

Data Mining



वीरमाता जिजाबाई टेक्नॉलॉजीकल इन्स्टिट्यूट
Veermata Jijabai Technological Institute
(VJTI Mumbai)

VJTI Mumbai



Scheme

| S. No. | Course Code | Course Title | L-T-P (Hours / Week) | Credits | TA | MST | ESE | ESE hours |
|-----------|----------------|--------------------------------------|----------------------------|---------|----|-----|-----|--------------|
| 1. | R4CO4001 T | Data Mining and Data Warehousing | 3-0-0 | 3 | 20 | 40 | 60 | 3 |
| 2. | R4CO4001 P | Data Mining and Data Warehousing Lab | 0-0-2 | 1 | 60 | 0 | 40 | |

Course Outcomes

1. Perform the pre-processing of data and apply mining and data warehousing techniques
2. Identify and Implement association rules, classification, and clustering algorithms
3. Solve real world problems in business and scientific information using data mining
4. Use data analysis tools for scientific applications

Evaluation

The structure: 3-0-2 (TH-TUT-Lab)

MST: 40 Marks (50% Weightage)

TA: 20 Marks:

Class attendance,
Assignment submission,
Quizzes,
Project,
Tutorial...

ESE: 100 Marks : (3hrs) : (60% Weightage)

Off-line / online ???

Theory

Syllabus:

Lecture Plan: 35 - 40 Lectures

Take home reading....

Laboratory

Lab Plan:

8/10 experiments + Project

Duration : 22 hrs

Lab Evaluation Scheme:

Individual performance,

Understanding,

Timely Submission,

Data Analysis.../ Q&A/

Programming Language-Platform:

Python, R, Java...

Validation- WEAKA, ORANGE,

TABLUS, ORACLE BI..

Laboratory

Data Sets:

Own Data set (more weightage),
Free data from UCI, IEEE, Kaggle...or other open
access repositories...

Smart Hackathon based problems...

Project Based Learning (PBL):

Activity...Problem based...(PBL):
Group of 2-3-4 Students, Socially relevant problem,
UN Sustainable Development Goals....

Reference

Reference Books:

1. Introduction to Data Mining. 2nd Edition. Pearson / Addison Wesley. -- Pang-Ning Tan, Michael Steinbach, Vipin Kumar
2. Data Mining Concepts and Techniques. 3rd Edition, Morgan Kaufmann. -- J. Han, M. Kamber and J. Pei
3. Data Mining and Machine Learning. Cambridge University Press. – Mohammed Zaki

On-Line resources:

1. Data Mining: <https://nptel.ac.in/courses/106/105/106105174/>
2. Introduction to Data Mining. University of Mannheim. --Prof. Bizer: Data Mining <https://www.uni-mannheim.de/dws/teaching/course-details/courses-for-master-candidates/ie-500-data-mining/>

DM and ML

People use Data Mining and Machine Learning interchangeably, unaware that the words mean two different things.

DM and ML have some shared characteristics.

DM is also called knowledge discovery in databases (KDD) (1990).

Machine learning first time presented in a checker-playing program (1950)



DM and ML

Data Mining:

- The process of extracting useful information from a vast amount of data.
- It discovers new, accurate, and useful patterns in the data
- It looks for meaningful and relevant information for the organization or individual.
- It is a tool used by humans.
- Data mining relies on vast stores of data (e.g., Big Data) which is used to make forecasts for businesses and other organizations.

Machine Learning:

- The process of discovering algorithms for improved experience derived from data.
- It refers to design, study, and development of algorithms that permit machines to learn without human intervention.
- It is a tool to make machines smarter, eliminating the human element (but not eliminating humans themselves; that would be wrong).
- Machine learning works with algorithms, not raw data.

DM and ML

Data Mining:

DM mining relies on human intervention and is ultimately created for use by people.

DM can't learn or adapt.

DM follows pre-set rules and is static

DM is only as smart as the users who enter the parameters

DM incorporates two elements: the database and machine learning.

Machine Learning:

ML can teach itself and not depend on human influence or actions

ML doesn't necessarily need data mining.

ML is based on learning and adaption

ML adjusts the algorithms as the right circumstances manifest themselves.

ML means those computers are getting smarter.

What is Data Mining

Data:

Any observation that have been collected

Mining:

Is the extraction of valuable minerals or other geological materials from the Earth, usually from an ore body, lode, vein, seam, reef or placer deposit.

Definitions of Data Mining

- Definitions

“Exploration & analysis, of large quantities of data in order to discover meaningful patterns”.

“A process used to extract usable patterns / data from a larger set of any raw data”.

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

- Data Mining methods:

1. **detect** interesting patterns in large quantities of data
2. **support** human decision making by providing such patterns
3. **predict** the outcome of a future observation based on the patterns

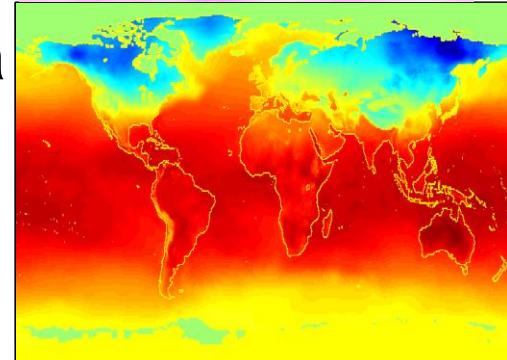
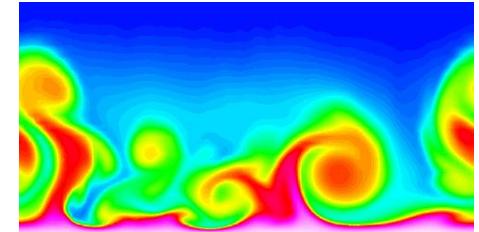
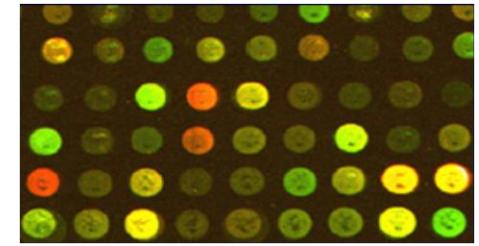
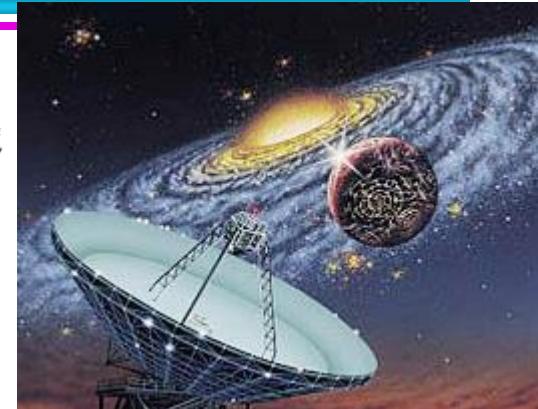
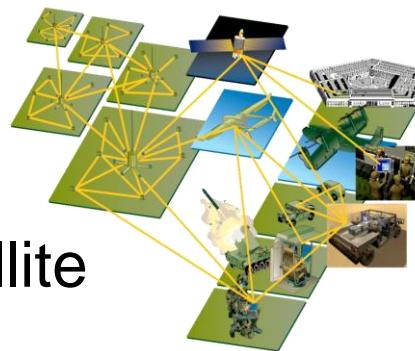
Why Mine Data? Commercial Viewpoint

- Large quantities of data are collected about all aspects of our lives
 - Web data, e-commerce
 - purchases at department/grocery stores
 - Bank/Credit Card transactions
- This data contains interesting patterns
- Competitive Pressure is Strong
 - Provide better, customized services for an edge (e.g. in Customer Relationship Management)



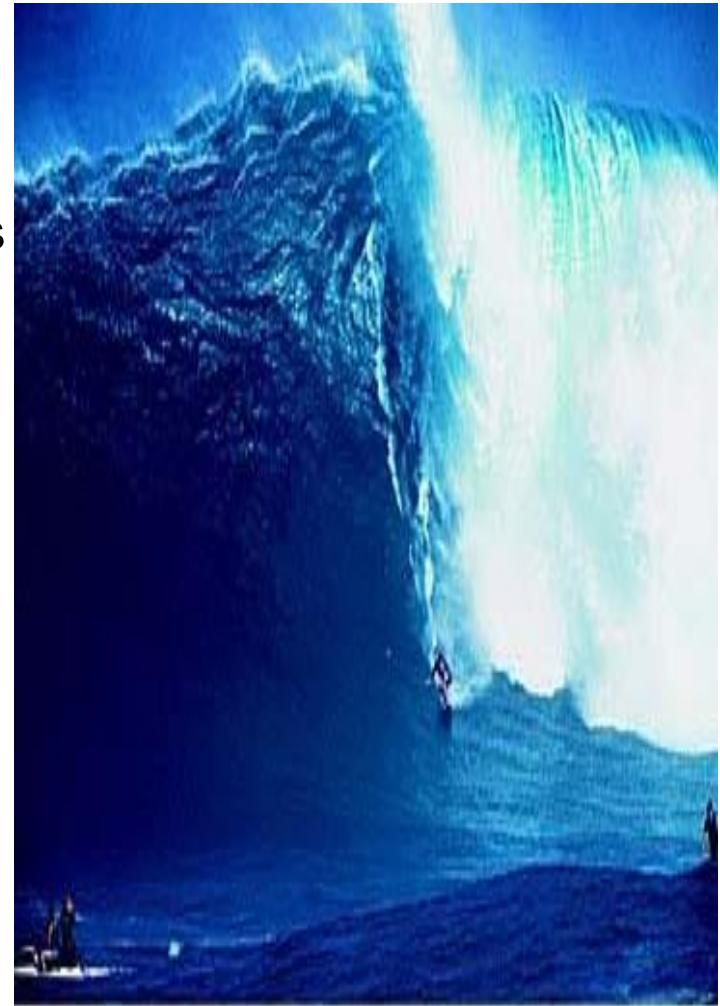
Why Mine Data? Scientific Viewpoint

- Data collected and stored at enormous speeds (GB/hour)
 - remote sensors on a satellite
 - telescopes scanning the skies
 - microarrays generating gene expression data, gene sequencing data
 - scientific simulations generating terabytes of data
- Traditional techniques infeasible for raw data

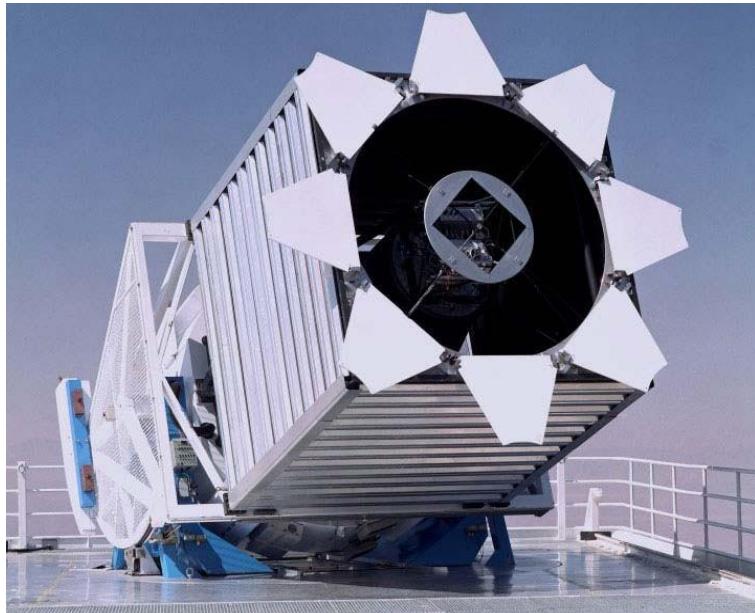


Why Mine Data? User Viewpoint

- Data mining may help us to
 1. discover these patterns and
 2. use them for decision making across all areas of society, including
 - Business and industry
 - Science and engineering
 - Medicine and biotech
 - Government
 - Individuals
 3. in Hypothesis Formation
 4. classifying and segmenting data



"We are Drowning in Data..."



Sloan Digital Sky Survey
 $\approx 200 \text{ GB/day}$
 $\approx 73 \text{ TB/year}$

Predict

- Type of sky object: Star or galaxy?

“We are Drowning in Data...”



US Library of Congress
≈ 235 TB archived
≈ 40 Wikipedias

Discover

- Topic distributions
- Historic trends*
- Citation networks

* Lansdall-Welfare, et al.: Content analysis of 150 years of British periodicals. PNSA, 2017.

“We are Drowning in Data...”



Facebook

- 4 Petabyte of new data generated every day
- over 300 Petabyte in
- Facebook's data warehouse

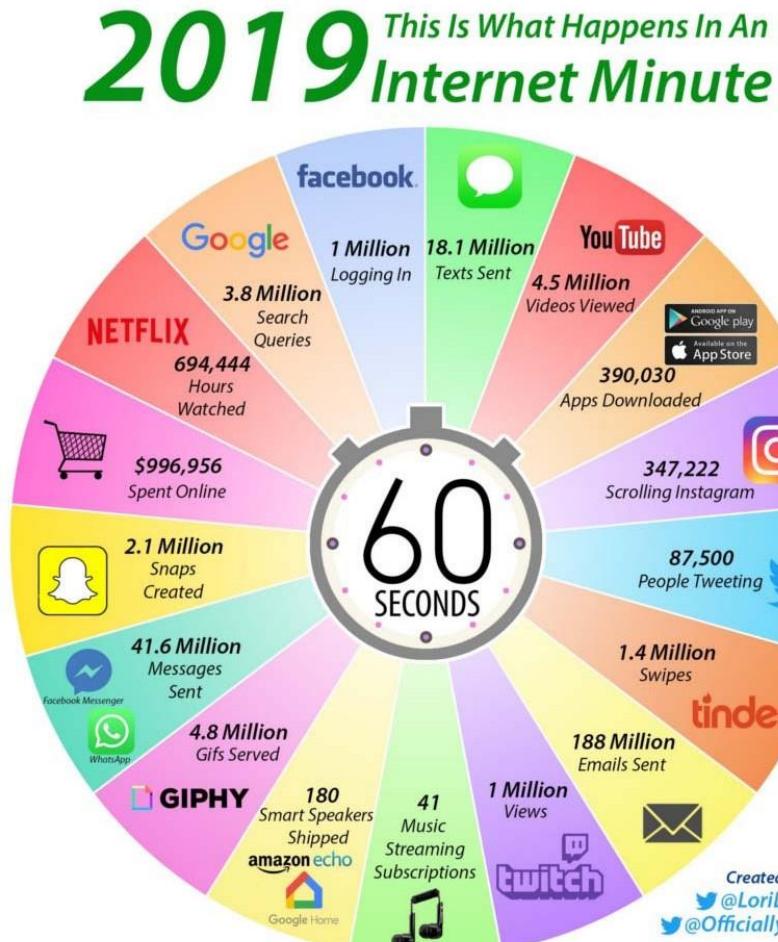
Predict

- Interests and behavior of over one billion people

<https://www.brandwatch.com/blog/facebook-statistics/>

<http://www.technologyreview.com/featuredstory/428150/what-facebook-knows/>

"We are Drowning in Data..."



Predict

- Interests and behavior of mankind

“We are Drowning in Data...”

Law enforcement agencies
collect unknown amounts of
data from various sources

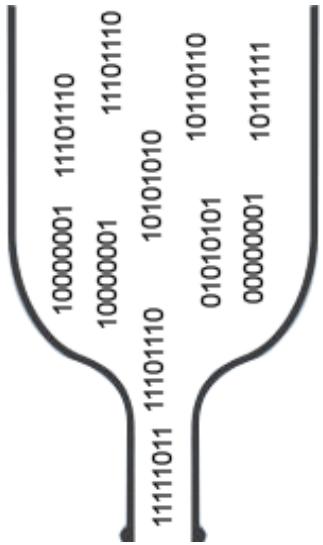
- Cell phone calls
- Location data
- Web browsing behavior
- Credit card transactions
- Online profiles (Facebook)
- ...

Predict

- Terrorist or not?
- Trustworthiness



“...but starving for knowledge!”



← Amount of data that is collected

← Amount of data that can be looked at by humans

We are interested in **the patterns, not the data**

itself! Data Mining methods help us to

- discover interesting patterns in large quantities of data
- take decisions based on the patterns

Mining Large Data Sets - Motivation

- There is often information “hidden” in the data that is not readily evident
- Human analysts may take weeks to discover useful information
- Much of the data is never analyzed at all

What is (not) Data Mining?

□ What is not Data Mining?

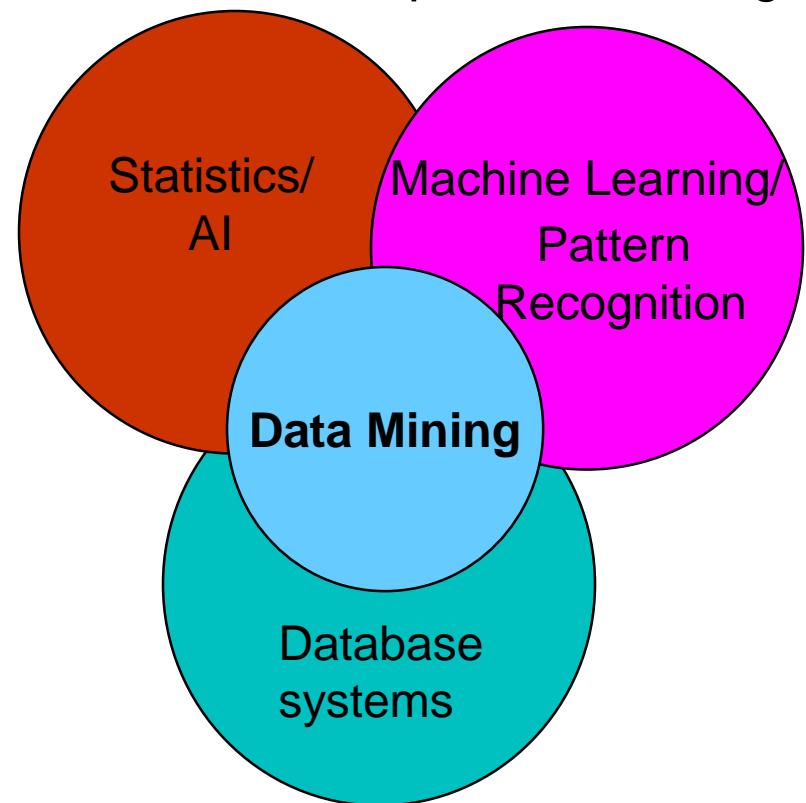
- Look up phone number in phone directory
- Query a Web search engine for information about “Amazon”

□ What is Data Mining?

- Certain names are more prevalent in certain US locations (O'Brien, O'Rurke, O'Reilly... in Boston area)
- Group together similar documents returned by search engine according to their context (e.g. Amazon rainforest, Amazon.com,)

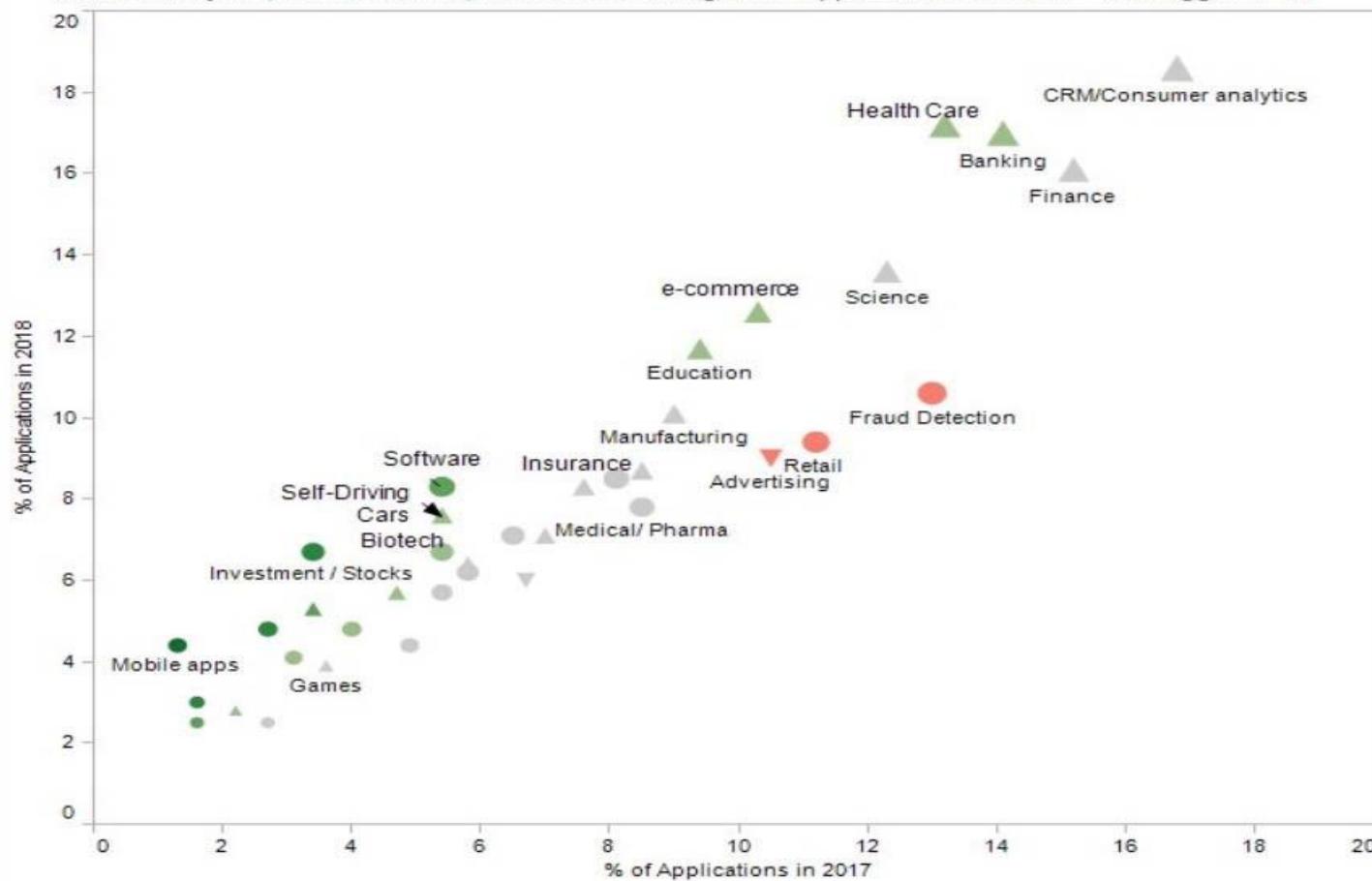
Origins of Data Mining

- Data Mining combines ideas from statistics, machine learning, artificial intelligence, and database systems
- Tries to overcome short- comings of traditional techniques concerning
 - large amount of data
 - high dimensionality of data
 - heterogeneous and complex nature
 - of data
 - explorative analysis beyond hypothesize-and-test paradigm



Survey on Data Mining Application Fields

Where Analytics, Data Science, Machine Learning were applied in 2018-2017 - KDnuggets Poll



- Source: KDnuggets online poll, 435 and 446 participants
- <https://www.kdnuggets.com/2019/03/poll-analytics-data-science-ml-applied-2018.html>

Data Mining Tasks

□ Predictive Task

- Goal: Predict unknown values of a variable
 - given observations (e.g., from the past)
- Example: *Will a person click a online advertisement?*
 - given his/her browsing history

□ Descriptive task

- Goal: Find patterns in the data.
- Example: Which products are often bought together?
- Machine Learning Terminology
 - descriptive = unsupervised
 - predictive = supervised

Data Mining Tasks...

- Classification [Predictive]
- Clustering [Descriptive]
- Association Rule Discovery [Descriptive]
- Sequential Pattern Discovery [Descriptive]
- Regression [Predictive]
- Deviation Detection [Predictive]

Classification: Definition

Goal: Previously unseen records should be assigned a class from a given set of classes as accurately as possible.

- Approach:
- Given a collection of records (*training set*)
 - each record contains a set of *attributes*
 - one attribute is the *class attribute (label)* that should be predicted



Find a *model* for predicting the class attribute as a function of the values of other attributes

- 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: Example

- Training set:



"tree"



"tree"



"tree"



"not a tree"



"not a tree"



"not a tree"

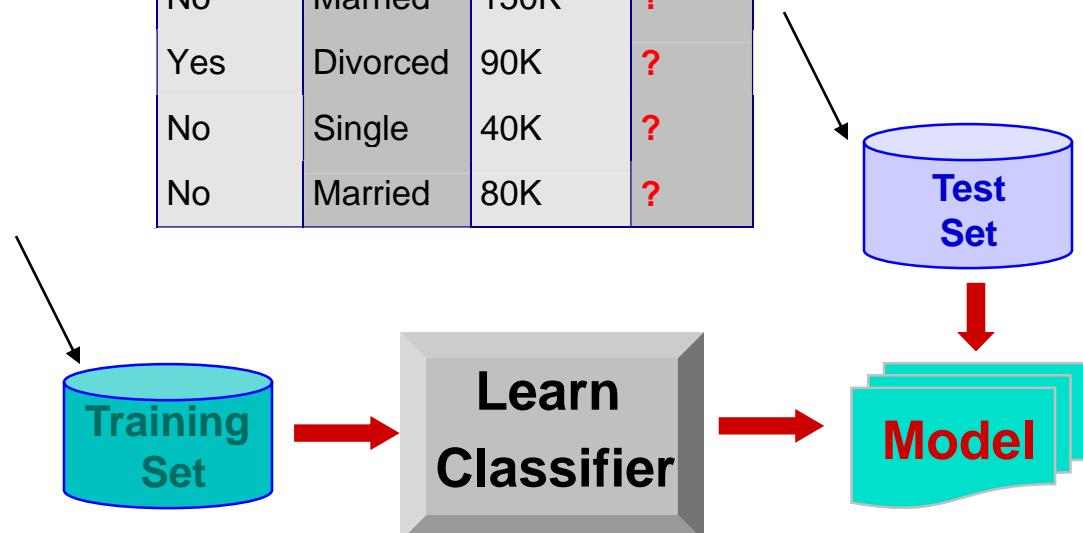
- Learned model: "Trees are big, green plants without wheels."

Classification Example

| 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 |

categorical
 categorical
 continuous
 class

| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Single | 75K | ? |
| Yes | Married | 50K | ? |
| No | Married | 150K | ? |
| Yes | Divorced | 90K | ? |
| No | Single | 40K | ? |
| No | Married | 80K | ? |



Classification: Application 1



- Application area: Direct Marketing
- Goal: Reduce cost of mailing by *targeting* a set of consumers likely to buy a new cell-phone product.
 - Approach:
 - ◆ Use the data for a similar product introduced before.
 - ◆ We know which customers decided to buy and which decided otherwise. This *{buy, don't buy}* decision forms the *class attribute*.
 - ◆ Collect various demographic, lifestyle, and company-interaction related information about all such customers.
 - Type of business, where they stay, how much they earn, etc
 - age, profession, location, income, marriage status, visits, logins, etc.
 - ◆ Use this information as input attributes to learn a classifier model. Apply model to decide which consumers to target

Classification: Application 2

- Application area: Fraud Detection
 - Goal: Predict fraudulent cases in credit card transactions.
 - Approach:
 1. Use credit card transactions and the information on its account-holder as attributes.
 1. When does a customer buy, what does he buy?
 2. How often he pays on time? etc
 2. Label past transactions as fraud or fair transactions. This forms the class attribute.
 3. Learn a model for the class of the transactions.
 4. Use this model to detect fraud by observing credit card transactions on an account.



Classification: Application 3

- Application area: Customer Attrition/Churn:
 - Goal: To predict whether a customer is likely to be lost to a competitor.
 - Approach:
 1. Use detailed record of transactions with each of the past and present customers, to find attributes.
 - How often the customer calls, where he calls, what time-of-the day he calls most, his financial status, marital status, etc.
 2. Label the customers as loyal or disloyal.
 3. Find a model for loyalty.

From [Berry & Linoff] Data Mining Techniques, 1997

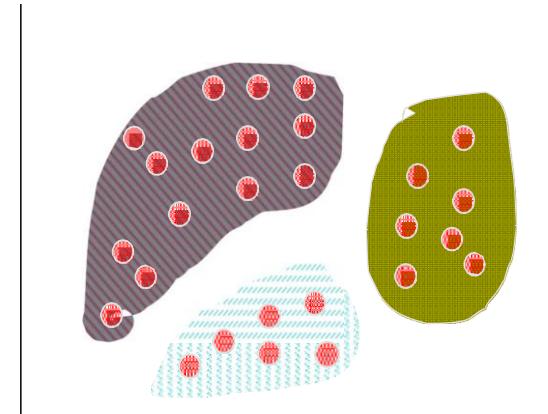
Classification: Application 4

- Application area: Sky Survey Cataloging
 - Goal: To predict class (star or galaxy) of sky objects, especially visually faint ones, based on the telescopic survey images (from Palomar Observatory).
 - 3000 images with 23,040 x 23,040 pixels per image.
 - Approach:
 - ◆ Segment the image.
 - ◆ Measure image attributes (features) - 40 of them per object.
 - ◆ Model the class based on these features.
 - ◆ Success Story: Could find 16 new high red-shift quasars, some of the farthest objects that are difficult to find!

From [Fayyad, et.al.] Advances in Knowledge Discovery and Data Mining, 1996

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.
- Goals
 1. intra-cluster distances are minimized
 2. inter-cluster distances are maximized
- Result
 - A descriptive grouping of data points

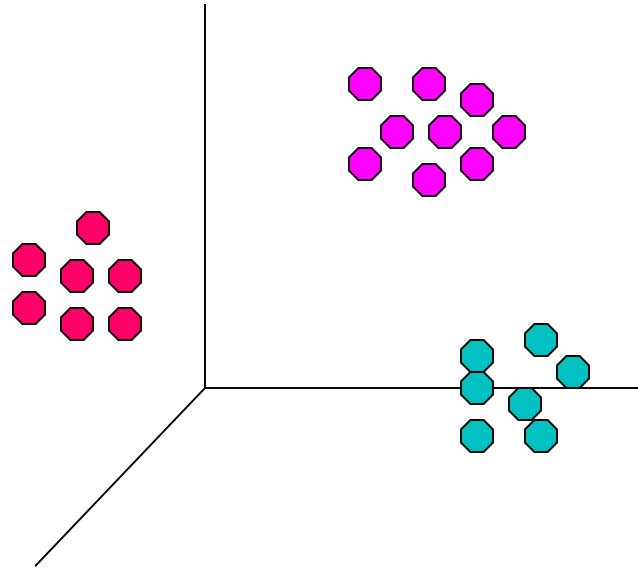


Illustrating Clustering

- Euclidean Distance Based Clustering in 3-D space.

Intraclasser distances
are minimized

Intercluster distances
are maximized



Clustering: Application 1

- Application area: Market segmentation
- Goal: Find groups of similar customers
 - where a group may be conceived as a market to be reached with a distinct marketing mix
- Approach:
 1. collect information about customers
 2. find clusters of similar customers
 3. measure the clustering quality by observing buying patterns after targeting customers with distinct marketing mixes



Clustering: Application 2

- Application area: Document Clustering
- Goal: Find groups of documents that are similar to each other based on terms appearing in them
- Approach
 - identify frequently occurring terms in each document
 - form a similarity measure based on the frequencies of different terms

Application Example: Grouping of articles in Google News



Illustrating Document Clustering

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

| <i>Category</i> | <i>Total Articles</i> | <i>Correctly Placed</i> |
|-----------------------------|-----------------------|-------------------------|
| <i>Financial</i> | 555 | 364 |
| <i>Foreign</i> | 341 | 260 |
| <i>National</i> | 273 | 36 |
| <i>Metro</i> | 943 | 746 |
| <i>Sports</i> | 738 | 573 |
| <i>Entertainment</i> | 354 | 278 |

Clustering of S&P 500 Stock Data

- Observe Stock Movements every day.
- Clustering points: Stock-{UP/DOWN}
- Similarity Measure: Two points are more similar if the events described by them frequently happen together on the same day.
 - We used association rules to quantify a similarity measure.

| | <i>Discovered Clusters</i> | <i>Industry Group</i> |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| 1 | Applied-Matl-DOWN,Bay-Net work-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Orac l-DOWN,SGI-DOWN, Sun-DOWN | Technology1-DOWN |
| 2 | Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOW N,Microsoft-DOWN,Scientific-Atl-DOWN | Technology2-DOWN |
| 3 | Fannie-Mae-DOWN,Fed-Home-Loan-DOW N, MBNA-Corp-DOWN,Morgan-Stanley-DOWN | Financial-DOWN |
| 4 | Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP | Oil-UP |

Association Rule Discovery: Definition

- Given a set of records each of which contain some number of items from a given collection;
 - discover **frequent itemsets** and produce **association rules** which will predict occurrence of an item based on occurrences of other items
 - 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 |

Frequent Itemsets
{Diaper, Milk, Beer}
{Milk, Coke}

Association Rules
{Diaper, Milk} --> {Beer}
{Milk} --> {Coke}

Association Rule Discovery: Application 1

- Application area: Marketing and Sales Promotion
 - Let the rule discovered be
 $\{Bagels, \dots\} \rightarrow \{Potato\ Chips\}$
 - 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!

Association Rule Discovery: Application 2

Application area: 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 –



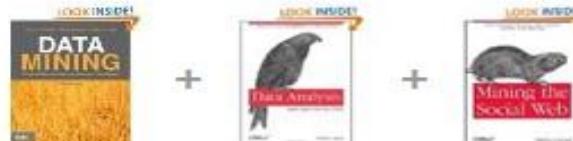
If a customer buys diaper and milk, then he is very likely to buy beer.

So, don't be surprised if you find six-packs stacked next to diapers!

- promote diapers to boost beer sales
- if selling diapers is discontinued, this will affect beer sales as well

Application area: Sales Promotion

amazon.com® Frequently Bought Together



Price For All Three: \$87.41

Add all three to Cart

Add all three to Wish List

Show availability and shipping details

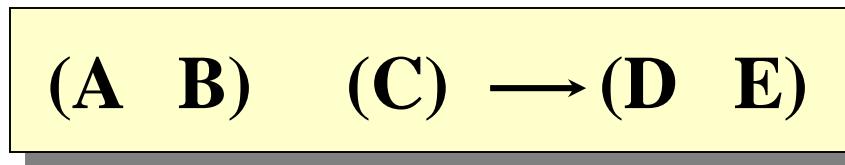
Association Rule Discovery: Application 3

- Application area: Inventory Management
 - Goal: A consumer appliance repair company wants to anticipate the nature of repairs on its consumer products and keep the service vehicles equipped with right parts to reduce on number of visits to consumer households.
 - Approach: Process the data on tools and parts required in previous repairs at different consumer locations and discover the co-occurrence patterns.

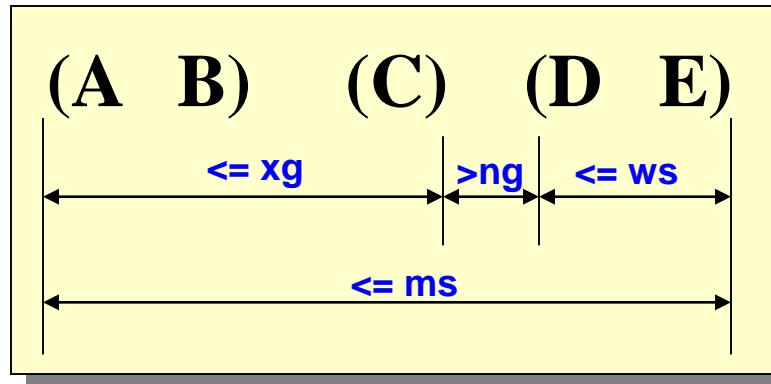


Sequential Pattern Discovery: Definition

- Given is a set of *objects*, with each object associated with its own *timeline of events*, find rules that predict strong **sequential dependencies** among different events.



- Rules are formed by first discovering patterns. Event occurrences in the patterns are governed by timing constraints.

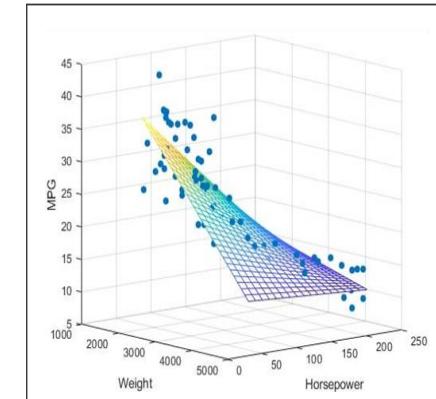
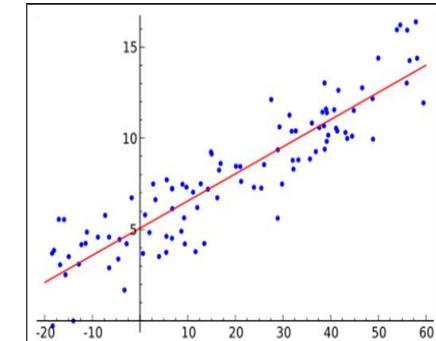


Sequential Pattern Discovery: Examples

- In telecommunications alarm logs,
 - (Inverter_Problem Excessive_Line_Current)
(Rectifier_Alarm) --> (Fire_Alarm)
- In point-of-sale transaction sequences,
 - Computer Store:
---- (PC) (Monitor) --> (Keyboard, Mouse....etc)
 - Computer Bookstore:
(Intro_To_Visual_C) (C++_Primer) -->
(Perl_for_dummies,Tcl_Tk)
 - Athletic Apparel Store:
(Shoes) (Racket, Racketball) --> (Sports_Jacket)

Regression

- Predict a value of a continuous variable based on the values of other variables, assuming a linear or nonlinear model of dependency.
Greatly studied in statistics
- Examples:
 - Predicting sales amounts of new product based on advertising expenditure.
 - Predicting the price of a house or car
 - Predicting miles per gallon (MPG) of a car as a function of its weight and horsepower
 - Predicting wind velocities as a function of temperature, humidity, air pressure, etc.
 - Time series prediction of stock market indices



Difference to classification: The predicted attribute is continuous, while classification is used to predict nominal attributes (e.g. yes/no)

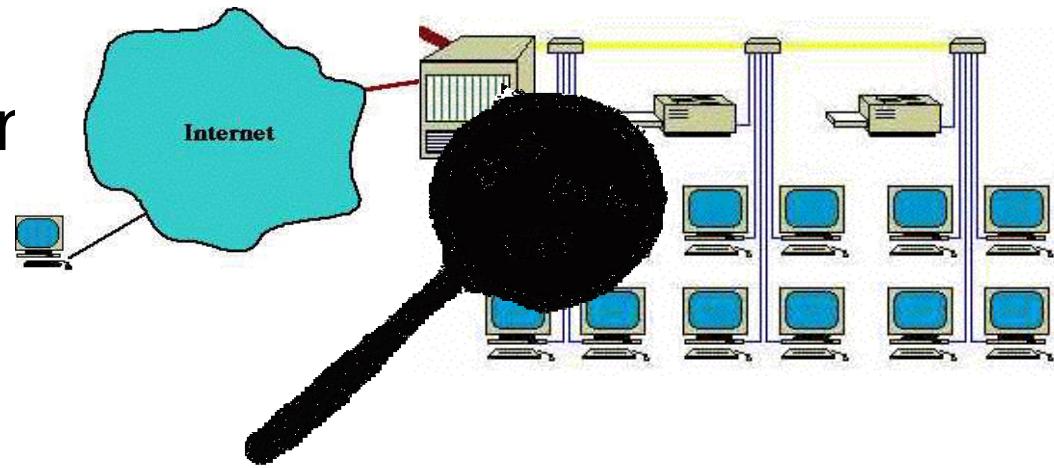
Deviation/Anomaly Detection

- Detect significant deviations from normal behavior (.....Outlier)
- Applications:

- Credit Card Fraud Detection

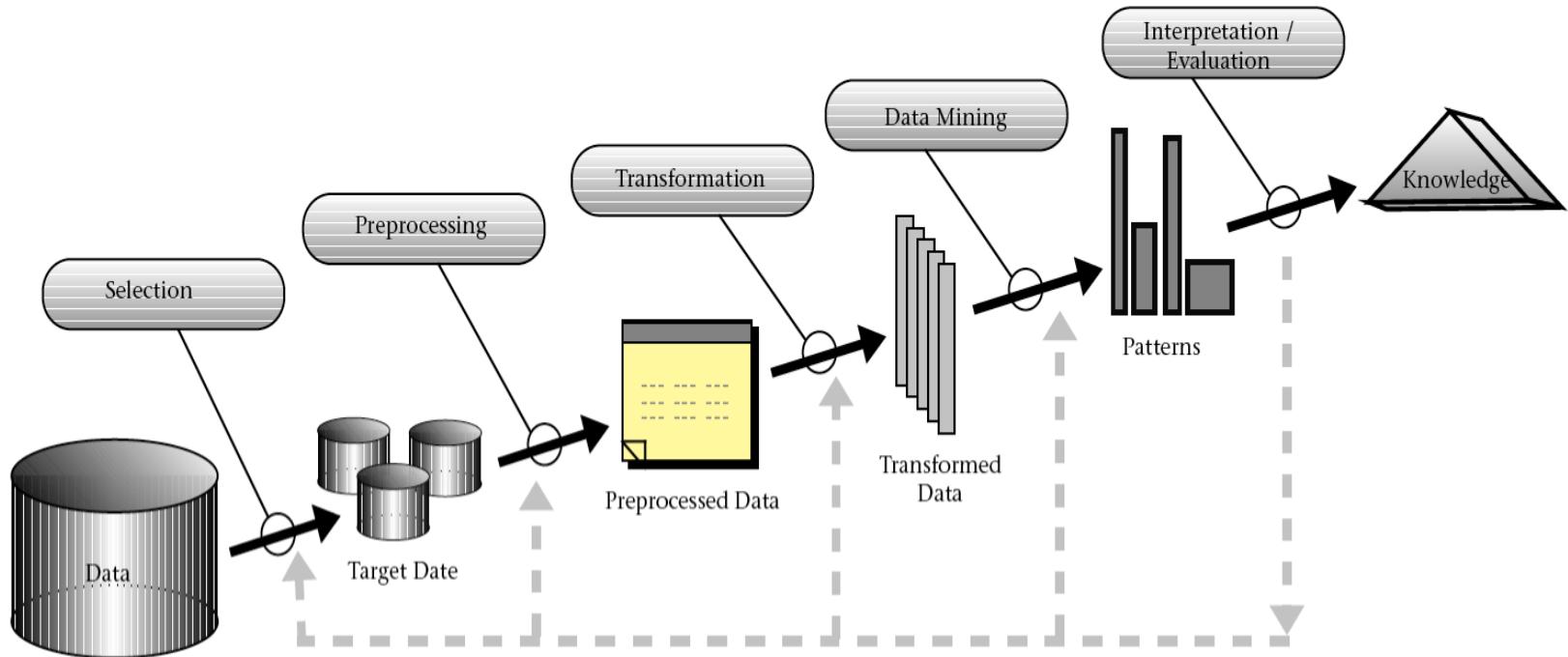


- Network Intrusion Detection



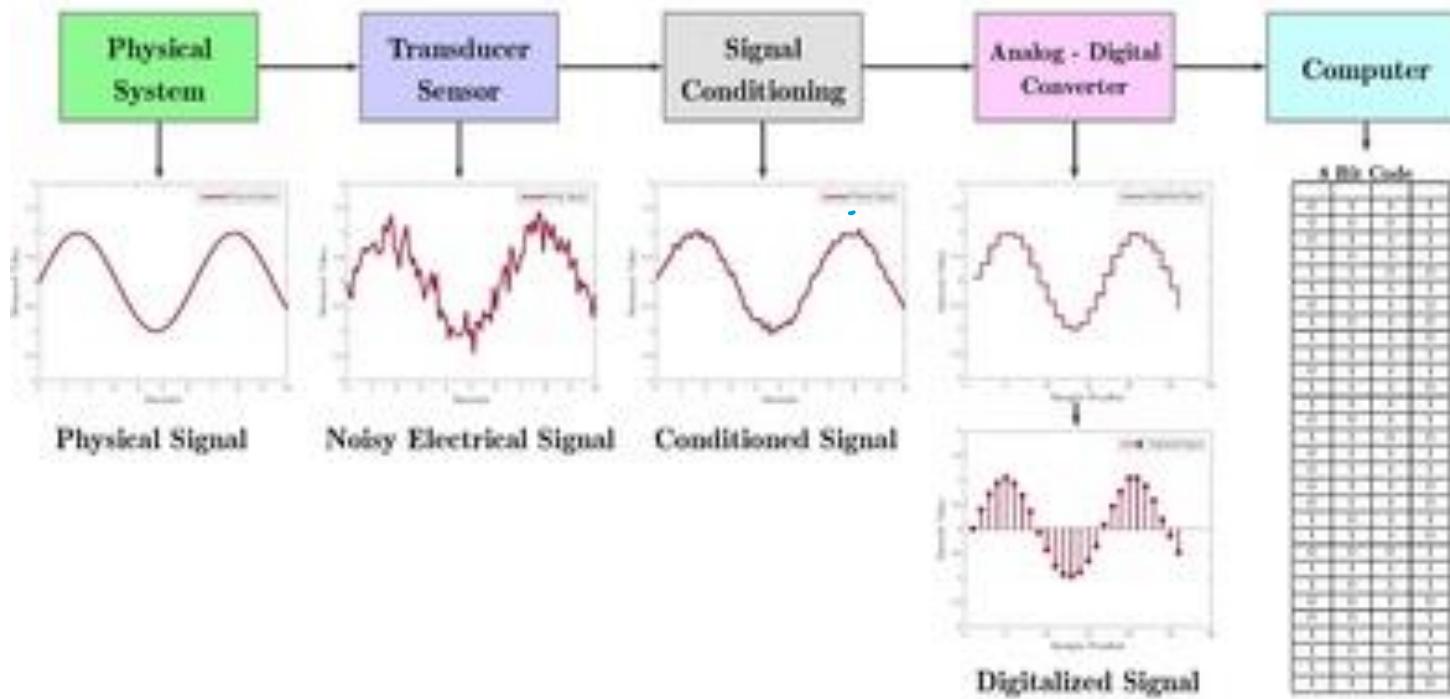
Typical network traffic at University level may reach over 100 million connections per day

The Data Mining Process



Data Acquisition System

Digital Data Acquisition System



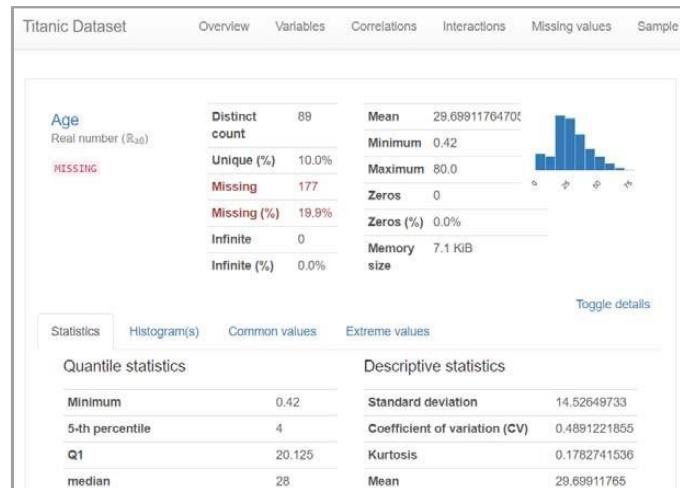
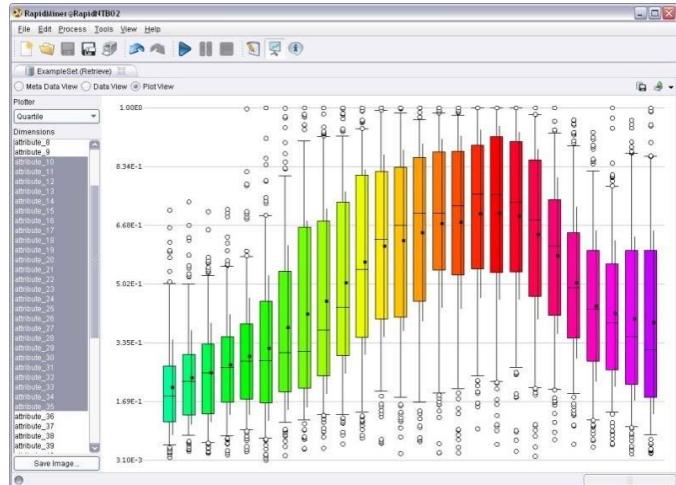
Selection and Exploration

– Selection

- What data is potentially useful for the task at hand?
- What data is available?
- What do I know about the quality of the data?

– Exploration / Profiling

- Get an initial understanding of the data
- Calculate basic summarization statistics
- Visualize the data
- Identify data problems such as outliers, missing values, duplicate records



Challenges of Data Mining

- Scalability
- Dimensionality
- Complex and Heterogeneous Data
- Data Quality
- Data Ownership and Distribution
- Privacy Preservation
- Streaming Data

Challenges of Data Mining

Scalability:

- Datasets with sizes of gigabytes, terabytes or even petabytes.
- Massive datasets cannot fit into main memory.
- Need to develop scalable data mining algorithms to mine massive datasets.
- Scalability can also be improved by using sampling or developing parallel and distributed algorithms.

High Dimensionality:

- Data sets with hundreds or thousands of attributes.
- Example: Dataset that contain measurement of temperature at various location.
- Traditional data analysis techniques that were developed for low dimensional data.
- Need to develop data mining algorithms to handle high dimensionality.

Challenges of Data Mining

Heterogenous and Complex Data:

- Traditional data analysis methods deal with datasets containing attributes of same type (Continuous or Categorical)
- Complex data sets contain image, video, text etc.
- Need to develop mining methods to handle complex datasets.

Data Ownership and Distribution:

- Data is not stored in one location or owned by one organization.
- Data is geographically distributed among resources belonging to multiple entries.
- Need to develop distributed data mining algorithms to handle distributed datasets.

Key challenges:

- How to reduce the amount of communication needed for distributed data.
- How to effectively consolidate the data mining results from multiple sources.
- How to address data security issues.

Challenges of Data Mining

Non Traditional Analysis:

- Traditional statistical approach is based on a hypothesize-and-test paradigm.
- A hypothesis is proposed, an experiment is designed to gather the data, and then data is analysed with respect to the hypothesis.
- This process is extremely labour-intensive.
- Need to develop mining methods to automate the process of hypothesis generation and evaluation.

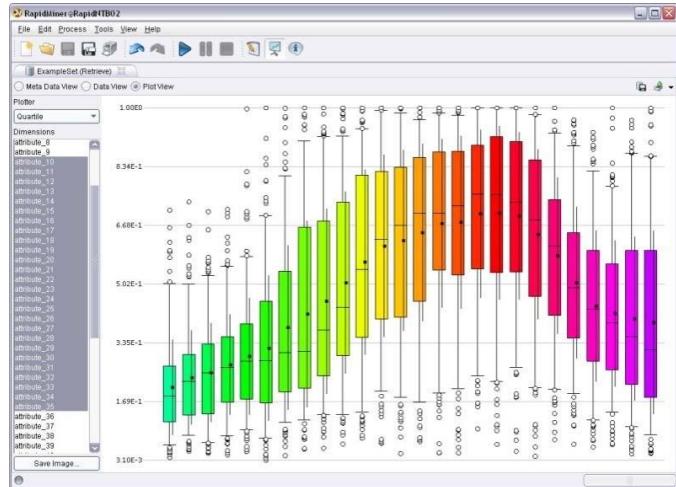
Selection and Exploration

– Selection

- What data is potentially useful for the task at hand?
- What data is available?
- What do I know about the quality of the data?

– Exploration / Profiling

- Get an initial understanding of the data
- Calculate basic summarization statistics
- Visualize the data
- Identify data problems such as outliers, missing values, duplicate records



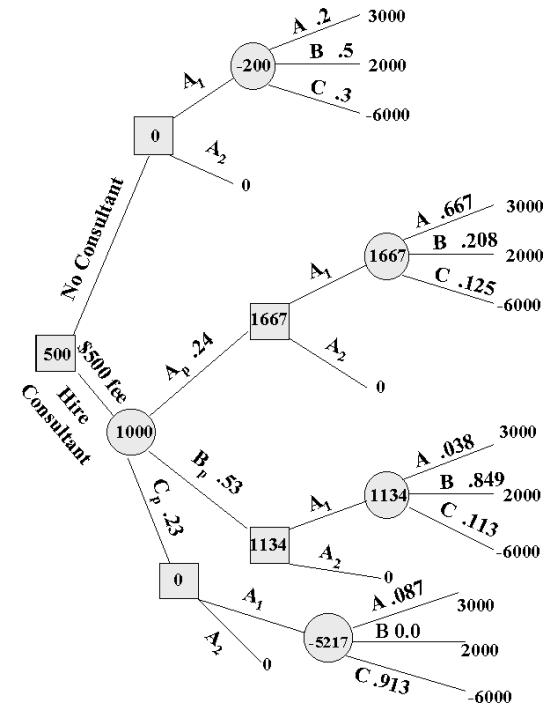
Preprocessing and Transformation

- Transform data into a representation that is suitable for the chosen data mining methods
 - scales of attributes (nominal, ordinal, numeric)
 - number of dimensions (represent relevant information using less attributes)
 - amount of data (determines hardware requirements)
- Methods
 - discretization and binarization
 - feature subset selection / dimensionality reduction
 - attribute transformation / text to term vector / embeddings
 - aggregation, sampling
 - integrate data from multiple sources
- Good data preparation is key to producing valid and reliable models
- Data integration and preparation is estimated to take **70-80%** of the time and effort of a data mining project

Data Mining

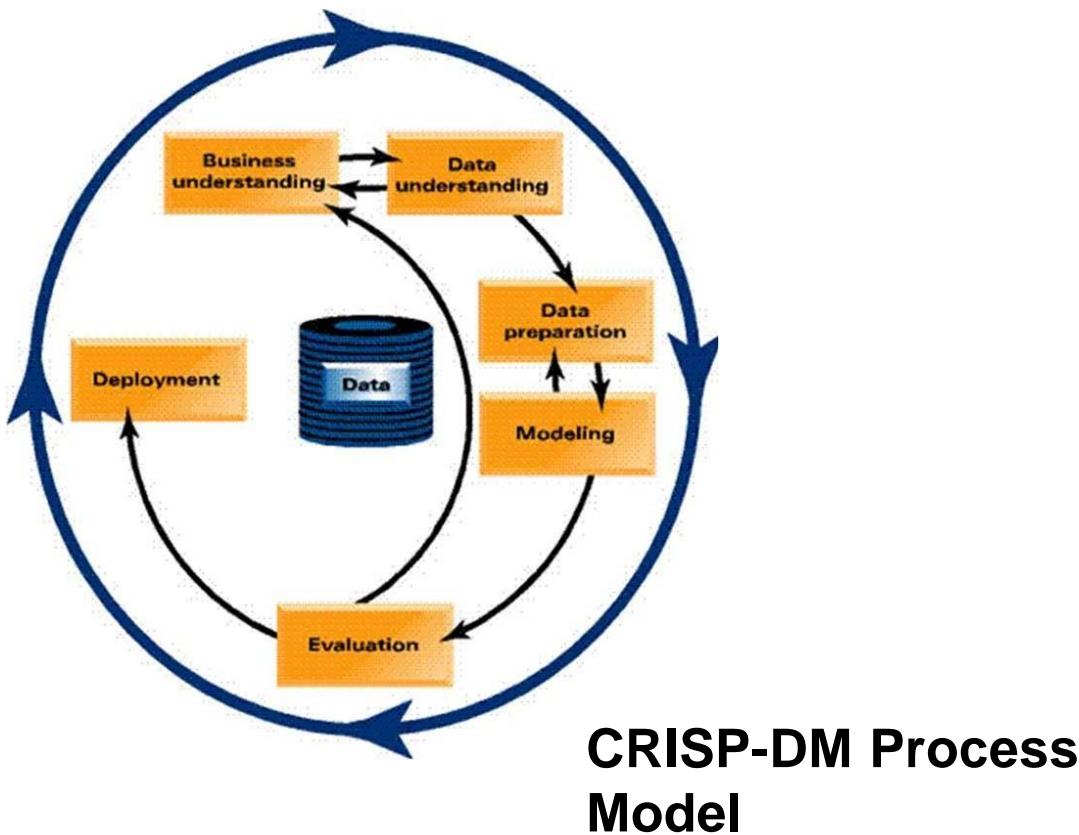
- Input: Preprocessed Data
- Output: Model / Patterns

1. Apply data mining method
2. Evaluate resulting model / patterns
3. Iterate
 - experiment with different parameter settings
 - experiment with multiple alternative methods
 - improve preprocessing and feature generation
 - increase amount or quality of training data

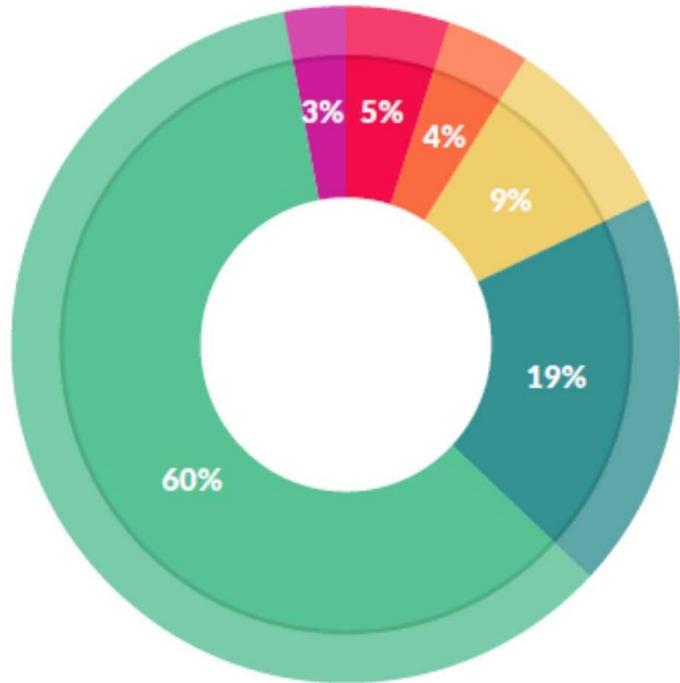


Deployment

- Use model in the business context
- Keep iterating in order to maintain and improve model



How Do Data Scientists Spend Their Days?



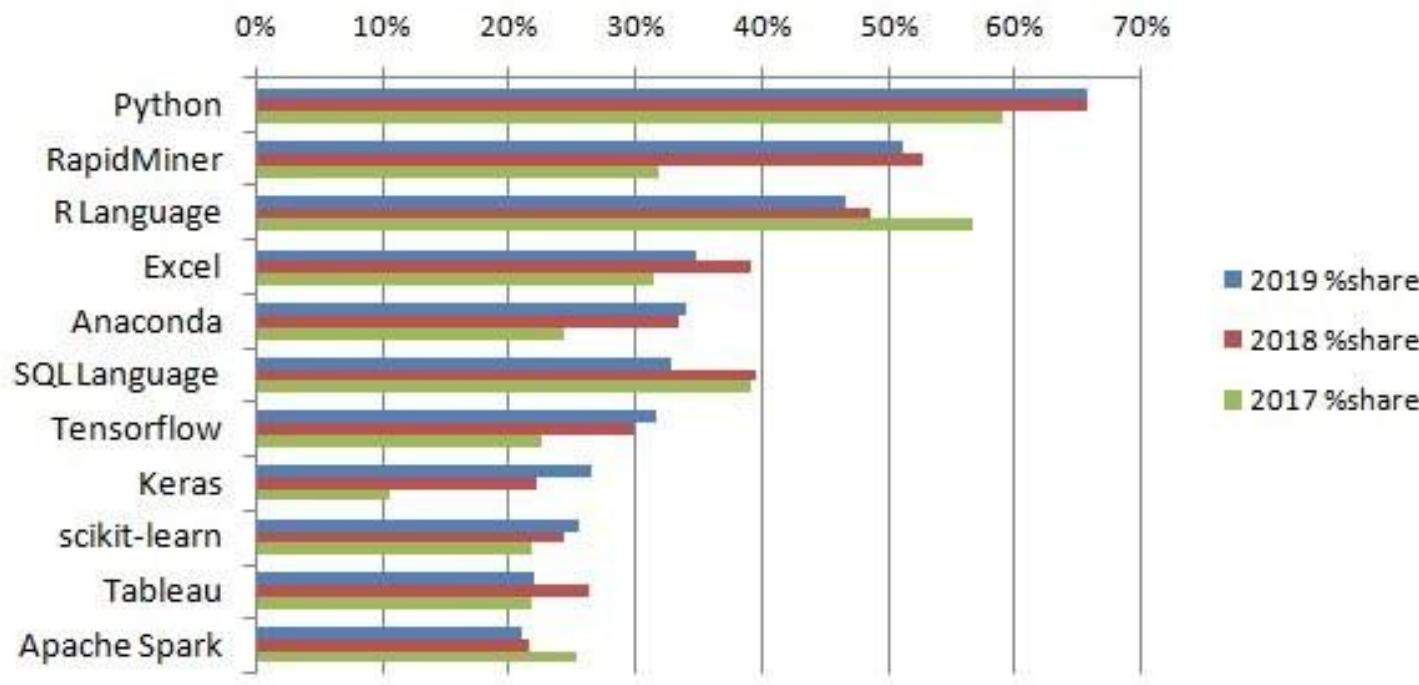
What data scientists spend the most time doing

- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

Source: CrowdFlower Data Science Report 2016: <http://visit.crowdflower.com/data-science-report.html>

Data Mining Software

Top Analytics, Data Science, Machine Learning Software 2017-2019, KDnuggets Poll



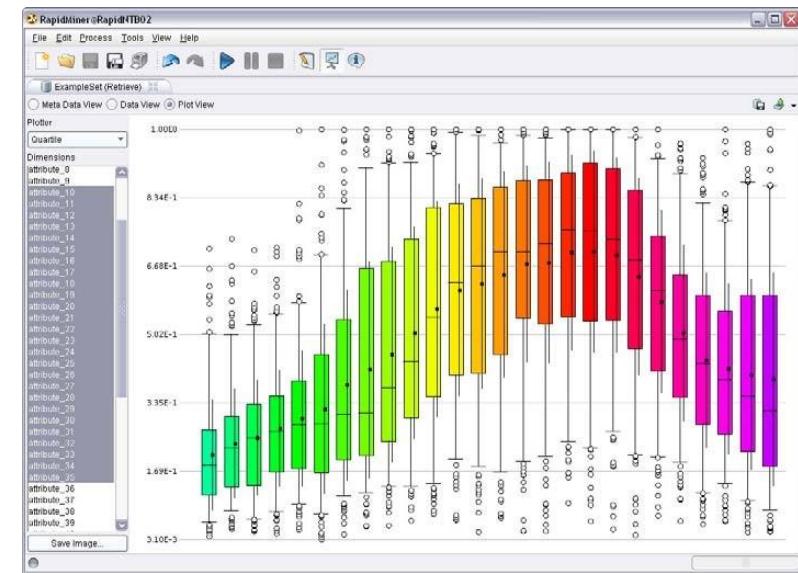
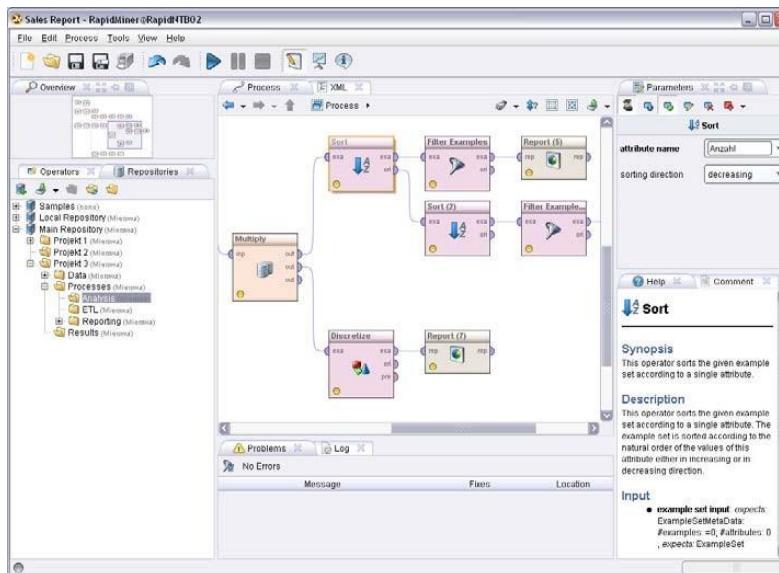
Source: KDnuggets online poll, 1800 votes

<https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html>

RapidMiner



- Powerful data mining suite
- Visual modelling of data mining pipelines
- Commercial tool, offering educational licenses



Gartner 2018 Magic Quadrant for Advanced Analytics Platforms



Literature – Rapidminer

1. Rapidminer – Documentation

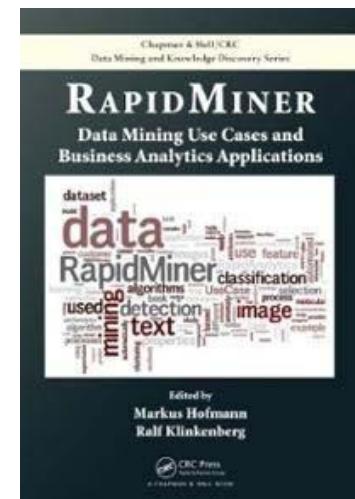
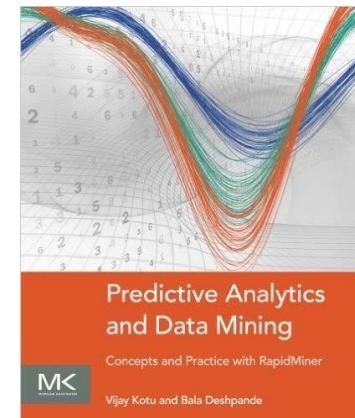
- <http://docs.rapidminer.com>
 - [**https://academy.rapidminer.com/catalog**](https://academy.rapidminer.com/catalog)

2. Vijay Kotu, Bala Deshpande: Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner. Morgan Kaufmann, 2014.

- covers theory and practical aspects using RapidMiner

3. Markus Hofmann, Ralf Klinkenberg:
RapidMiner: Data Mining Use Cases and Business Analytics Applications. Chapman & Hall, 2013.

- explains along case studies how to use simple and advanced Rapidminer features



Python

Use the Anaconda Python

–includes relevant packages, e.g.

- scikit-learn, pandas
- NumPy, Matplotlib

–includes Jupyter as development environment



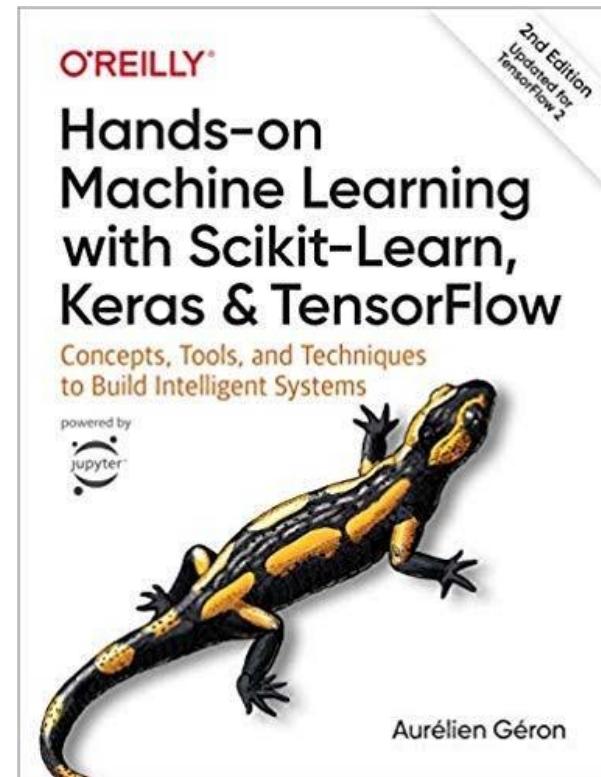
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import GridSearchCV

knn_estimator = KNeighborsClassifier()
parameters = {
    'n_neighbors': range(2, 9),
    'algorithm': ['ball_tree', 'kd_tree', 'brute']
}
stratified_10_fold_cv = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)
grid_search_estimator = GridSearchCV(knn_estimator, parameters, scoring='accuracy',
                                      cv=stratified_10_fold_cv)
grid_search_estimator.fit(iris_data, iris_target)
```

Slide Type Sub-Slide ▾

Literature – Python

1. Scikit-learn Documentation:
https://scikit-learn.org/stable/user_guide.html
2. Aurélien Géron: Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow.
2nd Edition, O'Reilly, 2019



Data Mining

2. Data

Prof Sunil Bhirud

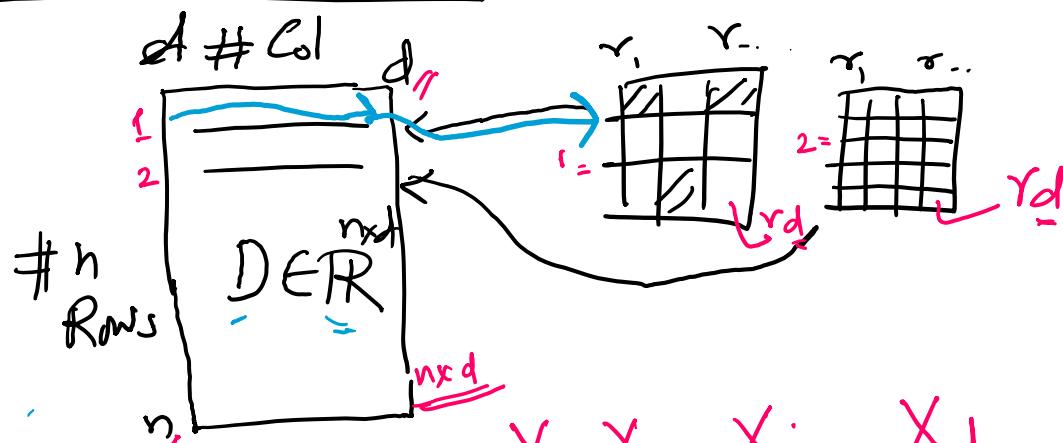


वीरमाता जिजाबाई टेक्नॉलॉजीकल इन्स्टिट्यूट
Veermata Jijabai Technological Institute
(VJTI Mumbai)



VJTI Mumbai

Data Matrix:



$$D = \begin{pmatrix} & \begin{matrix} X_1 & X_2 & \dots & X_d \end{matrix} \\ \begin{matrix} A_1 & A_2 & \dots & A_j & \dots & A_d \end{matrix} & \begin{matrix} x_{11} & x_{12} & x_{1j} & x_{1d} \\ x_{21} & x_{22} & x_{2j} & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{i1} & x_{i2} & x_{ij} & x_{id} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{nj} & x_{nd} \end{matrix} \end{pmatrix}$$

$\underline{n \times d}$

RN: $1 \times d$
Colm: $n \times 1$) View

Actual

Row View:-

$$\vec{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ij} \\ \vdots \\ x_{id} \end{bmatrix} = \begin{bmatrix} x_{i1} & x_{i2} & \dots & x_{id} \end{bmatrix}^T \quad \underline{1 \times d}$$

Column View:-

$$\vec{A}_{\cdot j} = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{bmatrix} = \begin{bmatrix} x_{1j} & x_{2j} & \dots & x_{nj} \end{bmatrix}^T \quad \underline{1 \times n}$$

a Important.

What is Data?

- Data:
 - Any observation that have been collected
 - 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 / Variable / Field / Characteristics / Feature
(Eye color, temperature etc..)

| 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 Objects

- Data sets are made up of data **objects**.
- A **data object** represents an **entity**.
- Examples:
 - **sales database**: customers, store items, sales
 - **medical database**: patients, treatments
 - **university database**: students, professors, courses
- Also called *samples* , *examples*, *instances*, *data points*, *objects*, *tuples*.
- Data objects are described by **attributes**.
- Database rows -> data objects; columns -> attributes.

Attribute Values

- **Attribute values** : Numbers or Symbols representing a characteristic or feature
 - E.g., customer _ID, Name, Address, Income etc.
- Distinction between attributes and attribute values
 - Same attribute can be mapped to different attribute values
 - ◆ Example: Attribute (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

Attribute Types

- **Nominal:** categories, states, or “names of things” : NOT Ordered
 - $Hair_color = \{auburn, black, blond, brown, grey, red, white\}$
 - marital status, occupation, ID numbers, zip codes
- **Binary**
 - Nominal attribute with only 2 states (0 and 1)
 - Symmetric binary: both outcomes equally important
 - ◆ e.g., gender
 - Asymmetric binary: outcomes not equally important.
 - ◆ e.g., medical test (positive vs. negative)
 - ◆ Convention: assign 1 to most important outcome (e.g., HIV positive)
- **Ordinal: Ordered- Differences are Meaningless**
 - Values have a meaningful order (ranking) but magnitude between successive values is not known.
 - $Size = \{small, medium, large\}$, grades, army rankings, Color (Spectrum)

Numeric Attribute Types

- Quantity (integer or real-valued)
- **Interval : Ordered. Differences are Meaningful**
 - ◆ No natural Zero
 - ◆ Measured on a scale of **equal-sized units**
 - E.g., *temperature in C° or F° (0° is a measured temperature), calendar dates*
 - ◆ No true zero-point
- **Ratio: Just like Interval**
 - ◆ Inherent **zero-point (Natural Zero)**
 - ◆ **Zero bank balance.**
 - ◆ 10 K° is twice as high as 5 K°).
 - e.g., *temperature in Kelvin, length, counts, monetary quantities*

Properties of Attribute Values

- The type of an attribute depends on which of the following properties it possesses:
 - Distinctness: $= \neq$
 - Order: $< >$
 - Addition: $+ -$
 - Multiplication: $* /$
 - Nominal attribute: distinctness
 - Ordinal attribute: distinctness & order
 - Interval attribute: distinctness, order & addition
 - Ratio attribute: all 4 properties

| 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. ($=, \neq$) | 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 |

| Attribute Level | Transformation | Comments |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Nominal | Any permutation of values | If all employee ID numbers were reassigned, would it make any difference? |
| Ordinal | <p>An order preserving change of values, i.e.,</p> $\text{new_value} = f(\text{old_value})$ <p>where f is a monotonic function.</p> | An attribute encompassing the notion of good, better best can be represented equally well by the values {1, 2, 3} or by {0.5, 1, 10}. |
| Interval | $\text{new_value} = \underline{a} * \underline{\text{old_value}} + \underline{b}$ <p>where a and b are constants</p> | Thus, the Fahrenheit and Celsius temperature scales differ in terms of where their zero value is and the size of a unit (degree). |
| Ratio | $\text{new_value} = \underline{a} * \underline{\text{old_value}}$ | Length can be measured in meters or feet. |

Discrete and Continuous Attributes

□ Discrete Attribute

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

□ Continuous Attribute

- Real numbers as attribute values (Infinite 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.
- Usually a Measurement...

Types of data sets

□ Record

- Data Matrix
- Document Data
- Transaction Data

□ Graph

- World Wide Web
- Molecular Structures

□ Ordered

- Spatial Data
- Temporal Data
- Sequential Data
- Genetic Sequence Data

Important Characteristics of Structured Data

- **Dimensionality**
 - ◆ Curse of Dimensionality
- **Sparsity**
 - ◆ Only presence counts
- **Resolution**
 - ◆ Patterns depend on the scale
- **Distribution**
 - ◆ Centrality and dispersion

Record Data

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

| <i>Tid</i> | 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

- 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

- 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 | wi | 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 or Market Basket Data

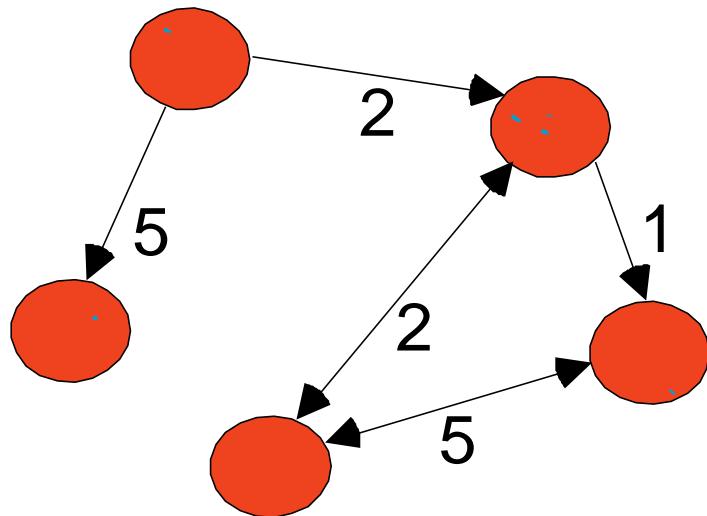
- A special type of record data, where
 - each **record (transaction)** involves a **set of items**.
 - For example, consider a grocery store. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items.



| <i>Transaction ID</i> | <i>Items</i> |
|-----------------------|---------------------------|
| 1 | Bread, Coke, Milk |
| 2 | Beer, Bread |
| 3 | Beer, Coke, Diaper, Milk |
| 4 | Beer, Bread, Diaper, Milk |
| 5 | Coke, Diaper, Milk |

Graph-Based Data

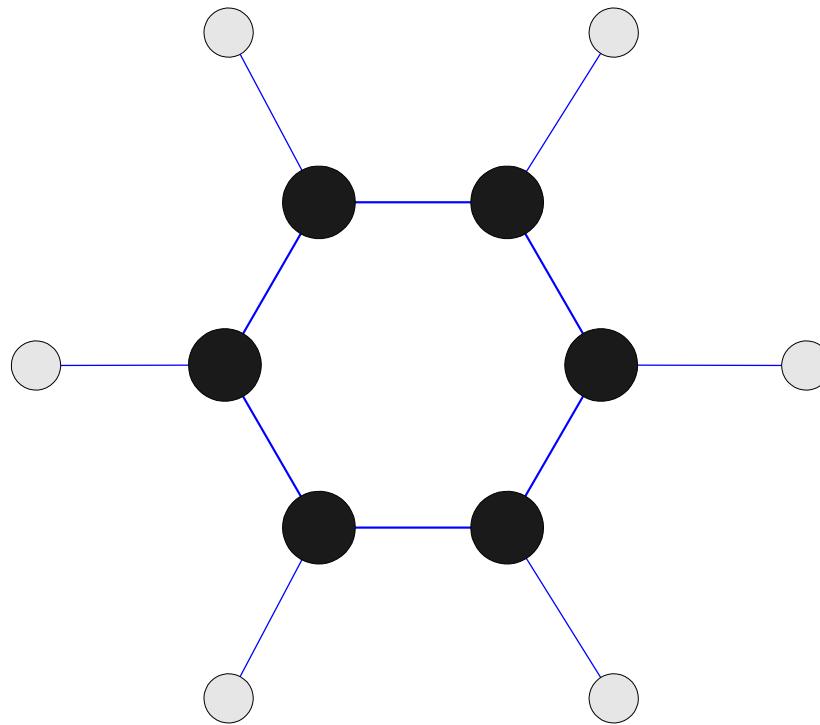
□ Examples: Generic graph and HTML Links



```
<a href="papers/papers.html#bbbb">  
Data Mining </a>  
<li>  
<a href="papers/papers.html#aaaa">  
Graph Partitioning </a>  
<li>  
<a href="papers/papers.html#aaaa">  
Parallel Solution of Sparse Linear System of Equations </a>  
<li>  
<a href="papers/papers.html#ffff">  
N-Body Computation and Dense Linear System Solvers
```

Chemical Data

□ Benzene Molecule: C_6H_6



Ordered Data

- Sequences of transactions

Items/Events

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



**An element of
the sequence**

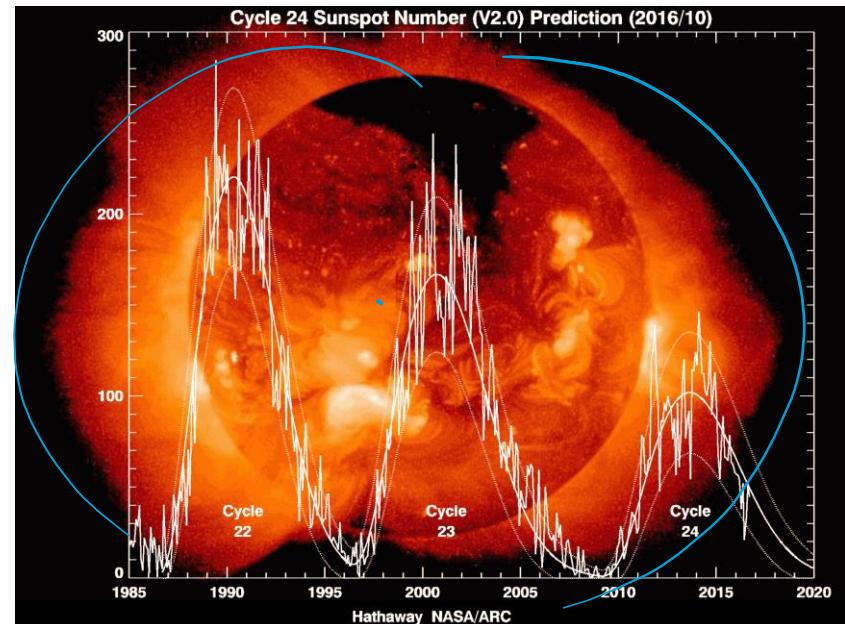
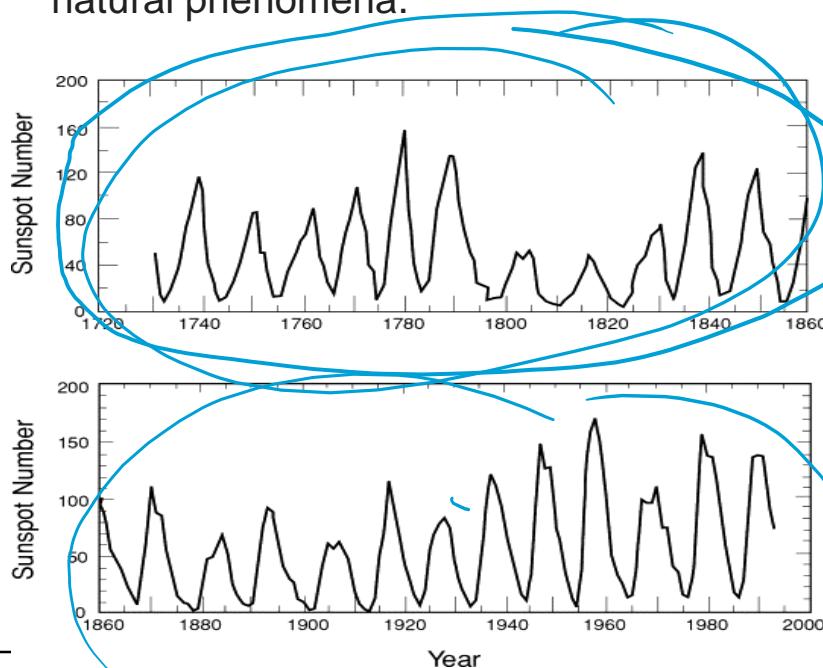
Ordered Data

- Genomic sequence data

```
GGTTCCGCCTTCAGCCCCGCGCC  
CGCAGGGCCCAGCCCCGCCGCCGTC  
GAGAAGGGCCCGCCTGGCGGGCG  
GGGGGAGGCAGGGGCCGCCGAGC  
CCAACCGAGTCCGACCAAGGTGCC  
CCCTCTGCTCGGCCTAGACCTGA  
GCTCATTAGGCAGCAGCGGACAG  
GCCAAGTAGAACACCGAAGCGC  
TGGGCTGCCTGCTGCGACCAGGG
```

Ordered Data: Time Series Data:

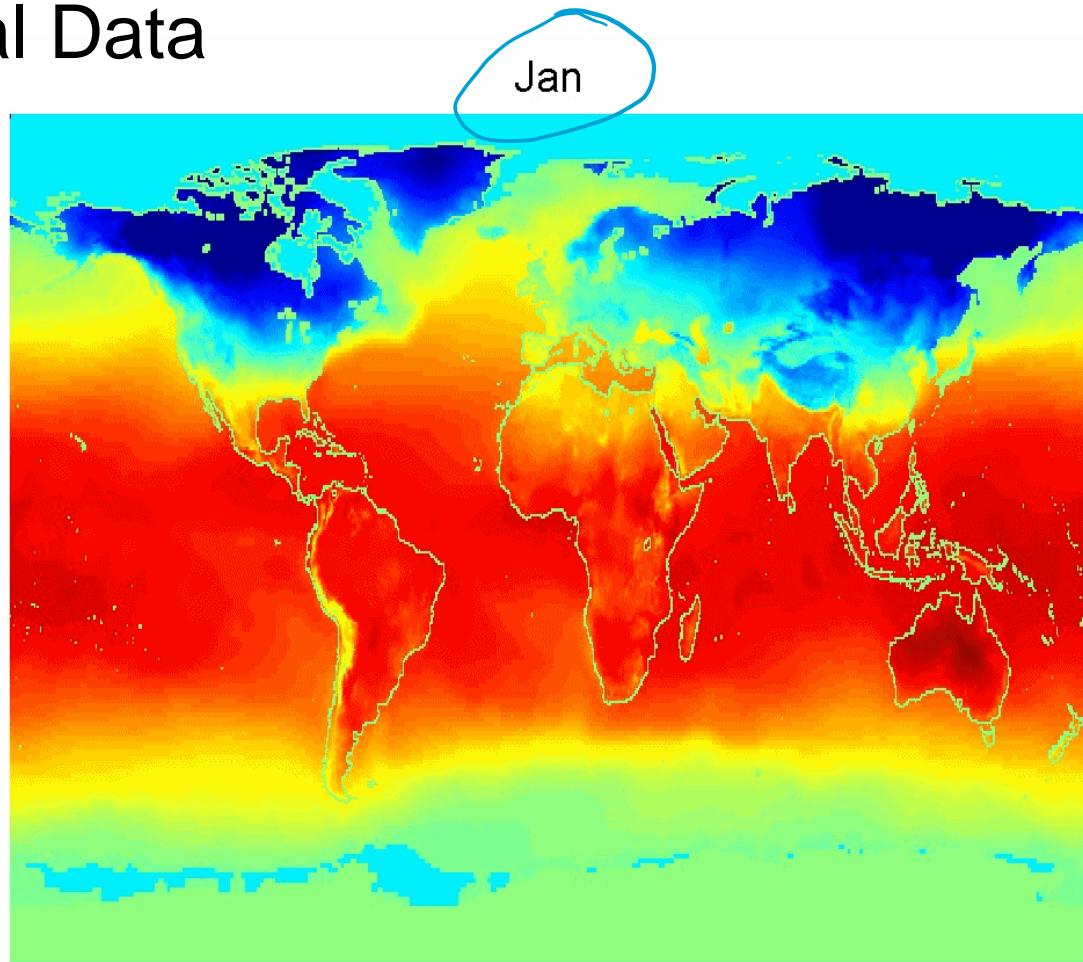
- The **solar cycle** or **solar magnetic activity cycle** is a nearly periodic 11-year change in the Sun's activity measured in terms of variations in the number of observed sunspots on the solar surface. Sunspots have been observed since the early 17th century and the sunspot time series is the longest continuously observed (recorded) time series of any natural phenomena.



Ordered Data

□ Spatio-Temporal Data

Average Monthly
Temperature of
land and ocean



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

Measurement and Data Collection Issues

Measurement Error:

- Error is the difference between the measured value and the actual value.

Data Collection Error:

- Omitting data objects
- Omitting attribute values
- Taking in-appropriate data objects: while collecting one specific species of animal data, you also collect data of similar species of animals.

Measurement and Errors

Any observation is composed of the true value plus some random error value.

But is that reasonable?

What if all error is random?

What if all error is not-random?
(Systematic Error)

$$X = T + e$$

Two Components:

- e_r • Random Error
- e_s • Systematic Error

$$X = T + e_r + e_s$$

X: Observed Value

T: True Value

e_r : Random Error

e_s : Systematic Error

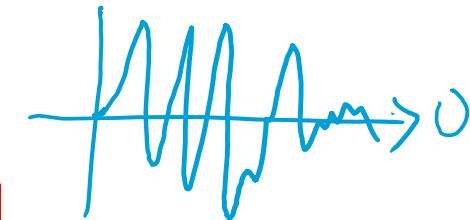
Random Error

Random errors: are caused by the sudden change in experimental conditions and noise and tiredness in the working persons.

- These errors are either positive or negative.
- It is due to factors which cannot be controlled.
- It may be too expensive to control them each time the experiment is conducted or the measurements are made.

Example: changes in humidity, unexpected change in temperature and fluctuation in voltage while taking readings.

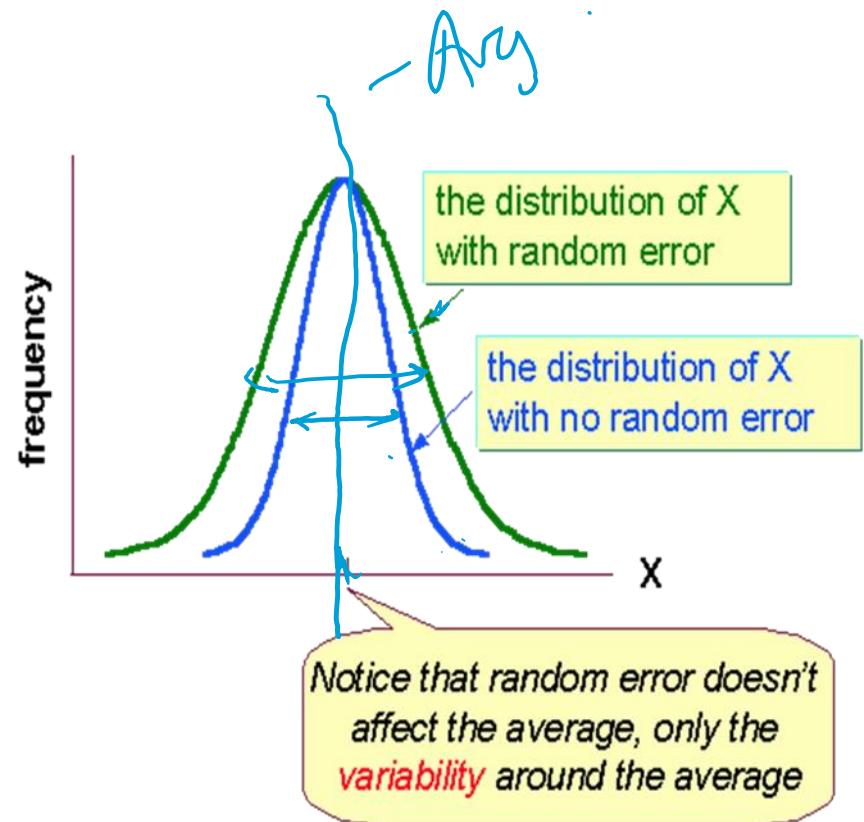
- These errors may be reduced by taking the average of a large number of readings.
- Random Error is sometimes called as NOISE.



For example:

- It is common for digital balances to exhibit random error in their least significant digit.
- Three measurements of a single object might read something like 0.9111g, 0.9110g, and 0.9112g.

- The concept of random error is closely related to the concept of precision.
- The higher the precision of a measurement instrument, the smaller the variability (standard deviation) of the fluctuations in its readings.



Systematic Error

Systematic error:

It occurs due to fault in the measuring device - are known as systematic errors.

- Usually they are called as Zero Error – a positive or negative error. Sometimes it is called as **BIAS in instrument**
- If the cause of the systematic error can be identified, then it usually **can be eliminated**.

Categories of Systematic Error:

- **Instrumental Error:** Imperfect calibration of measurement instruments (hysteresis or friction or Loading effect)

Categories of Systematic Error:

- **Environmental Error:**

Interference of the environment with the measurement process i.e external conditions (pressure, temp, humidity, magnetic field)

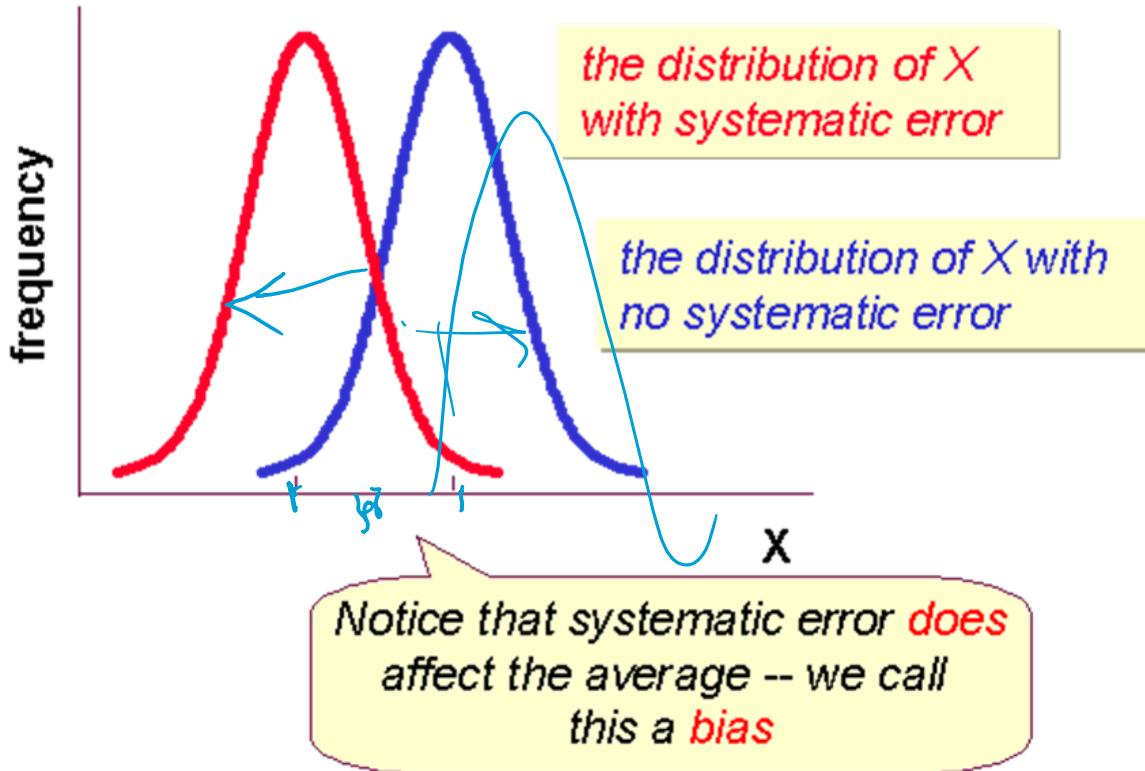
- **Observational Errors:**

Wrong observation or reading in the instrument
(Parallax)

- **Theoretical Errors:**

While designing it is assumed that temp of the surrounding will not change the reading, which is not true (some procedures / instruments are sensitive to temp / environment / other specific conditions)

Systematic Error



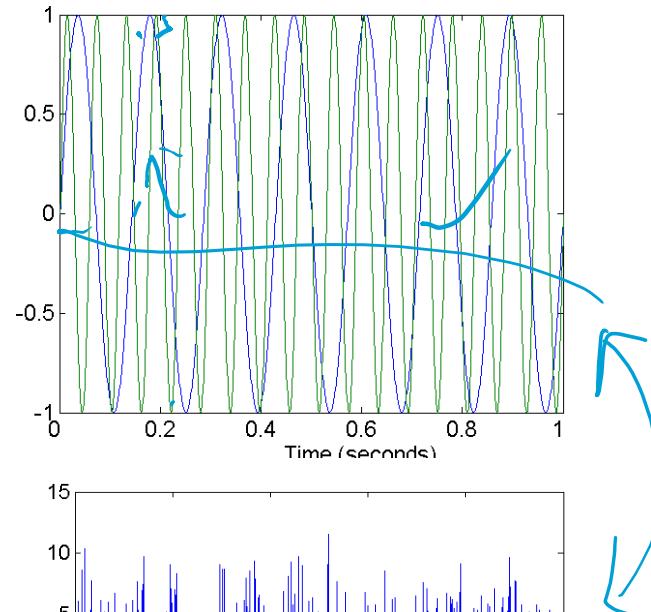
Noise and Artifacts

Noise: is the random component of a measurement.

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 + Noise

Data Artifacts

Is a data **flaw caused by equipment**, techniques or conditions.

Sources of data flaw:

Hardware, software errors, electromagnetic interference and
flaw in algorithm- prone to miscalculations

Types of Artefacts:

- Digital Artifacts : Digital camera that records a distorted or corrupted image.
- Visual Artifacts : Flaws in the visualizations such as user interfaces or streaming media
- Compression Artifacts : An Image become visibly distorted due to compression
- Noise Artifacts : Unwanted electrical fluctuations in radio reception
- Statistical Artifact : A flaw such as a bias in statistical data
- Radar Artifacts : Ghost objects in radar images/data/signal due to atmospheric effects or unfiltered echoes
- Sonic Artifacts : An unwanted sound as background noise on a film set. In some cases Artifacts are used a creative elements of music or films. Eg. Overdriving a bass signal for a fuzzy bass sound.

Precision

Precision: Closeness of repeated measurement (of the same quantity) to one another.

- Used for finding the consistency or reproducibility of the measurement.
- **High precision:** measurements are consistent or the repeated values of the reading are obtained.
- **Low precision:** value of the measurement varies.

Example:

Voltmeter readings :100V, 101V, 102V, 103V and 105V

The readings are nearly close to each other.

They are not exactly same because of the error.

The reading are close to each other, then we say that the readings are precise.

Bias:

Bias:

- A systematic variation of measurements from the quantity being measured.
- Difference between mean of the set of values and the known value of quantity being measured
- How close the measurements are to the true value.

Consider standard lab weighs of 1gm

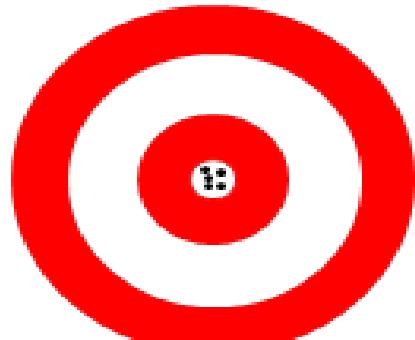
The weight of mass taken five times is { 1.015, 0.990, 1.013,
1.001, 0.986 }

Mean=1.001 and hence the bias is 0.001

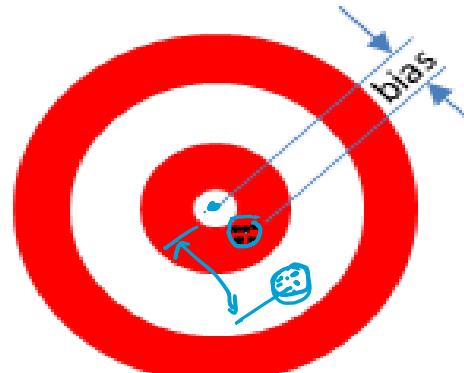
Precision is given by Standard Deviation=0.013

Bias and Precision

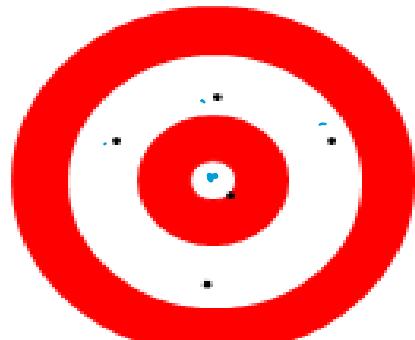
Being unbiased isn't always a good thing.



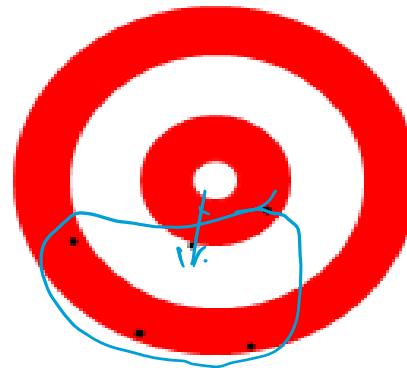
unbiased, precise



biased, precise



unbiased, imprecise

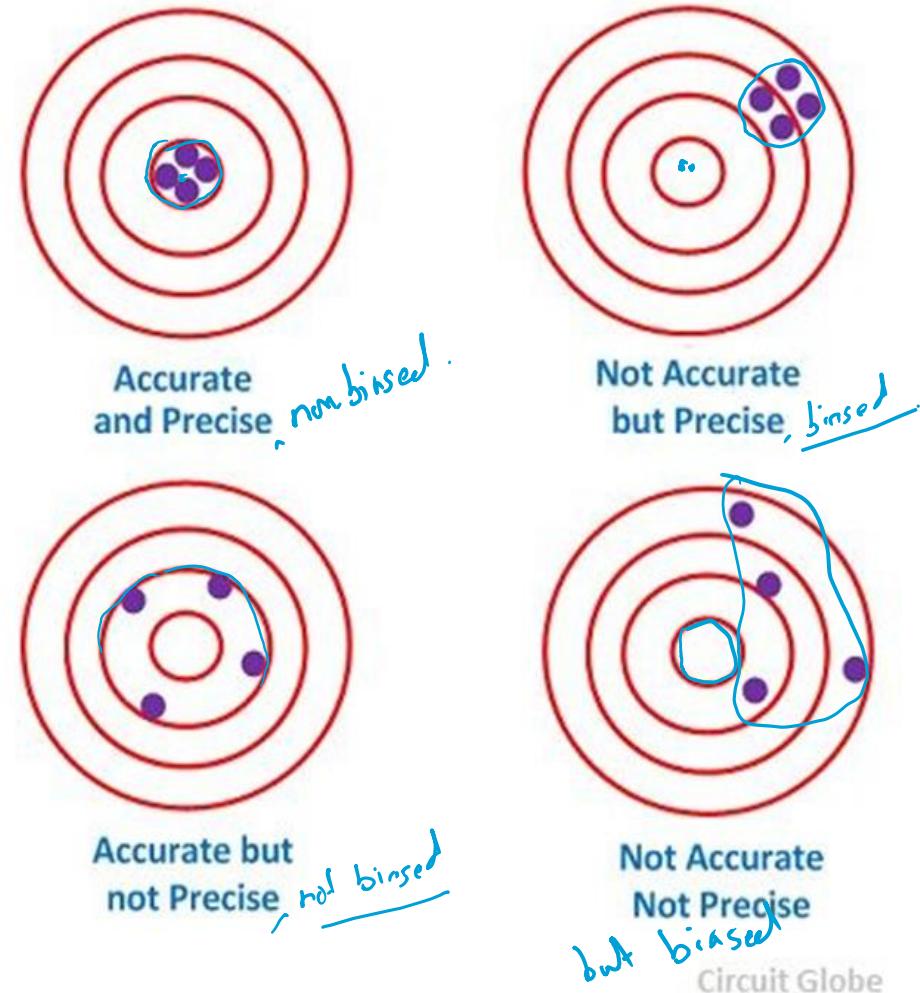


biased, imprecise

Accuracy

The closeness of the measurement value to the true/standard value of the quantity being measured.

It is the ability of the instrument to measure the accurate value.



Noise, Bias and Accuracy

Team A is *accurate*: The shots of the teammates are on the bull's-eye and close to one another.

The other three teams are inaccurate but in distinctive ways:

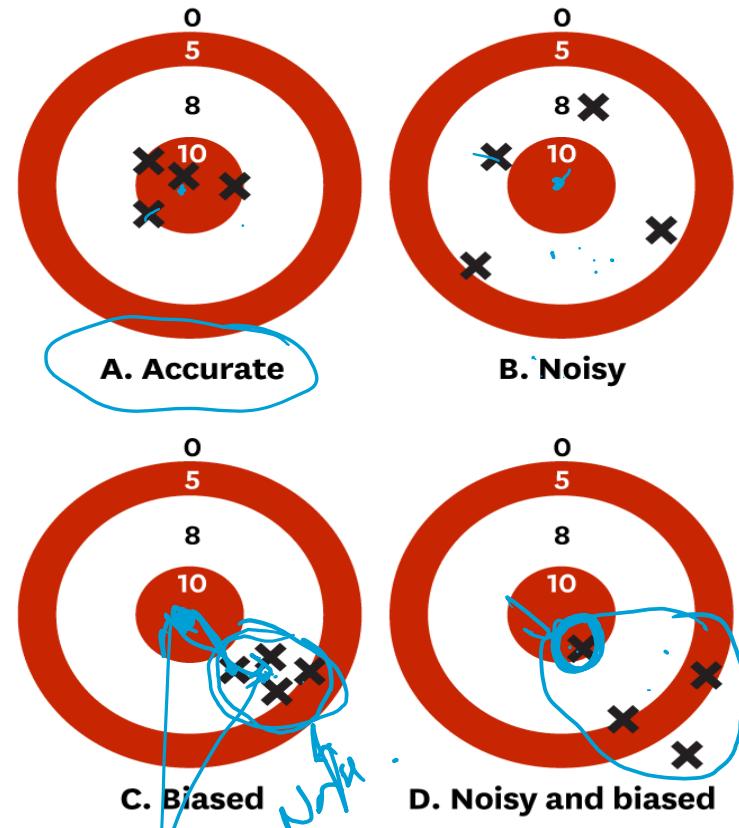
Team B is *noisy*: The shots of its members are centered around the bull's-eye but widely scattered.

Team C is *biased*: The shots all missed the bull's-eye but cluster together.

Team D is both *noisy* and *biased*.

As a comparison of teams A and B illustrates, an increase in noise always affects accuracy when there is no bias. When bias is present, increasing noise may actually cause a lucky hit, as happened for team D. Of course, no organization would put its trust in luck. Noise is always undesirable—and sometimes disastrous.

How Noise and Bias Affect Accuracy

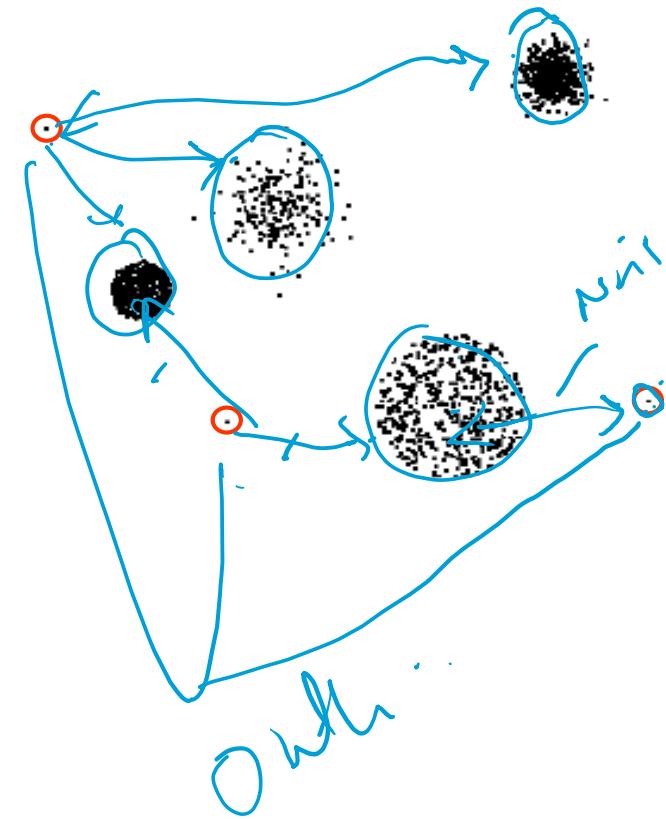


SOURCE DANIEL KAHNEMAN,
ANDREW M. ROSENFIELD,
LINNEA GANDHI, AND TOM BLASER
FROM "NOISE," OCTOBER 2016

© HBR.ORG

Outliers

- “Outliers” are data objects with characteristics that are considerably different than most of the other data objects in the data set
- “Outliers” is an observation that appears far away and diverges from an overall pattern in a sample.
- They are not necessarily wrong and are often the most interesting and informative observations in the sample



Outliers

Most common causes of outliers on a data set:

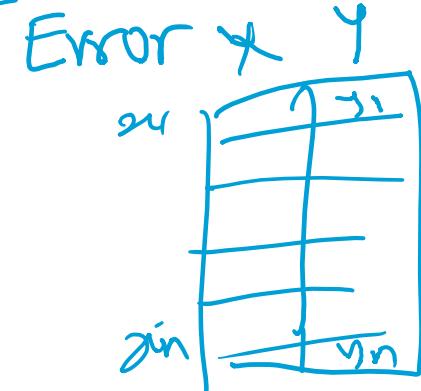
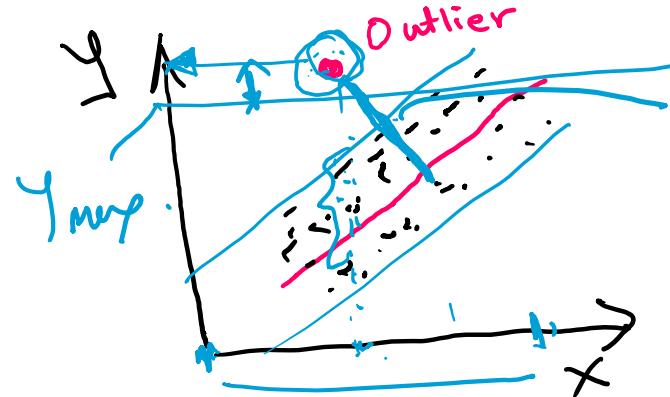
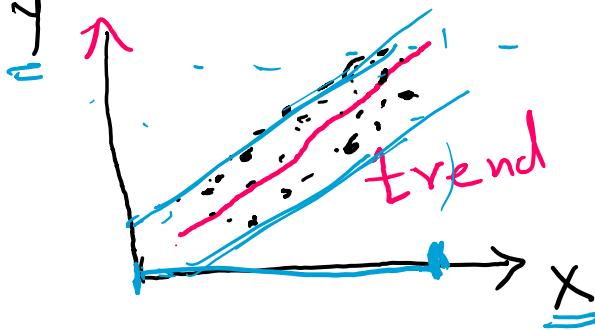
- Data entry errors (human errors)
- Measurement errors (instrument errors)
- Experimental errors (data extraction or experiment planning/executing errors)
- Intentional (dummy outliers made to test detection methods)
- Data processing errors (data manipulation or data set unintended mutations)
- Sampling errors (extracting or mixing data from wrong or various sources)
- Natural (not an error, novelties in data)

Some of the most popular methods for outlier detection are:

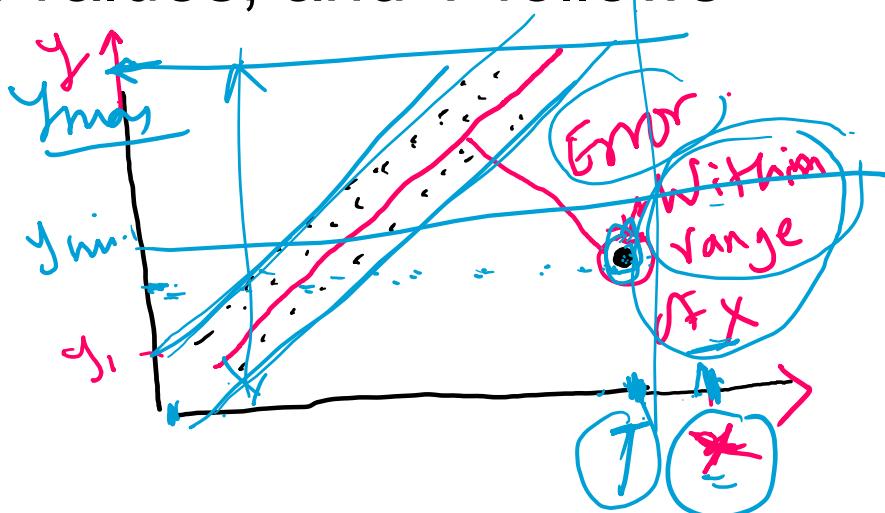
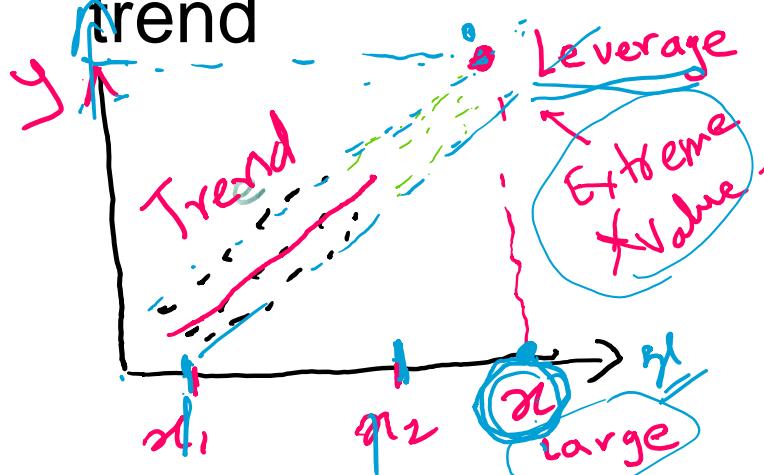
- Z-Score or Extreme Value Analysis (parametric)
- Probabilistic and Statistical Modeling (parametric)
- Linear Regression Models (PCA, LMS)
- Proximity Based Models (non-parametric)
- Information Theory Models
- High Dimensional Outlier Detection Methods (high dimensional sparse data)

Outlier and Leverage Point

- a. Outlier: Y- data that doesn't follow the general trend.

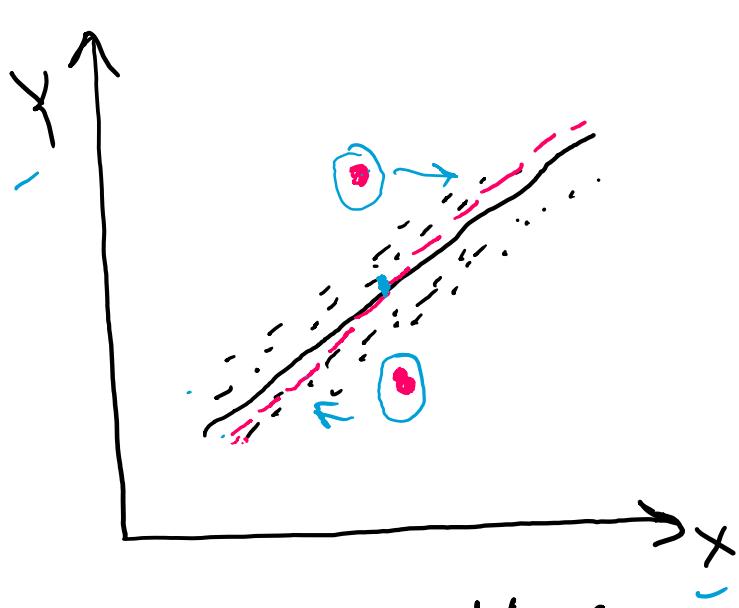


- b. Leverage: Extreme X values, and Y follows trend

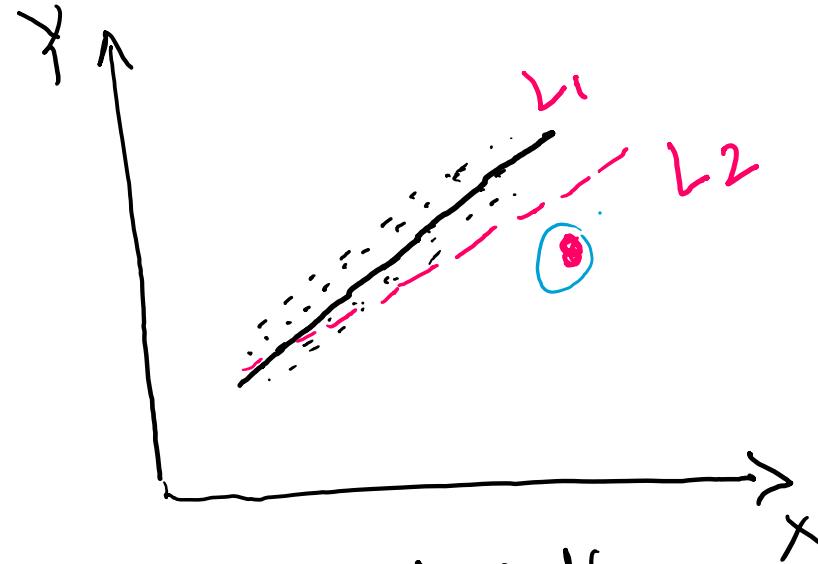


Outlier, Leverage Influence

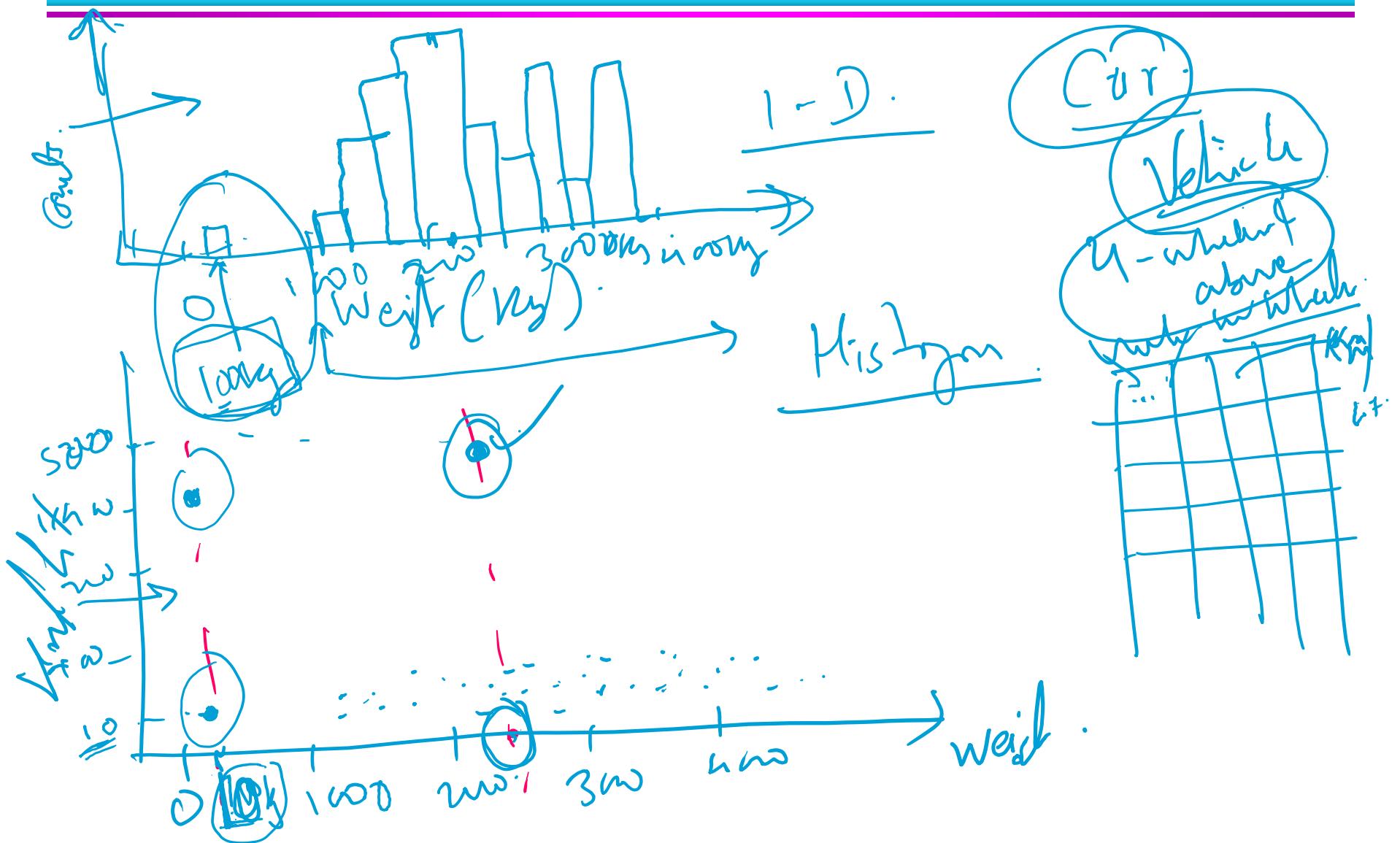
Outlier and Leverage influence the predictions



- Without outlier
- With Outlier



- Without Outlier
- With Outlier



Missing Values

□ Reasons for missing values

- Information is not collected (forgotten or lost)
(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)
- It is of no interest to the instance
- (measured parameter not related to patient condition)

□ Handling missing values

- Eliminate Data Objects
- Estimate Missing Values
- Ignore the Missing Value During Analysis
- Replace with all possible values (weighted by their probabilities)

Dealing with Missing Data

Use what you know about

- Why data are missing ?
- Distribution of missing data

Decide on the best analysis strategy to yield the estimates

Deletion Methods

Delete all cases with incomplete data and conduct analysis using only complete cases.

Advantages: Simple

Disadvantage: Loss of data if we discard all incomplete cases. So, in-efficient.

NOTE: if you use complete case analysis, then changes summary statistics for other variable, too.

Example: 15% missing data

| | Case 1 | | | | Case 2 | | | | Case 3 | | | |
|-----|--------|-----|-----|-----|--------|----|----|----|--------|----|----|----|
| | y1 | y2 | y3 | y4 | y1 | y2 | y3 | y4 | y1 | y2 | y3 | y4 |
| R1 | NA1 | NA2 | NA3 | NA4 | NA1 | | | | NA1 | | | |
| R2 | NA5 | NA6 | | | NA2 | | | | NA2 | | | |
| R3 | | | | | NA3 | | | | NA3 | | | |
| R4 | | | | | NA4 | | | | NA4 | | | |
| R5 | | | | | NA5 | | | | NA5 | | | |
| R6 | | | | | NA6 | | | | NA6 | | | |
| R7 | | | | | | | | | | | | |
| R8 | | | | | | | | | | | | |
| R9 | | | | | | | | | | | | |
| R10 | . | | | | | | | | | | | |

Case 1: Eliminate R1 and R2, Keep $8 \times 4 = 32$ data. 20% Loss

Case 2: Eliminate Y1 and Keep $10 \times 3 = 30$ data. 25% Loss

Case 3: Eliminate record R1 to R6, and Keep record 7 to 10 i.e. $4 \times 4 = 16$ data. 60% Loss

Listwise Deletion (Complete case analysis)

Only analyse cases with available data on each variable

Advantages: simple and compatible across analyses

Disadvantage: reduces statistical power (due to sample size), estimates may be biased.

Listwise deletion often produces unbiased regression slope estimates as long as missingness is not a function of outcome variable.

Pairwise Deletion (Available case analysis)

Analysis with all cases in which the variable of interest are present.

Advantages: Keeps as many cases as possible for each analysis, uses all information possible with each analysis.

Disadvantage: Cannot compare analyses because sample is different each time, sample size vary for each parameter estimation, can obtain nonsense results.

Compute the summary statistics using n_i observations not “n”
Compute correlation type statistics using complete pairs for both variables.

Example

List wise deletion

| Gender | Manpower | Sales |
|--------|----------|-------|
| M | 25 | 343 |
| F | 25 | 378 |
| M | 33 | 245 |
| F | 33 | 289 |
| M | 25 | 25 |
| M | 29 | 295 |
| M | 26 | 299 |

Pair wise deletion

| Gender | Manpower | Sales |
|--------|----------|-------|
| M | 25 | 343 |
| F | 25 | 378 |
| M | 33 | 245 |
| F | 33 | 289 |
| M | 25 | 25 |
| M | 29 | 295 |
| M | 26 | 299 |

Imputation Methods

1. Random sample from existing values:

Randomly generate an integer from 1 to $n - m_{\text{missing terms}}$, then replace the missing value with the corresponding observation that you chose randomly. ("m" number of missing points)

| Case | : 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8(n) |
|------|-------|-----|-----|-----|-----|-----|-----|------|
| Y_1 | : 3.4 | 3.9 | 2.6 | 1.9 | 2.2 | 3.3 | 1.7 | 2.4 |
| Y_2 | : 5.7 | 4.8 | 4.9 | 6.2 | 6.8 | 5.6 | -- | 5.8 |

Randomly generate number between 1 and 7: Say 3

Replace Y_2,7 by Y_2,3 = 4.9

Disadvantage: It may change the distribution of data

Imputation Method

2. Randomly sample from a reasonable distribution

e.g. If gender is missing and you have the information that there are about the same number of female and male in the population.

Gender $\sim \text{Ber}(p=0.5)$ or estimate p from the observed sample

Using random number generator from Bernoulli distribution for $p=0.5$, generate numbers for missing gender data

Disadvantage:

Distributional assumption may not be reliable (or correct even the assumption is correct, its representativeness is doubtful).

Imputation Methods

3. Mean / Mode Substitution

Replace missing value with the sample mean or mode.
Then, run analyses as if all complete cases.

Advantages: We can complete case analyses

Disadvantage: Reduces variability, weakens the correlation estimates because it ignores the relationship between variables, it creates artificial band.

Unless the proportion of missing data is low, do not use this method.

Last Observation Carried Forward

This method is specific to longitudinal data problems.

For each individual, NAN are replaced by the last observed value of that variable. Then, analyse data as if data were fully observed.

Disadvantage: The covariance structure and distribution change seriously.

Observation Time

| Cases | 1 | 2 | 3 | 4 | 5 | 6 |
|-------|-----|-----|-----|-----|-----|-----|
| 1 | 3.8 | 3.1 | 2.0 | 2.0 | 2.0 | 2.0 |
| 2 | 4.1 | 3.5 | 2.8 | 2.4 | 2.8 | 3.0 |
| 3 | 2.7 | 2.4 | 2.9 | 3.5 | 3.5 | 3.5 |

Imputation Methods

4. Dummy variable adjustment:

Create an indicator variable for missing value (1 for missing,
0 for observed)

Impute missing value to a constant (such as mean)

Include missing indicator in the regression

Advantage: Uses all information about missing observation

Disadvantage: results in biased estimates, not theoretically driven

Imputation Methods

5. Regression imputation:

Replace missing value with predicted score from regression equation.

Use complete cases to regress the variable with incomplete data on the other complete variables.

Advantages: Uses information from the observed data, gives better results than previous ones.

Disadvantage: Over-estimates model fit and correlation estimates, weakens variance.

Imputation Methods

Problem:

Regression assumes responses are normal distributed.

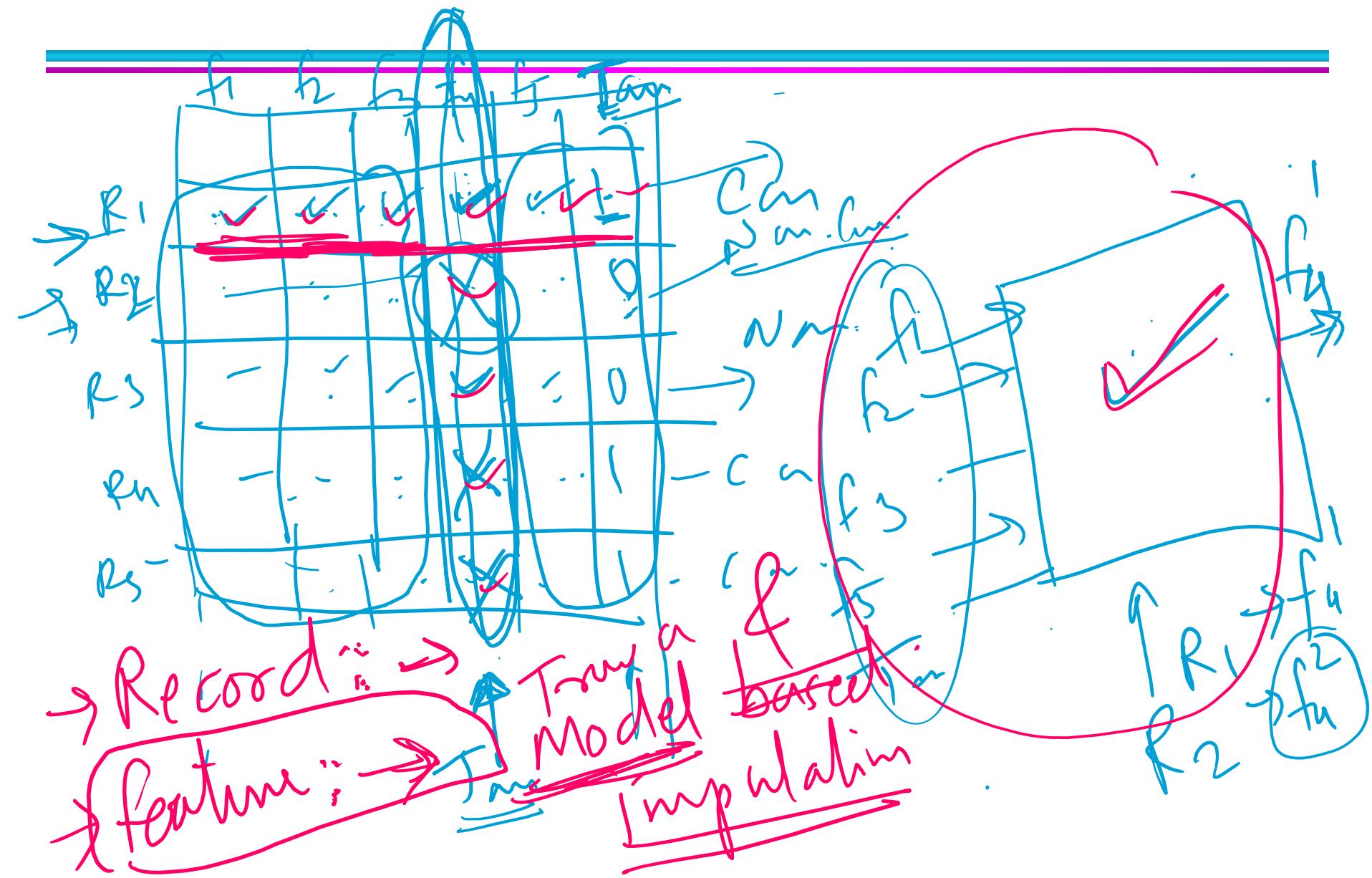
What if this assumption is unrealistic?

We can still use other models like logistic regression.

| Age | CC | RF | Brand |
|-----|-----|-----|--------|
| 14 | 350 | 165 | US |
| 31 | 200 | 75 | Europe |
| 17 | 302 | 110 | US |
| 25 | 400 | 150 | Japan |
| | 89 | 62 | |

mean mode
Predicting missing value

- Replace by some constant
- by mean:
- Ram replace from the observed values -
- Impute



Choice of the Imputation Method

1. Use a sample of your own dataset that does not contain any missing data (will serve as ground truth).
2. Introduce increasing proportions of missing data at random (e.g. 5–50 % in 5 % increments).
3. Reconstruct the missing data using the various methods.
4. Compute the sum of squared errors between the reconstructed and the original data, for each method and each proportion of missing data.
5. Repeat steps 1–4 a number of times (10 times for example) and compute the average performance of each method (average SSE).
6. Choose the method that performs best at the level of missing data in your dataset. E.g. if your data had 10 % of missing data, you would want to pick k-NN; at 40 % linear regression performs better (made-up data, for illustrative purpose only).

Duplicate Data

- Data set may include data objects that are duplicates, or almost duplicates of one another
 - Major issue when merging data from heterogeneous sources
- Examples:
 - Same person with multiple email addresses
- Data cleaning
 - Process of dealing with duplicate data issues

Dc - duplication

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
 - More “stable” data
 - ◆ Aggregated data tends to have less variability (some user conducts many sessions, his/her behavior can be represented by taking average of all sessions)

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

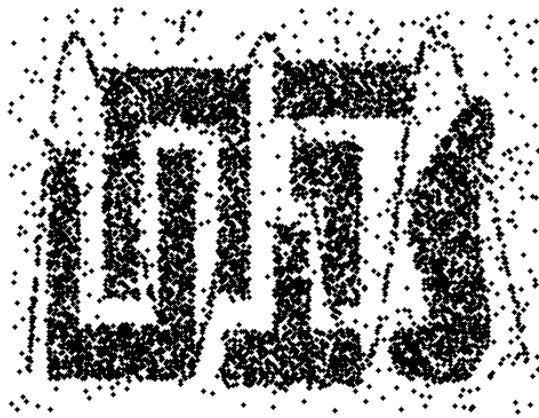
Sampling ...

- The key principle for effective sampling is the following:
 - using a sample will work almost as well as using the entire data sets, if the sample is representative
 - A sample is representative if it has approximately the same property (of interest) as the original set of data

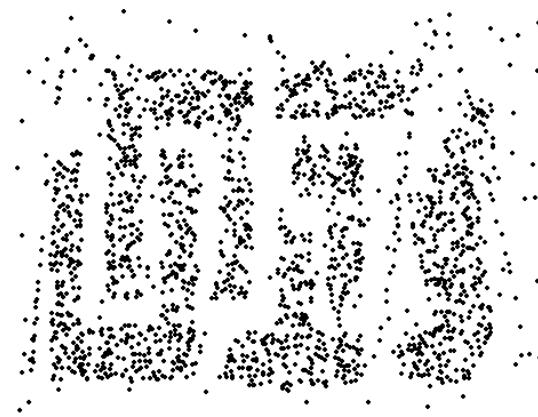
Types of Sampling

- 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

Sample Size



8000 points



2000 Points

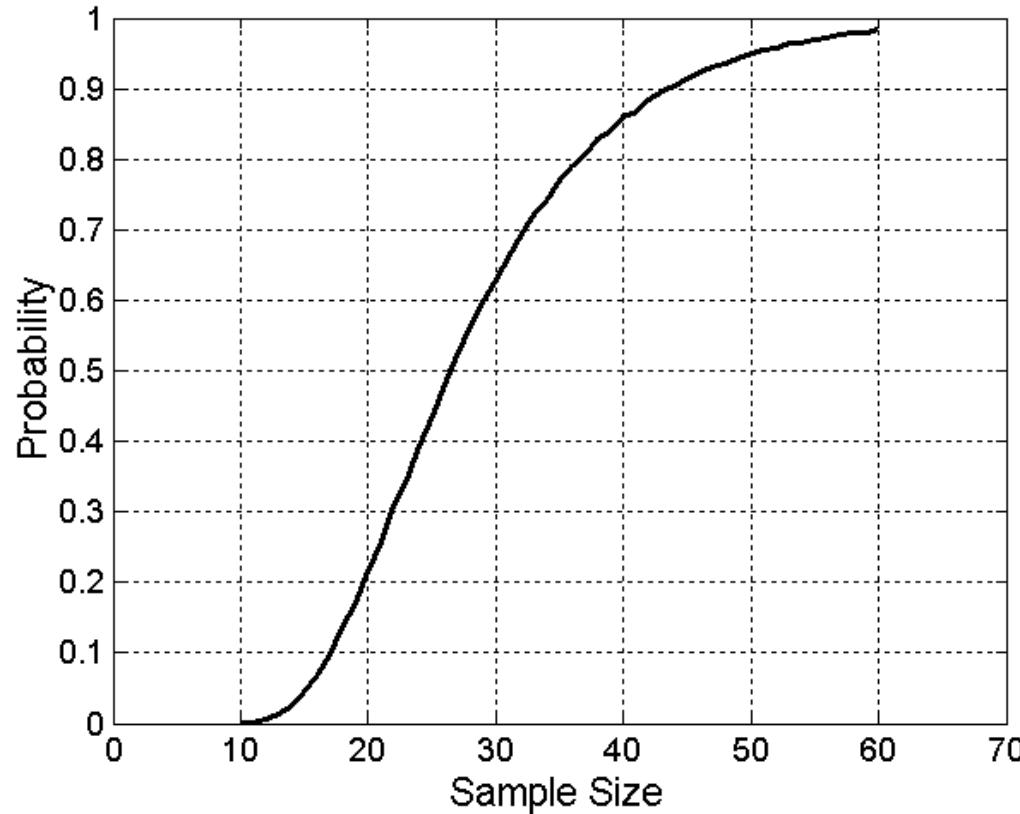


500 Points

Sample Size

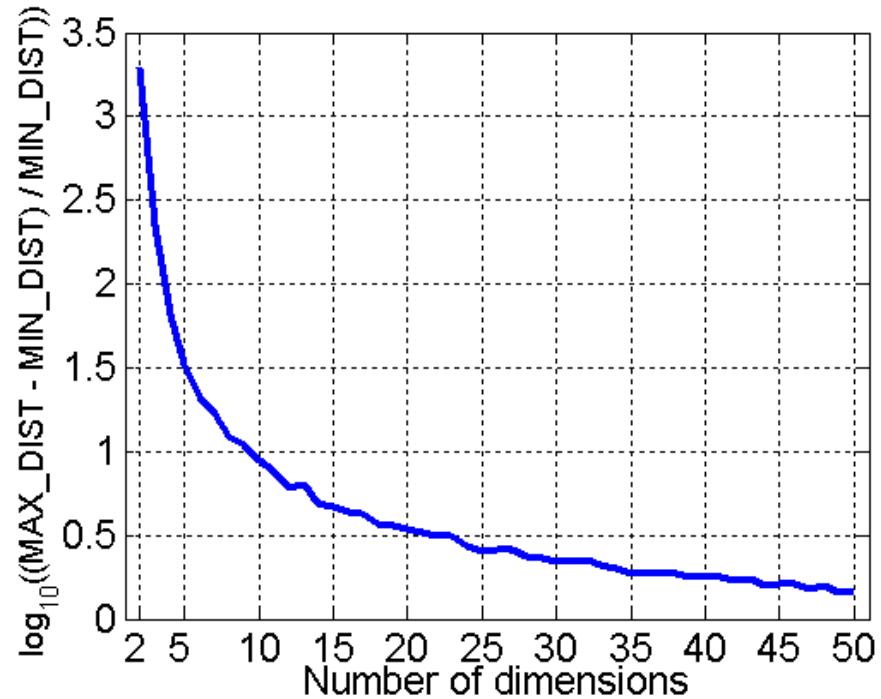
- What sample size is necessary to get at least one object from each of 10 groups.

• • • • •
• • • • •



Curse of Dimensionality

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



- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points

Dimensionality Reduction

□ Purpose:

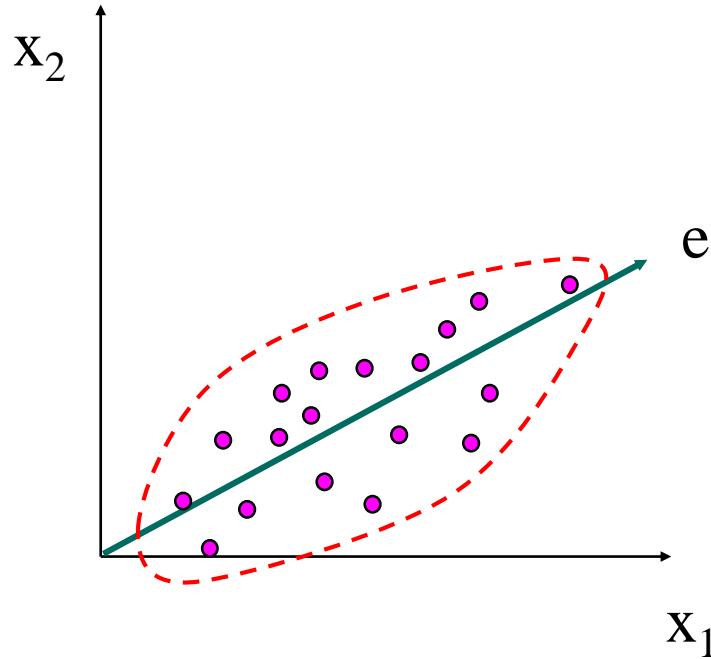
- Avoid curse of dimensionality
- Reduce amount of time and memory required by data mining algorithms
- Allow data to be more easily visualized
- May help to eliminate irrelevant features or reduce noise

□ Techniques

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

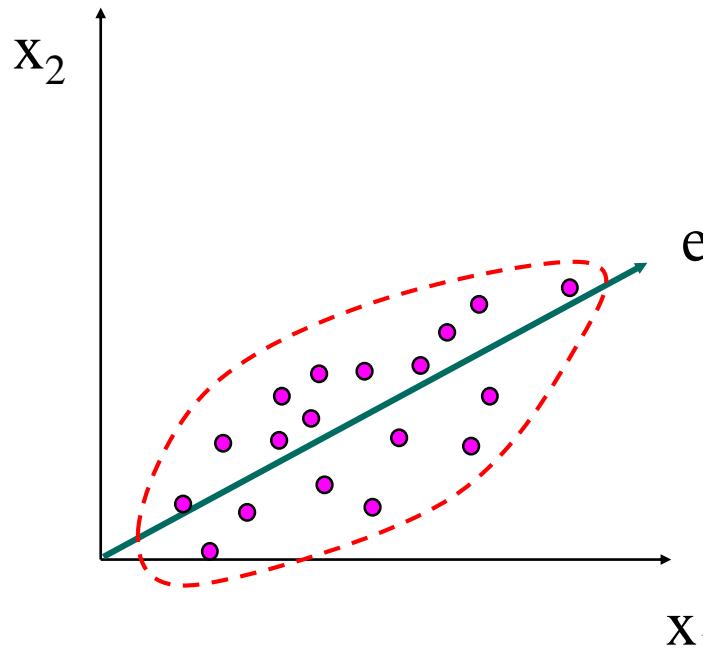
Dimensionality Reduction: PCA

- Goal is to find a projection that captures the largest amount of variation in data



Dimensionality Reduction: PCA

- Find the eigenvectors of the covariance matrix
- The eigenvectors define the new space



Feature Subset Selection

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

Feature Subset Selection

□ Techniques:

- Brute-force approach:
 - ◆ Try all possible feature subsets as input to data mining algorithm
- Embedded approaches:
 - ◆ Feature selection occurs naturally as part of the 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

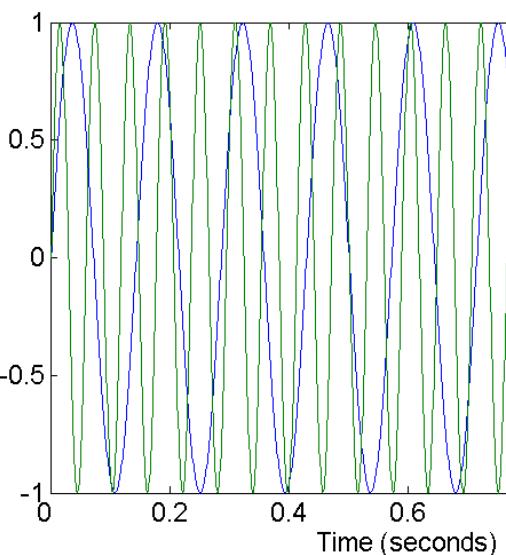
Feature Creation

- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes

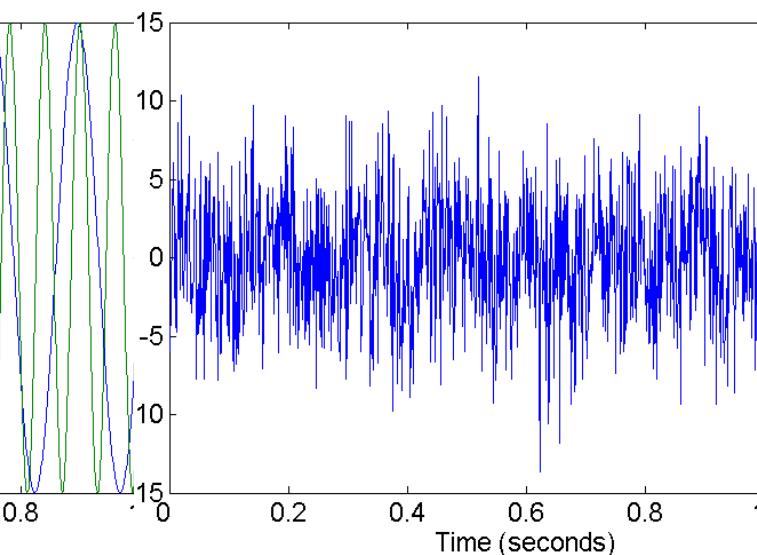
- Three general methodologies:
 - Feature Extraction
 - ◆ domain-specific
 - Mapping Data to New Space
 - Feature Construction
 - ◆ combining features

Mapping Data to a New Space

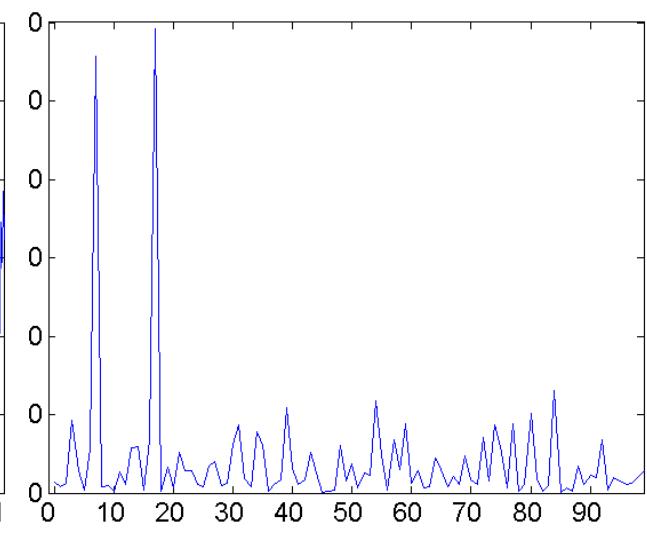
- Fourier transform
- Wavelet transform



Two Sine Waves



Two Sine Waves + Noise



Frequency

Discretization

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

Data Discretization

Data discretization transforms numeric data by mapping values to interval or concept labels.

- **Data discretization by binning:** This is a top-down unsupervised splitting technique based on a specified number of bins.
- **Data discretization by histogram analysis:** In this technique, a histogram partitions the values of an attribute into disjoint ranges called buckets or bins. It is also an unsupervised method.
- **Data discretization by cluster analysis:** In this technique, a clustering algorithm can be applied to discretize a numerical attribute by partitioning the values of that attribute into clusters or groups.
- **Data discretization by decision tree analysis:** Here, a decision tree employs a top-down splitting approach; it is a supervised method. To discretize a numeric attribute, the method selects the value of the attribute that has minimum entropy as a split-point, and recursively partitions the resulting intervals to arrive at a hierarchical discretization.
- **Data discretization by correlation analysis:** This employs a bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals, recursively. It is supervised method.

Simple Discretization: Binning

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

Binning Methods for Data Smoothing

The binning method can be used for smoothing the data (removing noise)

Unsorted data for price in dollars

Before sorting: 8, 16, 9, 15, 21, 21, 24, 30, 26, 27, 30, 34

First of all, sort the data

After Sorting: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34

Smoothing the data by equal frequency bins (Equal Depth)

Bin 1: 8, 9, 15, 16

Bin 2: 21, 21, 24, 26,

Bin 3: 27, 30, 30, 34

Smoothing by bin mean value

For Bin 1:

$$(8 + 9 + 15 + 16 / 4) = 12$$

Bin 1 = 12, 12, 12, 12

For Bin 2:

$$(21 + 21 + 24 + 26 / 4) = 23$$

Bin 2 = 23, 23, 23, 23

For Bin 3:

$$(27 + 30 + 30 + 34 / 4) = 30$$

Bin 3 = 30, 30, 30, 30

Smoothing by bin boundaries

How to smooth data by bin boundaries?

Put the minimum on the left side and maximum on the right side.

Middle values in bin boundaries move to its closest neighbor value with less distance.

Unsorted data for price in dollars:

Data: 8, 9, 15, 16, 21, 21, 24, 26, 27, 30, 30, 34

Smoothing the data by equal frequency bins

Bin 1: 8, 9, 15, 16

Bin 2: 21, 21, 24, 26,

Bin 3: 27, 30, 30, 34

Smooth data after bin Boundary

Before bin Boundary: Bin 1: 8, 9, 15, 16

Here, 8 is the minimum value and 16 is the maximum value. 9 is near to 8, so 9 will be treated as 8. 15 is more near to 16 and farther away from 8. So, 15 will be treated as 16.

After bin Boundary: Bin 1: 8, 8, 16, 16

Before bin Boundary: Bin 2: 21, 21, 24, 26,

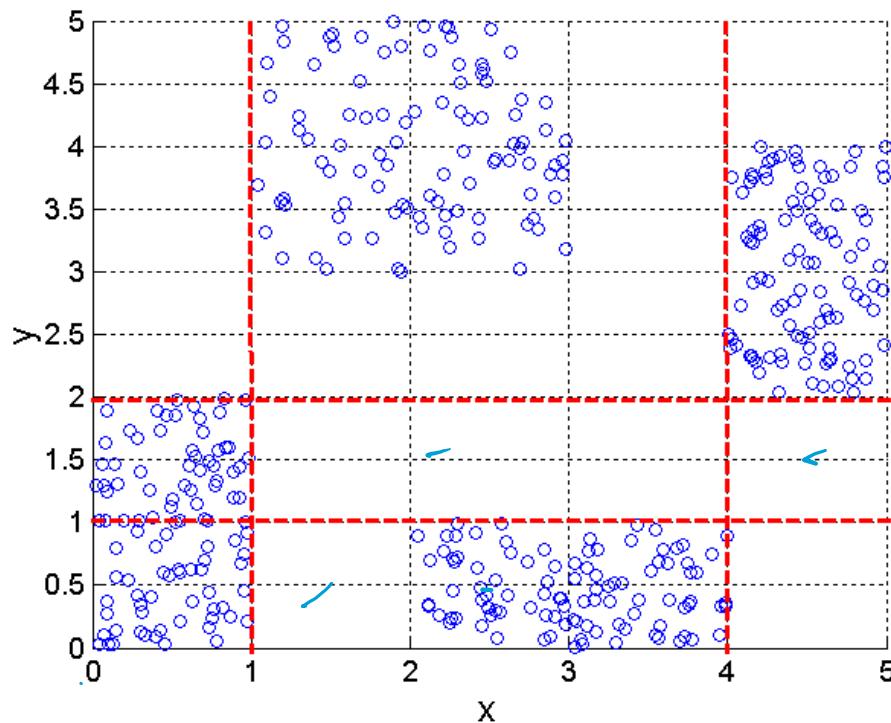
After bin Boundary: Bin 2: 21, 21, 26, 26,

Before bin Boundary: Bin 3: 27, 30, 30, 34

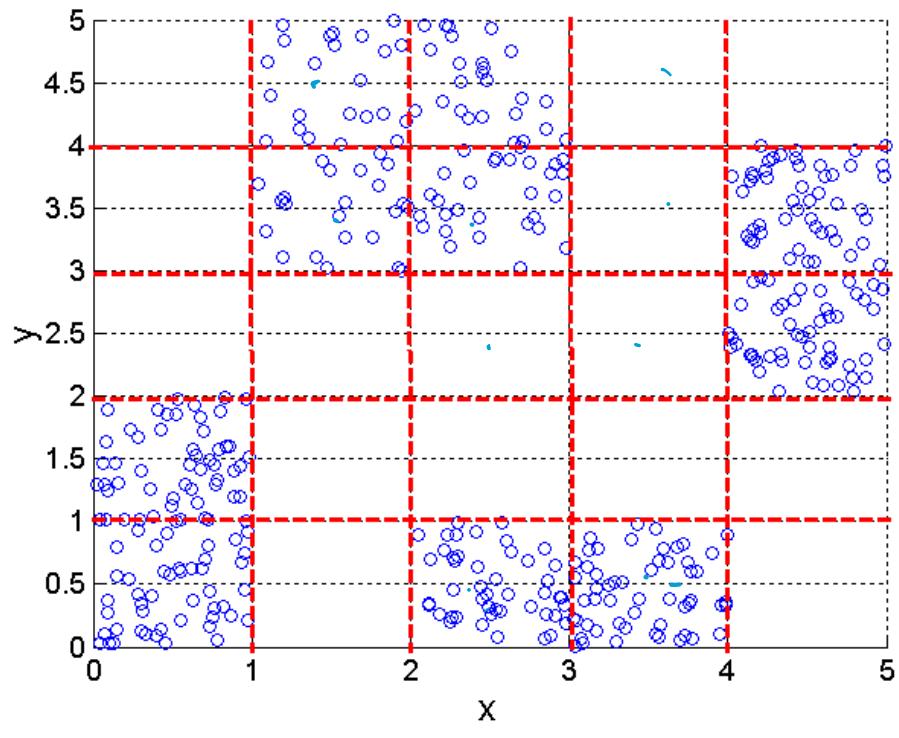
After bin Boundary: Bin 3: 27, 27, 27, 34

Discretization Using Class Labels

□ Entropy based approach

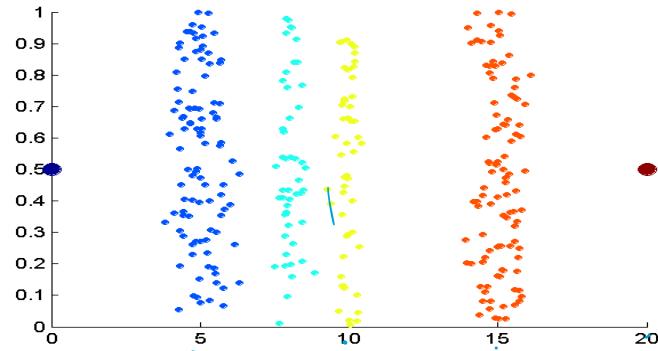


3 categories for both x and y

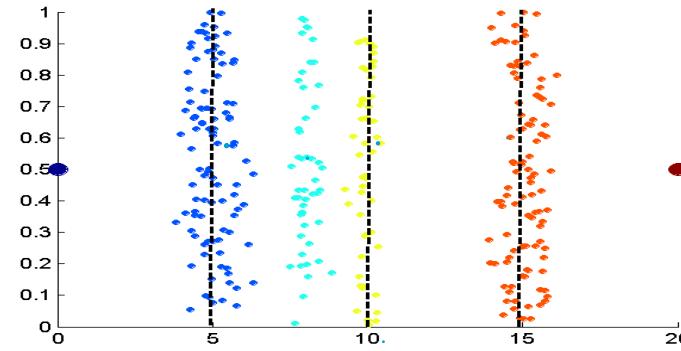


5 categories for both x and y

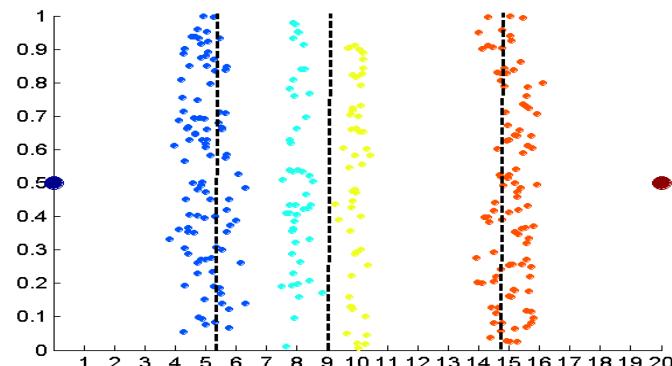
Discretization Without Using Class Labels



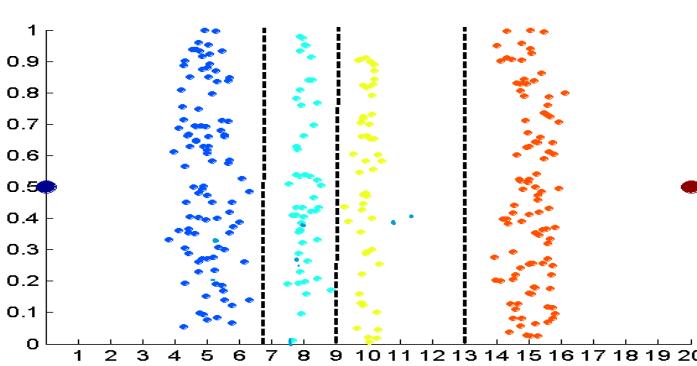
Data



Equal interval width



Equal frequency



K-means

Advantages (Pros) of data smoothing

Data smoothing clears the understandability of different important hidden patterns in the data set.

Data smoothing can be used to help predict trends. Prediction is very helpful for getting the right decisions at the right time.

Data smoothing helps in getting accurate results from the data.

Cons of data smoothing

Data smoothing doesn't always provide a clear explanation of the patterns among the data.

It is possible that certain data points being ignored by focusing the other data points.

Discretization by Classification & Correlation Analysis

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

Normalization

Normalization: used to scale the data of an attribute in range -1.0 to 1.0 or 0.0 to 1.0. It is generally useful for classification algorithms.

Need of Normalization –

Required when we are dealing with attributes on a different scale

It may lead to a dilution in effectiveness of an important equally important attribute (on lower scale) because of other attribute having values on larger scale.

Normalized to bring all the attributes on the same scale.

| person_name | Salary | Year_of_experience | Expected Position Level |
|-------------|--------|--------------------|-------------------------|
| Aman | 100000 | 10 | 2 |
| Abhinav | 78000 | 7 | 4 |
| Ashutosh | 32000 | 5 | 8 |
| Dishi | 55000 | 6 | 7 |
| Abhishek | 92000 | 8 | 3 |
| Avantika | 120000 | 15 | 1 |
| Ayushi | 65750 | 7 | 5 |

The attributes salary and year_of_experience are on different scale and hence attribute salary can take high priority over attribute year_of_experience in the model.

Min-Max Normalization

Normalization means transforming or mapping the data to a smaller or common range. All attributes gain an equal weight after this process.

- **Min-max normalization:** This preserves the relationships among the original data values and performs a linear transformation on the original data. The applicable ones of the actual maximum and minimum values of an attribute will be normalized in 0 to 1.

$$v' = \frac{v - \min(A)}{\max(A) - \min(A)} (\text{new_max}(A) - \text{new_min}(A)) + \text{new_min}(A)$$

- Where, A is the attribute data,
Min(A), Max(A) are the minimum and maximum absolute value of A respectively.
 v' is the new value of each entry in data.
 v is the old value of each entry in data.
 $\text{new_max}(A)$, $\text{new_min}(A)$ is the max and min value of the range(i.e boundary value of range required) respectively.

z-score normalization (Zero-Mean)

- **z-score normalization (Zero-Mean)**: Here the values for an attribute are normalized based on the mean and standard deviation of that attribute.
- It is useful when the actual minimum and maximum of an attribute to be normalized are unknown.

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

- v' and v is the new and old of each entry in data respectively. σ_A , and \bar{A} is the standard deviation and mean of A respectively.
- Takes care of outliers but does not provide data normalization with an identical scale.

Comparison of Min-Max Normalization and Z-Score Normalization

| Min-max normalization | Z-score normalization |
|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Not very well efficient in handling the outliers | Handles the outliers in a good way. |
| Min-max Guarantees that all the features will have the exact same scale. | Helpful in the normalization of the data but not with the <i>exact same scale</i>. |

Normalization by decimal scaling

Normalization by decimal scaling: This normalizes by moving the decimal point of values of attribute.

It normalizes by moving the decimal point of values of the data. To normalize the data by this technique, we divide each value of the data by the maximum absolute value of data.

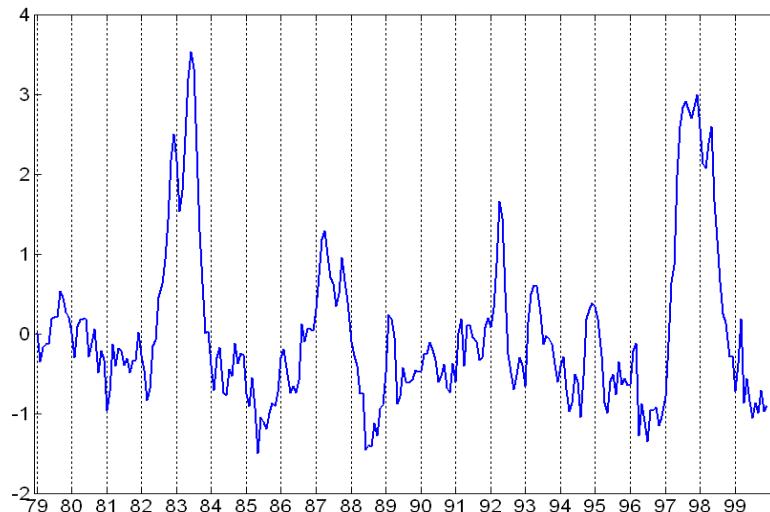
Where, j is the smallest integer such that $\max(|v_i|) < 1$.

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

- Ex: input data is: -10, 201, 301, -401, 501, 601, 701
- To normalize the above data,
 - Step 1: Maximum absolute value in given data(m): 701
 - Step 2: Divide the given data by 1000 (i.e $j=3$ =no of digits comprising the number)
- **Result:** The normalized data is: -0.01, 0.201, 0.301, -0.401, 0.501, 0.601, 0.701
- It follows that the means of the normalized data will always be between 0 and 1.

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

□ Proximity refers to a similarity or dissimilarity

Similarity/Dissimilarity for Simple Attributes

p and q are the attribute values for two data objects.

| 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 | $d = \frac{ p-q }{n-1}$ (values mapped to integers 0 to $n-1$, where n is the number of values) | $s = 1 - \frac{ p-q }{n-1}$ |
| Interval or Ratio | $d = p - q $ | $s = -d, s = \frac{1}{1+d}$ or $s = 1 - \frac{d - \min_d}{\max_d - \min_d}$ |

Table 5.1. Similarity and dissimilarity for simple attributes

Distance Measures

- Remember K-Nearest Neighbor are determined on the bases of some kind of “distance” between points.
- Two major classes of distance measure:
 1. *Euclidean* : based on position of points in some k -dimensional space.
 2. *Noneuclidean* : not related to position or space.

Distance Measures

For a pair of vectors (data points, or objects, or rows of a table), we can use some distance measures to compute how different or similar the vectors are.

Euclidean distance:

The distance between points

$A(x_1, y_1)$ and $B(x_2, y_2)$ is equal to

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$A(f_1, f_2, f_3, f_4, \dots, f_n) \quad B(g_1, g_2, g_3, g_4, \dots, g_n)$$

$$\sqrt{(f_1 - g_1)^2 + (f_2 - g_2)^2 + (f_3 - g_3)^2 + (f_4 - g_4)^2 + \dots + (f_n - g_n)^2}$$

$$\sqrt{\sum_{i=1}^n (f_i - g_i)^2}$$

Therefore, the distance between Row 2 and Row 5
is equal to

$$\sqrt{(5 - 9)^2 + (4 - 2)^2} = 4.472135954999579$$

$$\text{L}_2 \text{ Norm} = \|A - B\| = \sqrt{\sum_{i=1}^n (f_i - g_i)^2}$$

| | Feature 1 | Feature 2 |
|-------|-----------|-----------|
| Row 1 | 10 | 3 |
| Row 2 | 5 | 4 |
| Row 3 | 10 | 4 |
| Row 4 | 8 | 6 |
| Row 5 | 9 | 2 |

Manhattan distance

A simpler difference in every dimension can be computed without squaring the differences and without using the square root.

That is, just sum up the differences between the two vectors in every dimension.

The measure that just sums up the differences in each dimension of two points or vectors

$$A(f_1, f_2, f_3, f_4, \dots, f_n) \quad \text{and} \quad B(g_1, g_2, g_3, g_4, \dots, g_n)$$

Manhattan Distance between A and B =

$$|A - B|_1 = \sum_{i=1}^n |f_i - g_i|$$

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$\text{dist} = \left(\sum_{i=1}^n |f_i - g_i|^r \right)^{\frac{1}{r}}$$

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
 - A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors
- $r = 2$. Euclidean distance
- $r \rightarrow \infty$. “supremum” (L_{\max} norm, L_∞ norm) distance.
 - This is the maximum difference between any component of the vectors
- Do not confuse r with n , i.e., all these distances are defined for all numbers of dimensions.

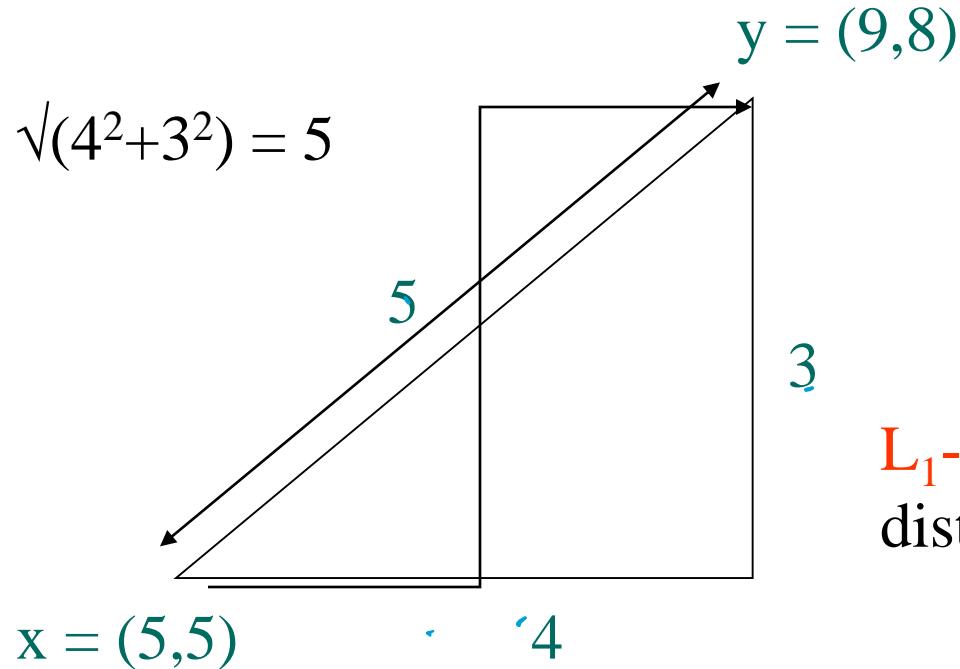
Another Euclidean Distance

- *L_∞ norm* : $d(x,y) = \text{the maximum of the differences between } x \text{ and } y \text{ in any dimension.}$

Examples L₁ and L₂ norms

L₂-norm:

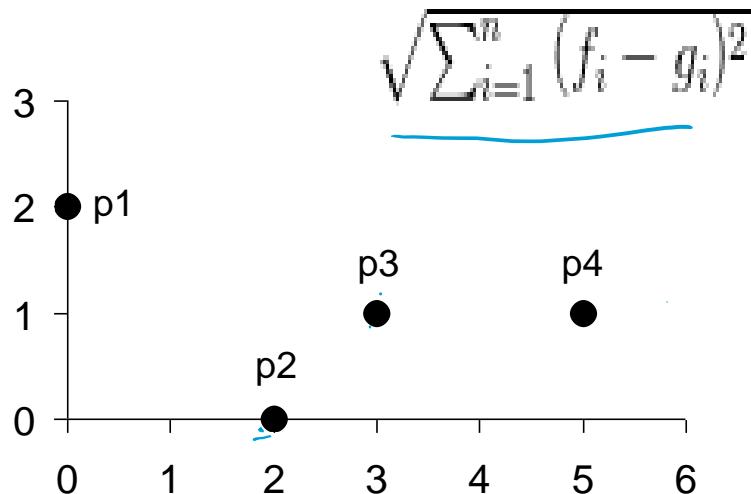
$$\text{dist}(x,y) = \sqrt{(4^2+3^2)} = 5$$



L₁-norm:

$$\text{dist}(x,y) = 4+3 = 7$$

Euclidean Distance



| 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 |

Distance Matrix

Minkowski Distance

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

| L1 | p1 | p2 | p3 | p4 |
|----|----|----|----|----|
| p1 | 0 | 4 | 4 | 6 |
| p2 | 4 | 0 | 2 | 4 |
| p3 | 4 | 2 | 0 | 2 |
| p4 | 6 | 4 | 2 | 0 |

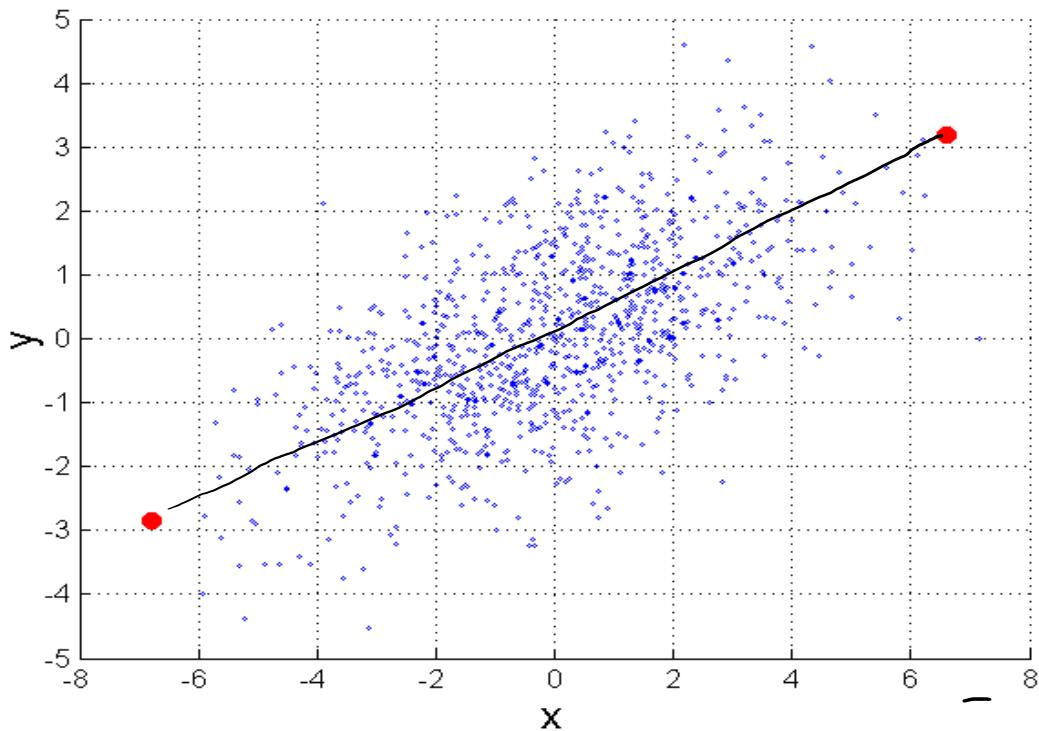
| L2 | p1 | p2 | p3 | p4 |
|----|-------|-------|-------|-------|
| p1 | 0 | 2.828 | 3.162 | 5.099 |
| p2 | 2.828 | 0 | 1.414 | 3.162 |
| p3 | 3.162 | 1.414 | 0 | 2 |
| p4 | 5.099 | 3.162 | 2 | 0 |

| L ∞ | p1 | p2 | p3 | p4 |
|------------|----|----|----|----|
| p1 | 0 | 2 | 3 | 5 |
| p2 | 2 | 0 | 1 | 3 |
| p3 | 3 | 1 | 0 | 2 |
| p4 | 5 | 3 | 2 | 0 |

Distance Matrix

Mahalanobis Distance

$$mahalanobis(p, q) = (p - q)^T \Sigma^{-1} (p - q)$$



Σ is the covariance matrix of the input data X

$$\Sigma_{j,k} = \frac{1}{n-1} \sum_{i=1}^n (X_{ij} - \bar{X}_j)(X_{ik} - \bar{X}_k)$$

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.

Hamming Distance

Hamming distance between two strings (same length) is the number of positions where the strings have different letters.

Example: compute the distance between the following two strings.

apricot
Abrikop

The two strings are different in three positions : 2, 5, and 7.

Hamming distance (apricot, abrikop) = 3

Used to compute the distance between binary bit vectors of the same length.

Sometimes, binary streams are repeated during transmission to increase reliability.

Hamming distance is a true distance measure and satisfies the four distance properties.

Edit distance (LCS-based)

The edit distance between two strings is the minimum number of edits (delete or insert) required to convert one string to the other.

Please note that given an infinite number of edits it is always possible to convert one string to another. We are talking about the “minimum” number of edits.

Example: consider that we have two strings S_1 and S_2 .

$$S_1 = xyzmn \quad S_2 = yzmopn$$

Let us try to convert S_1 to S_2 .

1. Delete x from S_1 . S_1 is now yzmn.
2. Insert o in the fourth position. S_1 now becomes yzmon.
3. Insert p in the fifth position. S_1 now becomes yzmopn.
4. Notice that S_1 has become $S_2 = yzmopn$.

To convert S_1 to S_2 , we needed three edits (one delete and two insertions.) We cannot do this conversion with any lesser edits than 3.

Therefore, the edit distance between xyzmn and yzmopn is = 3.

edit distance (S_1, S_2) = edit distance (S_2, S_1)

The alternative way to compute the edit distance: based on the length of the longest common subsequence (LCS)

$$\text{edit distance } (S_1, S_2) = |S_1| + |S_2| - 2 * |\text{LCS } (S_1, S_2)|$$

$|S_1|$ or $|S_2|$ indicates the length of the corresponding string.

$|\text{LCS } (S_1, S_2)|$ is the length of the longest common subsequence between both strings S_1 and S_2 from left to right

Example: consider strings $S_1 = \text{xyzmn}$ and $S_2 = \text{yzmopn}$.

yzm is present in both the strings from left to right.

However, the longest sequence that is present in both the strings from left to right is yzmn .

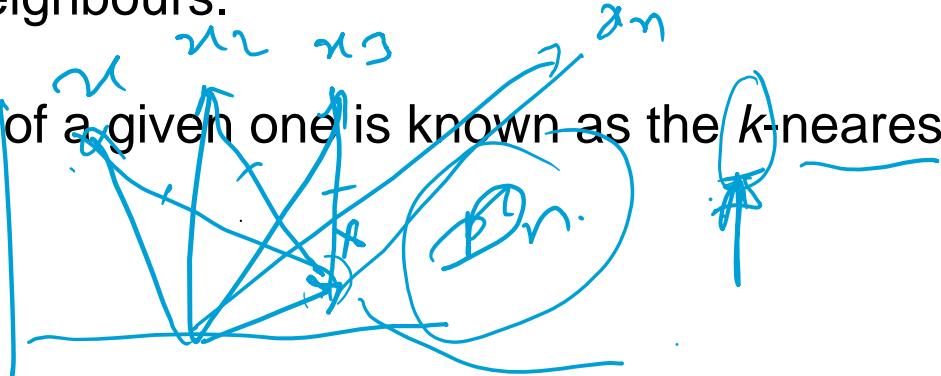
$$\begin{aligned}\text{Edit distance}(S_1, S_2) &= \text{edit distance}(\text{xyzmn}, \text{yzmopn}) \\ &= |\text{xyzmn}| + |\text{yzmopn}| - 2 |\text{yzmn}| = 5+6-2*4 = 3\end{aligned}$$

Nearest Neighbours

Given a data point, finding several closest points is called the computation of the nearest neighbours.

Finding k nearest data points of a given one is known as the k -nearest neighbours (knn) problem.

Problem statement for knn :



Given a vector \mathbf{x} and a data set D , order all N vectors of D such that

$$D = \{x_1, x_2, x_3, \dots, x_n\} \quad \text{distance}(x, x_i) \leq \text{distance}(x, x_{i+1})$$

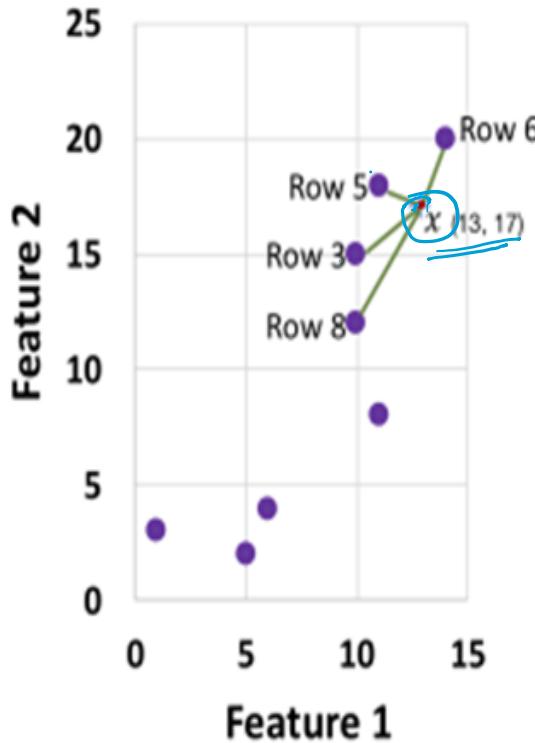
Return the first K vectors

$$S = \{x_1, x_2, x_3, \dots, x_k\}$$

For numerical objects, the length of S must be equal to the number of features/dimensions in D to be able to use a distance or similarity measure. knn returns the indices (row serial number) of the top k -nearest neighbours, instead of returning k complete vectors from the data.

Example of *knn*

| | Feature 1 | Feature 2 |
|-------|-----------|-----------|
| Row 1 | 5 | 2 |
| Row 2 | 1 | 3 |
| Row 3 | 10 | 15 |
| Row 4 | 6 | 4 |
| Row 5 | 11 | 18 |
| Row 6 | 14 | 20 |
| Row 7 | 11 | 8 |
| Row 8 | 10 | 12 |



With $k=4$ and a given point x , a *knn* function will return the following row IDs.

Row 5

Row 6

Row 3

Row 8

Row 5 is the 1st nearest neighbor.
Row 6 is the 2nd nearest neighbor.
Row 3 is the 3rd nearest neighbor.
Row 8 is the 4th nearest neighbor.

Find the 4 nearest neighbours of the point $x=(13,17)$

Find the 4 nearest neighbours of the point $x=(13,17)$

COMPUTE THE DISTANCE BETWEEN THE GIVEN POINT $x = (13, 17)$ AND EACH OF THE DATA POINTS IN THE DATA TABLE.

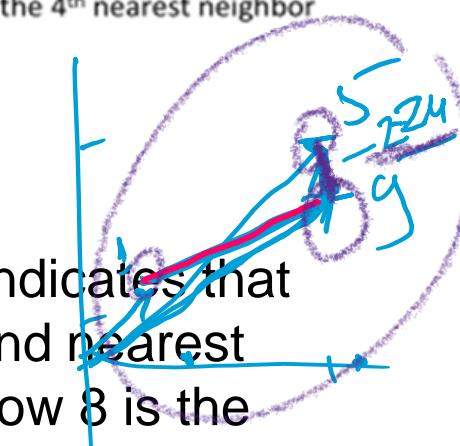
| | Feature 1 | Feature 2 |
|-------|-----------|-----------|
| Row 1 | 5 | 2 |
| Row 2 | 1 | 3 |
| Row 3 | 10 | 15 |
| Row 4 | 6 | 4 |
| Row 5 | 11 | 18 |
| Row 6 | 14 | 20 |
| Row 7 | 11 | 8 |
| Row 8 | 10 | 12 |

| Euclidean distance between x and row i |
|-------------------------------------------|
| $\sqrt{(13 - 5)^2 + (17 - 2)^2} = 17.00$ |
| $\sqrt{(13 - 1)^2 + (17 - 3)^2} = 18.44$ |
| $\sqrt{(13 - 10)^2 + (17 - 15)^2} = 3.61$ |
| $\sqrt{(13 - 6)^2 + (17 - 4)^2} = 14.76$ |
| $\sqrt{(13 - 11)^2 + (17 - 18)^2} = 2.24$ |
| $\sqrt{(13 - 14)^2 + (17 - 20)^2} = 3.16$ |
| $\sqrt{(13 - 11)^2 + (17 - 8)^2} = 9.22$ |
| $\sqrt{(13 - 10)^2 + (17 - 12)^2} = 5.83$ |

- Row 1 is the 7th nearest neighbor
Row 2 is the 8th nearest neighbor
Row 3 is the 3rd nearest neighbor
Row 4 is the 6th nearest neighbor
Row 5 is the 1st nearest neighbor
Row 6 is the 2nd nearest neighbor
Row 7 is the 5th nearest neighbor
Row 8 is the 4th nearest neighbor

First Compute Euclidean distance of x with each row.
For $K=4$ select first 4 from the ascending orders.

knn will return an array with content [5, 6, 3, 8], which indicates that Row 5 is the first nearest neighbour, Row 6 is the second nearest neighbour, Row 3 is the third nearest neighbour, and Row 8 is the fourth nearest neighbour.



Back to k-Nearest Neighbor (Pseudo-code)

- Missing values Imputation using k-NN.
- Input: Dataset (D), size of K
- for each record (x) with at least one missing value in D .
 - for each data object (y) in D .
 - ◆ Take the Distance (x,y)
 - ◆ Save the distance and y in array Similarity (S) array.
 - Sort the array S in descending order
 - Pick the top K data objects from S
 - ◆ Impute the missing attribute value (s) of x on the basis of known values of S (use Mean/Median or MOD).

K-Nearest Neighbor Drawbacks

- The major drawbacks of this approach are the
 - Choice of selecting exact distance functions.
 - Considering all attributes when attempting to retrieve the similar type of examples.
 - Searching through all the dataset for finding the same type of instances.
 - Algorithm Cost: ?

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, can not be negative)
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) The distance from one point to another cannot be greater than the distance between the same two points via another point.
This is commonly known as the triangle inequality property - the length of one side of a triangle cannot be greater than the sum of the other two sides

where $d(p, q)$ is the distance (dissimilarity) between points (data objects), p and q .

- A distance that satisfies these properties is a **metric**

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

$$= (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$$

$J = \text{number of 11 matches} / \text{number of not-both-zero attributes values}$

$$= (M_{11}) / (M_{01} + M_{10} + M_{11})$$

- The high similarity between a pair of points indicates that the points are nearby. Low similarity indicates a large distance.

$$\begin{array}{r} \text{antities} \\ P = 10101010 \\ Q = 10111100 \end{array}$$

SMC versus Jaccard: Example

$p = 1000000000$

$q = 0000001001$

$M_{01} = 2$ (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)

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

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

Jaccard index or Jaccard coefficient

- Jaccard index/coefficient/similarity is generally computed between two sets of items.
- It is a ratio of commonality between the sets over all the items.
- If X and Y are two sets, then the Jaccard index between two sets is computed using the ratio of the size of the intersection and the size of the union of the two sets.

$$\text{Jaccard}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

If $X = \{a, b, c\}$, and $Y = \{b, c, d, e\}$ then, the size of the intersection between X and Y is:

$$|X \cap Y| = |\{b, c\}| = 2 \quad |X \cup Y| = |\{a, b, c, d, e\}| = 5$$

$$\text{Jaccard}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{2}{5} = 0.4$$

Jaccard similarity varies between 0 (No similarity) to 1 (Similar).

Weighted Jaccard index/coefficient/similarity

- Jaccard index computed between two vectors/data points/objects is called a weighted Jaccard index.
- Given X and Y — two vectors each of length n — the formula for weighted Jaccard index or similarity between them is:

$$\text{Jaccard}(X, Y) = \frac{\sum_{k=1}^n \min(X_k, Y_k)}{\sum_{k=1}^n \max(X_k, Y_k)}$$

| | Feature 1 | Feature 2 | Feature 3 | Feature 4 |
|-------|-----------|-----------|-----------|-----------|
| Row 1 | 10 | 3 | 3 | 5 |
| Row 2 | 5 | 4 | 5 | 3 |
| Row 3 | 9 | 4 | 6 | 4 |
| Row 4 | 8 | 6 | 2 | 6 |
| Row 5 | 20 | 15 | 10 | 20 |

$$\text{Jaccard}(\text{Row 1}, \text{Row 3}) = \frac{9+3+3+4}{10+4+6+5} = \frac{19}{25} = 0.76 \quad < 1$$

$$\text{Jaccard}(\text{Row 1}, \text{Row 5}) = \frac{10+3+3+5}{20+15+10+20} = \frac{21}{65} = 0.323076923$$

That means Row 3 is more like Row 1 than Row 5

The set-based Jaccard similarity discussed earlier is a special case of weighted Jaccard similarity — in the set-based Jaccard similarity, the weight of an item (feature) can be either 1 (present) or 0 (absent.)

Consider $X = \{a, b, c\}$.

| | a | b | c | d | e |
|---|---|---|---|---|---|
| X | 1 | 1 | 1 | 0 | 0 |
| Y | 0 | 1 | 1 | 1 | 0 |

$$\text{Jaccard coeff.}(X, Y) = \frac{0+1+1+0+0}{1+1+1+1+1} \\ = \frac{2}{5} = 0.4$$

Proximity Measure for Binary Attributes

- A contingency table for binary data

| | | Object <i>j</i> | | |
|-----------------|---|-----------------|------------|------------|
| | | 1 | 0 | sum |
| Object <i>i</i> | 1 | <i>q</i> | <i>r</i> | <i>q+r</i> |
| | 0 | <i>s</i> | <i>t</i> | <i>s+t</i> |
| sum | | <i>q+s</i> | <i>r+t</i> | <i>p</i> |

- Distance measure for symmetric binary variables:

$$d(i, j) = \frac{r + s}{q + r + s + t}$$

- Distance measure for asymmetric binary variables:

$$d(i, j) = \frac{r + s}{q + r + s}$$

- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

$$sim_{Jaccard}(i, j) = \frac{q}{q + r + s}$$

- **Note: Jaccard coefficient is the same as “coherence”:**

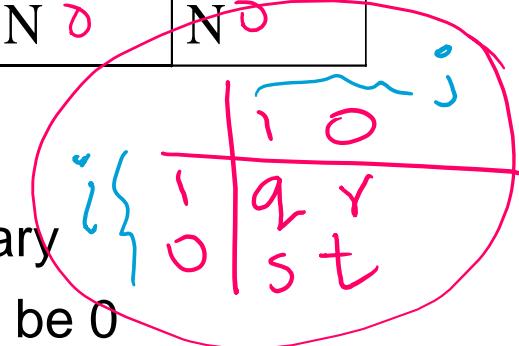
$$coherence(i, j) = \frac{sup(i, j)}{sup(i) + sup(j) - sup(i, j)} = \frac{q}{(q + r) + (q + s) - q}$$

Dissimilarity between Binary Variables

Example

| Name | Gender | Fever | Cough | Test-1 | Test-2 | Test-3 | Test-4 |
|------|--------|-------|-------|--------|--------|--------|--------|
| Jack | M | Y | N ○ | P | N ○ | N ○ | N ○ |
| Mary | F | Y | N ○ | P | N ○ | P | N ○ |
| Jim | M | Y ? | P | N ○ | N ⚡ | N ⚡ | N ○ |

- Gender is a symmetric attribute
- The remaining attributes are asymmetric binary
- Let the values Y and P be 1, and the value N be 0



$$d(i,j) = d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$

$$\frac{Y+S}{T+F+S}$$

$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$

$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

$q=11$
 $r=10$
 $s=01$
 $t=00$

Extended Jaccard Coefficient (Tanimoto)

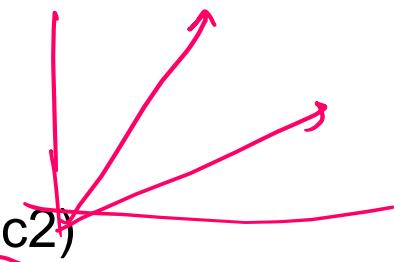
- Variation of Jaccard for continuous or count attributes
 - Reduces to Jaccard for binary attributes

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

Cosine Similarity

- Classical approach for computational linguistics is to measure similarity based on the content overlap between documents.
- For this we will represent documents as bag-of-words, so each document will be a sparse vector. And define measure of overlap as angle between vectors:

- Similarity (doc1, doc2) = $\cos(\theta) = \frac{\text{doc1} \cdot \text{doc2}}{\|\text{doc1}\| \|\text{doc2}\|}$



- By *cosine distance/dissimilarity* we assume following:
 - distance (doc1, doc2) = $1 - \text{similarity}(\text{doc1}, \text{doc2})$
- It is important to note, however, that this is not a proper distance metric in a mathematical sense as it does not have the triangle inequality property and it violates the coincidence axiom.

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 \bullet indicates vector dot product and $\|d\|$ is the length of vector d .

- Example:

$$d_1 = 3 \ 2 \ 0 \ 5 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0$$

$$d_2 = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2$$

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

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

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

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

Cosine Similarity

- A **document** can be represented by thousands of attributes, each recording the *frequency* of a particular word (such as **keywords**) or phrase in the document.

| Document | team | coach | hockey | baseball | soccer | penalty | score | win | loss | season |
|-----------|------|-------|--------|----------|--------|---------|-------|-----|------|--------|
| Document1 | 5 | 0 | 3 | 0 | 2 | 0 | 0 | 2 | 0 | 0 |
| Document2 | 3 | 0 | 2 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| Document3 | 0 | 7 | 0 | 2 | 1 | 0 | 0 | 3 | 0 | 0 |
| Document4 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 0 | 3 | 0 |

- Applications: information retrieval, biologic taxonomy, gene feature mapping, ...
- Cosine measure: If d_1 and d_2 are two vectors (e.g., term-frequency vectors), then

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

where \bullet indicates vector dot product, $\|d\|$: the length of vector d

Example: Cosine Similarity

- $\cos(d_1, d_2) = (d_1 \bullet d_2) / \|d_1\| \|d_2\|$,
where \bullet indicates vector dot product, $\|d\|$: the length of vector d
- Ex: Find the **similarity** between documents 1 and 2.

$$d_1 = (5, 0, 3, 0, 2, 0, 0, 2, 0, 0)$$

$$d_2 = (3, 0, 2, 0, 1, 1, 0, 1, 0, 1)$$

$$d_1 \bullet d_2 = 5*3+0*0+3*2+0*0+2*1+0*1+0*1+2*1+0*0+0*1 = 25$$

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

$$\|d_2\| = (3^2 + 0^2 + 2^2 + 0^2 + 1^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2)^{0.5} = (17)^{0.5} = 4.12$$

$$\cos(d_1, d_2) = \underline{0.94}$$

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:

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

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.

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

$$\text{distance}(p, q) = \left(\sum_{k=1}^n w_k |p_k - q_k|^r \right)^{1/r}.$$

Correlation

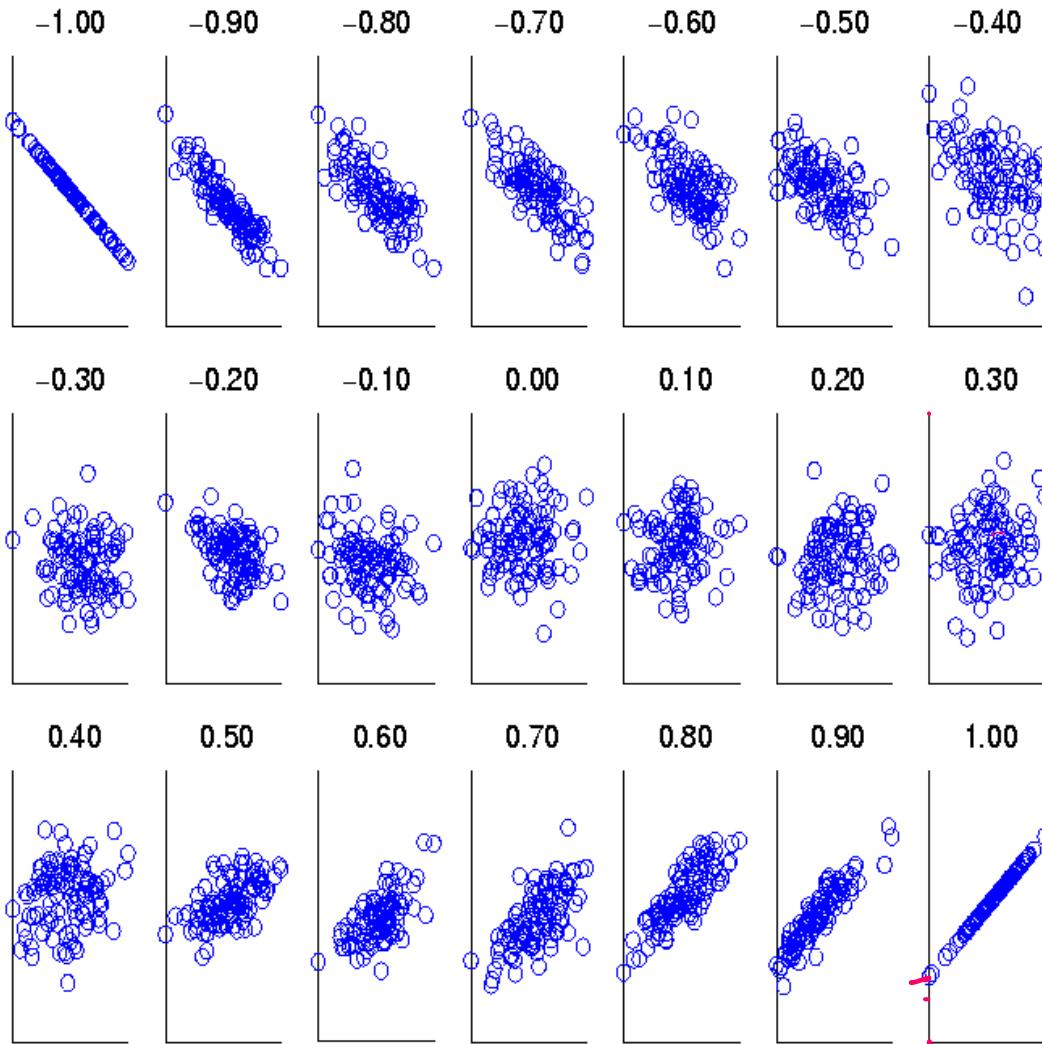
- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, p and q, and then take their dot product

$$p'_k = (p_k - \text{mean}(p)) / \text{std}(p)$$

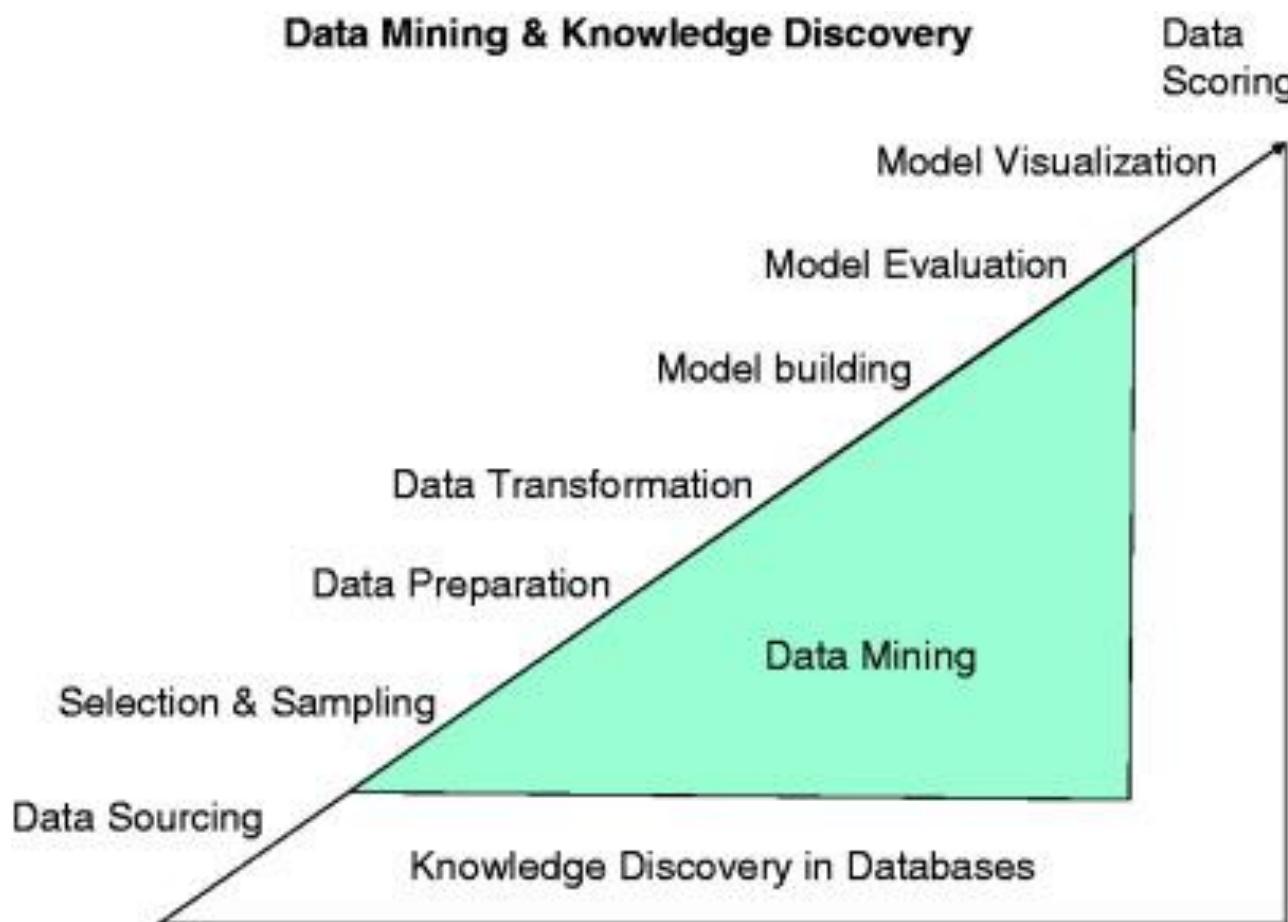
$$q'_k = (q_k - \text{mean}(q)) / \text{std}(q)$$

$$\text{correlation}(p, q) = p' \bullet q'$$

Visually Evaluating Correlation



**Scatter plots
showing the
similarity from
-1 to 1.**



Data Mining

3. 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
- Related to the area of Exploratory Data Analysis (EDA)
 - Created by statistician John Tukey
 - Seminal book is Exploratory Data Analysis by Tukey
 - A nice online introduction can be found in Chapter 1 of the NIST Engineering Statistics Handbook

<http://www.itl.nist.gov/div898/handbook/index.htm>

Techniques Used In Data Exploration

- In 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)

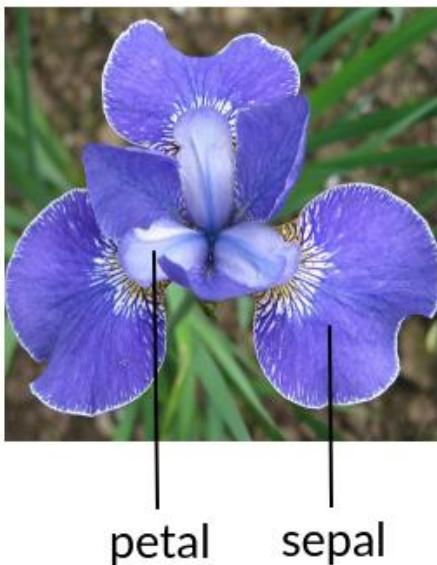
Iris Sample Data Set

- 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
 - Four (non-class) attributes
 - ◆ Sepal width and length
 - ◆ Petal width and length

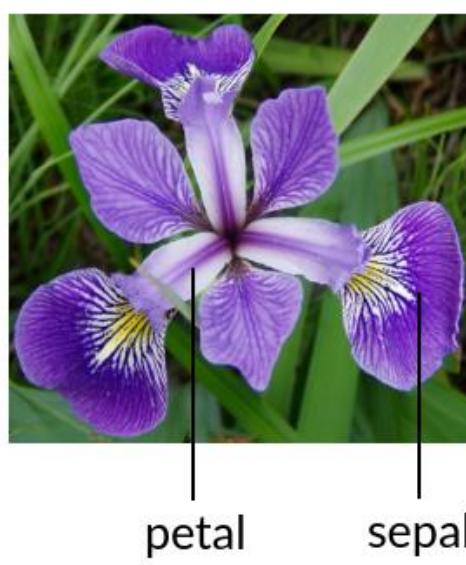


Virginica. Robert H. Mohlenbrock. USDA NRCS. 1995. Northeast wetland flora: Field office guide to plant species. Northeast National Technical Center, Chester, PA. Courtesy of USDA NRCS Wetland Science Institute.

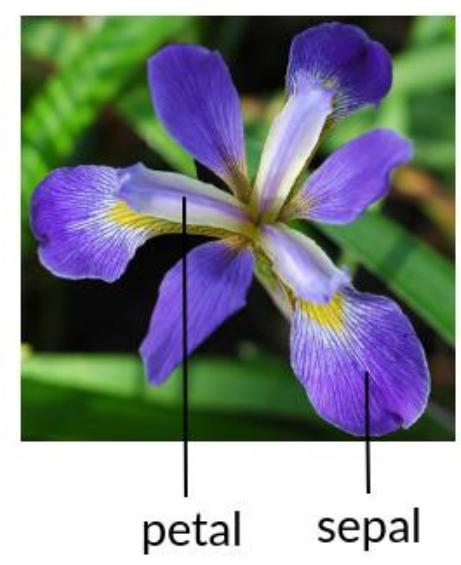
iris setosa



iris versicolor



iris 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 an attribute is the most frequent attribute value
- The notions of frequency and mode are typically used with categorical data

Percentiles

- For continuous data, the notion of a percentile is more useful.

Given an ordinal or continuous attribute x and a number p between 0 and 100, the p th percentile x_p 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 $x_{50\%}$.

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.

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

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

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.

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

However, this is also sensitive to outliers, so that other measures are often used.

Absolute Average Deviation (AAD),

Median Absolute Deviation (MAD),

Interquartile Range (IQ)

$$\text{AAD}(x) = \frac{1}{m} \sum_{i=1}^m |x_i - \bar{x}|$$

$$\text{MAD}(x) = \text{median}\left(\{|x_1 - \bar{x}|, \dots, |x_m - \bar{x}|\}\right)$$

$$\text{interquartile range}(x) = x_{75\%} - x_{25\%}$$

| Measure of central tendency $m(X)$ | Mean absolute deviation |
|------------------------------------|--------------------------------------------|
| Mean = 5 | $ 2-5 + 2-5 + 3-5 + 4-5 + 14-5 / 5 = 3.6$ |
| Median = 3 | $ 2-3 + 2-3 + 3-3 + 4-3 + 14-3 / 5 = 2.8$ |
| Mode = 2 | $ 2-2 + 2-2 + 3-2 + 4-2 + 14-2 / 5 = 3.0$ |

Seq.: $\{2 \ 2 \ 3 \ 4 \ 14\}$.

Basic Statistical Descriptions of Data

□ Motivation

- To better understand the data: central tendency, variation and spread

□ Data dispersion characteristics

- median, max, min, quantiles, outliers, variance, etc.

□ Numerical dimensions correspond to sorted intervals

- Data dispersion: analyzed with multiple granularities of precision
- Boxplot or quantile analysis on sorted intervals

□ Dispersion analysis on computed measures

- Folding measures into numerical dimensions
- Boxplot or quantile analysis on the transformed cube

Measuring the Central Tendency

- Mean (algebraic measure) (sample vs. population):

Note: n is sample size and N is population size.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \mu = \frac{\sum x}{N}$$

Weighted arithmetic mean:

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$$

Trimmed mean: chopping extreme values

Measuring the Central Tendency

□ Median:

- Middle value if odd number of values, or average of the middle two values otherwise
- Estimated by interpolation (for *grouped data*):

$$\text{Estimated Median} = L + \left(\frac{(n/2) - B}{G} \right) \times w$$

where:

- **L** is the lower class boundary of the group containing the median
- **n** is the total number of values 3194
- **B** is the cumulative frequency of the groups before the median group $200 + 450 + 300 = 950$
- **G** is the frequency of the median group 1500
- **w** is the group width $= 5$

□ Mode

$$mean - mode = 3 \times (mean - median)$$

- Value that occurs most frequently in the data
- Unimodal, bimodal, trimodal

| age | frequency |
|--------|-------------|
| 1–5 | 200 |
| 6–15 | 450 |
| 16–20 | 300 |
| 21–50 | 1500 |
| 51–80 | 700 |
| 81–110 | 44 |
| | <u>3194</u> |

Median # 1597

Given two school classes:

Morning class (20 students) = 62, 67, 71, 74, 76, 77, 78, 79, 79, 80, 80, 81, 81, 82, 83, 84, 86, 89, 93, 98 (Mean = 80)

Afternoon class (30 students) = 81, 82, 83, 84, 85, 86, 87, 87, 88, 88, 89, 89, 89, 89, 90, 90, 90, 90, 91, 91, 91, 92, 92, 93, 93, 94, 95, 96, 97, 98, 99 (Mean = 90)

The unweighted mean of the (80 + 90) / 2 = 85

Do not account for the difference in number of students in each class (20 versus 30);
Hence the value of 85 does not reflect the average student grade (independent of class).

The average student grade can be obtained by : adding all the grades up and divide by the total number of students:

$$\bar{x} = 4300 / 50 = 86$$

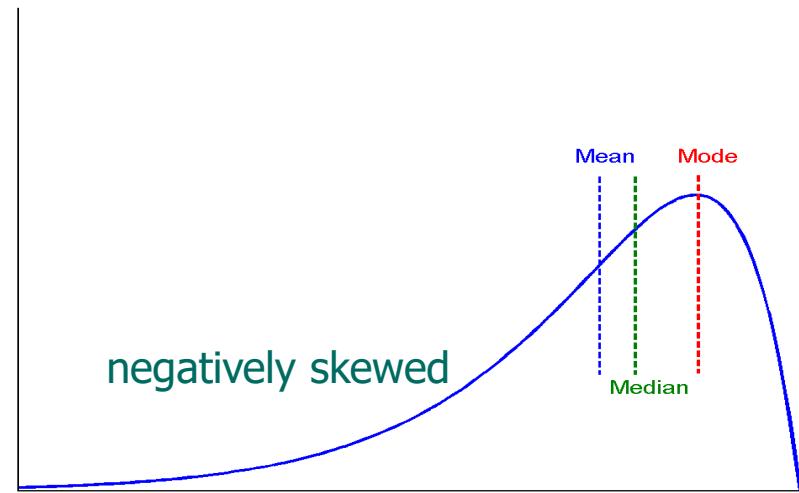
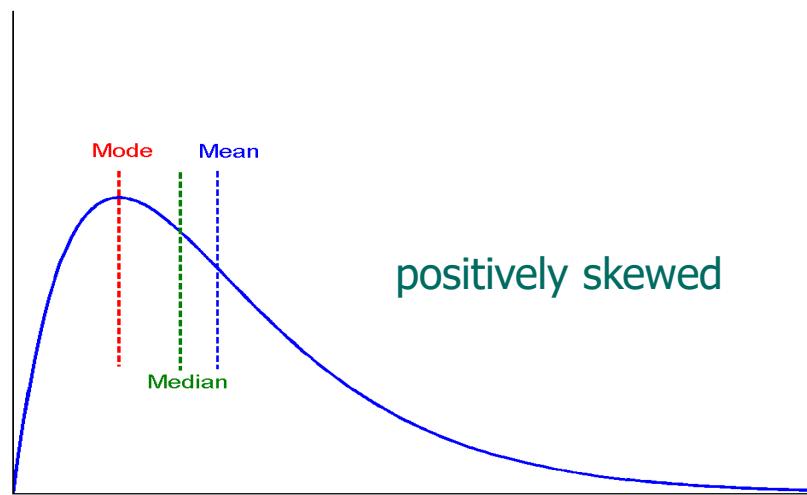
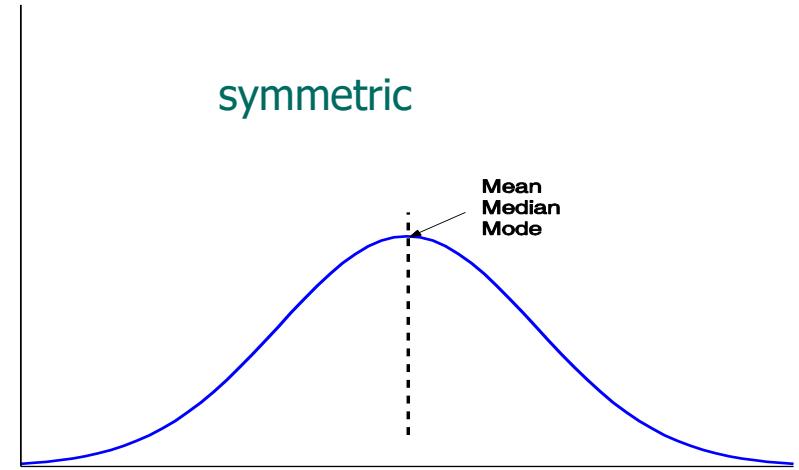
Or, this can be accomplished by weighting the class means by the number of students in each class. The larger class is given more "weight":

$$\bar{x} = \frac{(20 \times 80) + (30 \times 90)}{20 + 30} = 86$$

Thus, the weighted mean makes it possible to find the mean average student grade without knowing each student's score. Only the class means and the number of students in each class are needed.

Symmetric vs. Skewed Data

- Median, mean and mode of symmetric, positively and negatively skewed data



Measuring the Dispersion of Data

□ Quartiles, outliers and boxplots

- **Quartiles:** Q_1 (25^{th} percentile), Q_3 (75^{th} percentile)
- **Inter-quartile range:** $IQR = Q_3 - Q_1$
- **Five number summary:** min, Q_1 , median, Q_3 , max
- **Boxplot:** ends of the box are the quartiles; median is marked; add whiskers, and plot outliers individually
- **Outlier:** usually, a value higher/lower than $1.5 \times IQR$

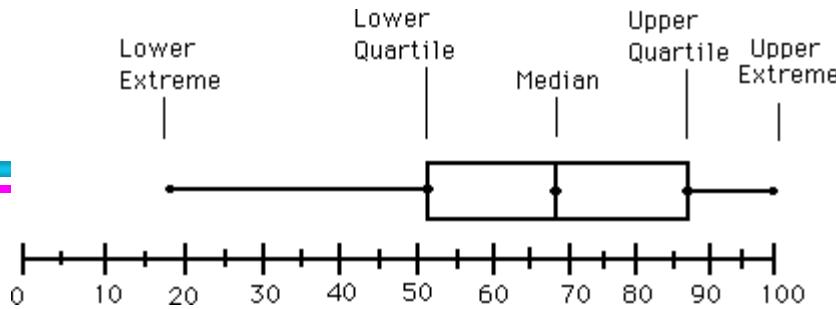
□ Variance and standard deviation (*sample: s, population: σ*)

- **Variance:** (algebraic, scalable computation)

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

- **Standard deviation s (or σ)** is the square root of variance s^2 (or σ^2)

Boxplot Analysis

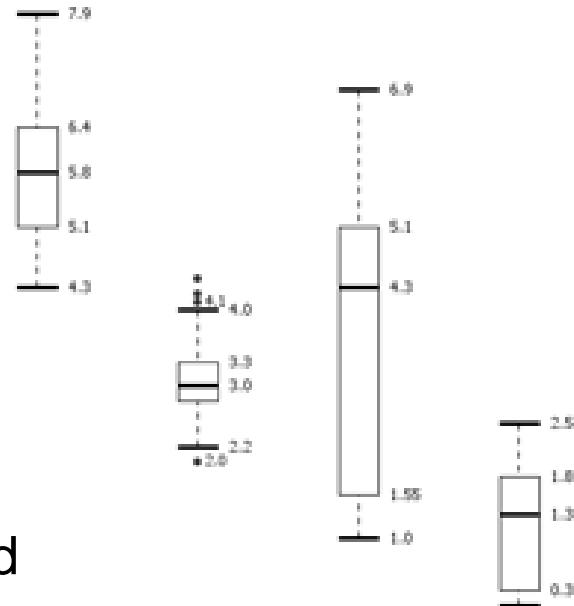


□ Five-number summary of a distribution

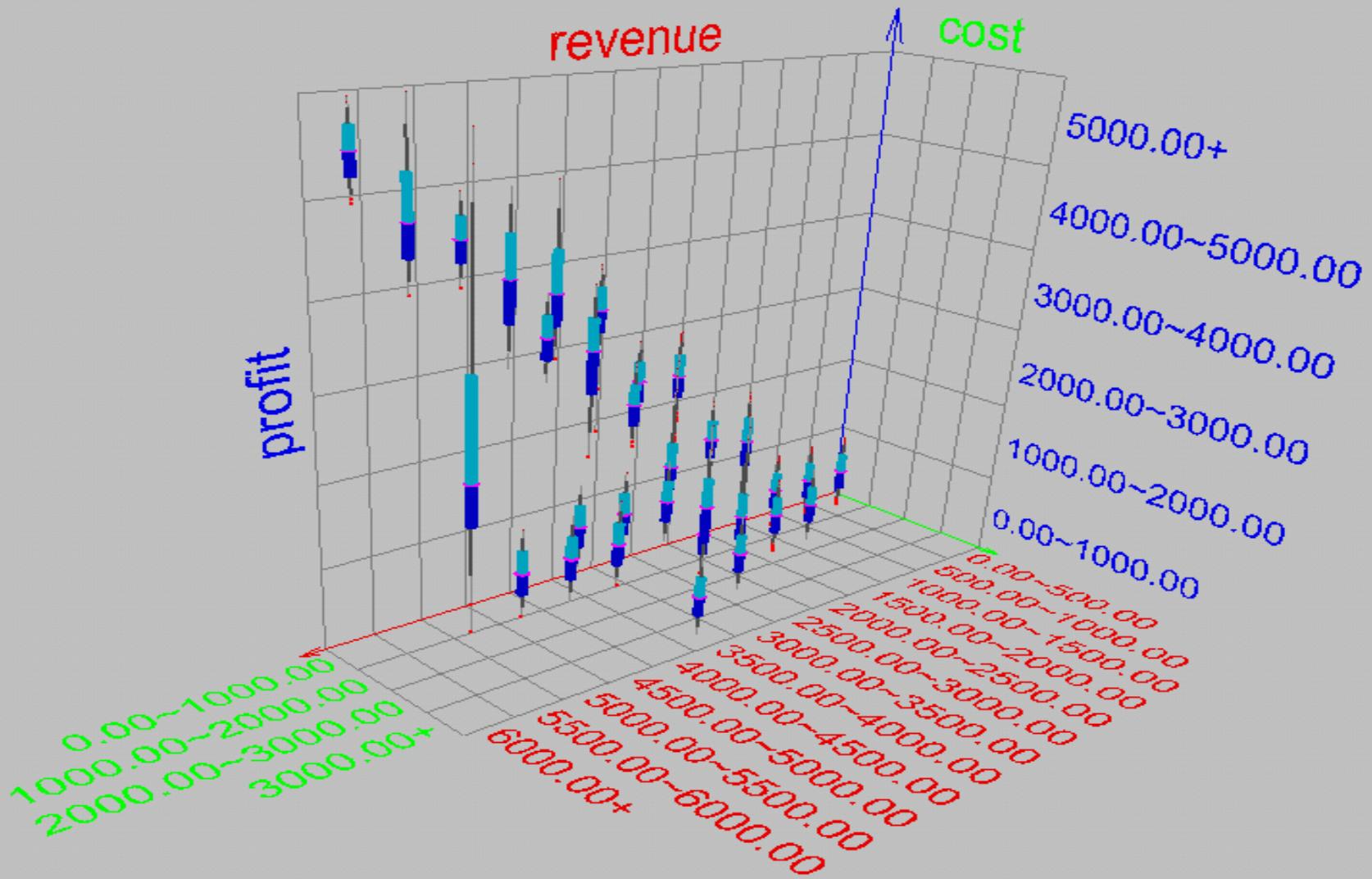
- Minimum, Q1, Median, Q3, Maximum

□ Boxplot

- Data is represented with a box
- The ends of the box are at the first and third quartiles, i.e., the height of the box is IQR
- The median is marked by a line within the box
- Whiskers: two lines outside the box extended to Minimum and Maximum
- Outliers: points beyond a specified outlier threshold, plotted individually



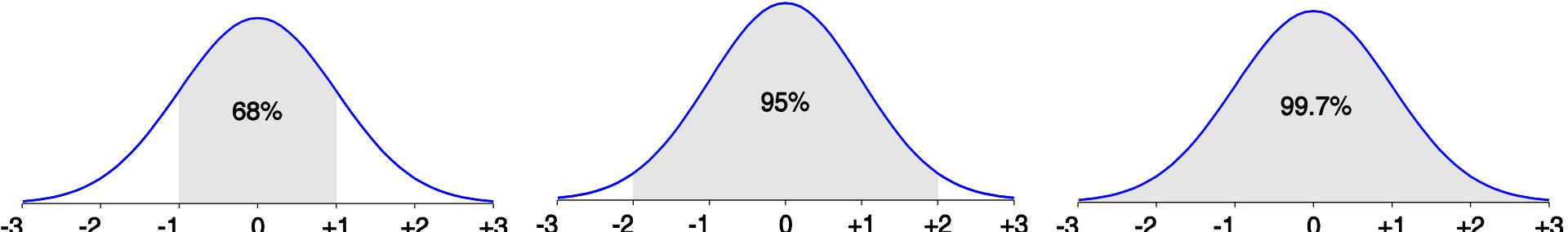
Visualization of Data Dispersion: 3-D Boxplots



Properties of Normal Distribution Curve

□ The normal (distribution) curve

- From $\mu-\sigma$ to $\mu+\sigma$: contains about 68% of the measurements (μ : mean, σ : standard deviation)
- From $\mu-2\sigma$ to $\mu+2\sigma$: contains about 95% of it
- From $\mu-3\sigma$ to $\mu+3\sigma$: contains about 99.7% of it



Graphic Displays of Basic Statistical Descriptions

- **Boxplot:** graphic display of five-number summary
- **Histogram:** x-axis are values, y-axis repres. frequencies
- **Quantile plot:** each value x_i is paired with f_i indicating that approximately $100 f_i \%$ of data are $\leq x_i$
- **Quantile-quantile (q-q) plot:** graphs the quantiles of one univariant distribution against the corresponding quantiles of another
- **Scatter plot:** each pair of values is a pair of coordinates and plotted as points in the plane

Visualization

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

Data Visualization

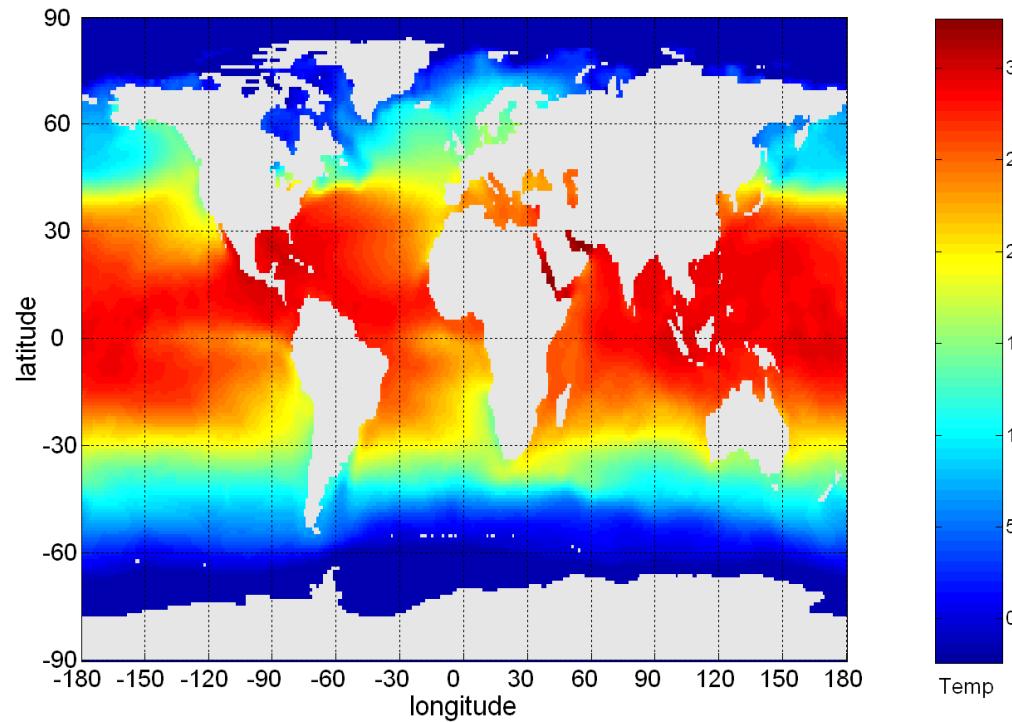
- Why data visualization?
 - Gain insight into an information space by mapping data onto graphical primitives
 - Provide qualitative overview of large data sets
 - Search for patterns, trends, structure, irregularities, relationships among data
 - Help find interesting regions and suitable parameters for further quantitative analysis
 - Provide a visual proof of computer representations derived
- Categorization of visualization methods:
 - Pixel-oriented visualization techniques
 - Geometric projection visualization techniques
 - Icon-based visualization techniques
 - Hierarchical visualization techniques
 - Visualizing complex data and relations

Geometric Projection Visualization Techniques

- Visualization of geometric transformations and projections of the data
- Methods
 - Direct visualization
 - Scatterplot and scatterplot matrices
 - Landscapes
 - Projection pursuit technique: Help users find meaningful projections of multidimensional data
 - Prosection views
 - Hyperslice
 - Parallel coordinates

Example: Sea Surface Temperature

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



Representation

- Is 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.**
- Example:
 - 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

- Is the placement of visual elements within a display
- Can make a large difference in how easy it is to understand the data
- Example: 9 Objects and 6 Attributes

| | 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 |

Re-arranged
Permuted

| | 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 |

3-1's &
3-0's .
--- --- ---
3-0's &
3-1's

Selection

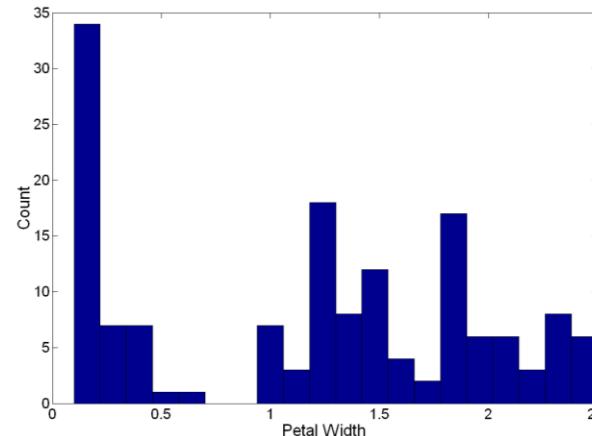
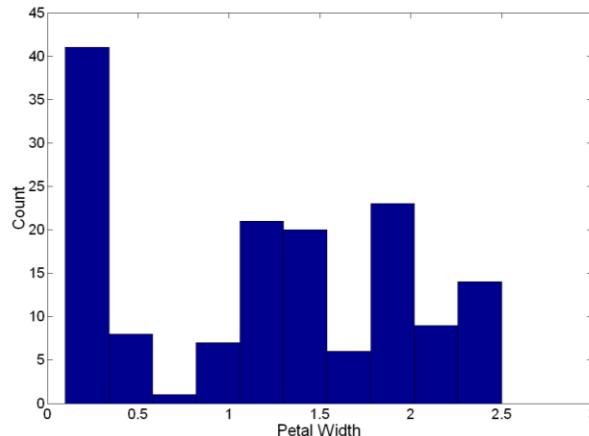
- Is the **elimination or the de-emphasis** of certain objects and attributes
- Selection may involve the **choosing a subset** of attributes
 - Dimensionality reduction is often used to reduce the number of dimensions to two or three
 - Alternatively, pairs of attributes can be considered
- Selection may also involve choosing a subset of objects
 - A region of the screen can only show so many points
 - Can sample, but want to preserve points in sparse areas

Visualization Techniques: Histograms

□ Histogram

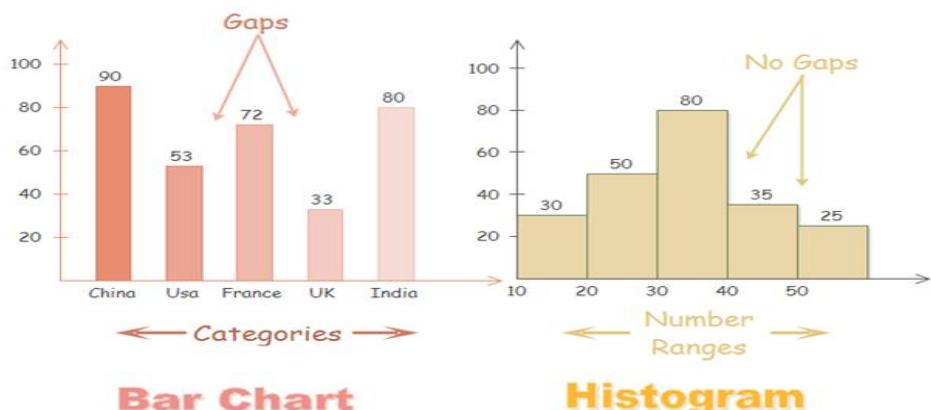
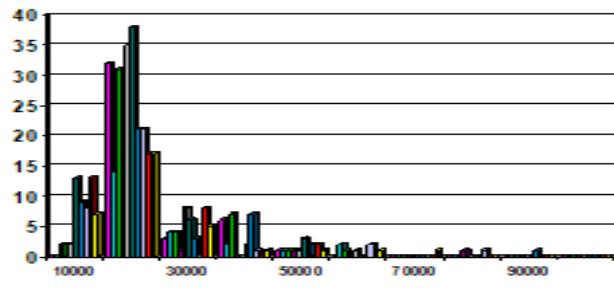
- Usually shows the **distribution of values** of a single variable
- Divide the values into bins and show a bar plot of the number of objects in each bin.
- The height of each bar indicates the number of objects
- Shape of histogram depends on the number of bins

□ Example: Petal Width (10 and 20 bins, respectively)

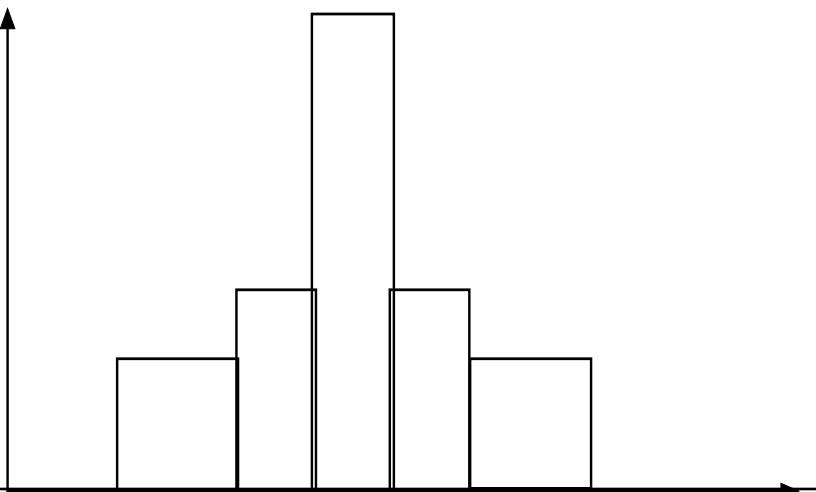
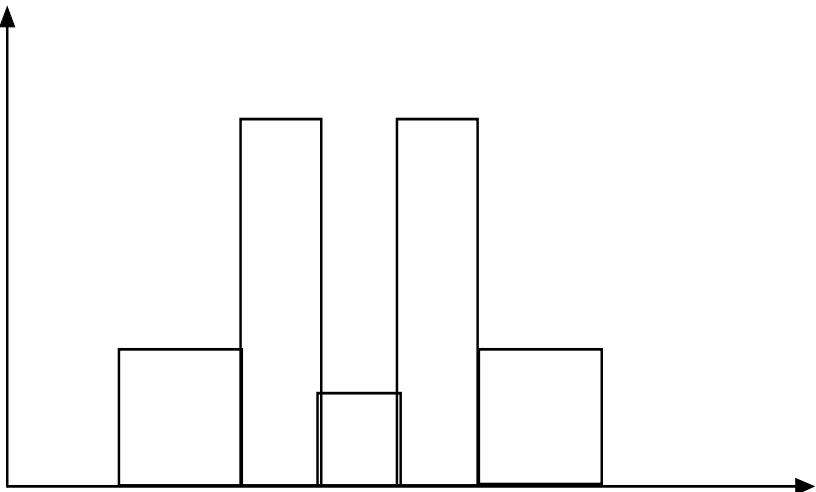


Histogram Analysis

- Histogram: Graph display of tabulated frequencies, shown as bars
- It shows what proportion of cases fall into each of several categories
- Differs from a bar chart in that it is the *area* of the bar that denotes the value, not the height as in bar charts, a crucial distinction when the categories are not of uniform width
- The categories are usually specified as non-overlapping intervals of some variable. The categories (bars) must be adjacent



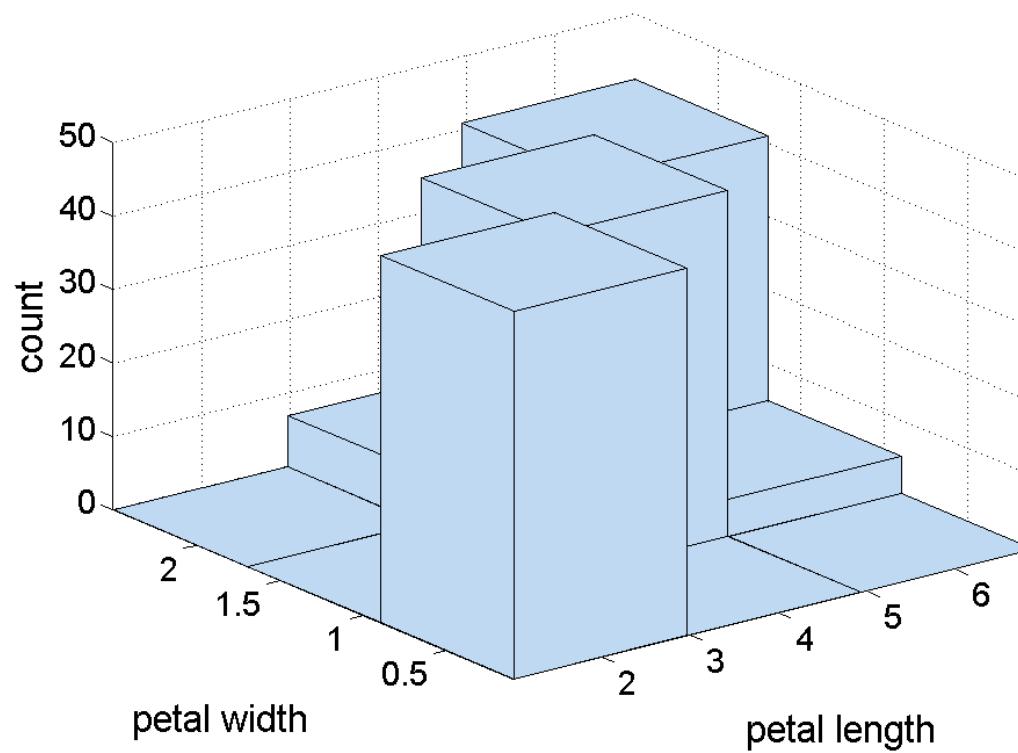
Histograms Often Tell More than Boxplots



- The two histograms shown in the left may have the same boxplot representation
 - The same values for: min, Q1, median, Q3, max
- But they have rather different data distributions

Two-Dimensional Histograms

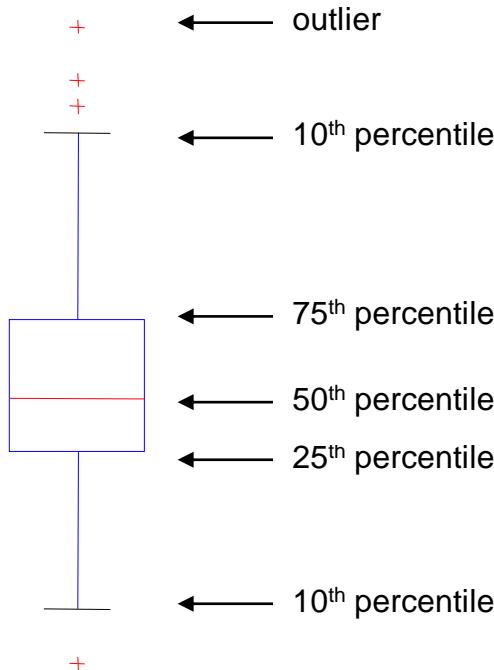
- Show the joint distribution of the values of two attributes
- Example: petal width and petal length
 - What does this tell us?



Visualization Techniques: Box Plots

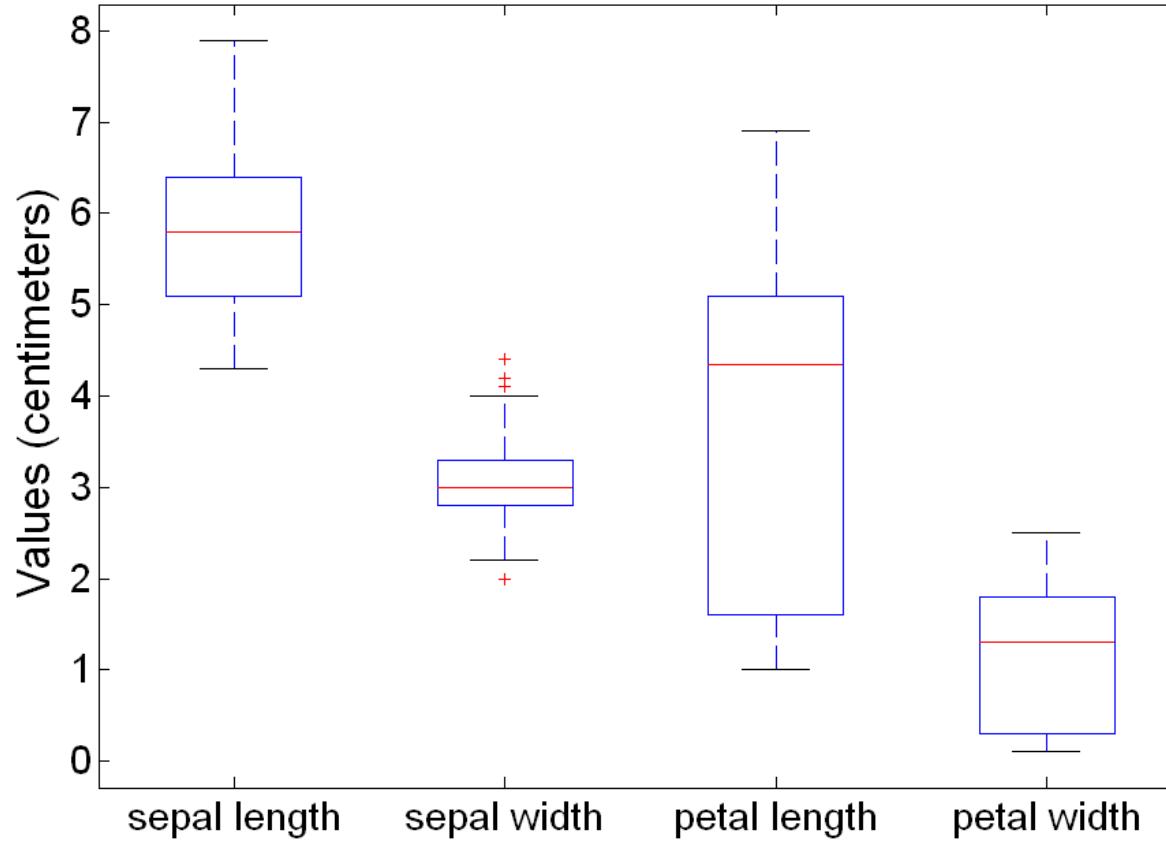
Box Plots

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



Example of Box Plots

- Box plots can be used to compare attributes

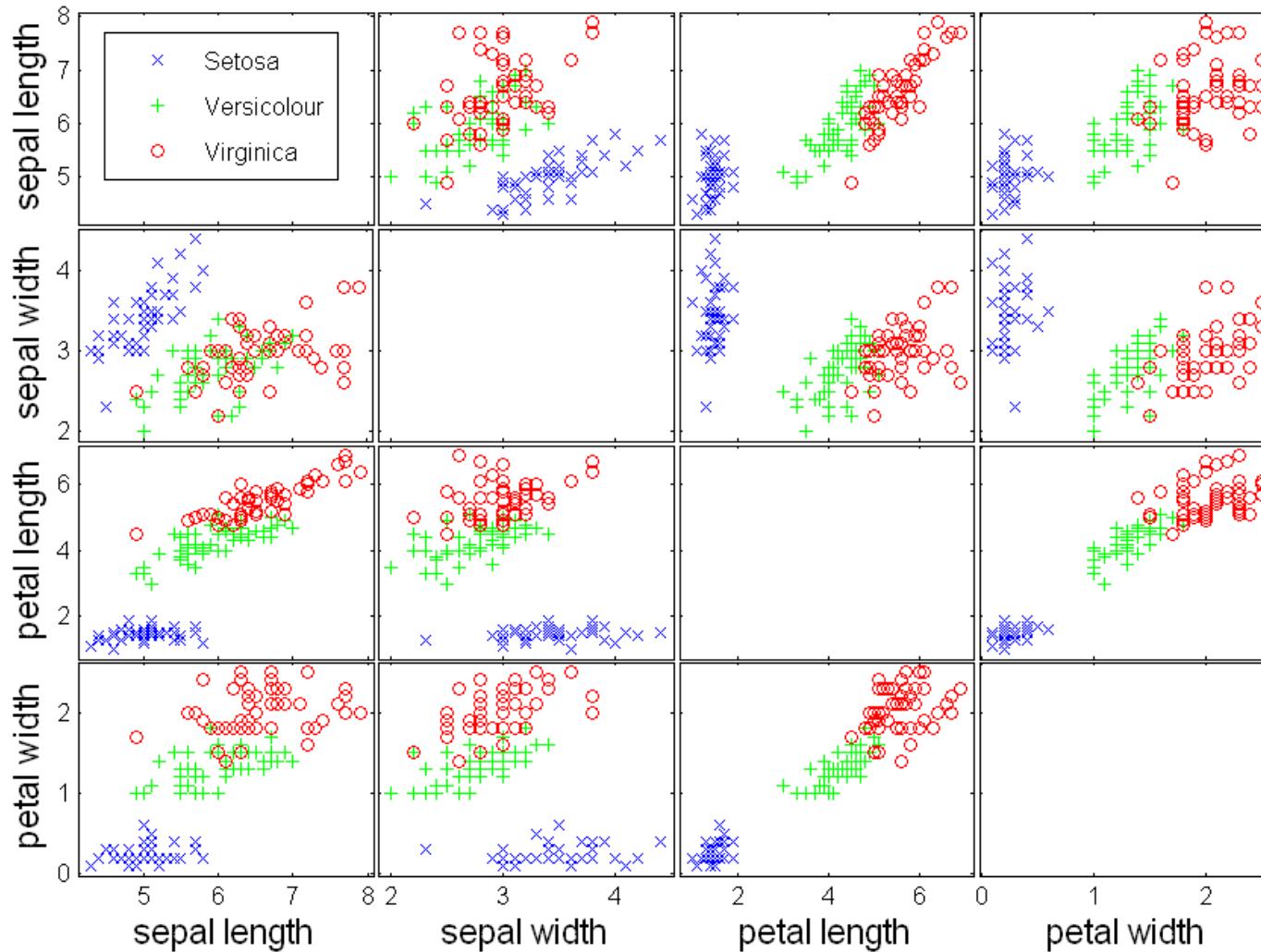


Visualization Techniques: Scatter Plots

□ Scatter plots

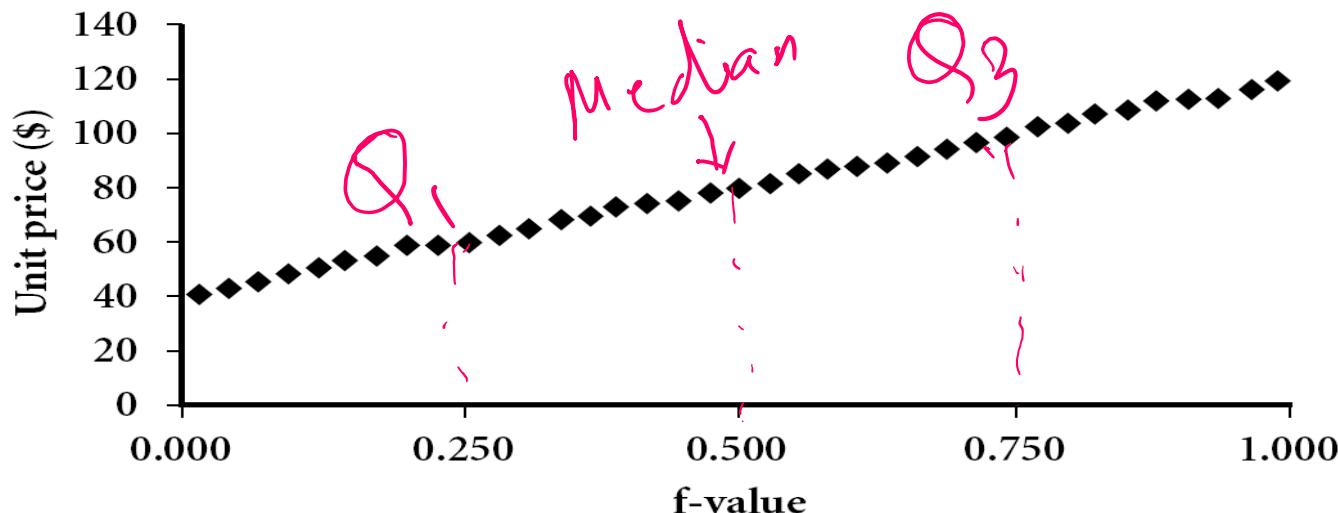
- Attributes values determine the position
- Two-dimensional scatter plots most common, but can have three-dimensional scatter plots
- Often additional attributes can be displayed by using the size, shape, and color of the markers that represent the objects
- It is useful to have arrays of scatter plots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

Scatter Plot Array of Iris Attributes



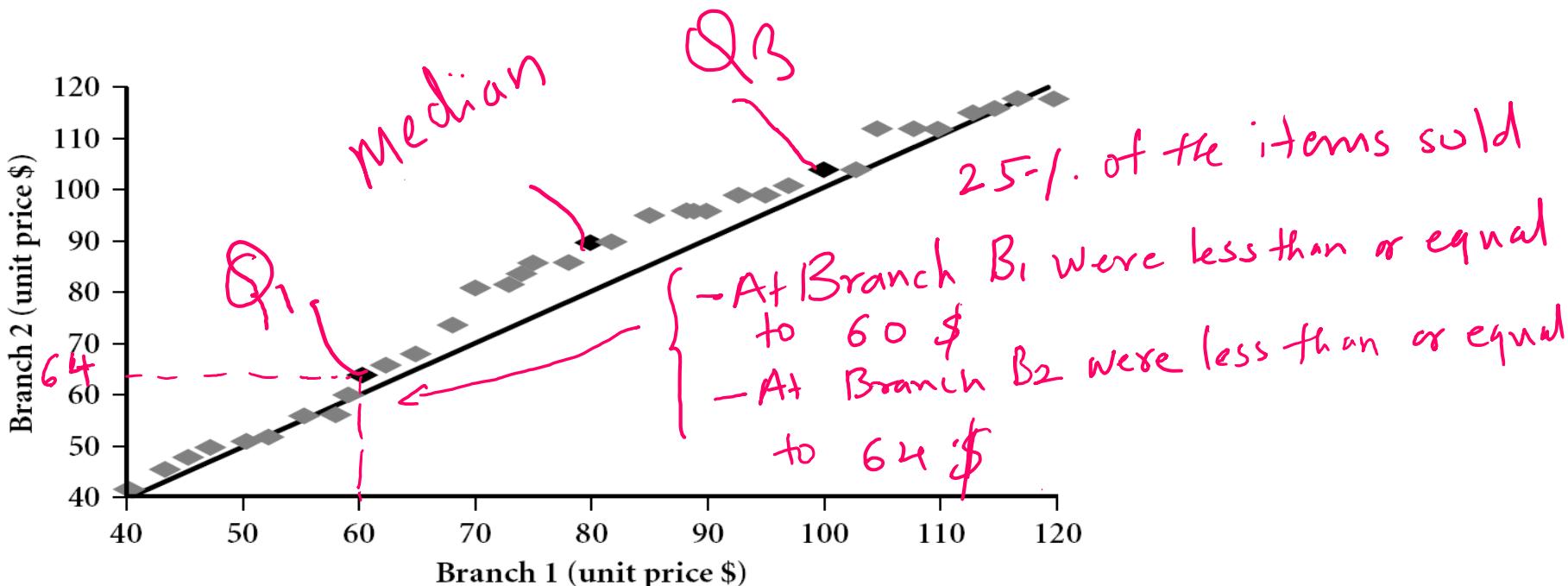
Quantile Plot

- Displays all of the data (allowing the user to assess both the overall behavior and unusual occurrences)
- Plots **quantile** information
 - For a data x_i sorted in increasing order, f_i indicates that approximately $f_i * 100\%$ of the data are below or equal to the value x_i



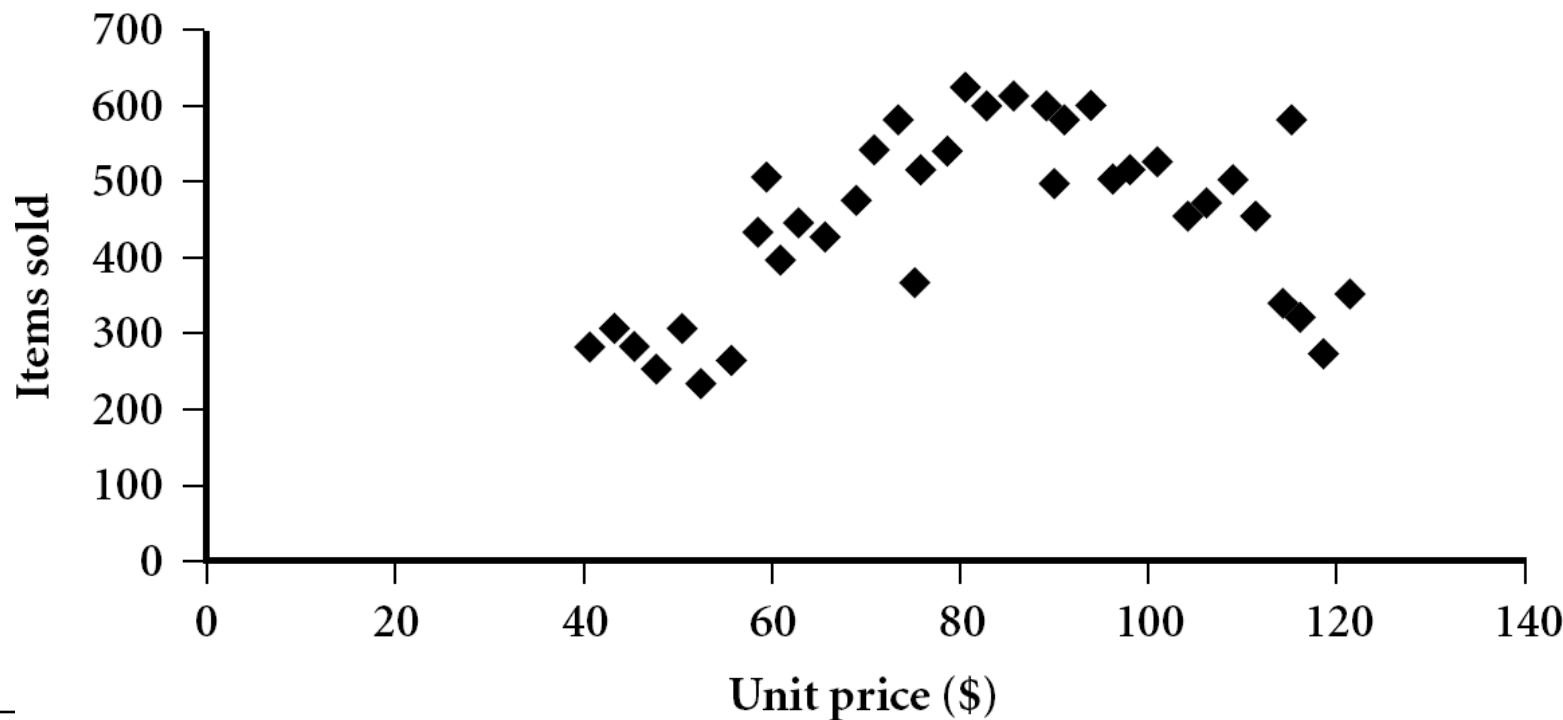
Quantile-Quantile (Q-Q) Plot

- Graphs the quantiles of one univariate distribution against the corresponding quantiles of another
- View: Is there is a shift in going from one distribution to another?
- Example shows unit price of items sold at Branch 1 vs. Branch 2 for each quantile. Unit prices of items sold at Branch 1 tend to be lower than those at Branch 2.

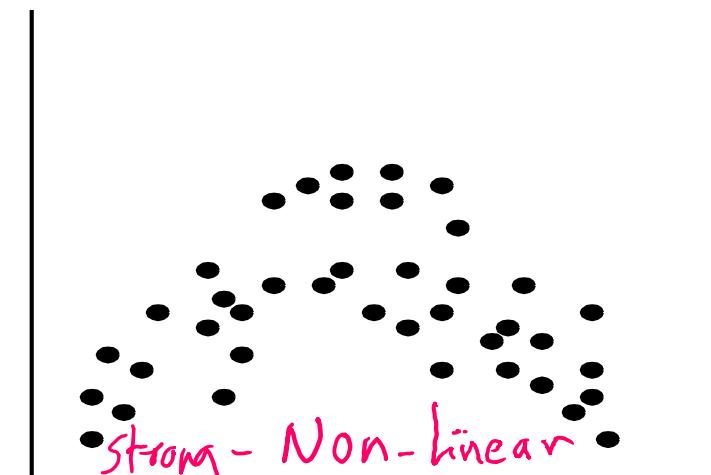
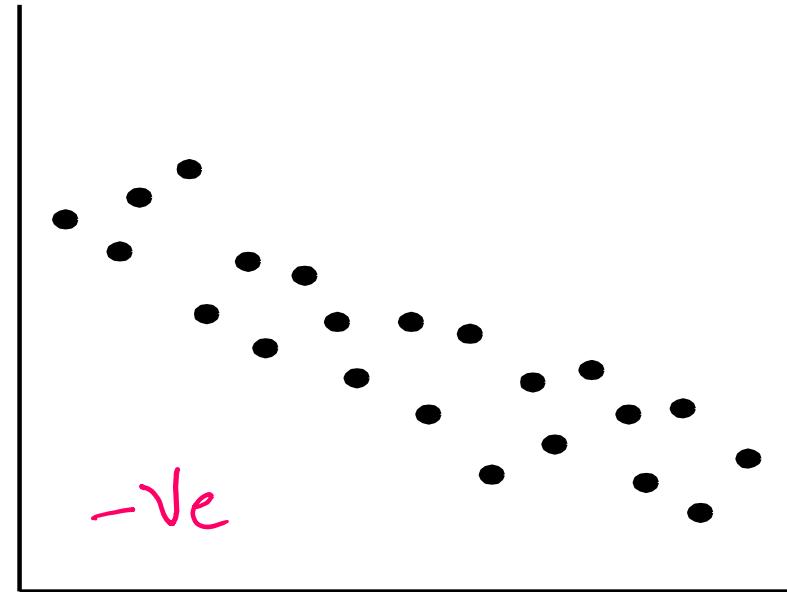
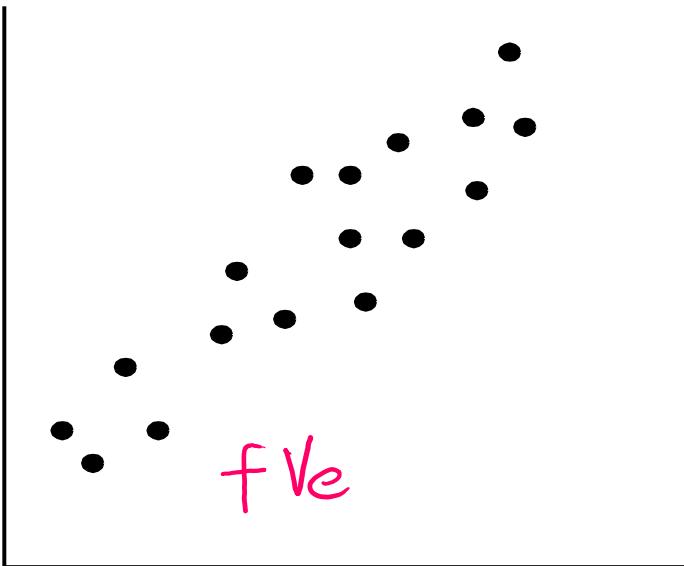


Scatter plot

- Provides a first look at bivariate data to see clusters of points, outliers, etc
- Each pair of values is treated as a pair of coordinates and plotted as points in the plane

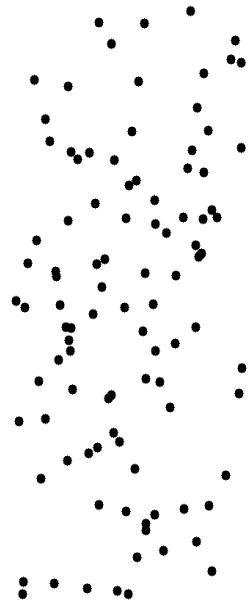
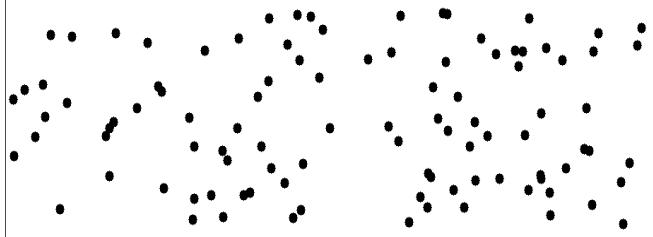
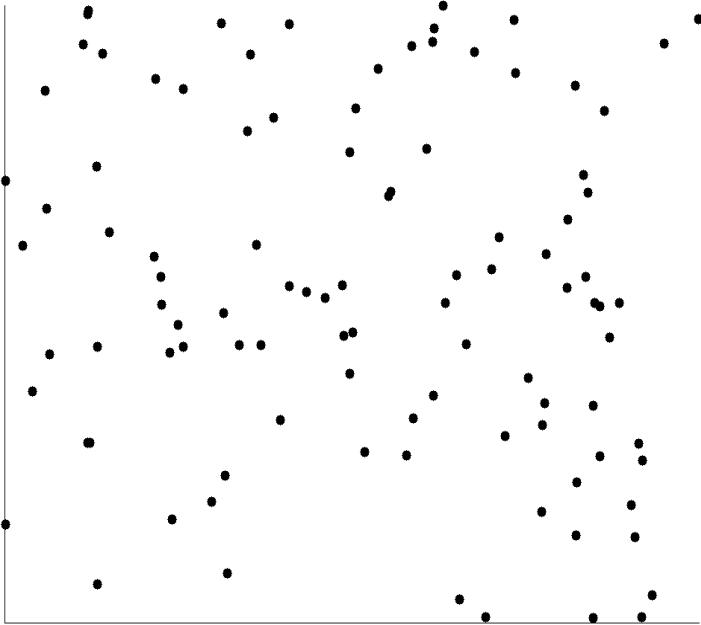


Positively and Negatively Correlated Data



- The left half fragment is positively correlated
- The right half is negative correlated

Uncorrelated Data

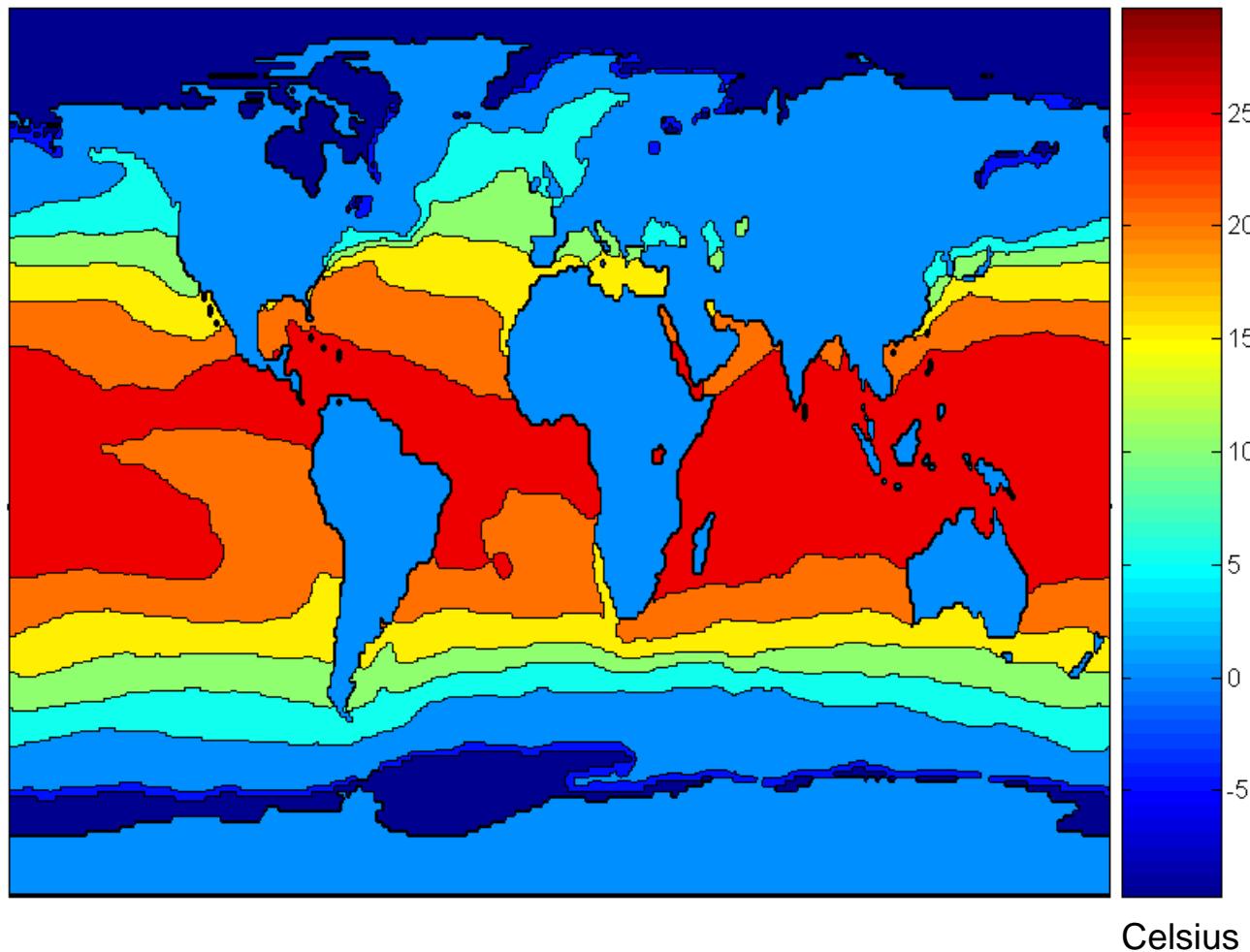


Visualization Techniques: Contour Plots

□ 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

Contour Plot Example: SST Dec, 1998

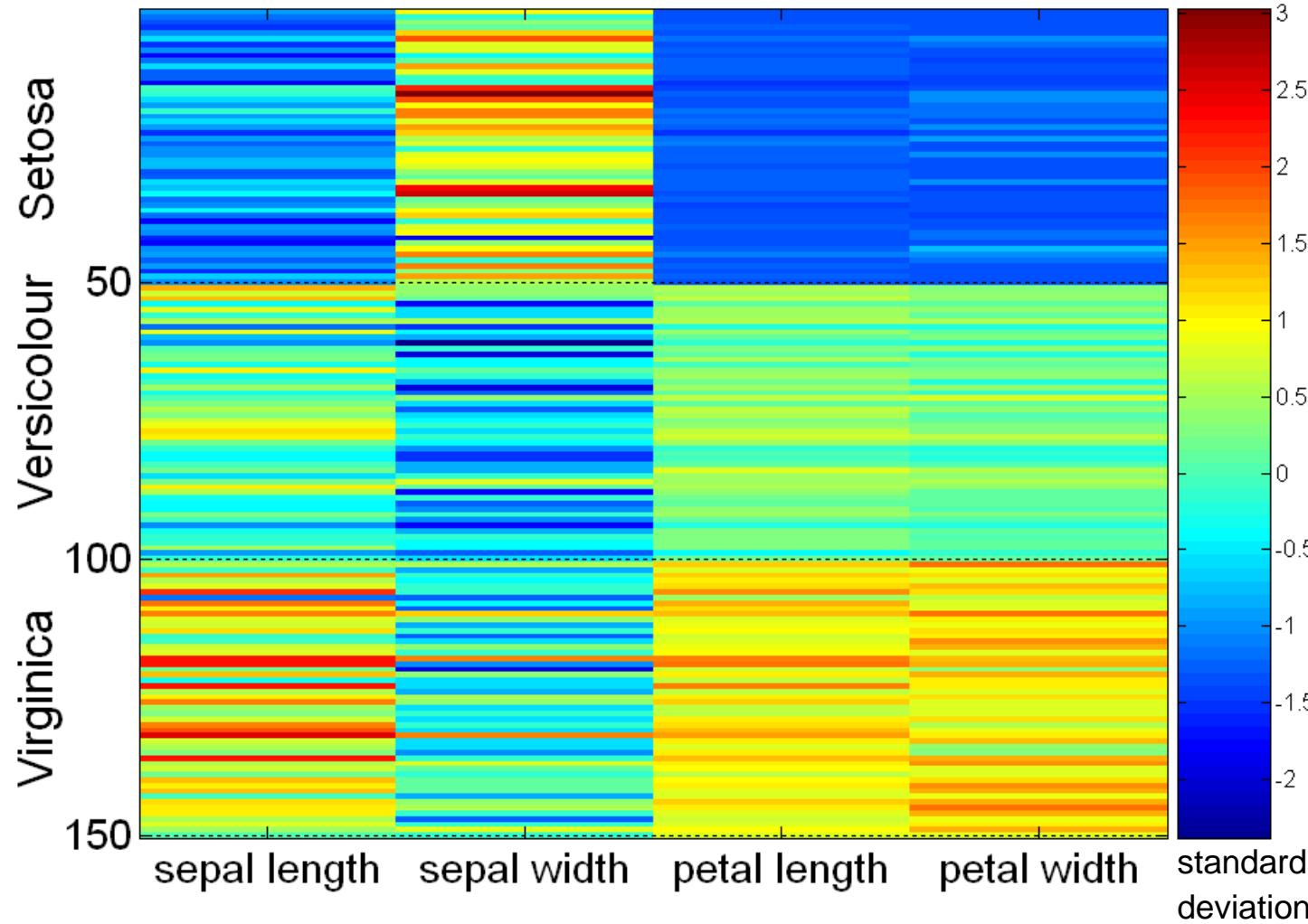


Visualization Techniques: Matrix Plots

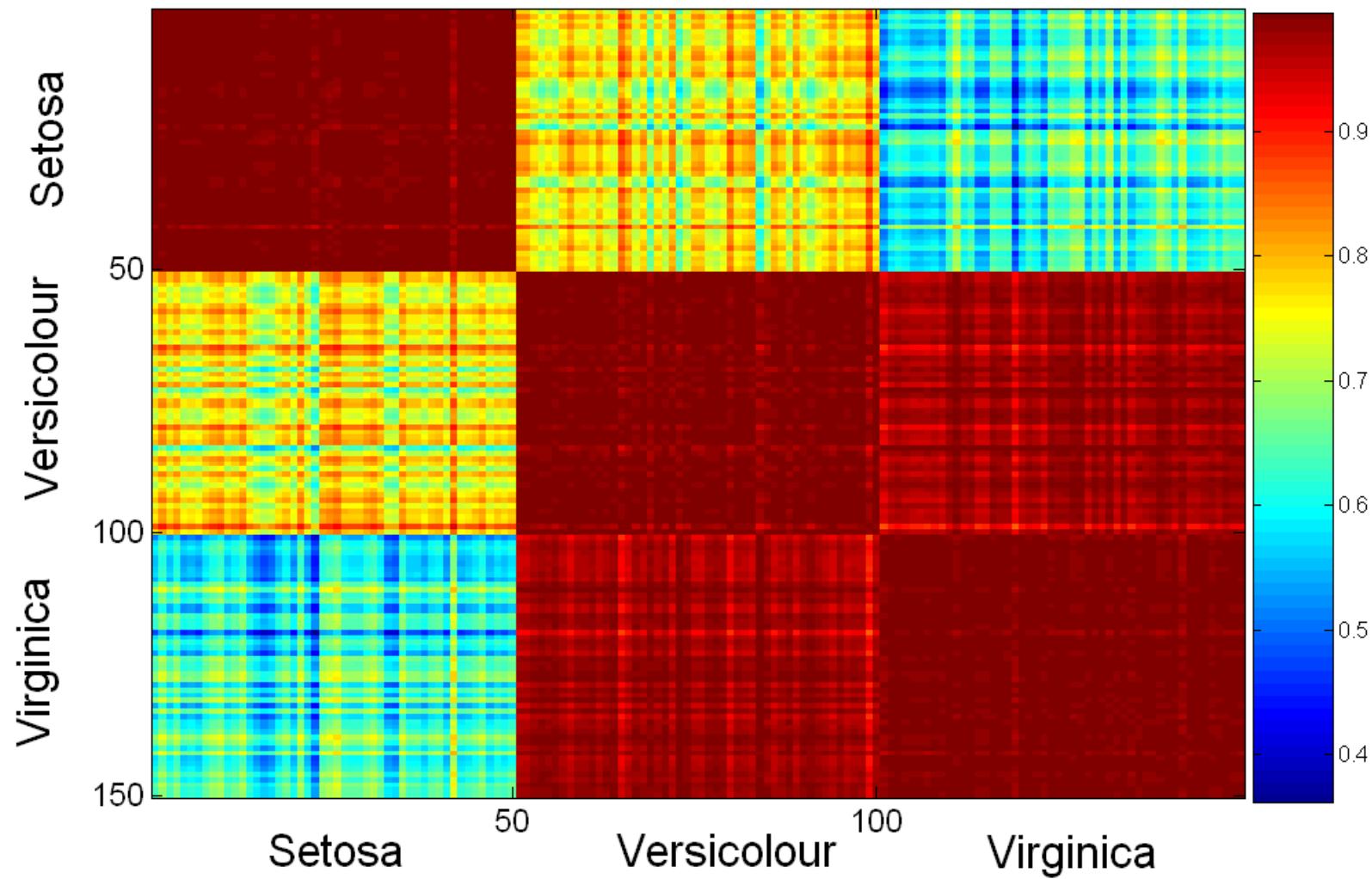
□ Matrix plots

- 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
- Examples of matrix plots are presented on the next two slides

Visualization of the Iris Data Matrix



Visualization of the Iris Correlation Matrix



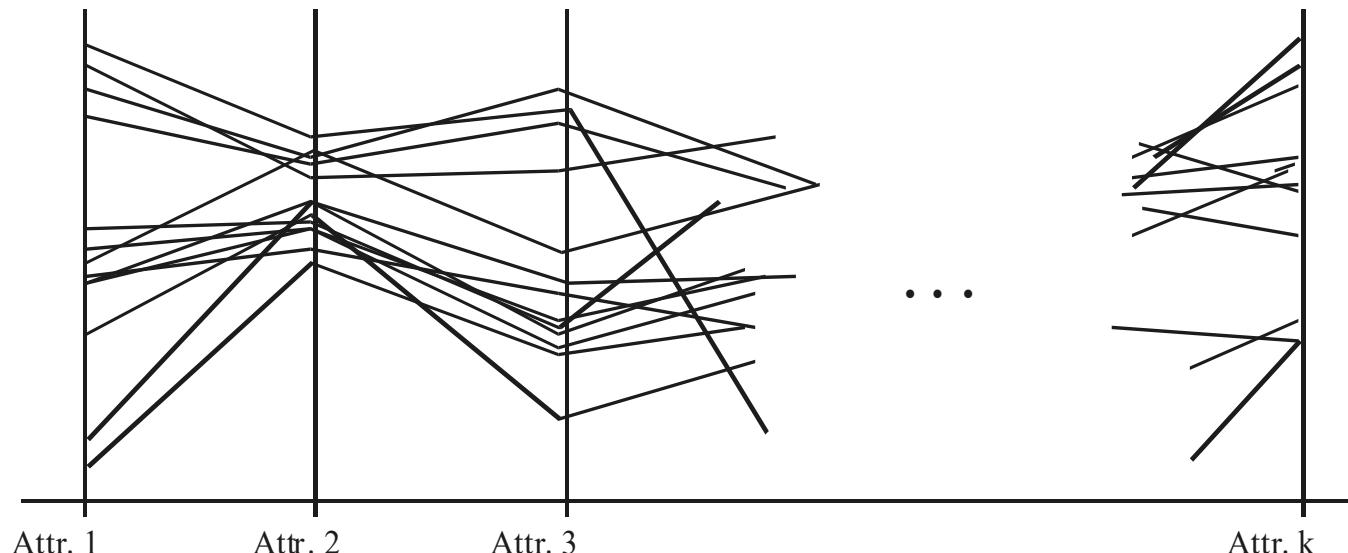
Visualization Techniques: Parallel Coordinates

□ Parallel Coordinates

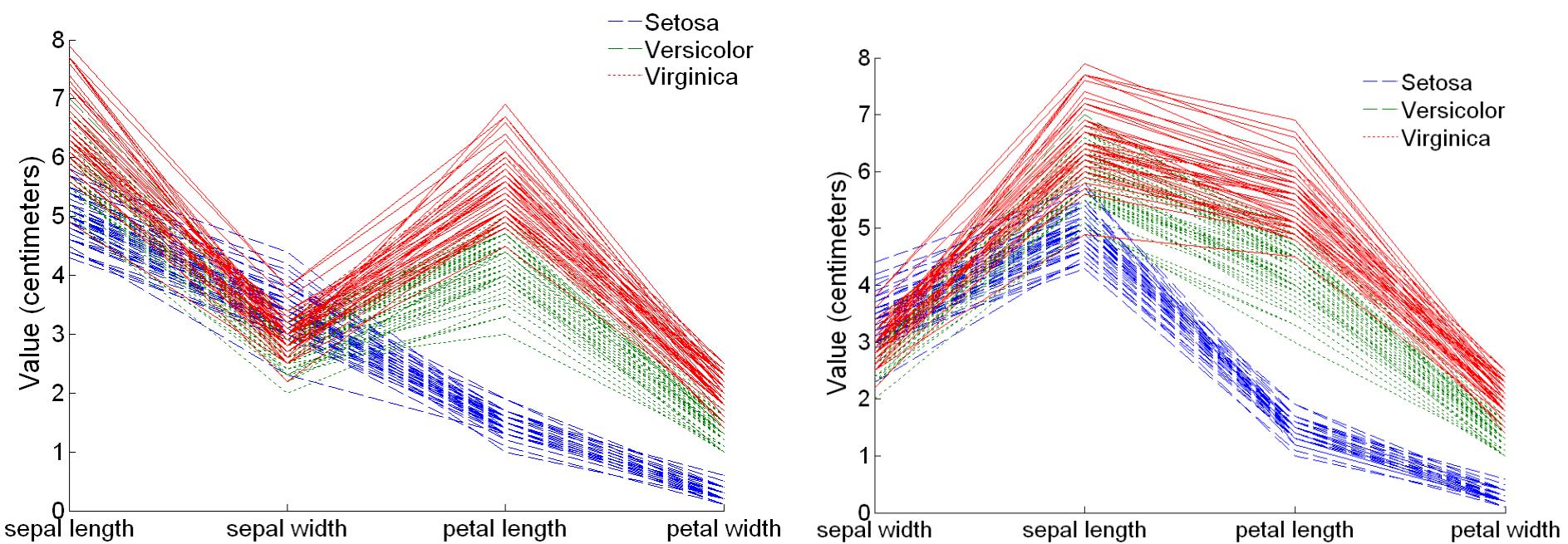
- Used to plot the attribute values of high-dimensional data
- Instead of using perpendicular axes, use a set of parallel axes
- The attribute values of each object are plotted as a point on each corresponding coordinate axis and the points are connected by a line
- Thus, each object is represented as a line
- Often, the lines representing a distinct class of objects group together, at least for some attributes
- Ordering of attributes is important in seeing such groupings

Parallel Coordinates

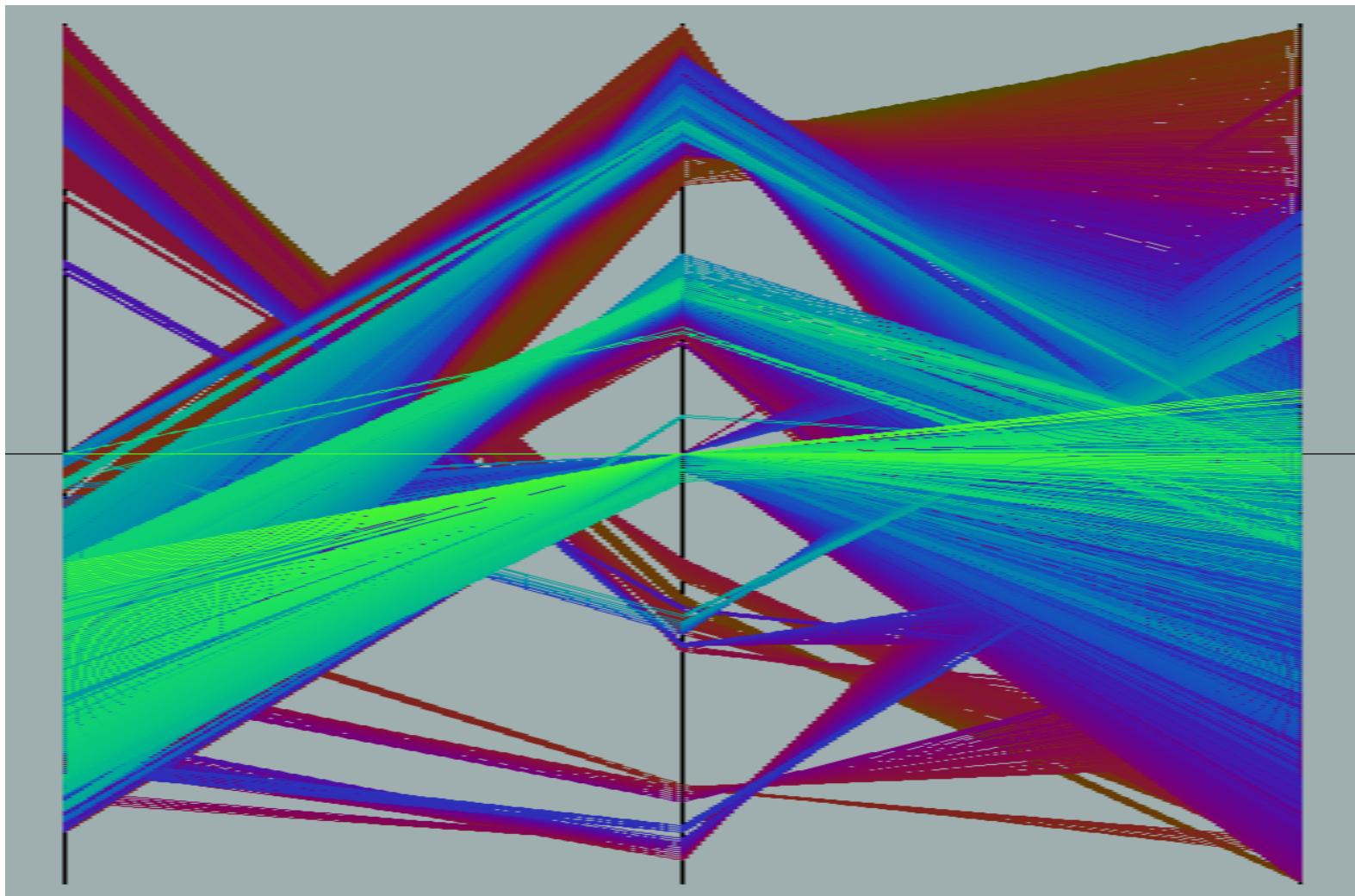
- n equidistant axes which are parallel to one of the screen axes and correspond to the attributes
- The axes are scaled to the [minimum, maximum]: range of the corresponding attribute
- Every data item corresponds to a polygonal line which intersects each of the axes at the point which corresponds to the value for the attribute



Parallel Coordinates Plots for Iris Data



Parallel Coordinates of a Data Set



Other Visualization Techniques

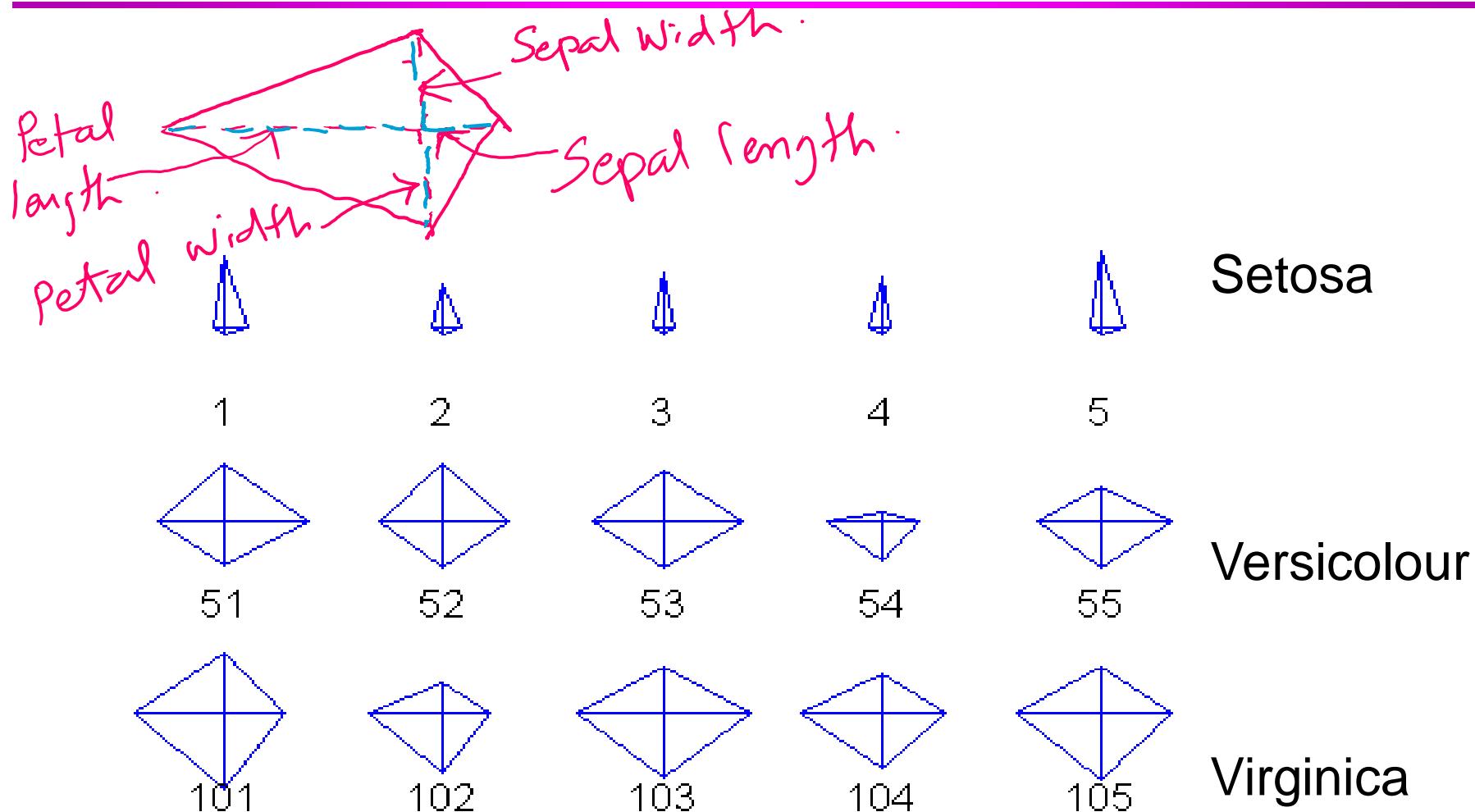
□ Star Plots

- 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

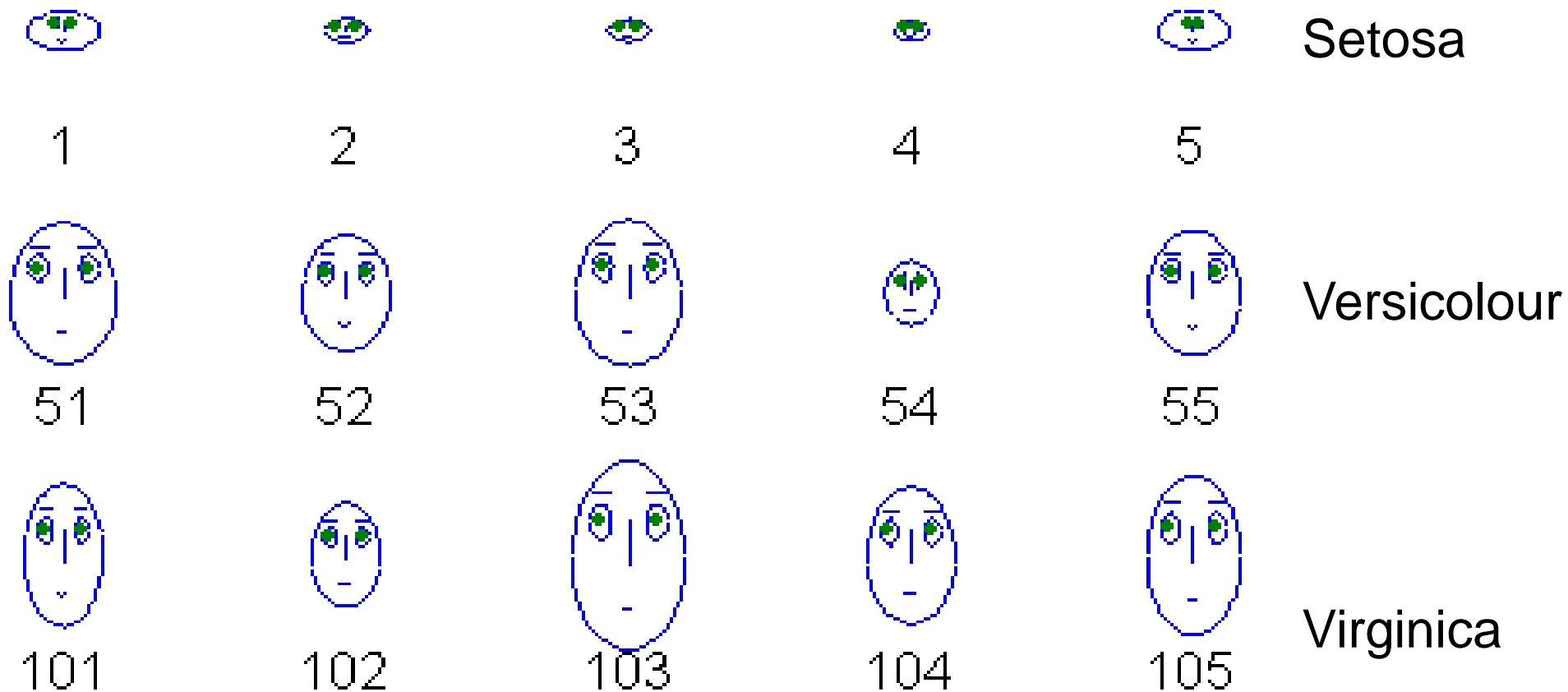
- Approach created by Herman Chernoff
- This approach associates each attribute with a characteristic of a face
- The values of each attribute determine the appearance of the corresponding facial characteristic
- Each object becomes a separate face
- Relies on human's ability to distinguish faces

Star Plots for Iris Data



Chernoff Faces for Iris Data

Plot of 15 Iris flowers using Chernoff Faces



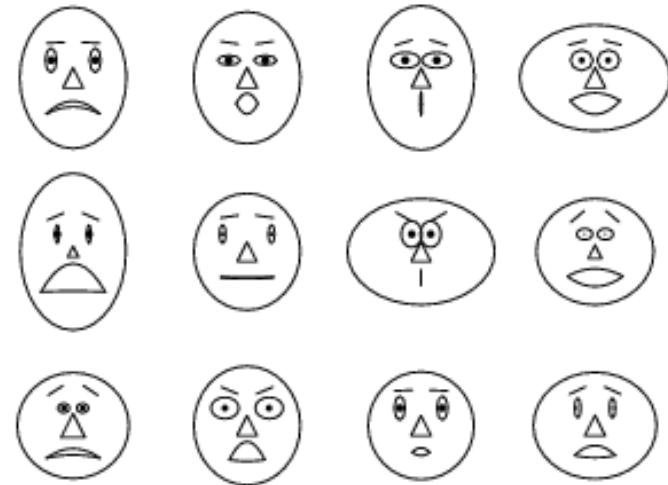
Chernoff Faces

- A way to display variables on a two-dimensional surface, e.g., let x be eyebrow slant, y be eye size, z be nose length, etc.
- The figure shows faces produced using 10 characteristics--head eccentricity, eye size, eye spacing, eye eccentricity, pupil size, eyebrow slant, nose size, mouth shape, mouth size, and mouth opening): Each assigned one of 10 possible values, generated using *Mathematica* (S. Dickson)

- REFERENCE: Gonick, L. and Smith, W. *The Cartoon Guide to Statistics*. New York:

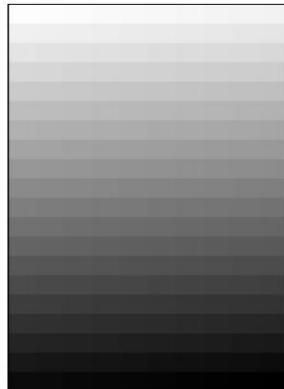
Harper Perennial, p. 212, 1993

- Weisstein, Eric W. "Chernoff Face." From *MathWorld--A Wolfram Web Resource*.
mathworld.wolfram.com/ChernoffFace.html

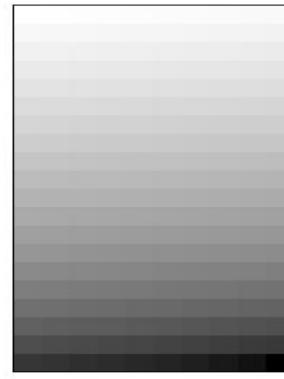


Pixel-Oriented Visualization Techniques

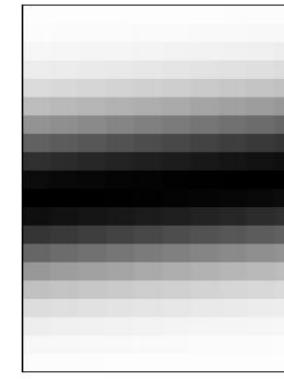
- For a data set of m dimensions, create m windows on the screen, one for each dimension
- The m dimension values of a record are mapped to m pixels at the corresponding positions in the windows
- The colors of the pixels reflect the corresponding values (Lighter = small value)
- Ex: Income vs Others : Credit limit increases with Income, Mid income people more likely to purchase more items, No correlation between Income and Age



(a) Income



(b) Credit Limit



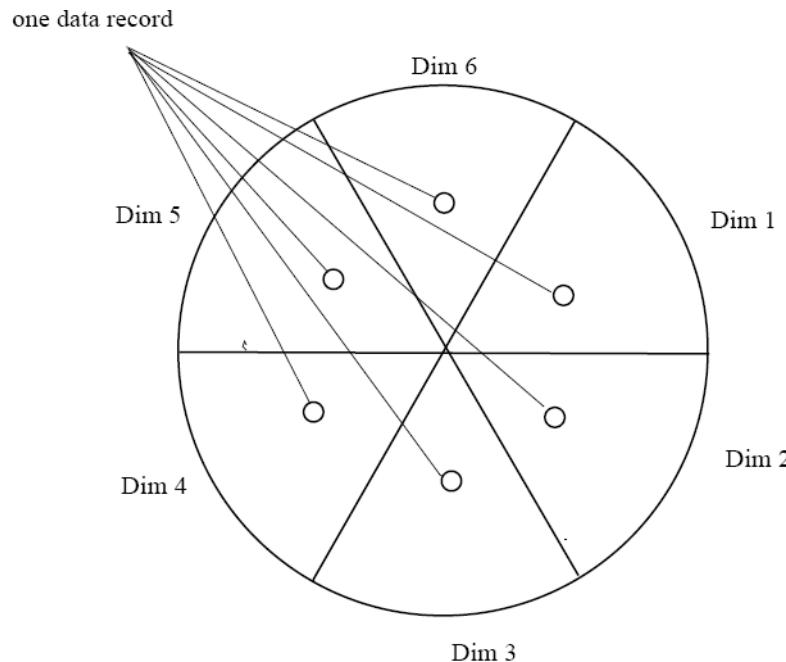
(c) transaction volume



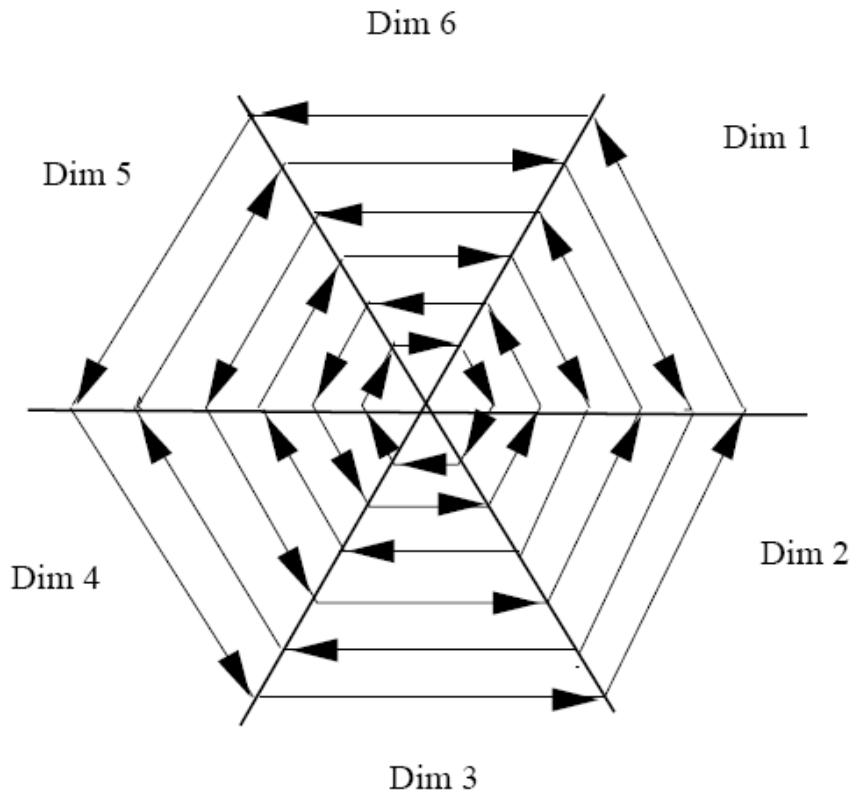
(d) age

Laying Out Pixels in Circle Segments

- To save space and show the connections among multiple dimensions, space filling is often done in a circle segment



(a) Representing a data record
in circle segment



(b) Laying out pixels in circle segment

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.
 - Such representations of data previously existed in statistics and other fields
- There are a number of data analysis and data exploration operations that are easier with such a data representation.

Creating a Multidimensional Array

- Two key steps in converting tabular data into a multidimensional array.
 - First, identify which attributes are to be the dimensions and which attribute is to be the target attribute whose values appear as entries in the multidimensional array.
 - ◆ The attributes used as dimensions must have discrete values
 - ◆ The target value is typically a count or continuous value, e.g., the cost of an item
 - ◆ Can have no target variable at all except the count of objects that have the same set of attribute values
 - Second, find the value of each entry in the multidimensional array by summing the values (of the target attribute) or count of all objects that have the attribute values corresponding to that entry.

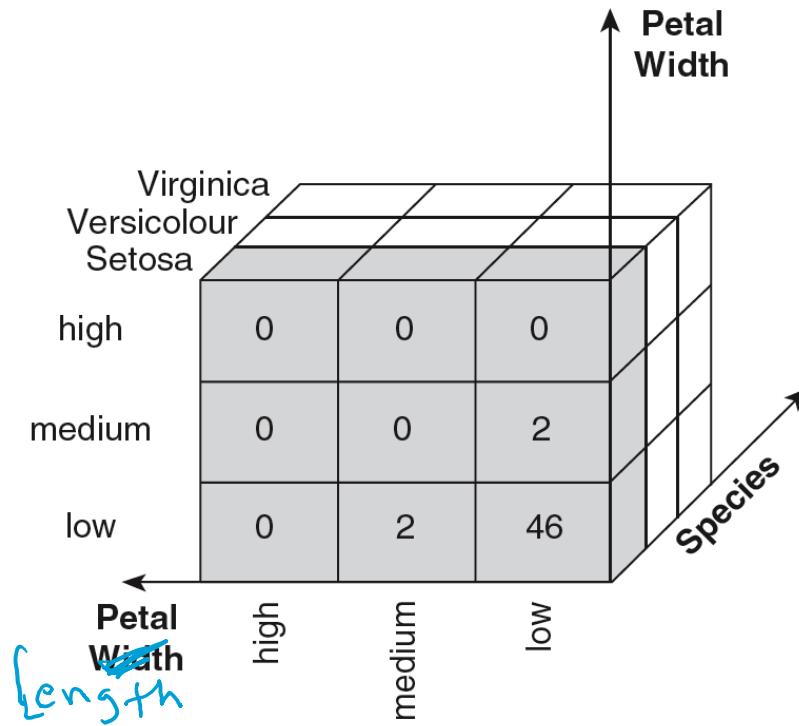
Example: Iris data

- We show how the attributes, petal length, petal width, and species type can be converted to a multidimensional array
 - First, we discretized the petal width and length to have categorical values: *low*, *medium*, and *high*
 - We get the following table - note the count attribute

| Petal Length | Petal Width | Species Type | Count |
|--------------|-------------|--------------|-------|
| low | low | Setosa | 46 |
| low | medium | Setosa | 2 |
| medium | low | Setosa | 2 |
| medium | medium | Versicolour | 43 |
| medium | high | Versicolour | 3 |
| medium | high | Virginica | 3 |
| high | medium | Versicolour | 2 |
| high | medium | Virginica | 3 |
| high | high | Versicolour | 2 |
| high | high | Virginica | 44 |

Example: Iris data (continued)

- Each unique tuple of petal width, petal length, and species type identifies one element of the array.
- This element is assigned the corresponding count value.
- The figure illustrates the result.
- All non-specified tuples are 0.



Example: Iris data (continued)

- Slices of the multidimensional array are shown by the following cross-tabulations
- What do these tables tell us?

| | | Width | | |
|--------|--------|-------|--------|------|
| | | low | medium | high |
| Length | low | 46 | 2 | 0 |
| | medium | 2 | 0 | 0 |
| | high | 0 | 0 | 0 |

Setosa

| | | Width | | |
|--------|--------|-------|--------|------|
| | | low | medium | high |
| Length | low | 0 | 0 | 0 |
| | medium | 0 | 43 | 3 |
| | high | 0 | 2 | 2 |

Versicolour

| | | Width | | |
|--------|--------|-------|--------|------|
| | | low | medium | high |
| Length | low | 0 | 0 | 0 |
| | medium | 0 | 0 | 3 |
| | high | 0 | 3 | 44 |

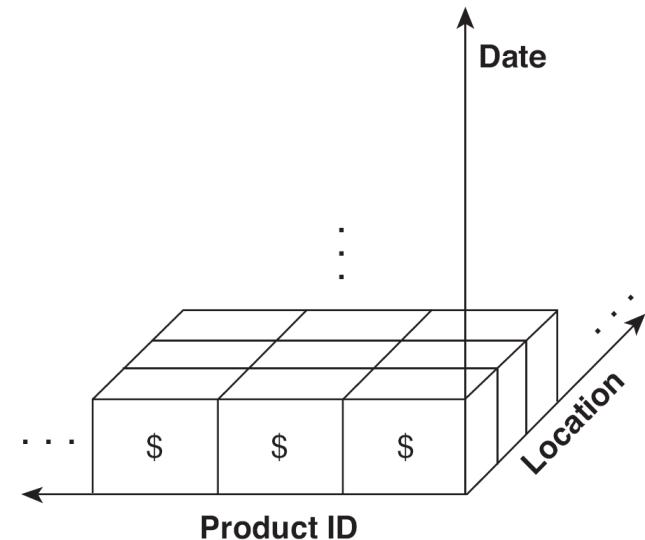
Virginica

OLAP Operations: Data Cube

- The key operation of a OLAP is the **formation of a data cube**
- A data cube is a multidimensional representation of data, together with all possible aggregates.
- By all possible aggregates, we mean the aggregates that result by selecting a proper subset of the dimensions and summing over all remaining dimensions.
- For example, if we choose the species type dimension of the Iris data and sum over all other dimensions, the result will be a one-dimensional entry with three entries, each of which gives the number of flowers of each type.

Data Cube Example

- Consider a data set that records the sales of products at a number of company stores at various dates.
- This data can be represented as a 3 dimensional array
- There are 3 two-dimensional aggregates (3 choose 2), 3 one-dimensional aggregates, and 1 zero-dimensional aggregate (the overall total)



Data Cube Example (continued)

- The following figure table shows one of the two dimensional aggregates, along with two of the one-dimensional aggregates, and the overall total

| product ID | date | | | | total |
|------------|-------------|-------------|-----|--------------|---------------|
| | Jan 1, 2004 | Jan 2, 2004 | ... | Dec 31, 2004 | |
| 1 | \$1,001 | \$987 | ... | \$891 | \$370,000 |
| : | : | | | : | : |
| 27 | \$10,265 | \$10,225 | ... | \$9,325 | \$3,800,020 |
| : | : | | | : | : |
| total | \$527,362 | \$532,953 | ... | \$631,221 | \$227,352,127 |

OLAP Operations: Slicing and Dicing

- Slicing is selecting a group of cells from the entire multidimensional array by specifying a specific value for one or more dimensions.
- Dicing involves selecting a subset of cells by specifying a range of attribute values.
 - This is equivalent to defining a subarray from the complete array.
- In practice, both operations can also be accompanied by aggregation over some dimensions.

OLAP Operations: Roll-up and Drill-down

- Attribute values often have a hierarchical structure.
 - Each date is associated with a year, month, and week.
 - A location is associated with a continent, country, state (province, etc.), and city.
 - Products can be divided into various categories, such as clothing, electronics, and furniture.
- Note that these categories often nest and form a tree or lattice
 - A year contains months which contains day
 - A country contains a state which contains a city

OLAP Operations: Roll-up and Drill-down

- This hierarchical structure gives rise to the roll-up and drill-down operations.
 - For sales data, we can aggregate (roll up) the sales across all the dates in a month.
 - Conversely, given a view of the data where the time dimension is broken into months, we could split the monthly sales totals (drill down) into daily sales totals.
 - Likewise, we can drill down or roll up on the location or product ID attributes.

Data Mining

- **4. Data Mining Classification:
Basic Concepts, Decision Trees, and Model
Evaluation**

What is Classification?

- Goal: Previously unseen records should be assigned a class from a given set of classes as accurately as possible
 - **Classification** : Classification is the task of learning a target function f that maps each attribute set \mathbf{x} to one of the predefined class labels y .
 - The target function is also known informally as a **classification model**.
 - A classification model is useful for the following purposes.
 - 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.
 - 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.
- Variants:
- Binary classification (e.g. fraud/no fraud or true/false)
 - Multi-class classification (e.g. low, medium, high)
 - Multi-label classification (more than one class per record, e.g. user interests)



Introduction to Classification

A Couple of Questions:

- **What is this?**
- **Why do you know?**
- **How have you come to that knowledge?**

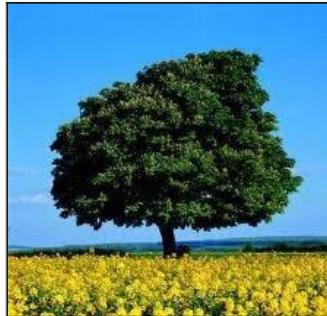


Introduction to Classification

- Goal: Learn a model for recognizing a concept, e.g. trees
- Training data:



"tree"



"tree"



"tree"



"not a tree"



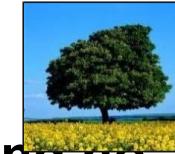
"not a tree"



"not a tree"

Introduction to Classification

- We (or the learning algorithm) look at positive and negative examples (**training data**)
- ... and derive a **model**
e.g., "Trees are big, green plants that have a trunk and no wheels."
- Goal: Classification of **unseen instances**



Warning:
Models are only
approximating examples!
Not guaranteed to be
correct or complete!

- A decision tree is a **flowchart-like structure**.
 - Each internal node **represents** a “test” on an attribute
(e.g. whether a coin flip comes up heads or tails),
 - Each **branch** represents the **outcome of the test**, and
 - Each **leaf node** represents a **class label**
(decision taken after computing all attributes).
 - The **paths from root to leaf** represent **classification rules**.
-
- Tree based learning algorithms: one of the **best and mostly used supervised learning methods**.
 - Tree based methods empower predictive models with **high accuracy, stability and ease of interpretation**.
 - Unlike linear models, they **map non-linear relationships quite well**.
 - They are adaptable at solving any kind of problem at hand (**classification or regression**).
 - Decision Tree algorithms are referred to as **CART (Classification and Regression Trees)**.

□ Predictive Modeling:

- A classification model can also be used to **predict the class label** of unknown records.
- Classification techniques are most **suited for predicting or describing data sets with binary or nominal categories**.
 - They are **less effective for ordinal categories** (e.g., to classify a person as a member of high-, medium-, or low income group) because they **do not consider the implicit order among the categories**.
 - Other forms of **relationships, such as the subclass–superclass relationships** among categories (e.g., humans and apes are primates, which in turn, is a subclass of mammals) **are also ignored**.

Common terms used with Decision trees:

- **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting:** It is a process of dividing a node into two or more sub-nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
- **Leaf / Terminal Node:** Nodes do not split is called Leaf or Terminal node.
- **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
- **Branch / Sub-Tree:** A sub section of entire tree is called branch or sub-tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes, whereas, sub-nodes are the child of parent node.

Types of Decision Trees

- Types of decision tree is **based on the type of target variable we have.**
- Two types:
- **Categorical Variable Decision Tree:**

Decision Tree which has categorical target variable then it called as categorical variable decision tree. E.g.: - the target variable “Student will play cricket or not” i.e. YES or NO.
- **Continuous Variable Decision Tree:**

Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

Decision Tree Algorithm Pseudocode

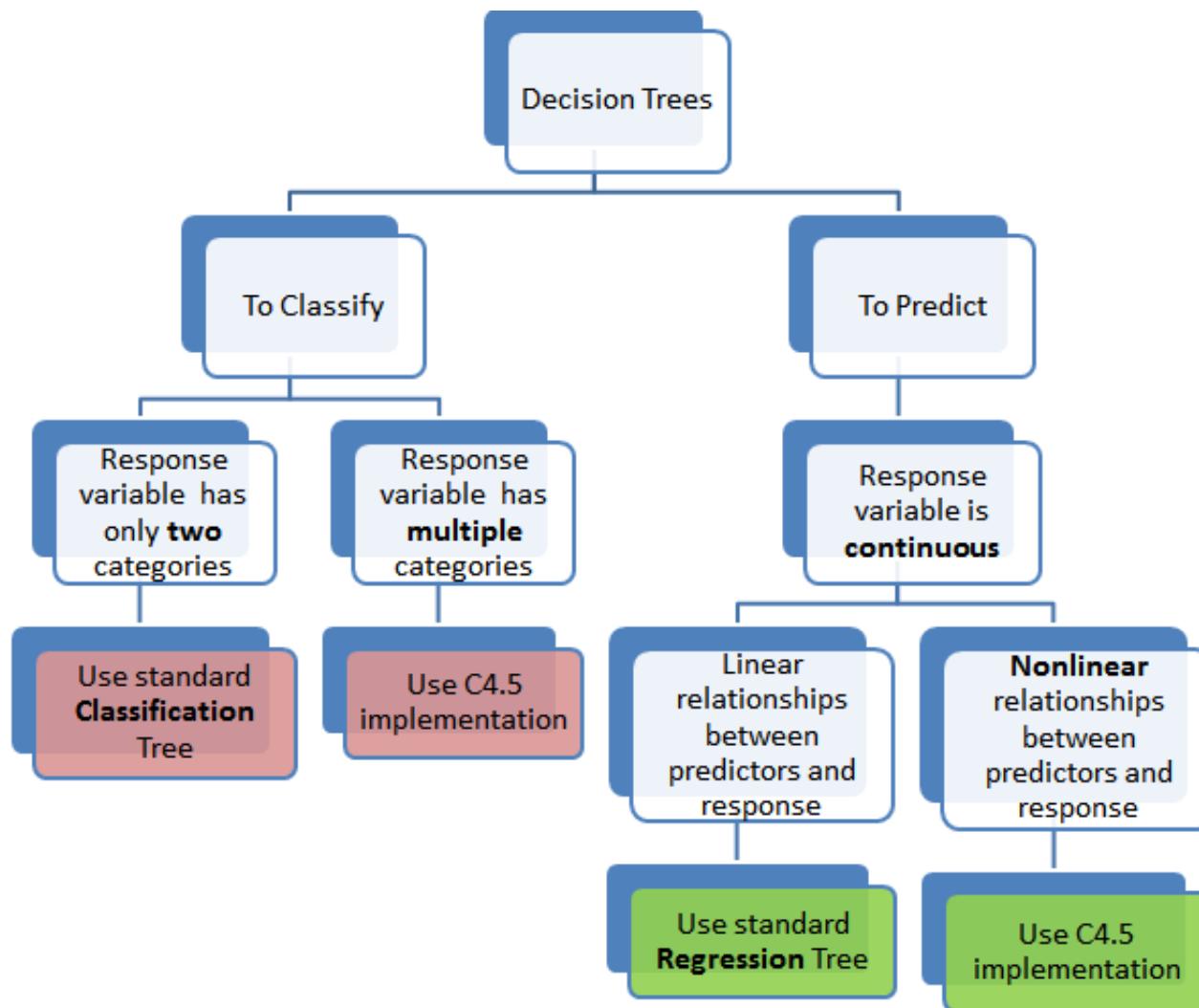
- **Decision Tree Algorithm Pseudocode**
 - The decision tree algorithm tries to solve the problem, by using tree representation.
 - Each internal node of the tree corresponds to an attribute, and
 - each leaf node corresponds to a class label.
1. Place the best attribute of the dataset at the **root** of the tree.
 2. Split the training set into **subsets**. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
- Repeat step 1 and step 2 on each subset until you find **leaf nodes** in all the branches of the tree.
 - In decision trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.
 - We continue comparing our record's attribute values with other **internal nodes** of the tree until we reach **a leaf node** with predicted class value. The modeled decision tree can be used to predict the target class or the value.

Assumptions while creating Decision Tree

Some of the assumptions we make while using Decision tree:

- At the beginning, the whole training set is considered as the **root**.
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are **distributed recursively** on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

Types of decision trees



Advantages of Decision Tree:

- **Easy to Understand:** Decision tree output is very easy to understand. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive.
- **Useful in Data exploration:** Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. With the help of decision trees, we can create new variables / features that has better power to predict target variable. It can also be used in data exploration stage. For e.g., we are working on a problem where we have information available in hundreds of variables, there decision tree will help to identify most significant variable.
- Decision trees implicitly perform variable screening or feature selection.
- Decision trees require relatively **little effort from users for data preparation**.
- **Less data cleaning required:** It requires less data cleaning. It is not influenced by outliers and missing values to a fair degree.
- **Data type is not a constraint:** It can handle both numerical and categorical variables. Can also *handle multi-output problems*.
- **Non-Parametric Method:** Decision tree is considered as non-parametric method. i.e decision trees have no assumptions about the space distribution and the classifier structure.
- Non-linear relationships between parameters do not affect tree performance.
- The number of hyper-parameters to be tuned is almost null.

Disadvantages of Decision Tree:

- **Over fitting:** Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting. Over fitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning.
- **Not fit for continuous variables:** While working with continuous numerical variables, decision tree loses information, when it categorizes variables in different categories.
- Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This is called **variance**, which needs to be lowered by methods like **bagging** and **boosting**.
- Greedy algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees, where the features and samples are randomly sampled with replacement.
- Decision tree learners create *biased* trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.
- Information gain in a decision tree with categorical variables gives a biased response for attributes with greater no. of categories.
- Generally, it gives low prediction accuracy for a dataset as compared to other machine learning algorithms.
- Calculations can become complex when there are many class label.

Regression Trees vs Classification Trees

- The terminal nodes (or leaves) lies at the bottom of the decision tree. This means that decision trees are typically drawn upside down such that leaves are the bottom & roots are the tops.
- Both the trees work almost similar to each other.
- The primary differences and similarities between Classification and Regression Trees are:
- Regression trees are used when dependent variable is continuous. Classification Trees are used when dependent variable is categorical.
- In case of Regression Tree, the value obtained by terminal nodes in the training data is **the mean response of** observation falling in that region. Thus, if an unseen data observation falls in that region, we'll make its prediction with mean value.
- In case of Classification Tree, the value (class) obtained by terminal node in the training data is **the mode of** observations falling in that region. Thus, if an unseen data observation falls in that region, we'll make its prediction with mode value.
- Both the trees divide the predictor space (independent variables) into distinct and non-overlapping regions.
- Both the trees follow a top-down greedy approach known as recursive binary splitting. We call it as 'top-down' because it begins from the top of tree when all the observations are available in a single region and successively splits the predictor space into two new branches down the tree. It is known as 'greedy' because, the algorithm cares (looks for best variable available) about only the current split, and not about future splits which will lead to a better tree.

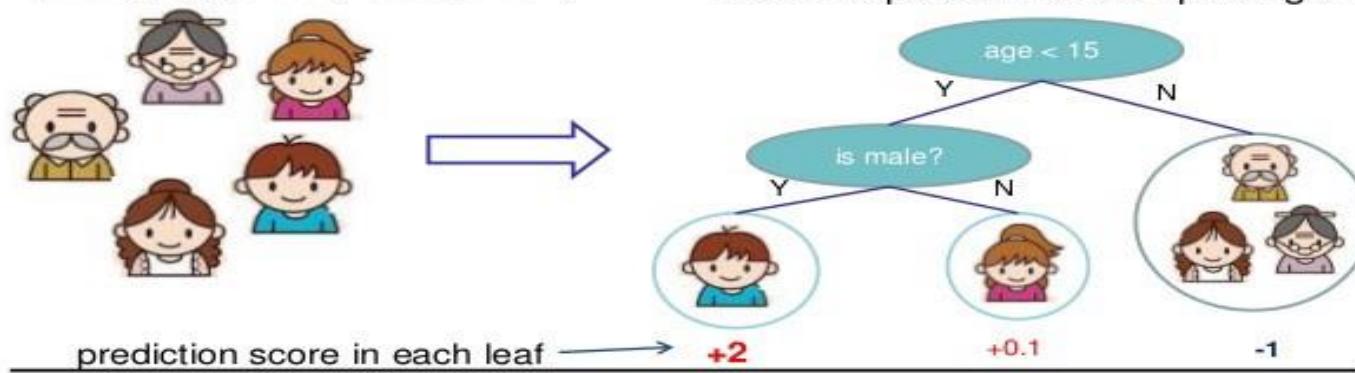
This splitting process is continued until a user defined stopping criteria is reached. For e.g.: we can tell the algorithm to stop once the number of observations per node becomes less than 50.

In both the cases, the splitting process results in fully grown trees until the stopping criteria is reached. But, the fully grown tree is likely to over fit data, leading to poor accuracy on unseen data. This bring 'pruning'. Pruning is one of the technique used tackle overfitting.

Regression Tree (CART)

- regression tree (also known as classification and regression tree):
 - Decision rules same as in decision tree
 - Contains one score in each leaf value

Input: age, gender, occupation, ... Does the person like computer games



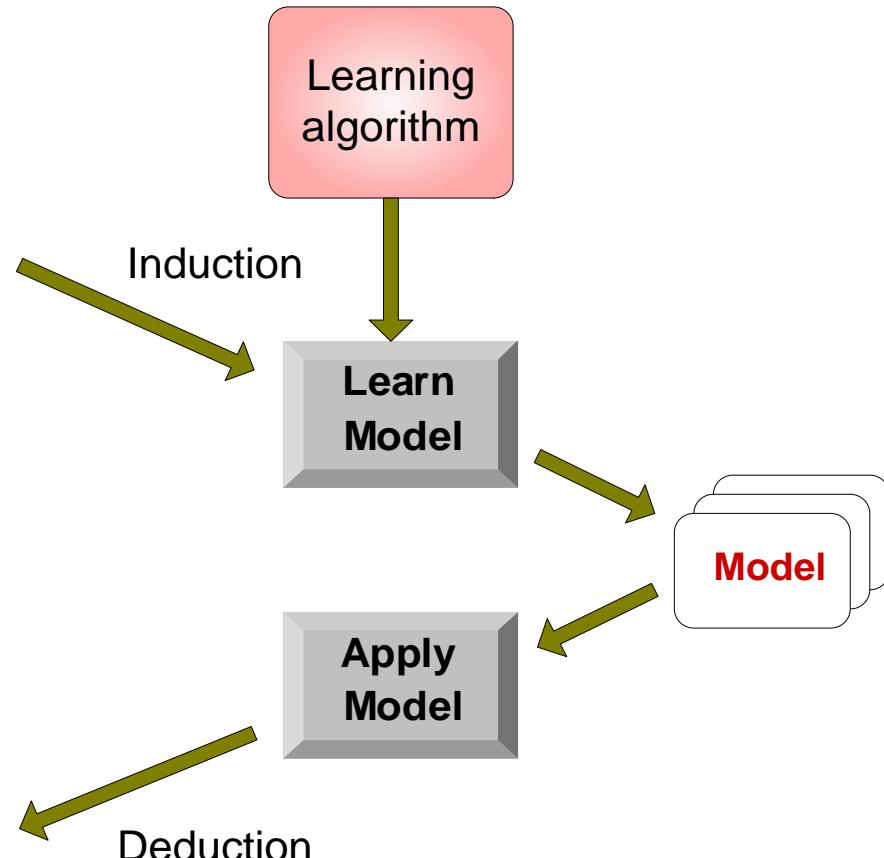
Illustrating Classification Task

| 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



Examples of Classification Task

Credit Risk Assessment

Attributes: your age, income, debts, ...

Class: are you getting credit by your bank?



Marketing

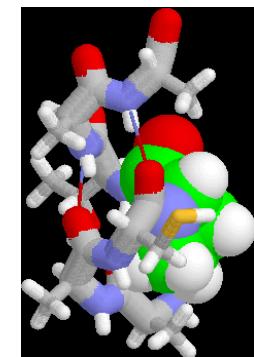
Attributes: previously bought products, browsing behavior

Class: are you a target customer for a new product?

SPAM Detection

Attributes: words and header fields of an e-mail

Class: regular e-mail or spam e-mail?



Identifying Tumor Cells

Attributes: features extracted from x-rays or MRI scans

Class: malignant or benign cells

Classifying secondary structures of protein

as alpha-helix, beta-sheet, or random coil

Categorizing news stories as finance, weather, entertainment, sports...

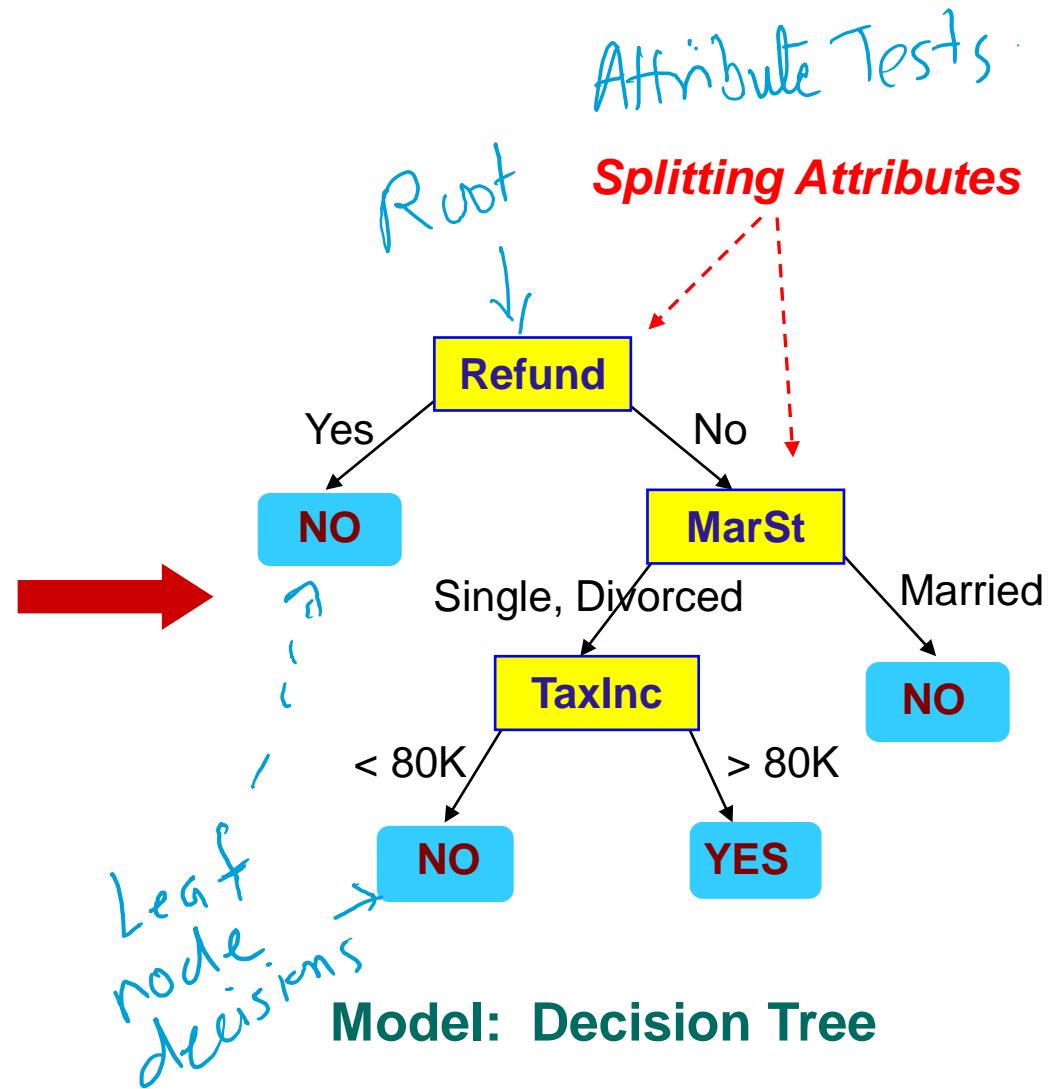
Classification Techniques

- 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

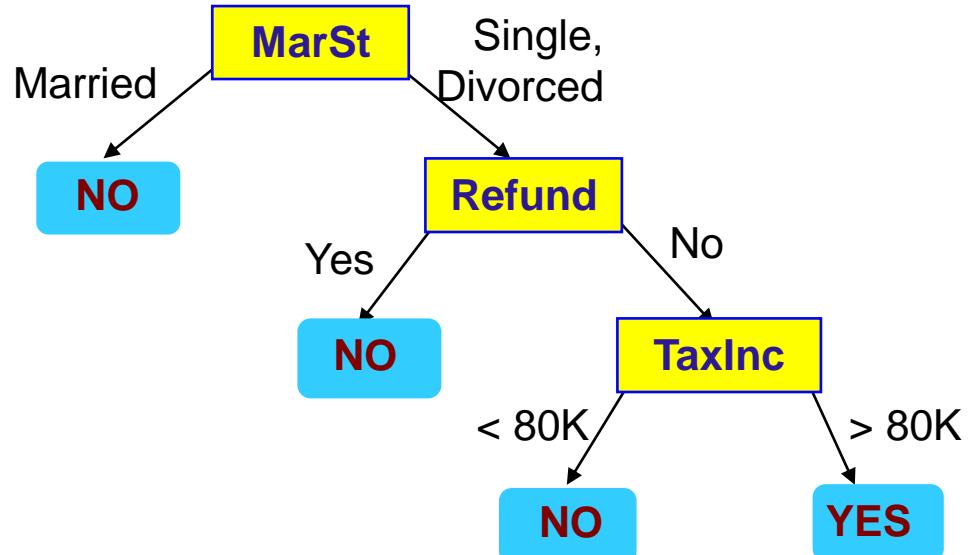
| Tid | Refund | Marital Status | Taxable Income | Cheat |
|-----|--------|----------------|----------------|-------|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married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

Training Data



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!

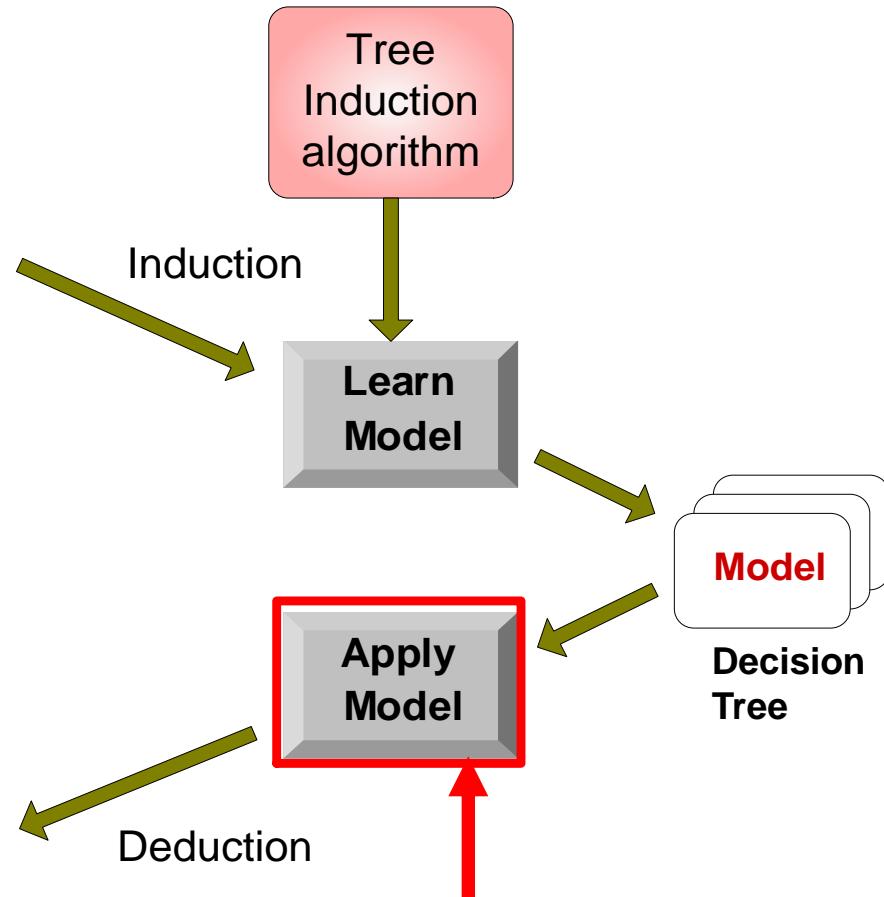
Decision Tree Classification Task

| 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

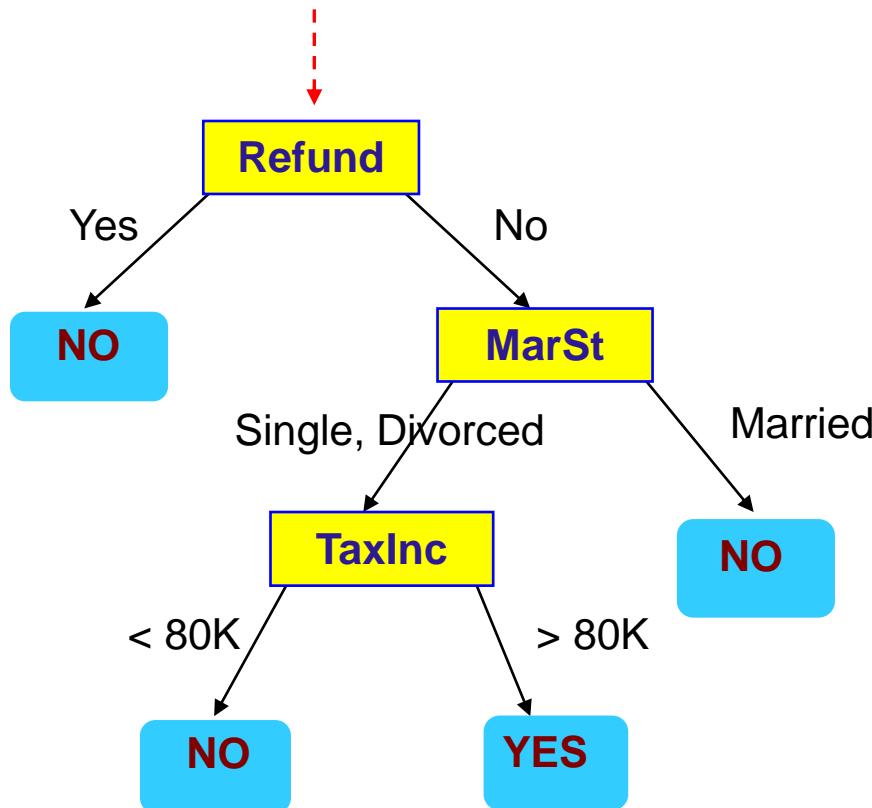
| Tid | Attrib1 | Attrib2 | Attrib3 | Class |
|-----|---------|---------|---------|-------|
| 11 | No | Small | 55K | ? |
| 12 | Yes | Medium | 80K | ? |
| 13 | Yes | Large | 110K | ? |
| 14 | No | Small | 95K | ? |
| 15 | No | Large | 67K | ? |

Test Set



Apply Model to Test Data

Start from the root of tree.



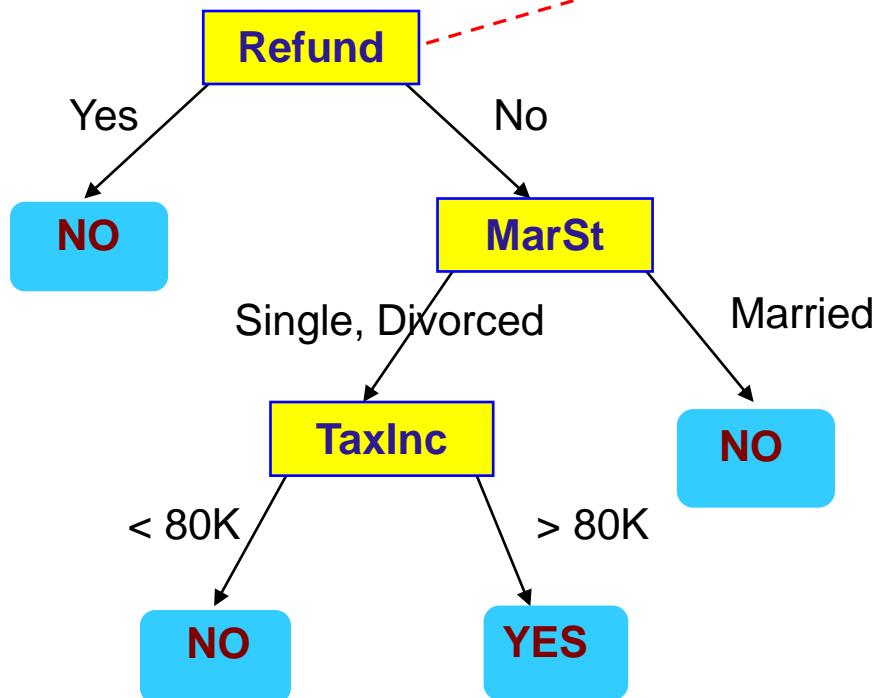
Test Data

| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Married | 80K | ? |

Apply Model to Test Data

Test Data

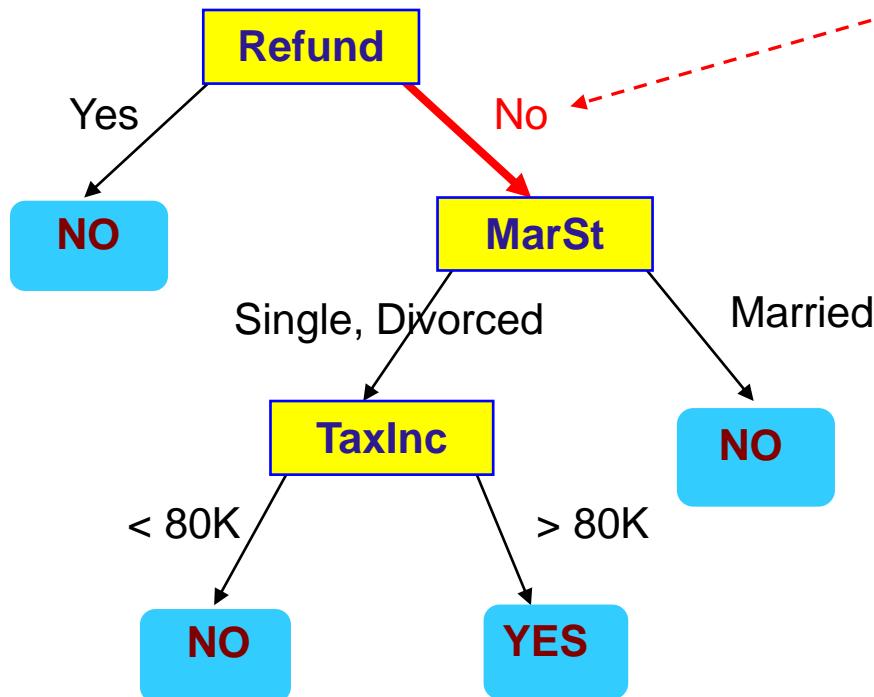
| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Married | 80K | ? |



Apply Model to Test Data

Test Data

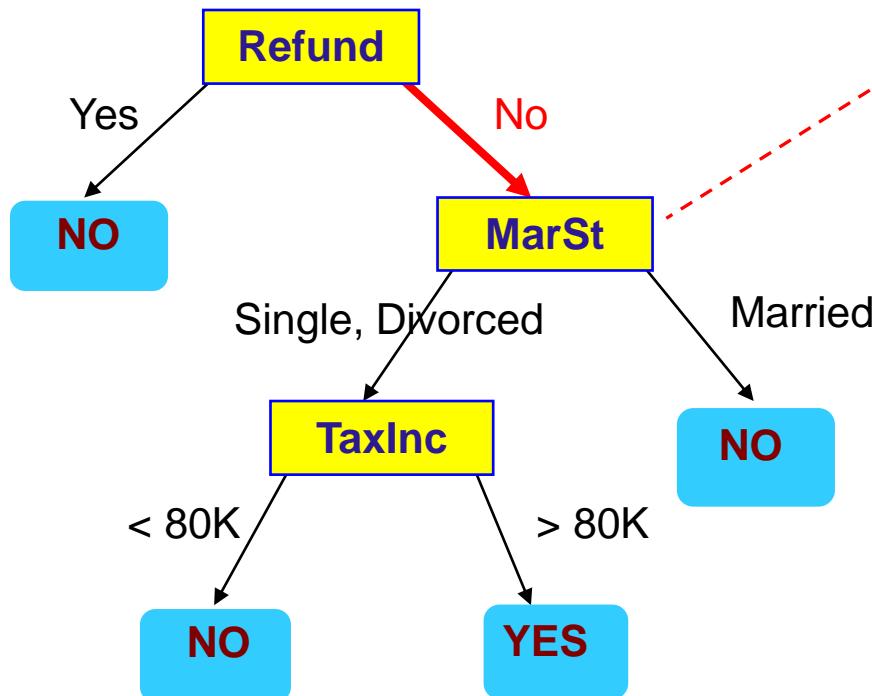
| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Married | 80K | ? |



Apply Model to Test Data

Test Data

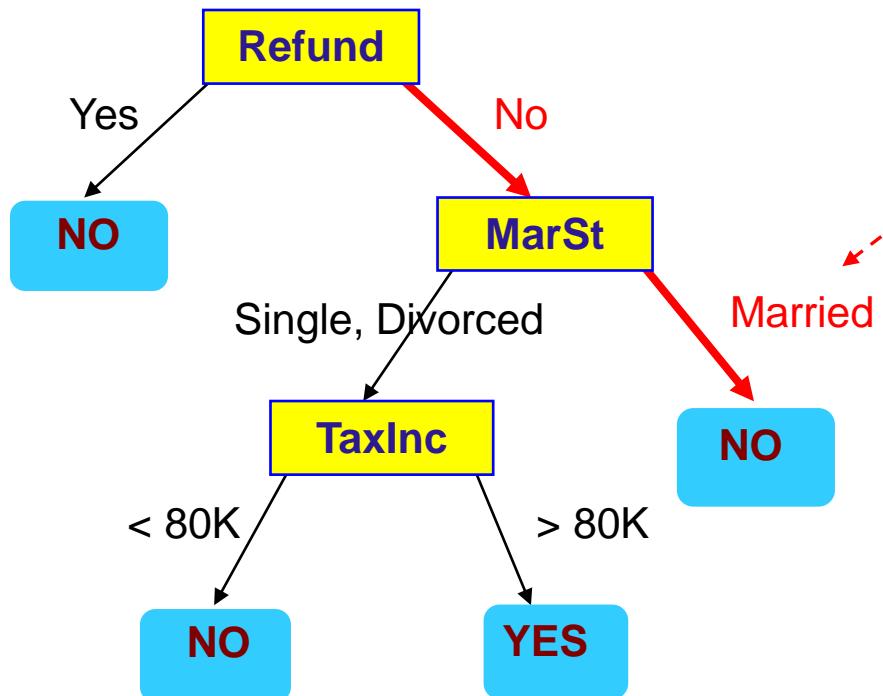
| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Married | 80K | ? |



Apply Model to Test Data

Test Data

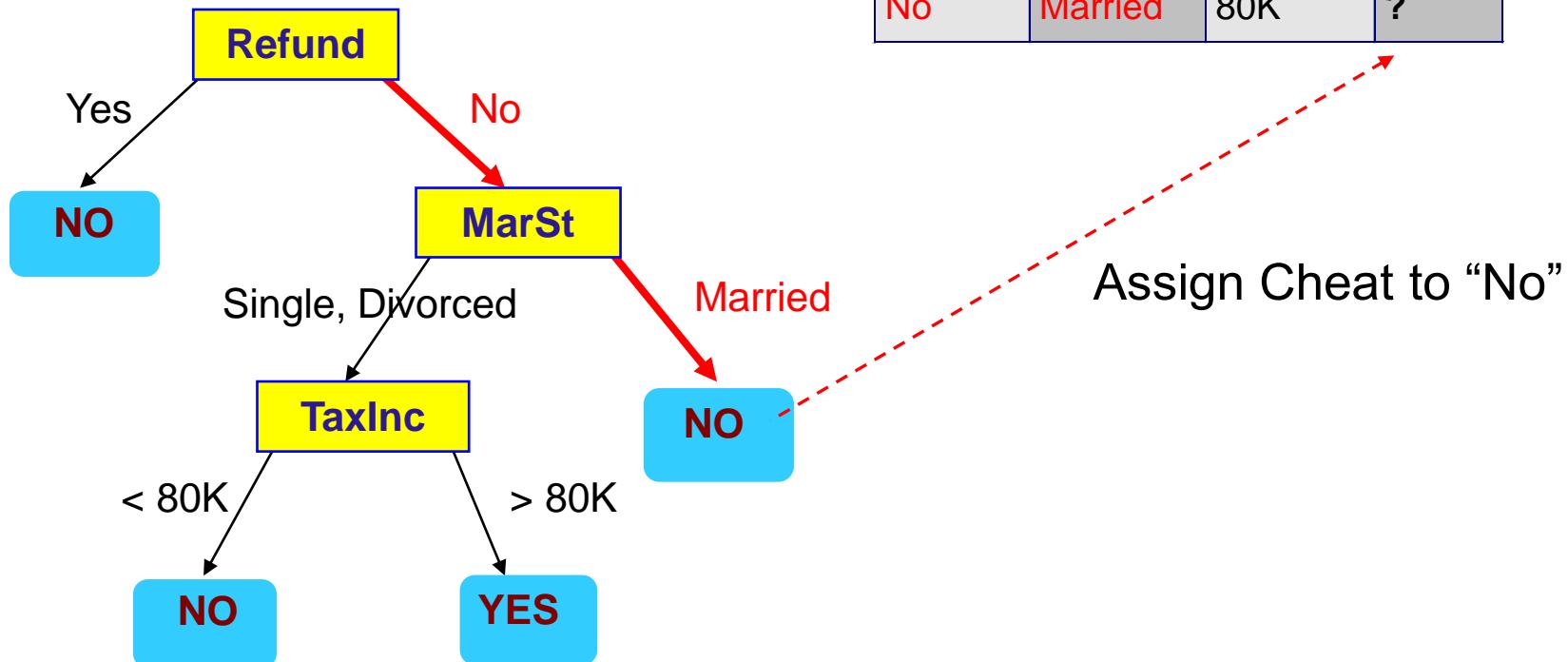
| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Married | 80K | ? |



Apply Model to Test Data

Test Data

| Refund | Marital Status | Taxable Income | Cheat |
|--------|----------------|----------------|-------|
| No | Married | 80K | ? |



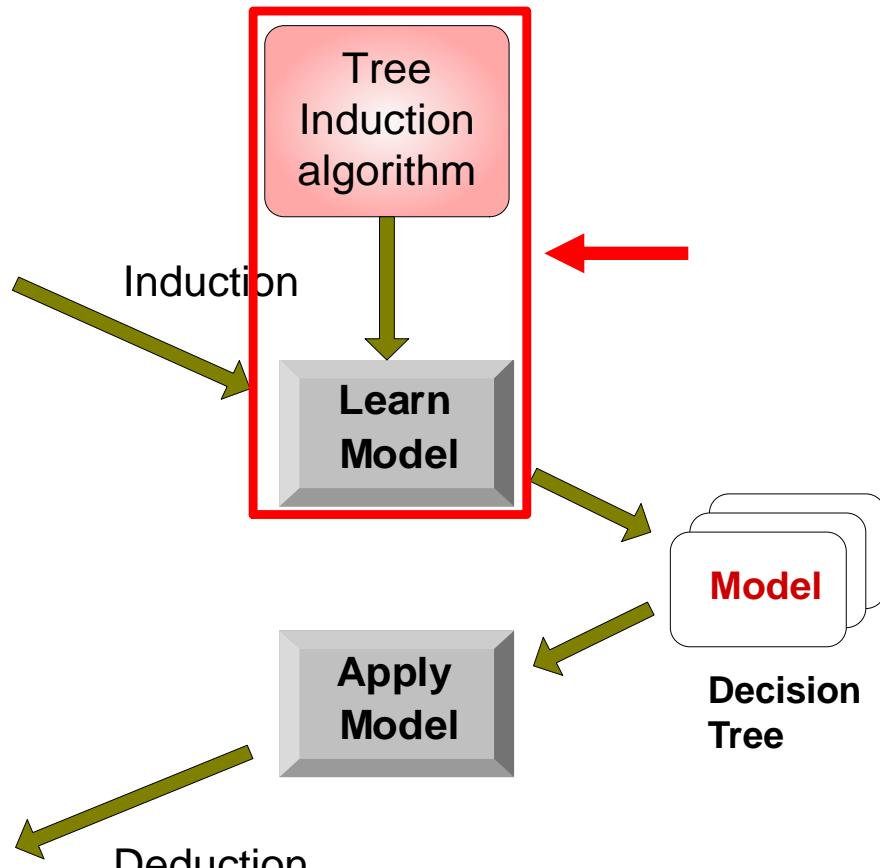
Decision Tree Classification Task

| 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

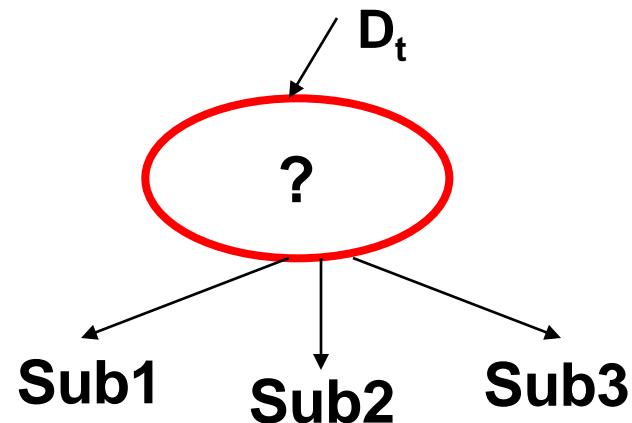
- Many Algorithms:

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

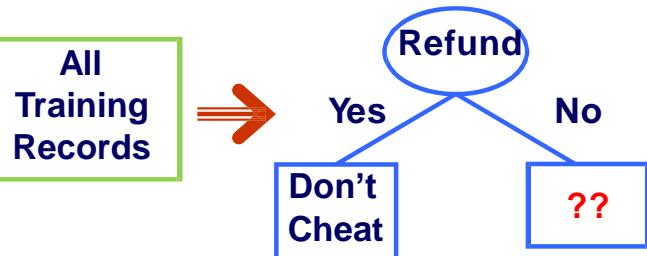
General Structure of Hunt's Algorithm

- Let D_t be the set of training records that are associated with node t and
- $y = \{y_1, y_2, \dots, y_c\}$ be the class labels.
- General Procedure:
 - Generate leaf node or attribute test:
 - if D_t only contains records that belong to the **same class** y_t , then t is a **leaf node** labeled as y_t
 - if D_t contains records that belong to **more than one class**, use an **attribute test** to split the data into subsets having a higher **purity**.
 - for all possible tests: calculate purity of the resulting subsets
 - choose test resulting in highest purity
- Recursively apply this procedure to each subset

| 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 |



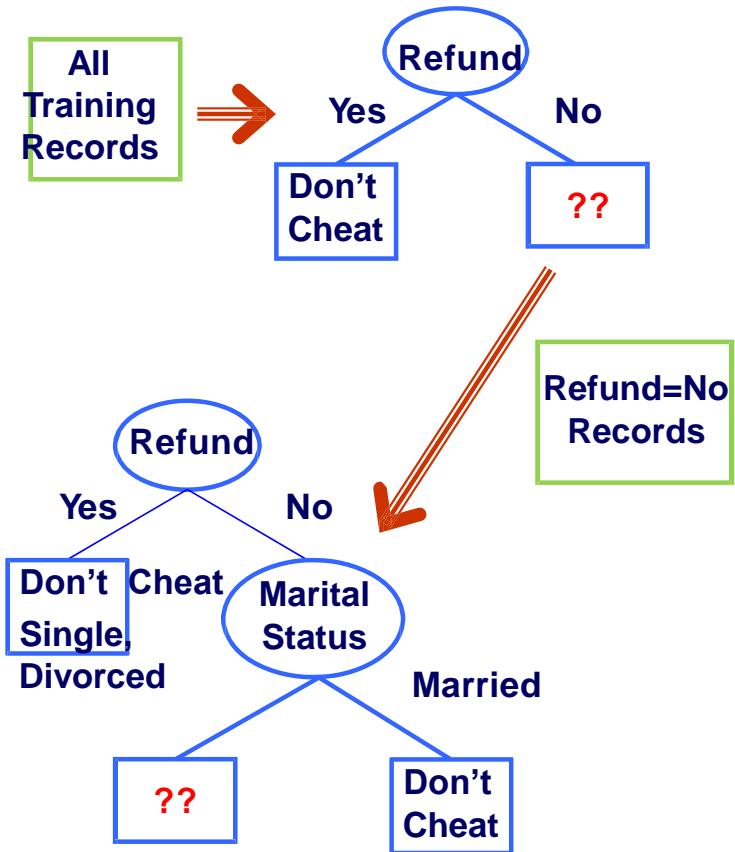
Hunt's Algorithm – Step 1



1. We calculate the purity of the resulting subsets for all possible splits
 - Purity of split on Refund
 - Purity of split on Marital Status
 - Purity of split on Taxable Income
2. We find the split on Refund to produce the purest subsets

| 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 |

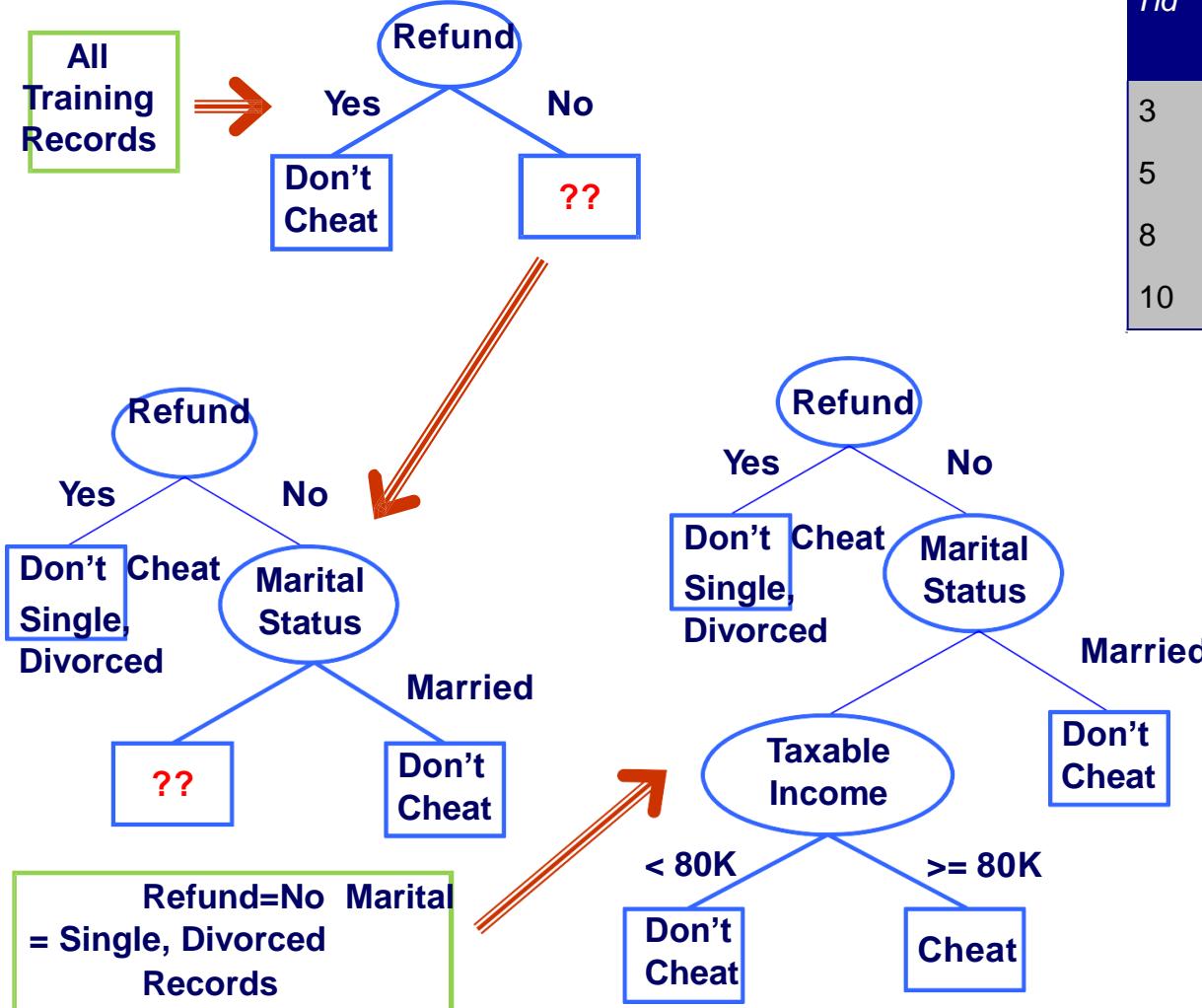
Hunt's Algorithm – Step 2



| Tid | Refund | Marital Status | Taxable Income | Cheat |
|-----|--------|----------------|----------------|-------|
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

1. We further examine the Refund=No records
2. Again, we test all possible splits
3. We find the split on Marital Status to produce the purest subsets

Hunt's Algorithm – Step 3



| Tid | Refund | Marital Status | Taxable Income | Cheat |
|-----|--------|----------------|----------------|-------|
| 3 | No | Single | 70K | No |
| 5 | No | Divorced | 95K | Yes |
| 8 | No | Single | 85K | Yes |
| 10 | No | Single | 90K | Yes |

1. We further examine the Marital Status=Single or =Divorced records
2. We find a split on Taxable Income to produce pure subsets
3. We stop splitting as no sets containing different classes are left

Design Issues of Decision Tree Induction

A learning algorithm for inducing decision trees must address the following two issues.

1. **How should training records be split?**

- **How to specify the attribute test condition?**
 - ⑩ **Depends on number of ways to split: 2-way split, multi-way split**
 - ⑩ **Depends on attribute data type: nominal, ordinal, continuous**
- **How to determine the best split?**
 - ⑩ **Different purity measures can be used**

Each recursive step of the tree-growing process must select an attribute test condition to divide the records into smaller subsets. To implement this step, the algorithm must provide a method for specifying the test condition for different attribute types as well as an objective measure for evaluating the goodness of each test condition.

Design Issues of Decision Tree Induction

2. When should the splitting procedure stop?

- Shallow trees might generalize better to unseen records
- Fully grown trees might overfit training data

A stopping condition is needed to terminate the tree-growing process.

A possible strategy is to continue expanding a node until either all the records belong to the same class or

All the records have identical attribute values.

Although both conditions are sufficient to stop any decision tree induction algorithm, other criteria can be imposed to allow the tree-growing procedure to terminate earlier.

The advantages of early termination will be discussed later.

Tree Induction

- Greedy strategy:
 - Split the records based on an attribute test that optimizes certain criterion.

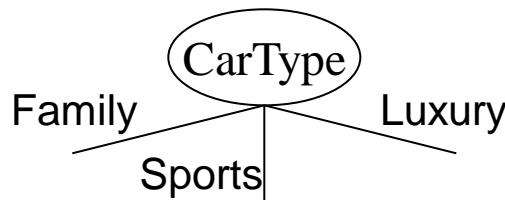
- Issues
 - Determine how to split the records
 - ◆ How to specify the attribute test condition?
 - ◆ How to determine the best split?
 - Determine when to stop splitting

How to Specify Test Condition?

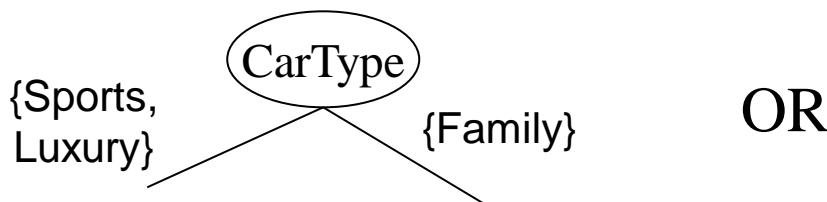
- Depends on attribute types
 - Nominal
 - Ordinal
 - Continuous
- Depends on number of ways to split
 - 2-way split
 - Multi-way split

Splitting Based on Nominal Attributes

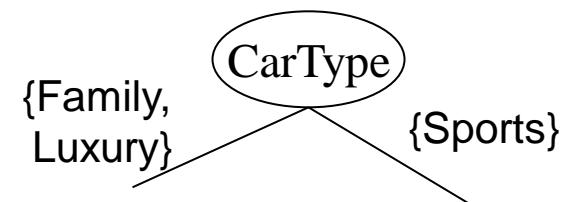
- **Multi-way split:** Use as many partitions as distinct values.



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

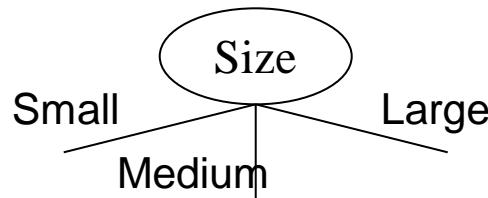


OR

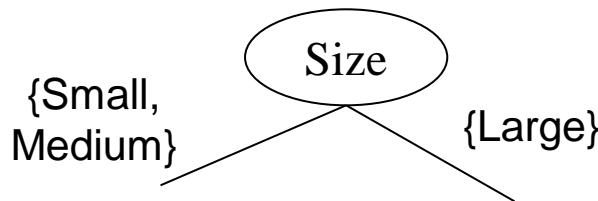


Splitting Based on Ordinal Attributes

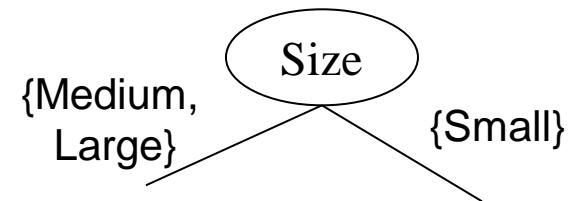
- **Multi-way split:** Use as many partitions as distinct values.



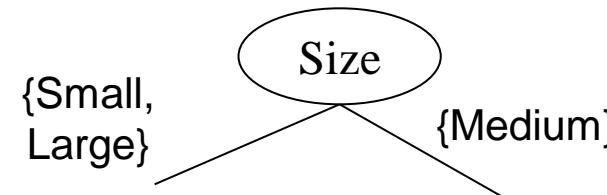
- **Binary split:** Divides values into two subsets.
Need to find optimal partitioning.



OR



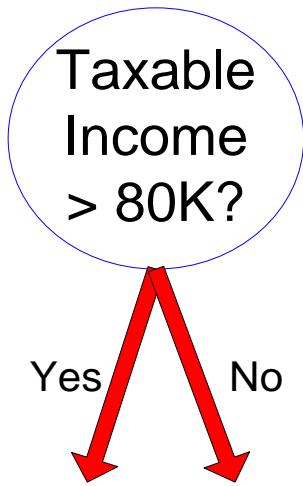
- What about this split?



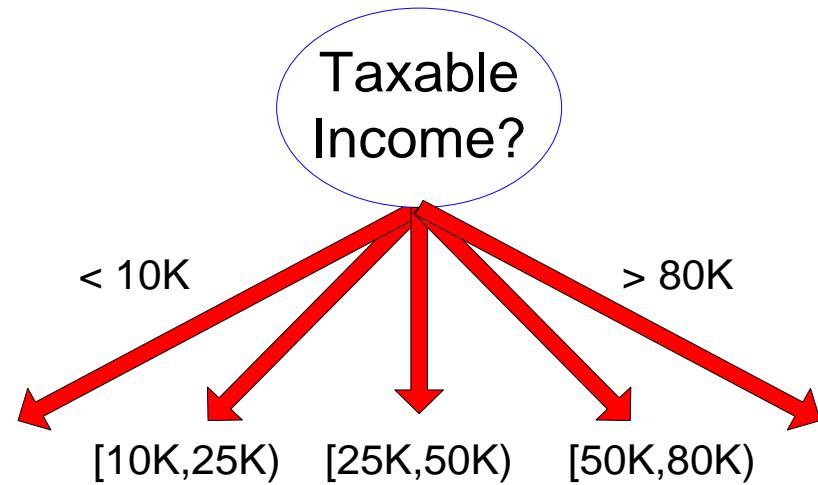
Splitting Based on Continuous Attributes

- Different ways of handling Continuous attributes
 - Discretization to form an ordinal categorical attribute
 - ◆ Static – discretize once at the beginning
 - ◆ Dynamic – ranges can be found by
 - equal interval bucketing/binning,
 - equal frequency bucketing/binning.
 - binning based on user-provided boundaries-percentiles/clustering
 - Binary Decision: $(A < v)$ or $(A \geq v)$
 - ◆ usually sufficient in practice
 - ◆ consider all possible splits and finds the best cut
 - ◆ can be more compute intensive

Splitting Based on Continuous Attributes



(i) Binary split



(ii) Multi-way split

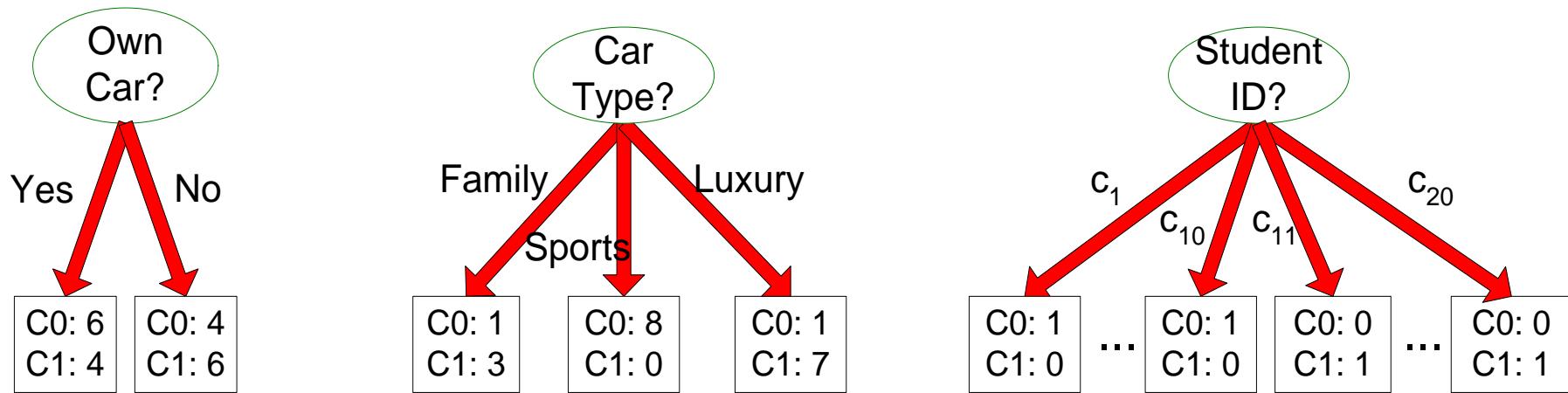
Discretization Example

- Values of the attribute, e.g., age of a person:
 - 0, 4, 12, 16, 16, 18, 24, 26, 28
- Equal-interval binning – for bin width of e.g., 10:
 - Bin 1: 0, 4 [-,10) bin
 - Bin 2: 12, 16, 16, 18 [10,20) bin
 - Bin 3: 24, 26, 28 [20,+) bin
 - denote negative infinity, + positive infinity
- Equal-frequency binning – for bin density of e.g., 3:
 - Bin 1: 0, 4, 12 [-, 14) bin
 - Bin 2: 16, 16, 18 [14, 21) bin
 - Bin 3: 24, 26, 28 [21,+] bin

How to determine the Best Split

In a two-class problem, the class distribution at any node is given by (p_0, p_1) , where, $p_1 = 1 - p_0$.

Before Splitting the dataset contains: **10 records of class 0,**
10 records of class 1



Which test condition is the best?

The degree of split is based on the degree of impurity of the child nodes.

Smaller degree of impurity: more skewed class distribution

Class distribution $(0, 1)$: zero impurity; Class distribution $(0.5, 0.5)$: highest impurity

How to determine the Best Split

- Greedy approach: Test all possible splits and use the one that results in the most **homogeneous (= pure)** nodes
 - Nodes with **homogeneous** class distribution are preferred
- Need a measure of **node impurity**:

| |
|-------|
| C0: 5 |
| C1: 5 |

Non-homogeneous,
High degree of impurity

| |
|-------|
| C0: 9 |
| C1: 1 |

Homogeneous,
Low degree of impurity

Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification error

How to Find the Best Split?

1. Compute impurity measure (P) before splitting
2. Compute impurity measure (M) after splitting for all possible splits
 - compute impurity measure of each child node
 - M is the weighted impurity of children
3. Choose the attribute test condition (split) that produces the highest purity gain

$$\text{Gain} = P - M$$

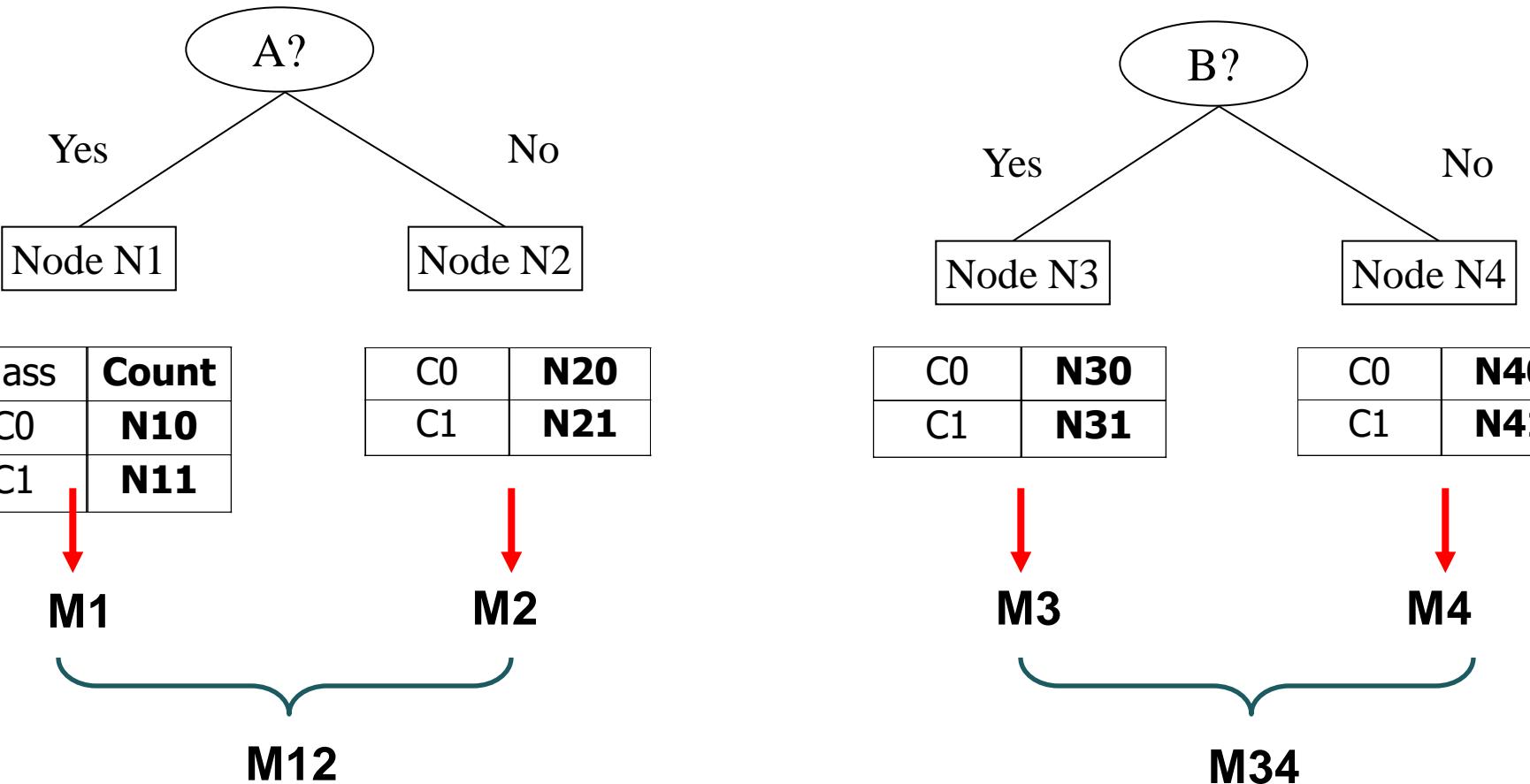
or equivalently, lowest impurity measure after splitting (M)

How to Find the Best Split

Before Splitting:

| | |
|----|-----|
| C0 | N00 |
| C1 | N01 |

→ P



Measure of Impurity: GINI

- Gini Index for a given node t :

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

(NOTE: $p(j | t)$ is the relative frequency of class “j” at node “t”).

- Maximum ($1 - 1/n_c$) : 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

| Class | Count |
|-------------------|-------|
| 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

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

| | |
|----|---|
| C1 | 0 |
| C2 | 6 |

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Gini} = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

| | |
|----|---|
| C1 | 1 |
| C2 | 5 |

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

| | |
|----|---|
| C1 | 2 |
| C2 | 4 |

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Gini} = 1 - (2/6)^2 - (4/6)^2 = 0.444$$

Splitting Based on GINI

- Used in Classification And Regression Tree (CART)
- Supervised Learning In Quest (SLIQ)
- Scalable Parallel Classifier (SPRINT)

When a node p is split into k partitions (children/subsets), the GINI index of each partition is weighted according to the partition's size

- The quality of split is computed as:

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

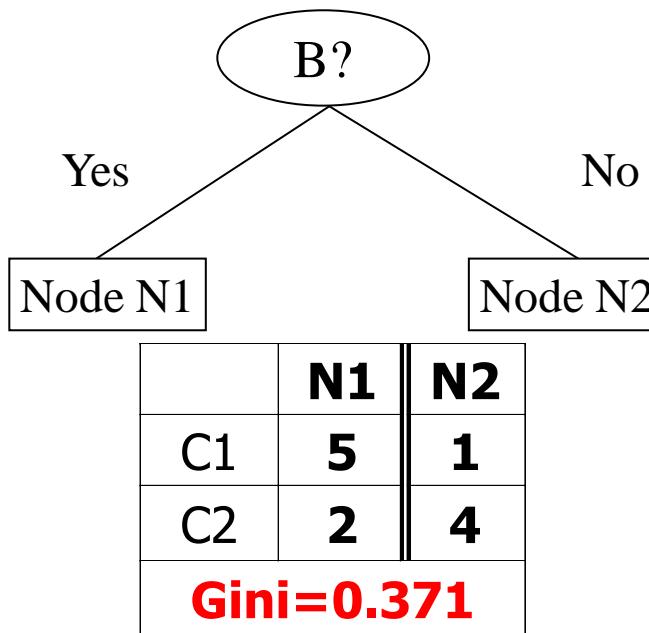
where, n_i = number of records at child i ,
 n = number of records at node p .

Binary Attributes: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
 - Larger and **Purer Partitions** are sought for.

$$\begin{aligned}\text{Gini}(N1) &= 1 - (5/7)^2 - (2/7)^2 \\ &= 0.408\end{aligned}$$

$$\begin{aligned}\text{Gini}(N2) &= 1 - (1/5)^2 - (4/5)^2 \\ &= 0.32\end{aligned}$$



| | Parent |
|---------------------|--------|
| C1 | 6 |
| C2 | 6 |
| Gini = 0.500 | |

$$\begin{aligned}\text{Gini(Children)} &= 7/12 * 0.408 + \\ &\quad 5/12 * 0.32 \\ &= 0.371\end{aligned}$$

– Purity Gain = $0.5 - 0.371 = 0.129$

Categorical Attributes: Computing Gini Index

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

Multi-way split

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

Two-way split

(find best partition of values)

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

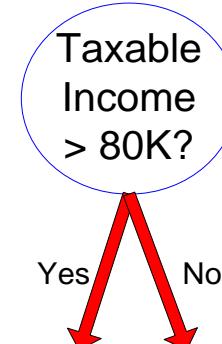
| | CarType | |
|------|--------------|------------------|
| | {Sports} | {Family, Luxury} |
| C1 | 2 | 2 |
| C2 | 1 | 5 |
| Gini | 0.419 | |

The Multi-way split has smaller GINI index

Continuous Attributes: Computing Gini Index

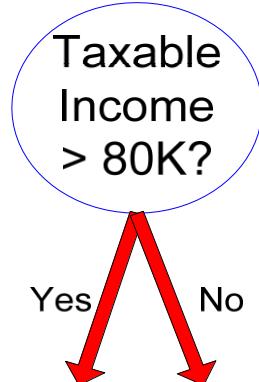
- Use Binary Decisions based on one value
- Several Choices for the splitting value
 - Number of possible splitting values = Number of distinct values
- Each splitting value has a count matrix associated with it
 - Class counts in each of the partitions, $A < v$ and $A \geq v$
- Simple method to choose best v
 - For each v , scan the database to gather count matrix and compute its Gini index
 - Computationally Inefficient!
Repetition of work.

| 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 |



Continuous Attributes: Computing Gini Index...

- How to find best binary split for a continuous attribute?
- For efficient computation: for each attribute,
 - Sort the attribute on values (Annual income $\leq v$ is used to split the training records)
 - Linearly scan these values, each time updating the count matrix and computing gini index
 - Choose the split position that has the least gini index



| Cheat | No | No | No | Yes | Yes | Yes | No | No | No | No | |
|-------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Taxable Income | | | | | | | | | | | |
| Sorted Values → | 60 | 70 | 75 | 85 | 90 | 95 | 100 | 120 | 125 | 220 | |
| Split Positions → | 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 |

Alternative Splitting Criteria based on INFO

- Entropy at a given node t:

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

(NOTE: $p(j | t)$ is the relative frequency of class j at node t).

- Measures homogeneity of a node.
 - ◆ Maximum ($\log n_c$) : 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

Examples for computing Entropy

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

| | |
|----|---|
| C1 | 0 |
| C2 | 6 |

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Entropy} = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

| | |
|----|---|
| C1 | 1 |
| C2 | 5 |

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Entropy} = -(1/6) \log_2 (1/6) - (5/6) \log_2 (1/6) = 0.65$$

| | |
|----|---|
| C1 | 2 |
| C2 | 4 |

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Entropy} = -(2/6) \log_2 (2/6) - (4/6) \log_2 (4/6) = 0.92$$

Splitting Based on INFO...

□ 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;

n_i 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)
- Used in ID3 and C4.5
- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

Splitting Based on INFO...

□ Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

Parent Node, p is split into k partitions;

n_i is number of records in partition i

- Gain RATIO is designed to overcome the tendency to generate a large number of small partitions (disadvantage of Information Gain)
- Gain RATIO Adjusts Information Gain by the entropy of the partitioning (SplitINFO).
- Higher entropy partitioning (large number of small partitions) is penalized!
- Used in C4.5

Splitting Criteria based on Classification Error

- Classification error at a node t :

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

- 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

Examples for Computing Error

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

| | |
|----|---|
| C1 | 0 |
| C2 | 6 |

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$\text{Error} = 1 - \max(0, 1) = 1 - 1 = 0$$

| | |
|----|---|
| C1 | 1 |
| C2 | 5 |

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

$$\text{Error} = 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$$

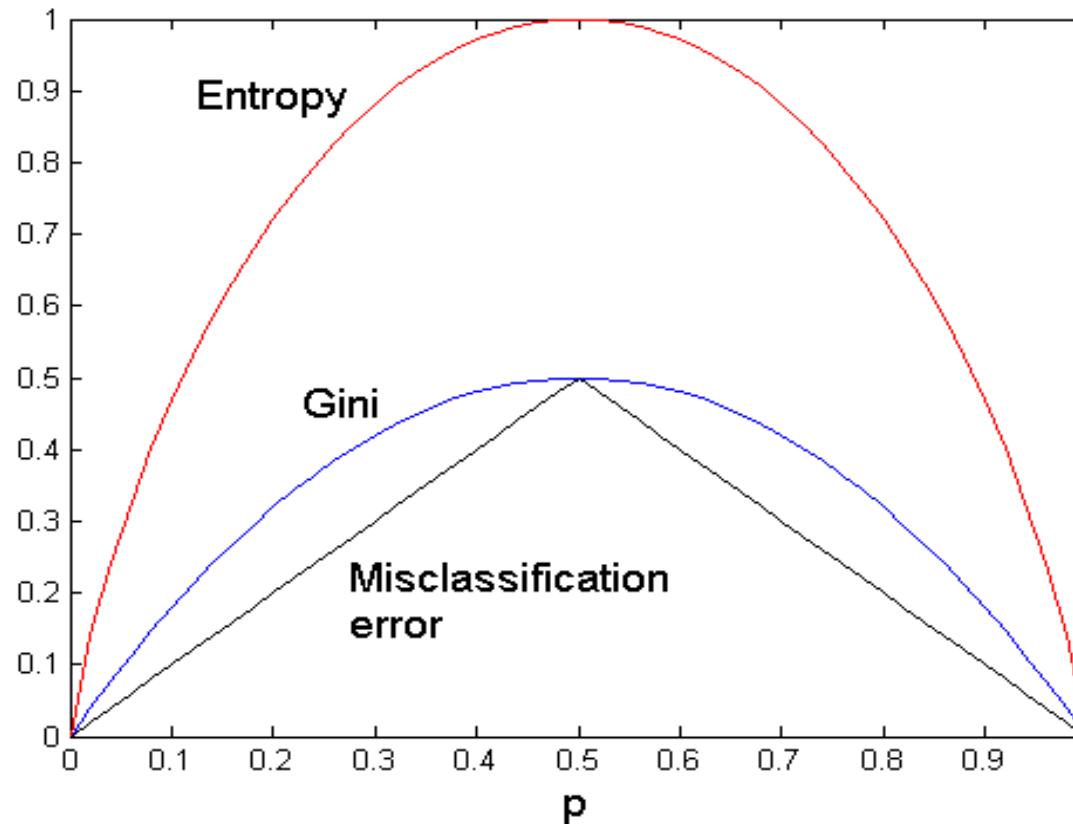
| | |
|----|---|
| C1 | 2 |
| C2 | 4 |

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

$$\text{Error} = 1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3$$

Comparison among Splitting Criteria

For a 2-class problem:

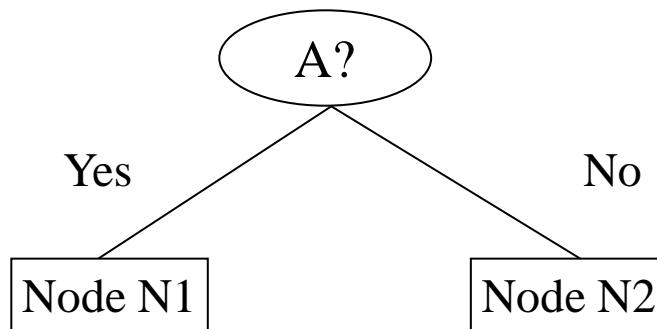


P : fraction of records that belong to one of the two classes.

P = 0.5: Three measures attain max value: uniform class distribution

P = 0 / 1: Three measures attain min value : all records belong to same class

Misclassification Error vs Gini



| | Parent |
|--------------------|---------------|
| C1 | 7 |
| C2 | 3 |
| Gini = 0.42 | |

Gini(N1)

$$= 1 - (3/3)^2 - (0/3)^2$$

$$= 0$$

Gini(N2)

$$= 1 - (4/7)^2 - (3/7)^2$$

$$= 0.489$$

| | N1 | N2 |
|-------------------|-----------|-----------|
| C1 | 3 | 4 |
| C2 | 0 | 3 |
| Gini=0.361 | | |

Gini(Children)

$$= 3/10 * 0$$

$$+ 7/10 * 0.489$$

$$= 0.342$$

Gini improves !!

But Error increases!!

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

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
- Early termination (to be discussed later)

Practical Issues of Classification

- Underfitting and Overfitting
- Missing Values
- Costs of Classification

Other Issues

- Data Fragmentation
- Search Strategy
- Expressiveness
- Tree Replication

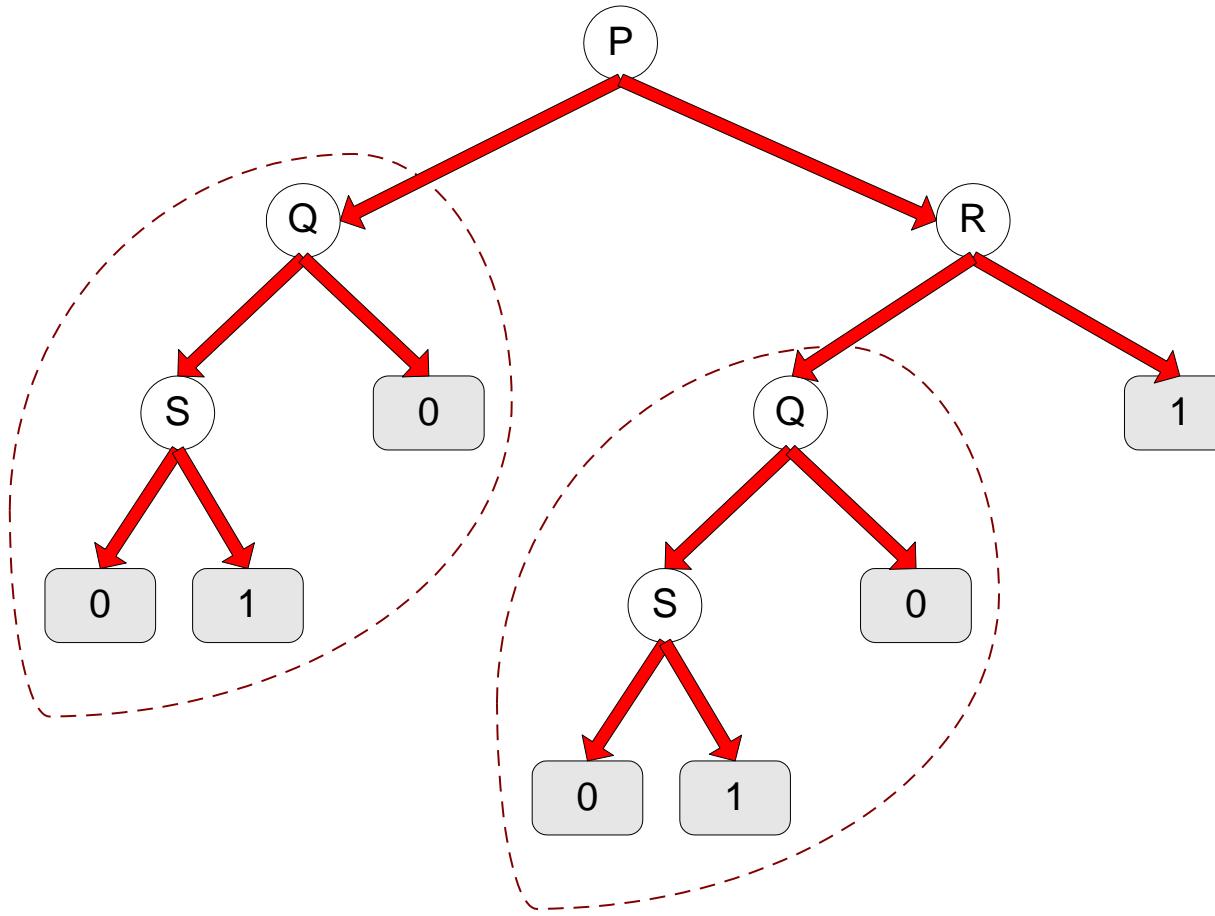
Data Fragmentation

- Number of instances gets smaller as you traverse down the tree
- Number of instances at the leaf nodes could be too small to make any statistically significant decision

Search Strategy

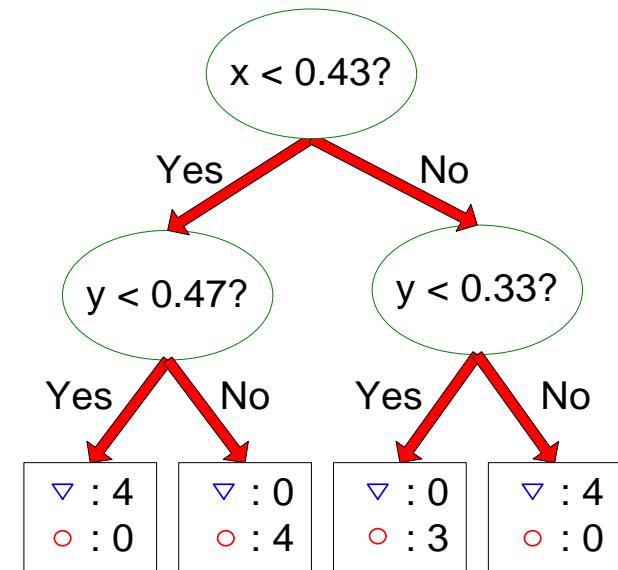
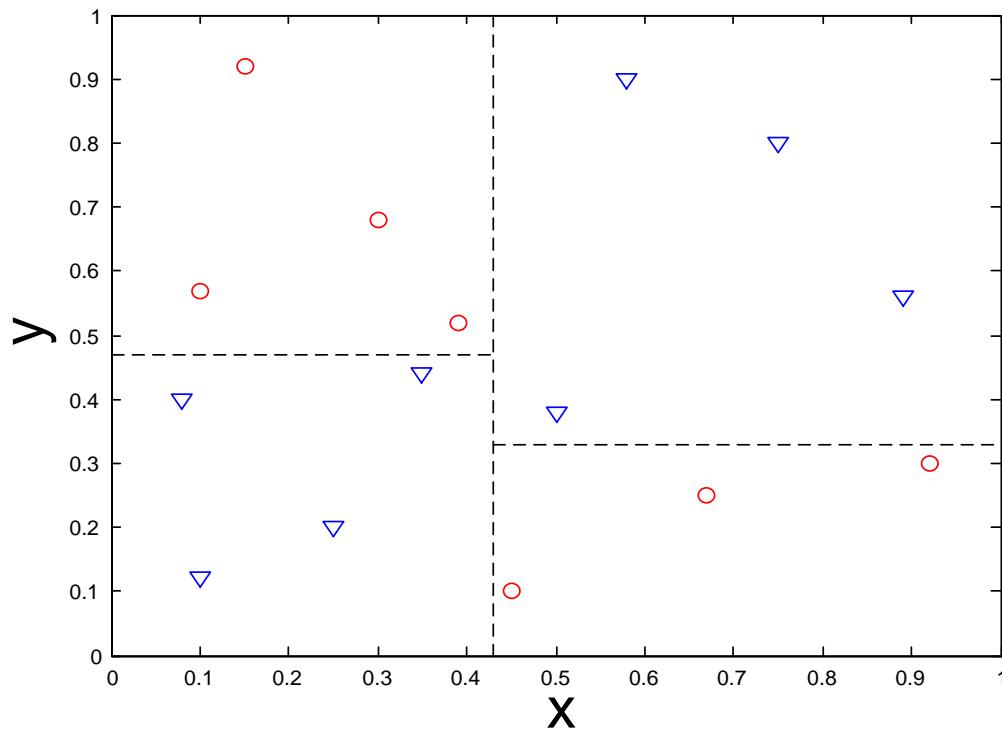
- Finding an optimal decision tree is NP-hard
- The algorithm presented so far uses a greedy,
top-down, recursive partitioning strategy to induce a reasonable solution
- Other strategies?
 - Bottom-up
 - Bi-directional

Tree Replication



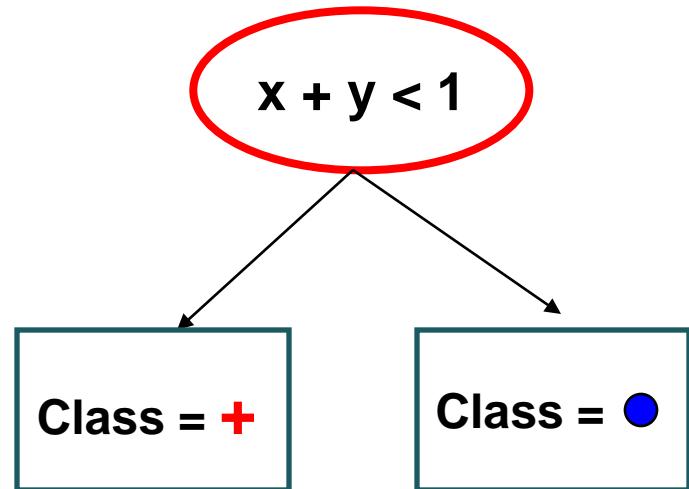
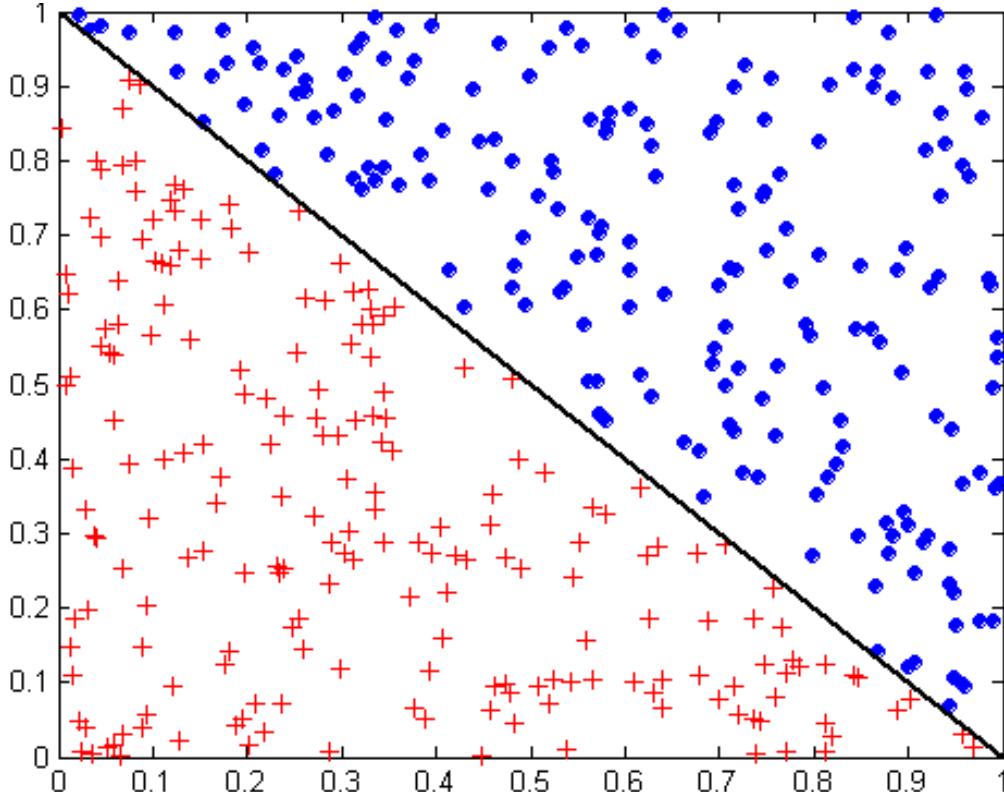
- Same subtree appears in multiple branches

Decision Boundary



- Border line between two neighboring regions of different classes is known as **decision boundary**
- Decision boundary is parallel to axes because test condition involves a single attribute at-a-time

Oblique Decision Trees



- Test condition may involve multiple attributes
- More expressive representation
- Finding optimal test condition is computationally expensive

Discussion on Decision Trees

Advantages:

- Inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to interpret by humans for small-sized trees (eager learning)
- Can easily handle redundant or irrelevant attributes
- Accuracy is comparable to other classification techniques for many low dimensional data sets (not texts and images)

Disadvantages:

- Space of possible decision tree is exponentially large
- Greedy approaches are often unable to find the best tree
- Trees do not take into account interactions between attributes.

Overfitting

- We want to learn models that are good at classifying **unseen records**
- **Overfitting:** Learned models can fit the training data too closely and thus work poorly on unseen data
- Model perfectly fitting the training data:

"Trees are **big**, **green** **plants** that have a **trunk** and **no wheels**"

- Unseen example:

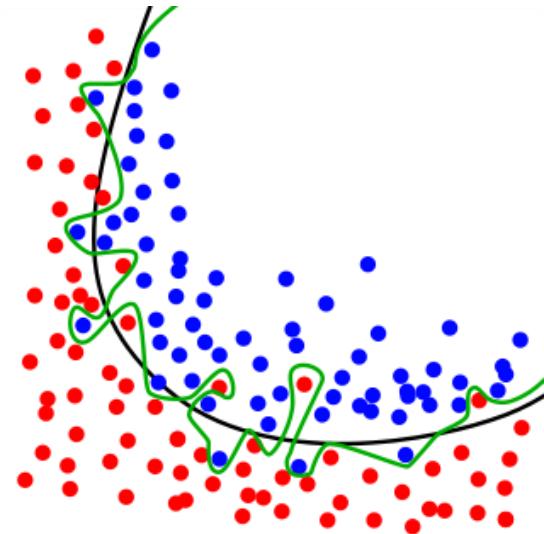
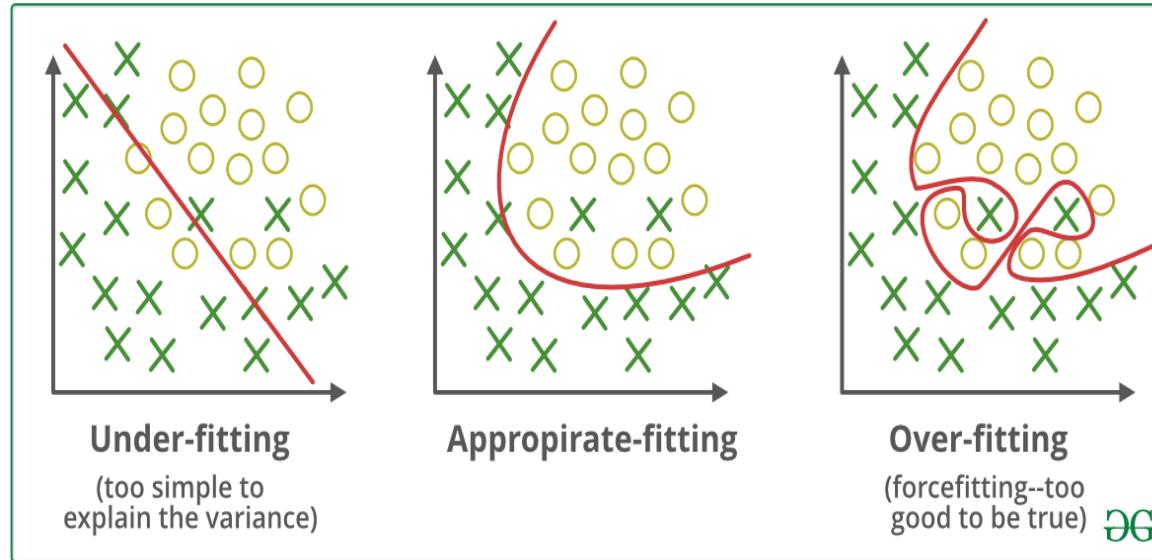


Training data



- **Goal:** Find good compromise between specificity and generality of the learned model

Underfitting and Overfitting



*The black line fits the data well,
the green line is overfitting.*

The cause of the poor performance of a model in machine learning is either overfitting or underfitting the data.

Generalization in Data Mining /Machine Learning: refers to how well the concepts learned by a learning model generalizes to specific examples or data not yet seen by the model.

Overfitting vs Underfitting

Underfitting:

- Occurs when a model is too simple.
- Informed by too few features or regularized too much - which makes it inflexible in learning from the dataset.
- Simple learners tend to have less variance in their predictions but more bias towards wrong outcomes.
- On the other hand, complex learners tend to have more variance in their predictions.
- Underfitted models are like those Engineers who wanted to be cricketers but forced by their parents to take up engineering.
- They will neither know engineering nor cricket pretty well. They never had their heart in what they did and have insufficient knowledge of everything.
- In terms of machine learning, we can state them as too little focus on the training set. Neither good for training not testing.

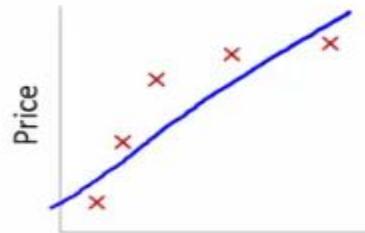
Underfitting:

How to detect underfitting?

A model under fits when it is too simple with regards to the data it is trying to model.

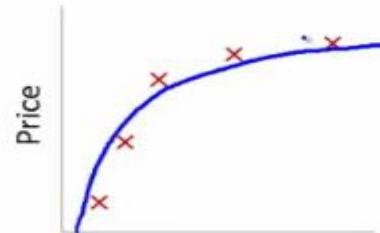
One way to detect such a situation is to use the bias-variance approach.

Model is *under fitted* when you have a *high bias*.

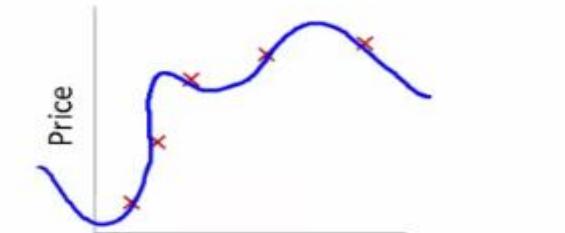


$$\theta_0 + \theta_1 x$$

High bias
(underfit)



"Just right"



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

High variance
(overfit)

How to avoid underfitting :

More data will not generally help. It will, in fact, likely increase the training error.

Therefore **we should increase more features**. Because that expands the hypothesis space. This **includes making new features from existing features**. Same way more parameters may also expand the hypothesis space.

Overfitting:

Overfitted models are like subject matter experts

They know a lot about a particular field for example subject matter experts.

You ask them anything about the functionality of their tool (even in details), they'll probably be able to answer you and that too pretty precisely.

But when you ask them why the oil price fluctuate, they'll probably make an informed guess and say something peculiar.

In terms of machine learning, we can state them as too much focus on the training set (programmers) and learns complex relations which may not be valid in general for new data (test set).

Overfitting:

How to detect Overfitting?

A key challenge with overfitting, and with machine learning in general, is that we can't know how well our model will perform on new data until we actually test it.

To address this, we can split our initial dataset into separate *training* and *test subsets*. This method can approximate how well our model will perform on new data.

If our model does much better on the training set than on the test set, then we're likely overfitting.

For example, it would be a big red flag if our model saw 95% accuracy on the training set but only 48% accuracy on the test set.

How to Prevent Overfitting:

Detecting overfitting is useful, but it doesn't solve the problem. Fortunately, you have several options to try.

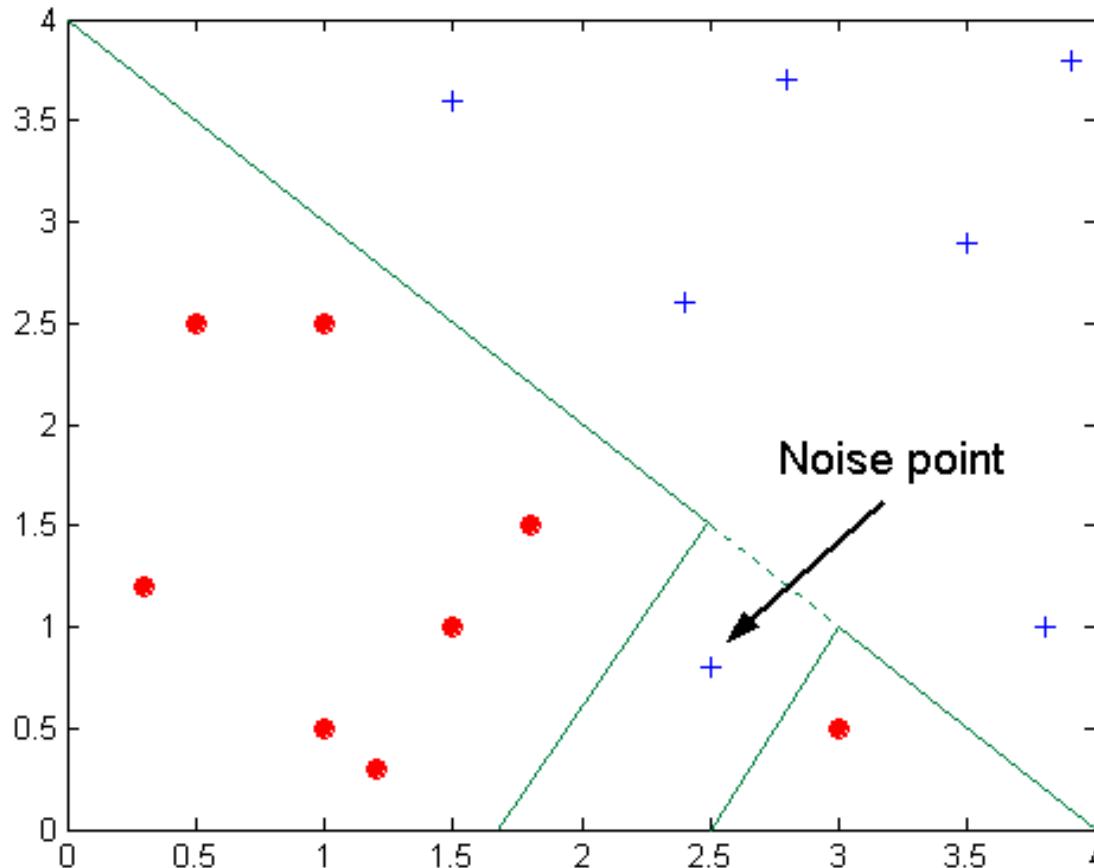
Here are a few of the most popular solutions for overfitting:

Cross Validation, Early Stopping, Pruning, Regularization, Remove Feature, Train with more data and Ensemble

Reasons for Model Overfitting

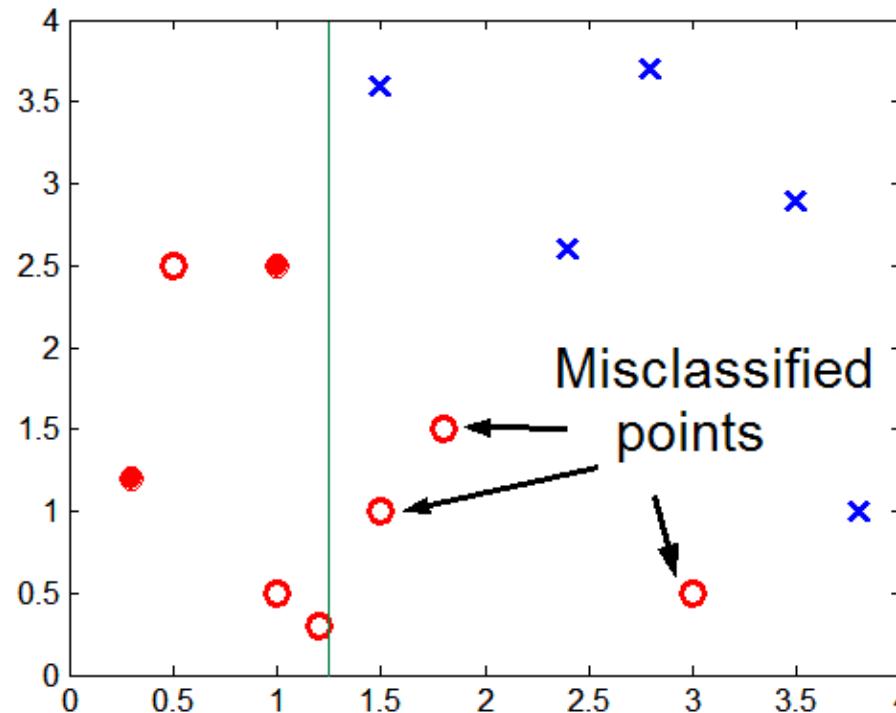
- Limited Training Size
- High Model Complexity
 - Multiple Comparison Procedure

Overfitting due to Presence of Noise



Decision boundary is distorted by noise point

Overfitting due to Insufficient Examples, Lack of Representative samples



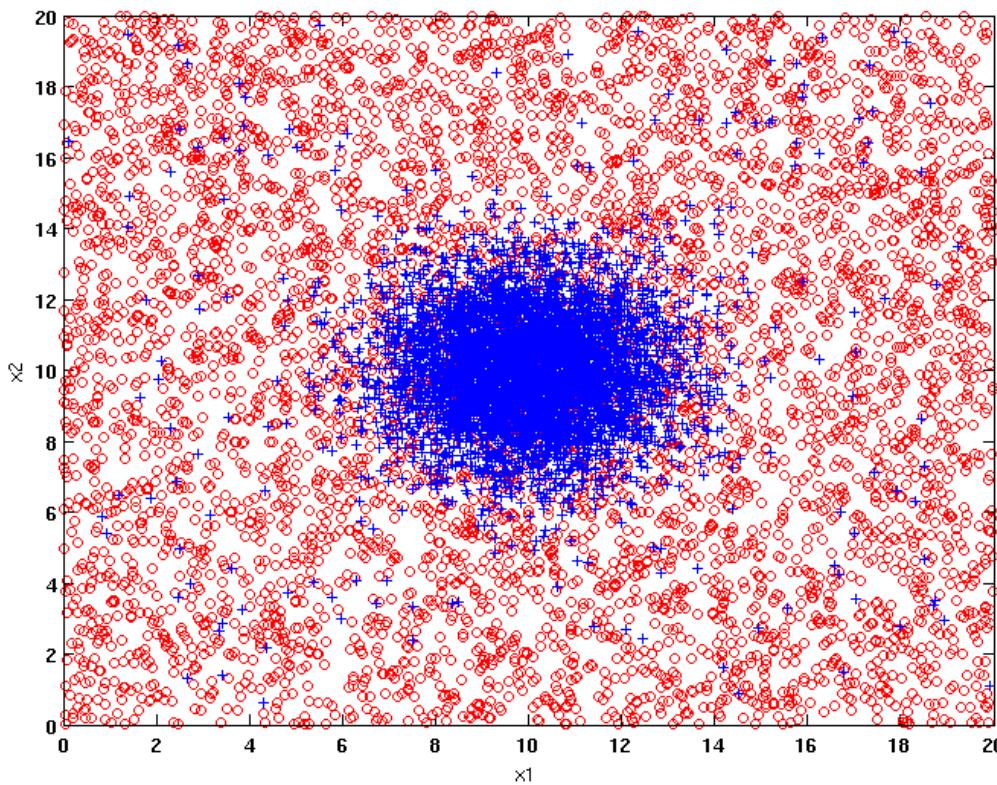
Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels of that region

Insufficient number of training records in the region causes the decision tree to predict the test examples using other training records that are irrelevant to the classification task

Notes on Overfitting

- Overfitting results in decision trees that are more complex than necessary
- Training error does not provide a good estimate of how well the tree will perform on previously unseen records
- Need ways for estimating generalization errors

Example Data Set



Two class problem:

+ : 5200 instances

- 5000 instances generated from a Gaussian centered at (10,10)

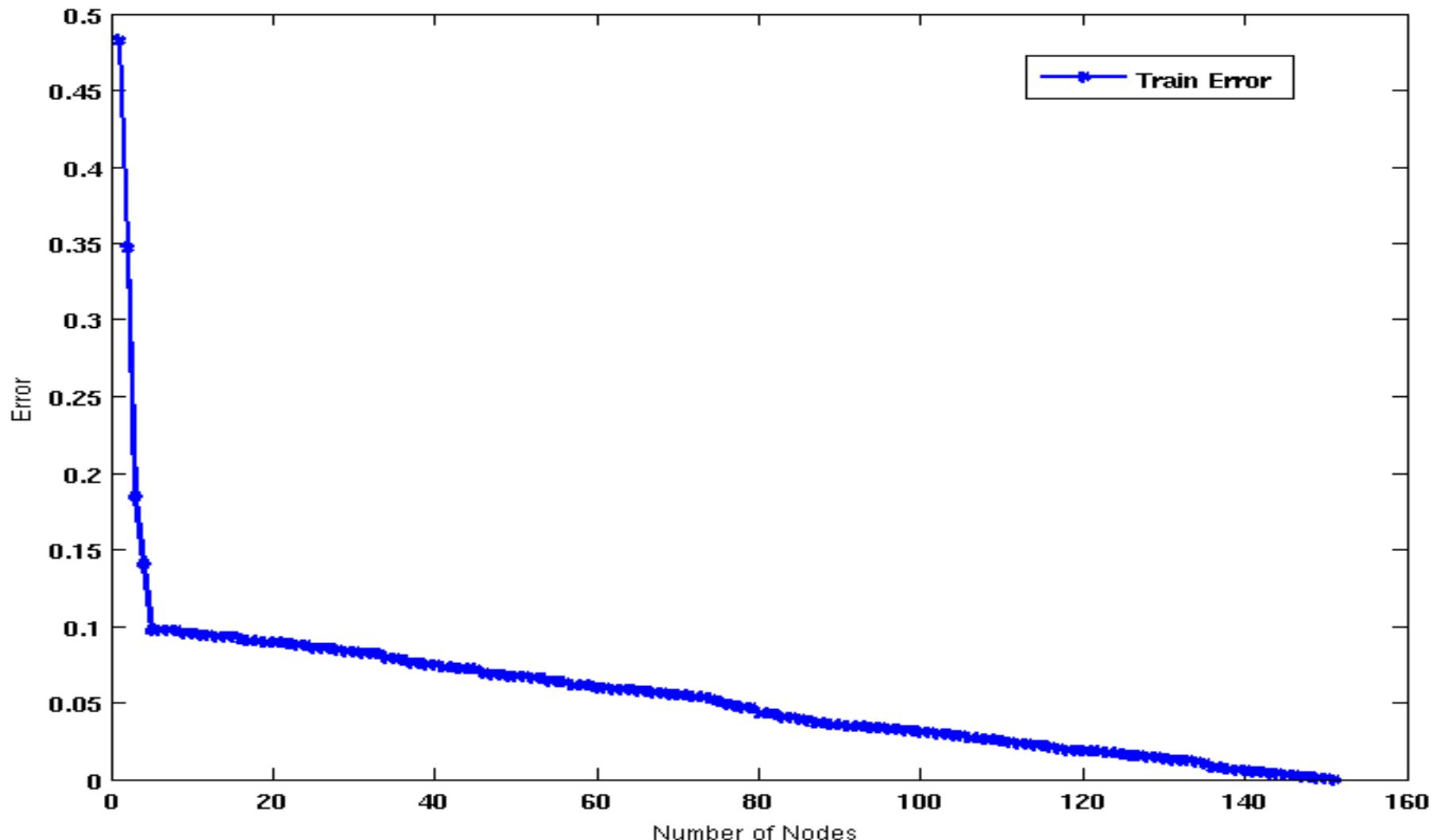
- 200 noisy instances added

o : 5200 instances

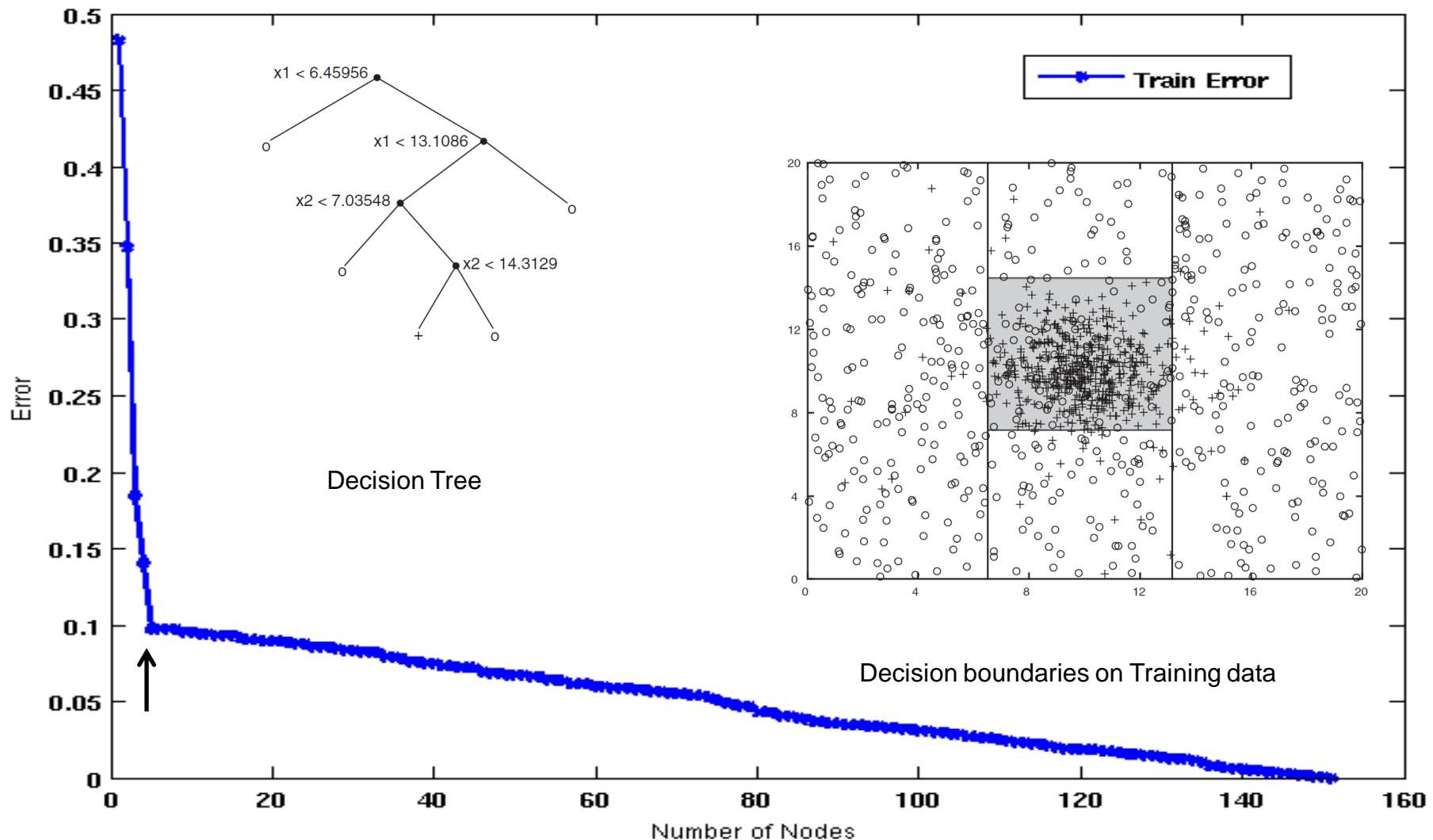
- Generated from a uniform distribution

10 % of the data used for training and 90% of the data used for testing

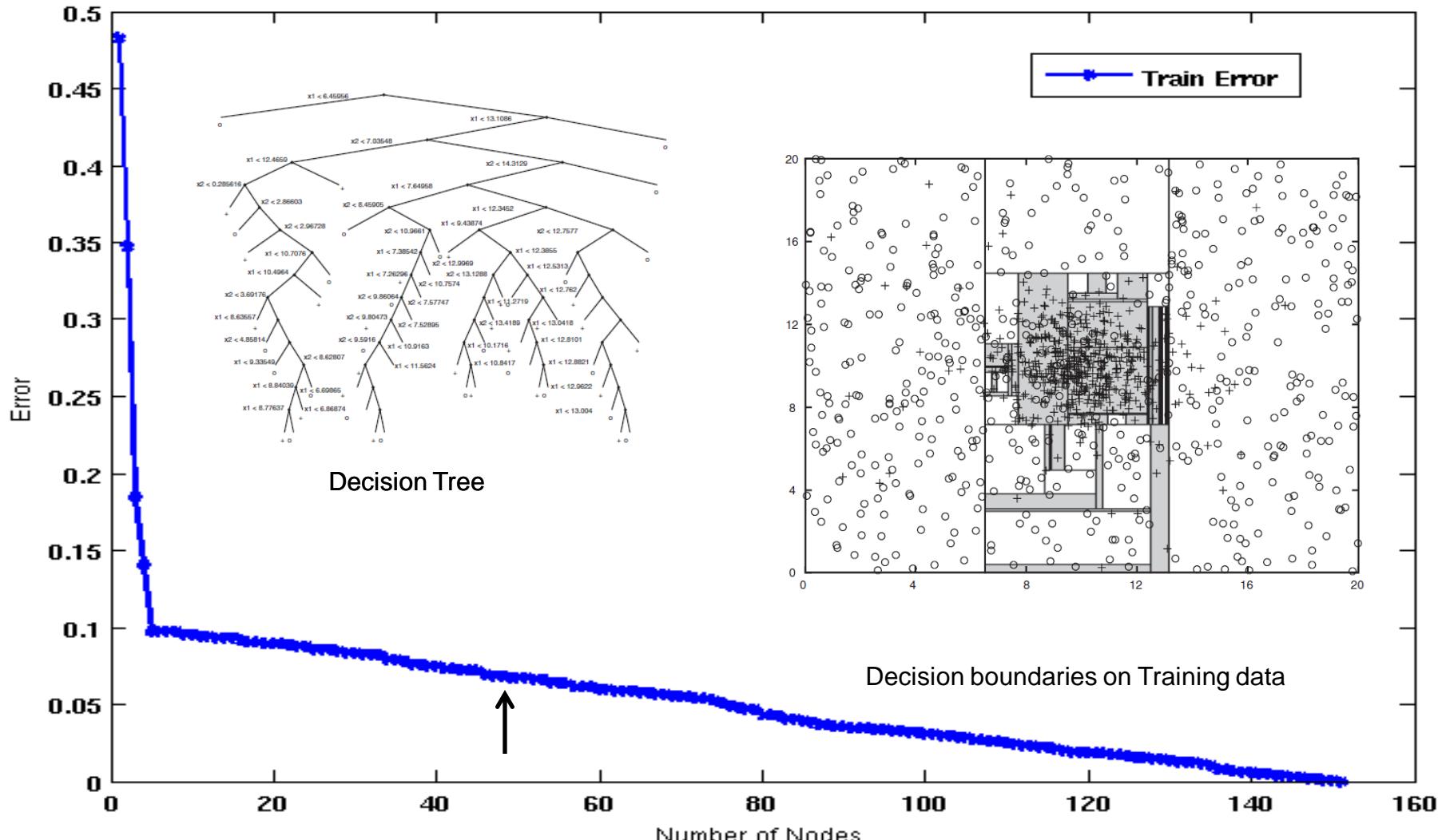
Increasing number of nodes in Decision Trees



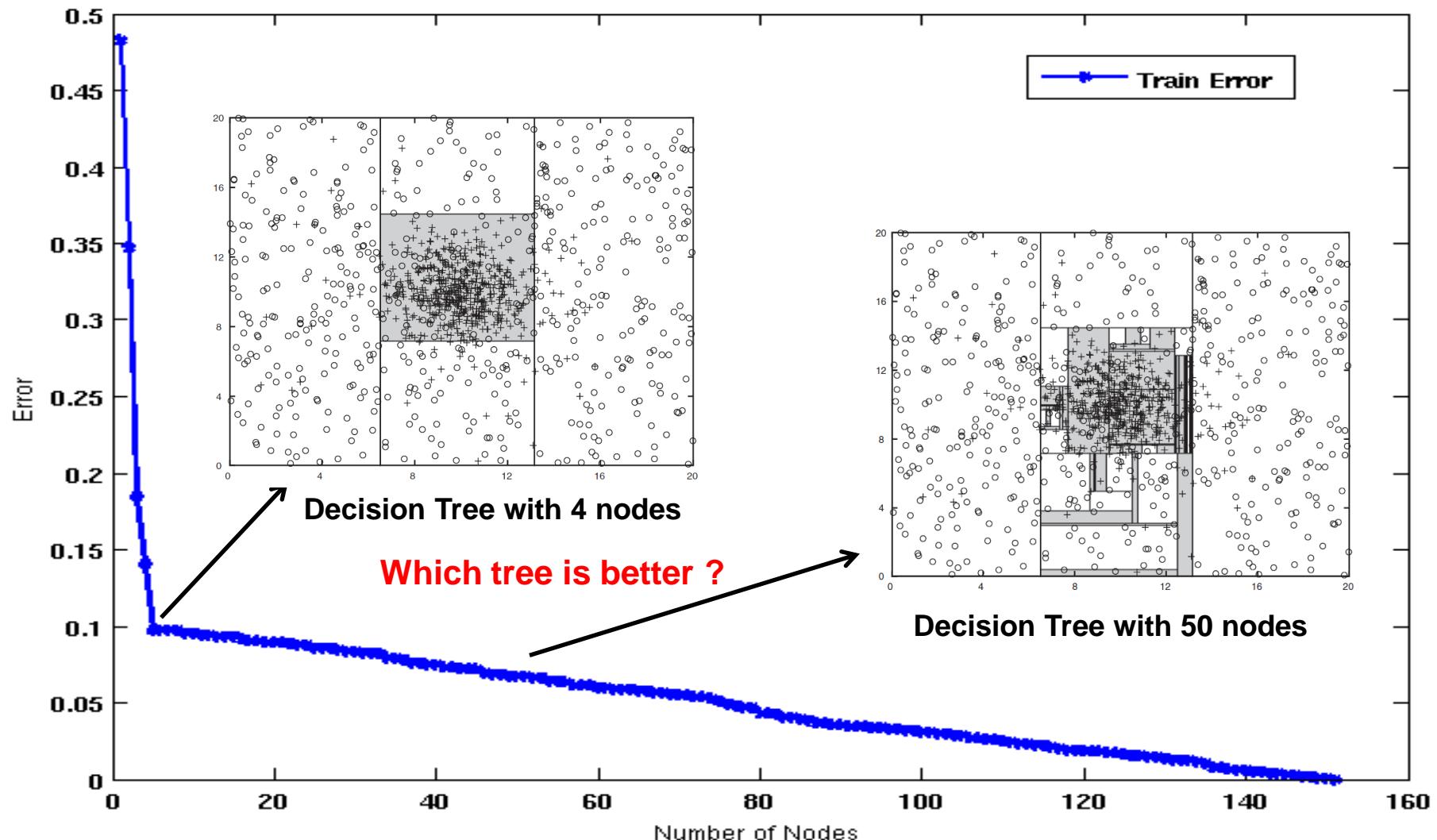
Decision Tree with 4 nodes



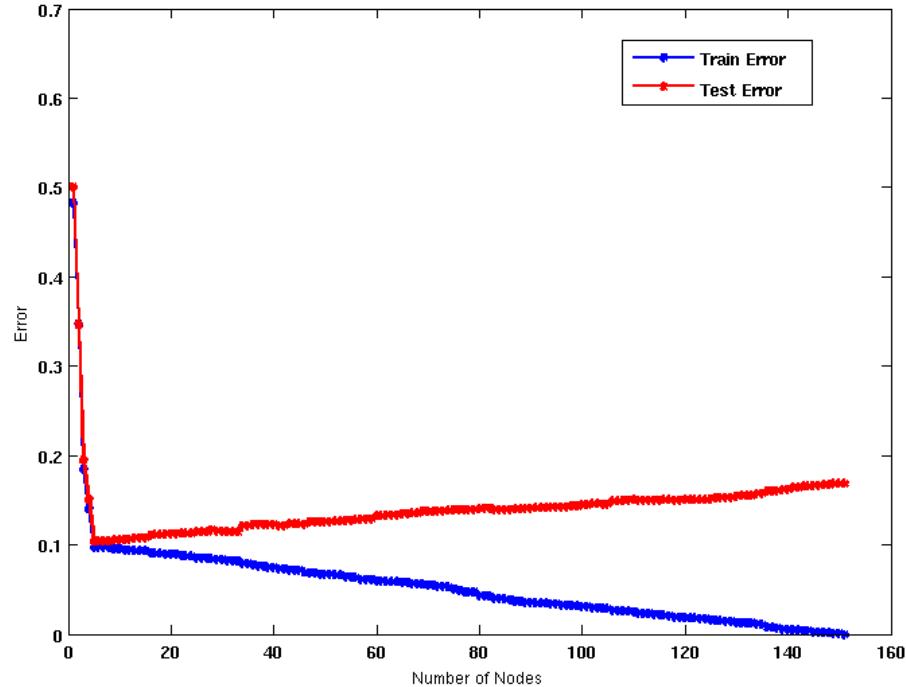
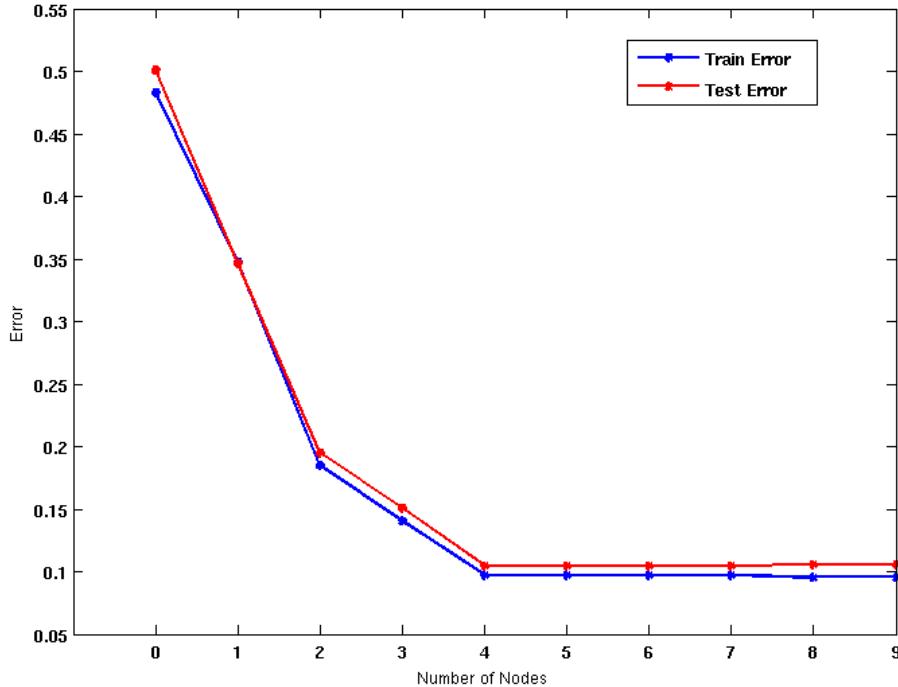
Decision Tree with 50 nodes



Which tree is better?



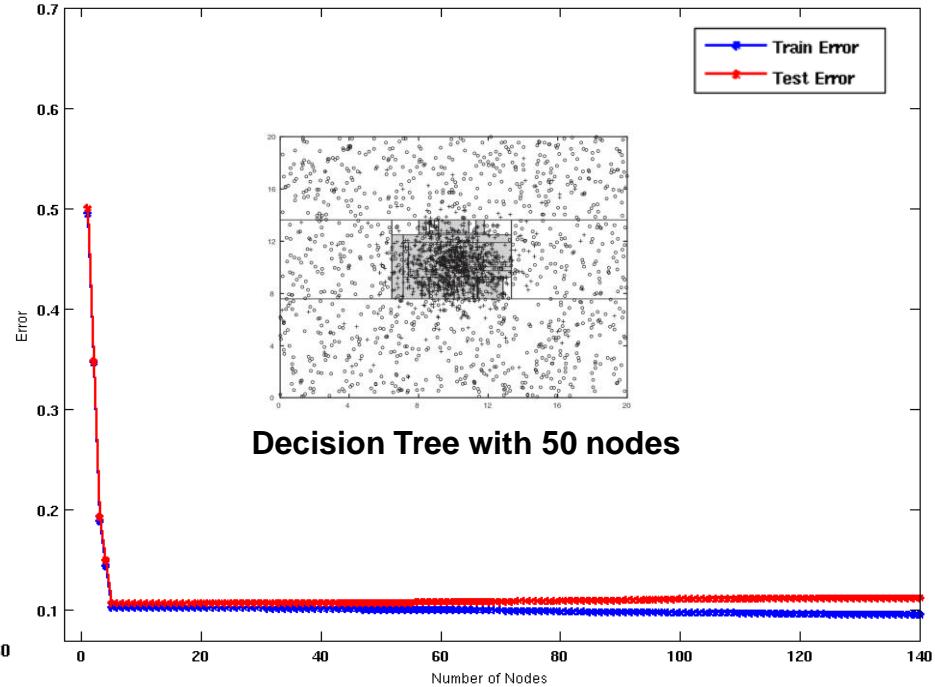
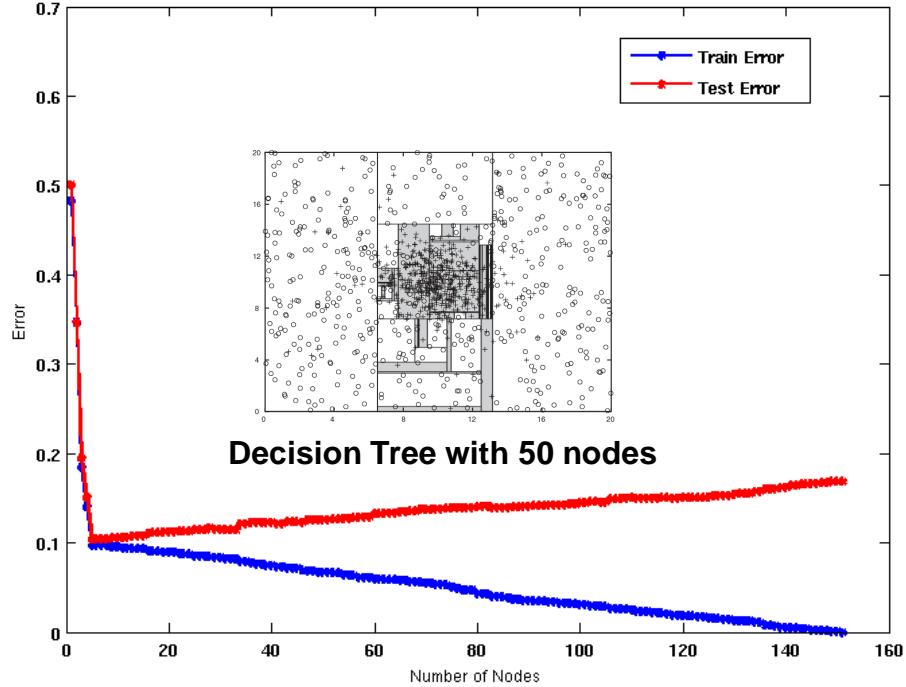
Model Overfitting



Underfitting: when model is too simple, both training and test errors are large

Overfitting: when model is too complex, training error is small, but test error is large

Model Overfitting



Using twice the number of data instances

- If training data is under-representative, testing errors increase and training errors decrease on increasing number of nodes
- Increasing the size of training data reduces the difference between training and testing errors at a given number of nodes

Effect of Multiple Comparison Procedure

- Consider the task of predicting whether stock market will rise/fall in the next 10 trading days

- Random guessing:

$$P(\text{correct}) = 0.5$$

- Make 10 random guesses in a row:

$$P(\#\text{correct} \geq 8) = \frac{\binom{10}{8} + \binom{10}{9} + \binom{10}{10}}{2^{10}} = 0.0547$$

| | |
|--------|------|
| Day 1 | Up |
| Day 2 | Down |
| Day 3 | Down |
| Day 4 | Up |
| Day 5 | Down |
| Day 6 | Down |
| Day 7 | Up |
| Day 8 | Up |
| Day 9 | Up |
| Day 10 | Down |

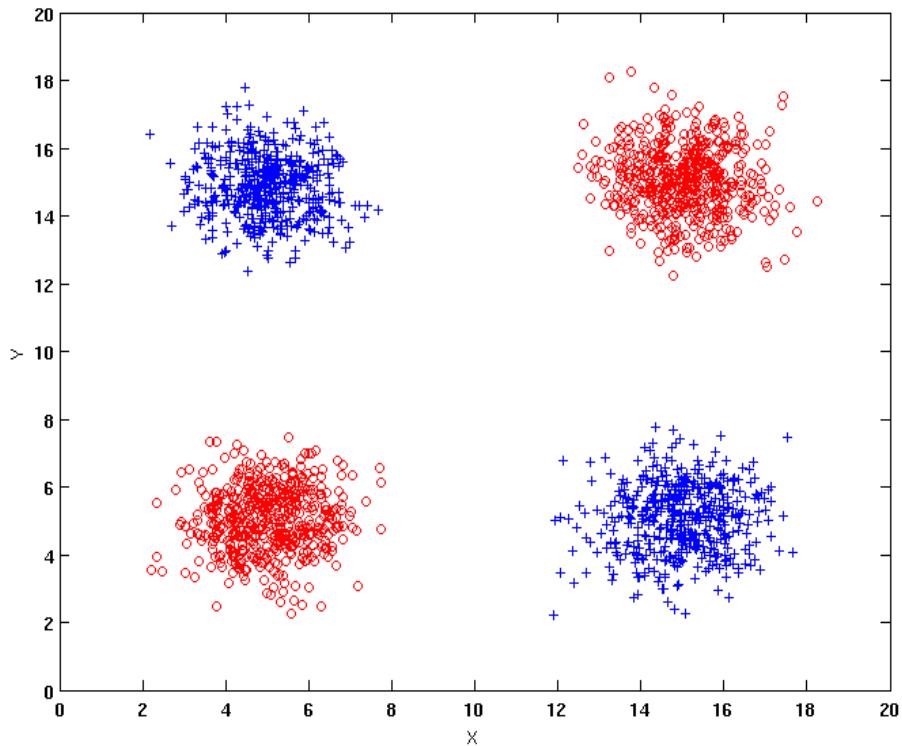
Effect of Multiple Comparison Procedure

- Approach:
 - Get 50 analysts
 - Each analyst makes 10 random guesses
 - Choose the analyst that makes the most number of correct predictions

- Probability that at least one analyst makes at least 8 correct predictions

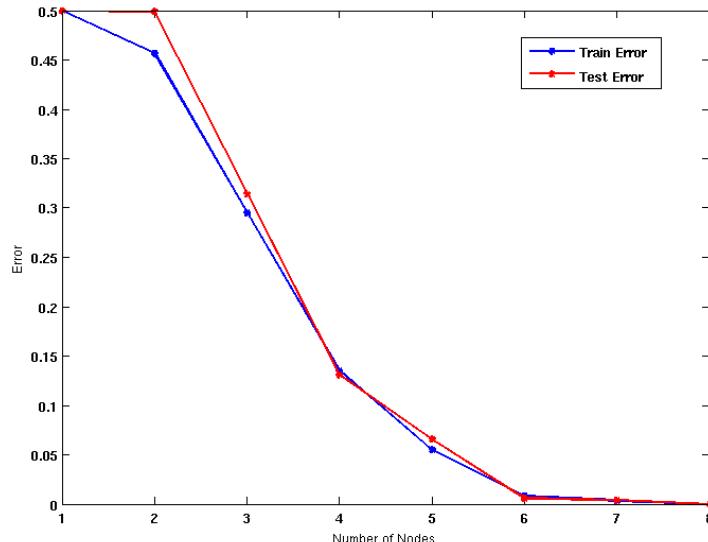
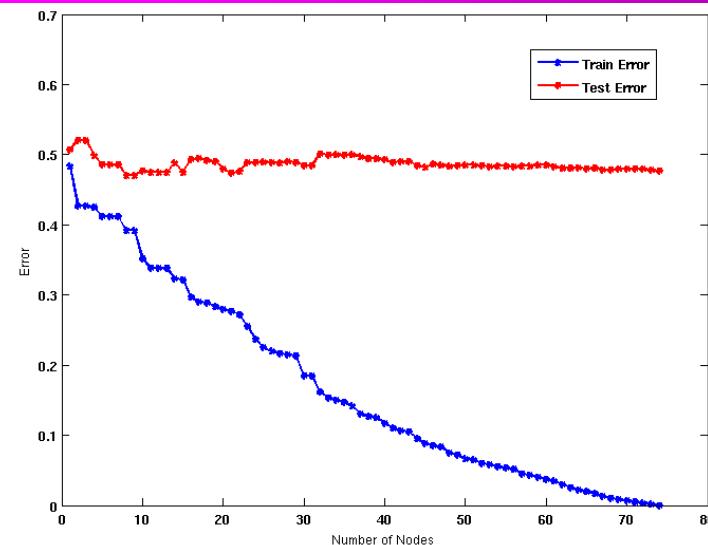
$$P(\# \text{correct} \geq 8) = 1 - (1 - 0.0547)^{50} = 0.9399$$

Effect of Multiple Comparison - Example



Use additional 100 noisy variables generated from a uniform distribution along with X and Y as attributes.

Use 30% of the data for training and 70% of the data for testing



Using only X and Y as attributes

Classification Errors

- Training errors (apparent errors)
 - Errors committed on the training set
- Test errors
 - Errors committed on the test set
- Generalization errors
 - Expected error of a model over random selection of records from same distribution

Estimating Generalization Errors

- Re-substitution errors: error **on training** data ($\sum e(t)$)
- Generalization errors: error **on testing** data ($\sum e'(t)$)
- Methods for estimating generalization errors:
 - Optimistic approach: $e'(t) = e(t)$
 - Reduced error pruning (REP):
 - ◆ uses validation data set to estimate generalization error

- **Pessimistic Error Estimate**
- **Computes generalization error (on testing data) as the sum of training error and a penalty term for model complexity.**
- **The pessimistic error estimate of a decision tree T , $\text{eg}(T)$, can be computed as follows:**

$$\text{eg}(T) = [e(T) + \Omega(T)] / N_t, \text{ Where,}$$

$e(T)$ is the overall training error of the decision tree,

N_t is the number of training records, and

$\Omega(t_i)$ is the penalty term associated with each leaf node t_i .

- **Thus out of many possible tree the one which has a better pessimistic error is selected.**
- **Also a node should not be expanded into its child nodes unless it reduces the misclassification error for more than one training record.**

– Pessimistic approach:

- ◆ For each leaf node: $e'(t) = (e(t)+0.5)$ (assume $\Omega(T)=0.5$)
- ◆ Total errors: $e'(T) = e(T) + N \times 0.5$ (N: number of leaf nodes)
- ◆ For a tree with 30 leaf nodes and 10 errors on training (out of 1000 instances):
◆ Training error = $10/1000 = 1\%$

Generalization error = $(10 + 30 \times 0.5)/1000 = 2.5\%$

Occam's Razor

- Given two models of similar generalization errors, one should prefer the simpler model over the more complex model
- For complex models, there is a greater chance that it was fitted accidentally by errors in data
- Therefore, one should include model complexity when evaluating a model

Incorporating Model Complexity

□ Rationale: Occam's Razor

- Given two models of similar generalization errors, one should prefer the simpler model over the more complex model
- A complex model has a greater chance of being fitted accidentally by errors in data
- Therefore, one should include model complexity when evaluating a model

$$\text{Gen. Error(Model)} = \text{Train. Error(Model, Train. Data)} + \alpha \times \text{Complexity(Model)}$$

Estimating the Complexity of Decision Trees

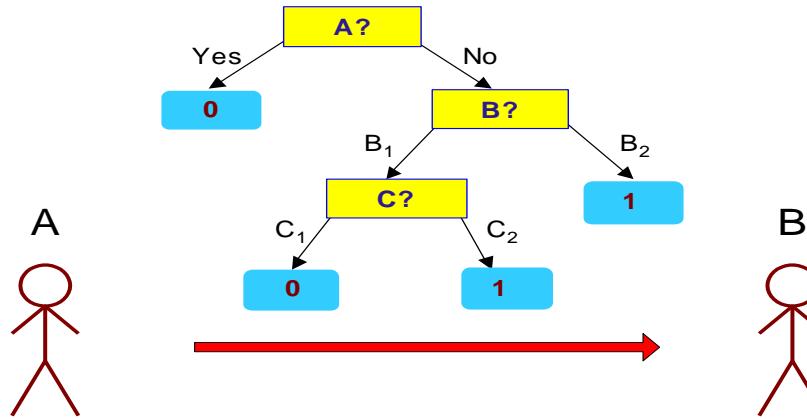
- **Pessimistic Error Estimate** of decision tree T with k leaf nodes:

$$err_{gen}(T) = err(T) + \Omega \times \frac{k}{N_{train}}.$$

- $err(T)$: error rate on all training records
- Ω : trade-off hyper-parameter (similar to α)
 - ◆ Relative cost of adding a leaf node
- k : number of leaf nodes
- N_{train} : total number of training records

Minimum Description Length (MDL)

| X | y |
|-------|-----|
| X_1 | 1 |
| X_2 | 0 |
| X_3 | 0 |
| X_4 | 1 |
| ... | ... |
| X_n | 1 |

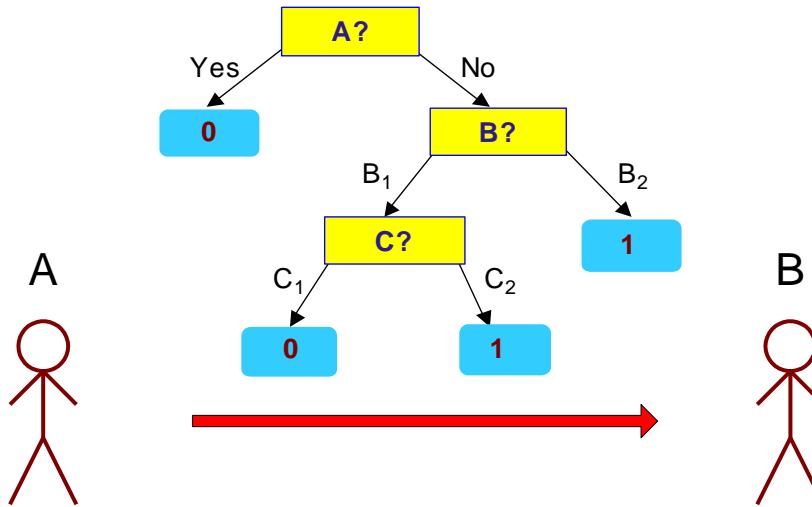


| X | y |
|-------|-----|
| X_1 | ? |
| X_2 | ? |
| X_3 | ? |
| X_4 | ? |
| ... | ... |
| X_n | ? |

- Person A and B are given a set of records with known attribute values x.
- 'A' knows the exact class label for each record, while person 'B' knows none of this information.
- 'B' can obtain the classification of each record by requesting that 'A' transmits the class labels sequentially.
- Such a message would require $\Theta(n)$ bits of information, where n is the total number of records.
- Alternatively, A may decide to build a classification model that summarizes the relationship between x and y.
- The model can be encoded in a compact form before being transmitted to B.
- If the model is 100% accurate, then the cost of transmission is equivalent to the cost of encoding the model.
- Otherwise, A must also transmit information about which record is classified incorrectly by the model.

Minimum Description Length (MDL)

| X | y |
|-------|-----|
| X_1 | 1 |
| X_2 | 0 |
| X_3 | 0 |
| X_4 | 1 |
| ... | ... |
| X_n | 1 |



| X | y |
|-------|-----|
| X_1 | ? |
| X_2 | ? |
| X_3 | ? |
| X_4 | ? |
| ... | ... |
| X_n | ? |

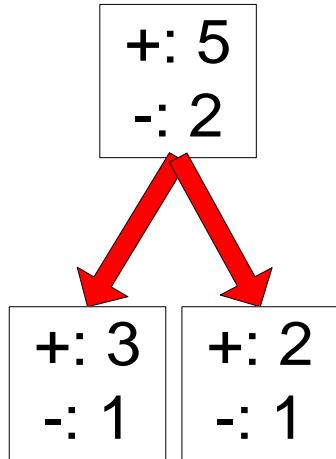
- $\text{Cost}(\text{Model}, \text{Data}) = \text{Cost}(\text{Data}|\text{Model}) + \alpha \times \text{Cost}(\text{Model})$
 - Cost is the number of bits needed for encoding.
 - Search for the least costly model.
- $\text{Cost}(\text{Data}|\text{Model})$ encodes the misclassification errors.
- $\text{Cost}(\text{Model})$ uses node encoding (number of children) plus splitting condition encoding.

Among all equally good classifier you would like to pick up one which requires least number of bits to describe it. So the description length should be as small as possible given that the classifier has some acceptable level of performance. (Classifier description: no of support vectors in SVM, no of weight and other parameters in ANN etc. and encoding them for communication or so.)

Small error requires less number of bits. If classifier is complex we need lots of bits to describe it but error is minimum and number of bits required to specify error are minimum. Trade off between classifier description and error description). We need to select based on some trade off criterion.

4.4.4. Estimating Statistical Bounds

$$e'(N, e, \alpha) = \frac{e + \frac{z_{\alpha/2}^2}{2N} + z_{\alpha/2} \sqrt{\frac{e(1-e)}{N} + \frac{z_{\alpha/2}^2}{4N^2}}}{1 + \frac{z_{\alpha/2}^2}{N}}$$



Before splitting: $e = 2/7$, $e'(7, 2/7, 0.25) = 0.503$

$$e'(T) = 7 \times 0.503 = 3.521$$

After splitting:

$$e(T_L) = 1/4, \quad e'(4, 1/4, 0.25) = 0.537$$

$$e(T_R) = 1/3, \quad e'(3, 1/3, 0.25) = 0.650$$

$$e'(T) = 4 \times 0.537 + 3 \times 0.650 = 4.098$$

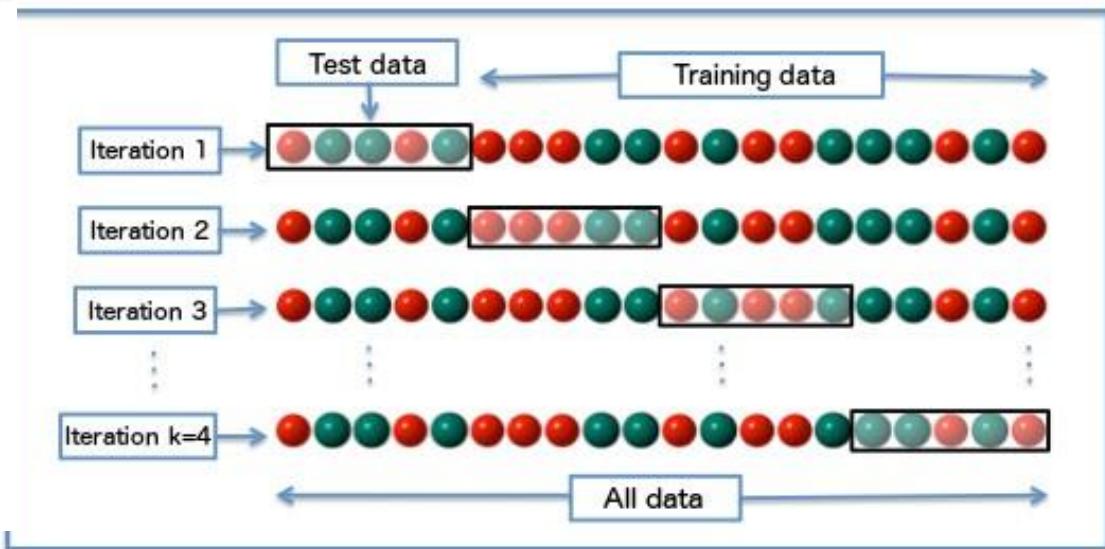
Therefore, do not split

Using Validation Set

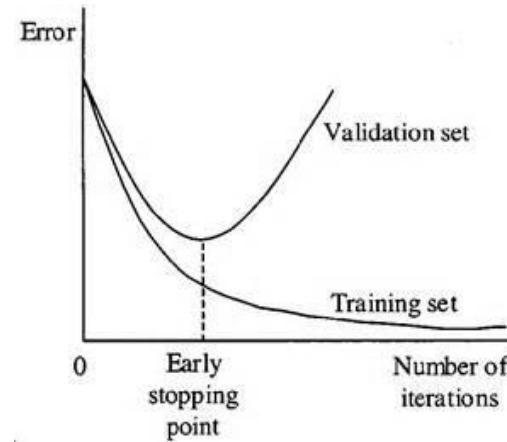
- Divide training data into two parts:
 - Training set:
 - ◆ use for model building
 - Validation set:
 - ◆ use for estimating generalization error
 - ◆ Note: validation set is not the same as test set
- Drawback:
 - Less data available for training

Here are a few of the most popular solutions for overfitting:

- **Cross-Validation:** A standard way to find out-of-sample prediction error is to use 5-fold cross-validation.



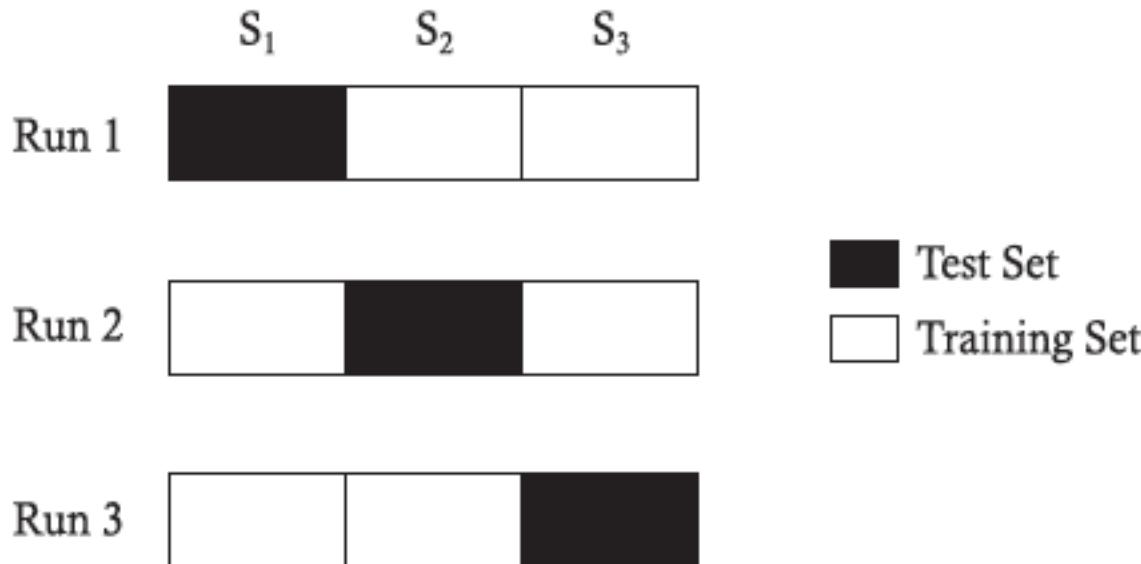
- **Early Stopping:** Its rules provide us with guidance as to how many iterations can be run before the learner begins to over-fit.



- **Pruning:** Pruning is extensively used while building related models. It simply removes the nodes which add little predictive power for the problem in hand.
- **Regularization:** It introduces a cost term for bringing in more features with the objective function. Hence it tries to push the coefficients for many variables to zero and hence reduce cost term.
- **Remove features:** Some algorithms have built-in feature selection. For those that don't, you can manually improve their generalizability by removing irrelevant input features. An interesting way to do so is to tell a story about how each feature fits into the model.
- **Train with more data:** It won't work every time, but training with more data can help algorithms detect the signal better.
- If we just add more noisy data, this technique won't help. That's why you should always ensure your data is clean and relevant.
-

Cross-validation Example

□ 3-fold cross-validation



How to Address Overfitting

□ Pre-Pruning (Early Stopping Rule)

- Stop the algorithm before it becomes a fully-grown tree

shallow tree potentially generalizes better (Occam's razor)

- Typical stopping conditions for a node (no pruning):

- ◆ Stop if all instances belong to the same class
 - ◆ Stop if all the attribute values are the same

- More restrictive conditions:

- ◆ Stop if number of instances is less than some user-specified threshold (High threshold leads to underfitting problem, Low threshold leads to overfitting problem)
 - ◆ Stop if class distribution of instances are independent of the available features (e.g., using χ^2 test)
 - ◆ Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).

How to Address Overfitting...

Post-pruning

1. Grow decision tree to its entirety
2. Trim the nodes of the decision tree in a bottom-up fashion(using a validation set)
3. Estimate generalization error before and after trimming
4. If generalization error improves after trimming
 - replace sub-tree by a leaf node or
 - replace subtree by most frequently used branch

Class label of leaf node is determined from majority class of instances in the sub-tree

Example of Post-Pruning

| | |
|---------------|----|
| Class = Yes | 20 |
| Class = No | 10 |
| Error = 10/30 | |

Training Error (Before splitting) = 10/30

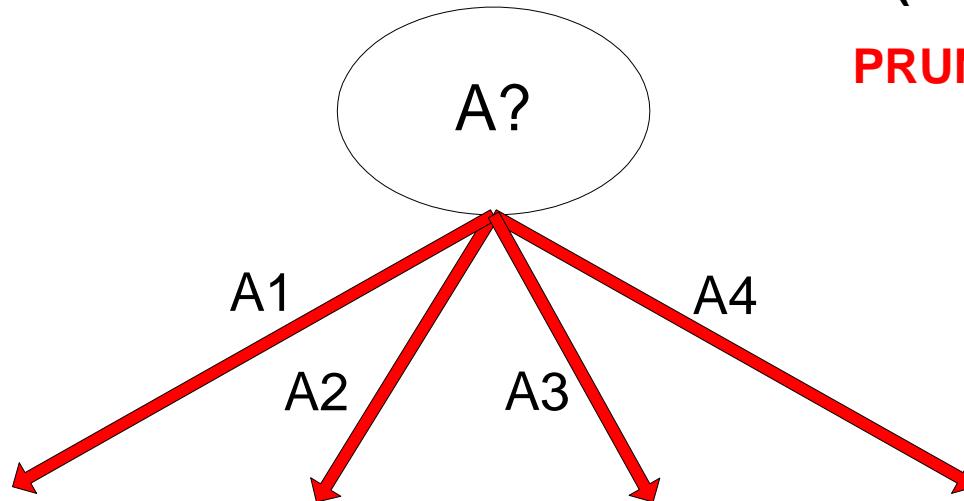
Pessimistic error = $(10 + 0.5)/30 = 10.5/30$

Training Error (After splitting) = 9/30

Pessimistic error (After splitting)

$$= (9 + 4 \times 0.5)/30 = 11/30$$

PRUNE!



| | |
|-------------|---|
| Class = Yes | 8 |
| Class = No | 4 |

| | |
|-------------|---|
| Class = Yes | 3 |
| Class = No | 4 |

| | |
|-------------|---|
| Class = Yes | 4 |
| Class = No | 1 |

| | |
|-------------|---|
| Class = Yes | 5 |
| Class = No | 1 |

Handling Missing Attribute Values

- Missing values affect decision tree construction in three different ways:
 - Affects how impurity measures are computed
 - Affects how to distribute instance with missing value to child nodes
 - Affects how a test instance with missing value is classified

Computing Impurity Measure

| Tid | Refund | Marital Status | Taxable Income | 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 | ? | Single | 90K | Yes |

Missing value

Before Splitting:

Entropy(Parent)

$$= -0.3 \log(0.3) - (0.7)\log(0.7) = 0.8813$$

| | Class = Yes | Class = No |
|------------|-------------|------------|
| Refund=Yes | 0 | 3 |
| Refund>No | 2 | 4 |
| Refund=? | 1 | 0 |

Split on Refund:

Entropy(Refund=Yes) = 0

Entropy(Refund>No)

$$= -(2/6)\log(2/6) - (4/6)\log(4/6) = 0.9183$$

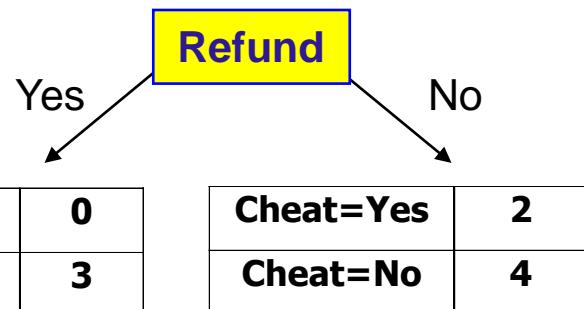
Entropy(Children)

$$= 0.3 (0) + 0.6 (0.9183) = 0.551$$

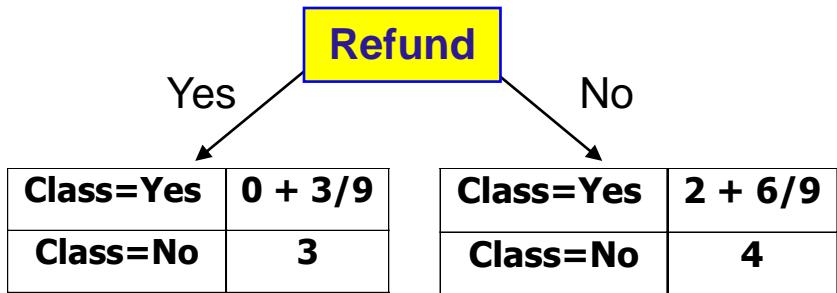
$$\text{Gain} = 0.9 \times (0.8813 - 0.551) = 0.3303$$

Distribute Instances

| Tid | Refund | Marital Status | Taxable Income | 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 |



| Tid | Refund | Marital Status | Taxable Income | Class |
|-----|--------|----------------|----------------|-------|
| 10 | ? | Single | 90K | Yes |



Probability that Refund=Yes is 3/9

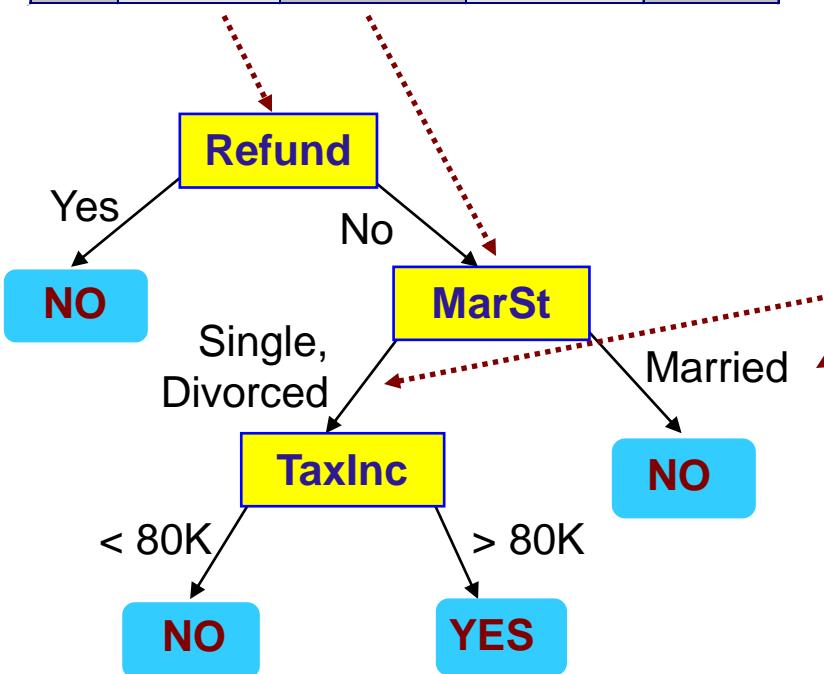
Probability that Refund=No is 6/9

Assign record to the left child with weight = 3/9 and to the right child with weight = 6/9

Classify Instances

New record:

| Tid | Refund | Marital Status | Taxable Income | Class |
|-----|--------|----------------|----------------|-------|
| 11 | No | ? | 85K | ? |



| | Married | Single | Divorced | Total |
|-----------|---------|--------|----------|-------|
| Class=No | 3 | 1 | 0 | 4 |
| Class=Yes | 6/9 | 1 | 1 | 2.67 |
| Total | 3.67 | 2 | 1 | 6.67 |

Probability that Marital Status = Married is $3.67/6.67$

Probability that Marital Status = {Single,Divorced} is $3/6.67$

Model Evaluation

- How good is a model at classifying unseen records?
- Metrics for Performance Evaluation
 - How to evaluate/measure 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?

Model Evaluation

- Purpose:
 - To estimate performance of classifier on previously unseen data (test set)
- Holdout
 - Reserve $k\%$ for training and $(100-k)\%$ 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: $k=n$

Model Selection

- Performed during model building
- Purpose is to ensure that model is not overly complex (to avoid overfitting)
- Need to estimate generalization error
 - Using Validation Set
 - Incorporating Model Complexity
 - Estimating Statistical Bounds

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.
- A **confusion matrix** is a matrix (table) that can be used to measure the performance of an **Classifier** (machine learning algorithm), usually in a supervised learning mode on a set of test data for which the true values are known.
- It allows the visualization of the performance of an algorithm.
- The confusion matrix shows the ways in which your classification model is confused when it makes predictions.
- It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made

Confusion Matrix

- Positive (P) (Yes) (+) (T): Observation is positive (for example: is an apple).
- Negative (N) (No) (-) (F) : Observation is not positive (for example: is not an apple)

| | | PREDICTED CLASS | |
|----------------------------------|-----------|---------------------------------------|----------------------------------------|
| ACTUAL CLASS / Observation | | Class= | Class= |
| | Class=Yes | a (TP) | b (FN) <small>Type II error</small> |
| | Class>No | c (FP) <small>Type I error</small> | d (TN) |

- a: TP (true positive)
- b: FN (false negative)
- c: FP (false positive)
- d: TN (true negative)

True +ve: Observation is positive, and is predicted to be positive.

If the person is actually having cancer (Actual class = Yes) and we predict correctly as **Yes** he is actually having a cancer (Predicted Class = Yes)

False -ve: (Type 2 Error) Observation is positive, but is predicted negative

If the person is actually having a cancer(Actual class = Yes) and we predict wrong as **No**, he is actually having a cancer (Predicted class = No)

False +ve: (Type 1 Error) Observation is negative, but is predicted positive

if the person is not having a cancer (Actual class = No) and we predict as **Yes**,he is actually not having cancer but we predict as he is having cancer, which is wrong (Predicted class = Yes)

True -ve: Observation is negative, and is predicted to be negative.

If the person is not having a cancer (Actual class = No) and we predict correctly as **No**, that means we predict correctly as the person is not having cancer (Predicted Class = No)

Class Imbalance

Model performance for classification models is usually debatable in terms of **which model performance is most relevant**, especially when the dataset is imbalanced.

The usual model performance measures for evaluating a classification model are accuracy, sensitivity or recall, specificity, precision, KS statistic and Area under the curve (AUC).

The Class Imbalance Problem

- Sometimes, classes have **very unequal frequency**
- Fraud detection: 98% transactions OK, 2% fraud
- E-commerce: 99% surfers don't buy, 1% buy
- Intruder detection: 99.99% of the users are no intruders
- Security: >99.99% of Americans are not terrorists

The Class Imbalance Problem

- Consider a 2-class problem : The class of interest is commonly called Positive Class
 - Number of Class 0 (Negative) examples = 9990
 - Number of Class 1 (Positive) examples = 10
- If model predicts everything to be class 0, accuracy is $9990+0/10000 = 99.9\%$
- Accuracy is misleading because model does not detect any class 1 example
- In this case we get accuracy as 99.9% but we cannot evaluate the performance of a model on the basis of accuracy because we were not able to predict class 1 examples.
- If we see carefully the proportion of Class 0 examples is high which is 9990 and the proportion of class 1 examples is very low which is 10 for the 2-class problem.
- Hence in the above case accuracy will not be a correct measure to evaluate the performance of model.

Metrics for Performance Evaluation...

Accuracy: Widely-used metric: -Treat every class as equally important

$$\text{Accuracy} = \frac{\text{Correct Prediction}}{\text{All Predictions}}$$

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

It may be defined as the **number of correct predictions** made by our ML model.

Accuracy is the number of **correct (True) predictions** made by the model by the total number of records. The best accuracy is 100% indicating that all the predictions are correct.

- For an imbalanced dataset, accuracy is not a valid measure of model performance.
- For a dataset where the default rate is 5%, even if all the records are predicted as 0, the model will still have an accuracy of 95%. It assumes equal costs for both kinds of errors.
- The model may ignore all the defaults and can be very detrimental to the business.

Alternate: Use performance metrics from information retrieval which are biased towards the positive class by ignoring TN

Recall or Sensitivity:

Recall may be defined as the **number of positives returned** by our ML model. **When it's actually yes, how often does it predict yes?**

Recall is the ratio of the total number of correctly classified positive examples divide to the total number of positive examples.

High Recall indicates the class is correctly recognized (small number of FN).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Precision:

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. **When it predicts yes, how often is it correct?**

High Precision indicates an example labeled as positive is indeed positive (small number of FP).

Precision is given by the relation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- sensitivity, recall, hit rate, or true positive rate (TPR)
- specificity, selectivity or true negative rate (TNR)
- precision or positive predictive value (PPV)
- negative predictive value (NPV)
- miss rate or false negative rate (FNR)
- fall-out or false positive rate (FPR)
- false discovery rate (FDR)
- false omission rate (FOR)
- Threat score (TS) or Critical Success Index (CSI)

KS statistic

KS statistic is a measure of degree of separation between the positive and negative distributions.

KS value of 100 indicates that the scores partition the records exactly such that one group contains all positives and the other contains all negatives.

In practical situations, a KS value higher than 50% is desirable.

High recall, low precision: This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives (high FP). $R = \frac{TP}{TP+FN}$

Low recall, high precision: This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP) $P = \frac{TP}{TP+FP}$

| | |
|----|----|
| TP | FN |
| FP | TN |

Recall or sensitivity gives us information about a model's performance on false negatives (incorrect prediction of customers who will default), while precision gives us information of the model's performance of false positives.

Based on what is predicted, precision or recall might be more critical for a model.

The cost function comes into play in deciding which of the incorrect predictions can be more detrimental — the false positive or the false negative (in other words, which performance measure is important — precision or recall).

It is difficult to compare two models with low precision and high recall or vice versa. So to make them comparable, we use F-Score. F-score helps to measure Recall and Precision at the same time.

F-measure:

Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses **Harmonic Mean** in place of **Arithmetic Mean** as it punishes the extreme values more.

The F-Measure will always be nearer to the smaller value of Precision or Recall.

$$F\text{-measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

- Let's consider an example now, in which we have **infinite data elements of class B** and **a single element of class A** and the **model is predicting class A against all the instances in the test data**.
Here, Precision : 0.0, Recall : 1.0
- Now: Arithmetic mean: 0.5 , Harmonic mean: 0.0
When taking the arithmetic mean, it would have 50% correct. Despite being the worst possible outcome! While taking the harmonic mean, the F-measure is 0.

Example

| n=165 | Predicted: NO | Predicted: YES | |
|-------------|---------------|----------------|-----|
| Actual: NO | TN = 50 | FP = 10 | 60 |
| Actual: YES | FN = 5 | TP = 100 | 105 |
| | 55 | 110 | |

| | | P | N | |
|---|-----|-----|----|-----|
| P | TP | 100 | FN | 5 |
| | FP | 10 | TN | 50 |
| | 110 | | 55 | 165 |

• **Accuracy:** Overall, how often is the classifier correct?

- $(TP+TN)/\text{total} = (100+50)/165 = 0.91$

• **Misclassification Rate:** Overall, how often is it wrong?

- $(FP+FN)/\text{total} = (10+5)/165 = 0.09$
- equivalent to 1 minus Accuracy also known as "Error Rate"

• **True Positive Rate:** When it's actually yes, how often does it predict yes?

- $TP/\text{actual yes} = 100/105 = 0.95$ also known as "Sensitivity" or "Recall"

• **False Positive Rate:** When it's actually no, how often does it predict yes?

- $FP/\text{actual no} = 10/60 = 0.17$

• **True Negative Rate:** When it's actually no, how often does it predict no?

- $TN/\text{actual no} = 50/60 = 0.83$
- equivalent to 1 minus False Positive Rate also known as "Specificity"

• **Precision:** When it predicts yes, how often is it correct?

- $TP/\text{predicted yes} = 100/110 = 0.91$

• **Prevalence:** How often does the yes condition actually occur in our sample?

- $\text{actual yes}/\text{total} = 105/165 = 0.64$

$$\begin{aligned}
 P &= \frac{TP}{TP+FN} = \frac{100}{100+5} = 0.95 \\
 F &= \frac{2+RXP}{R+P} = \frac{2+0.95 \times 0.91}{0.95 + 0.91} = 0.729 \\
 &= 1 - 0.729 = 0.271 \\
 &= 27.1\%
 \end{aligned}$$

How to Calculate a Confusion Matrix

- Here, is step by step process for calculating a confusion Matrix in data mining
- Step 1) First, you need to test dataset with its expected outcome values.
- Step 2) Predict all the rows in the test dataset.
- Step 3) Calculate the expected predictions and outcomes:
 - The total of correct predictions of each class.
 - The total of incorrect predictions of each class.
- After that, these numbers are organized in the below-given methods:
 - Every row of the matrix links to a predicted class.
 - Every column of the matrix corresponds with an actual class.
- The total counts of correct and incorrect classification are entered into the table.
 - The sum of correct predictions for a class go into the predicted column and expected row for that class value.
 - The sum of incorrect predictions for a class goes into the expected row for that class value and the predicted column for that specific class value.

- **Null Error Rate:**

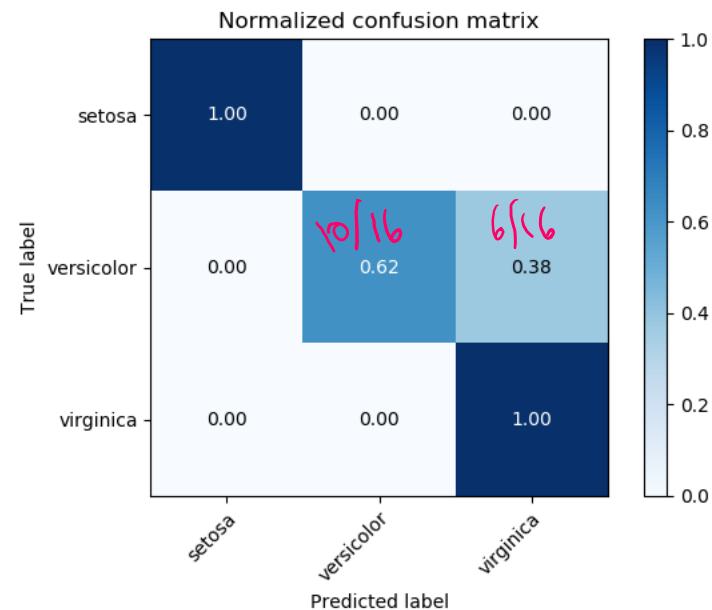
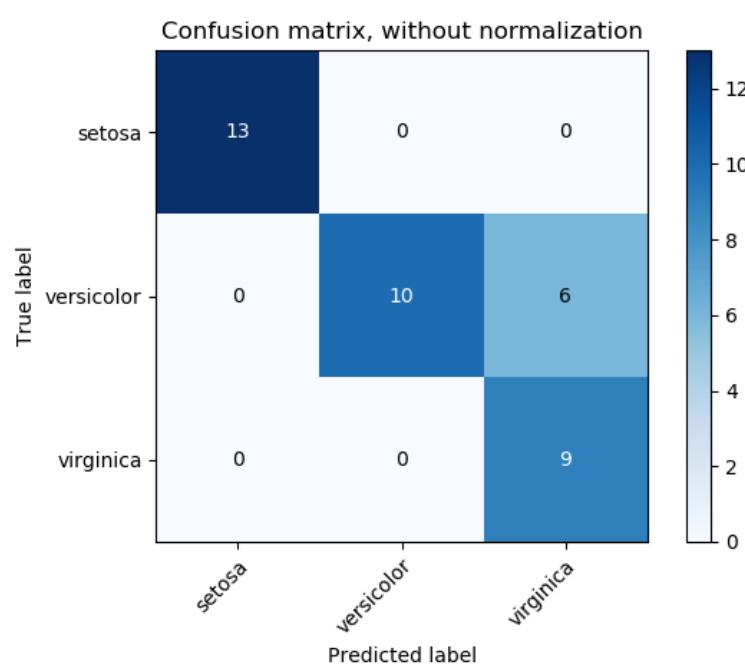
- This is how often you would be wrong if you always predicted the majority class. (In our example, the null error rate would be $60/165=0.36$ because if you always predicted yes, you would only be wrong for the 60 "no" cases.)
- This can be a useful baseline metric to compare your classifier against.
- However, the best classifier for a particular application will sometimes have a higher error rate than the null error rate
- **Cohen's Kappa: (Range 0-1; ideal=1)**
- This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance.
- In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate.

Why you need Confusion matrix?

- Pros/benefits of using a confusion matrix:
- It shows how any classification model is confused when it makes predictions.
- Confusion matrix not only gives you insight into the errors being made by your classifier but also types of errors that are being made.
- This breakdown helps you to overcomes the limitation of using classification accuracy alone.
- Every column of the confusion matrix represents the instances of that predicted class.
- Each row of the confusion matrix represents the instances of the actual class.

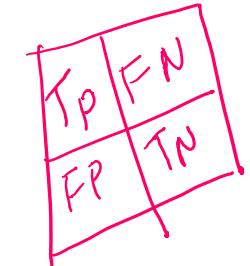
Confusion matrix with and without normalization by class support size (number of elements in each class). T

This kind of normalization can be interesting in case of class imbalance to have a more visual interpretation of which class is being misclassified.



Cost Matrix

| | | PREDICTED CLASS | |
|--------------|--------------------|-------------------------------------------|---------------------------|
| ACTUAL CLASS | $C(i j)$ | Class=Yes | Class>No |
| | Class=Yes | $C(\text{Yes} \text{Yes})$ | $C(\text{No} \text{Yes})$ |
| | Class>No | $C(\text{Yes} \text{No})$ <i>i j</i> | $C(\text{No} \text{No})$ |



$C(i|j)$: Cost of misclassifying class j example as class i

Cost matrix is similar to the confusion matrix except the fact that we are calculating the cost of wrong prediction or right prediction.

Computing Cost of Classification

| Cost Matrix | | PREDICTED CLASS | |
|--------------|---------|-----------------|-----|
| ACTUAL CLASS | C(i j) | + | - |
| | + | -1 | 100 |
| | - | 1 | 0 |

| Model M ₁ | PREDICTED CLASS | | |
|----------------------|-----------------|-----|-----|
| ACTUAL CLASS | | + | - |
| | + | 150 | 40 |
| | - | 60 | 250 |

| Model M ₂ | PREDICTED CLASS | | |
|----------------------|-----------------|-----|-----|
| ACTUAL CLASS | | + | - |
| | + | 250 | 45 |
| | - | 5 | 200 |

$$\text{Accuracy} = 150+250/150+250+40+60=80\%$$

$$\text{Cost} = 150(-1)+40(100)+60(1)+250(0)=3910$$

$$\text{Accuracy} = 250+200/250+200+45+5=90\%$$

$$\text{Cost} = 250(-1)+45(100)+5(1)+200(0)=4255$$

Cost vs Accuracy

| Count | PREDICTED CLASS | |
|--------------|-----------------|----------|
| ACTUAL CLASS | Class=Yes | Class>No |
| | Class=Yes | a |
| | Class>No | d |

Accuracy is proportional to cost if

1. $C(Yes|No)=C(No|Yes) = q$
2. $C(Yes|Yes)=C(No|No) = p$

$$N = a + b + c + d$$

$$\text{Accuracy} = (a + d)/N$$

| Cost | PREDICTED CLASS | |
|--------------|-----------------|----------|
| ACTUAL CLASS | Class=Yes | Class>No |
| | Class=Yes | p |
| | Class>No | q |

$$\text{Cost} = p (a + d) + q (b + c)$$

$$= p (a + d) + q (N - a - d)$$

$$= q N - (q - p)(a + d)$$

$$= N [q - (q-p) \times \text{Accuracy}]$$

Two models **Model M1 and M2** both of which are having correct predictions and wrong predictions.

Accuracy for Model M2 is higher compare to Model M1, however the cost for Model M2 is higher compare to Model M1.

So it depends on what kind of problem statement we are facing.

If we are focusing on accuracy then we will go with the Model M2 (In this case we need to compromise on cost) , however if we are focusing on cost then we will go with the Model M1 (In this case we need to compromise on accuracy).

Let us understand some model performance measures based on an example of predicting loan default.

A loan default dataset is a typical example of an imbalanced dataset where the two classes are Loan default Y and Loan default N.

The number of loan defaulters is usually a very small fraction of the total dataset — not more than 7–8%.

This provides a classical imbalanced dataset to understand why cost functions are critical in deciding on which model to use.

Since the false negative cost is the highest, the most ***optimal model will be the one with the minimum false negatives***. In other words, a model with higher sensitivity / recall will fetch a higher net revenue compared to other models. Now that we have the method to calculate the net revenue, let us compare 2 models based on their confusion matrix:

| | | Predicted | |
|----------|---|-----------|----|
| | | 0 | 1 |
| Observed | 0 | 1880 | 47 |
| | 1 | 82 | 84 |

1880*12000

47*-12000

84*-3000

82*-10,00,000

-6,02,56,000

| | | Predicted | |
|----------|---|-----------|-----|
| | | 0 | 1 |
| Observed | 0 | 1916 | 47 |
| | 1 | 10 | 120 |

1916*12000

47*-12000

10*-3000

120*-10,00,000

1,20,68,000

Cost-Sensitive Measures

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

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

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

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

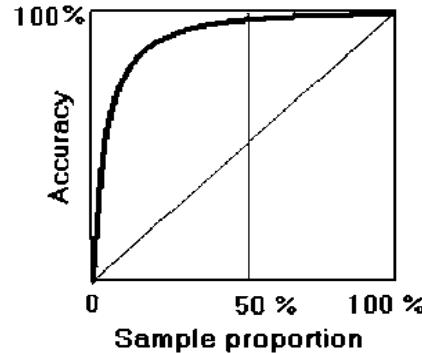
Methods for Performance Evaluation

- 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

Learning Curve

ACCURACY AS A FUNCTION OF SAMPLE SIZE

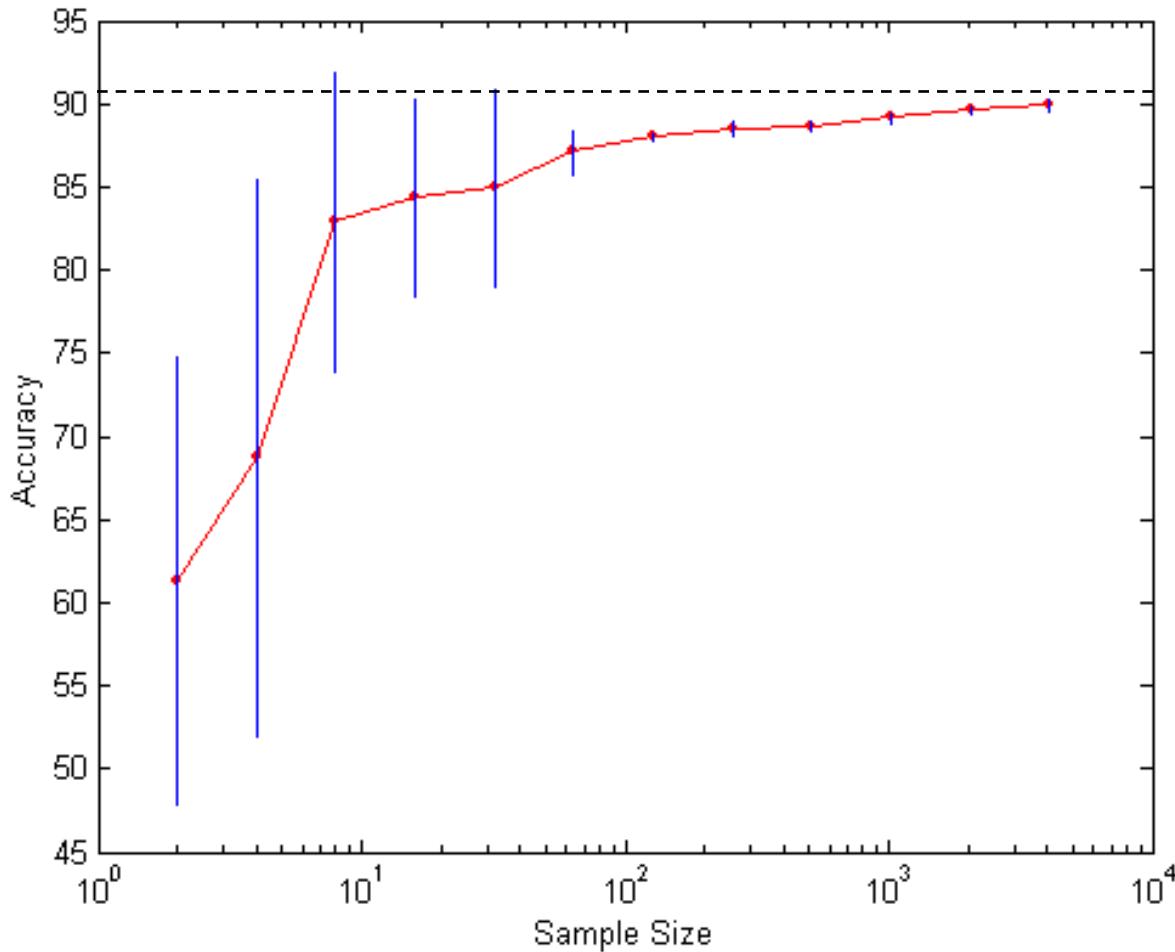
The following diagram illustrates the pattern of accuracy growth when sample size increases.



To be noted that:

- Accuracy is 100% when the entire population has been examined (as in the case of a census).
- The pattern of accuracy growth is not linear. The accuracy of a sample equal to half the data population size is not 50% but very near to 100%.
- Good accuracy levels can be achieved at relatively small sample sizes, provided that the samples are representative.
- The result of this relationship is that beyond a certain sample size the gains in accuracy are negligible, while sampling costs increase significantly

Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve:
 - Arithmetic sampling (Langley, et al)
 - Geometric sampling (Provost et al)
- Effect of small sample size:
 - Bias in the estimate
 - Variance of estimate

Methods of Estimation

- Holdout
 - Reserve 2/3 for training and 1/3 for testing
- Random subsampling
 - Repeated holdout
- Cross validation (CV)
 - 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 (in hold out if no representation of the class- ensure at least equal representation from every class)
 - oversampling vs undersampling
- Bootstrap
 - Sampling with replacement (CV with replacement)

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?

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
 - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TP (on the y-axis) against FP (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
 - changing the threshold of algorithm, sample distribution or cost matrix changes the location of the point

ROC chart & Area under the curve (AUC)

ROC chart is a plot of (1-specificity) (FP) in the X axis and sensitivity (TP) in the Y axis.

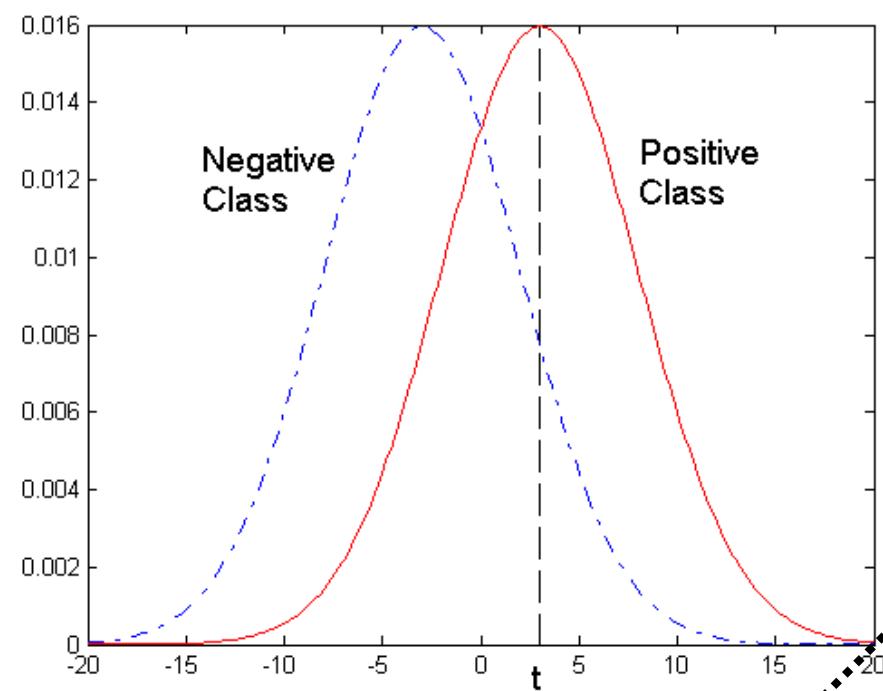
Area under the ROC curve is a measure of model performance.

The AUC of a random classifier is 50% and that of a perfect classifier is 100%.

For practical situations, an AUC of over 70% is desirable.

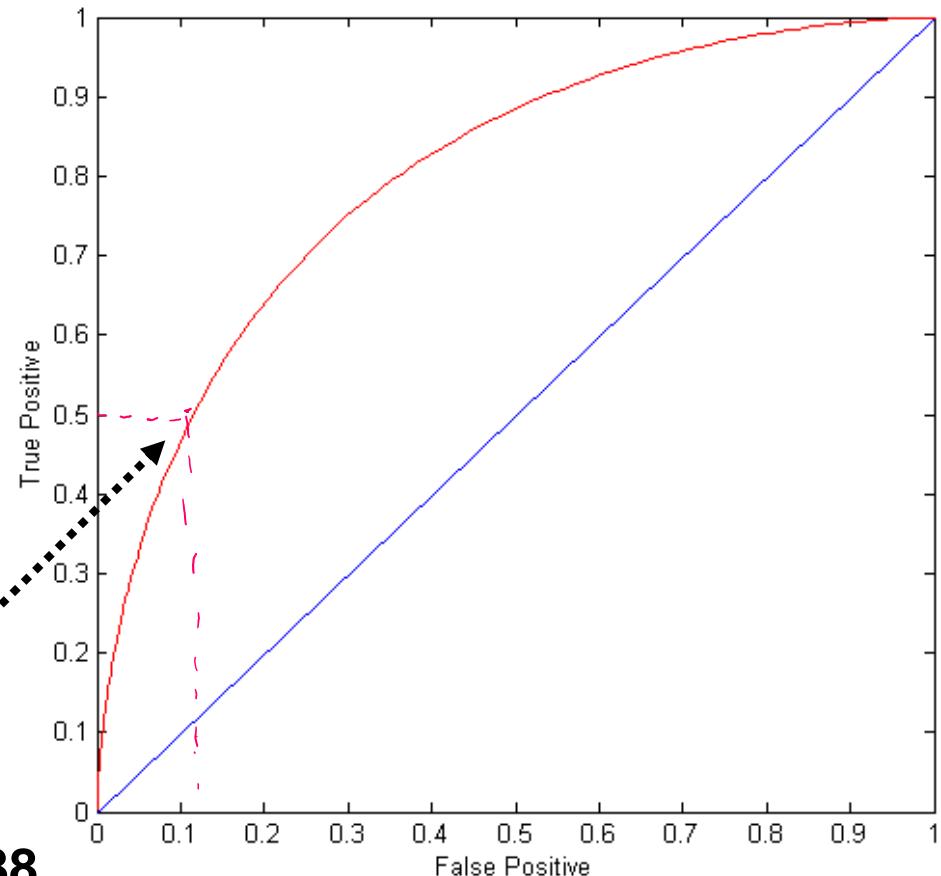
ROC Curve

- 1-dimensional data set containing 2 classes (positive and negative)
- any points located at $x > t$ is classified as positive



At threshold t :

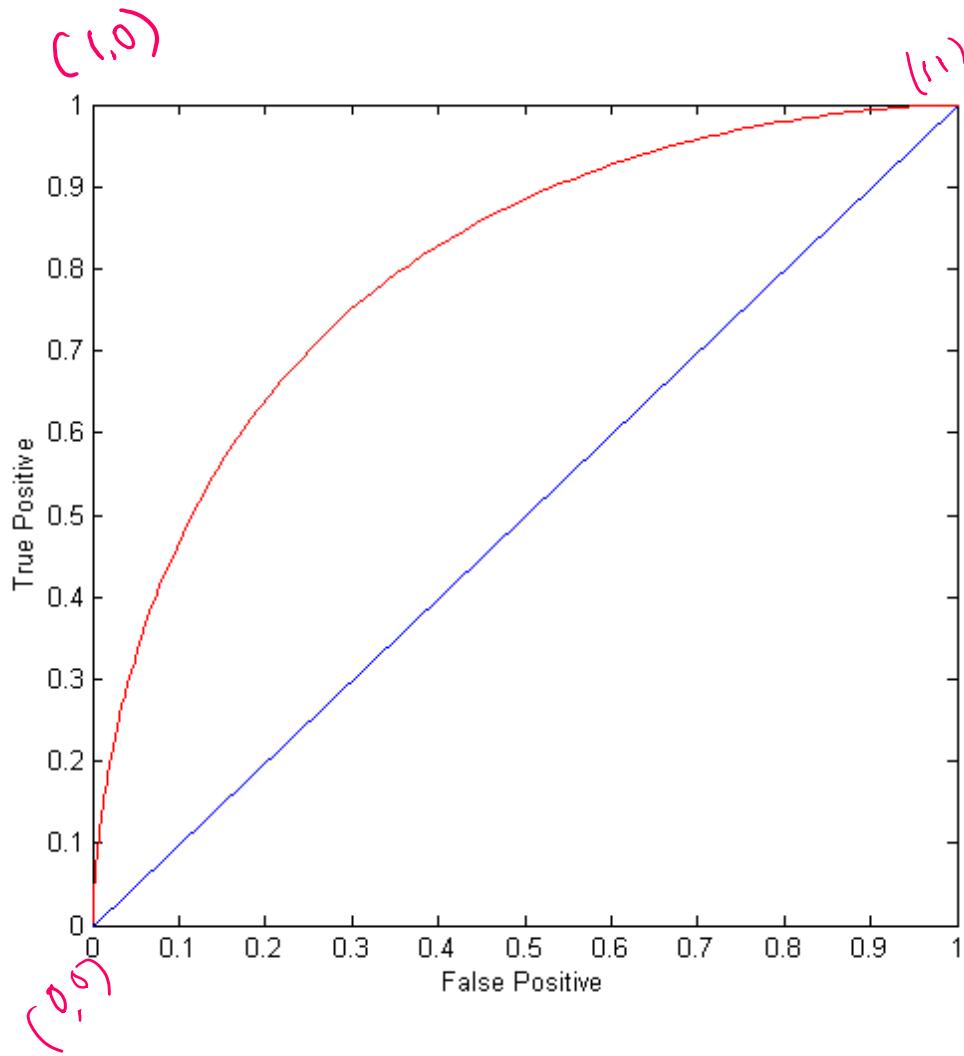
$\text{TP}=0.5, \text{FN}=0.5, \text{FP}=0.12, \text{TN}=0.88$



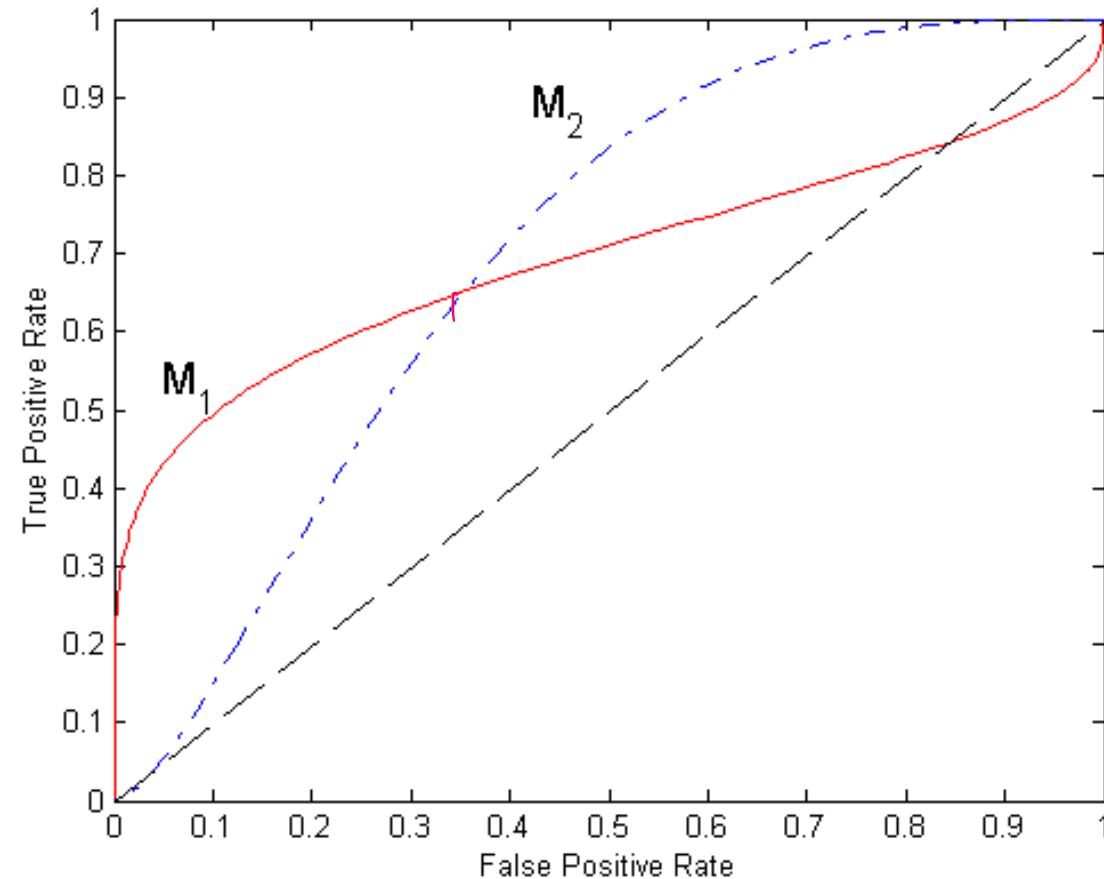
ROC Curve

(TP,FP):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal
- Diagonal line:
 - Random guessing
 - Below diagonal line:
 - ◆ prediction is opposite of the true class



Using ROC for Model Comparison



- No model consistently outperform the other
 - M_1 is better for small FPR
 - M_2 is better for large FPR
- Area Under the ROC curve
 - Ideal:
 - Area = 1
 - Random guess:
 - Area = 0.5

How to Construct an ROC curve

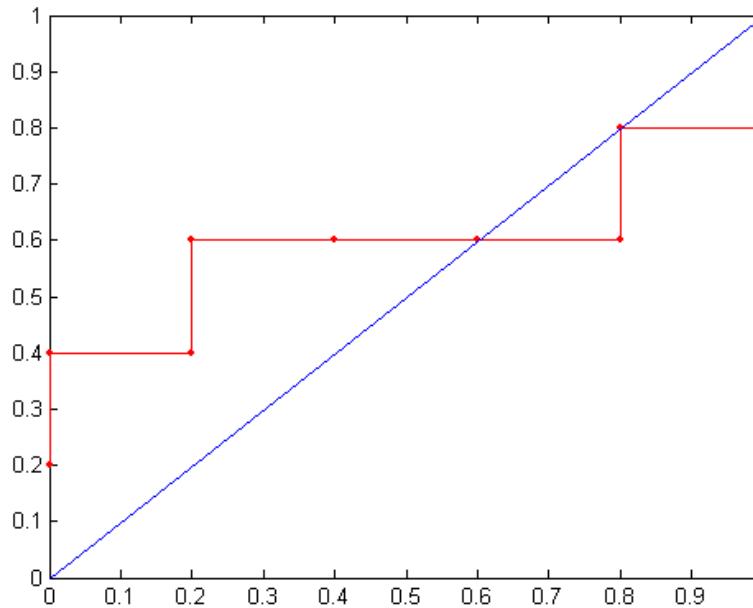
| Instance | $P(+ A)$ | True Class |
|----------|----------|------------|
| 1 | 0.95 | + |
| 2 | 0.93 | + |
| 3 | 0.87 | - |
| 4 | 0.85 | - |
| 5 | 0.85 | - |
| 6 | 0.85 | + |
| 7 | 0.76 | - |
| 8 | 0.53 | + |
| 9 | 0.43 | - |
| 10 | 0.25 | + |

- Use classifier that produces posterior probability for each test instance $P(+|A)$
- Sort the instances according to $P(+|A)$ in decreasing order
- Apply threshold at each unique value of $P(+|A)$
- Count the number of TP, FP, TN, FN at each threshold
- TP rate, $TPR = TP/(TP+FN)$
- FP rate, $FPR = FP/(FP + TN)$

How to construct an ROC curve

| Class | + | - | + | - | - | - | + | - | + | + | |
|--------------|------|------|------|------|------|------|------|------|------|------|------|
| Threshold >= | 0.25 | 0.43 | 0.53 | 0.76 | 0.85 | 0.85 | 0.85 | 0.87 | 0.93 | 0.95 | 1.00 |
| TP | 5 | 4 | 4 | 3 | 3 | 3 | 3 | 2 | 2 | 1 | 0 |
| FP | 5 | 5 | 4 | 4 | 3 | 2 | 1 | 1 | 0 | 0 | 0 |
| TN | 0 | 0 | 1 | 1 | 2 | 3 | 4 | 4 | 5 | 5 | 5 |
| FN | 0 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 4 | 5 |
| → TPR | 1 | 0.8 | 0.8 | 0.6 | 0.6 | 0.6 | 0.6 | 0.4 | 0.4 | 0.2 | 0 |
| → FPR | 1 | 1 | 0.8 | 0.8 | 0.6 | 0.4 | 0.2 | 0.2 | 0 | 0 | 0 |

ROC Curve:



Test of Significance

- Given two models:
 - Model M1: accuracy = 85%, tested on 30 instances
 - Model M2: accuracy = 75%, tested on 5000 instances
- Can we say M1 is better than M2?
 - How much confidence can we place on accuracy of M1 and M2?
 - Can the difference in performance measure be explained as a result of random fluctuations in the test set?