Lec 1

What is Data Mining?

- extracting knowledge from ***large*** amount of data
  - cleaning, integration, selection, transformation, mining/processing, pattern evaluation, presentation

What kind of patterns can be mined/found by data mining techniques?

- Characterization and Discrimination
- Frequent patterns, Associations, and Correlations
- Classification and Prediction
- Cluster analysis
- Outlier analysis
- Evolution analysis

Give examples of each of the following:

- Characterization and Discrimination
  - Characteristics of customers who buy a certain kind of product
  - Customers who buy product A vs customers who buy another product B
- Frequent patterns, Associations, and Correlations
  - What kind of products do customers buy together?
  - If customer buys product A, what's the chance that he/she will buy product B as well
- Classification and Prediction
  - Classify the sales of various products into different classes
  - Predict the sales of the product
- Cluster analysis
  - Divide the data into different groups of similar items
  - No. of cluster are not known apriori
- Outlier analysis
  - Deviant data from the expected
- Evolution analysis
  - Time series analysis of data

Discrimination – only one attribute, Cluster – multiple attributes/features/parameters

Classification – past data - learn– model → testing on new data, Discrimination – existing data

Discuss whether or not each of the following activities is a data mining task.

Dividing the customers of a company according to their gender.

No. This is a simple database query.

Dividing the customers of a company according to their profitability.

No. This is an accounting calculation, followed by the application of a threshold. However, predicting the profitability of a new customer would be data mining.

Predicting the outcomes of tossing a (fair) pair of dice.

No. Since the die is fair, this is a probability calculation.

Predicting the future stock price of a company using historical records.

Yes. We would attempt to create a model that can predict the continuous value of the stock price. This is an example of the area of data mining known as predictive modelling.

Monitoring seismic waves for earthquake activities.

Yes. In this case, we would build a model of different types of seismic wave behavior associated with earthquake activities and raise an alarm when one of these different types of seismic activity was observed. This is an example of the area of data mining known as classification.

Lec 2

Data vs. Information

- **Data and information are interrelated.**
- **Data usually refers to raw data, or unprocessed data.**
- **Once the data is analyzed, it is considered as information.**
- **Information is "knowledge communicated or received concerning a particular fact or circumstance."**

Students Examination data

In examination data about obtained marks of different subjects for all students is collected.

Merit List

- merit is calculated on the basis of obtained marks of each candidate.
- Merit list is used to decide whether a candidate will get admission in the college or not.

Differences between data and information

- Data is used as input for the computer system. Information is the output of data.
- Data is unprocessed facts figures. Information is processed data.
- Data doesn't depend on Information. Information depends on data.
- Data doesn't carry a meaning. Information must carry a logical meaning.
- Data is the raw material. Information is the product.

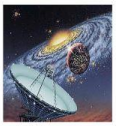
Why Mine Data? Commercial Viewpoint

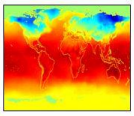
Lots of data is being collected and warehoused

- Web data, e-commerce
- purchases at department/ grocery stores
- Bank/Credit Card transactions

Computers have become cheaper and more powerful

Why Mine Data? Scientific Viewpoint



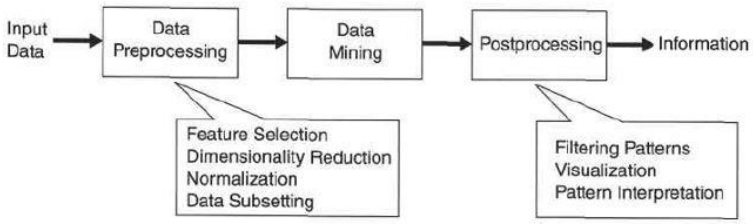


- Data collected and stored at enormous speeds (GB/hour)
  - remote sensors on a satellite
  - microarrays generating gene expression data
- Traditional techniques infeasible for raw data
- Data mining may help scientists
  - in classifying and segmenting data
  - in Hypothesis Formation

What is Data Mining?

Many Definitions

– **Non-trivial extraction of implicit, previously unknown and potentially useful information from data.**



What is (not) Data Mining?

- What is not Data Mining?
  - Look up phone number in phone directory.
  - Query a Web search engine for information about "Amazon"

- What is Data Mining?
  - Certain names are more prevalent.
  - Group together similar documents returned by search engine according to their context

Feature Selection – select relevant features – A, B, C, D, E → A, B, C (No effect of D and E)

Feature Reduction – Derive new features with all existing features – A, B, C, D, E → F, G

Data Sub setting – Training data for Train the model and Testing data to test the Model

What Kind of Data can be mined?

- **Flat files:** Flat files are simple data files in text or binary format with a structure known by the data mining algorithm to be applied.
- **Relational Databases**: Tables have columns and rows, where columns represent attributes and rows represent tuples.
- **Data Warehouses**: A data warehouse as a storehouse, is a repository of data collected from multiple data sources (often heterogeneous).

  A data warehouse gives the option to analyze data from different sources.

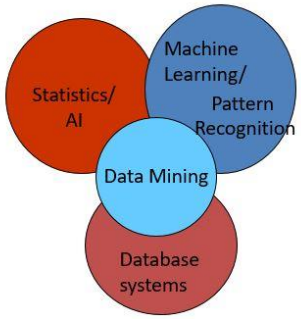
- **Transaction Databases**: A transaction database is a set of records representing transactions, each with a time stamp, an identifier and a set of items.

- **Multimedia Databases**: Multimedia databases include video, images, audio and text media.

- **Time-Series Databases**:Time-series databases contain time related data such stock market data or logged activities.

Origins of Data Mining

- **Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems**
- **Traditional Techniques may be unsuitable due to**
  - **High dimensionality of data**
  - **Heterogeneous, distributed nature of data**



Data Mining Tasks

- **Descriptive** Analytics: which use data aggregation and data mining to provide insight into the past and answer: "What has happened?"

- **Predictive** Analytics: which use statistical models and forecasts techniques to understand the future and answer: "What could happen?"

Descriptive – Based on descriptive characteristics (min, max, median, mean, sd etc.) find pattern

Predictive – Based on data prepare model – that will predict the data

- Classification [Predictive]
- Clustering [Descriptive]
- Association Rule Discovery [Descriptive]
- Regression [Predictive]

Classification – two different groups, Regression – to find a value – continues value

Classification: Definition

innovate achieve lead

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Data – input features/independent variable (discreet value) – output – variable (predict)

An Example

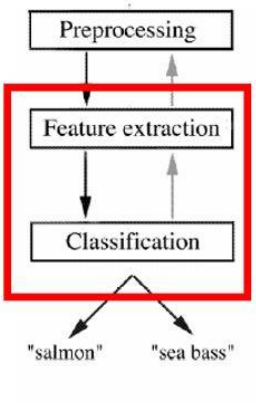
innovate achieve lead

- **from *Pattern Classification by* Duda & Hart & Stork – Second Edition, 2001)**
- A fish-packing plant wants to automate the process of sorting incoming fish according to category.
- As a pilot project, it is decided to try to separate sea bass from salmon using optical sensing

Features

Length
Lightness
Width
Position of mouth

- **Preprocessing:** Images of different fishes are isolated from one another and from background;
- **Feature extraction:** The information of a single fish is then sent to a feature extractor, that measure certain “features” or “properties”;
- **Classification:** The values of these features are passed to a classifier that evaluates the evidence presented, and build a model to discriminate between the two species

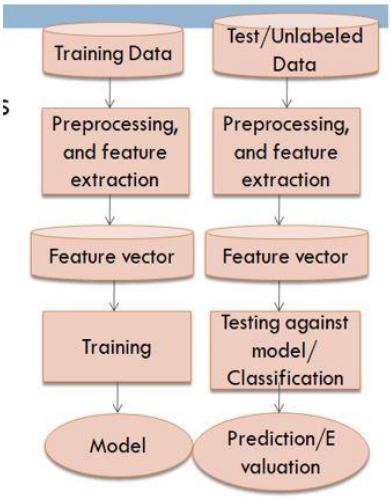


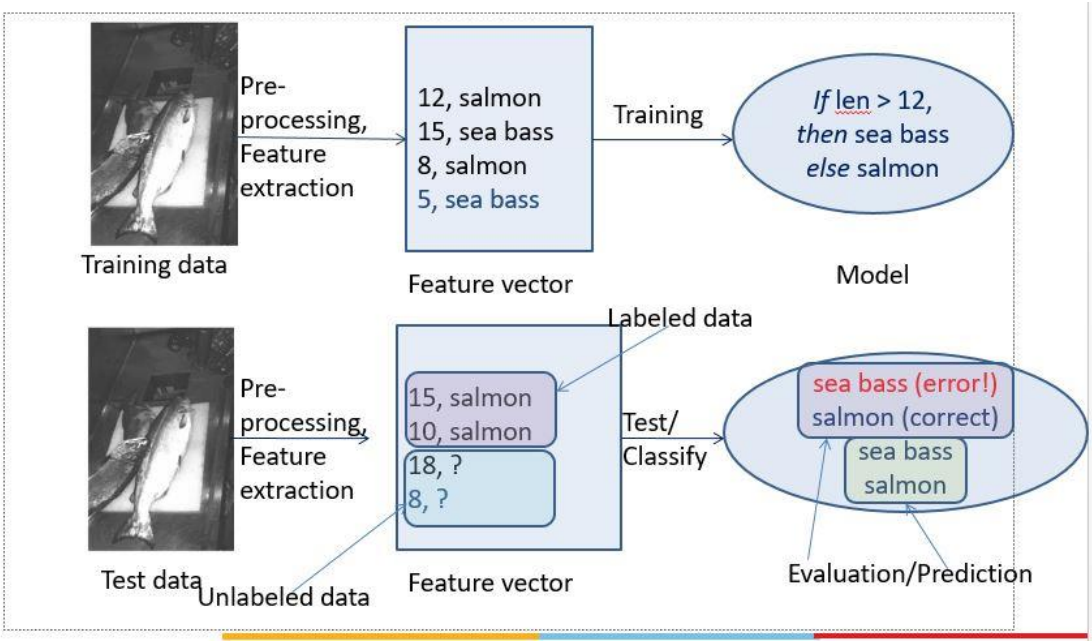
Pre Processing – How many features, then remove irrelevant features then classification algorithm

- Domain knowledge:
  - A sea bass is generally longer than a salmon
- Related feature*:* (or attribute)
  - Length
- Training the classifier:
  - Some examples are provided to the classifier in this form: <fish_length, fish_name>
  - These examples are called training examples
  - The classifier *learns* itself from the training examples, how to distinguish Salmon from Bass based on the *fish_length*

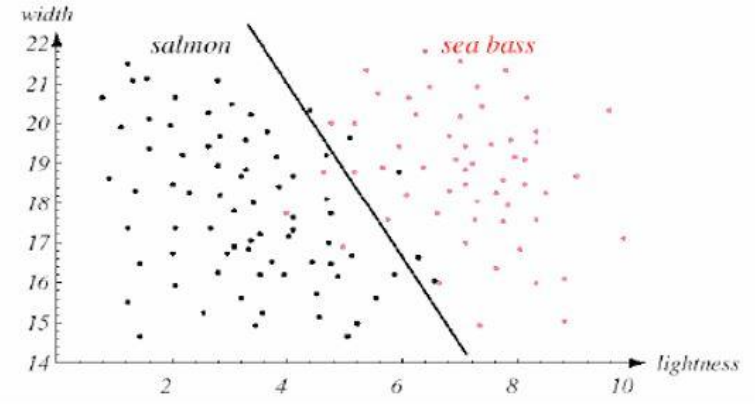
- Classification model (hypothesis):
  - The classifier generates a model from the training data to classify future examples (test examples)
  - An example of the model is a rule like this:
  - If *Length >= l* then sea bass* otherwise *salmon*
  - Here the value of *l** determined by the classifier
- Testing the model
  - Once we get a model out of the classifier, we may use the classifier to test future examples
  - The test data is provided in the form <fish_length>
  - The classifier outputs <fish_type> by checking *fish_length* against the model

So the overall classification process goes like this →





Classification and Prediction – Supervised, Clustering and Association rule – Unsupervised



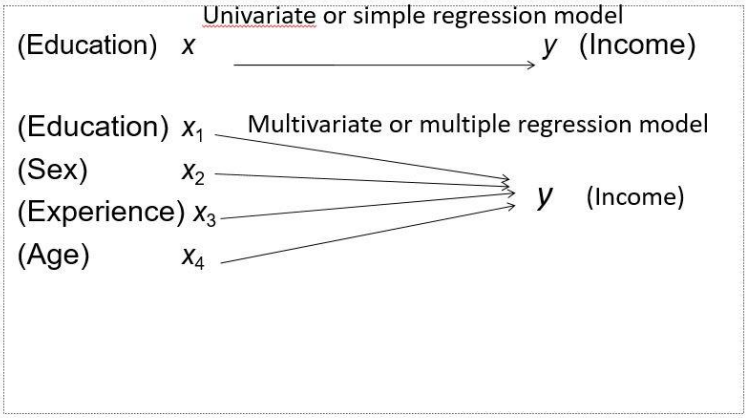
Decision rule: Classify the fish as a sea bass if its feature vector falls above the decision boundary shown, and as salmon otherwise

Lec 3 Data

Regression

- Predict a value of a given continuous valued variable based on the values of other variables, assuming a linear or nonlinear model of dependency.
- Greatly studied in statistics, neural network fields.
- Examples:
  - Predicting sales amounts of new product based on advertising expenditure.
  - Predicting wind velocities as a function of temperature, humidity, air pressure, etc.
  - Time series prediction of stock market.

Univariate and multivariate models



Classification – Dependent variable has some fix value

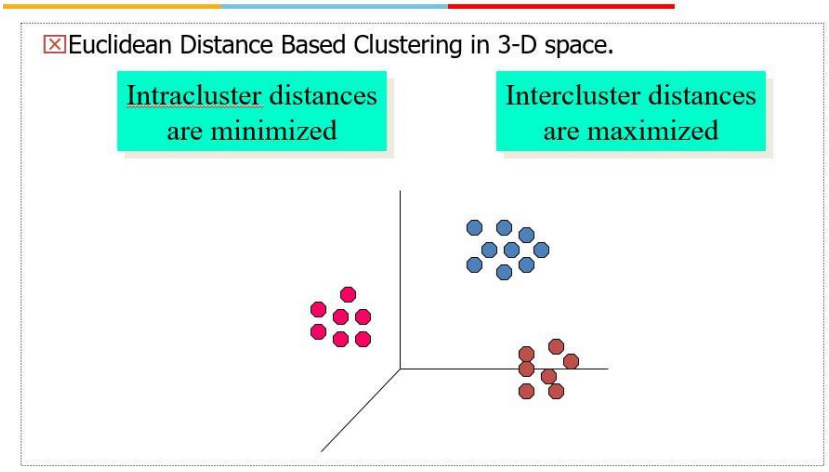
Clustering Definition

- Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that
  - Data points in one cluster are more similar to one another.
  - Data points in separate clusters are less similar to one another.
- Similarity Measures:
  - Euclidean Distance if attributes are continuous.
  - Other Problem-specific Measures.

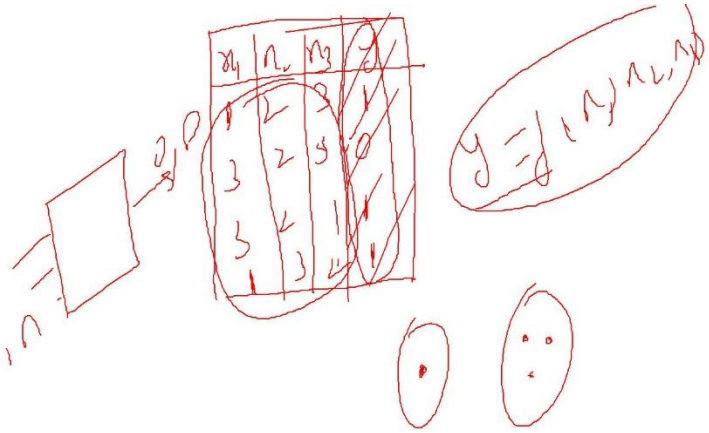
Based on characteristic of data – categorize data – Clustering (Unsupervised)

Minimize – Inter cluster distance, Maximize – Intra cluster distance

Clustering



Supervised Learning – output Is known → y = f(x), find pattern, past/historical data – predict future



When data is not labeled – use clustering, unsupervised is used to find past (what types groups)

Clustering: Application

Document Clustering:

- Goal: To find groups of documents that are similar to each other based on the important terms appearing in them.
- Approach: To identify frequently occurring terms in each document. Form a similarity measure based on the frequencies of different terms. Use it to cluster.
- Gain: Information Retrieval can utilize the clusters to relate a new document or search term to clustered documents.

- Clustering Points: 3204 Articles of TOI.
- Similarity Measure: How many words are common in these documents (after some word filtering).

Category	Total Articles	Correctly Placed
Financial	555	364
Foreign	341	260
National	273	36
Metro	943	746
Sports	738	573
Entertainment	354	278

Association Rule Discovery: Definition

- Given a set of records each of which contain some number of items from a given collection;
  - Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

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

Association Rule Discovery: Application 1

Marketing and Sales Promotion:

- Let the rule discovered be

  {Bagels, … } --> {Potato Chips}

- Potato Chips as consequent => Can be used to determine what should be done to boost its sales.
- Bagels in the antecedent => Can be used to see which products would be affected if the store discontinues selling bagels.
- Bagels in antecedent *and* Potato chips in consequent => Can be used to see what products should be sold with Bagels to promote sale of Potato chips!

- Supermarket shelf management.
  - Goal: To identify items that are bought together by sufficiently many customers.
  - Approach: Process the point-of-sale data collected with barcode scanners to find dependencies among items.
  - A classic rule --
    - If a customer buys bread, then he is very likely to buy butter.

Challenges of Data Mining

- Scalability
- Dimensionality
- Complex and Heterogeneous Data
- Data Ownership and Distribution

  (1) how to reduce the amount of communication needed to perform the distributed computation.

  (2) how to address data security issues.
- Non-traditional Analysis

What is Data?

- **Collection of data objects and their attributes**
- **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**

Attributes

Objects

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

Attribute Values

- **Attribute values are numbers or symbols assigned to an attribute**
- **Distinction between attributes and attribute values**
  - 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

There are different types of attributes

- Nominal: **Data** are neither measured nor ordered but subjects are merely allocated to distinct categories

  Examples: ID numbers, eye color, zip codes
- Ordinal: **Ordinal data** is a categorical where the variables have natural, ordered categories and the distances between the categories is not known.

  Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
- Interval: In **interval** measurement the distance between attributes does have meaning.

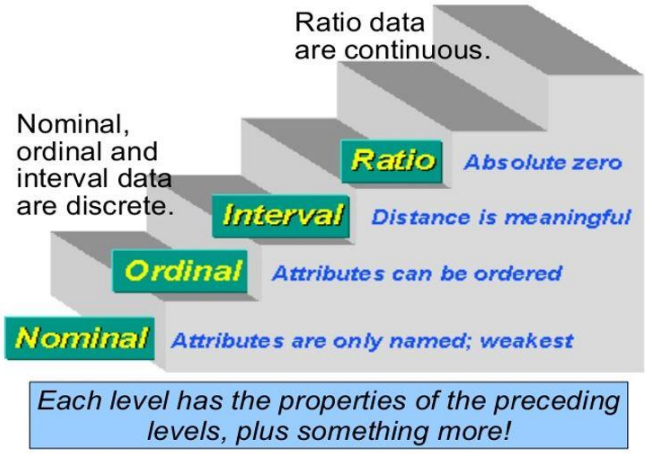
  Examples: calendar dates, temperatures in Celsius or Fahrenheit.
- Ratio:

  Examples: temperature in Kelvin, length, time, counts

The type of an attribute depends on which of the following properties it possesses:

- **Distinctness:** = ≠
- **Order:** < >
- **Addition:** + -
- **Multiplication:** * /

- **Nominal attribute**: distinctness
- **Ordinal attribute**: distinctness & order
- **Interval attribute**: distinctness, order & addition
- **Ratio attribute**: all 4 properties



Each level has the properties of the preceding levels, plus something more!

Attribute Type	Description	Examples	Operations
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, sex: {*male, female*}	mode, entropy, contingency correlation, χ^2 test
Ordinal	The values of an ordinal attribute provide enough information to order objects. (<, >)	hardness of minerals, {*good, better, best*}, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. (+, -)	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, *t* and *F* tests
Ratio	For ratio variables, both differences and ratios are meaningful. (*, /)	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

Discrete and Continuous Attributes

- **Discrete Attribute**
  - Has only a finite or countable 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.

Temperature – Low and High – Discrete value, Discrete (values is defined), continuous – defined

Types of data sets

innovate achieve lead

- **Record**
  - Data Matrix
  - Document Data
  - Transaction Data
- **Graph**
  - World Wide Web
  - Molecular Structures
- **Ordered**
  - Spatial Data
  - Temporal Data
  - Sequential Data
  - Genetic Sequence Data

Collection of similar data –called data sets, Ordered Data group – order matters

Record Data

innovate achieve lead

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

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

Data Matrix

innovate achieve lead

- 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	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Document Data

innovate achieve lead

- 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	play	ball	score	game	win	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

Transaction Data

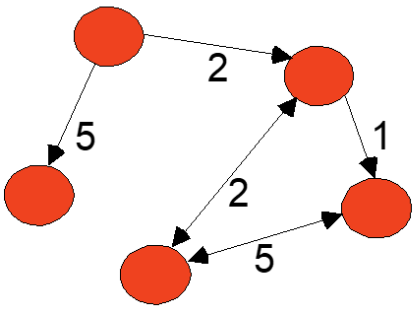
innovate achieve lead

- 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

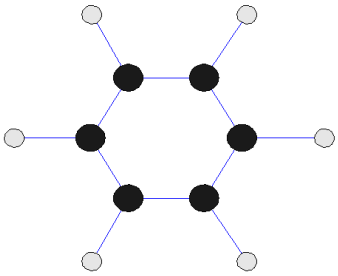
Graph Data

innovate achieve lead



Edges – relation between two Nodes, Node – attribute

Molecular Structures : Chemical Data



Ordered Data

Sequences of transactions

(A B) (D) (C E)
(B D) (C) (E)
(C D) (B) (A E)

An element of the sequence

Genomic sequence data

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

Data Quality

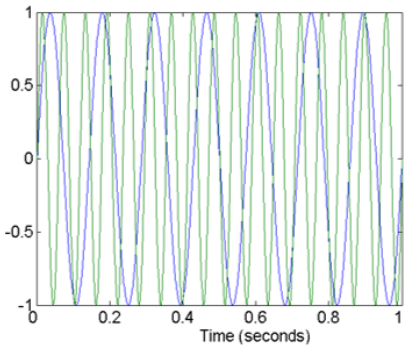
- What kinds of data quality problems?
- How can we detect problems with the data?
- What can we do about these problems?

Examples of data quality problems:

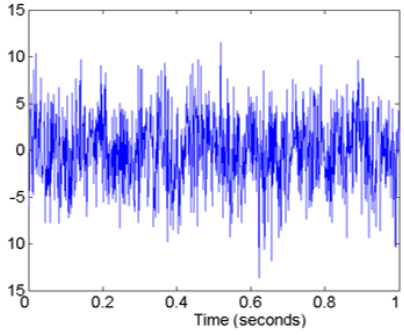
- Noise and outliers
- missing values
- duplicate data

Noise

- **Noise refers to modification of original values**
  - Examples: distortion of a person's voice when talking on a poor phone and "snow" on television screen

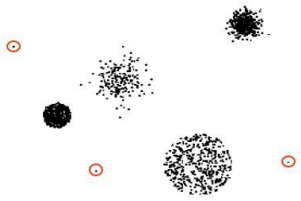


Two Sine Waves



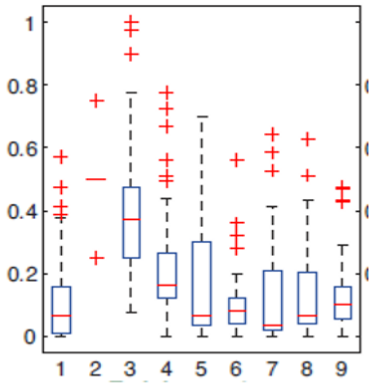
Two Sine Waves + Noise

- **Outliers are data objects with characteristics that are considerably different than most of the other data objects in the data set**



- Effectiveness of outliers is examined by using the following equation:

$$e_i = \begin{cases} \text{if } |y_{ji} - \hat{y}_j| > 3 * \sigma & \text{for Effective outliers} \\ \text{if } |y_{ji} - \hat{y}_j| <= 3 * \sigma & \text{for Non Effective outliers} \end{cases}$$



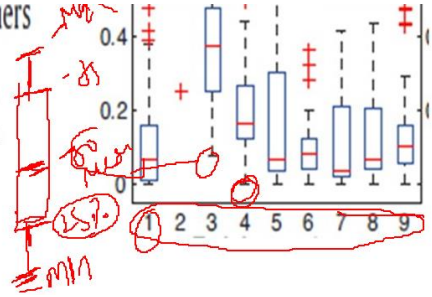
Standard deviation(σ): how much the members of a group differ from the mean value for the group=

$$\sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N-1}}.$$

$\text{if } |y_{ji} - y_j| <= 3 * \sigma$ for Non Effective outliers

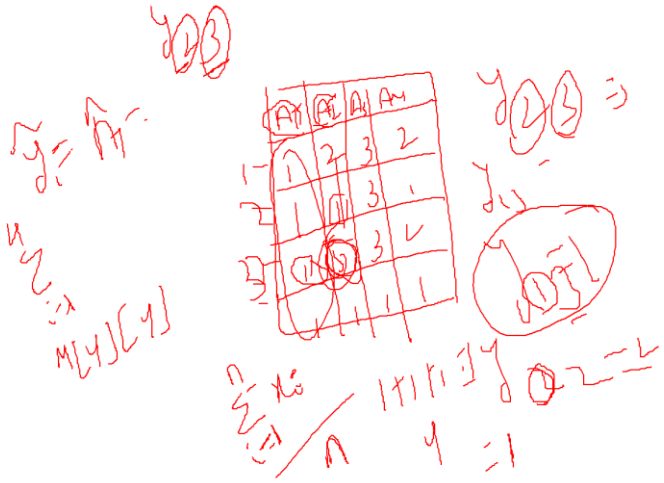
Standard deviation(σ): how much the members of a group differ from the mean value for the group=

$$\sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N-1}}.$$



Effective outliers – which will affect the modal's performance

Y_{ji} – Object i, j^{th} feature, yi^ - value of Object i



Lec 4 Preprocessing

Missing Values

- **Reasons for missing values**
  - Information is not collected
    (e.g., people decline to give their age and weight)
  - Attributes may not be applicable to all cases
    (e.g., annual income is not applicable to children)
- **Handling missing values**
  - Eliminate Data Objects
  - Estimate Missing Values
  - Ignore the Missing Value During Analysis
  - Replace with all possible values (weighted by their probabilities)

Duplicate Data

- **Data set may include data objects that are duplicates, or almost duplicates of one another**

Examples:

- Same person with multiple email addresses

Data cleaning

- Process of dealing with duplicate data issues

```
xmean=mean(x)

sd1=sd(x)

for i=1:length(x)

        if mod(x(i)-xmean)>3*sd1

                outef(i)=1;
```

```
        else outef(i)=0;

        end

end
```

Data Preprocessing

- Aggregation
- Sampling
- Dimensionality Reduction
- Feature subset selection
- Feature creation
- Discretization and Binarization
- Attribute Transformation

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

Transaction ID	Item	Store Location	Date	Price	...
⋮	⋮	⋮	⋮	⋮	
101123	Watch	Chicago	09/06/04	$25.99	...
101123	Battery	Chicago	09/06/04	$5.99	...
101124	Shoes	Minneapolis	09/06/04	$75.00	...
⋮	⋮	⋮	⋮	⋮	

Reducing the possible values for date from 365 days to 12 months.
This type of aggregation is commonly used in Online Analytical Processing (OLAP).

Arithmetic mean:

$$\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}$$

Standard deviation: how much the members of a group differ from the mean value for the group.

$$s_x = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}}$$

n = The number of data points
$\bar{x}$ = The mean of the x_i
x_i = Each of the values of the data

Sampling

- **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.**

- **Sampling is used in data mining because processing the entire set of data of interest is too expensive or time consuming.**

Types of Sampling

- **Simple Random Sampling**
  - **There is an equal probability of selecting any particular item.**
- **Sampling without replacement**
  - **As each item is selected, it is removed from the population.**
- **Sampling with replacement**
  - **Objects are not removed from the population as they are selected for the sample.**
    - **In sampling with replacement, the same object can be picked up more than once**
- **Stratified sampling**
  - **Split the data into several partitions; then draw random samples from each partition**

Stratified Sampling – generally used, Sampling – only reducing objects – not attr. like aggregation

8000 points **2000 Points** **500 Points**

Dimensionality Reduction

- **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
  - Others: supervised and non-linear techniques

Student name, Student id, Student Branch, Student Address, Student CGPI, Student DOB

How many students – how many students from CSE department – Only Branch is relevant

Feature Subset Selection

innovate achieve lead

- **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:**
  - **Brute-force approaches:**
    - **Try all possible feature subsets as input to data mining algorithm**
  - **Filter approaches:**
    - **Features are selected before data mining algorithm is run**
  - **Wrapper approaches:**
    - **Use the data mining algorithm as a black box to find best subset of attributes**

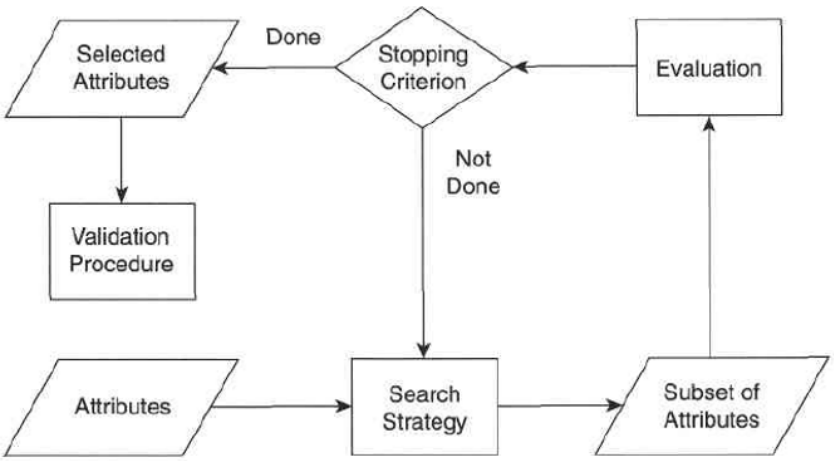
Brute force – All subsets - 2^n-1 – model based on each subset – check performance of each model

Filter approach – without using any learning algorithm select features – then use algorithm

Wrapper – searching techniques (using learning algo.) (Too many features – brute force won't work)

Feature Subset Selection

innovate achieve lead



Filter approaches

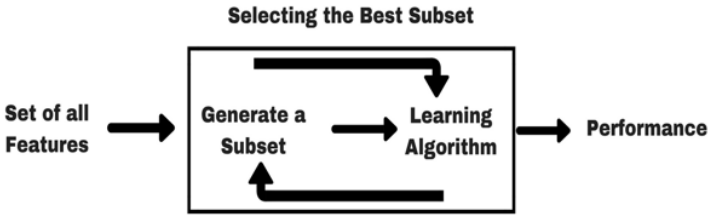
Pearson's Correlation: It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1. Pearson's correlation is given as:

$$r = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sqrt{\sum (x-\bar{x})^2 \sum (y-\bar{y})^2}}$$

Correlation(A,B) → 0.8 – highly related – if A's value is increased B's value will also increase – so no need to consider both values -> i.e. Tax calculated from salary (only consider salary)

Wrapper approaches

- **we try to use a subset of features and train a model using them.**



Searching strategy to generate subset. Wrapper – gives better result commonly used

Sequential Forward Selection (SFS)

Sequential Forward Selection (SFS)

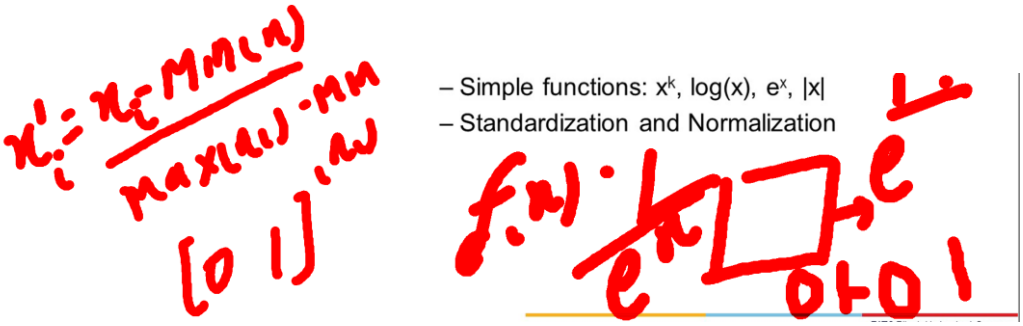
- Start with the empty set, *X=0*
- Repeatedly add the most significant feature with respect to *X*

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:**
  - **Mapping Data to New Space**
  - **Feature Construction**

Attribute Transformation

- A function that maps the entire set of values of a given attribute to a new set of replacement values such that each old value can be identified with one of the new values
  - Simple functions: x^k, log(x), e^x, |x|
  - Standardization and Normalization



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/Dissimilarity for Simple Attributes

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{\lvert p-q \rvert}{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values)	$s = 1 - \frac{\lvert p-q \rvert}{n-1}$
Interval or Ratio	$d = \lvert p - q \rvert$	$s = -d$, $s = \frac{1}{1+d}$ or $s = 1 - \frac{d - min\_d}{max\_d - min\_d}$

Table 5.1. Similarity and dissimilarity for simple attributes

- Consider an attribute that measures the quality of a product on the scale **{poor, fair, OK, good, wonderful}**

P1: wonderful

P2: fair

- The values of the ordinal attribute are often mapped to successive integers, beginning at 0 or 1, e.g., {poor:0, fair:1, OK:2, good:3, wonderful:4).

Ordinal Attributes - P1, P2 → d = 0.75 (Highly dissimilar) , s = 0.25 – (Low Similar)

Euclidean Distance

innovate achieve lead

- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

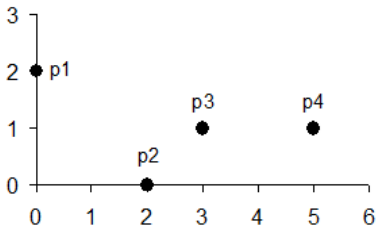
- Where *n* is the number of dimensions (attributes) and p_k and q_k are, respectively, the k^{th} attributes (components) or data objects *p* and *q*.

Interval/Ratio – Euclidean Distance – between object – more distance – objects - dissimilarity matrix

n – Attributes - P_k – value of kth attribute of Object P

Euclidean Distance

innovate achieve lead



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

P1 and P2 are more similar than P1 and P3

Common Properties of a Distance

- Distances, such as the Euclidean distance, have some well known properties.

1. $d(p, q) \geq 0$ for all p and q and $d(p, q) = 0$ only if $p = q$. (Positive definiteness)
2. $d(p, q) = d(q, p)$ for all p and q. (Symmetry)
3. $d(p, r) \leq d(p, q) + d(q, r)$ for all points p, q, and r. (Triangle Inequality)

- where $d(p, q)$ is the distance (dissimilarity) between points (data objects), p and q.
- Measures that satisfy all three properties are known as **metrics**.

Non-metric Dissimilarities:

A: { I ,2,3,4} and B : {2,3,4}

A- B: {1}

B - A : φ

Set difference – Non – metric dissimilarities – metric unit of measurement

Common Properties of a Similarity

Similarities, also have some well known properties.

1. $s(p, q) = 1$ (or maximum similarity) only if $p = q$.
2. $s(p, q) = s(q, p)$ for all p and q. (Symmetry)

where $s(p, q)$ is the similarity between points (data objects), p and q.

Similarity Between Binary Vectors

- Common situation is that objects, p and q, have only binary attributes
- Compute similarities using the following quantities

M_{01} = the number of attributes where p was 0 and q was 1
M_{10} = the number of attributes where p was 1 and q was 0
M_{00} = the number of attributes where p was 0 and q was 0
M_{11} = the number of attributes where p was 1 and q was 1

- Simple Matching and Jaccard Coefficients

SMC = number of matches / number of attributes
$= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

J = number of 11 matches / number of not-both-zero attributes values
$= (M_{11}) / (M_{01} + M_{10} + M_{11})$

SMC versus Jaccard: Example

p = 1 0 0 0 0 0 0 0 0 0
q = 0 0 0 0 0 0 1 0 0 1

M_{01} = 2 (the number of attributes where p was 0 and q was 1)
M_{10} = 1 (the number of attributes where p was 1 and q was 0)
M_{00} = 7 (the number of attributes where p was 0 and q was 0)
M_{11} = 0 (the number of attributes where p was 1 and q was 1)

SMC = $(M_{11} + M_{00})/(M_{01} + M_{10} + M_{11} + M_{00})$ = (0+7) / (2+1+0+7) = 0.7

J = (M_{11}) / $(M_{01} + M_{10} + M_{11})$ = 0 / (2 + 1 + 0) = 0

Hamming Distance

- **The Hamming Distance is a number used to denote the difference between two Binary Vectors.**

Steps to Calculate Hamming Distance

1. Ensure the two strings are of equal length. The Hamming distance can only be calculated between two strings of equal length.
2. Compare the first two bits in each string. If they are the same, record a "0" for that bit. If they are different, record a "1" for that bit.
3. Compare each bit in succession and record either "1" or "0" as appropriate.
4. Add all the ones and zeros in the record together to obtain the Hamming distance.

Cosine Similarity

If d_1 and d_2 are two document vectors, then

$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2||$,

where • indicates vector dot product and || d || is the length of vector d.

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

Example:

d_1 = 3 2 0 5 0 0 0 2 0 0
d_2 = 1 0 0 0 0 0 0 1 0 2

$d_1 \bullet d_2$ = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

$||d_1||$ = $(3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5}$ = $(42)^{0.5}$ = 6.481

$||d_2||$ = $(1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5}$ = $(6)^{0.5}$ = 2.245

$\cos(d_1, d_2)$ = .3150

Extended Jaccard Coefficient

- The extended Jaccard coefficient can be used for document data.

$$T(p,q) = \frac{p \bullet q}{\|p\|^2 + \|q\|^2 - p \bullet q},$$

Pearson's Correlation

- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, p and q, and then take their dot product

$$p'_k = (p_k - mean(p)) / std(p)$$

$$q'_k = (q_k - mean(q)) / std(q)$$

$$correlation(p,q) = p' \bullet q'$$

Linear relation between two objects / attributes

$$\text{corr}(\mathbf{x},\mathbf{y}) = \frac{\text{covariance}(\mathbf{x},\mathbf{y})}{\text{standard\_deviation}(\mathbf{x}) * \text{standard\_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x\ s_y}$$

$$\text{covariance}(\mathbf{x},\mathbf{y}) = s_{xy} = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \overline{x})(y_k - \overline{y})$$

$$\text{standard\_deviation}(\mathbf{x}) = s_x = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(x_k - \overline{x})^2}$$

$$\text{standard\_deviation}(\mathbf{y}) = s_y = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}(y_k - \overline{y})^2}$$

Corr(A,B) -> Negative -> A value increases -> B values – decreases

Corr(A,B) -> Positive -> A value increases -> B values – increases

Corr(A,B) -> 0.8 → 80% cases -> A value increases -> B values – increases

Visually Evaluating Correlation

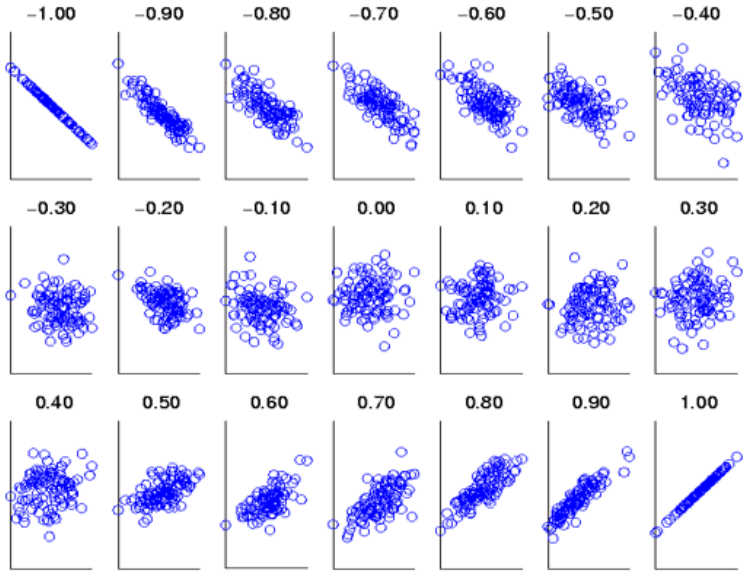


Figure 5.11. Scatter plots illustrating correlations from -1 to 1.

Perfect Correlation.

- **Correlation is always in the range -1 to 1.**
- **A correlation of 1 (-1) means that x and y have a perfect positive (negative) linear relationship.**

x: (-3, 6, 0, 3, -6)
y: (1, -2, 0,-7, 2)

x: (3 , 6 , 0 , 3 , 6)
y : (1 , 2 , 0 , 1 , 2)

General Approach for Combining Similarities

- Sometimes attributes are of many different types, but an overall similarity is needed.

1. For the k^{th} attribute, compute a similarity, s_k, in the range $[0, 1]$.
2. Define an indicator variable, δ_k, for the k_{th} attribute as follows:

$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have a value of 0, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$

3. Compute the overall similarity between the two objects using the following formula:

$$similarity(p, q) = \frac{\sum_{k=1}^{n} \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

Using Weights to Combine Similarities

- **May not want to treat all attributes the same.**
  - **Use weights w_k which are between 0 and 1 and sum to 1.**

$$similarity(p,q) = \frac{\sum_{k=1}^{n} w_k \delta_k s_k}{\sum_{k=1}^{n} \delta_k}$$

Lec 5 Data Mining: Exploring Data

What is data exploration?

- **A preliminary exploration of the data to better understand its characteristics.**
- Key motivations of data exploration include
  - Helping to select the right tool for preprocessing or analysis
  - Making use of humans' abilities to recognize patterns
    - People can recognize patterns not captured by data analysis tools

Before apply any Data Mining algorithm to understand characteristics of Data – we have to explore data – Storing the data in rows – attributes are characteristics of objects

Techniques Used In Data Exploration

- **In Exploratory Data Analysis (EDA), as originally defined by Tukey**
  - The focus was on visualization
  - Clustering and anomaly detection were viewed as exploratory techniques
  - In data mining, clustering and anomaly detection are major areas of interest, and not thought of as just exploratory
- **In our discussion of data exploration, we focus on**
  - Summary statistics
  - Visualization
  - Online Analytical Processing (OLAP)

OLAP – new Data Mining technique to handle multidimensional data

- Many of the exploratory data techniques are illustrated with the Iris Plant data set.
  - Can be obtained from the UCI Machine Learning Repository http://www.ics.uci.edu/~mlearn/MLRepository.html
  - From the statistician Douglas Fisher
  - Three flower types (classes):
    - Setosa
    - Virginica
    - Versicolour

- **Each flower is characterized by five attributes:**

1. sepal length in centimeters
2. sepal width in centimeters
3. petal length in centimeters
4. petal width in centimeters
5. class (Setosa, Versicolour, Virginica)

Summary Statistics

- Summary statistics are numbers that summarize properties of the data
  - Summarized properties include frequency, location and spread
    - Examples: location - mean
      spread - standard deviation
  - Most summary statistics can be calculated in a single pass through the data

Frequency and Mode

- **The frequency of an attribute value is the percentage of time the value occurs in the data set**
  - **For example, given the attribute 'gender' and a representative population of people, the gender 'female' occurs about 50% of the time.**
- **The mode of a an attribute is the most frequent attribute value**
- **The notions of frequency and mode are typically used with categorical data**

Not for continuous data – only for categorical / discrete data

Find the mode of the following set of scores.

14 11 15 9 11 15 11 7 13 12

Weight (kg)	45	50	55	60	65	70	75	80
Bags of rice (Frequency)	8	11	7	10	9	10	12	8

Mode – 75 (most frequent – 12)

Percentiles

innovate achieve lead

- For continuous data, the notion of a percentile is more useful.
- Given an continuous attribute *x* and a number *p* between 0 and 100, the *p*th percentile is a value of x such that *p*% of the observed values of x are less than x_p
- For instance, the 50th percentile is the value $x_{50\%}$ such that 50% of all values of x are less than x_{50}

1. Order all the values in the data set from smallest to largest.
2. Multiply *k* percent by the total number of values, *n*. This number is called the index.
3. If the index obtained in Step 2 is not a whole number, round it up to the nearest whole number and go to Step 4a. If the index obtained in Step 2 is a whole number, go to Step 4b.

Step 4a.Count the values in your data set from left to right until you reach the number indicated by Step 3. The corresponding value in your data set is the k^{th} percentile.

Step 4b. Count the values in your data set from left to right until you reach the number indicated by Step 2.

43, 54, 56, 61, 62, 66, 68, 69, 69, 70, 71, 72, 77, 78, 79, 85, 87, 88, 89, 93, 95, 96, 98, 99, 99

Find the 90th percentile.

Find the 60th percentile.

X90 → n = 25, Index → 22.5 ~ 23 → 98

X60 → n = 25, Index → 15 → 79

Measures of Location: Mean and Median

- **The mean is the most common measure of the location of a set of points.**
- **However, the mean is very sensitive to outliers.**
- **Thus, the median or a trimmed mean is also commonly used.**

$$\text{mean}(x) = \overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\text{median}(x) = \begin{cases} x_{(r+1)} & \text{if } m \text{ is odd, i.e., } m = 2r+1 \\ \frac{1}{2}(x_{(r)} + x_{(r+1)}) & \text{if } m \text{ is even, i.e., } m = 2r \end{cases}$$

Find the mean, median, and range for the following list of values:

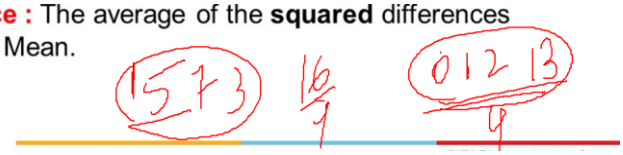
13, 18, 13, 14, 13, 16, 14, 21, 13

The "range" of a list a numbers is just the difference between the largest and smallest values.

Mean – 15, Median – 14, Range – 8, Mode – 13

Total number of samples are odd then 50th Percentile is Median value

- **Variance :** The average of the **squared** differences from the Mean.



Variance - Mean is same but spread is more in set B(0, 1, 2, 13) while other set is dense (1 to 7)

Measures of Spread: Range and Variance

- Range is the difference between the max and min
- The variance or standard deviation is the most common measure of the spread of a set of points.

$$\text{variance}(x) = s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x})^2$$

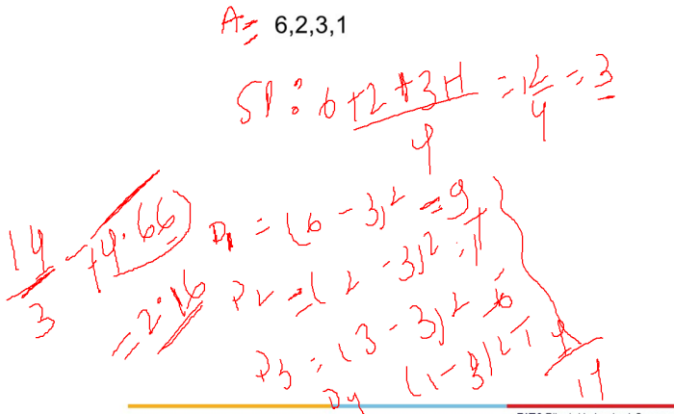
- **Variance :** The average of the **squared** differences from the Mean.

Standard deviation

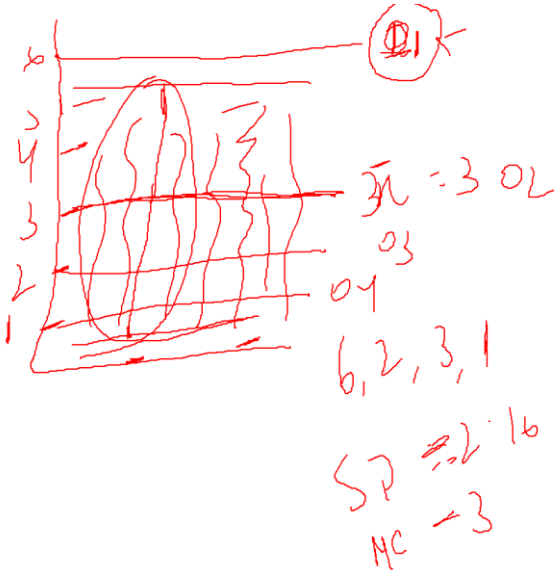
Step 1: Find the mean.
Step 2: For each data point, find the square of its distance to the mean.
Step 3: Sum the values from Step 2.
Step 4: Divide by the number of data points.
Step 5: Take the square root.
Note: If you're dealing with a sample,
Step 4: Divide by the (number of data points-1).

example for calculating standard deviation

6,2,3,1



A (1, 5, 7, 3) – sd → 2.58, B(0 ,1 , 2 ,13) – sd → 6.055 (more spread)



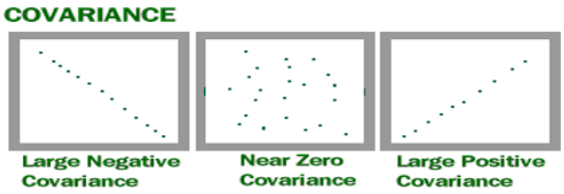
Upper bound – 5.16 (value more than this – extra-large), Lower – 0.84 (lower than extra small)

Variance →Most of the values are between 0.84 and 5.16

Covariance

innovate achieve lead

- Covariance is a measure of how much two random variables vary together. It's similar to variance, but where variance tells you how a *single* variable varies, **co** variance tells you how **two** variables vary together.



Calculate covariance for the following data set:

x: 2.1, 2.5, 3.6, 4.0

y: 8, 10, 12, 14

$$cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Answer – 2.266

Covariance may be or may not be in range of [-1 , 1]

Covariance(x, y) == Covariance(y, x)

Correlation Coefficient

innovate achieve lead

- **Correlation coefficient formulas are used to find how strong a relationship is between data. The formulas return a value between -1 and 1, where:**
- 1 indicates a strong positive relationship.
- -1 indicates a strong negative relationship.
- A result of zero indicates no relationship at all.

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$

Covariance and correlation both are similar (used to find relation between objects) but two different ways to measure – formulas, Correlation(x , y) == Correlation(y , x)

SUBJECT	AGE X	GLUCOSE LEVEL Y
1	43	99
2	21	65
3	25	79
4	42	75
5	57	87
6	59	81

x – 41.66, y – 81 , c – (+0.79)

When Age is increases – 80% chance that Glucose level will increase

$$r = \frac{\sum (x-\bar{x})(y-\bar{y})}{\sqrt{\sum (x-\bar{x})^2 \sum (y-\bar{y})^2}}$$

Visualization

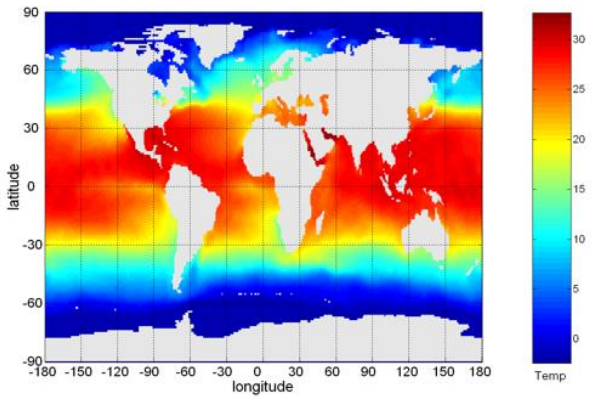
innovate achieve lead

- Visualization is the conversion of data into a visual or tabular format so that the characteristics of the data and the relationships among data items or attributes can be analyzed or reported.
- Visualization of data is one of the most powerful and appealing techniques for data exploration.
  - Humans have a well developed ability to analyze large amounts of information that is presented visually
  - Can detect general patterns and trends
  - Can detect outliers and unusual patterns

Sea Surface Temperature

innovate achieve lead

- The following shows the Sea Surface Temperature (SST) for July 1982
  - Tens of thousands of data points are summarized in a single figure



Representation

- In the mapping of information to a visual format
- Data objects, their attributes, and the relationships among data objects are translated into graphical elements such as **points, lines, shapes, and colors**.
  - Objects are often represented as points
  - Their attribute values can be represented as the position of the points or the characteristics of the points, e.g., color, size, and shape
  - If position is used, then the relationships of points, i.e., whether they form groups or a point is an outlier, is easily perceived.

Arrangement

- Arrangement Is the placement of visual elements within a display
- Can make a large difference in how easy it is to understand the data

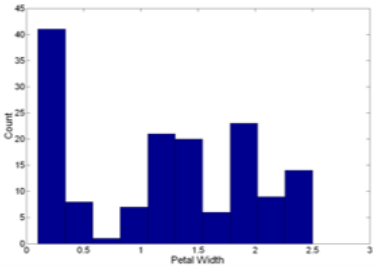
	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	1	0	1	0	0	1
5	0	1	0	1	1	0
6	1	0	1	0	0	1
7	0	1	0	1	1	0
8	1	0	1	0	0	1
9	0	1	0	1	1	0

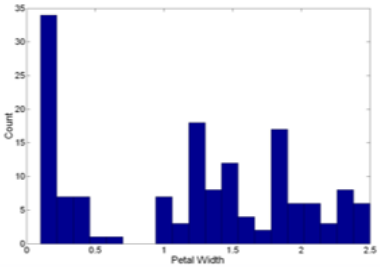
	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

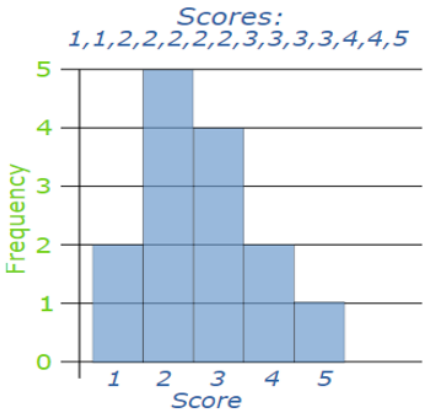
Visualization Techniques: Histograms

Histogram

- Usually shows the distribution of values of a single variable
- Divide the values into bins and show a bar plot of the number of objects in each bin.
- The height of each bar indicates the number of objects
- Shape of histogram depends on the number of bins





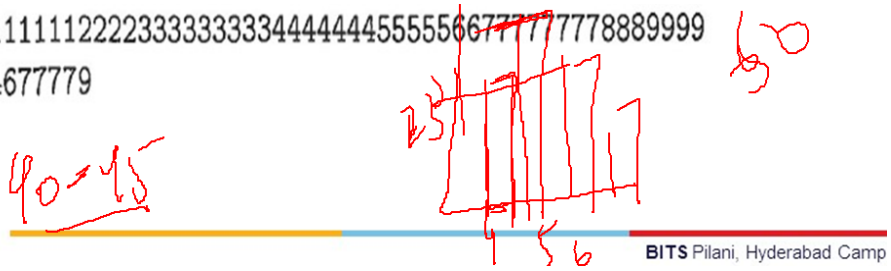


43 44 44 44 45 46 46 46 46 47 47 48 48 48 48 48 49 49 49 49 49 49 50
50 50 50 50 50 50 50 50 50 51 51 51 51 51 51 51 51 51 52 52 52 52 53
54 54 54 54 54 54 55 55 55 55 55 55 55 56 56 56 56 56 56 57 57 57 57
57 57 57 57 58 58 58 58 58 58 58 59 59 59 60 60 60 60 60 60 61 61 61
61 61 61 62 62 62 62 63 63 63 63 63 63 63 63 63 64 64 64 64 64 64 64
65 65 65 65 65 66 66 67 67 67 67 67 67 67 67 68 68 68 69 69 69 69 70
71 72 72 72 73 74 76 77 77 77 77 79

4 : 3444456666778888899999
5 : 00000000001111111112222344444455555556666667777777788888899999
6 : 000000111111222233333333344444445555566777777778889999
7 : 0122234677779

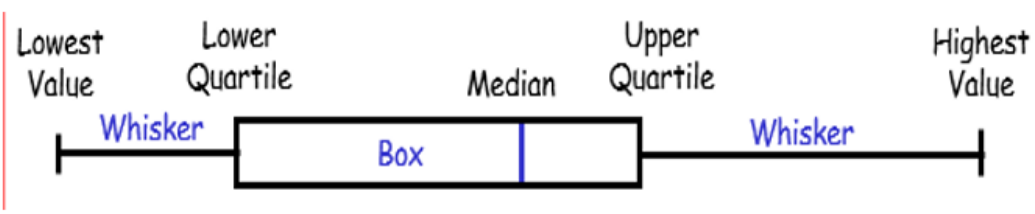
No of distinct elements – huge → Divide into bins/range → Bar chart

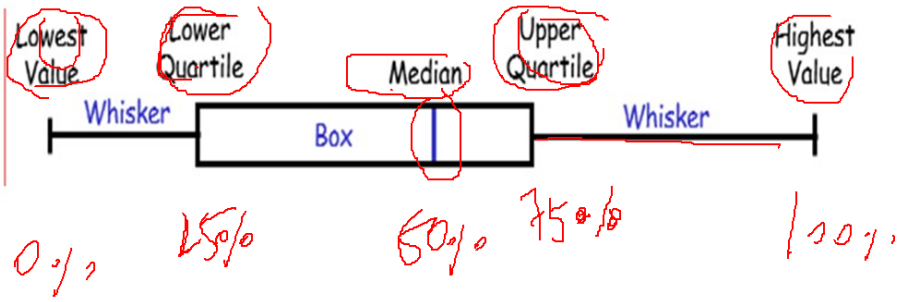
6 : 000000111111222233333333344444445555566777777778889999
7 : 0122234677779



Visualization Techniques: Box Plots

Box Plots
- Invented by J. Tukey
- Another way of displaying the distribution of data
- Following figure shows the basic part of a box plot





Finding the Median of Odd Numbered Data Sets

- Once the pieces of data (numbers) are arranged in order from least to greatest, then the middle number of the set is the median
  - [3, 4, 4, 5, 8, 8, 9, 10,11]
- The median for this set of data = 8
- The median splits the data set in half.
  - [3, 4, 4, 5,] 8, [8, 9, 10,11]
- Find the upper and lower quartiles as well as the upper and lower extremes.

Lower Quartile

- The lower quartile is the median of the bottom half of the data (to the left of the median).

[3, 4, 4, 5,] 8, [8, 9, 10,11]

4 + 4 = 8 8 divided by 2 = 4

The lower quartile for this set of data = **4**

Upper Quartile

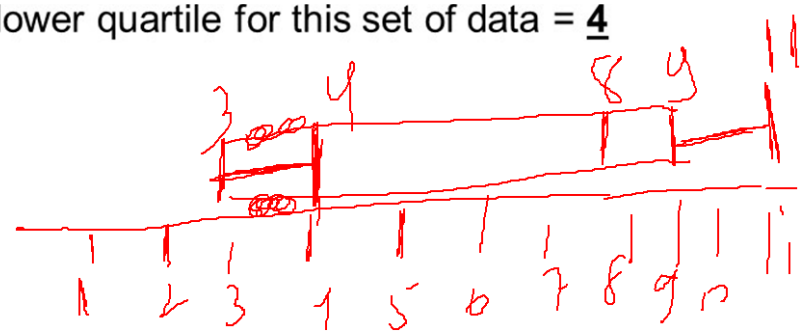
- The upper quartile is the median of the top half of the data (to the right of the median).
  - [3, 4, 4, 5,] 8, [8, 9, 10,11]
  - 9 + 10 = 19; 19 divided by 2 = 9.5
- The upper quartile for this data set = **9.5**

Interquartile Range

- To find the interquartile range, subtract the lower quartile from the upper quartile.
  - [3, 4,] **4** [4, 5,] 8, [8, 9] **9.5** [10,11]
  - Upper Quartile – Lower Quartile = _____
    - 9.5 – 4 = 5.5
  - The interquartile range for this data = 5.5

Higher Whisker = Highest value – Upper Quartile, Lower = Lowest – Lower Quartile

The lower quartile for this set of data = **4**



Lower Extreme and Upper Extreme

- The lower extreme is the lowest number in the data set.
  - [3, 4,] **4** [4, 5,] 8, [8, 9] **9.5** [10,11]
- The lower extreme for this data set = 3
- The upper extreme is the highest number in the data set.
  - [3, 4,] **4** [4, 5,] 8, [8, 9] **9.5** [10,11]
- The upper extreme for this data set = 11

5 Number Summary

[3, 4, 4, 5, 8, 8, 9, 10,11]

Median = 8

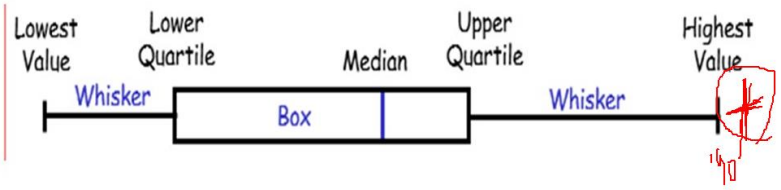
Lower Quartile = 4

Upper Quartile = 9.5

Lower Extreme = 3

Upper Extreme = 11

Lec 6 Data Mining: Exploring Data



In case of outlier detection technique >90% → Outlier – not all outliers are effective

Interquartile Range

innovate achieve lead

- To find the interquartile range, subtract the lower quartile from the upper quartile.
  - [3, 4,] **4** [4, 5,] 8, [8, 9] **9.5** [10,11]
  - Upper Quartile – Lower Quartile = _____
    - 9.5 – 4 = 5.5
  - The interquartile range for this data = 5.5

Criteria for Outliers

innovate achieve lead

- The following is criteria to identify outliers in a data set.
- The inter-quartile range (IQR) is the value of Q3 – Q1
- *x* is an outlier if of one the following inequalities is true:
  - $x > Q_3 + 1.5 \times IQR$
  - $x < Q_1 - 1.5 \times IQR$
- Does our data have any outliers?

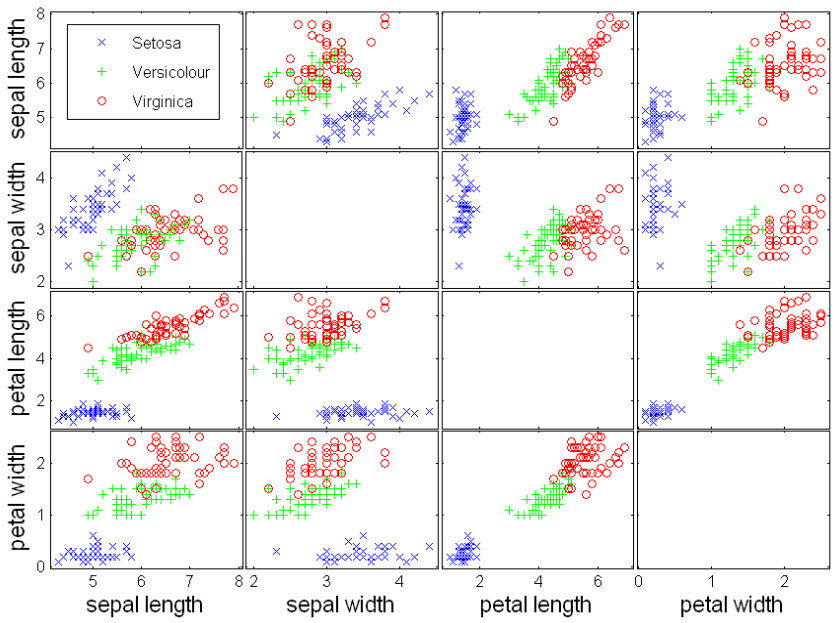
X > 9+(1.5*5.5) = 17.75 and x < 4 – (1.5*5.5) = -4.25, x > 17.75 & x < -4.25 → Outlier

Visualization Techniques: Scatter Plots

innovate achieve lead

Scatter plots

- Attributes values determine the position
- Two-dimensional scatter plots most common, but can have three-dimensional scatter plots
- Often additional attributes can be displayed by using the size, shape, and color of the markers that represent the objects
- It is useful to have arrays of scatter plots can compactly summarize the relationships of several pairs of attributes

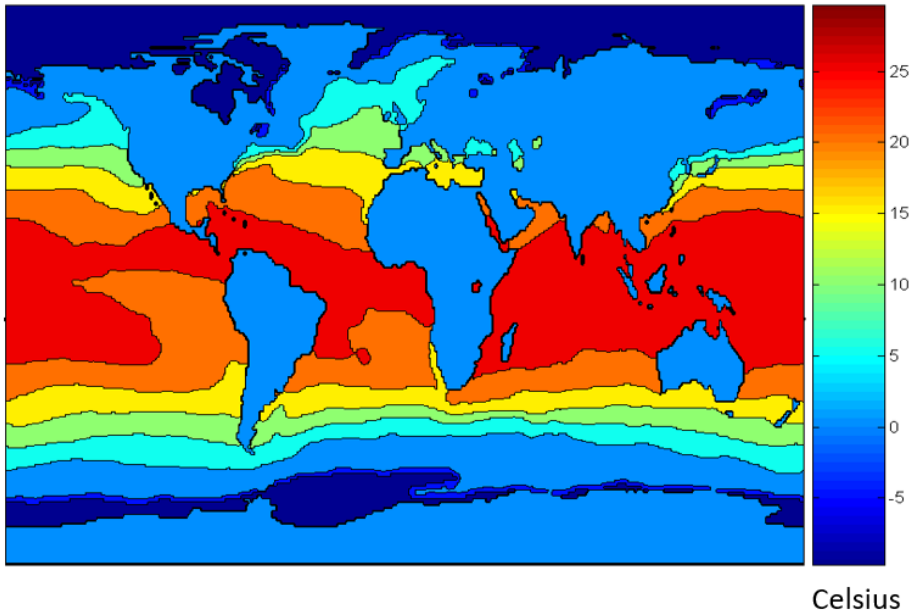


Visualization Techniques: Contour Plots

innovate achieve lead

Contour plots

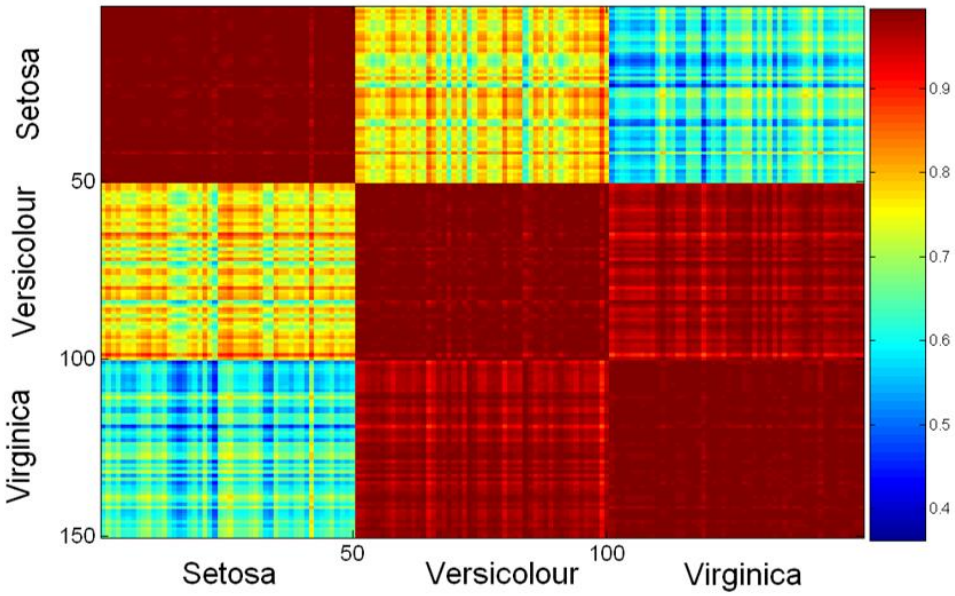
- Useful when a continuous attribute is measured on a spatial grid
- They partition the plane into regions of similar values
- The contour lines that form the boundaries of these regions connect points with equal values
- The most common example is contour maps of elevation
- Can also display temperature, rainfall, air pressure, etc.
  - An example for Sea Surface Temperature (SST) is provided on the next slide



Visualization Techniques: Matrix Plots

Matrix plots

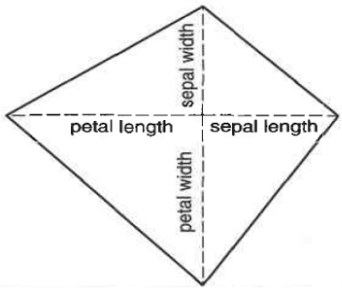
- Can plot the data matrix
- This can be useful when objects are sorted according to class
- Typically, the attributes are normalized to prevent one attribute from dominating the plot
- Plots of similarity or distance matrices can also be useful for visualizing the relationships between objects

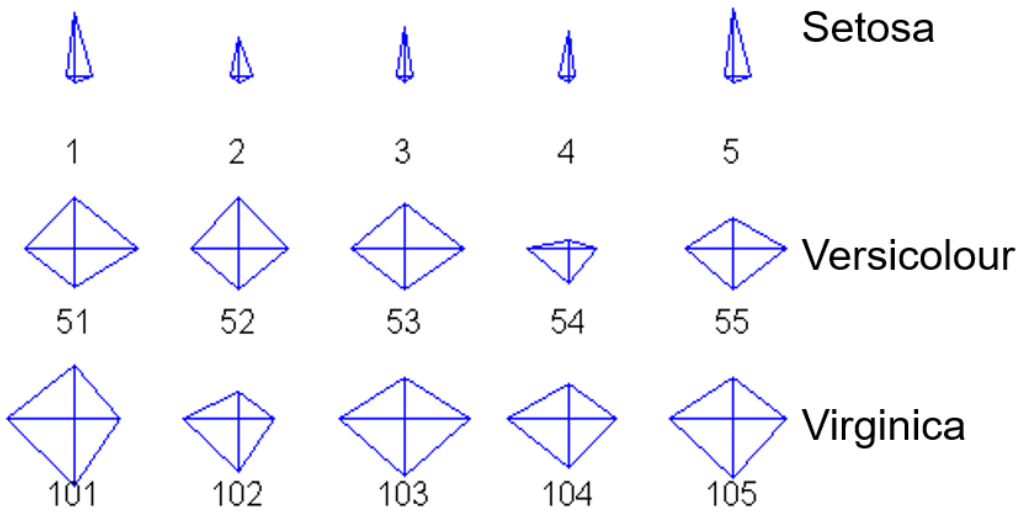


Other Visualization Techniques

Star Plots

- Similar approach to parallel coordinates, but axes radiate from a central point
- The line connecting the values of an object is a polygon

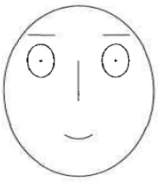


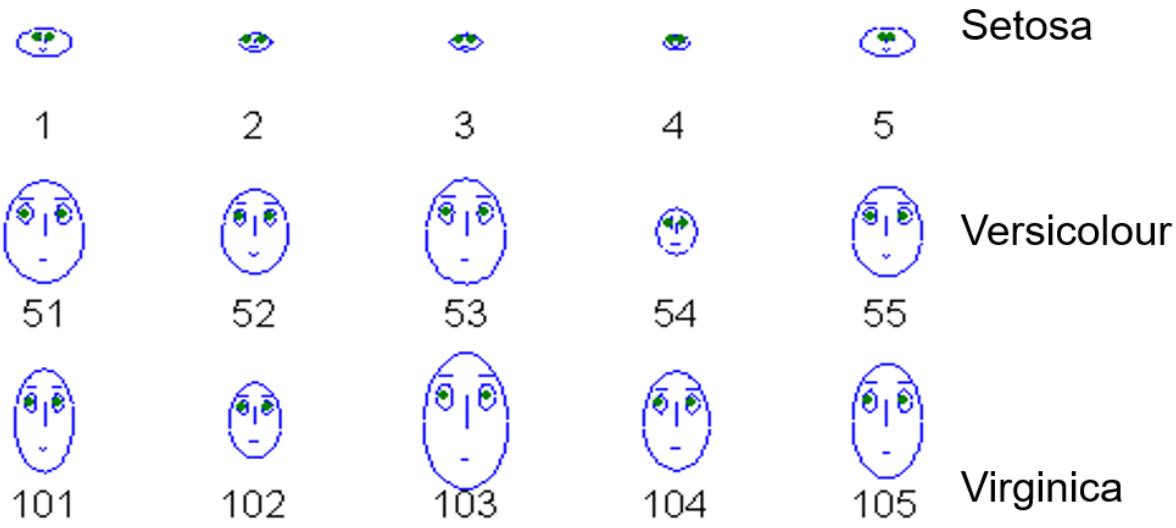


Chernoff Faces

innovate achieve lead

- Approach created by Herman Chernoff.
- This approach associates each attribute with a characteristic of a face.
- The values of each attribute determine the appearance of the corresponding facial characteristic.
- Each object becomes a separate face.
- Relies on human's ability to distinguish faces.





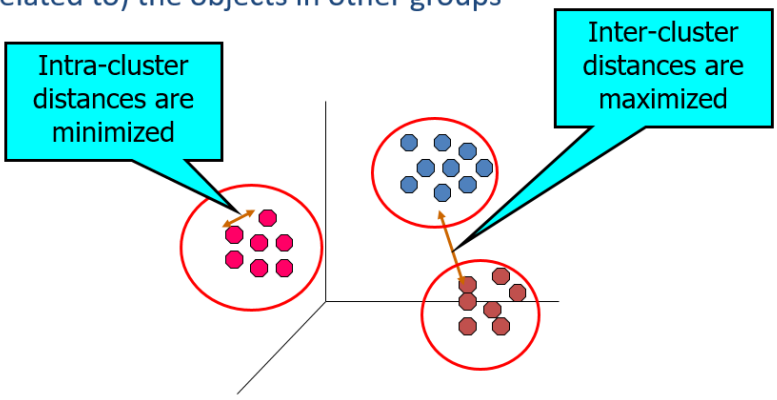
OLAP

- On-Line Analytical Processing (OLAP) was proposed by E. F. Codd, the father of the relational database.
- Relational databases put data into tables, while OLAP uses a multidimensional array representation.
- There are a number of data analysis and data exploration operations that are easier with such a data representation.

Lec 7 Cluster Analysis: Basic Concepts and Algorithms

What is Cluster Analysis?

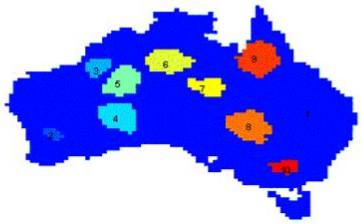
- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Organize the data in different groups

Applications of Cluster Analysis

- Understanding
  - Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations
- Summarization
  - Reduce the size of large data sets



What is not Cluster Analysis?

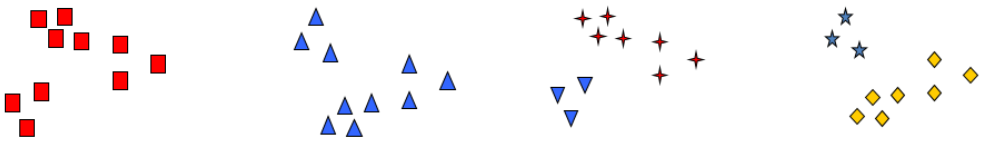
- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing students into different registration groups alphabetically, by last name
- Results of a query
  - Groupings are a result of an external specification

Clustering is unsupervised classification

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters



Two Clusters

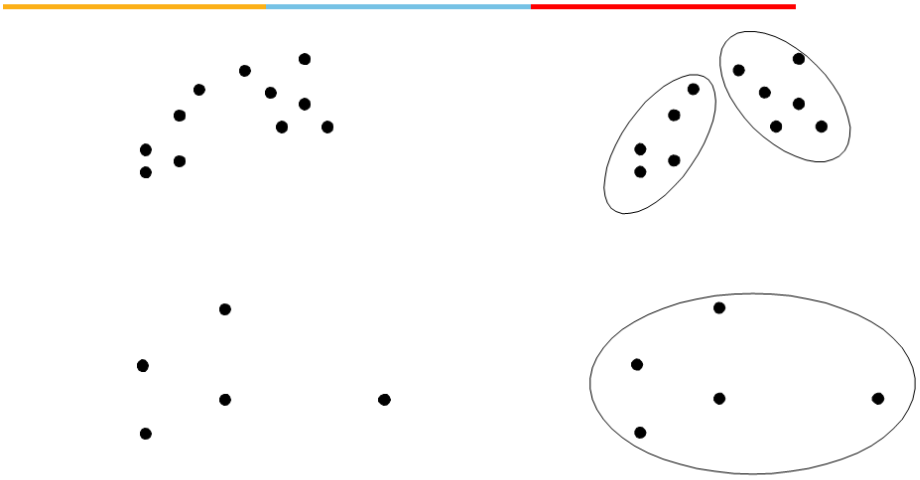
Four Clusters

Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

- Partitional Clustering

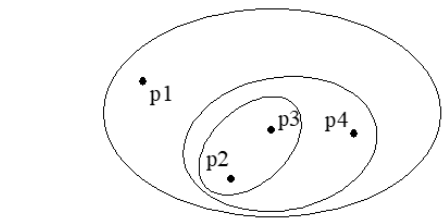
Partitional Clustering



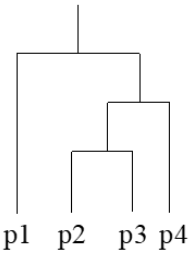
Original Points

A Partitional Clustering

Hierarchical Clustering



Hierarchical Clustering



Dendrogram

Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
  - In non-exclusive clusterings, 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

Partitional – exclusive, Fuzzy - same object multiple cluster with weight – 0.7 – A, 0.3 – B

Non Fuzzy – assign object to the cluster with higher similarity, Partitional is non fuzzy one object belongs to one cluster

- 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

Partial – noise in data – remove noise then do clustering, Different sizes – Heterogeneous

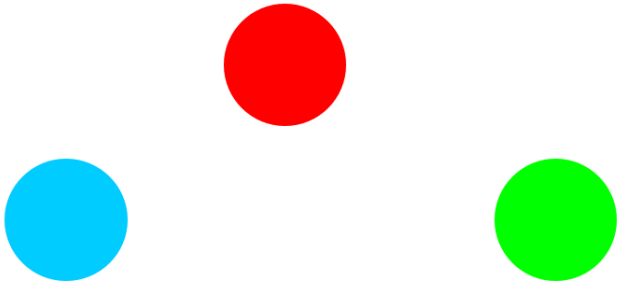
Mostly partial used –Java project –based on classes –group into two more complex and less complex – weighted method class (WML) – tester will more focus on high complexity class – bugs prone

Types of Clusters

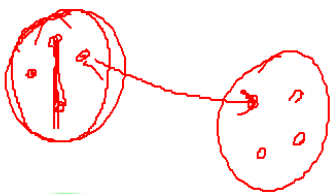
- Well-separated clusters
- Center-based clusters
- Contiguous clusters
- Density-based clusters
- Property or Conceptual
- Described by an Objective Function

Well-separated clusters

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



3 well-separated clusters



Intra Cluster distance < Inter Cluster distance

Center-Based

- Center-based
  - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster
  - 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

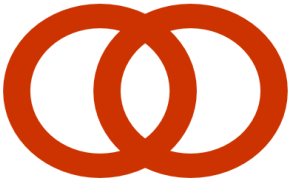
- Center-based

Contiguity-Based

- Contiguous Cluster (Nearest neighbor or Transitive)
  - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.
  - two objects are connected only if they are within a specified distance of each other.
  - Each point is closer to at least one point in its cluster than to any point in another cluster.

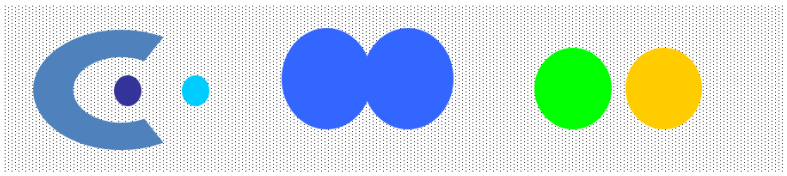
Conceptual Clusters

- Shared Property or Conceptual Clusters
  - Finds clusters that share some common property or represent a particular concept.

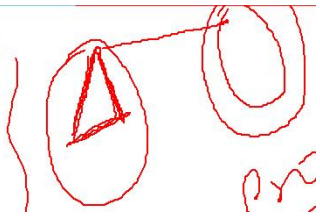


Density-Based

- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Clusters are regions of high density separated by regions of low density.



- Well-separated clusters
- Center-based clusters



Center Based

Contingency

Density based

Conceptual

Objective Function

innovate achieve lead

- Clusters Defined 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

Objective Function – minimize Intra cluster distance and maximize Inter cluster distance

Clustering Algorithms

innovate achieve lead

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

K-means Clustering

innovate achieve lead

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

One object belongs to one cluster – Partitional - Either one of cluster

Common Distance measures

innovate achieve lead

- Distance measure will determine how the similarity of two elements is calculated and it will influence the shape of the clusters.
- They include:
  - The Euclidean distance (also called 2-norm distance) is given by:

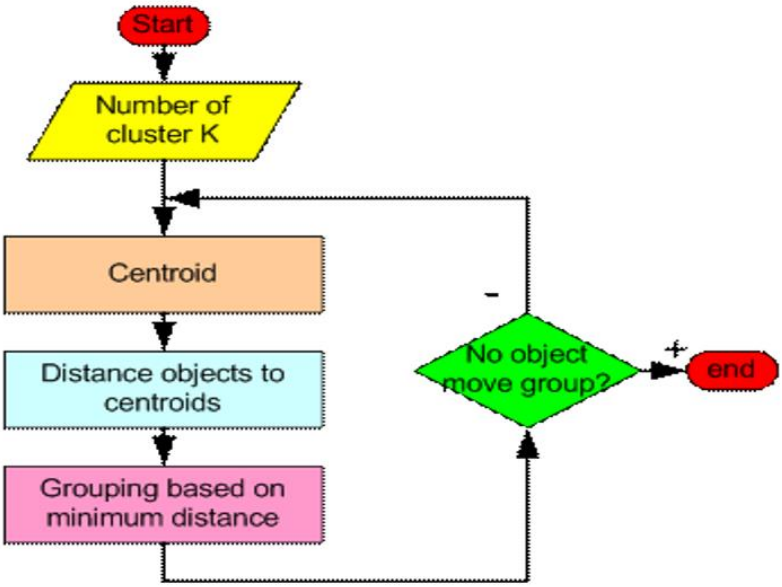
$$Euclidean\ Distance = d = \sqrt{\sum_{i=1}^{N}(Xi - Yi)^2}$$

Euclidian distance shows similarity between two objects – less than more similar, Euclidian distance = 0 -> both are same objects

K-MEANS CLUSTERING

- The k-means algorithm is an algorithm to cluster n objects based on attributes into k partitions, where k < n.
- Simply speaking k-means clustering is an algorithm to classify or to group the objects based on attributes/features into K number of group.
- K is positive integer number.
- The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

How the K-Mean Clustering algorithm works?



k-means algorithm (using K=2)

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

n = 7

Step 1:

- Initialization: Randomly we choose following two centroids (k=2) for two clusters.
- In this case the 2 centroid are: m1=(1.0,1.0) and m2=(5.0,7.0).

Individual	Variable 1	Variable 2
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Step 2:

$$d(m_1,2)=\sqrt{|1.0-1.5|^2+|1.0-2.0|^2}=1.12$$

$$d(m_2,2)=\sqrt{|5.0-1.5|^2+|7.0-2.0|^2}=6.10$$

Individual	Centroid 1	Centroid 2
1	0	7.21
2 (1.5, 2.0)	1.12	6.10
3	3.61	3.61
4	7.21	0
5	4.72	2.5
6	5.31	2.06
7	4.30	2.92

- Thus, we obtain two clusters containing: {1,2,3} and {4,5,6,7}.
- Their new centroids are:

$$m_1=\left(\frac{1}{3}(1.0+1.5+3.0),\frac{1}{3}(1.0+2.0+4.0)\right)=(1.83,2.33)$$

$$m_2=\left(\frac{1}{4}(5.0+3.5+4.5+3.5),\frac{1}{4}(7.0+5.0+5.0+4.5)\right)$$

$$=(4.12,5.38)$$

- Now using these centroids we compute the Euclidean distance of each object, as shown in table.
- Therefore, the new clusters are:

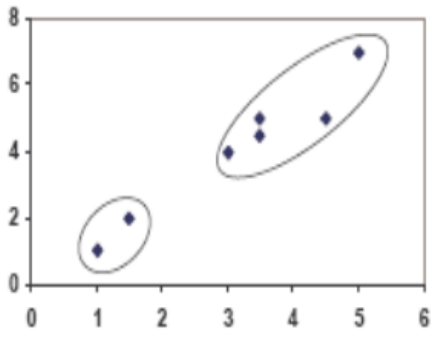
  {1,2} and {3,4,5,6,7}

Individual	Centroid 1	Centroid 2
1	1.57	5.38
2	0.47	4.28
3	2.04	1.78
4	5.64	1.84
5	3.15	0.73
6	3.78	0.54
7	2.74	1.08

- Next centroids are: m1=(1.25,1.5) and m2 = (3.9,5.1)
  - The clusters obtained are:
  - {1,2} and {3,4,5,6,7}
  - Therefore, there is no change in the cluster.
  - Thus, the algorithm comes to a halt here and final result consist of 2 clusters {1,2} and {3,4,5,6,7}.

Individual	Centroid 1	Centroid 2
1	0.56	5.02
2	0. 56	3.92
3	3.05	1.42
4	6.66	2.20
5	4.16	0.41
6	4.78	0.61
7	3.75	0.72

PLOT



Evaluating K-means Clusters

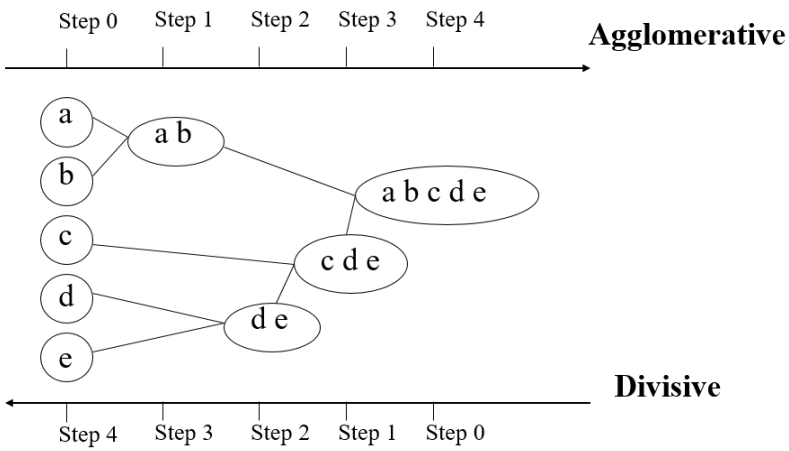
- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

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

  - x is a data point in cluster Ci and mi is the representative point for cluster Ci
  - can show that mi corresponds to the center (mean) of the cluster
  - 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

- Hierarchical Clustering Approach
  - A typical clustering analysis approach via partitioning data set sequentially
  - Construct nested partitions layer by layer via grouping objects into a tree of clusters (without the need to know the number of clusters in advance)
  - Use (generalised) distance matrix as clustering criteria
- Agglomerative: a bottom-up strategy
  - Initially each data object is in its own (atomic) cluster
  - Then merge these atomic clusters into larger and larger clusters
- Divisive: a top-down strategy
  - Initially all objects are in one single cluster
  - Then the cluster is subdivided into smaller and smaller clusters



Problem with K mean cluster – how to decide value of k

Bottom up – each object is one cluster – find similarity – then put same objects together till reaches single cluster, Top – down – start with single cluster – divide into sub clusters

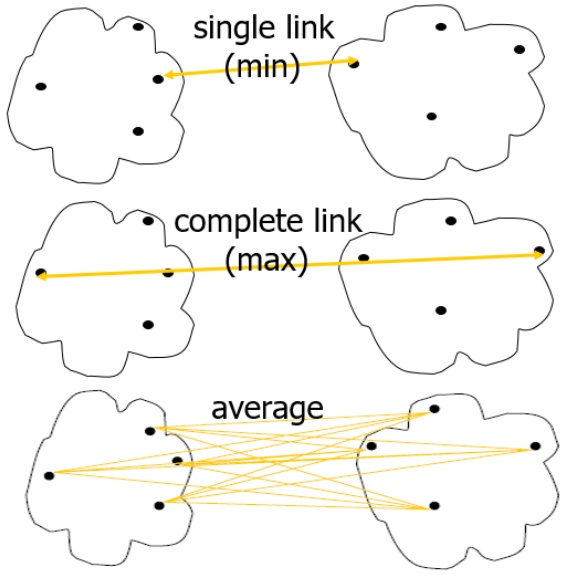
Cluster Distance Measures

innovate achieve lead

Single link: smallest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \min\{d(x_{ip}, x_{jq})\}$

Complete link: largest distance between an element in one cluster and an element in the other, i.e., $d(C_i, C_j) = \max\{d(x_{ip}, x_{jq})\}$

Average: avg distance between elements in one cluster and elements in the other, i.e., $d(C_i, C_j) = \text{avg}\{d(x_{ip}, x_{jq})\}$



- Given a data set of five objects characterised by a single continuous feature, assume that there are two clusters: C1: {a, b} and C2: {c, d, e}.

	a	b	c	d	e
a	0	1	3	4	5
b	1	0	2	3	4
c	3	2	0	1	2
d	4	3	1	0	1
e	5	4	2	1	0

Single link

$$\text{dist}(C_1, C_2) = \min\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} = \min\{3,4,5,2,3,4\} = 2$$

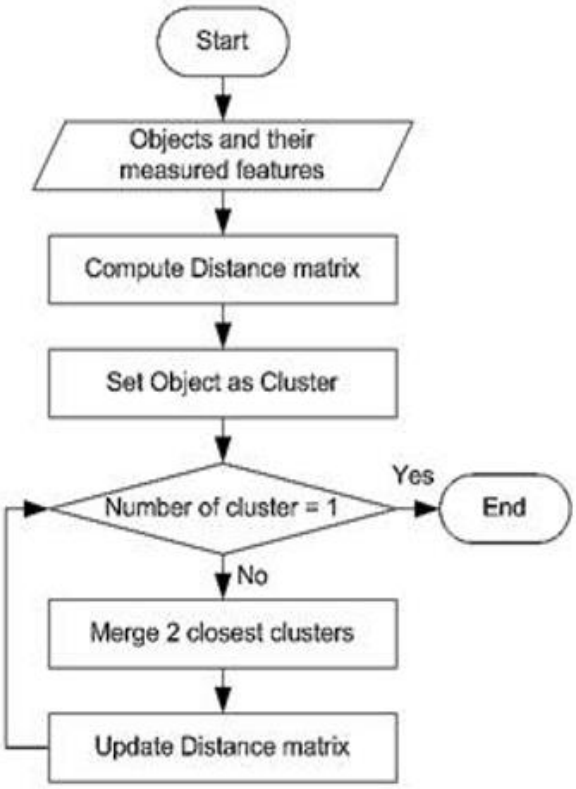
Complete link

$$\text{dist}(C_1, C_2) = \max\{d(a,c), d(a,d), d(a,e), d(b,c), d(b,d), d(b,e)\} = \max\{3,4,5,2,3,4\} = 5$$

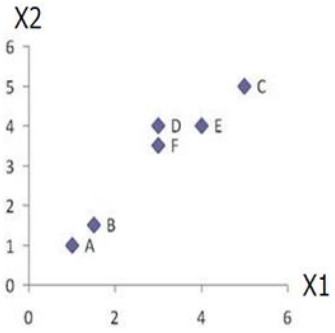
Average

$$\text{dist}(C_1, C_2) = \frac{d(a,c) + d(a,d) + d(a,e) + d(b,c) + d(b,d) + d(b,e)}{6} = \frac{3+4+5+2+3+4}{6} = \frac{21}{6} = 3.5$$

Agglomerative Algorithm Bottom to Top



Data containing objects (rows) – features (columns)



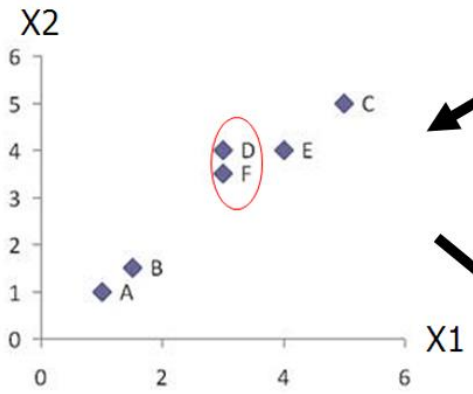
	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

data matrix

$$d_{AB} = \left((1-1.5)^2 + (1-1.5)^2\right)^{\frac{1}{2}} = \sqrt{\tfrac{1}{2}} = 0.7071$$

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

distance matrix



Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

$$d_{(D,F)\to A} = \min\left(d_{DA}, d_{FA}\right) = \min\left(3.61, 3.20\right) = 3.20$$

$$d_{(D,F)\to B} = \min\left(d_{DB}, d_{FB}\right) = \min\left(2.92, 2.50\right) = 2.50$$

$$d_{(D,F)\to C} = \min\left(d_{DC}, d_{FC}\right) = \min\left(2.24, 2.50\right) = 2.24$$

$$d_{E\to(D,F)} = \min\left(d_{ED}, d_{EF}\right) = \min\left(1.00, 1.12\right) = 1.00$$

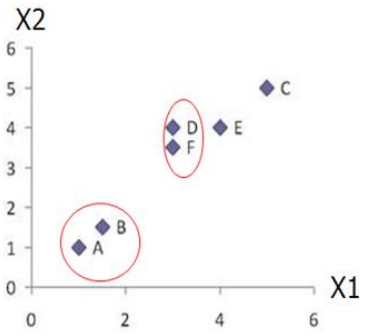
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	?	4.24
B	0.71	0.00	4.95	?	3.54
C	5.66	4.95	0.00	?	1.41
D, F	?	?	?	0.00	?
E	4.24	3.54	1.41	?	0.00

Min Distance (Single Linkage)

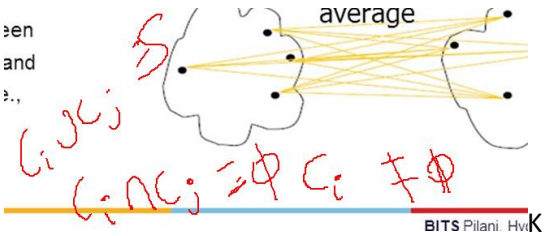
Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00

Min Distance (Single Linkage)

Dist	A	B	C	D, F	E
A	0.00	0.71	5.66	3.20	4.24
B	0.71	0.00	4.95	2.50	3.54
C	5.66	4.95	0.00	2.24	1.41
D, F	**3.20**	**2.50**	**2.24**	0.00	1.00
E	4.24	3.54	1.41	**1.00**	0.00



Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0



K mean cluster property

$$d_{C\to(A,B)} = \min\left(d_{CA}, d_{CB}\right) = \min\left(5.66, 4.95\right) = 4.95$$

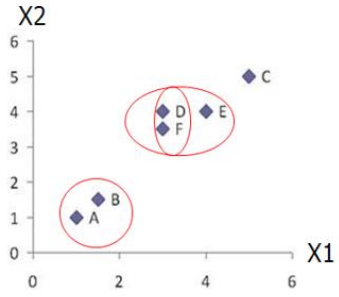
$$d_{(D,F)\to(A,B)} = \min\left(d_{DA}, d_{DB}, d_{FA}, d_{FB}\right) = \min\left(3.61, 2.92, 3.20, 2.50\right) = 2.50$$

$$d_{E\to(A,B)} = \min\left(d_{EA}, d_{EB}\right) = \min\left(4.24, 3.54\right) = 3.54$$

Dist	A,B	C	(D, F)	E
A,B	0	?	?	?
C	?	0	2.24	1.41
(D, F)	?	2.24	0	1.00
E	?	1.41	1.00	0

Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

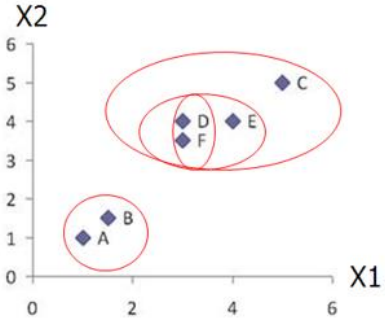


Min Distance (Single Linkage)

Dist	A,B	C	(D, F)	E
A,B	0	4.95	2.50	3.54
C	**4.95**	0	2.24	1.41
(D, F)	**2.50**	2.24	0	1.00
E	**3.54**	1.41	1.00	0

Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00



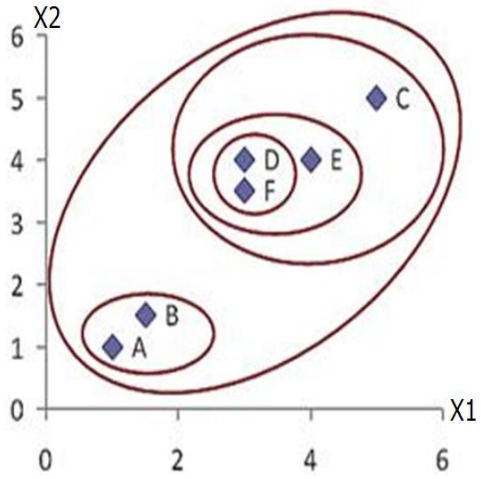
Min Distance (Single Linkage)

Dist	(A,B)	C	(D, F), E
(A,B)	0.00	4.95	2.50
C	4.95	0.00	1.41
(D, F), E	**2.50**	**1.41**	0.00

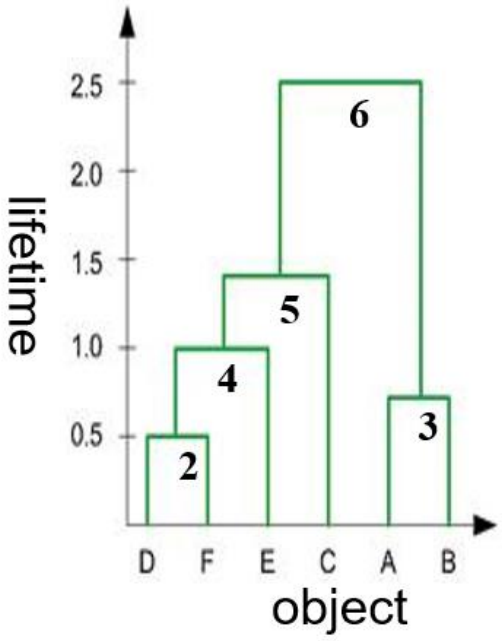
Min Distance (Single Linkage)

Dist	(A,B)	((D, F), E),C
(A,B)	0.00	2.50
((D, F), E),C	**2.50**	0.00

	X1	X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5



- In the beginning we have 6 clusters: A, B, C, D, E and F
- We merge clusters D and F into cluster (D, F) at distance 0.50
- We merge cluster A and cluster B into (A, B) at distance 0.71
- We merge clusters E and (D, F) into ((D, F), E) at distance 1.00
- We merge clusters ((D, F), E) and C into (((D, F), E), C) at distance 1.41
- We merge clusters (((D, F), E), C) and (A, B) into ((((D, F), E), C), (A, B)) at distance 2.50
- The last cluster contain all the objects, thus conclude the computation



Hierarchical clustering is better than K mean clustering

Fuzzy c-means clustering

- In fuzzy clustering, every point has a degree of belonging to clusters
- Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster.
- The FCM algorithm is one of the most widely used fuzzy clustering algorithms.
- This technique was originally introduced by Professor Jim Bezdek in 1981.

How much portion of an object belongs to cluster 1 and cluster 2

i.e. Uij = 0.8 → Membership/Probability value of Object i w.r.t. Cluster j → Uij >=0 & Uij<=1

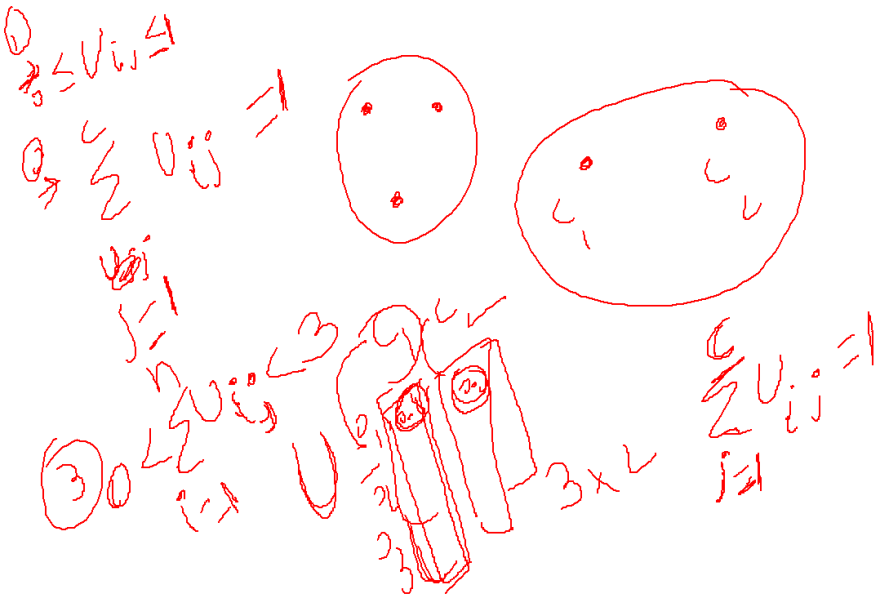
Fuzzy c-means clustering

- The FCM algorithm attempts to partition a finite collection of elements X={,, ...,} into a collection of c fuzzy clusters with respect to some given criterion.
- The algorithm is based on minimization of the following objective function

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2$$

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2$$

- where m (the Fuzziness Exponent) is any real number greater than 1
- N is the number of data
- C is the number of clusters
- uij is the degree of membership of xi in the cluster j
- xi is the ith of d-dimensional measured data
- cj is the d-dimension center of the cluster
- ||*|| is any norm expressing the similarity between any measured data and the center.



Fuzzy c-means clustering

- Given a finite set of data, the algorithm returns a list of c cluster centers V, such that V=vi, i =1, 2, ... , c
- Membership matrix U such that

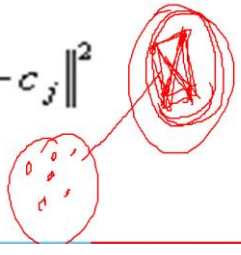
  U = uij , i =1, ..., c , j =1,..., n
  - Where uij is a numerical value in [0, 1] that tells the degree to which the element xj belongs to the i-th cluster.
- Summation of membership of each data point should be equal to one.

Parameters of the FCM algorithm

- Before using the FCM algorithm, the following parameters must be specified:
  - the number of clusters, c,
  - the fuzziness exponent, m, where m is any real number greater than 1

The algorithm is based on minimization of the following objective function

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2$$



Steps

innovate achieve lead

- Step1: choose random centroid at least 2
- Step2: compute membership matrix.

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}} = \frac{1}{\left(\frac{\|x_i - c_j\|}{\|x_i - c_1\|}\right)^{2/(m-1)} + \left(\frac{\|x_i - c_j\|}{\|x_i - c_2\|}\right)^{2/(m-1)} + \cdots + \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{2/(m-1)}}$$

where ||xi - cj|| is the Distance from point i to current cluster centre j, ||xi - ck|| is the Distance from point i to other cluster centers k.(note if we have 2D data we use euclidean distance).

$$u_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}} = \frac{1}{\left(\frac{\|x_i - c_j\|}{\|x_i - c_1\|}\right)^{2/(m-1)} + \left(\frac{\|x_i - c_j\|}{\|x_i - c_2\|}\right)^{2/(m-1)} + \cdots + \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{2/(m-1)}}$$

Step3: calculate the c cluster centers.

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}}$$

Step4: If the difference between the original centroid and the next one is below a certain threshold value say, €, then the algorithm stops, else it continues till this condition is true.

Let x=[2 3 4 5 6 7 8 9 10 11] , m=2, number of cluster C=2, c1=3, c2=11.

Step 1: for first iteration calculate membership matrix.

U11 → membership of 1st Object with 1st Cluster

$$U11 = \frac{1}{\left(\frac{2-3}{2-3}\right)^{\frac{2}{2-1}} + \left(\frac{2-3}{2-11}\right)^{\frac{2}{2-1}}} = \frac{1}{1+\frac{1}{81}} = \frac{81}{82} = 98.78\%$$

The membership of first node to first cluster

$$U12 = \frac{1}{\left(\frac{2-11}{2-3}\right)^{\frac{2}{2-1}} + \left(\frac{2-11}{2-11}\right)^{\frac{2}{2-1}}} = \frac{1}{81+1} = \frac{1}{82} = 1.22\%$$

The membership of first node to second cluster

X	cluster1	cluster2
2	0.9878	0.0122
3	1.0000	0
4	0.9800	0.0200
5	0.9000	0.1000
6	0.7353	0.2647
7	0.5000	0.5000
8	0.2647	0.7353
9	0.1000	0.9000
10	0.0200	0.9800
11	0	1.0000

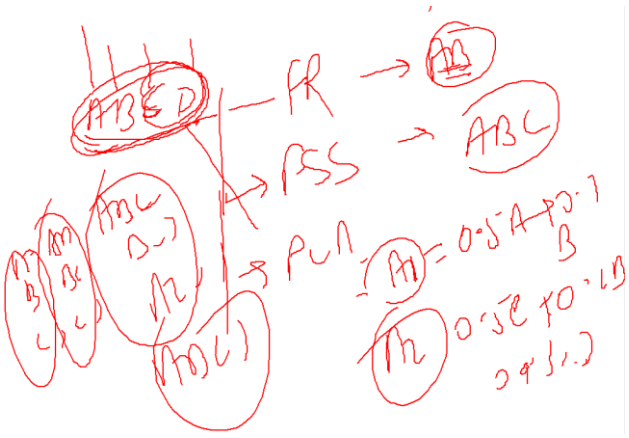
C1=4.0049
C2=9.4576

Mid sem → Up to Clustering technique **– Clustering techniques, Visualization techniques**, Characteristics of objects (mean, median etc.), Similarity between objects – Distance, Simple matching etc., Type of Attributes

Real example of clustering technique – Program – classes in three group – high complexity, medium and low complexity class – features – no. of lines, no. of conditions etc.

High complexity classes – more error prone – high priority

Dimension reduction – Feature Ranking, Feature Selection – Subset – only selected features have impact, Feature Extraction (PCA) –construct new features have impact on data similar to old features



Lec 8 Classification: Basic Concepts, Decision Tree, Model Evaluation

Unsupervised – no idea about o/p, labels not given – features given – cluster them – clustering

Supervised – collection of historical data – create model – classify test data using model - accuracy

Classification: Definition

innovate achieve lead

- Given a collection of records (training set)
  - Each record contains a set of attributes, one of the attributes is the class.
- Find a model for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A test set is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Classification Task

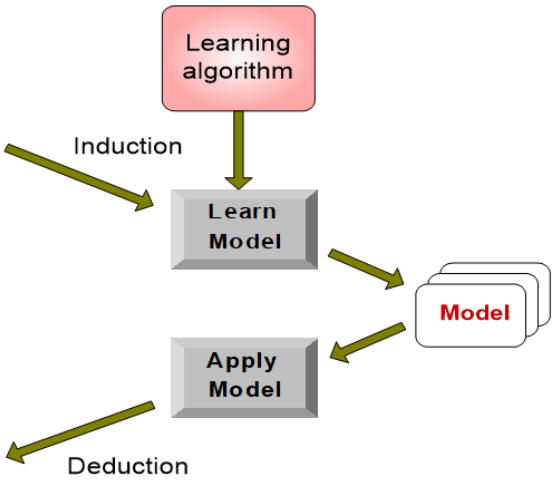
innovate achieve lead

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

Training Set

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

Test Set

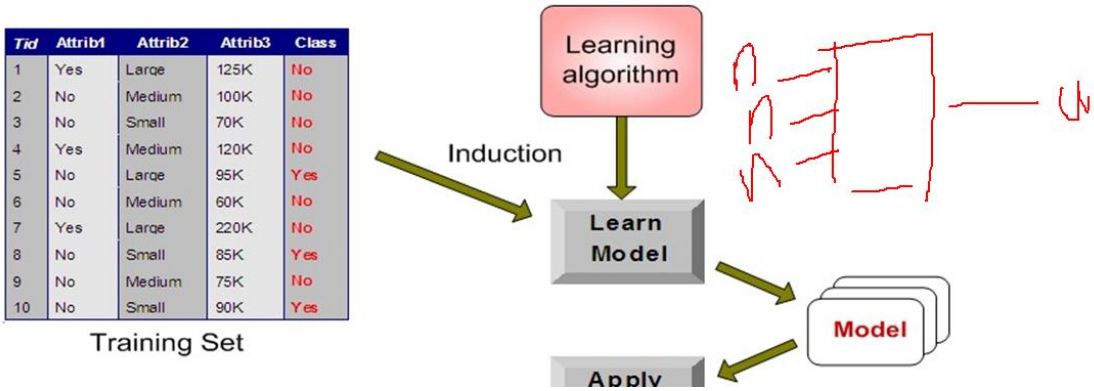


Learning algorithm – classification technique – to develop model

Classification Techniques

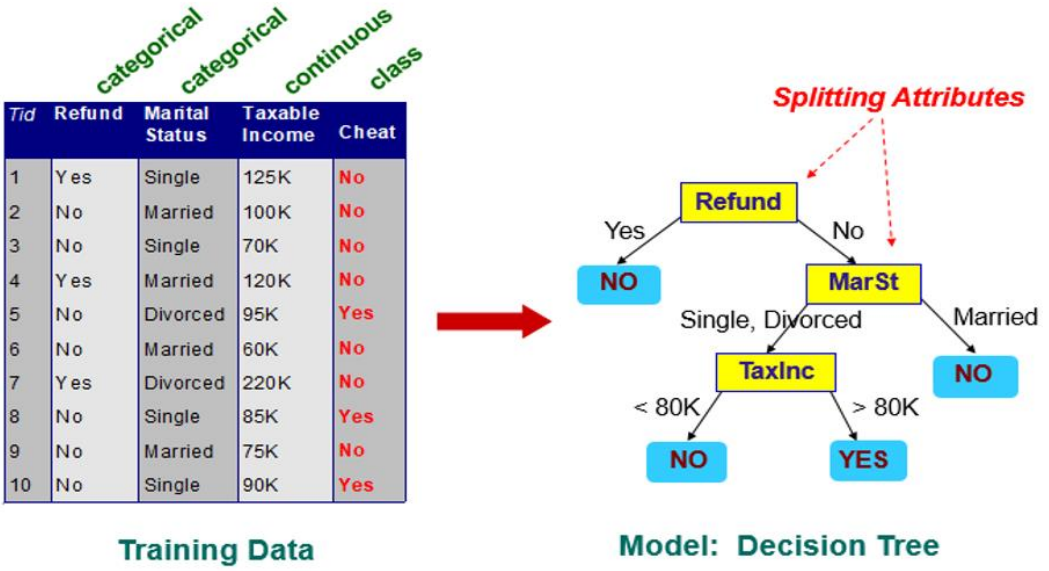
innovate achieve lead

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

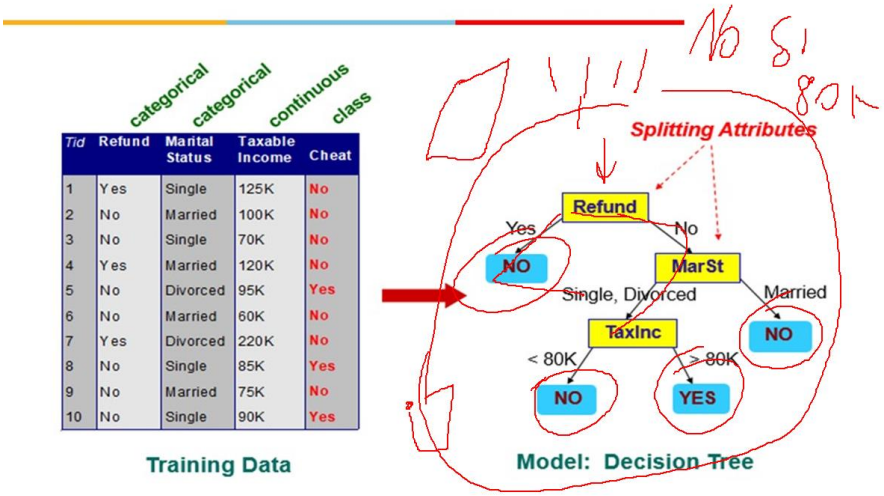


Example of a Decision Tree

innovate achieve lead

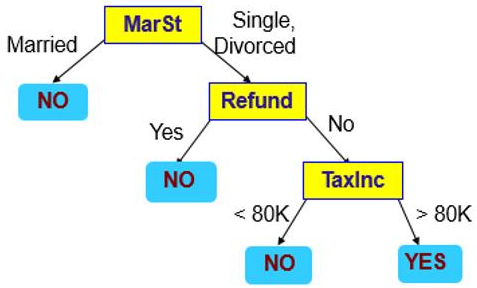


Tree - Cycle free graph, unique path between every node pair of tree, except root each node has only one incoming edge, leaf node – no outgoing edge



Another Example of Decision Tree

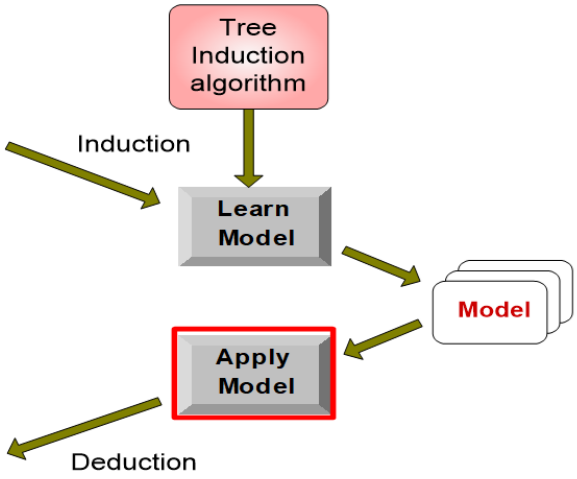
Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



There could be more than one tree that fits the same data!

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

Training Set

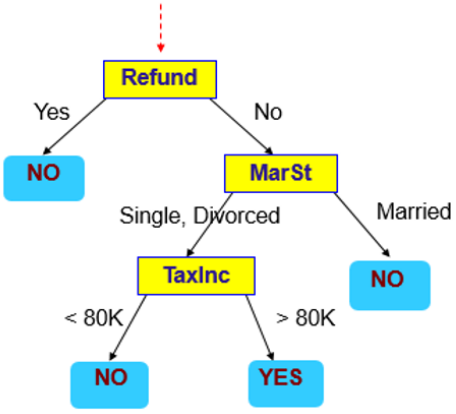


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

Apply Model to Test Data

Start from the root of tree.



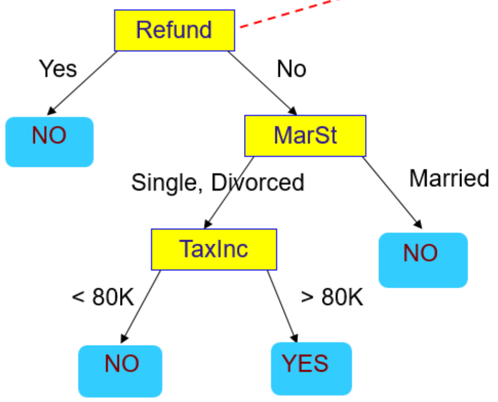
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

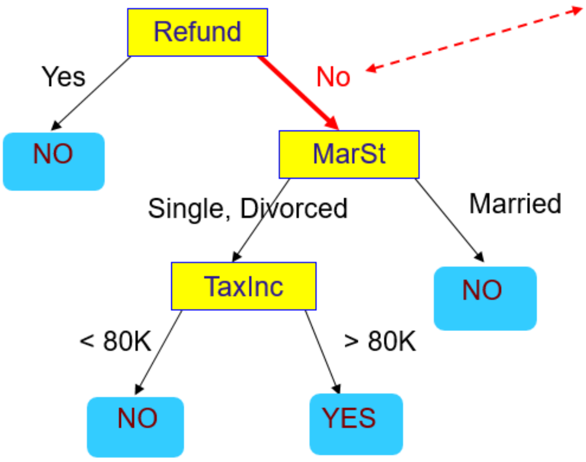
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



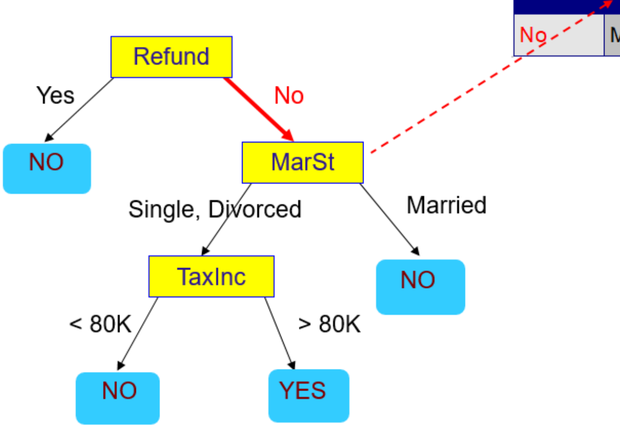
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



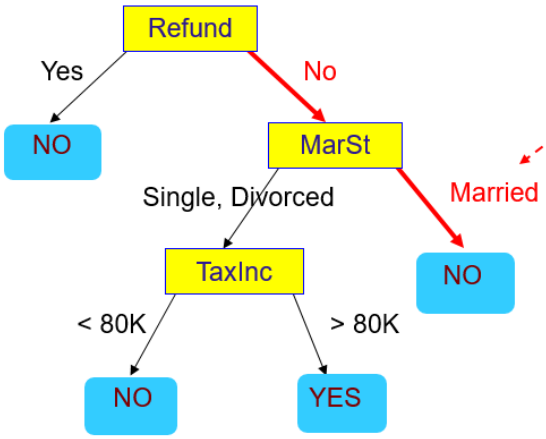
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



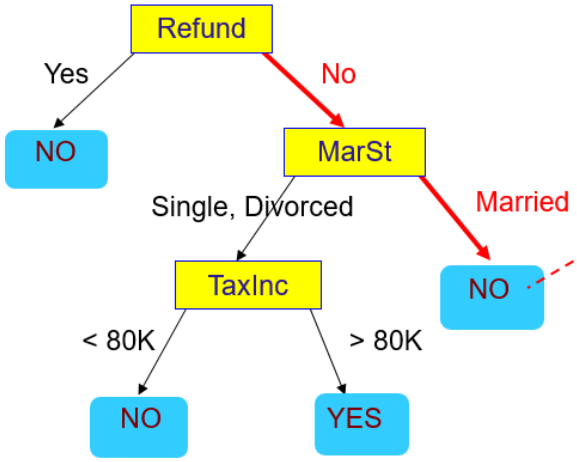
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Test Data

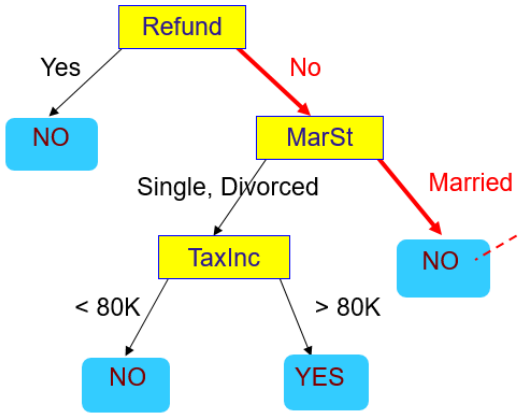
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to “No”

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to “No”

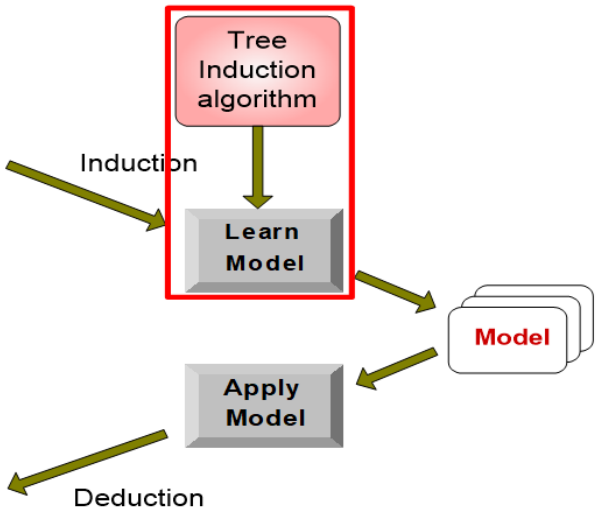
Decision Tree Classification Task

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

Training Set

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

Test Set



Decision Tree Induction

- Many Algorithms:
  - Hunt's Algorithm (one of the earliest)
  - CART
  - ID3, C4.5
  - SLIQ,SPRINT

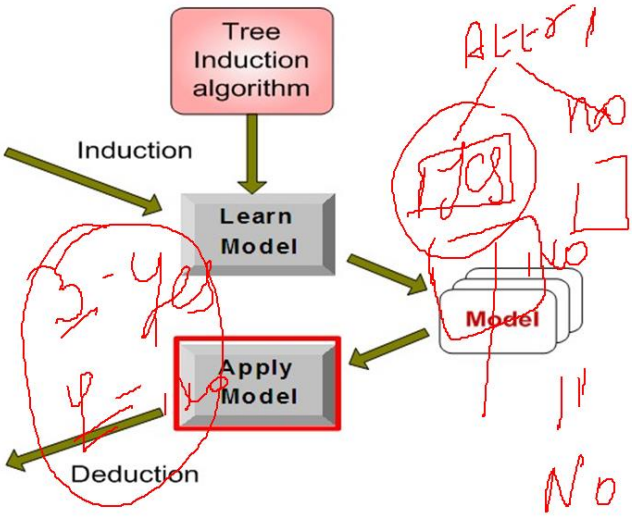
General Structure of Hunt's Algorithm

- Let Dt be the set of training records that reach a node t
- General Procedure:
  - If Dt contains records that belong the same class yt, then t is a leaf node labeled as yt
  - If Dt is an empty set, then t is a leaf node labeled by the default class, yd
  - If Dt contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

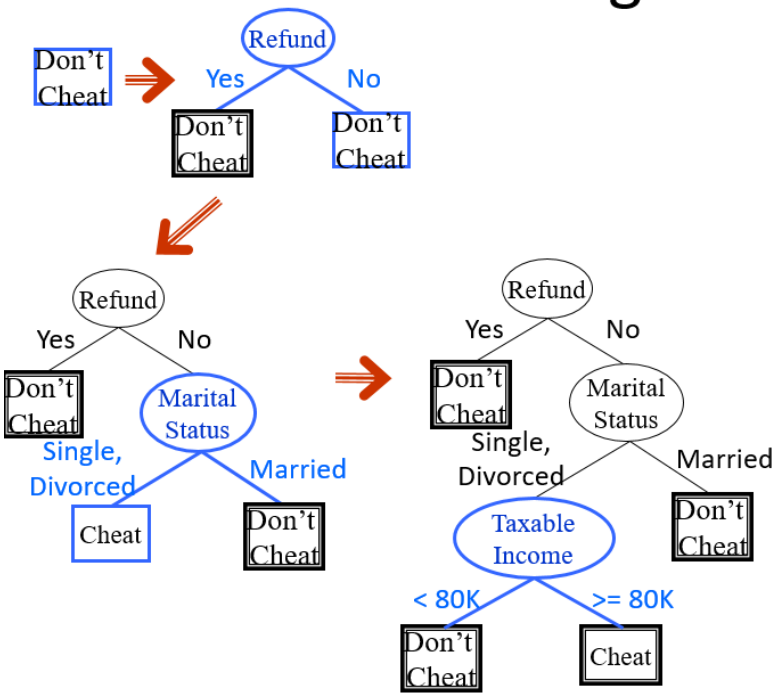
Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

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



Hunt's Algorithm

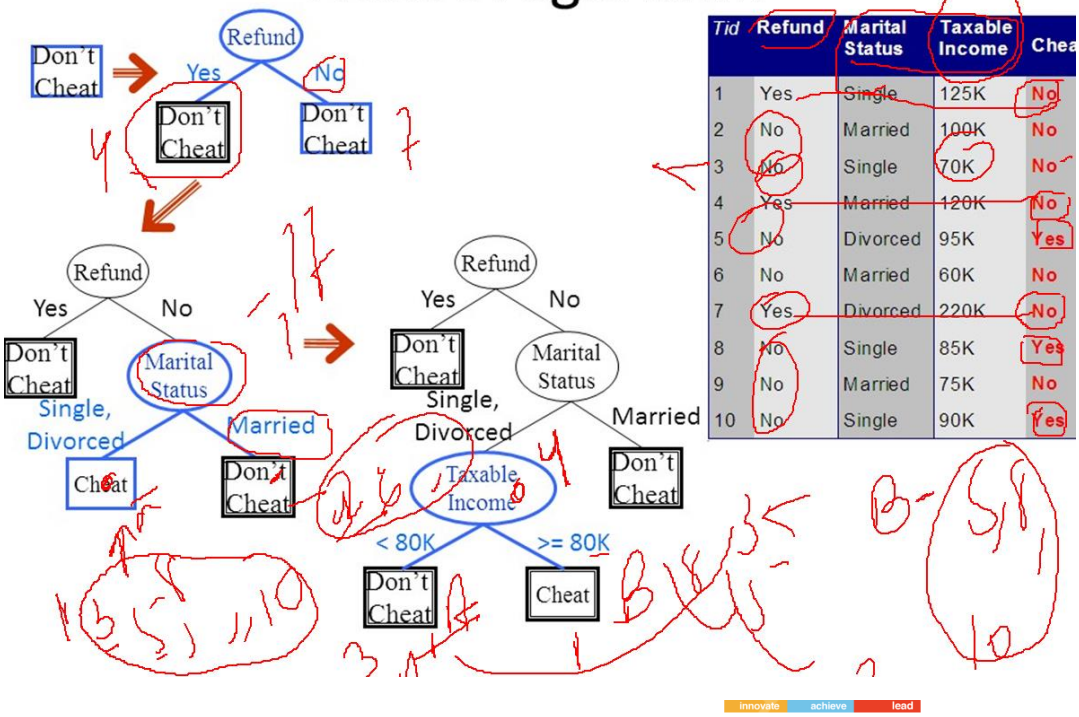


Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.
- Issues
- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

Hunt's Algorithm



Tid	Refund	Marital Status	Taxable Income	Chea
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

innovate achieve lead

How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Nominal (no comparison), Ordinal – Discrete Data

Types of Attributes

innovate achieve lead

There are different types of attributes

- Nominal: **Data** are neither measured nor ordered but subjects are merely allocated to distinct categories

  Examples: ID numbers, eye color, zip codes
- Ordinal: **Ordinal data** is a categorical where the variables have natural, ordered categories and the distances between the categories is not known.

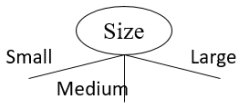
  Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
- Interval: In **interval** measurement the distance between attributes does have meaning.

  Examples: calendar dates, temperatures in Celsius or Fahrenheit.
- Ratio:

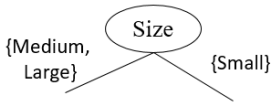
  Examples: temperature in Kelvin, length, time, counts

Splitting Based on Ordinal Attributes

Multi-way split: Use as many partitions as distinct values.

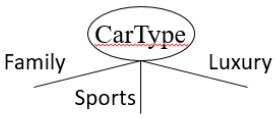


Binary split: Divides values into two subsets.
Need to find optimal partitioning.

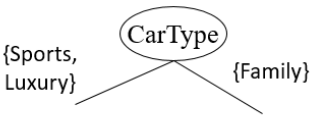


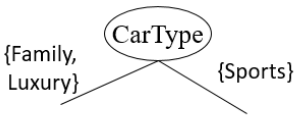
Splitting Based on Nominal Attributes

Multi-way split: Use as many partitions as distinct values.



Binary split: Divides values into two subsets.
Need to find optimal partitioning.



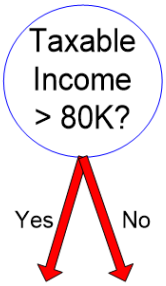


Splitting Based on Continuous Attributes

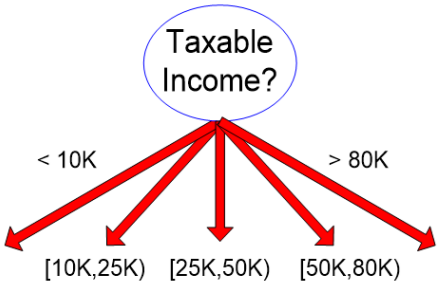
Different ways of handling

- Discretization to form an ordinal categorical attribute
  - Static – discretize once at the beginning
  - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.

- Binary Decision: $(A < v)$ or $(A \geq v)$
  - consider all possible splits and finds the best cut
  - can be more compute intensive



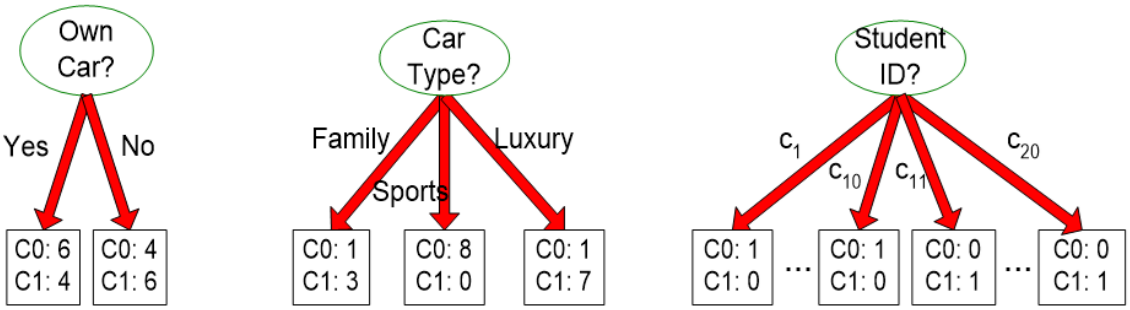
(i) Binary split



(ii) Multi-way split

How to determine the Best Split

Before Splitting: 10 records of class 0,
10 records of class 1



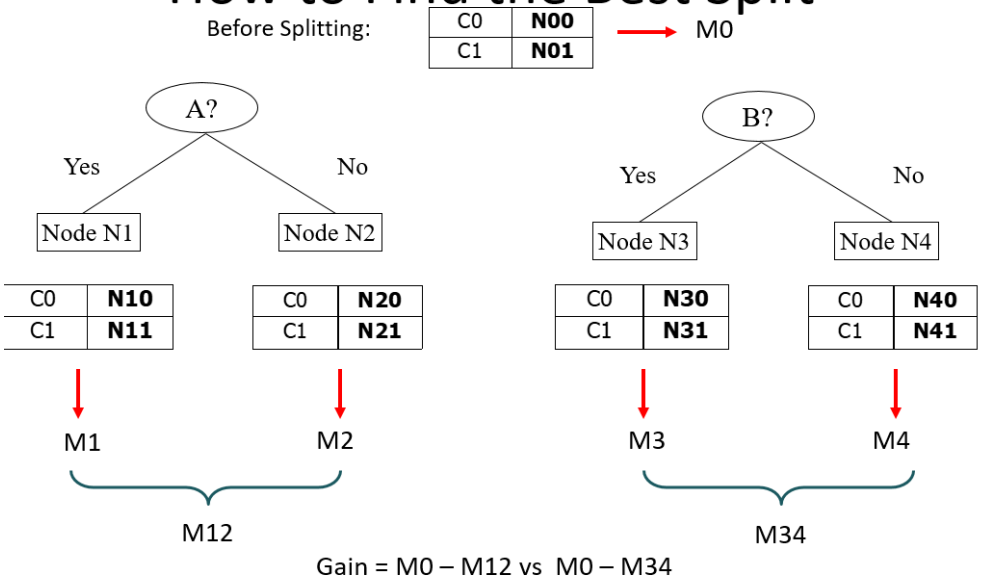
Which test condition is the best?

Best case – performance of attribute – performance parameter – attribute – splitting information in two groups – one group contains data of one class only

Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification error

How to Find the Best Split



Gain = M0 – M12 vs M0 – M34

Measure of Impurity: GINI

innovate achieve lead

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

- (NOTE: p(j | t) is the relative frequency of class j at node t).
  - Maximum (1 - 1/nc) when records are equally distributed among all classes, implying least interesting information
  - Minimum (0.0) when all records belong to one class, implying most interesting information

C1	0
C2	6
Gini=0.000	

C1	1
C2	5
Gini=0.278	

C1	2
C2	4
Gini=0.444	

C1	3
C2	3
Gini=0.500	

Examples for computing GINI

innovate achieve lead

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	0
C2	6

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – $P(C1)^2$ – $P(C2)^2$ = 1 – 0 – 1 = 0

C1	1
C2	5

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – $(1/6)^2$ – $(5/6)^2$ = 0.278

C1	2
C2	4

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – $(2/6)^2$ – $(4/6)^2$ = 0.444

Splitting Based on GINI

innovate achieve lead

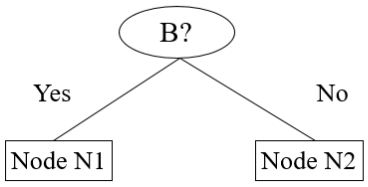
- When a node p is split into k partitions (children), the quality of split is computed as,

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child i,
n = number of records at node p.

Binary Attributes: Computing GINI Index

- Splits into two partitions



	Parent
C1	6
C2	6
Gini = 0.500	

Gini(N1)
= 1 – $(5/6)^2$ – $(2/6)^2$
= 0.194

Gini(N2)
= 1 – $(1/6)^2$ – $(4/6)^2$
= 0.528

	N1	N2
C1	5	1
C2	2	4
Gini=0.333		

Gini(Children)
= 7/12 * 0.194 +
5/12 * 0.528
= 0.333

We consider attributes – with low GINI Index value

Categorical Attributes: Computing Gini Index

- For each distinct value, gather counts for each class in the dataset
- Use the count matrix to make decisions

	CarType		
	Family	Sports	Luxury
C1	1	2	1
C2	4	1	1
Gini	0.393		

	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

	CarType	
	{Sports}	{Family, Luxury}
C1	2	2
C2	1	5
Gini	0.419	

Continuous Attributes: Computing Gini Index...

- For efficient computation: for each attribute,
  - Sort the attribute on values
  - Linearly scan these values, each time updating the count matrix and computing gini index
  - Choose the split position that has the least gini index

Sorted Values →
Split Positions →

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
Taxable Income										
	60	70	75	85	90	95	100	120	125	220

	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		***0.300***		0.343		0.375		0.400		0.420	

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

Alternative Splitting Criteria based on INFO

- Entropy at a given node t:

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

  - (NOTE: p(j | t) is the relative frequency of class j at node t).
- Measures homogeneity of a node.
  - Maximum (log nc) when records are equally distributed among all classes implying least information
  - Minimum (0.0) when all records belong to one class, implying most information
- Entropy based computations are similar to the GINI index computations

$$Entropy(t) = -\sum_j p(j \mid t) \log_2 p(j \mid t)$$

C1	0
C2	6

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	1
C2	5

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	2
C2	4

P(C1) = 2/6 P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

  - Parent Node, p is split into k partitions;
  - ni is number of records in partition i
- Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)

Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

Consider attributes with – maximum GainRATIO

Splitting Criteria based on Classification Error

- Classification error at a node t :

$$Error(t) = 1 - \max_i P(i \mid t)$$

- Measures misclassification error made by a node.
  - Maximum (1 - 1/nc) when records are equally distributed among all classes, implying least interesting information
  - Minimum (0.0) when all records belong to one class, implying most interesting information

$$Error(t) = 1 - \max_i P(i \mid t)$$

C1	0
C2	6

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	1
C2	5

P(C1) = 1/6 P(C2) = 5/6

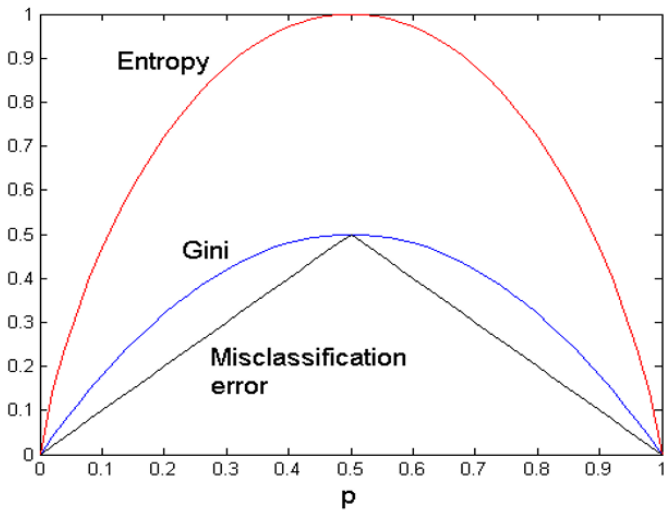
Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

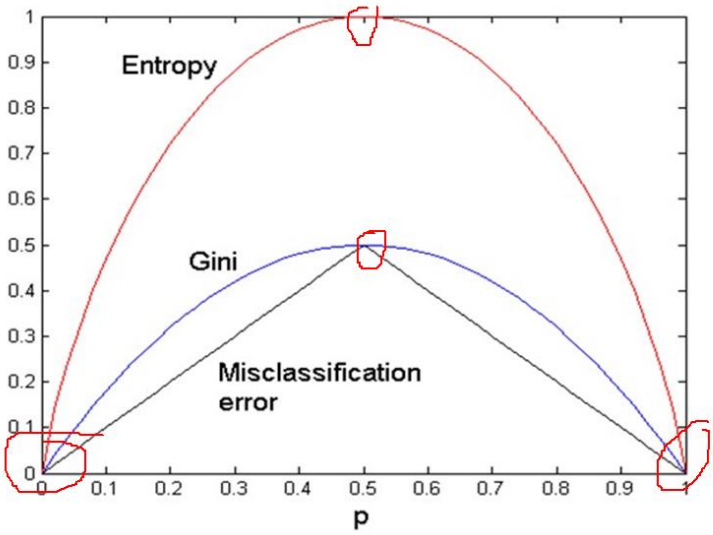
C1	2
C2	4

P(C1) = 2/6 P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Comparison among Splitting Criteria





GINI, Error – 0 – Best case, Max – worst case, Entropy – 0 – worst case, Max – best case

Stopping Criteria for Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values

Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance among competing models?

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Metrics for Performance Evaluation…

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Metrics for Performance Evaluation

$$\text{Precision (p)} = \frac{a}{a+c}$$

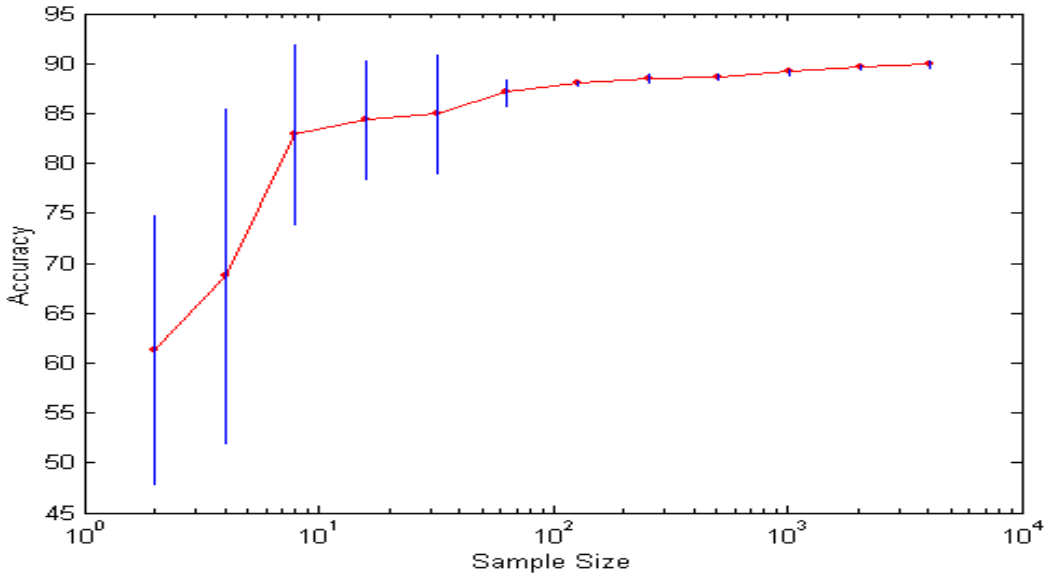
$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

Methods for Performance Evaluation

- How to obtain a reliable estimate of performance?
- Performance of a model may depend on other factors besides the learning algorithm:
  - Class distribution
  - Cost of misclassification
  - Size of training and test sets

Increase sample size (training data) – accuracy increases



Methods of Estimation

innovate achieve lead

Holdout
- Reserve 2/3 for training and 1/3 for testing

Random subsampling
- Repeated holdout

Cross validation
- Partition data into k disjoint subsets
- k-fold: train on k-1 partitions, test on the remaining one
- Leave-one-out: k=n

Stratified sampling
- oversampling vs undersampling

Bootstrap
- Sampling with replacement

Methods of Estimation

innovate achieve lead

Holdout
- Reserve 2/3 for training and 1/3 for testing

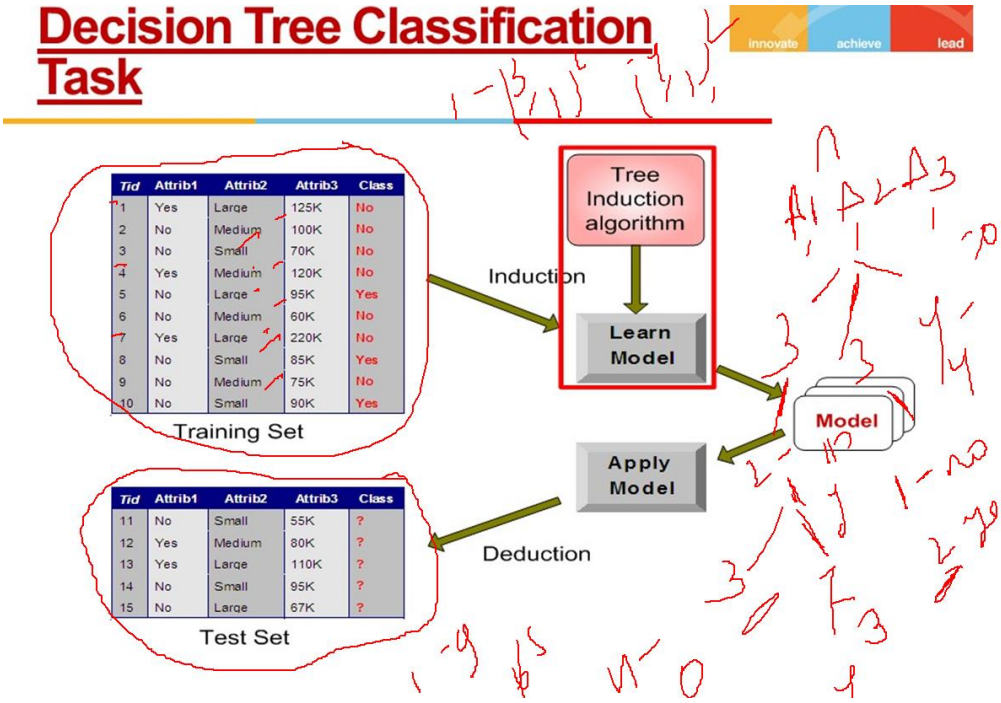
Random subsampling
- Repeated holdout

Cross validation
- Partition data into k disjoint subsets
- k-fold: train on k-1 partitions, test on the remaining one
- Leave-one-out: k=n

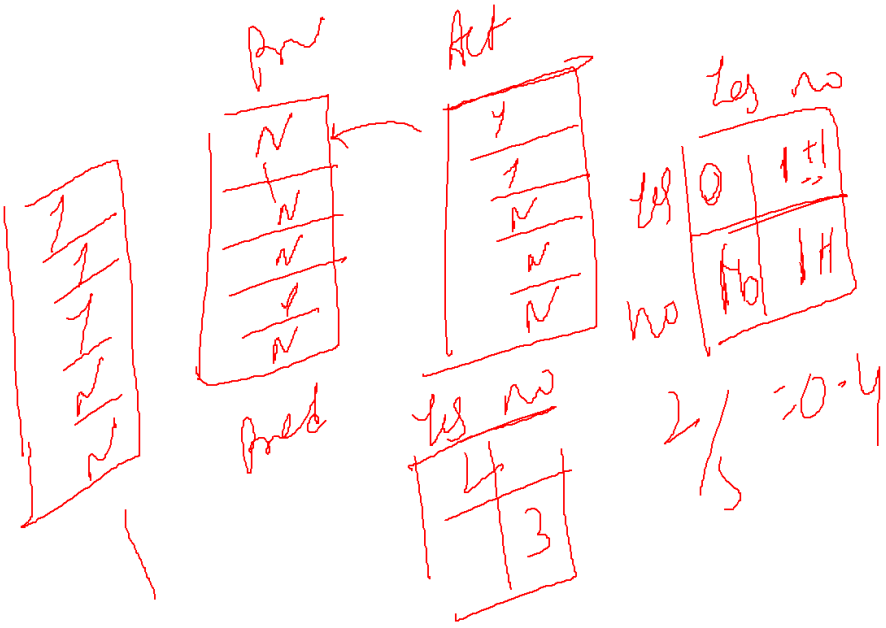
Stratified sampling
- oversampling vs undersampling

Bootstrap
- Sampling with replacement

Cross validation – each data is used for training and testing simultaneously, Holdout – specific data for training and specific for training – performance – based on training data



Until – whole object belongs to same class – splitting



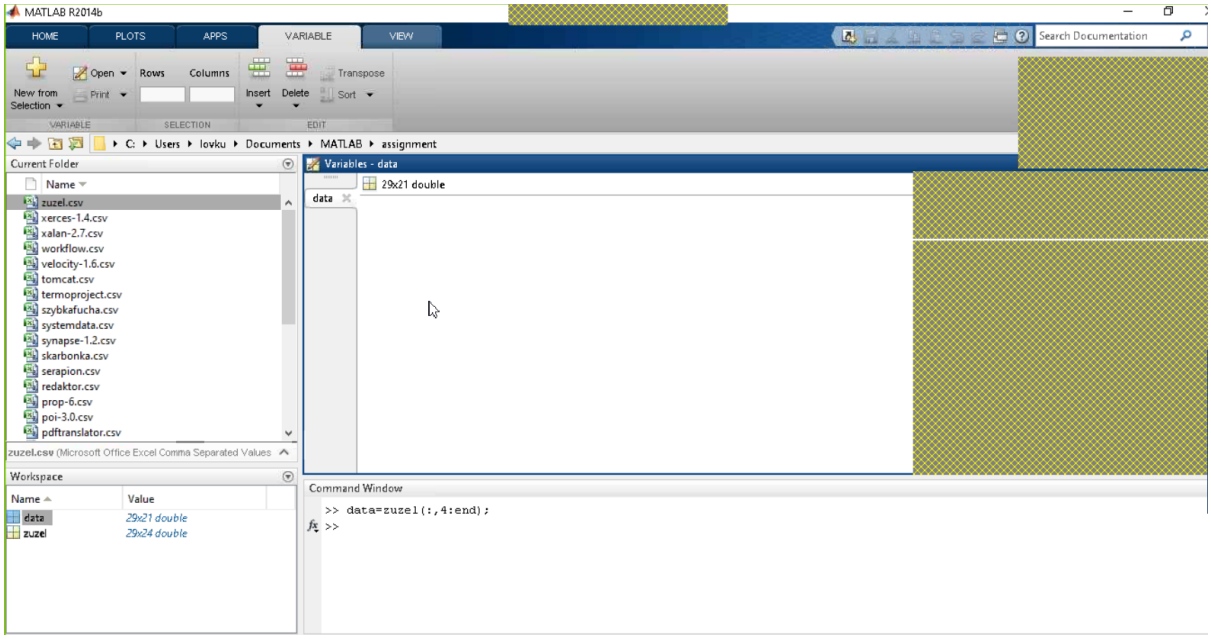
Accuracy – first case – 4/5

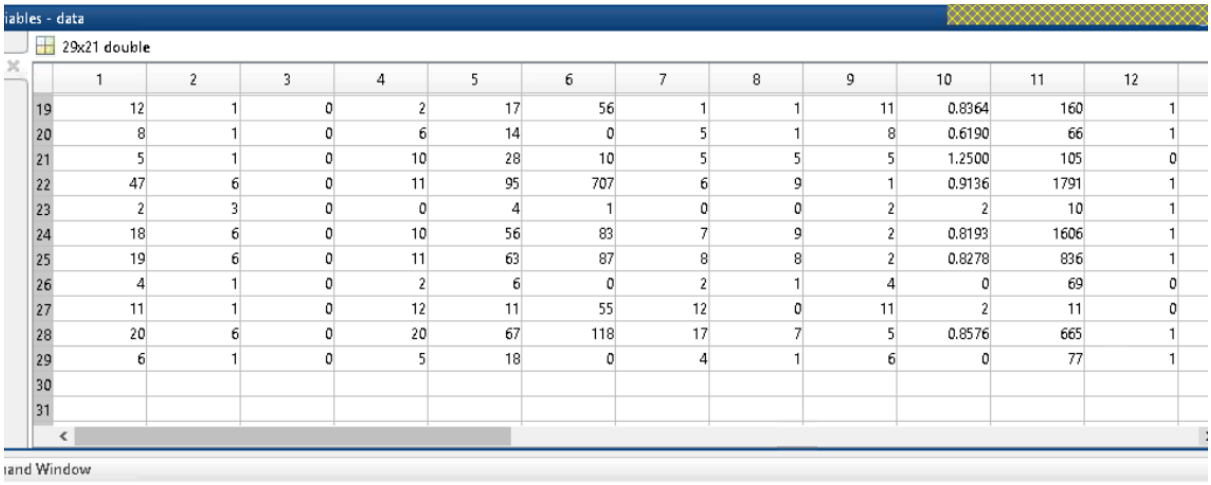
Matrix => [2, 0]

[1, 2]

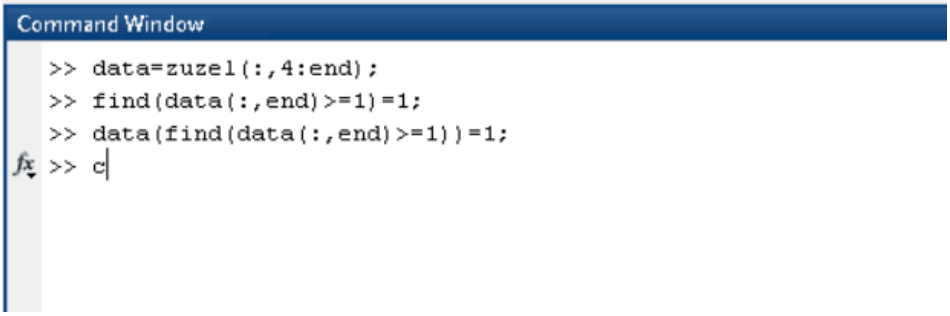
Lec 9 MATLAB Hands on

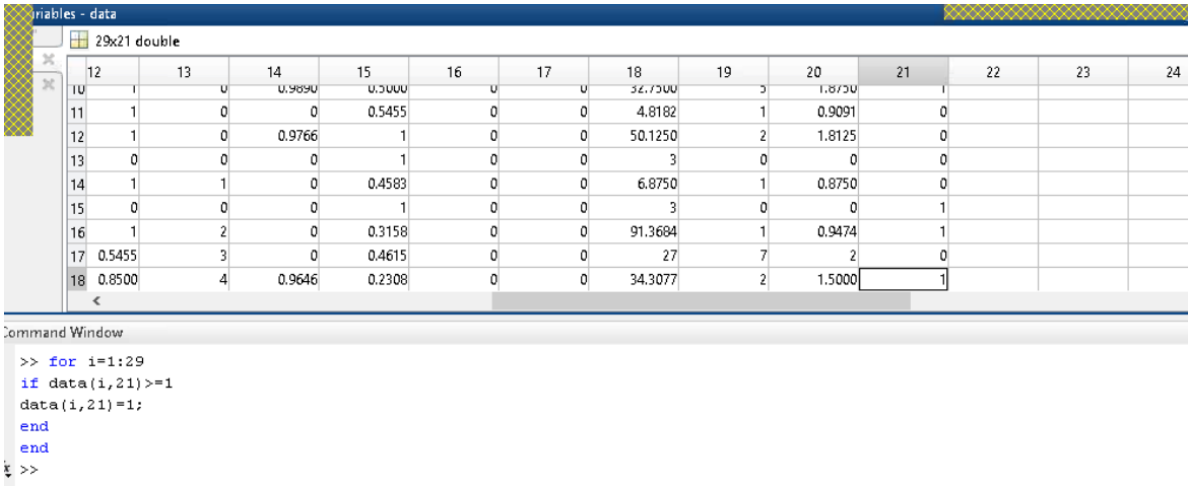
K Mean – clustering – select k value first



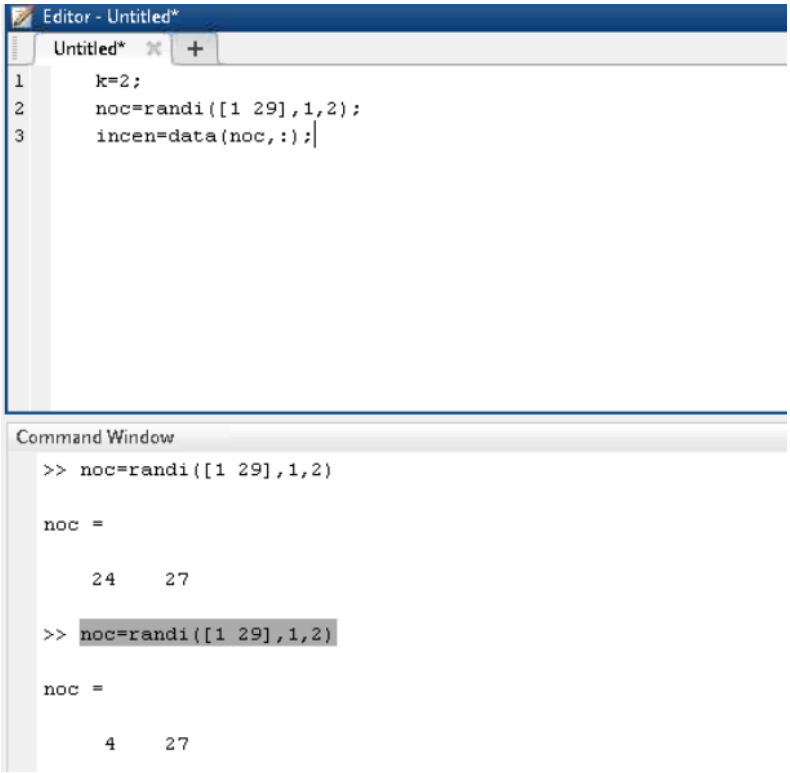


29 – Rows – objects – 20 features – group in class of 2

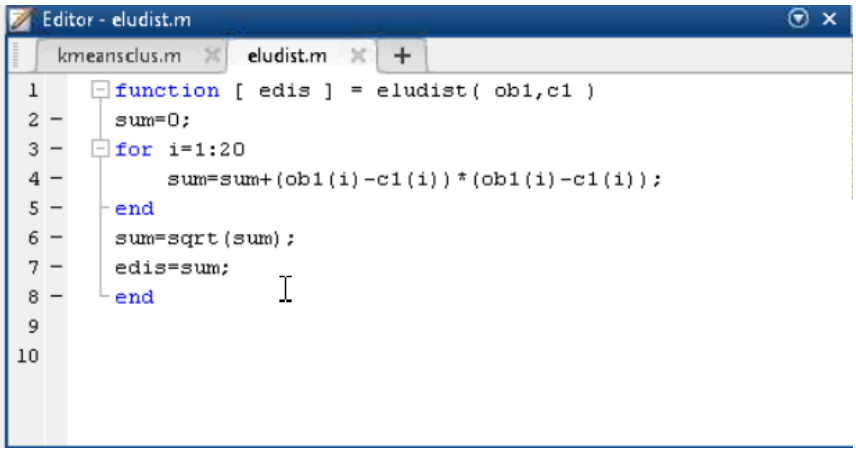


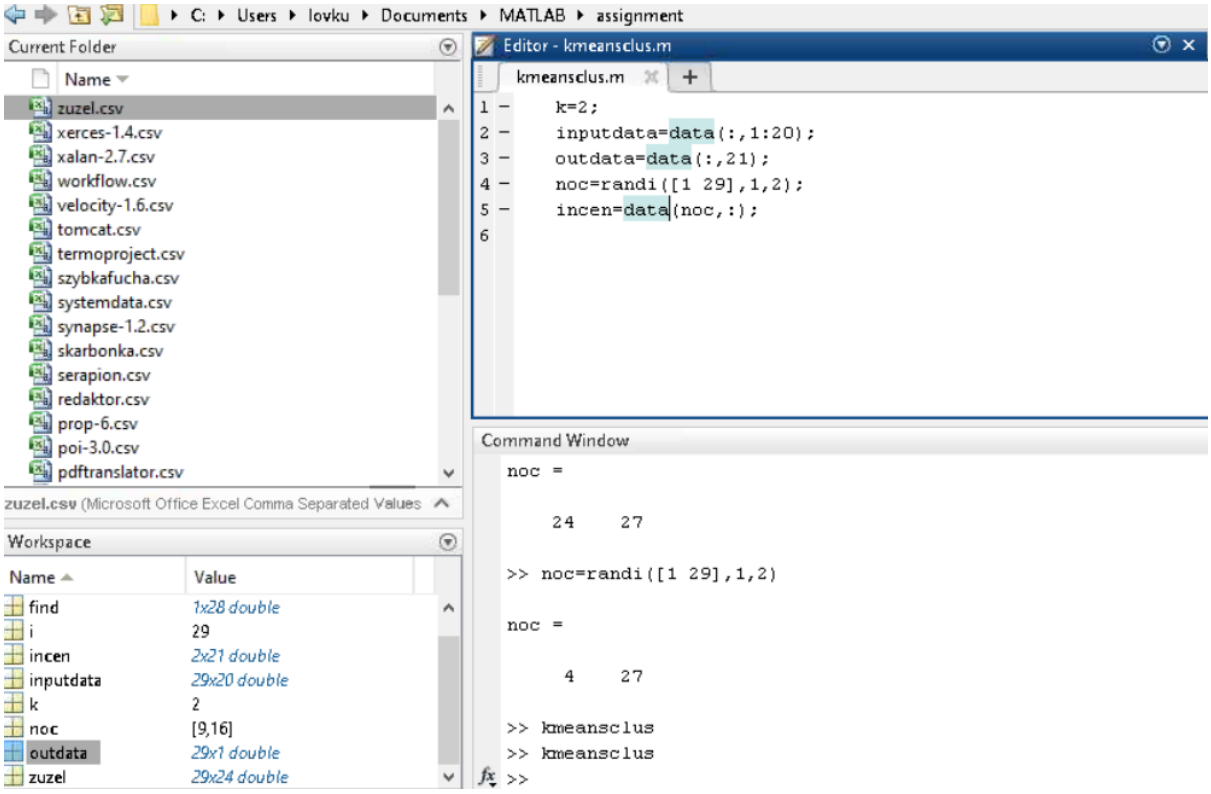


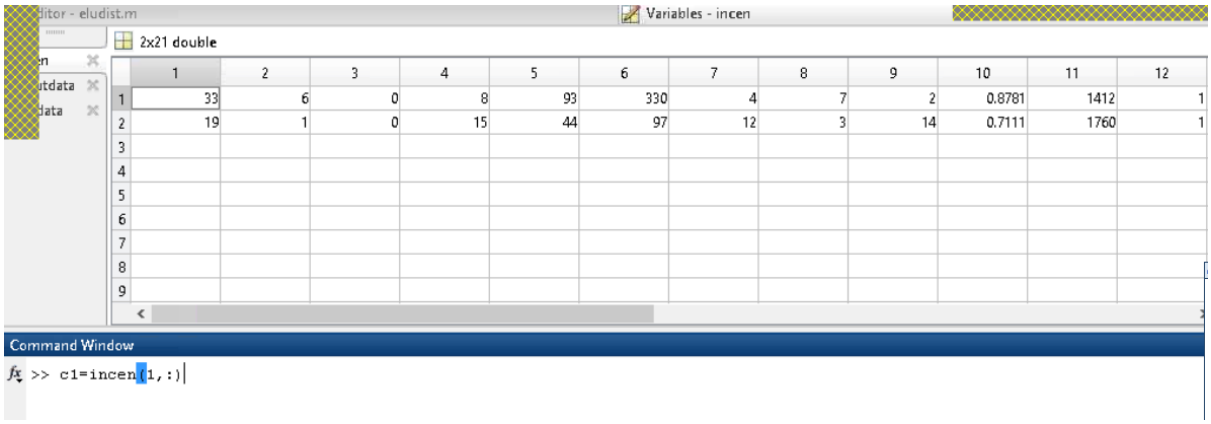
K = 2; randomly generate – centroid

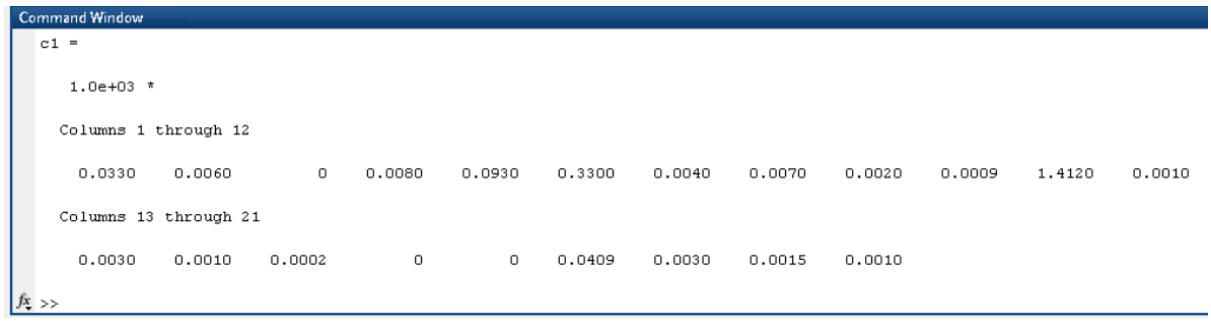


Find Euclidian distance of each object from this two clusters










```
edis =

  503.1946
```

```
Command Window
>> ob1

ob1 =

  Columns 1 through 12

    1.0000    6.0000         0         0   27.0000    7.0000         0         0    1.0000    0.7000  177.0000    1.0000

  Columns 13 through 20

         0    0.9921    1.0000         0         0   27.5000    2.0000    1.5000

fx >>
```

Standard - Training Data – 70%, Rest Testing

```
kmeansclus.m*   eludist.m   +
-     k=2;
-     inputdata=data(:,1:20);
-     outdata=data(:,21);
-     trdata=inputdata(1:20,:);
-     tsdata=inputdata(21:29,:);
-     trouptdata=outdata(1:20);
      tsouptdata=outdata(21:29);



-     noc=randi([1 29],1,2);
-     incen=inputdata(noc,:);
```

```
noc=randi([1 29],1,2);
incen=inputdata(noc,:);

for j=1:20
for i=1:2
    disco(j,i)=eludist(trdata(j,:),incen(i,:));



end
end
```

20x2 double

	1	2	3	4	5
1	257.1877	102.4679			
2	361.9693	18.2489			
3	347.5351	11.3496			
4	339.2459	40.0051			
5	0	356.4911			
6	123.5994	314.8563			
7	1.4921e+03	1.8455e+03			
8	275.5963	196.2523			
9	1.0221e+03	1.3778e+03			

```
neansclus.m*   eludist.m
for i=1:2
    disco(j,i)=eludist(trdata(j,:),incen(i,:));
end
end
c1=0;
for i=1:20
    if disco(i,1)<disco(i,2)
        c1(i)=1;
    else
        c1(i)=2;
    end
end
```

Variables - c1

1x20 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	2	2	2	2	1	1	1	2	1	1	2	1
2												
3												

```
neansclus.m*   eludist.m
c2=0;
l=0;
m=0;
for i=1:20
    if disco(i,1)<disco(i,2)
        l=l+1;
        c1(l,1:20)=trdata(i,:);
    else
        m=m+1;
        c2(m,1:20)=trdata(i,:);
    end
end
```

Variables - c1

9x20 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	15	6	0	7	49	73	3	6	1	0.8631	424	0.916
2	25	1	0	42	40	150	39	3	23	0.7917	347	
3	30	6	0	6	72	199	3	5	2	0.8714	1910	
4	33	6	0	8	93	330	4	7	2	0.8781	1412	
5	8	6	0	5	44	14	5	2	1	0.7619	276	
6	16	6	0	0	42	92	0	0	1	0.8754	841	
7	19	1	0	15	44	97	12	3	14	0.7111	1760	
8	13	1	0	8	36	0	3	5	10	0.7652	375	0.545
9	26	6	0	10	80	189	6	8	3	0.8340	938	0.850

11x20 double

	1	2	3	4	5	6	7	8	9	10	11
1	1	6	0	0	27	7	0	0	1	0.7000	177
2	10	1	0	14	11	3	14	0	10	0.6667	72
3	5	1	0	2	19	0	0	2	5	0	86
4	2	1	0	3	17	1	0	3	2	2	95
5	23	1	0	4	32	171	3	1	23	0.8807	169
6	11	1	0	3	12	25	3	0	11	0.8000	69
7	1	1	0	0	2	0	0	0	1	2	4
8	8	1	0	1	14	0	0	1	8	0.6190	66
9	1	1	0	0	2	0	0	0	1	2	4

```
    c1updata=mean(c1);
    c2updata=mean(c2);

    a=eludist(c1updata,incen(1,:));
    b=eludist(c2updata,incen(2,:));
    if a<=0.05 && b<=0.05
       stop=1;
    end

```

kmeansclus.m | eludist.m

```
    k=2;
    inputdata=data(:,1:20);
    outdata=data(:,21);
    trdata=inputdata(1:20,:);
    tsdata=inputdata(21:29,:);
    trouptdata=outdata(1:20);
    tsouptdata=outdata(21:29);



    noc=randi([1 29],1,2);
    incen=inputdata(noc,:);
```

```
    noc=randi([1 29],1,2);
    incen=inputdata(noc,:);

    stop=0;
    while stop==0

    for j=1:20
    for i=1:2
       disco(j,i)=eludist(trdata(j,:),incen(i,:));
    end
    end
    c1=0;
    c2=0;
```

```
    c2=0;
    l=0;
    m=0;
    for i=1:20
        if disco(i,1)<disco(i,2)
           l=l+1;
           c1(l,1:20)=trdata(i,:);
        else
            m=m+1;
           c2(m,1:20)=trdata(i,:);
        end
    end
```

```
34 -    end
35
36 -    c1updata=mean(c1);
37 -    c2updata=mean(c2);
38
39 -    a=eludist(c1updata,incen(1,:));
40 -    b=eludist(c2updata,incen(2,:));
41 -    if a<=0.05 && b<=0.05
42 -       stop=1;
43 -    end
44 -    incen=[c1updata;c2updata];
45
46 -    end
```

```
39 -    a=eludist(c1updata,incen(1,:));
40 -    b=eludist(c2updata,incen(2,:));
41      count=count+1;
42 -    if (a<=0.05 && b<=0.05) || count==100
43 -       stop=1;
44 -    end
45 -    incen=[c1updata;c2updata];
46
```

```
incen =

  Columns 1 through 12

   20.5556    4.3333         0   11.2222   55.5556  127.1111    8.3333    4.3333    6.3333    0.8169  920.3333    0.9236
    7.4545    1.4545         0    3.1818   15.1818   23.9091    2.3636    0.8182    7.3636    1.0111   88.0000    0.7273

  Columns 13 through 20

    1.8889    0.6465    0.4009         0         0   41.9174    4.5556    1.7937
    0.1818    0.0902    0.5970         0         0   12.5713    1.0909    0.7990
```

```
c2updata =

  Columns 1 through 12

    9.5333    2.0000         0    6.4667   22.4000   33.3333    5.0667    1.6667    7.7333    0.9536  159.3333    0.7641

  Columns 13 through 20

    0.4667    0.1973    0.5378         0         0   15.8041    2.0000    1.0149

fx >>
```

Editor - kmeansclus.m*

kmeansclus.m* | eludist.m

```
43 -       stop=1;
44 -    end
45 -    incen=[c1updata;c2updata];
46 -    end
47
48 -    for j=1:9
49 -       for i=1:2
50 -    discotst(j,i)=eludist(tsdata(j,:),incen(i,:));
51        end
52 -    end
53
54
55
```

Testing data

```
56 -    c1=0;
57 -   for i=1:9
58 -        if discotst(i,1)<discotst(i,2)
59 -          c1(i)=1
60 -        else
61             c1(i)=2;
62 -        end
63 -    end
64
```

```
      c1(i)=1;
    else
      c1(i)=2;
    end
end
>>
>> c1

c1 =

     2     1     2     1     1     2     2     2     2
```

```
c1 =

     2     1     2     1     1     2     2     2     2

>> tsouptdata

tsouptdata =

     1
     1
     0
     1
     1
     0
     0
     1
     0
```

```
58 -        if discotst(i,1)<discotst(i,
59 -          c1(i)=1;
60 -        else
61 -          c1(i)=2;
62 -        end
63 -    end
64
65 -    cmat=zeros(2,2);
66 -   for i=1:9
67 -        if tsouptdata(i)==0
68 -            if c1(i)==1
69 -                cmat(1,1)=cmat(1,1)+1;
70 -            end
71 -        end
```

```
kmeansclus.m*   eludist.m
66 -  for i=1:9
67 -      if tsouptdata(i)==0
68 -          if c1(i)==1
69 -              cmat(1,1)=cmat(1,1)+
70 -          else
71 -              cmat(1,2)=cmat(1,2)+1;
72 -          end
73 -      elseif tsouptdata(i)==1
74 -          if c1(i)==2
75 -              cmat(2,2)=cmat(2,2)+1;
76 -          else
77 -              cmat(2,1)=cmat(2,1)+1;
78 -          end
79 -      end
```

```
>> cmat

cmat =

     0     4
     3     2

>>
```

Random centers

```
cmat =

     4     0
     2     3

>>
```

Random centers

Accuracy – depends on initial centroids, threshold value / iterations

Precision – How many classes are classified as 1 class

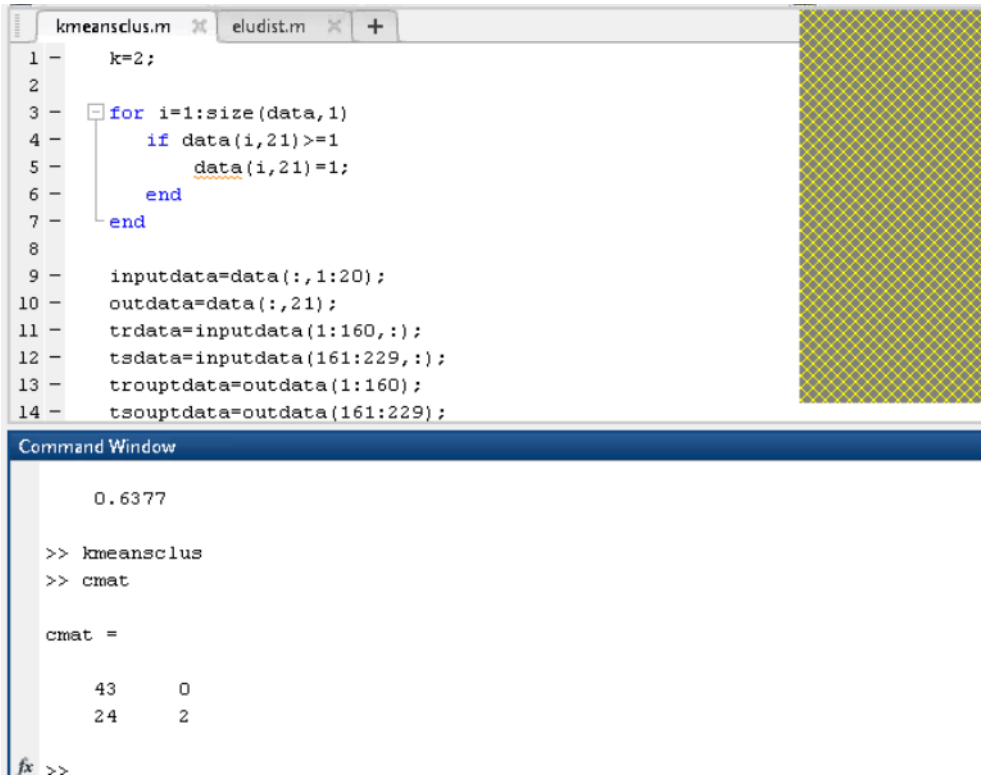
Accuracy – How many classes accurately classified

Recall – How many classes are classified as 0 class

Proj1	77	90	80	87											
			Kmean			FCM				HC				DT		
Projname	Acc	Pre	Recall	F-Measure	Acc	Pre	Recall	F-Measure	Acc	Pre	Recall	F-Measure	Acc	Pre	Recall	F-Measure
Proj1	77.8	98	78	87												

Cross validation – k fold → k-1 sets for training, 1 for testing

```
>> data=velocity1(:,4:end);
>> 229*0.7

ans =

  160.3000

>> 229-160
```



Association Analysis: Basic Concepts and Algorithms

Association Rule Mining

innovate achieve lead

- Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction

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

Marketing purpose – if Person buys product A then he will buy product B

Definition: Frequent Itemset

innovate achieve lead

- Itemset
  - A collection of one or more items
    - **Example: {Milk, Bread, Diaper}**
  - k-itemset
    - **An itemset that contains k items**
- Support count (σ)
  - Frequency of occurrence of an itemset
    - **E.g. σ({Milk, Bread,Diaper}) = 2**
- Support
  - Fraction of transactions that contain an itemset
    - **E.g. s({Milk, Bread, Diaper}) = 2/5**
- Frequent Itemset
  - An itemset whose support is greater than or equal to a minsup threshold

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

Support count – how many times particular item set is appearing

Definition: Association Rule

innovate achieve lead

- Association Rule
  - An implication expression of the form X → Y, where X and Y are itemsets
    - **Example: {Milk, Diaper} → {Beer}**
- Rule Evaluation Metrics
  - Support (s)
    - **Fraction of transactions that contain both X and Y**
  - Confidence (c)
    - **Measures how often items in Y appear in transactions that contain X**

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

Example:

$$\{\text{Milk, Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk, Diaper, Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

Support of association rule X → Y = (X U Y) / Total number of support count

Confidence of association rule → (X U Y) / X

Association Rule Mining Task

innovate achieve lead

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ minsup threshold
  - confidence ≥ minconf threshold
- Brute-force approach:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Prune rules that fail the minsup and minconf thresholds

Mining Association Rules

innovate achieve lead

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

Example of Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

Observations:

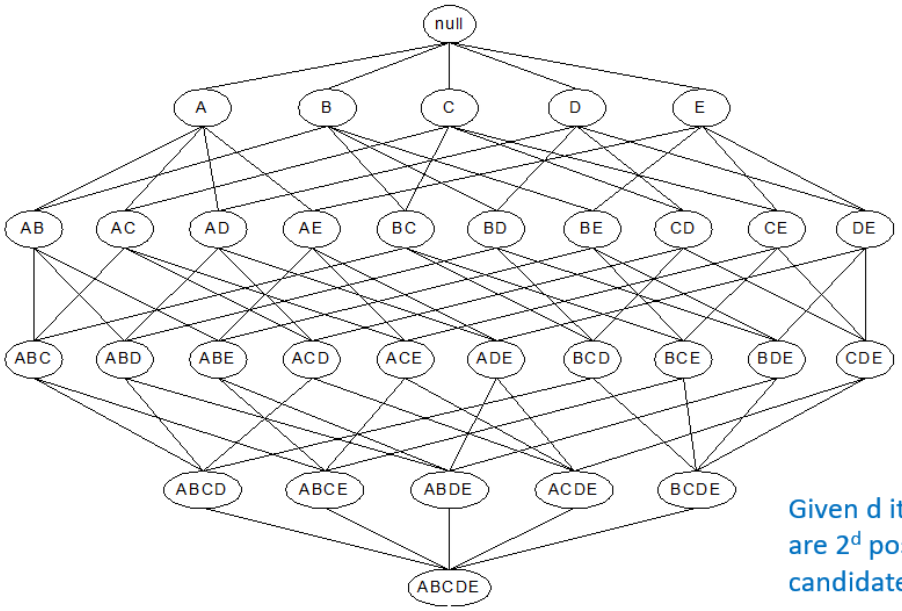
- All the above rules are binary partitions of the same itemset: {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

Mining Association Rules

innovate achieve lead

- Two-step approach:
  - Frequent Itemset Generation
    - **Generate all itemsets whose support ≥ minsup**
  - Rule Generation
    - **Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset**
- Frequent itemset generation is still computationally expensive

Frequent Itemset Generation

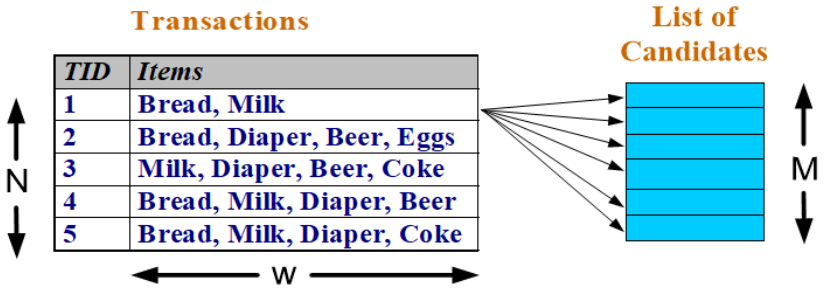


Given d items, there are 2^d possible candidate itemsets

Consider item set whose support count >= min

Frequent Itemset Generation

- Brute-force approach:
  - Count the support of each candidate by scanning the database



  - Match each transaction against every candidate
  - Complexity ~ O(NMw) => Expensive since M = 2d !!!

Frequent Itemset Generation Strategies

- Reduce the number of candidates (M)
  - 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

Reducing Number of Candidates

Apriori principle:

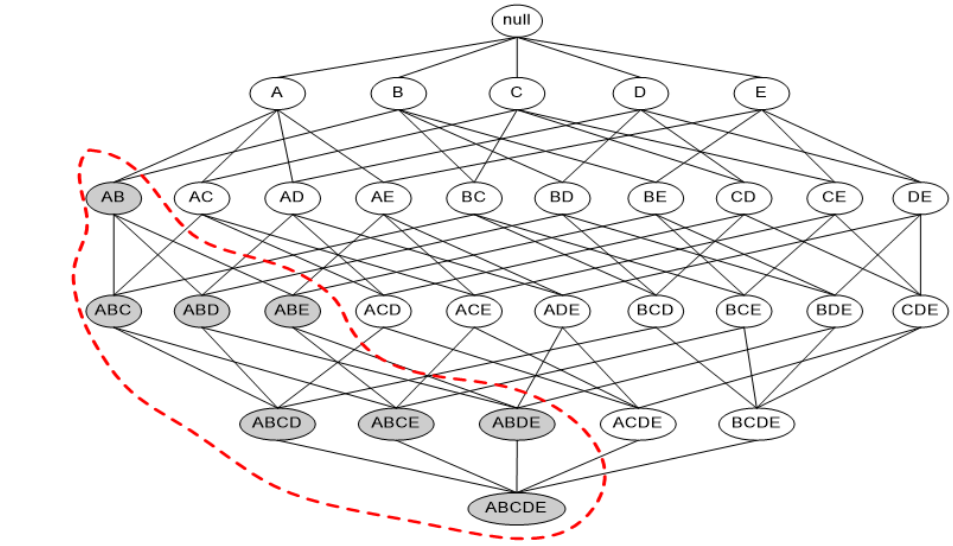
- If an itemset is frequent, then all of its subsets must also be frequent
- Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

  - Support of an itemset never exceeds the support of its subsets

Support {Bread} > {Bread,Milk}

Illustrating Apriori Principle



Illustrating Apriori Principle

Item	Count
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Minimum Support = 3

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

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

If every subset is considered,
$^6C_1 + {}^6C_2 + {}^6C_3 = 41$
With support-based pruning,
$6 + 6 + 1 = 13$

Itemset	Count
{Bread,Milk,Diaper}	3

Support count > minimum support → frequent item

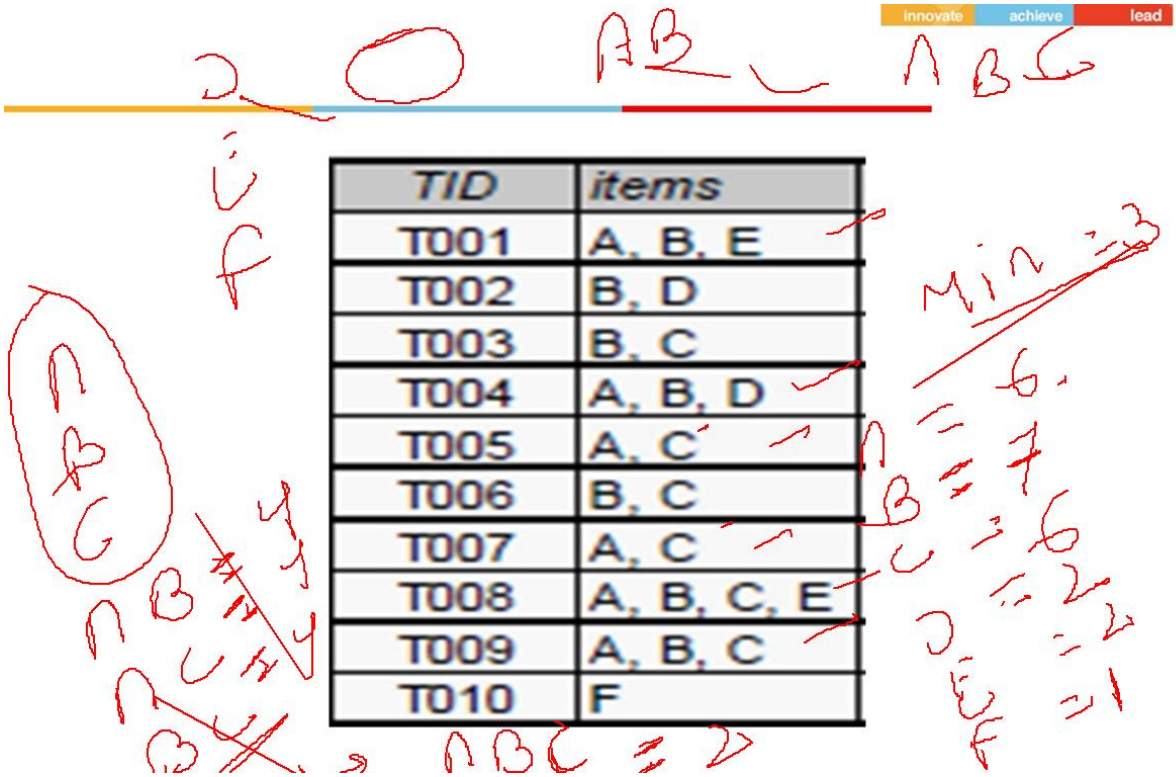
Apriori Algorithm

Method:

- Let k=1
- Generate frequent itemsets of length 1
- Repeat until no new frequent itemsets are identified
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Prune candidate itemsets containing subsets of length k that are infrequent
  - Count the support of each candidate by scanning the DB
  - Eliminate candidates that are infrequent, leaving only those that are frequent

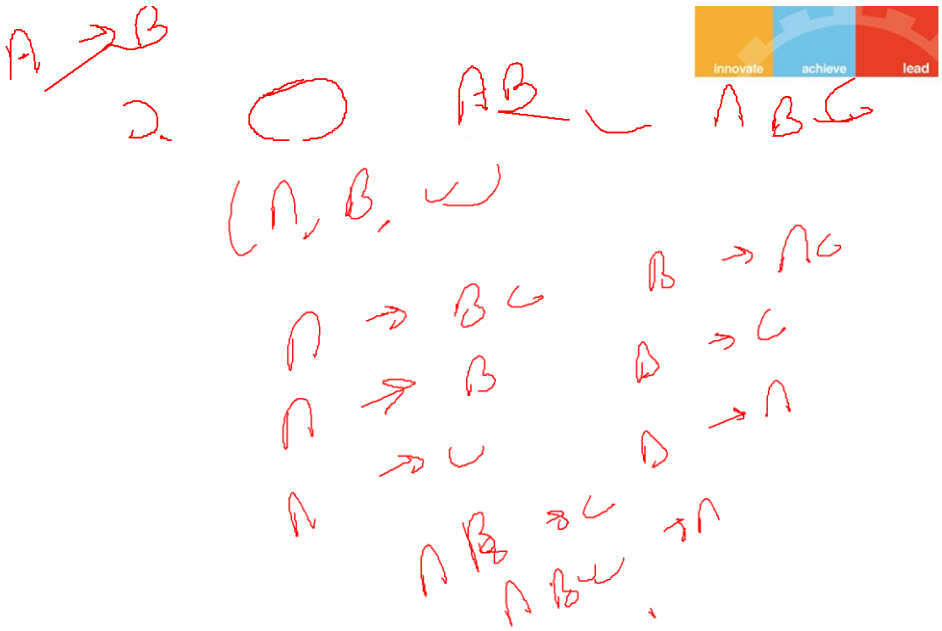
TID	*items*
T001	A, B, E
T002	B, D
T003	B, C
T004	A, B, D
T005	A, C
T006	B, C
T007	A, C
T008	A, B, C, E
T009	A, B, C
T010	F

Lec 10 Association Analysis: Basic Concepts & Algorithms cont…



TID	items
T001	A, B, E
T002	B, D
T003	B, C
T004	A, B, D
T005	A, C
T006	B, C
T007	A, C
T008	A, B, C, E
T009	A, B, C
T010	F

If {A, B}, {B, C}, {A, C} are frequent then only consider {A, B, C}



Find all combinations – confidence of each rules – if c > min confidence – consider, i.e. min = 0.6

Confidence of A → B → Support (A, B) / Support (A) = 4 / 6 > 0.6 → strong rule

Minimum support and minimum confidence – depends on user

c({B,C}) → 4 / 7, c({A,C}) = 4 / 6, c({C,A}) = 4 / 6 → strong rules

Always take high minimum support (frequent items) and high minimum confidence

Bayes Classifier

- A probabilistic framework for solving classification problems
- Conditional Probability:

$$P(C \mid A) = \frac{P(A,C)}{P(A)}$$

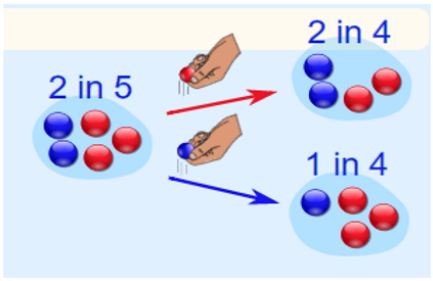
$$P(A \mid C) = \frac{P(A,C)}{P(C)}$$

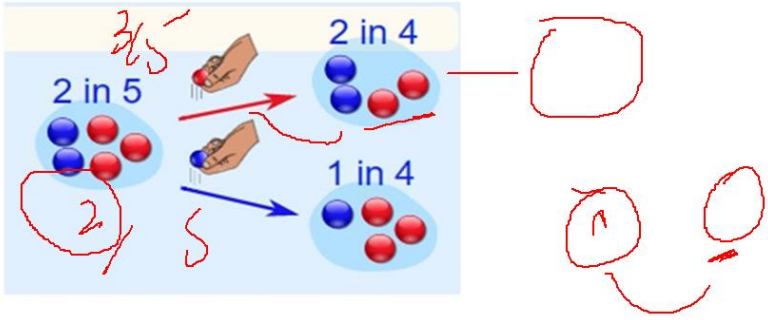
- Bayes theorem:

$$P(C \mid A) = \frac{P(A \mid C)P(C)}{P(A)}$$

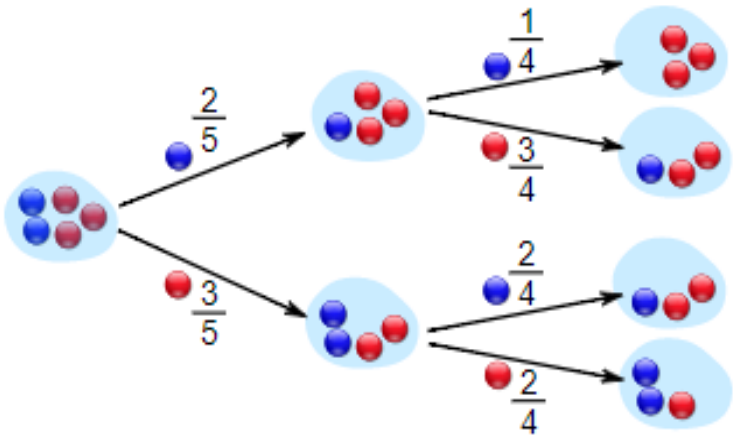
Example: Marbles in a Bag

- Dependent Events

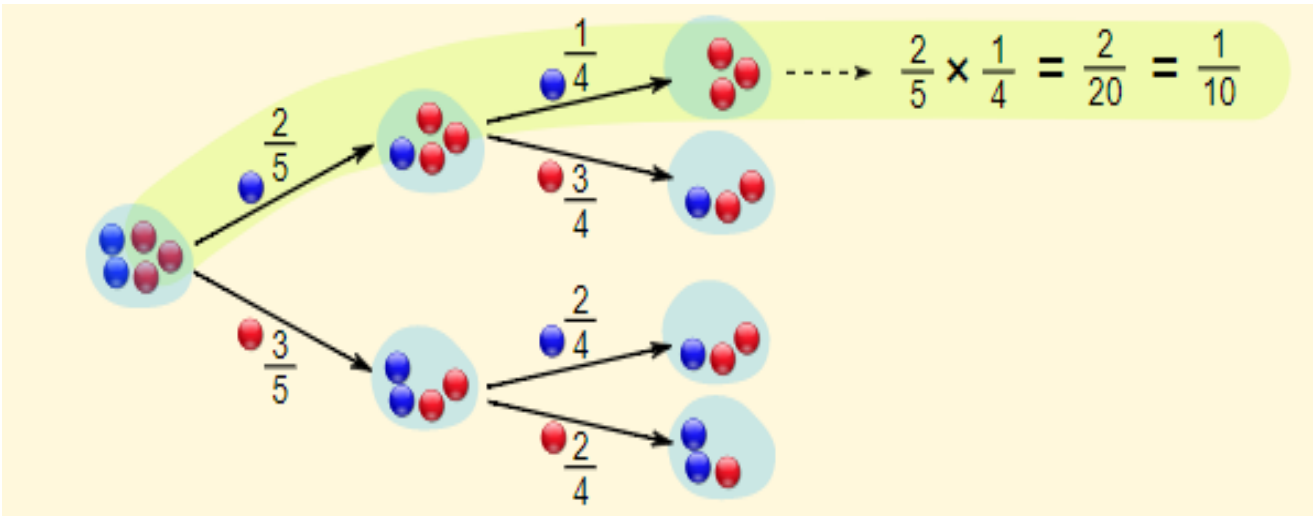




Toss a coin – independent event, next event is not dependent on previous event

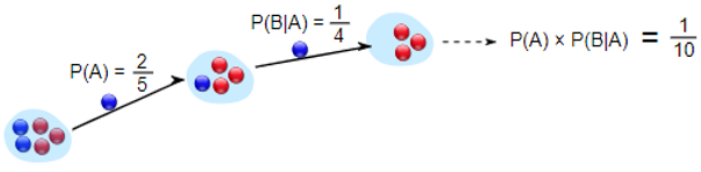


What are the chances of drawing 2 blue marbles?



Notation

- P(A) means "Probability Of Event A"
- P(B|A) means "Event B given Event A"
  - Or event A has already happened, now what is the chance of event B?
  - P(B|A) is also called the "Conditional Probability" of B given A.



"Probability Of" "Given"

$$P(A \text{ and } B) = P(A) \times P(B \mid A)$$

Event A Event B

$$P(B \mid A) = \frac{P(A \text{ and } B)}{P(A)}$$

Bayesian Classifiers

- Consider each attribute and class label as random variables
- Given a record with attributes (A1, A2,…,An)
- Goal is to predict class C
- Specifically, we want to find the value of C that maximizes P(C| A1, A2,…,An)
- Can we estimate P(C| A1, A2,…,An) directly from data?

Approach:

– compute the posterior probability P(C | A1, A2, …, An) for all values of C using the Bayes theorem

$$P(C \mid A_1 A_2 \mathrm{K}\ A_n) = \frac{P(A_1 A_2 \mathrm{K}\ A_n \mid C) P(C)}{P(A_1 A_2 \mathrm{K}\ A_n)}$$

– Choose value of C that maximizes

$P(C \mid A_1, A_2, \ldots, A_n)$

– Equivalent to choosing value of C that maximizes

$P(A_1, A_2, \ldots, A_n|C)\ P(C)$

How to estimate P(A1, A2, …, An | C)?

- Assume independence among attributes Ai when class is given:
- P(A1, A2, …, An |C) = P(A1| Cj) P(A2| Cj)… P(An| Cj)

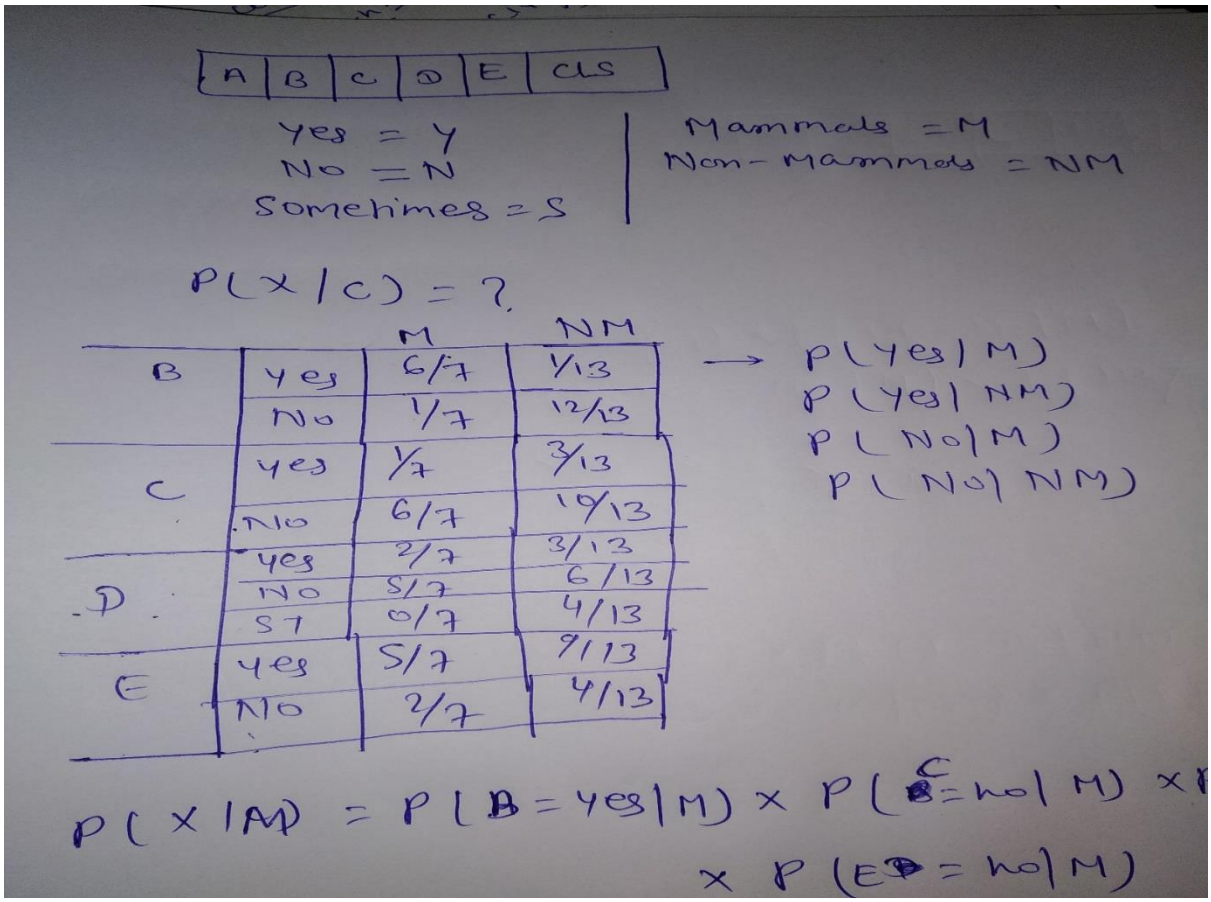
- Estimate P(Ai| Cj) for all Ai and Cj.
- New point is classified to C_j if $P(C_j) \prod P(A_i| C_j)$ is maximal.

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

P(CLASS=MAMMALS/FEATURES) > P(CLASS=NON MAMMALS/FEATURES) → CLASS =MAMMALS

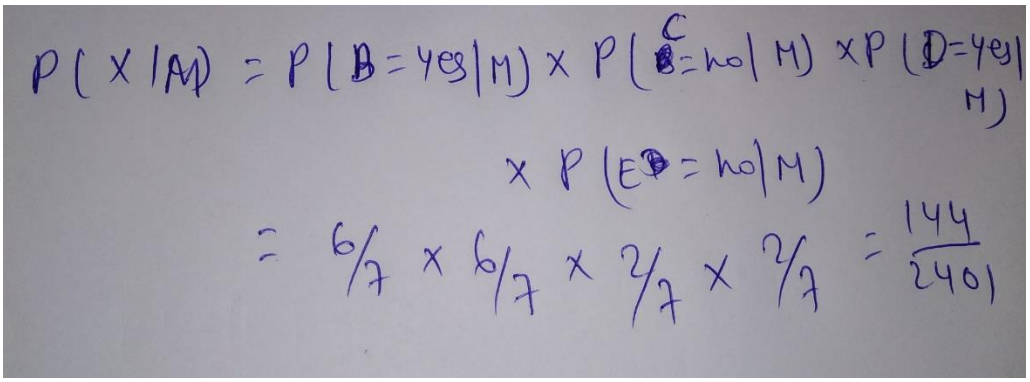
P(CLASS=MAMMALS/FEATURES), P(FEATURES/MAMMALS)

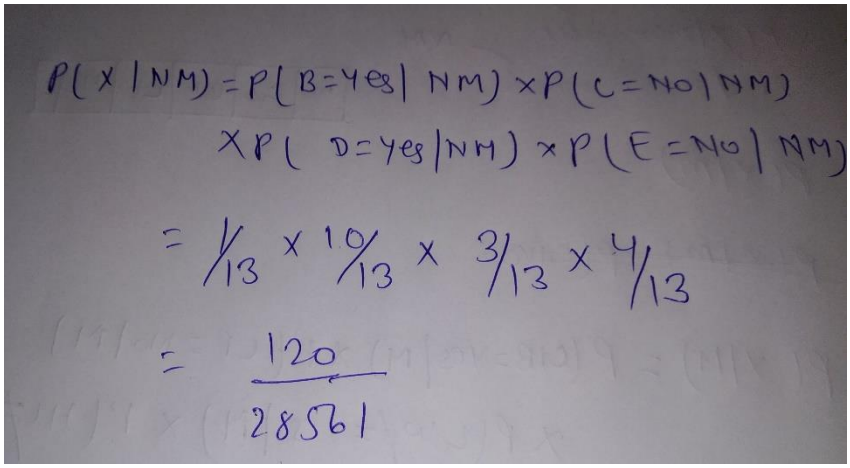


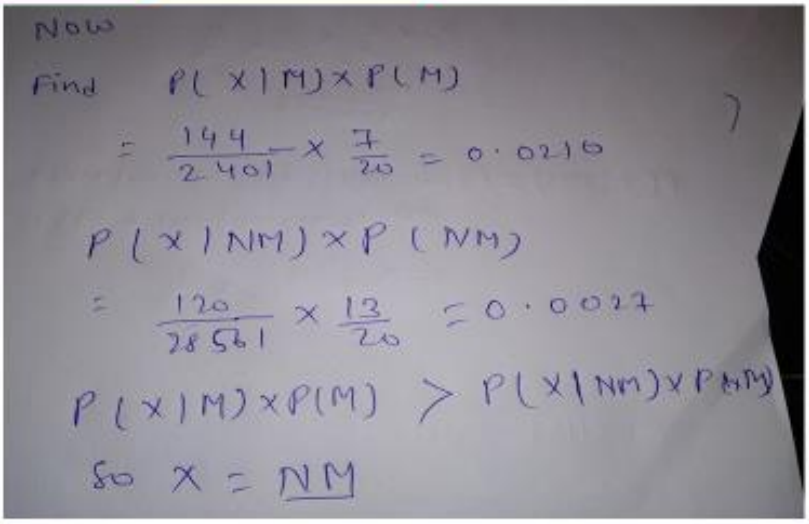
P(X|M)=P(M/X)*P(X)/P(M), P(X|MAMMALS), P(X|NON-MAMMALS)

P(X|MAMMALS)= P(GIVE BIRTH=YES|MAMMALS) * P(CAN FLY=NO|MAMMALS) * P(LIVE IN WATER=YES|MAMMALS) * P(HAVE LEGS=NO|MAMMALS) – probability which maximizes – M

P(X|NONMAMMALS) = P(GIVE BIRTH=YES|NONMAMMALS) * P(CAN FLY = NO|NON MAMMALS) * P(LIVE IN WATER = YES|NONMAMMALS) * P(HAVE LEGS=NO|NONMAMMALS)







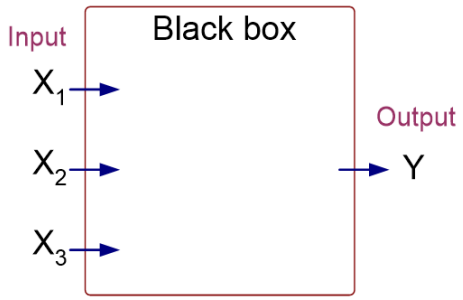
P(C|features) = P(feature|c=yes) * P(c=yes)/P(features) → both expr → P(features) common - skip

P(C=no|features) = P(feature|C=no) * P(C=no)/P(features)

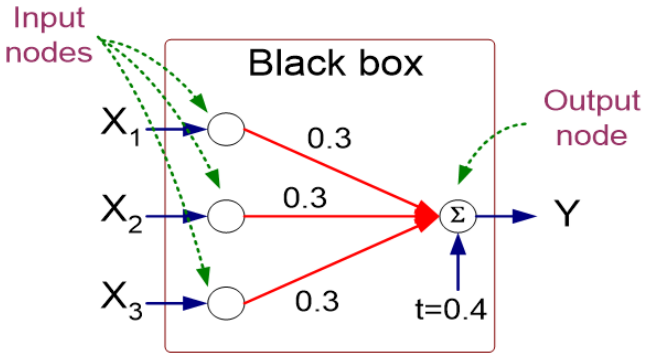
Continues data – convert them in discrete values

Artificial Neural Networks (ANN)

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



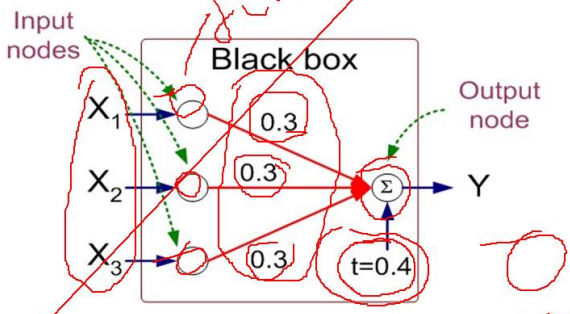
X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0



$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

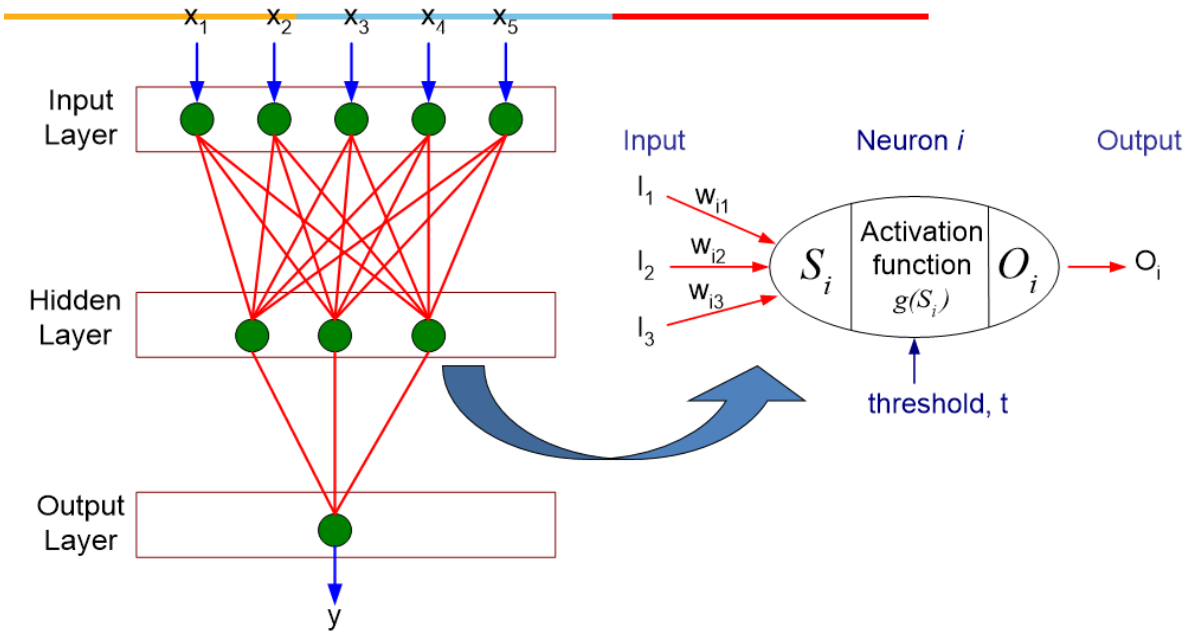
$$\text{where} \quad I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

X_1	X_2	X_3	Y
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	0
0	1	0	0
0	1	1	1
0	0	0	0

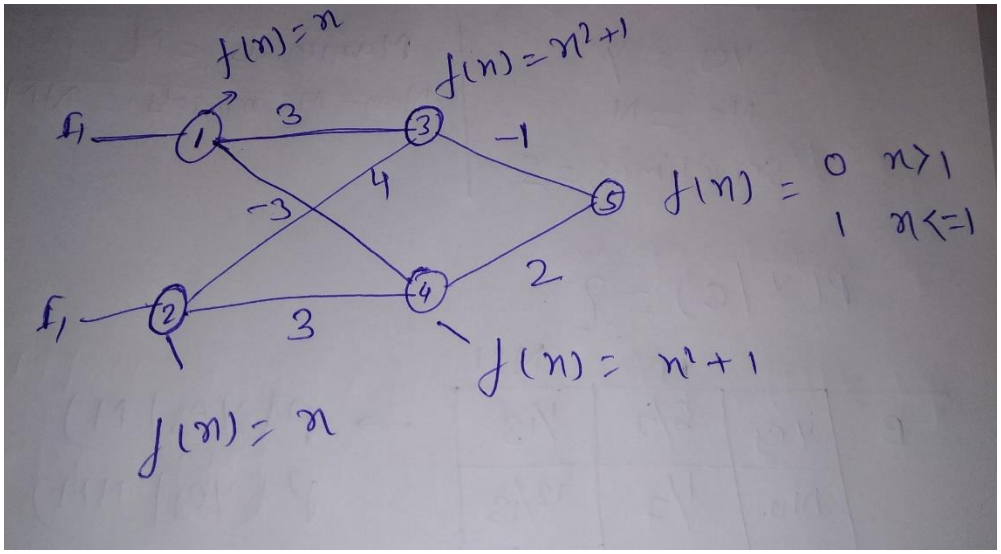


$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

$$\text{where} \quad I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$



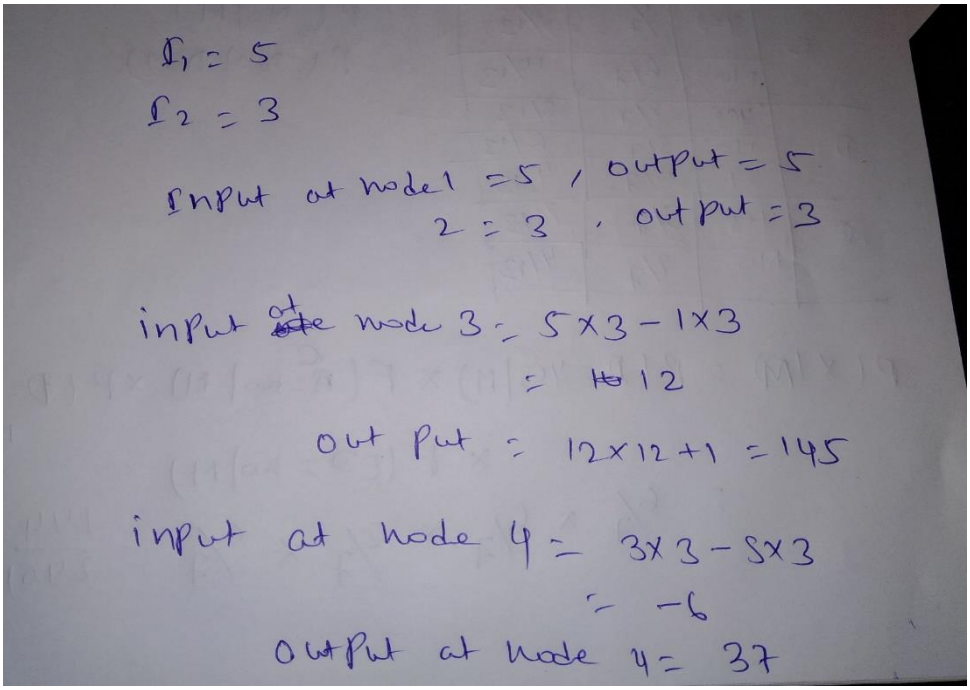
- Initialize the weights (w0, w1, …, wk)
- Adjust the weights in such a way that the output of ANN is consistent with class labels of training examples
- Objective function: $E = \sum_i [Y_i - f(w_i, X_i)]^2$
- Find the weights wi's that minimize the above objective function

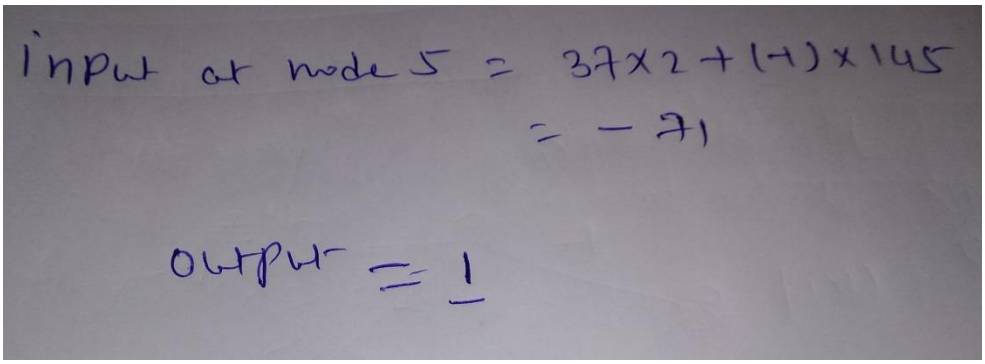


More than one neuron and more than one hidden layer

Architecture – 2 - 2 – 1 → Input node – hidden node – output node

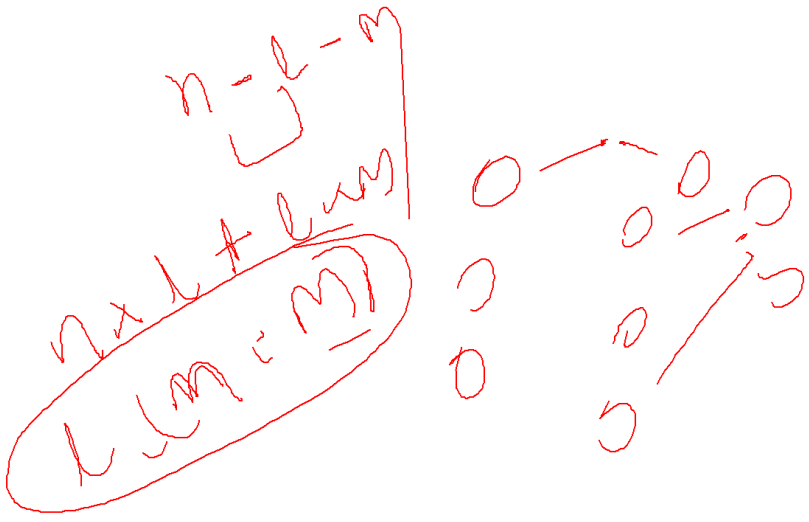
All nodes are connected to each other





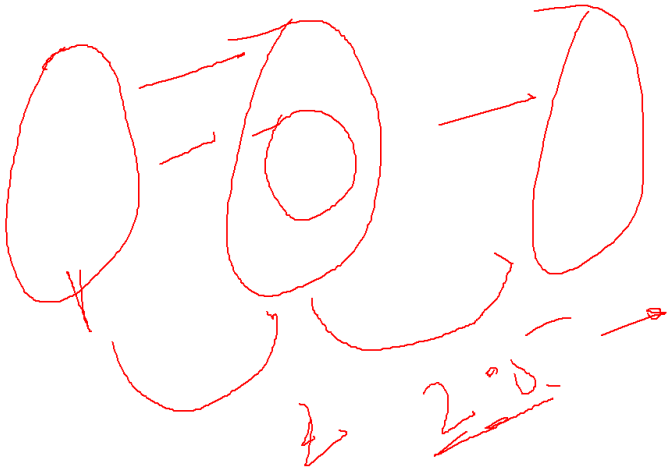
Prediction – we want to find exact value

N – L – M → 5 – 6 – 1, (5*6) + (6*1) = 36



Req. of hidden nodes – Normal/liner/polynomial regression → $y = mx + c$ or $y = mx^2 + nx + c$

If y1 and y2 are not linearly correlated – we can't find model using liner regression – NN – hidden N



Weights – transfer input as we are getting good results – increase hidden nodes to get good R

No. of layers increase – performance increases – complexity increases – trade off

Hidden nodes varies from - no. of hidden nodes to 2 * no. of input nodes - standard

$$\mathrm{RMSE} = \sqrt{\sum \frac{(y_{pred} - y_{ref})^2}{N}}$$

$$MAE = \frac{\sum_{i=1}^{n} |p_i - a_i|}{n} \qquad MARE = \frac{1}{n} \sum_{i=1}^{n} \frac{|w_o - w_p|}{w_o} \qquad (9)$$

NN – used for classification and predication both

Parameters used to validate performance of NN – evolution of prediction model

RMSE – root mean square error, MAE - Mean absolute error

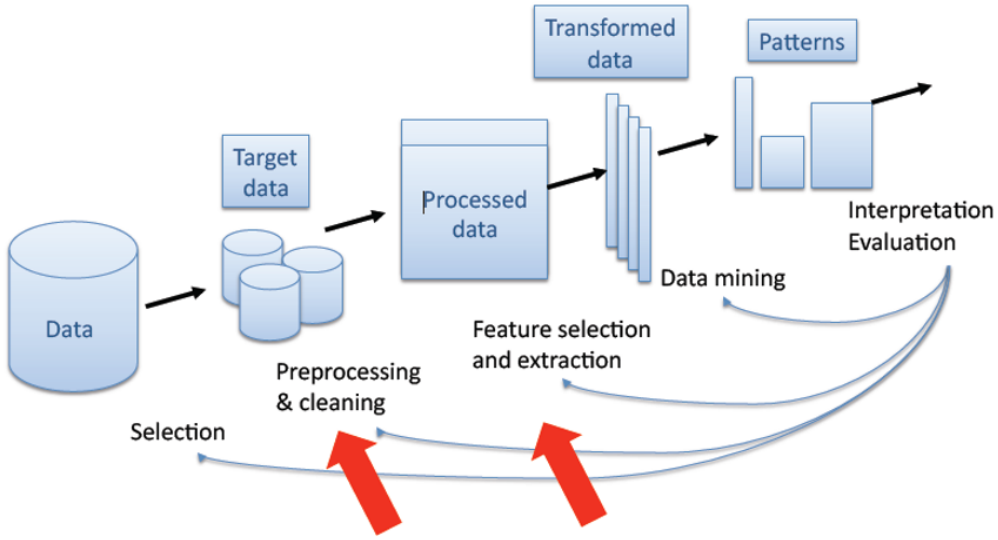
Lec 11 Feature Selection, Dimensionality Reduction

Collection of data - Rows – records – objects, column – features that describes objects

Features selection – not all features are important – select optimal/right set of features

Dimension reduction – reduce size of data (Reduce features)

Feature Selection, Dimensionality Reduction



- In the presence of millions of features/attributes/inputs/variables, select the most relevant ones.

- Why we need FS:
  - to improve performance (in terms of speed, predictive power, simplicity of the model).
  - to visualize the data for model selection.
  - To reduce dimensionality and remove noise.

- Feature Selection is a process that chooses an optimal subset of features according to a certain criterion.

- Reasons for performing FS may include:
  - removing irrelevant data.
  - increasing predictive accuracy of learned models.
  - improving learning efficiency, such as reducing storage requirements and computational cost.
  - reducing the complexity of the resulting model description, improving the understanding of the data and the model.

Steps

- searching for the best subset of features.
- criteria for evaluating different subsets.
- principle for selecting, adding, removing or changing new features during the search.

- Feature selection methods can be broadly classified into two subclasses:
  - Feature ranking methods: In Feature ranking method, some decisive factors have been considered to rank each individual feature and then some of these features are selected which are suitable for a given project.
  - Feature subset selection method: In feature subset selection, subset of features are searched which collectively have good predictive capability.

For every feature find performance parameter – based on that rank the feature (univariate)

Feature subset – worst case consider all the possible combinations

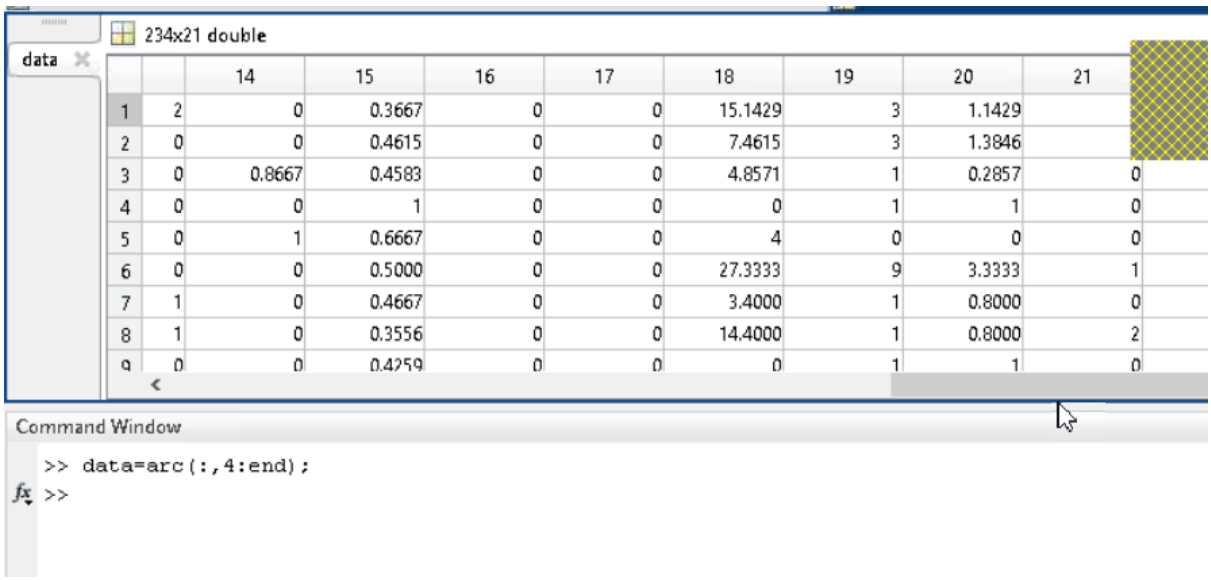
Feature ranking methods:

- Feature ranking methods rank features independently without using any learning algorithm.
- In feature ranking methods, ranking of features are based on the score of the features.
- Further top [log2 n] metrics out of "n" number of metrics have been considered to develop a model.
- Example

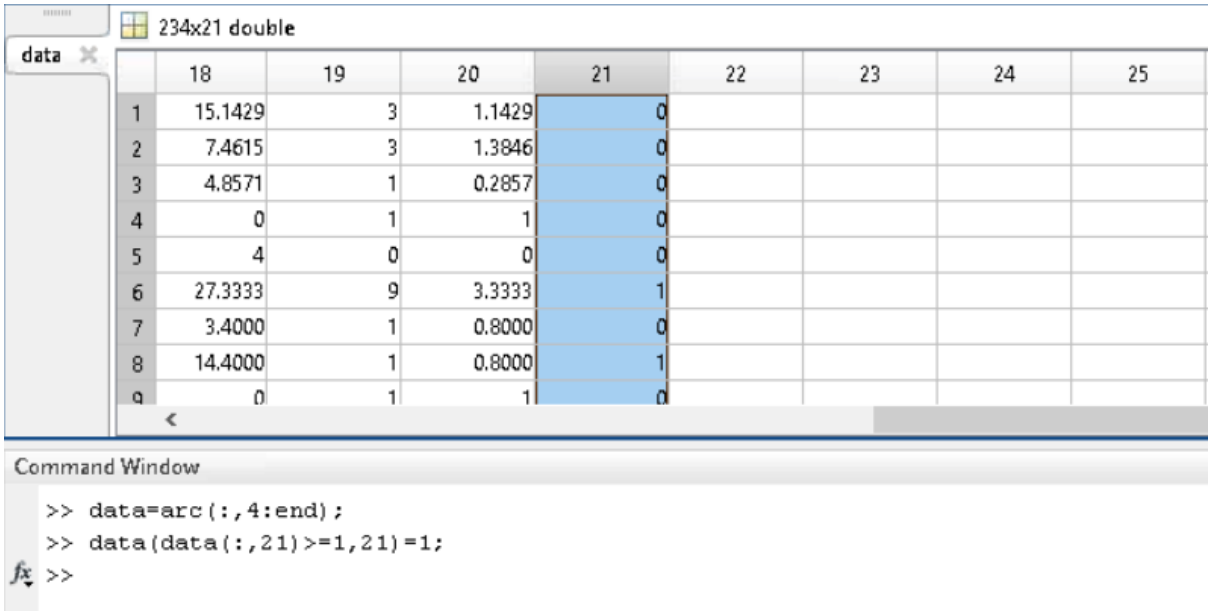
$$GINI_{Feature} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

$$GINI(t) = 1 - \sum_{j} [p(j \mid t)]^2$$

p(j | t) is the relative frequency of class j at node t

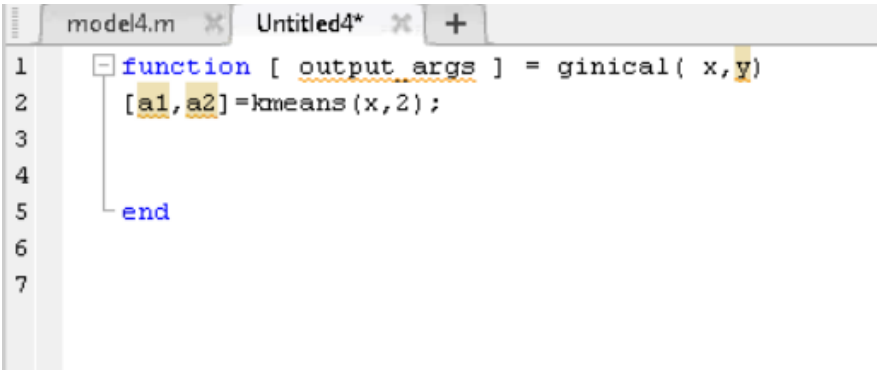


20 features – independent – 1 output



Select best set of features – using GINI index – High GINI index – good feature

Continuous data – discrete data – K mean clustering or mean value technique (value > mean or not)



234x21 double

data | a1 | a2

	1	2	3	4	5	6	7	8
88	2	1	0	2	3	1	0	2
89	7	1	0	4	7	21	2	2
90	4	1	0	2	4	6	2	0
91	22	1	0	16	114	165	0	16
92	21	1	0	3	24	184	3	0
93	3	1	0	4	3	3	3	1
94	16	4	0	21	56	78	0	21
95	26	1	0	11	65	0	0	11
96	5	1	0	5	10	4	0	5

Command Window

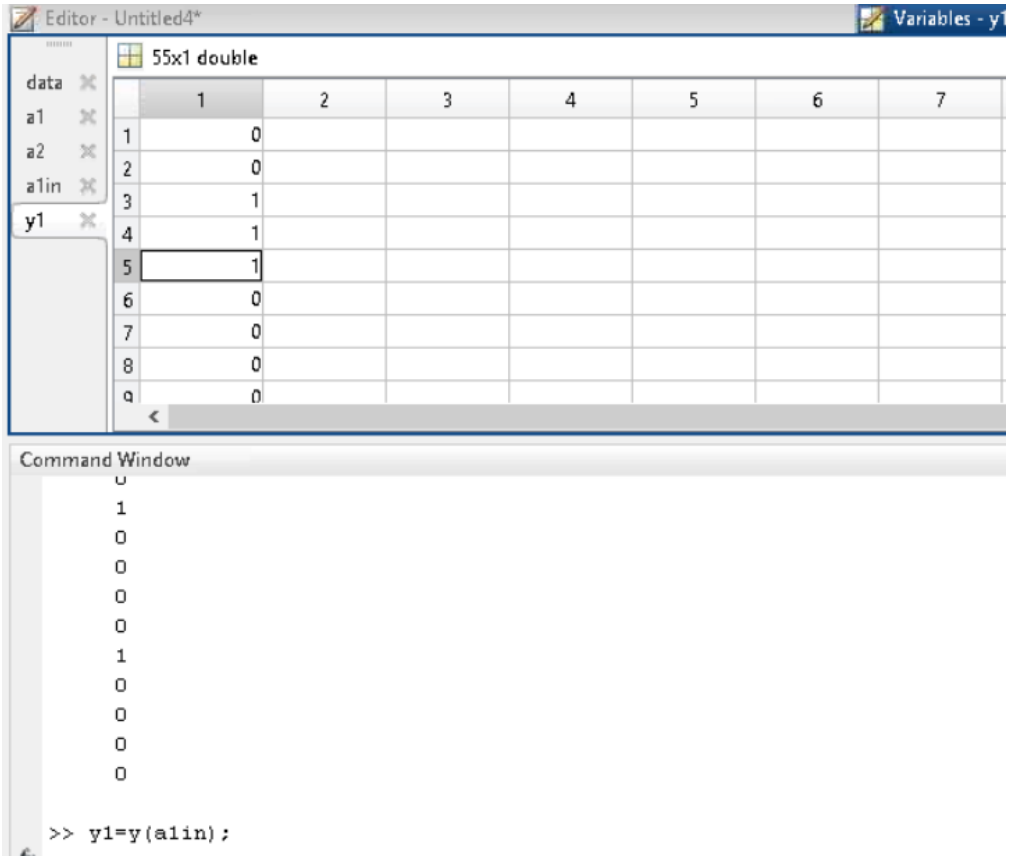
```
     2
     1
     2
     2
     1
     2


a2 =

   22.6727
    5.0950
```

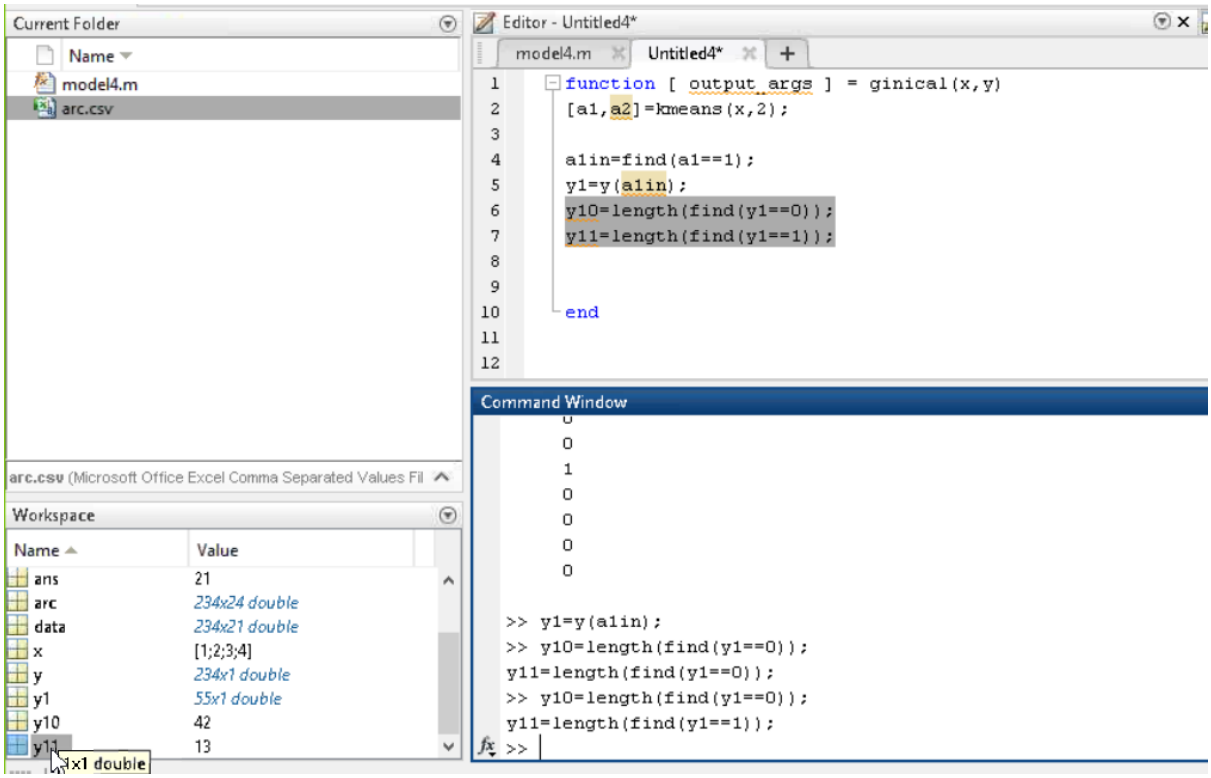
model4.m | Untitled4*

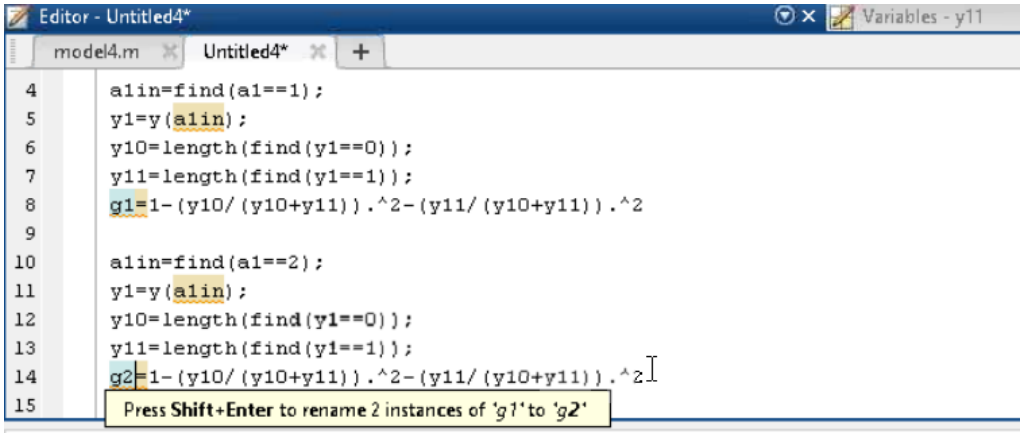
```
1   function [ output_args ] = ginical(x,y)
2   [a1,a2]=kmeans(x,2);
3
4   a1in=find(a1==1);
5   y1=y(a1in);
6
7
8
9   end
10
11
```

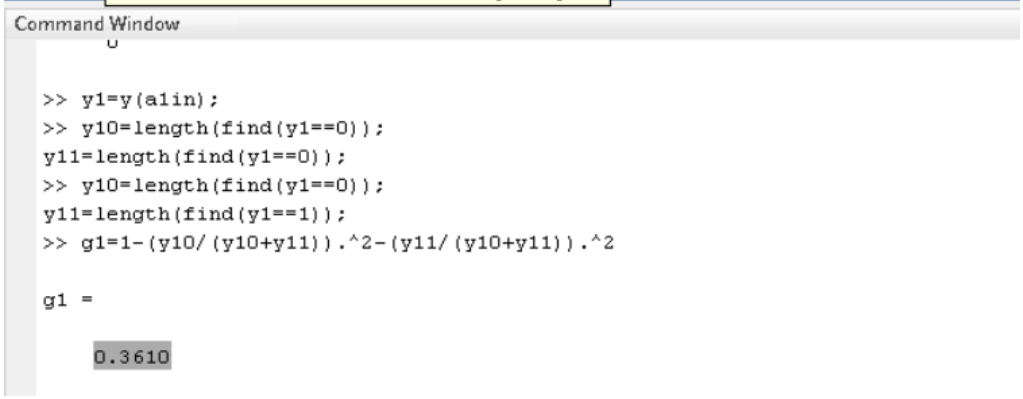
Command Window

```
     1
     2
     2
     1
     2


a2 =

   22.6727
    5.0950

>> a1in=find(a1==1);
>> y=data(:,21)
```


Editor - Untitled4*
Variables - y1
55x1 double
data
a1
a2
a1in
y1
1 2 3 4 5 6 7
1 0
2 0
3 1
4 1
5 1
6 0
7 0
8 0
Command Window
1
0
0
0
0
1
0
0
0
0
>> y1=y(a1in);


Current Folder
Name
model4.m
arc.csv
arc.csv (Microsoft Office Excel Comma Separated Values Fil
Workspace
Name
Value
ans 21
arc 234x24 double
data 234x21 double
x [1;2;3;4]
y 234x1 double
y1 55x1 double
y10 42
y11 13
Editor - Untitled4*
model4.m
Untitled4*
1 function [output_args] = ginical(x,y)
2 [a1,a2]=kmeans(x,2);
3
4 a1in=find(a1==1);
5 y1=y(a1in);
6 y10=length(find(y1==0));
7 y11=length(find(y1==1));
8
9
10 end
11
12
Command Window
0
0
1
0
0
0
0
>> y1=y(a1in);
>> y10=length(find(y1==0));
y11=length(find(y1==0));
>> y10=length(find(y1==0));
y11=length(find(y1==1));
>>



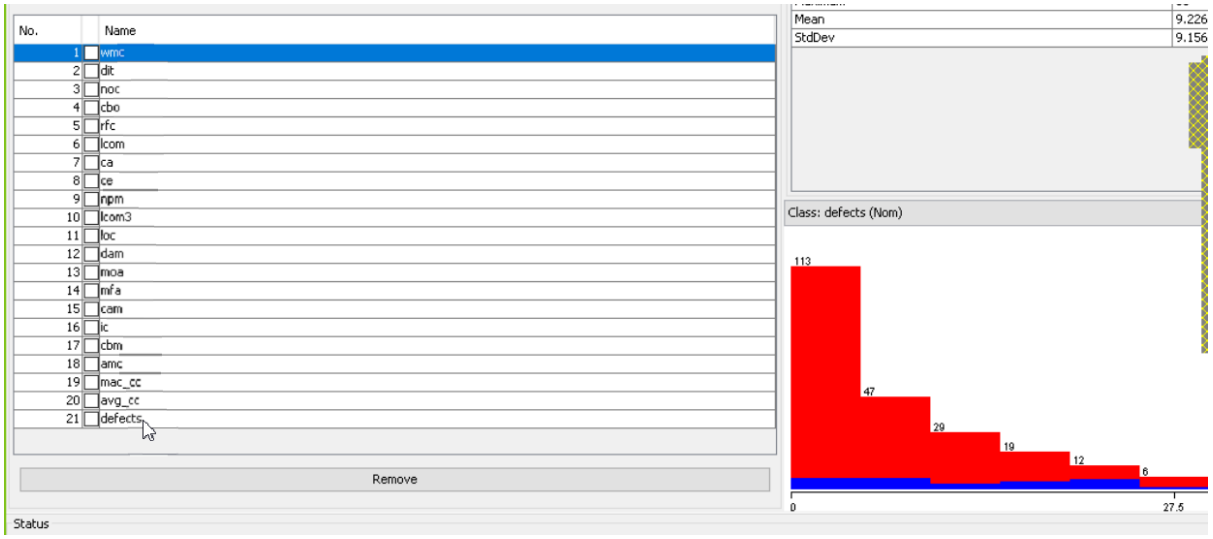


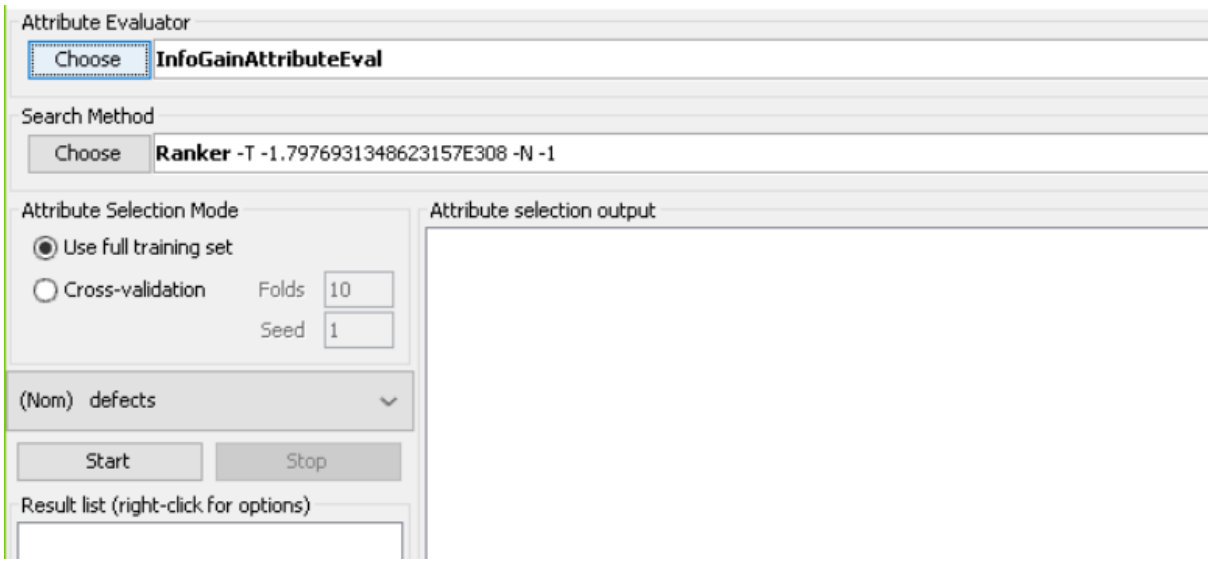

```
giniind=(length(find(a1==1))/length(y))*g1+(length(find(a1==2))/length(y))*g2
```

```
for i=1:20
    [ giniind(i) ] = ginical(data(:,i),data(:,21));
end
```

```
    0.1681    0.1816    0.1819    0.1844    0.1948    0.1975    0.1977    0.1983    0.

  Columns 13 through 20

    0.2027    0.2033    0.2034    0.2035    0.2036    0.2039    0.2040    0.2040


b =

     8     4     5    11     1    13    18     9    15    14    19    20    16     7
```

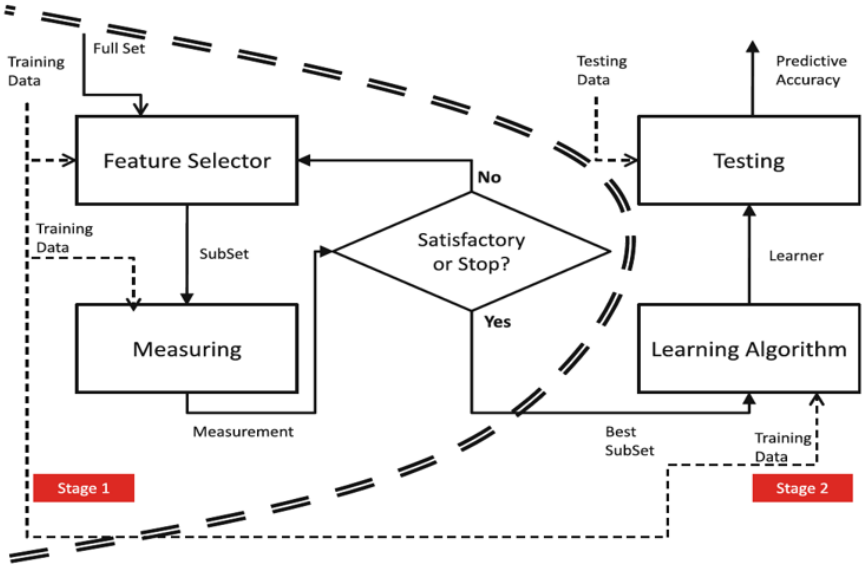
```
@attribute mac_cc numeric
@attribute avg_cc numeric
@attribute defects {1,0}

@data
7,1,0,6,19,7,0,6,6,0.750000000000000,117,1,2,0,0.366666667000000,0,0,15.1428571400000,3,1.142900
13,1,0,1,17,64,0,1,12,0.916666667000000,117,1,0,0,0.461538462000000,0,0,7.46153846200000,3,1.384
7,4,4,48,12,19,47,1,6,0.944444444000000,44,0.333333333000000,0,0.866666667000000,0.458333333000
```





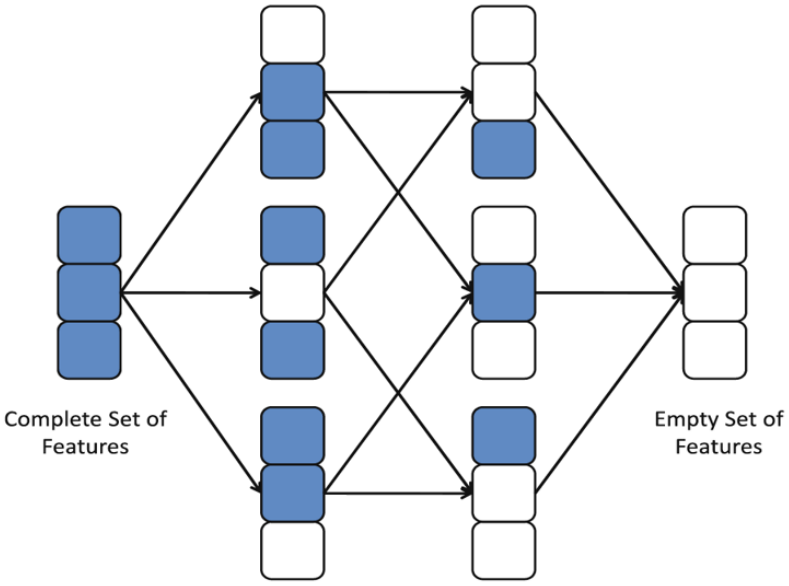

```
Attribute Evaluator (supervised, Class (nominal): 21 defects):
        Information Gain Ranking Filter

Ranked attributes:
 0.0842    8 ce
 0.0693    4 cbo
 0.062     5 rfc
 0.0502   11 loc
 0.0457   15 cam
 0.0448    9 npm
 0.0409    1 wmc
 0         6 lcom
 0         3 noc
 0         2 dit
 0         7 ca
 0        20 avg_cc
 0        19 mac_cc
 0        12 dam
 0        18 amc
 0        17 cbm
 0        16 ic
 0        14 mfa
 0        13 moa
 0        10 lcom3

Selected attributes: 8,4,5,11,15,9,1,6,3,2,7,20,19,12,18,17,16,14,13,10 : 20
```

Filters:



Satisfactory stop – after how many features you will stop – $\log_2 n$ features

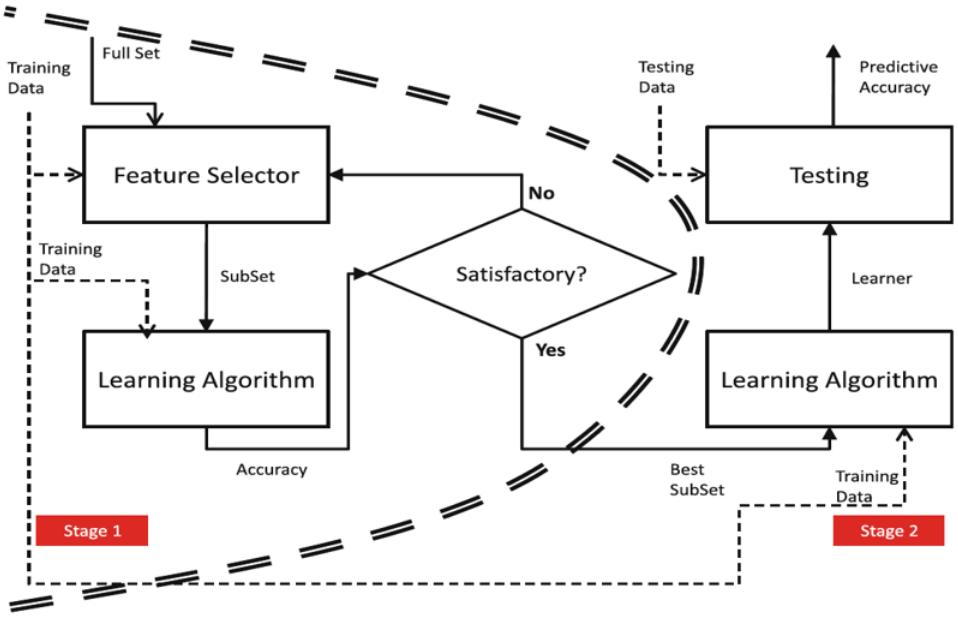
- Feature-subset selection methods are used to find suitable subset of features which collectively have good predictive capability.
- FS can be considered as a search problem, where each state of the search space corresponds to a concrete subset of features selected.
- The selection can be represented as a binary array, with each element corresponding to the value 1, if the feature is currently selected by the algorithm and 0, if it does not occur.



- Search Directions:
  - Sequential Forward Generation (SFG): It starts with an empty set of features S. As the search starts, features are added into S according to some criterion that distinguish the best feature from the others. S grows until it reaches a full set of original features. The stopping criteria can be a threshold for the number of relevant features m or simply the generation of all possible subsets in brute force mode.

  - Sequential Backward Generation (SBG): It starts with a full set of features and, iteratively, they are removed one at a time. Here, the criterion must point out the worst or least important feature. By the end, the subset is only composed of a unique feature, which is considered to be the most informative of the whole set. As in the previous case, different stopping criteria can be used.

Wrappers



Evaluation

- Accuracy
- Complexity
- Number of Features Selected
- Speed of the FS method
- Generality of the features selected

Feature ranking – low complexity as compared to feature subset selection

```
>> inp=data(:,1:4);
>> outp=data(:,21);
>> x=combntns(1:4,1)
Warning: The COMBNTNS function will be removed in a future release.
> In combntns at 22

x =

     1
     2
     3
     4

>>
```

```
>> 2^20-1

ans =

     1048575
```

All the features – performance issue

```
1 -    inp=data(:,1:5);
2 -    outpu=data(:,21);
3
```

Command Window

```
     3
     4

>> x=combntns(1:4,2)
Warning: The COMBNTNS function will be removed in a future release.
> In combntns at 22

x =

     1     2
     1     3
     1     4
     2     3
     2     4
     3     4

>> x=combntns(1:4,2)
```

```
x =

     1     2     3
     1     2     4
     1     3     4
     2     3     4

>> x=combntns(1:4,4)
Warning: The COMBNTNS function will be removed in a fut
> In combntns at 22

x =

     1     2     3     4

>>
```

model4.m | ginical.m | ginv.m | fsss.m

```
function [ ypre ] = model4(trdata,tsdata )
CMdl = fitctree(trdata(:,1:end-1),trdata(:,end));
Y_fit = predict(CMdl,tsdata(:,1:end-1));
ypre=Y_fit;

Y_fit = predict(CMdl,trdata(:,1:end-1));
tpre=Y_fit;
end
```

Return predicted value of testing data

```
>> 234*0.7

ans =

  163.8000
```

```
trdata(:,end+1)=trdef;
tsdata(:,end+1)=tsdef;
for i=1:5
x=combntns(1:5,i)
```

Command Window

```
     0
     0
     0
     0
     0
     1
     0
     1
     0
     0
     0
     1
     0
     0
     0
```

```
x =

     1     2
     1     3
     1     4
     1     5
     2     3
     2     4
     2     5
     3     4
     3     5
     4     5
```

I = 2

```
i=0;
for i=1:5
x=combntns(1:5,i)
for j=1:size(x,1)
    trdatanew=trdata(:,x(j,:));
    trdatanew(:,end+1)=trdata(:,end);

    tsdatanew=tsdata(:,x(j,:));
    tsdatanew(:,end+1)=tsdata(:,end);
```

```
    tsdatanew=tsdata(:,x(j,:));
    tsdatanew(:,end+1)=tsdata(:,end);

    [ ypre ] = model4(trdatanew,tsdatanew );

cmat=confusionmat(tsdata(:,end),ypre);
acc=(cmat(1,1)+cmat(2,2))/(cmat(1,1)+cmat(1,2)+cmat(2,1)+cmat(2,2));
end
```

Command Window

```
     1     5
     2     3
     2     4
     2     5
     3     4
     3     5
     4     5

>> cmat=confusionmat(tsdata(:,end),ypre);
>> cmat

cmat =

    57     5
     6     2
```

1x31 double

	18	19	20	21	22	23	24	25	26	27	28	29
1	0.8857	0.8857	0.8857	0.8429	0.8857	0.8571	0.8429	0.8429	0.8857	0.8857	0.8429	0.842
2												
3												

2^5 – 1 values

```
cmat=confusionmat(tsdata(:,end),ypre);
acc=(cmat(1,1)+cmat(2,2))/(cmat(1,1)+cmat(1,2)+cmat(2,1)+cmat(2,2));
l=l+1;
accv(l)=acc;
subv(l)=x(j,:);
end
end
```

Command Window

```
>> cmat=confusionmat(tsdata(:,end),ypre);
>> cmat

cmat =

    57     5
     6     2

>> acc=(cmat(1,1)+cmat(2,2))/(cmat(1,1)+cmat(1,2)+cmat(2,1)+cmat(2,2));
>> acc

acc =

    0.8429
```

```
>> [a,b]=max(accv)

a =

    0.8857


b =

     8
```

Model developed by 2nd feature performance is 88% - we are training based on 2nd feature

In a specific domain we can compare these two methods – feature ranking and subset selection

Principal Component Analysis (PCA)

- Attribute reduction using Principal Component analysis (PCA) is achieved by transforming high dimension data space into lower dimension data space.
- takes a data matrix of n objects by p variables, which may be correlated, and summarizes it by uncorrelated axes (principal components or principal axes) that are linear combinations of the original p variables

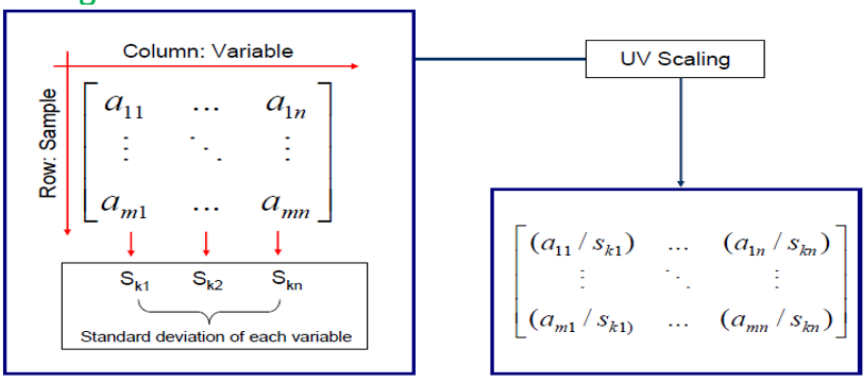
(A, B, C) → (F1, F2) -- F1=A+B, F2=B+C

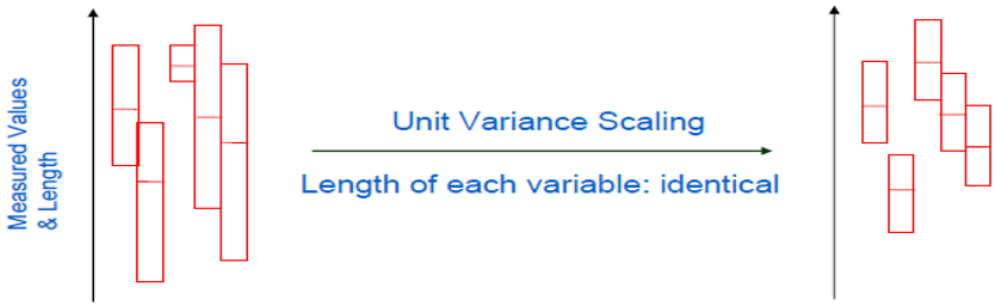
Procedures of PCA

- 1st Step: Pre-treatment of Data Matrix→Scaling
- 2nd Step: Calculation of covariance matrix
- 3rd Step: Calculation of eigenvalues and eigenvectors of covariance matrix
- 4th Step: Calculation of scores

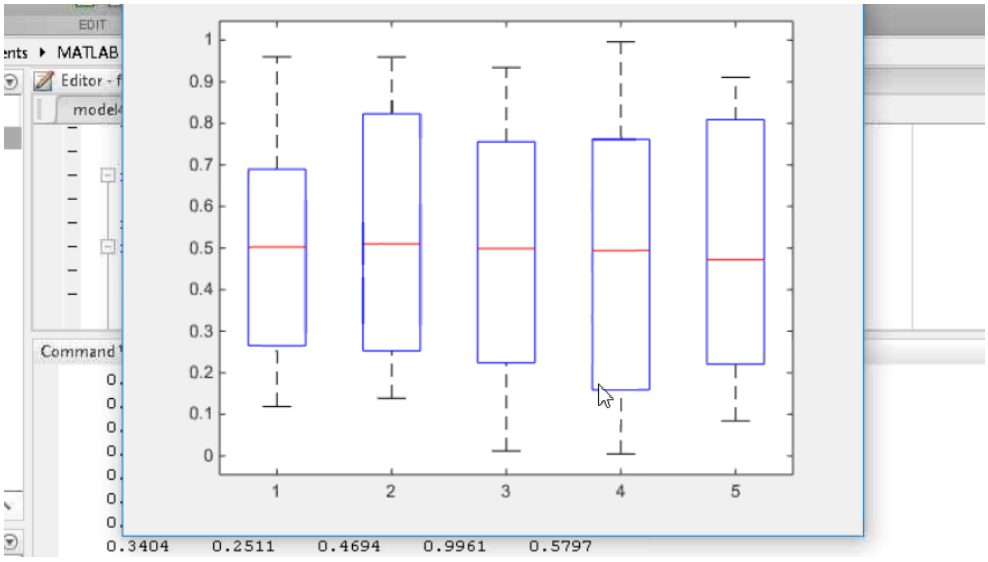
1st Step: Pre-treatment of Data Matrix→Scaling

- Pre-treatment of Data Matrix-Scaling
  - Unless the data are normalized, a variable with a large variance will dominate
  - Most common scaling technique – Unit variance (UV) scaling





Note: However, the mean values still remain different
Therefore **mean-centering** as a second part of pre-data processing
Step1) Average value of each variable is calculated
Step2) Subtracted from the data

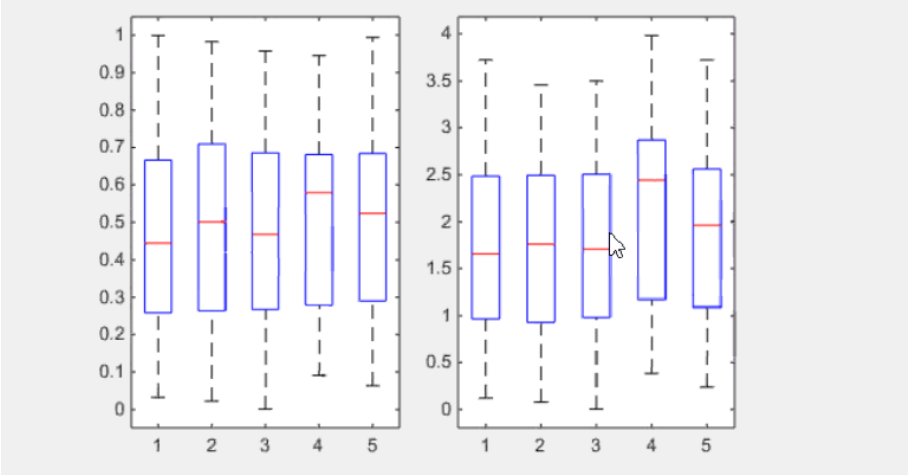


Using sd – normalize the data

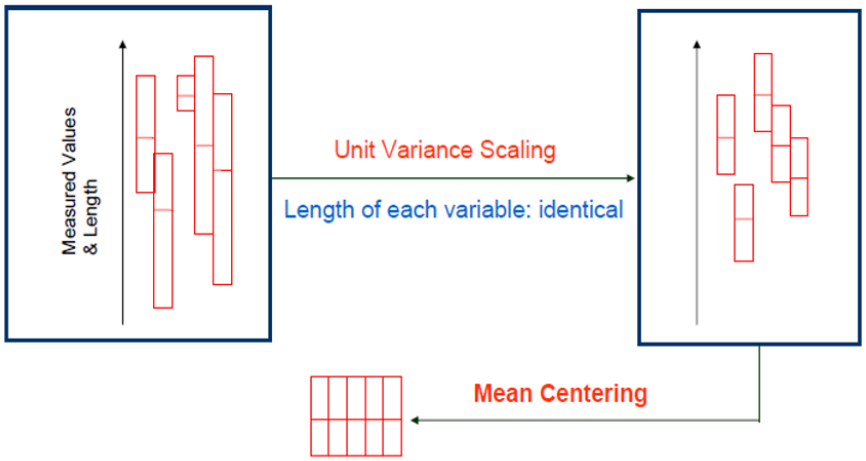
```
boxplot(x);
for i=1:5
    xn(1:40,i)=x(:,i)/std(x(:,i));
end
```

Command Window

```
    0.0154    0.2305    0.2316    0.0287    0.9841
    0.0430    0.8443    0.4889    0.4899    0.1672
    0.1690    0.1948    0.6241    0.1679    0.1062
    0.6491    0.2259    0.6791    0.9787    0.3724
    0.7317    0.1707    0.3955    0.7127    0.1981
    0.6477    0.2277    0.3674    0.5005    0.4897
    0.4509    0.4357    0.9880    0.4711    0.3395
    0.5470    0.3111    0.0377    0.0596    0.9516
    0.2963    0.9234    0.8852    0.6820    0.9203
    0.7447    0.4302    0.9133    0.0424    0.0527
    0.1890    0.1848    0.7962    0.0714    0.7379

>> for i=1:5
   xn(1:40,i)=x(:,i)/std(x(:,i));

end
```

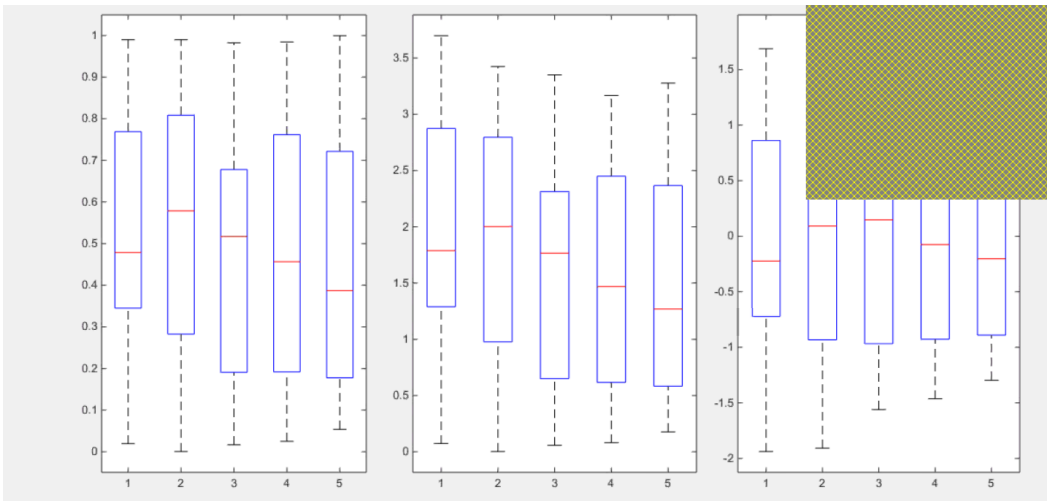


1) Pre-treatment of Data Matrix- Scaling (Continued)




```
 9 -     boxplot(xn);
10
11 -    for i=1:5
12 -        xnn(1:40,i)=xn(:,i)-mean(xn(:,i));
13 -    end
14
15
16 -    subplot(1,3,3);
17 -    boxplot(xn)
```

```
Command Window
    0.5470    0.3111    0.0377    0.0596    0.9516
    0.2963    0.9234    0.8852    0.6820    0.9203
    0.7447    0.4302    0.9133    0.0424    0.0527
    0.1890    0.1848    0.7962    0.0714    0.7379

>> for i=1:5
   xn(1:40,i)=x(:,i)/std(x(:,i));

end
>> pcaana
>> mean(xn)

ans =

    1.7937    1.7064    1.7720    2.1998    1.8821
```



```
>> mean(xnn)

ans =

   1.0e-15 *

    0.5274   -0.5274    0.3886    0.0389   -0.1721
```

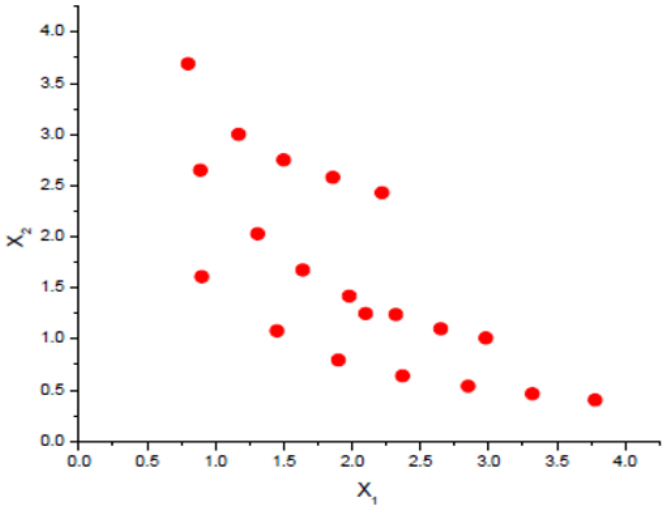
Mean value near to 0

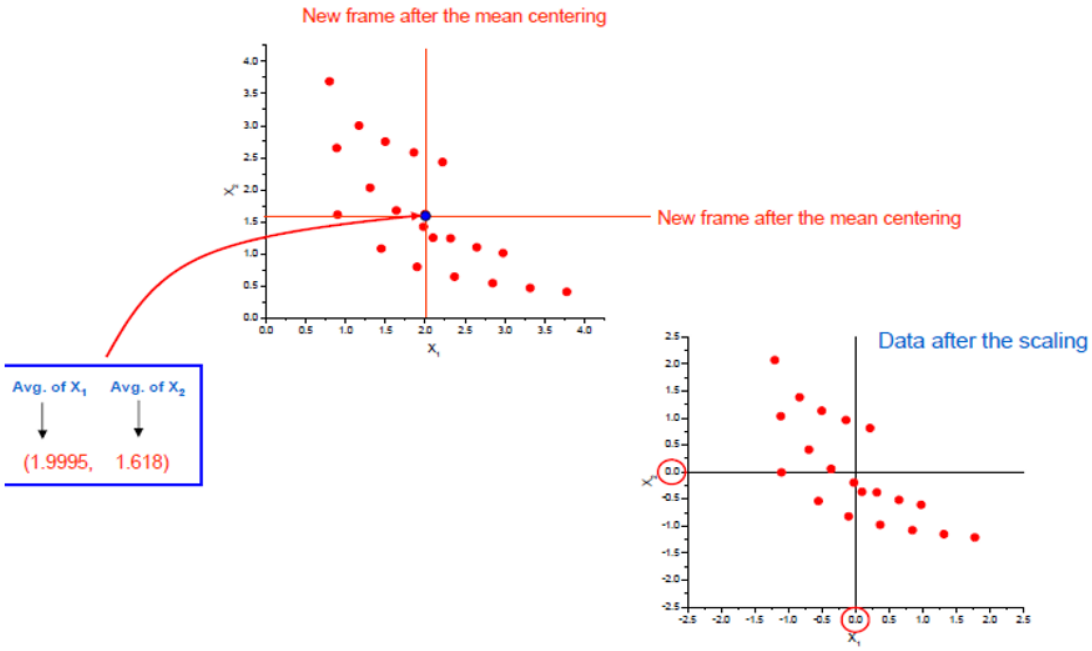
Example of Data Matrix: X

Sample NO.	Element	Martynov-Batsanov's Electronegativity (X_1)	Zunger's pseudopotenial core radii sum (X_2)	Sample NO.	Element	Martynov-Batsanov's Electronegativity (X_1)	Zunger's pseudopotential core radii sum (X_2)
1	H	2.1	1.25	11	Al	1.64	1.675
2	Li	0.9	1.61	12	Si	1.98	1.42
3	Be	1.45	1.08	13	P	2.32	1.24
4	B	1.9	0.795	14	S	2.65	1.1
5	C	2.37	0.64	15	Cl	2.98	1.01
6	N	2.85	0.54	16	K	0.8	3.69
7	O	3.32	0.465	17	Ca	1.17	3
8	F	3.78	0.405	18	Sc	1.5	2.75
9	Na	0.89	2.65	19	Ti	1.86	2.58
10	Mg	1.31	2.03	20	V	2.22	2.43

Example of Data Matrix: X (continued)

The scatter plot of X





- 2nd Step: Calculation of covariance matrix
  - Calculation of Covariance matrix(S) of Data Matrix(X)

- Variance (1 dimensional concept): Measure of the spread of data in a given data set
- Covariance (Multi-dimensional concept): Measure of the spread of data between dimensions (variables)

$$\operatorname{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{(n-1)}$$

```
19 -    covmn=cov(xnn);
20
21
```

Command Window

```
ans =

   1.0e-15 *

    0.5274   -0.5274    0.3886    0.0389   -0.1721

>> covmn=cov(xnn)

covmn =

    1.0000    0.2631    0.0685   -0.0175   -0.0068
    0.2631    1.0000   -0.2721    0.2850   -0.1215
    0.0685   -0.2721    1.0000   -0.0777    0.0904
   -0.0175    0.2850   -0.0777    1.0000   -0.0624
   -0.0068   -0.1215    0.0904   -0.0624    1.0000
```

Calculation of Covariance matrix(S) of Data Matrix(X):

$$S = \begin{bmatrix} \text{cov}(x,x) & \text{cov}(x,y) & \text{cov}(x,z) \\ \text{cov}(y,x) & \text{cov}(y,y) & \text{cov}(y,z) \\ \text{cov}(z,x) & \text{cov}(z,y) & \text{cov}(z,z) \end{bmatrix}$$

- 3rd Step of PCA: Calculation of eigenvalues and eigenvectors of covariance matrix
  - 1) Calculation of Eigenvalues of Covariance matrix (S)
  - 2) Calculation of the corresponding Eigenvectors of Covariance matrix (S)

$$S = \begin{bmatrix} 0.6881 & -0.5929 \\ -0.5929 & 0.9026 \end{bmatrix} \rightarrow \begin{vmatrix} 0.6881-\lambda & -0.5929 \\ -0.5929 & 0.9026-\lambda \end{vmatrix} = 0$$

$$\downarrow$$

$$(0.6881-\lambda)(0.9026-\lambda)-(0.5929)^2 = 0$$

$$\downarrow$$

$$\lambda^2 - 1.5907\lambda + 0.27 = 0$$

$$\downarrow$$

$$\lambda = \frac{1.5907 \pm \sqrt{(1.5907)^2 - 4\times 0.27}}{2}$$

$$\downarrow$$

$$\lambda_1 = 1.3978,\ \lambda_2 = 0.1928$$

Eigenvalues of covariance matrix S

Eigen value → (C- Lambda) * Identity matrix = 0

```
   -0.4566   -0.1100    0.1791   -0.8213   -0.2698
    0.7057   -0.1062   -0.0862   -0.1760   -0.6726
    0.3970    0.6712   -0.0100   -0.4542    0.4306
   -0.3653    0.6074   -0.5155    0.1586   -0.4546
    0.0501   -0.3964   -0.8334   -0.2509    0.2877


eva =

    0.5206         0         0         0         0
         0    0.9081         0         0         0
         0         0    0.9514         0         0
         0         0         0    1.0955         0
         0         0         0         0    1.5244
```

For eigenvalue $\lambda_1 = 1.3978$

$$\begin{bmatrix} 0.6881-1.3978 & -0.5929 \\ -0.5929 & 0.9026-1.3978 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -0.7097 & -0.5929 \\ -0.5929 & -0.4952 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

Is it possible to solve this problem? (NOT for the case of $x_1 = x_2 = 0$)

No, because of two same values (-0.5929)

For eigenvalue $\lambda_2 = 0.1928$

$$\begin{bmatrix} 0.6881-0.1928 & -0.5929 \\ -0.5929 & 0.9026-0.1928 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.4953 & -0.5929 \\ -0.5929 & 0.7098 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

Is it possible to solve this problem? (NOT for the case of $x_1 = x_2 = 0$)

No, because of two same values (-0.5929)

The command is **[P, Λ]=eig(S)**. Then you will get followings in Matlab.

- Total eigenvalues will be 0.1928+1.3978=1.5906 and this is the same as total variance
- Eigenvalues represent the lengths of the two principal axes of ellipse. Therefore the axes represent the total variance of the data set.
- The first principal axis contains 1.3978/1.5906=87.88% of the total variance.

  Second principal axis represents 0.1928/1.5906=12.12% of the total variance.

- 4th Step of PCA: Calculation of scores

$$PC1 = \alpha 1X1 + \alpha 2X2$$

$$PC2 = \beta 1\, X1 + \beta 2\, X2$$

α 's are the elements of the **first eigenvector**

β 's are the elements of the **second eigenvector**

Or

PCA=Data*Eigenvector

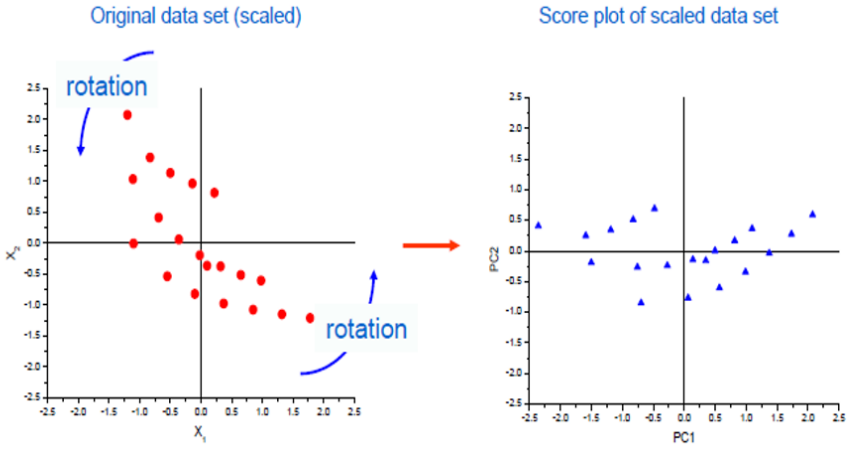
Select principle components with Eigen values above threshold i.e. 0.9

inp
outp
trdata
ypre
accv
xn
pcav

	1	2	3	4	5	6	7
1	-0.3654	-0.4586	1.2147	2.5535	0.6773		
2	-0.2447	0.1347	1.3766	-0.7264	0.4825		
3	1.1716	0.3648	-0.9880	0.3655	0.1921		
4	0.9688	0.9908	-0.0972	-0.0890	-0.5103		
5	1.0367	0.0974	-1.1092	-1.6510	0.4237		

Command Window

```
    -0.4566   -0.1100    0.1791   -0.8213   -0.2698
     0.7057   -0.1062   -0.0862   -0.1760   -0.6726
     0.3970    0.6712   -0.0100   -0.4542    0.4306
    -0.3653    0.6074   -0.5155    0.1586   -0.4546
     0.0501   -0.3964   -0.8334   -0.2509    0.2877


eva =

    0.5206         0         0         0         0
         0    0.9081         0         0         0
         0         0    0.9514         0         0
         0         0         0    1.0955         0
         0         0         0         0    1.5244

>> pcav=xnn*ev;
```

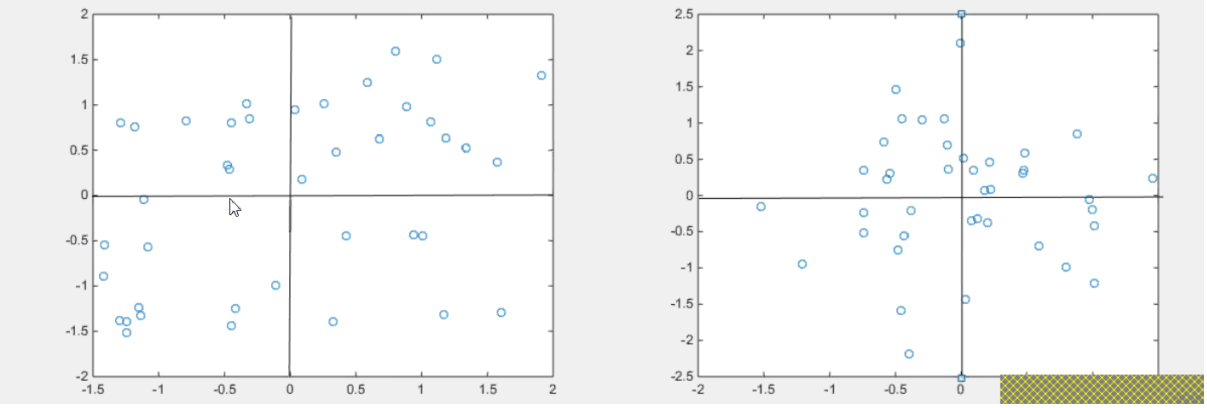


model4.m | ginical.m | ginv.m | fsss.m | pcaana.m*

```
16      % subplot(1,3,3);
17      % boxplot(xnn);
18
19 -    covmn=cov(xnn);
20 -    [ev,eva]=eig(covmn);
21 -    pcav=xnn*ev;
22 -    subplot(1,2,1);
23 -    plot(xnn(:,1),xnn(:,2),'o');
24      subplot(1,2,2);
```

Command Window

```
    -0.4566   -0.1100    0.1791   -0.8213   -0.2698
     0.7057   -0.1062   -0.0862   -0.1760   -0.6726
     0.3970    0.6712   -0.0100   -0.4542    0.4306
    -0.3653    0.6074   -0.5155    0.1586   -0.4546
     0.0501   -0.3964   -0.8334   -0.2509    0.2877


eva =

    0.5206         0         0         0         0
         0    0.9081         0         0         0
         0         0    0.9514         0         0
         0         0         0    1.0955         0
         0         0         0         0    1.5244

>> pcav=xnn*ev;
>> [ev,eva]=eig(covmn)
```



Lec 12 Anomaly Detection

$$S = \begin{bmatrix} 0.6881 & -0.5929 \\ -0.5929 & 0.9026 \end{bmatrix} \rightarrow \begin{vmatrix} 0.6881-\lambda & -0.5929 \\ -0.5929 & 0.9026-\lambda \end{vmatrix} = 0$$

$$(0.6881-\lambda)(0.9026-\lambda)-(0.5929)^2 = 0$$

$$\lambda^2 - 1.5907\lambda + 0.27 = 0$$

$$\lambda = \frac{1.5907 \pm \sqrt{(1.5907)^2 - 4\times 0.27}}{2}$$

$$\lambda_1 = 1.3978,\ \lambda_2 = 0.1928$$

Eigenvalues of covariance matrix S

Anomaly/Outlier Detection

- What are anomalies/outliers?
  - The set of data points that are considerably different than the remainder of the data
- Variants of Anomaly/Outlier Detection Problems
  - Given a database D, find all the data points x ∈ D with anomaly scores greater than some threshold t
  - Given a database D, find all the data points x ∈ D having the top-n largest anomaly scores f(x)
  - Given a database D, containing mostly normal (but unlabeled) data points, and a test point x, compute the anomaly score of x with respect to D
- Applications:
  - Credit card fraud detection, telecommunication fraud detection, network intrusion detection, fault detection

Anomaly Detection

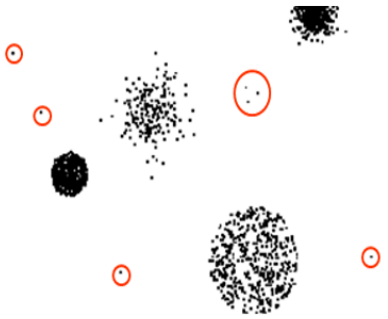
innovate achieve lead

- Challenges
  - How many outliers are there in the data?
  - Method is unsupervised
  - Validation can be quite challenging (just like for clustering)

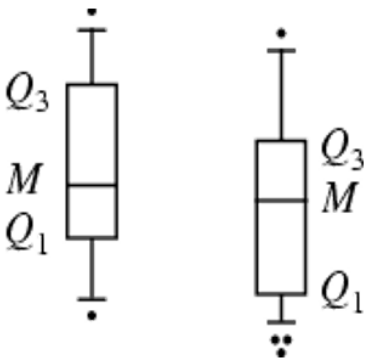
- Working assumption:
  - There are considerably more "normal" observations than "abnormal" observations (outliers/anomalies) in the data

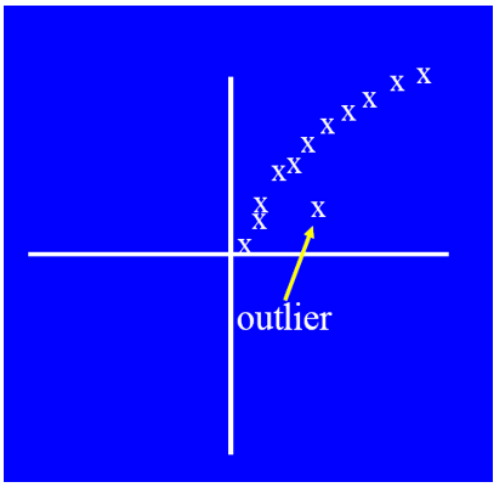
- General Steps
  - Build a profile of the "normal" behavior
    - Profile can be patterns or summary statistics for the overall population
  - Use the "normal" profile to detect anomalies
    - Anomalies are observations whose characteristics differ significantly from the normal profile

Types of anomaly detection schemes
- Graphical & Statistical-based
- Distance-based
- Model-based



Boxplot (1-D), Scatter plot (2-D), Spin plot (3-D)





Box-plot

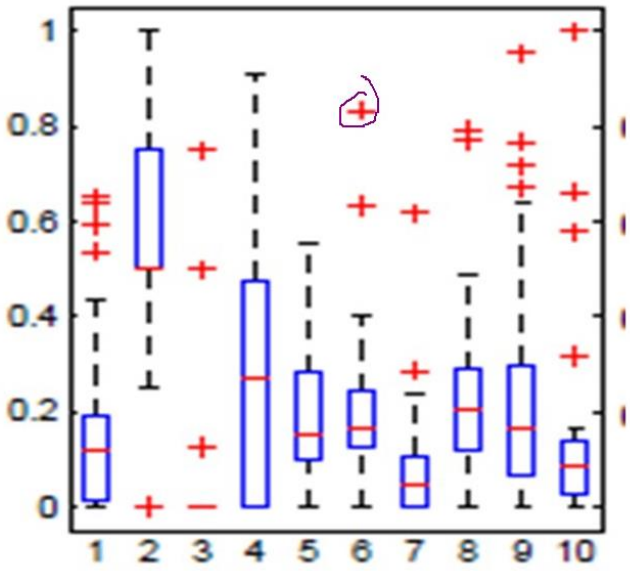
innovate achieve lead

- Effectiveness of outliers is examined by using the following equation

$$e_i = \begin{cases} \text{if } |y_{ji} - \hat{y}_j| > 3 * \sigma & \text{for Effective outliers} \\ \text{if } |y_{ji} - \hat{y}_j| <= 3 * \sigma & \text{for Non Effective outliers} \end{cases}$$

- Or

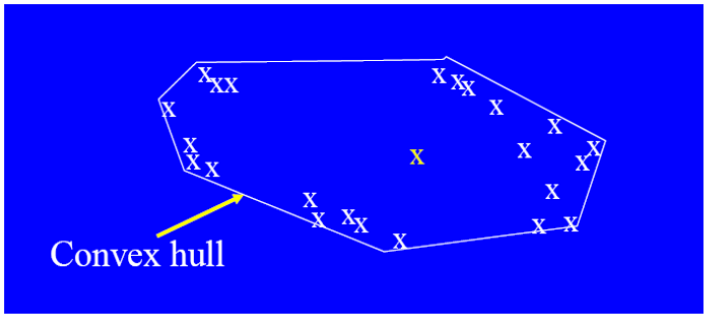
Lower Limit = Q1 – 1.5 IQR.
Upper Limit = Q3 + 1.5 IQR
IQR=Q3-Q1



Convex Hull Method

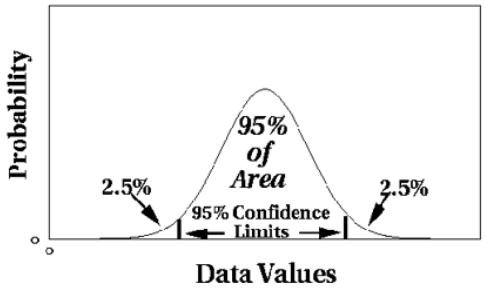
innovate achieve lead

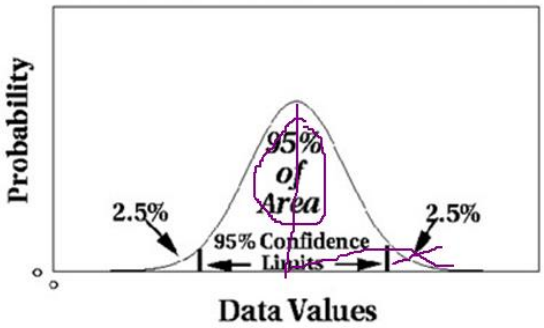
- Extreme points are assumed to be outliers Use convex hull method to detect extreme values



Statistical Approaches

- Assume a parametric model describing the distribution of the data (e.g., normal distribution)
- Apply a statistical test that depends on
  - Data distribution
  - Parameter of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)





Grubbs' Test

- Detect outliers in univariate data
- Assume data comes from normal distribution
- Detects one outlier at a time, remove the outlier, and repeat
  - H0: There is no outlier in data
  - HA: There is at least one outlier
- Grubbs' test statistic: $$G = \frac{\max\left|X - \overline{X}\right|}{s}$$
- Reject H0 if: $$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/N,N-2)}}{N-2+t^2_{(\alpha/N,N-2)}}}$$

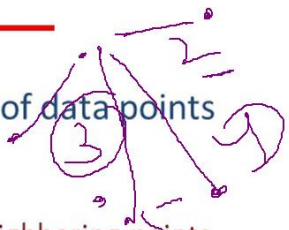
Distance-based Approaches

- Data is represented as a vector of features
- Three major approaches
  - Nearest-neighbor based
  - Density based
  - Clustering based

Nearest-Neighbor Based Approach

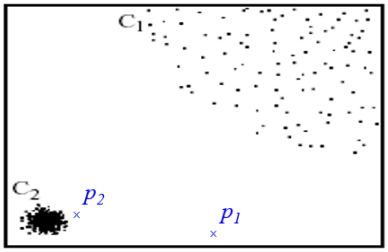
- Approach:
- Compute the distance between every pair of data points
- There are various ways to define outliers:
  - Data points for which there are fewer than p neighboring points within a distance D
  - The top n data points whose distance to the kth nearest neighbor is greatest
  - The top n data points whose average distance to the k nearest neighbors is greatest

- Approach:
- Compute the distance between every pair of data points
- There are various ways to define outliers:
  - Data points for which there are fewer than p neighboring points within a distance D
  - The top n data points whose distance to the kth nearest neighbor is greatest
  - The top n data points whose average distance to the k nearest neighbors is greatest

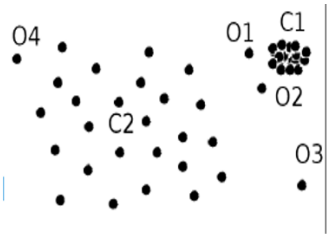


Density-based: LOF approach

- For each point, compute the density of its local neighborhood
- Compute local outlier factor (LOF) of a sample p as the average of the ratios of the density of sample p and the density of its nearest neighbors
- Outliers are points with largest LOF value



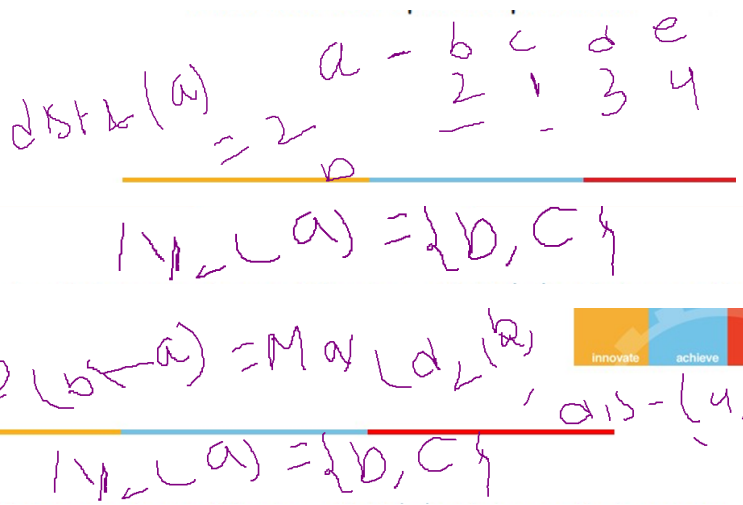
- In Fig., o1 and o2 are local outliers to C1, o3 is a global outlier, but o4 is not an outlier. However, proximity-based clustering cannot find o1 and o2 are outlier (e.g., comparing with O4).



- k-distance of an object o, distk(o): distance between o and its k-th NN
- k-distance neighborhood of o, Nk(o) = {o'| o' in D, dist(o, o') ≤ distk(o)}
- Reachability distance from o' to o:

$$reachdist_k(o \leftarrow o') = \max\{dist_k(o), dist(o, o')\}$$

  - where k is a user-specified parameter



- Local reachability density of o:

$$lrd_k(o) = \frac{\|N_k(o)\|}{\sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)}$$

- LOF (Local outlier factor) of an object o is the average of the ratio of local reachability of o and those of o's k-nearest neighbors

$$LOF_k(o) =: \sum_{o' \in N_k(o)} lrd_k(o') \cdot \sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)$$

LOF(Local Outlier Factor) Example

Consider the following 4 data points:

- a(0, 0), b(0, 1), c(1, 1), d(3, 0)
- Calculate the LOF for each point and show the top 1 outlier, set k = 2 and use Manhattan Distance.

Step 1: calculate all the distances between each two data points

There are 4 data points:

a(0, 0), b(0, 1), c(1, 1), d(3, 0)

(Manhattan Distance here)

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

dist(a, b) = 1

dist(a, c) = 2

dist(a, d) = 3

dist(b, c) = 1

dist(b, d) = 3+1=4

dist(c, d) = 2+1=3

Step 2: calculate all the dist2(o)

- distk(o): distance between o and its k-th NN(k-th nearest neighbor)

dist2(a) = **dist(a, c) = 2** ***(c is the 2nd nearest neighbor)***

dist2(b) = **dist(b, a) = 1** ***(a/c is the 2nd nearest neighbor)***

dist2(c) = **dist(c, a) = 2** ***(a is the 2nd nearest neighbor)***

dist2(d) = **dist(d, a) = 3** ***(a/c is the 2nd nearest neighbor)***

Step 3: calculate all the Nk(o)

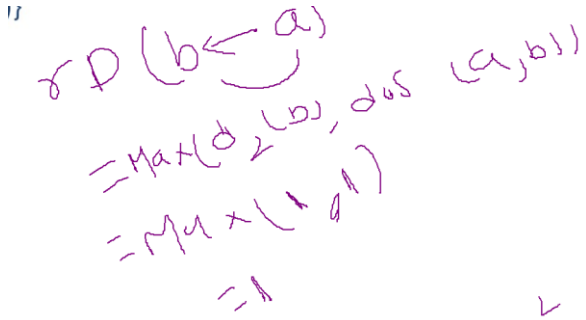
- k-distance neighborhood of o, Nk(o) = {o' | o' in D, dist(o, o') ≤ distk(o)}

N2(a) = {b, c}

N2(b) = {a, c}

N2(c) = {b, a}

N2(d) = {a, c}



Step 4: calculate all the lrdk(o)

- $lrd_k(o)$: Local Reachability Density of o

$$lrd_k(o) = \frac{\|N_k(o)\|}{\sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)}$$

$$reachdist_k(o \leftarrow o') = \max\{dist_k(o), dist(o, o')\}$$

|| $N_k(o)$ || means the number of objects in $N_k(o)$,

For example: || $N_2(a)$ || = || {b, c} || = 2

$$lrd_k(a) = \frac{\| N_2(a) \|}{reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a)}$$

$$reachdist_k(o \leftarrow o') = \max\{dist_k(o), dist(o, o')\}$$

$$reachdist_2(b \leftarrow a) = \max\{dist_2(b), dist(b, a)\}$$

$$= \max\{1, 1\} = 1$$

$$reachdist_2(c \leftarrow a) = \max\{dist_2(c), dist(c, a)\}$$

$$= \max\{2, 2\} = 2$$

Thus, $lrd_2(a)$

$$= \frac{\| N_2(a) \|}{reachdist_2(b \leftarrow a) + reachdist_2(c \leftarrow a)} = 2/(1+2) = 0.667$$

Similarly,

$$lrd_2(b) = \frac{||N_2(b)||}{reachdist_2(a \leftarrow b) + reachdist_2(c \leftarrow b)} = 2/(2+2) = 0.5$$

$$lrd_2(c) = \frac{||N_2(c)||}{reachdist_2(b \leftarrow c) + reachdist_2(a \leftarrow c)} = 2/(1+2) = 0.667$$

$$lrd_2(d) = \frac{||N_2(b)||}{reachdist_2(a \leftarrow d) + reachdist_2(c \leftarrow d)} = 2/(3+3) = 0.33$$

innovate achieve lead

Step 5: calculate all the LOFk(o)

Local outlier factor (LOF)

$$LOF_k(o) =: \sum_{o' \in N_k(o)} lrd_k(o') \cdot \sum_{o' \in N_k(o)} reachdist_k(o' \leftarrow o)$$

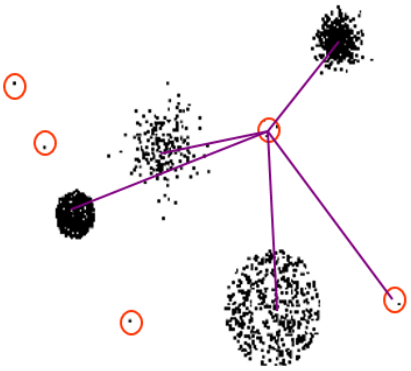
LOF2(a) =
(lrd2(b) + lrd2(c)) * (reachdist2(b←a) + reachdist2(c←a))
= (0.5+0.667) * (1+2) = 3.501
LOF2(b) =
(lrd2(a) + lrd2(c)) * (reachdist2(a←b) + reachdist2(c←b))
= (0.667+0.667) * (2+2) = 5.336

LOF2(c) =
(lrd2(b) + lrd2(a)) * (reachdist2(b←c) + reachdist2(a←c))
= (0.5+0.667) * (1+2) = 3.501
LOF2(d) =
(lrd2(a) + lrd2(c)) * (reachdist2(a←d) + reachdist2(c←d))
= (0.667+0.667) * (3+3) = 8.004

The sorted order is:
LOF2(**d**) = 8.004
LOF2(**b**) = 5.336
LOF2(**a**) = 3.501
LOF2(**c**) = 3.501
Obviously, top 1 outlier is point d.

Clustering-Based

- Basic idea:
  - Cluster the data into groups of different density
  - Choose points in small cluster as candidate outliers
  - Compute the distance between candidate points and non-candidate clusters.
    - If candidate points are far from all other non-candidate points, they are outliers



Box-plot

- Effectiveness of outliers is examined by using the following equation

$$e_i = \begin{cases} \text{if } |y_{ji} - \hat{y}_j| > 3 * \sigma & \text{for Effective outliers} \\ \text{if } |y_{ji} - \hat{y}_j| <= 3 * \sigma & \text{for Non Effective outliers} \end{cases}$$

- Or

Lower Limit = Q1 – 1.5 IQR.
Upper Limit = Q3 + 1.5 IQR
IQR=Q3-Q1