



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

S2-21_DSECLZC415

Introduction to Data Mining



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

Textbooks/Reference Books

Text Books

T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education, 2019
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han and Micheline Kamber Morgan Kaufmann Publishers, 2011

Reference Book(s) & other resources

R1	Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner by Vijay Kotu and Bala Deshpande Morgan Kaufmann Publishers © 2015
	Additional references may be given during lectures

Modular Structure

<u>No</u>	<u>Title of the Module</u>
M1	Introduction to Data Mining
M2	Data Preprocessing
M3	Data Exploration
M4	Classification and Prediction
M5	Clustering
M6	Association Analysis
M7	Anomaly Detection
M8	Data mining on unstructured (Big) data
M9	Data Mining Applications

Evaluation Scheme

No	Name	Type	Weight
1.	Quiz-I	Online	5%
	Quiz-II	Online	5%
	Assignment	Group	10%
2.	Mid-Semester Test		30%
3.	Comprehensive Exam		50%

Data Mining Defined

What Is Data Mining?

Data mining (knowledge discovery from data)

- Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data

Alternative names

- Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.

Watch out: Is everything “data mining”?

- Simple search and query processing
- (Deductive) expert systems

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,)

Why Data Mining?

The Explosive Growth of Data: from terabytes to petabytes

- Data collection and data availability
 - Automated data collection tools, database systems, Web, computerized society
- Major sources of abundant data
 - Business: Web, e-commerce, transactions, stocks, ...
 - Science: Remote sensing, bioinformatics, scientific simulation, ...
 - Society and everyone: news, digital cameras, YouTube

We are drowning in data, but starving for knowledge!

“Necessity is the mother of invention”—Data mining—
Automated analysis of massive data sets

Why Data Mining

A search engine (e.g., Google) receives hundreds of millions of queries every day. Each query can be viewed as a transaction where the user describes her or his information need.

What novel and useful knowledge can a search engine learn from such a huge collection of queries collected from users over time? Some patterns found in user search queries can disclose invaluable knowledge that cannot be obtained by reading individual data items alone.

For example, Google's *Flu Trends* uses specific search terms as indicators of flu activity. It found a close relationship between the number of people who search for flu-related information and the number of people who actually have flu symptoms. A pattern emerges when all of the search queries related to flu are aggregated. Using aggregated Google search data, *Flu Trends* can estimate flu activity up to two weeks faster than traditional systems can.

This example shows how data mining can turn a large collection of data into knowledge that can help meet a challenge.

Evolution of Database Technology

1960s:

- Data collection, database creation, IMS and network DBMS

1970s:

- Relational data model, relational DBMS implementation

1980s:

- RDBMS, advanced data models (extended-relational, OO, deductive, etc.)
- Application-oriented DBMS (spatial, scientific, engineering, etc.)

1990s:

- Data mining, data warehousing, multimedia databases, and Web databases

2000s

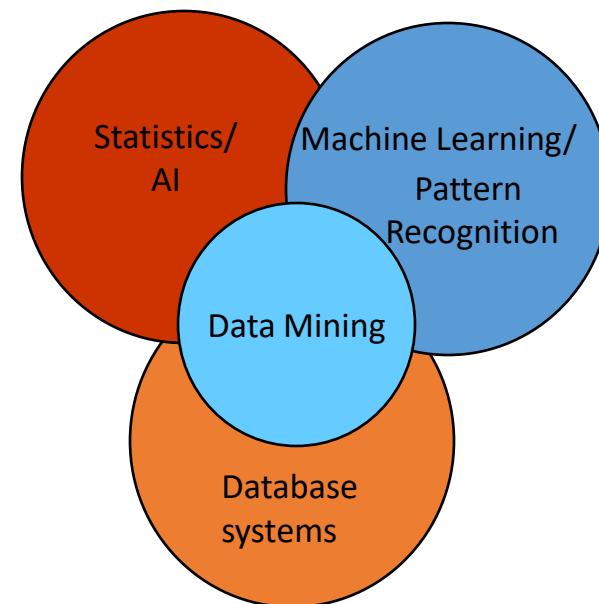
- Stream data management and mining
- Data mining and its applications
- Web technology (XML, data integration) and global information systems

Origins of Data Mining

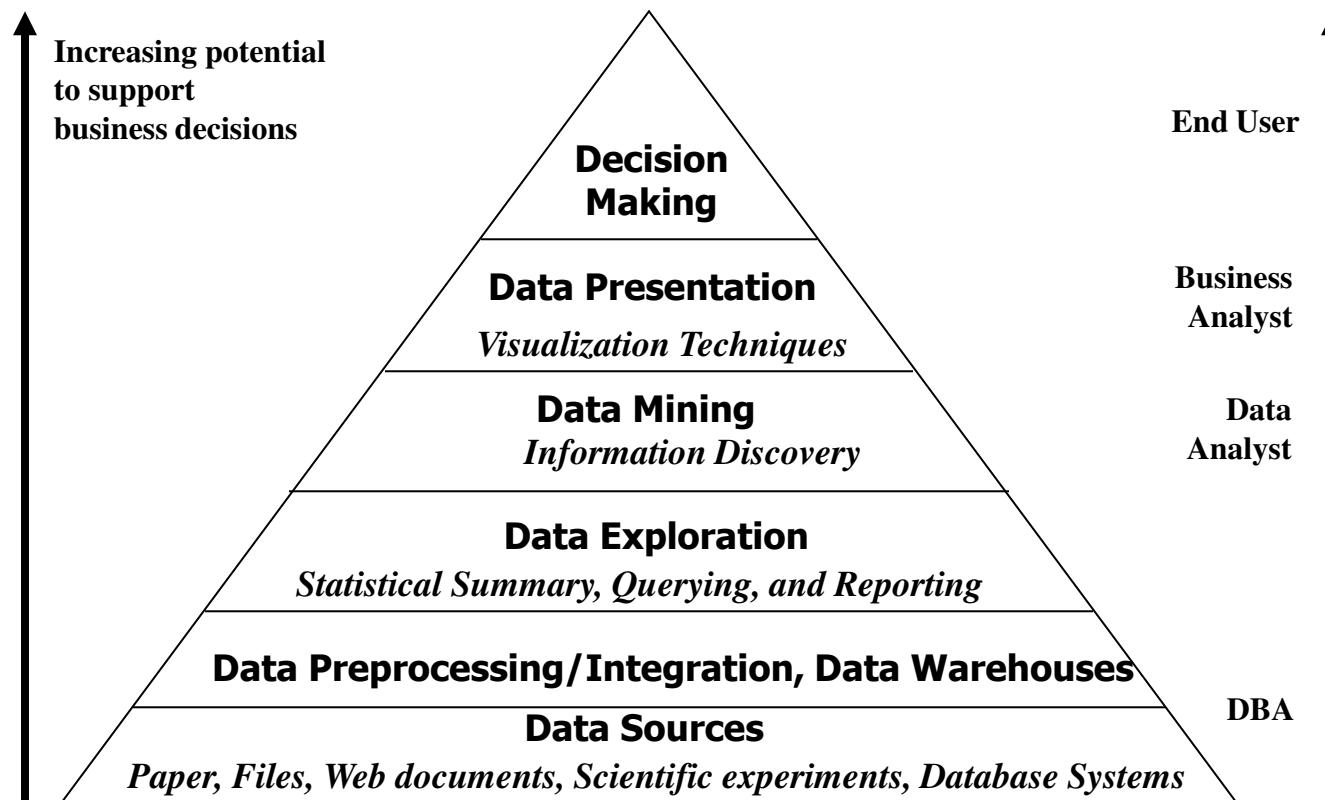
Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems

Traditional Techniques
may be unsuitable due to

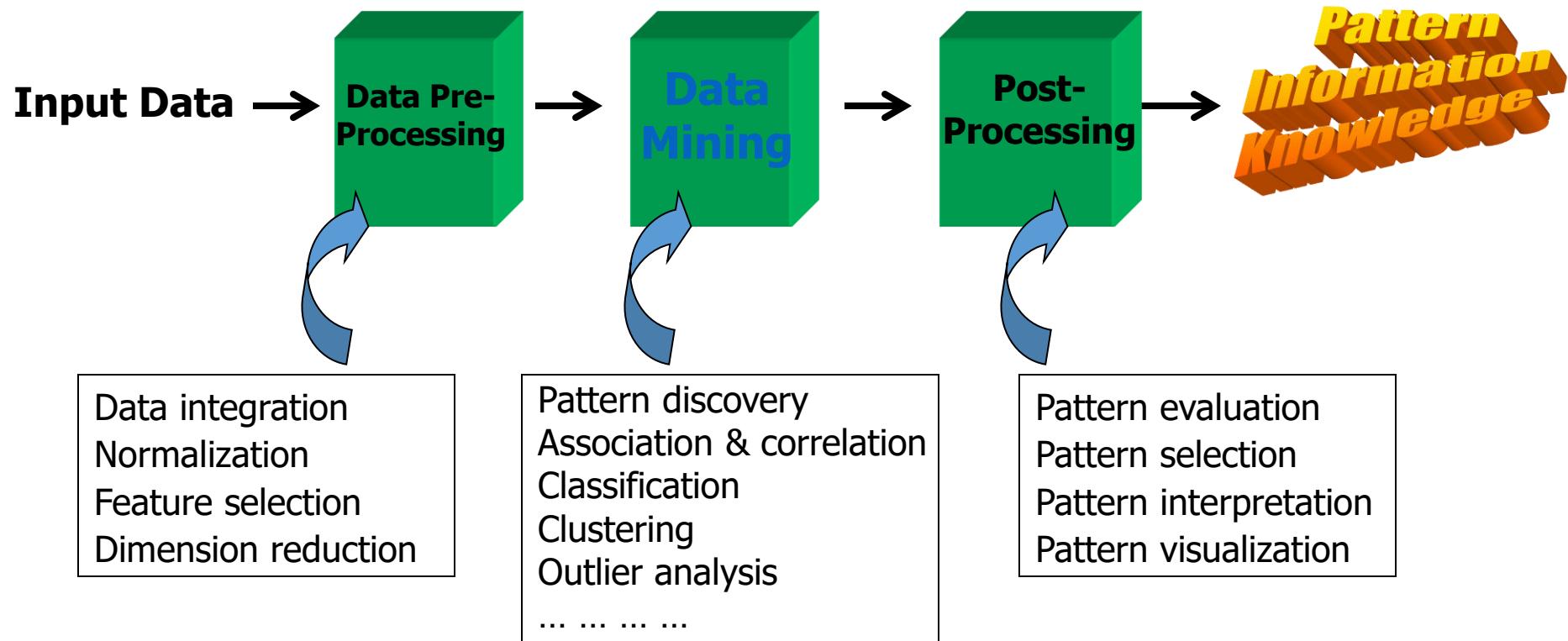
- Enormity of data
- High dimensionality of data
- Heterogeneous, distributed nature of data



Data Mining in Business Intelligence



Data Mining/KDD Process



KDD – Knowledge Discovery in Databases

Data Mining & Machine Learning

According to Tom M. Mitchell, Chair of Machine Learning at Carnegie Mellon University and author of the book *Machine Learning* (McGraw-Hill),

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with the experience E.

We now have a set of objects to define machine learning:

Task (T), Experience (E), and Performance (P)

With a computer running a set of tasks, the experience should be leading to performance increases (to satisfy the definition)

Many data mining tasks are executed successfully with help of machine learning

Multi-Dimensional View of Data Mining

Data to be mined

- Database data (extended-relational, object-oriented, heterogeneous, legacy), data warehouse, transactional data, stream, spatiotemporal, time-series, sequence, text and web, multi-media, graphs & social and information networks

Knowledge to be mined (or: Data mining functions)

- Characterization, discrimination, association, classification, clustering, trend/deviation, outlier analysis, etc.
- Descriptive vs. predictive data mining
- Multiple/integrated functions and mining at multiple levels

Techniques utilized

- Data-intensive, data warehouse (OLAP), machine learning, statistics, pattern recognition, visualization, high-performance, etc.

Applications adapted

- Retail, telecommunication, banking, fraud analysis, bio-data mining, stock market analysis, text mining, Web mining, etc.

Data Mining on Diverse kinds of Data

Besides relational database data (from operational or analytical systems), there are many other kinds of data that have diverse forms and structures and different semantic meanings.

Examples of data can be :

time-related or sequence data (e.g., historical records, stock exchange data, and time-series and biological sequence data),

data streams (e.g., video surveillance and sensor data, which are continuously transmitted),

spatial data (e.g., maps),

engineering design data (e.g., the design of buildings, system components, or integrated circuits),

hypertext and multimedia data (including text, image, video, and audio data),

graph and networked data (e.g., social and information networks), and

the Web (a widely distributed information repository).

Diversity of data brings in new challenges such as handling special structures (e.g., sequences, trees, graphs, and networks) and specific semantics (such as ordering, image, audio and video contents, and connectivity)

Data Mining Activities

Data Mining Tasks

Prediction Methods

- Use some variables to predict unknown or future values of other variables.

Description Methods

- Find human-interpretable patterns that describe the data.

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

Experts have more terms:

Gartner Analyst View:

https://twitter.com/doug_laney/status/611172882882916352

SCM Expert View:

<https://elsekuipers.files.wordpress.com/2014/08/scm-analytics1.png>

Data Mining Tasks...

Classification [Predictive]

Clustering [Descriptive]

Association Rule Discovery [Descriptive]

Sequential Pattern Discovery [Descriptive]

Regression [Predictive]

Deviation Detection [Predictive]

Classification: Definition

Given a collection of records (*training set*)

- Each record contains a set of *attributes*, one of the attributes is the *class*.

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

Goal: previously unseen records should be assigned a class as accurately as possible.

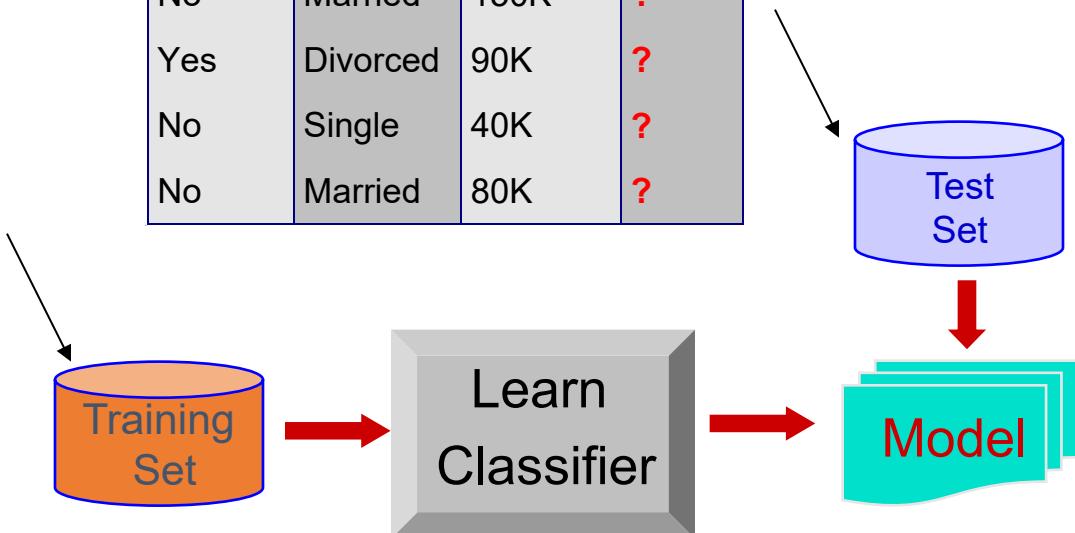
- A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

Classification 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

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.
 - Use this information as input attributes to learn a classifier model.

From [Berry & Linoff] Data Mining Techniques, 1997

Classification: Application 2

Fraud Detection

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

Classification: Application 3

Customer Attrition/Churn:

- Goal: To predict whether a customer is likely to be lost to a competitor.
- Approach:
 - 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.
 - Label the customers as loyal or disloyal.
 - Find a model for loyalty.

From [Berry & Linoff] Data Mining Techniques, 1997

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:

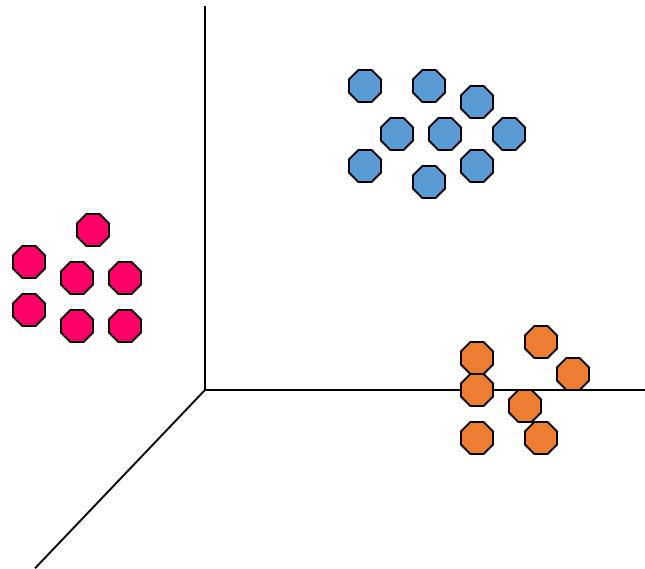
- Euclidean Distance if attributes are continuous.
- Other Problem-specific Measures.

Illustrating Clustering

Intracluster distances
are minimized

Intercluster distances
are maximized

Euclidean Distance Based
Clustering in 3-D space



Clustering: Application 1

Market Segmentation:

- Goal: subdivide a market into distinct subsets of customers where any subset may conceivably be selected as a market target to be reached with a distinct marketing mix.
- Approach:
 - Collect different attributes of customers based on their geographical and lifestyle related information.
 - Find clusters of similar customers.
 - Measure the clustering quality by observing buying patterns of customers in same cluster vs. those from different clusters.

Clustering: Application 2

Document Clustering:

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

Association Rule Discovery: Definition

Given a set of records each of which contain some number of items from a given collection;

- Produce dependency rules which will predict occurrence of an item based on occurrences of other items.

Example of Association Rules

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

$\{\text{Diaper}\} \rightarrow \{\text{Butter}\}$,
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Beans, Coke}\}$,
 $\{\text{Butter, Bread}\} \rightarrow \{\text{Milk}\}$,

Association Rule Discovery: Application 1

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

Inventory Management:

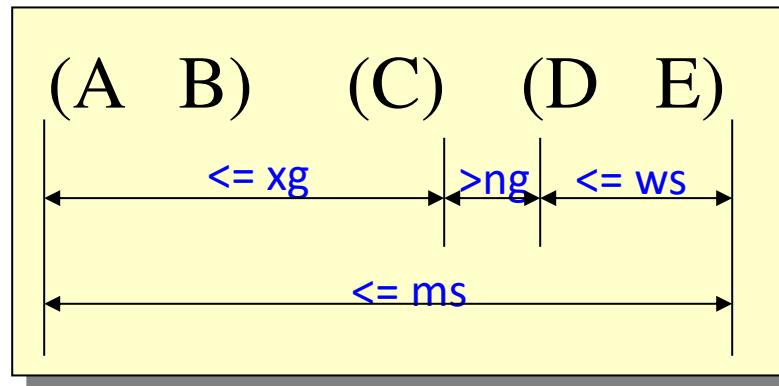
- Goal: A consumer appliance repair company wants to anticipate the nature of repairs on its consumer products and keep the service vehicles equipped with right parts to reduce on number of visits to consumer households.
- Approach: Process the data on tools and parts required in previous repairs at 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.

$$(A \ B) \quad (C) \longrightarrow (D \ E)$$

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



Timing constraints include maxgap (xg), mingap (ng), windowsize (ws), maxspan (ms)

Sequential Pattern Discovery: Examples

In telecommunications alarm logs,

- (Inverter_Problem Excessive_Line_Current)
(Rectifier_Alarm) --> (Fire_Alarm)

In point-of-sale transaction sequences,

- Computer Bookstore:
(Intro_To_Visual_C) (C++_Primer) -->
(Perl_for_dummies,Tcl_Tk)
- Athletic Apparel Store:
(Shoes) (Racket, Racketball) --> (Sports_Jacket)

Prediction/Regression

Predict a value of a given continuous valued variable based on the values of other variables, assuming a linear or nonlinear model of dependency.

Greatly studied in statistics, neural network fields.

Examples:

- Predicting sales amounts of new product based on advertising expenditure.
- Predicting wind velocities as a function of temperature, humidity, air pressure, etc.
- Time series prediction of stock market indices.

Deviation/Anomaly Detection

Detect significant deviations from normal behavior

Applications:

- Credit Card Fraud Detection
- Network Intrusion Detection

Gartner's Magic Quadrant

DM Process & Challenges

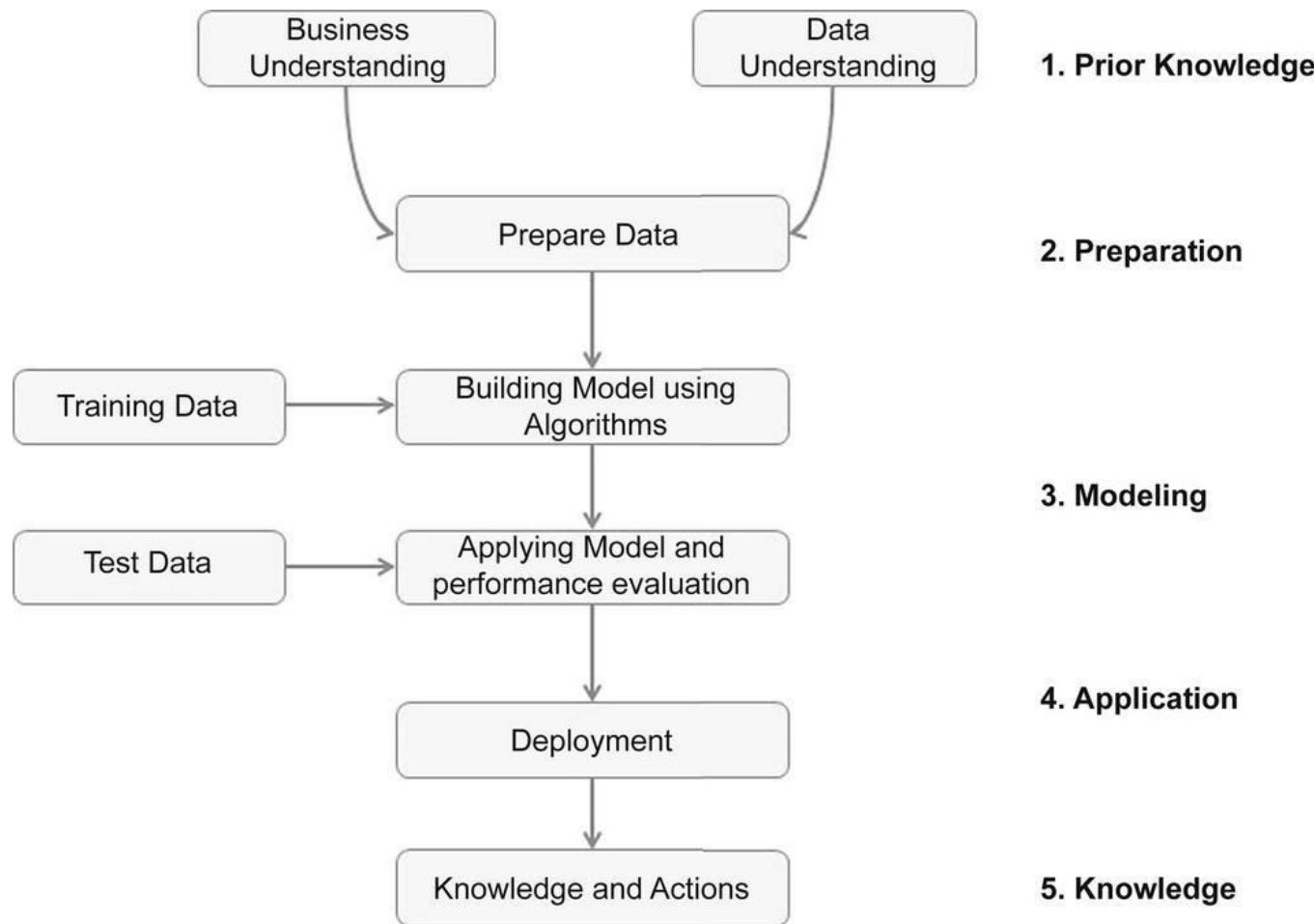
DM Process

The standard data mining process involves

1. understanding the problem,
2. preparing the data (samples),
3. developing the model,
4. applying the model on a data set to see how the model may work in real world, and
5. production deployment.

A popular data mining process frameworks is CRISP-DM (Cross Industry Standard Process for Data Mining). This framework was developed by a consortium of companies involved in data mining

Generic Data Mining Process



Prior Knowledge

Data Mining tools/solutions identify hidden patterns.

- Generally we get many patterns
- Out of them many could be false or trivial.
- Filtering false patterns requires domain understanding.

Understanding how the data is collected, stored, transformed, reported, and used is essential.

Data Preparation

Data needs to be understood. It requires descriptive statistics such as mean, median, mode, standard deviation, and range for each attribute

Data quality is an ongoing concern wherever data is collected, processed, and stored.

- The data cleansing practices include elimination of duplicate records, quarantining outlier records that exceed the bounds, standardization of attribute values, substitution of missing values, etc.
- it is critical to check the data using data exploration techniques in addition to using prior knowledge of the data and business before building models to ensure a certain degree of data quality

Missing Values

- Need to track the data lineage of the data source to find right solution

Data Types and Conversion

- The attributes in a data set can be of different types, such as continuous numeric (interest rate), integer numeric (credit score), or categorical
- data mining algorithms impose different restrictions on what data types they accept as inputs

Transformation

- Can go beyond type conversion, may include dimensionality reduction or numerosity reduction

Outliers are anomalies in the data set

- May occur legitimately or erroneously.

Feature Selection

- Many data mining problems involve a data set with hundreds to thousands of attributes, most of which may not be helpful. Some attributes may be correlated, e.g. sales amount and tax.

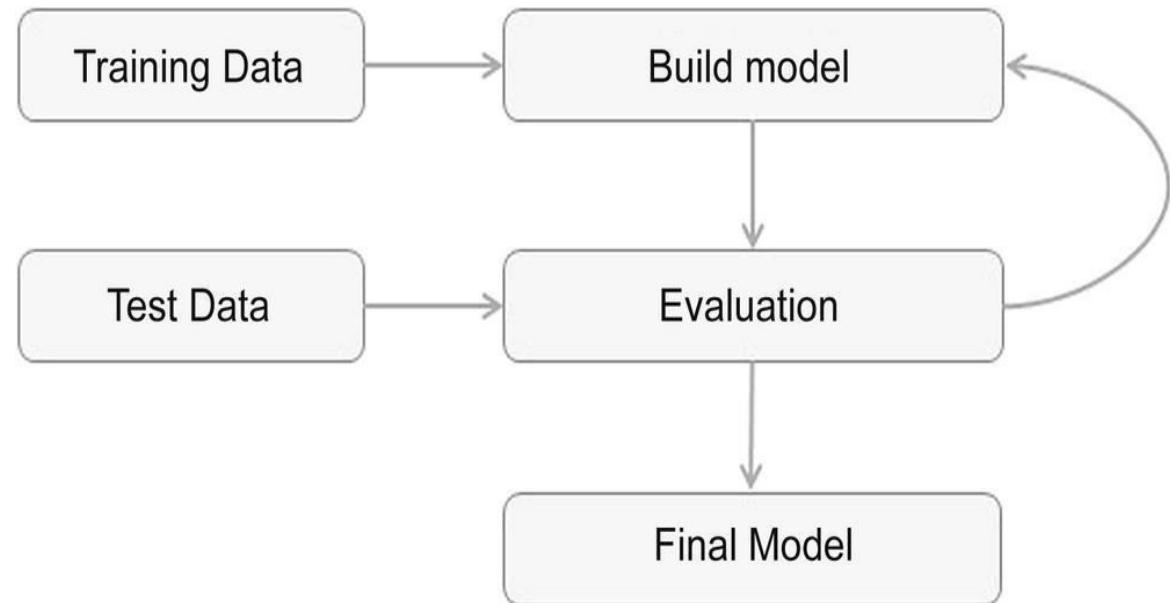
Data Sampling may be adequate in many cases

Modeling & Evaluation

A model is the abstract representation of the data and its relationships in a given data set.

Data mining models can be classified into the following categories: classification, regression, association analysis, clustering, and outlier or anomaly detection.

Each category has a few dozen different algorithms; each takes a slightly different approach to solve the problem at hand



Application

The model deployment stage considerations:

- assessing model readiness, technical integration, response time, model maintenance, and assimilation

Production Readiness

- Real-time response capabilities, and other business requirements

Technical Integration

- Use of modeling tools (e.g. RapidMiner), Use of PMML for portable and consistent format of model description, integration with other tools

Timeliness

- The trade-offs between production responsiveness and build time need to be considered

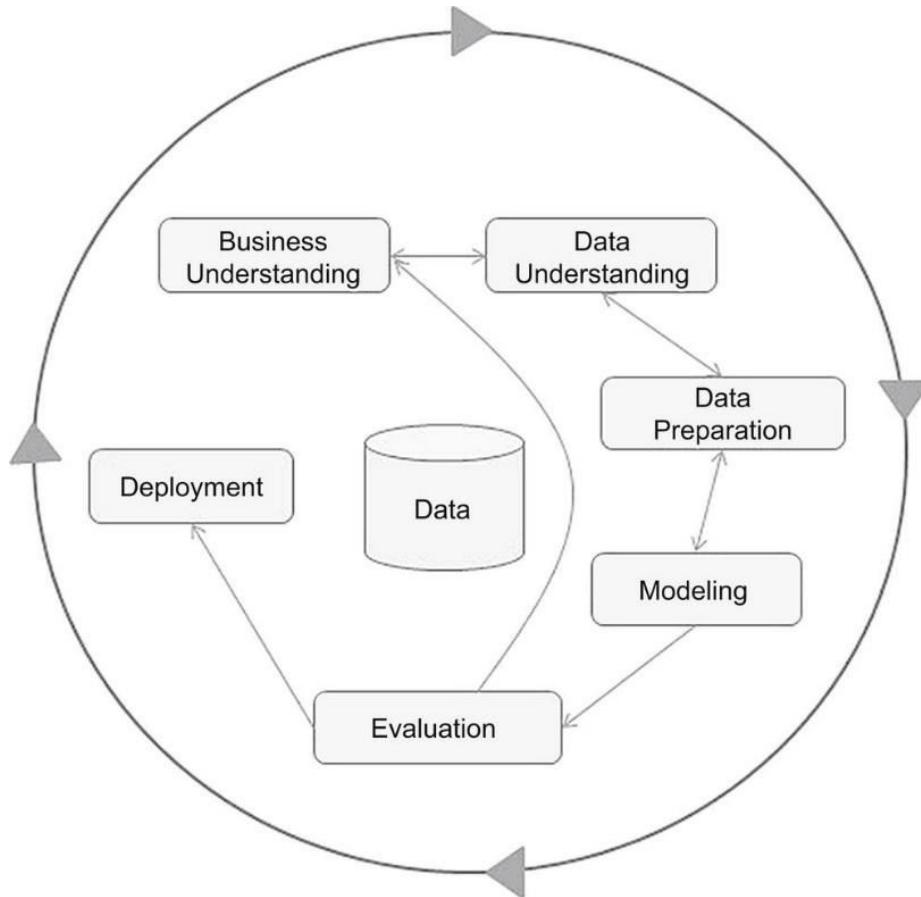
Remodeling

- The conditions in which the model is built may change after deployment

Assimilation

- The challenge is to assimilate the knowledge gained from data mining in the organization. For example, the objective may be finding logical clusters in the customer database so that separate treatment can be provided to each customer cluster.

CRISP data mining framework



CRISP is the most popular methodology for analytics, data mining, and data science projects, with 43% share as per 2014 KDnuggets Poll.

CRISP-DM was conceived in 1996. In 1997 it got underway as a European Union project, led by SPSS, Teradata, Daimler AG, NCR Corporation and OHRA.

DM Issues/Challenges

DM Issues/Challenges – Mining Methodology

Mining various and new kinds of knowledge

Mining knowledge in multidimensional space

Data mining—an interdisciplinary effort

Boosting the power of discovery in a networked environment

Handling uncertainty, noise, or incompleteness of data

Pattern evaluation and pattern- or constraint-guided mining

DM Issues/Challenges

DM Issues/Challenges – User Interaction

Interactive mining

Incorporation of background knowledge

Ad hoc data mining and data mining query languages

Presentation and visualization of data mining results

DM Issues/Challenges - Efficiency and Scalability

Efficiency and scalability of data mining algorithms

Parallel, distributed, and incremental mining algorithms

Cloud computing and cluster computing

DM Issues/Challenges

DM Issues/Challenges - Diversity of Database Types

Handling complex types of data

Mining dynamic, networked, and global data repositories

DM Issues/Challenges - Society

Social impacts of data mining

Privacy-preserving data mining

Invisible data mining

Text Books

	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
R1	Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner by Vijay Kotu and Bala Deshpande Morgan Kaufmann Publishers

Thank You

Source: <https://stackoverflow.com/>

	Database	Data Mart	Data Warehouse	Data Lake
Source	Single	Single	Multiple	Multiple
Structure	Structured	Structured	Structured	Unstructured
Purpose	Determined	Determined	Determined	Undetermined
Storage	Centralized	Decentralized	Centralized	Centralized
Granularity	Detailed	Summarized	Detailed & Summary	All
Flexibility	Low	Medium	Medium	High
Primary Use	Transactional	Reporting	Analytics & Reporting	Analytics
Data Volume	Low	Low	Medium	High
Development	Top-down	Bottom-up	Top-down	All
Design Time	Medium	Medium	High	Low
Volatility	Medium	Low	None	None
Data Operations	CRUD	CR	CRU	CR
Subject Area	Single	Single	Multiple	Multiple
Design Schema	Relational	Multi-dimensional	Multi-dimensional	No Schema



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

S2-21_DSECLZC415

Data Pre-processing



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

Data Preprocessing Concepts

Preprocessing Objectives

- To improve data quality
- To modify data to better fit specific data mining technique

Major Tasks in Data Preprocessing

- **Data cleaning**
 - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
 - Integration of multiple databases, data cubes, or files
- **Data reduction**
 - Dimensionality reduction
 - Numerosity reduction
 - Data compression
- **Data transformation and data discretization**
 - Normalization
 - Concept hierarchy generation

Data Quality: Multidimensional View

- Measures for data quality: A multidimensional view
 - Accuracy: correct or wrong, accurate or not
 - Completeness: not recorded, unavailable, ...
 - Consistency: some modified but some not, dangling, ...
 - Timeliness: timely update?
 - Believability: how trustable the data are correct?
 - Interpretability: how easily the data can be understood?

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

Data Cleaning

Data Cleaning

- Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error
 - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 - e.g., *Occupation* = “ ” (missing data)
 - noisy: containing noise, errors, or outliers
 - e.g., *Salary* = “-10” (an error)
 - inconsistent: containing discrepancies in codes or names, e.g.,
 - *Age* = “42”, *Birthday* = “03/07/2010”
 - Was rating “1, 2, 3”, now rating “A, B, C”
 - discrepancy between duplicate records
 - Intentional (e.g., *disguised missing data*)
 - Jan. 1 as everyone’s birthday?

Incomplete (Missing) Data

- Data is not always available
 - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
 - equipment malfunction
 - inconsistent with other recorded data and thus deleted
 - data not entered due to misunderstanding
 - certain data may not be considered important at the time of entry
 - not register history or changes of the data
- Missing data may need to be inferred

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
 - a global constant : e.g., “unknown”, a new class?!
 - the attribute mean
 - the attribute mean for all samples belonging to the same class: smarter
 - the most probable value: inference-based such as Bayesian formula or decision tree

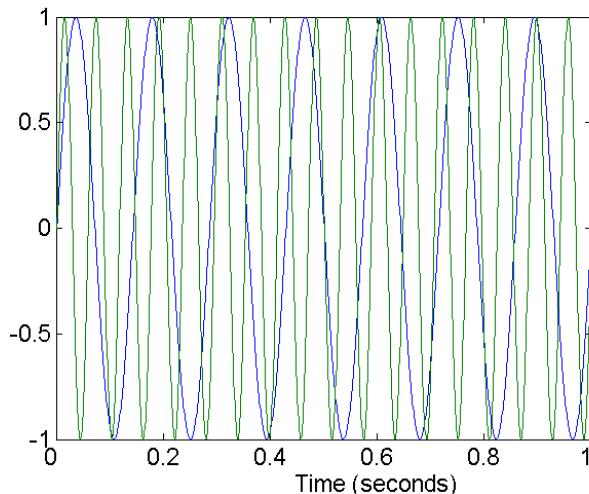
Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
 - faulty data collection instruments
 - data entry problems
 - data transmission problems
 - technology limitation
 - inconsistency in naming convention
- Other data problems which require data cleaning
 - duplicate records
 - incomplete data
 - inconsistent data

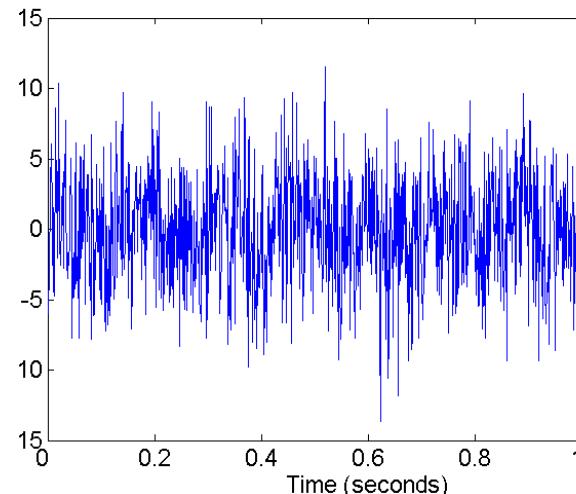
Noise

Noise refers to modification of original values

- Examples: distortion of a person's voice when talking on a poor phone and “snow” on television screen



Two Sine Waves



Two Sine Waves + Noise

How to Handle Noisy Data?

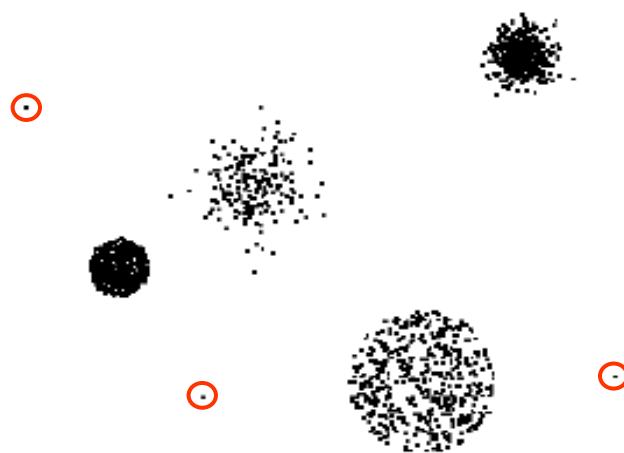
- Binning (also used for discretization)
 - first sort data and partition into (equal-frequency) bins
 - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
 - Binning methods smooth a sorted data value by consulting its "neighborhood," that is, the values around it, i.e. they perform *local* smoothing.
- Regression
 - smooth by fitting the data into regression functions
- Clustering
 - detect and remove outliers
- Combined computer and human inspection
 - detect suspicious values and check by human (e.g., deal with possible outliers)

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

Outliers

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



Data Cleaning as a Process

- Data discrepancy detection
 - Use metadata (e.g., domain, range, dependency, distribution)
 - Check field overloading
 - Check uniqueness rule, consecutive rule and null rule
 - Use commercial tools
 - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
 - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)
- Data migration and integration
 - Data migration tools: allow transformations to be specified
 - ETL (Extraction/Transformation>Loading) tools: allow users to specify transformations through a graphical user interface
- Integration of the two processes
 - Iterative and interactive (e.g., Potter's Wheels)



Data Preprocessing Techniques

Major Tasks in Data Preprocessing

- **Data cleaning**
 - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
 - Integration of multiple databases, data cubes, or files
- **Data reduction**
 - Dimensionality reduction
 - Numerosity reduction
 - Data compression
- **Data transformation and data discretization**
 - Normalization
 - Concept hierarchy generation

Data Integration

- **Data integration:** Combines data from multiple sources into a coherent store
 - Schema integration: e.g., A.cust-id \equiv B.cust-#
 - Integrate metadata from different sources
 - Entity identification problem:
 - Identify real world entities from multiple data sources, e.g., Bill Clinton = William Clinton
 - Detecting and resolving data value conflicts
 - For the same real world entity, attribute values from different sources are different
 - Possible reasons: different representations, different scales, e.g., metric vs. British units

Any problems with the Data?

Name	Age	DateOfFirstBuy	Profession	DateOfBirth
Bill Gates	34	15-Jan-2015	MGR	Feb 24, 1981
John	39	27-Jan-2015		Mar 11, 1982
Anjaneya	57	29-Feb-2011	PROF	Apr 17, 1965
William Gates	34	15-Jan-2015	MGR	Feb 24, 1981
Kennedy	39	30-Jan-2015	DOC	Nov 03,1982

Any problems with the Data?

Name	Age	DateOfFirstBuy	Profession	DateOfBirth
Bill Gates	34	15-Jan-2015	MGR	Feb 24, 1981
John	39	27-Jan-2015		Mar 11, 1982
Anjaneya	57	29-Feb-2011	PROF	Apr 17, 1965
William Gates	34	15-Jan-2015	MGR	Feb 24, 1981
Kennedy	39	30-Jan-2015	DOC	Nov 03,1982

- 1) Missing values in Profession column
- 2) Format of DateOfFirstBuy and DateOfBirth are different, needs standardization. Invalid date value
- 3) Row 1 and Row 3 are potentially duplicate data.
- 4) Both Age and DateOfBirth are stored. Age is derived attribute.
- 5) Inconsistent format for name, missing first or last names
- 6) Entity identification issues

Handling Redundancy

Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
 - *Object identification:* The same attribute or object may have different names in different databases
 - *Derivable data:* One attribute may be a “derived” attribute in another table, e.g., annual revenue
- Redundant attributes may be able to be detected by *correlation analysis* and *covariance analysis*
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Correlation Analysis (Numeric Data)

- Correlation coefficient (also called **Pearson's product moment coefficient**)

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

where n is the number of tuples, \bar{A} and \bar{B} are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B.

- $+1 \geq r_{A,B} \geq -1$
- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
- $r_{A,B} = 0$: independent;
- $r_{A,B} < 0$: negatively correlated

Correlation (viewed as linear relationship)

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

$$a'_k = (a_k - \text{mean}(A)) / \text{std}(A)$$

$$b'_k = (b_k - \text{mean}(B)) / \text{std}(B)$$

$$\text{correlation}(A, B) = A' \bullet B'$$

Covariance (Numeric Data)

- Covariance is similar to correlation

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

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

where n is the number of tuples, \bar{A} and \bar{B} are the respective mean or **expected values** of A and B, σ_A and σ_B are the respective standard deviation of A and B

- **Positive covariance:** If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values
- **Negative covariance:** If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value
- **Independence:** $Cov_{A,B} = 0$

Co-Variance: An Example

- It can be simplified in computation as

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

- Suppose two stocks A and B have the following values in one week: (2, 5), (3, 8), (5,

10), (4, 11), (6, 14).

$$Cov(A, B) = E(A \cdot B) - \bar{A}\bar{B}$$

- Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?

- $E(A) = (2 + 3 + 5 + 4 + 6) / 5 = 20/5 = 4$

- $E(B) = (5 + 8 + 10 + 11 + 14) / 5 = 48/5 = 9.6$

- $Cov(A, B) = (2 \times 5 + 3 \times 8 + 5 \times 10 + 4 \times 11 + 6 \times 14) / 5 - 4 \times 9.6 = 4$

- Thus, A and B rise together since $Cov(A, B) > 0$.

Correlation Analysis (Nominal Data)

- χ^2 (chi-square) test

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

- The larger the χ^2 (chi-square) value, the more likely the variables are related
- The cells that contribute the most to the χ^2 value are those whose actual count is very different from the expected count
- Correlation does not imply causality
 - # of hospitals and # of car-theft in a city are correlated
 - Both are causally linked to the third variable: population

Chi-Square Calculation: An Example

	Play chess	Not play chess	Sum (row)
Like science fiction	250	200	450
Not like science fiction	50	1000	1050
Sum(col.)	300	1200	1500

Do you think there is correlation between 'Play chess' and 'Like science fiction'

For the above cross-tabulation

- There are two possible values in rows: Likes science fiction, Not like science fiction
- There are two possible values in columns: Play chess, Not play chess

$$\text{Degrees of Freedom} = (\text{rows}-1) * (\text{cols}-1) = (2-1)(2-1) = 1$$

Chi-Square Calculation: Computation

	Play chess	Not play chess	Sum (row)
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

- χ^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

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

- We can check statistical significance of chi-square value in standard (χ^2) table.
- It shows that like_science_fiction and play_chess are correlated in the dataset

	P	0.995	0.975	0.2	0.1	0.05	0.025	0.02	0.01	0.005	0.002	0.001
DF	.0004	.00016	1.642	2.706	3.841	5.024	5.412	6.635	7.879	9.55	10.828	
1	0.01	0.0506	3.219	4.605	5.991	7.378	7.824	9.21	10.597	12.429	13.816	
3	0.0717	0.216	4.642	6.251	7.815	9.348	9.837	11.345	12.838	14.796	16.266	
4	0.207	0.484	5.989	7.779	9.488	11.143	11.668	13.277	14.86	16.924	18.467	
5	0.412	0.831	7.289	9.236	11.07	12.833	13.388	15.086	16.75	18.907	20.515	
6	0.676	1.237	8.558	10.645	12.592	14.449	15.033	16.812	18.548	20.791	22.458	
7	0.989	1.69	9.803	12.017	14.067	16.013	16.622	18.475	20.278	22.601	24.322	
8	1.344	2.18	11.03	13.362	15.507	17.535	18.168	20.09	21.955	24.352	26.124	
9	1.735	2.7	12.242	14.684	16.919	19.023	19.679	21.666	23.589	26.056	27.877	
10	2.156	3.247	13.442	15.987	18.307	20.483	21.161	23.209	25.188	27.722	29.588	
11	2.603	3.816	14.631	17.275	19.675	21.92	22.618	24.725	26.757	29.354	31.264	
12	3.074	4.404	15.812	18.549	21.026	23.337	24.054	26.217	28.3	30.957	32.909	
13	3.565	5.009	16.985	19.812	22.362	24.736	25.472	27.688	29.819	32.535	34.528	
14	4.075	5.629	18.151	21.064	23.685	26.119	26.873	29.141	31.319	34.091	36.123	
15	4.601	6.262	19.311	22.307	24.996	27.488	28.259	30.578	32.801	35.628	37.697	
16	5.142	6.908	20.465	23.542	26.296	28.845	29.633	32	34.267	37.146	39.252	
17	5.697	7.564	21.615	24.769	27.587	30.191	30.995	33.409	35.718	38.648	40.79	
18	6.265	8.231	22.76	25.989	28.869	31.526	32.346	34.805	37.156	40.136	42.312	
19	6.844	8.907	23.9	27.204	30.144	32.852	33.687	36.191	38.582	41.61	43.82	
20	7.434	9.591	25.038	28.412	31.41	34.17	35.02	37.566	39.997	43.072	45.315	

Normalization, Discretization

Normalization

- **Min-max normalization:** to $[new_min_A, new_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A$$

- Ex. Let income range \$12,000 to \$98,000 normalized to [0.0, 1.0]. Then
\$73,600 is mapped to $\frac{73,600 - 12,000}{98,000 - 12,000} (1.0 - 0) + 0 = 0.716$

- **Z-score normalization** (μ : mean, σ : standard deviation):

$$v' = \frac{v - \mu_A}{\sigma_A}$$

- Ex. Let $\mu = 54,000$, $\sigma = 16,000$. Then $\frac{73,600 - 54,000}{16,000} = 1.225$

- **Normalization by decimal scaling**

$$v' = \frac{v}{10^j} \quad \text{Where } j \text{ is the smallest integer such that } \text{Max}(|v'|) < 1$$

Discretization

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

Data Discretization Methods

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

Simple Discretization: Binning

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

Binning Methods for Data Smoothing

Sorted data for *price* (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15

Bin 2: 21, 21, 24

Bin 3: 25, 28, 34

Smoothing by bin means:

Bin 1: 9, 9, 9

Bin 2: 22, 22, 22

Bin 3: 29, 29, 29

Smoothing by bin boundaries:

Bin 1: 4, 4, 15

Bin 2: 21, 21, 24

Bin 3: 25, 25, 34

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
 - Details to be covered in Chapter “Classification”
- 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

Data Reduction

Data Reduction Strategies

- **Data reduction:** Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Data reduction strategies
 - Dimensionality reduction, e.g., remove unimportant attributes
 - Wavelet transforms
 - Principal Components Analysis (PCA)
 - Feature subset selection, feature creation
 - Numerosity reduction (some simply call it: Data Reduction)
 - Regression and Log-Linear Models
 - Histograms, clustering, sampling
 - Data cube aggregation
 - Data compression

Data Reduction : Dimensionality Reduction

- **Curse of dimensionality**

- When dimensionality increases, data becomes increasingly sparse
- Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
- The possible combinations of subspaces will grow exponentially

- **Dimensionality reduction**

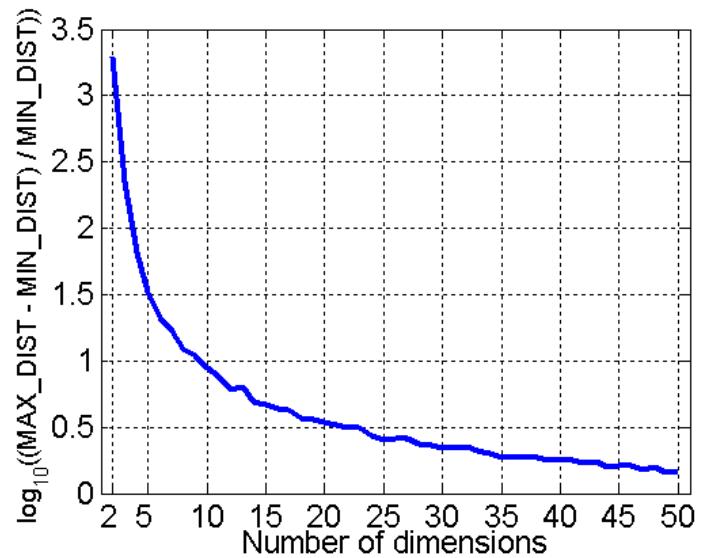
- Avoid the curse of dimensionality
- Help eliminate irrelevant features and reduce noise
- Reduce time and space required in data mining
- Allow easier visualization

- **Dimensionality reduction techniques**

- Wavelet transforms
- Principal Component Analysis
- Supervised and nonlinear techniques (e.g., feature selection)

Curse of Dimensionality

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

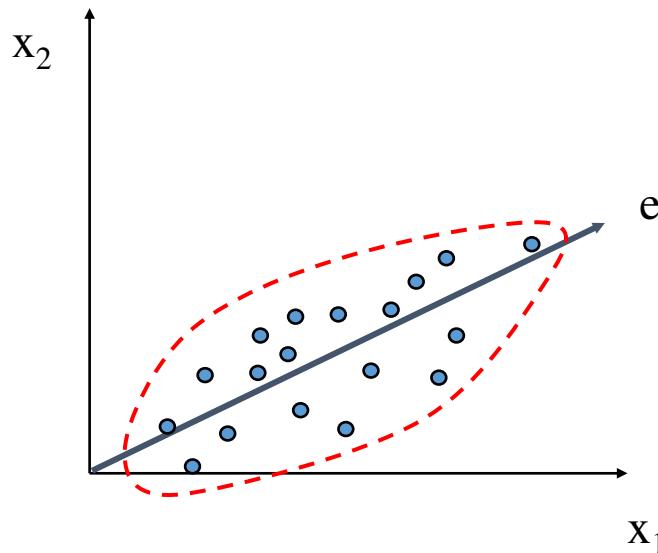


- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points

Principal Component Analysis (PCA)

Find a projection that captures the largest amount of variation in data

The original data are projected onto a much smaller space, resulting in dimensionality reduction.



Principal Component Analysis (Steps)

- Given N data vectors from n -dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
 - Normalize input data: Each attribute falls within the same range
 - Compute k orthonormal (unit) vectors, i.e., *principal components*
 - Each input data (vector) is a linear combination of the k principal component vectors
 - The principal components are sorted in order of decreasing “significance” or strength
 - Since the components are sorted, the size of the data can be reduced by eliminating the *weak components*, i.e., those with low variance (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data)
- Works for numeric data only

Attribute Subset Selection

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

Heuristic Search in Attribute Selection

- There are 2^d possible attribute combinations of d attributes
- Typical heuristic attribute selection methods:
 - Best single attribute under the attribute independence assumption: choose by significance tests
 - Best step-wise feature selection:
 - The best single-attribute is picked first
 - Then next best attribute condition to the first, ...
 - Step-wise attribute elimination:
 - Repeatedly eliminate the worst attribute
 - Best combined attribute selection and elimination
 - Optimal branch and bound:
 - Use attribute elimination and backtracking

Attribute Creation (Feature Generation)

- Create new attributes (features) that can capture the important information in a data set more effectively than the original ones
- Three general methodologies
 - Attribute extraction
 - Domain-specific
 - Mapping data to new space (see: data reduction)
 - E.g., Fourier transformation, wavelet transformation
 - Attribute construction
 - Combining features
 - Data discretization

Data Reduction: Numerosity Reduction

- Reduce data volume by choosing alternative, *smaller forms* of data representation
- **Parametric methods** (e.g., regression)
 - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
 - Ex.: Log-linear models—obtain value at a point in m -D space as the product on appropriate marginal subspaces
- **Non-parametric** methods
 - Do not assume models
 - Major families: histograms, clustering, sampling, ...

Sampling

Sampling: obtaining a small sample s to represent the whole data set N

Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data

Key principle: Choose a **representative** subset of the data

- Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods, e.g., stratified sampling:

Note: Sampling may not reduce database I/Os (page at a time)

Types of Sampling

Simple random sampling

- There is an equal probability of selecting any particular item

Sampling without replacement

- Once an object is selected, it is removed from the population

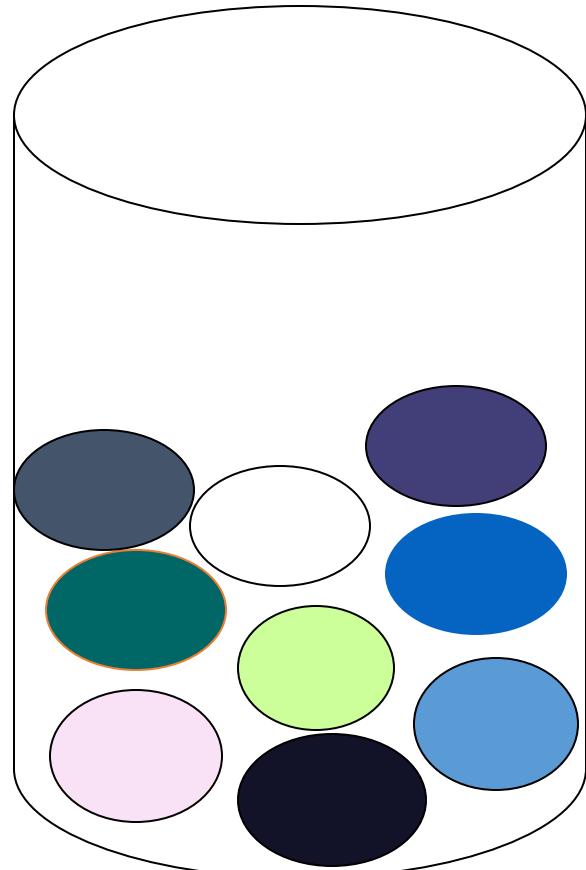
Sampling with replacement

- A selected object is not removed from the population

Stratified sampling:

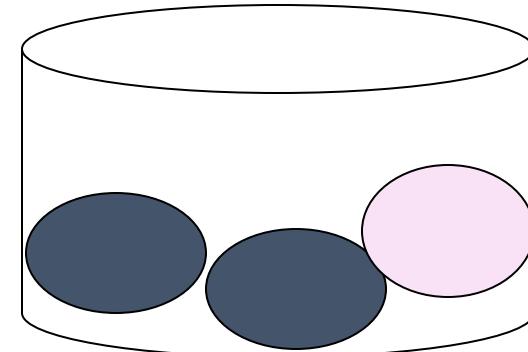
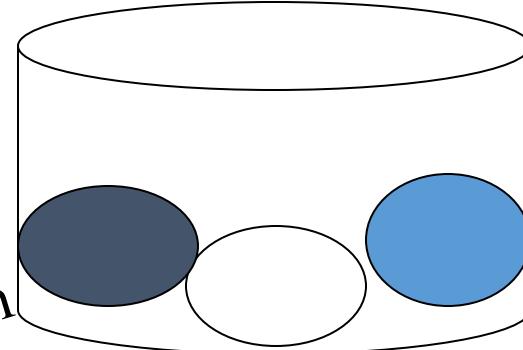
- Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)
- Used in conjunction with skewed data

Sampling: With or without Replacement



SRSWOR
(simple random
sample without
replacement)

SRSWR



Raw Data

Text Books

Author(s), Title, Edition, Publishing House	
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers

Thank You



S2-21_DSECLZC415

Data Exploration

BITS Pilani

Pilani|Dubai|Goa|Hyderabad



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

Agenda

- Data objects and Attributes types
- Basic Statistical Descriptions of Data
- Measuring Data Similarity and Dissimilarity

Data Description

Types of Data Sets

- Record
 - Relational records
 - Data matrix, e.g., numerical matrix, crosstabs
 - Document data: text documents: term-frequency vector
 - Transaction data
- Graph and network
 - World Wide Web
 - Social or information networks
 - Molecular Structures
- Ordered
 - Video data: sequence of images
 - Temporal data: time-series
 - Sequential Data: transaction sequences
 - Genetic sequence data
- Spatial, image and multimedia:
 - Spatial data: maps
 - Image data:
 - Video data:

	team	coach	play	ball	score	game	winnings	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

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

Important Characteristics of Structured Data

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

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.

Attributes

- **Attribute (or dimensions, features, variables)**: a data field, representing a characteristic or feature of a data object.
 - *E.g., customer_ID, name, address*
- Types:
 - Nominal
 - Binary
 - Numeric: quantitative
 - Interval-scaled
 - Ratio-scaled

Attribute Types

- **Nominal:** categories, states, or “names of things”
 - $Hair_color = \{auburn, black, blond, brown, grey, red, white\}$
 - 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**
 - Values have a meaningful order (ranking) but magnitude between successive values is not known.
 - $Size = \{small, medium, large\}$, grades, army rankings

Numeric Attribute Types

- Quantity (integer or real-valued)
- **Interval**
 - Measured on a scale of **equal-sized units**
 - Values have order
 - E.g., *temperature in C° or F°, calendar dates*
 - No true zero-point
- **Ratio**
 - Inherent **zero-point**
 - We can speak of values as being an order of magnitude larger than the unit of measurement (10 K° is twice as high as 5 K°).
 - e.g., *temperature in Kelvin, length, counts, monetary quantities*

Discrete vs. Continuous Attributes

- **Discrete Attribute**
 - Has only a finite or countably infinite set of values
 - E.g., zip codes, profession, or the set of words in a collection of documents
 - Sometimes, represented as integer variables
 - Note: Binary attributes are a special case of discrete attributes
- **Continuous Attribute**
 - Has real numbers as attribute values
 - E.g., 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

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):

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

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

- 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*):

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

Median interval →

age	frequency
1–5	200
6–15	450
16–20	300
21–50	1500
51–80	700
81–110	44

Measuring the Central Tendency

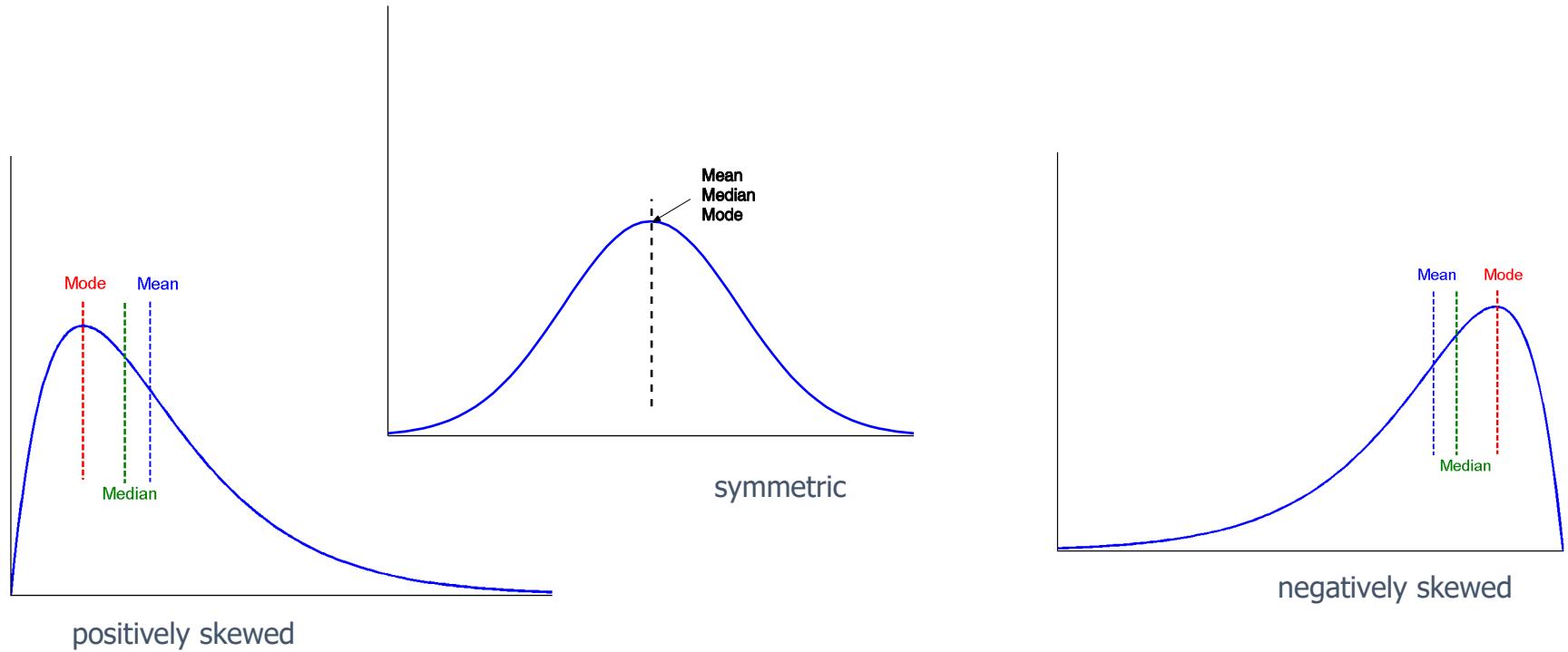
- Mode

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

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

Symmetric vs. Skewed Data

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



Measuring the Dispersion of Data

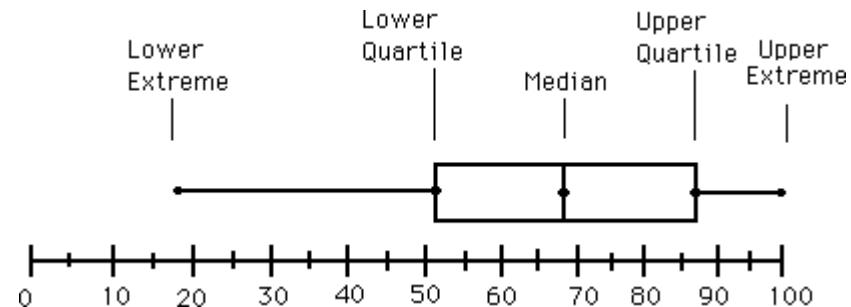
- Quartiles, outliers and boxplots
 - **Quartiles:** Q_1 (25th percentile), Q_3 (75th percentile)
 - **Inter-quartile range:** $IQR = Q_3 - Q_1$
 - **Five number summary:** min, Q_1 , median, Q_3 , max
 - **Boxplot:** ends of the box are the quartiles; median is marked; add whiskers, and plot outliers individually
 - **Outlier:** usually, a value higher/lower than $1.5 \times IQR$ (on both sides of box from Q_1 to Q_3)
- Variance and standard deviation (*sample: s, population: σ*)
 - **Variance:** (algebraic, scalable computation)

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

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^n (x_i - \mu)^2 = \frac{1}{N} \sum_{i=1}^n x_i^2 - \mu^2$$
 - **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

Example

Following is an ordered list of observations of a variable. Compute 5 point summary.

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

Solution:

Min: 13

Q1: 20

Median: 25

Q3: 35

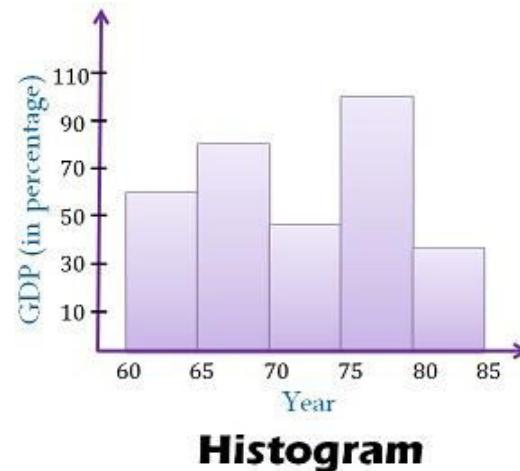
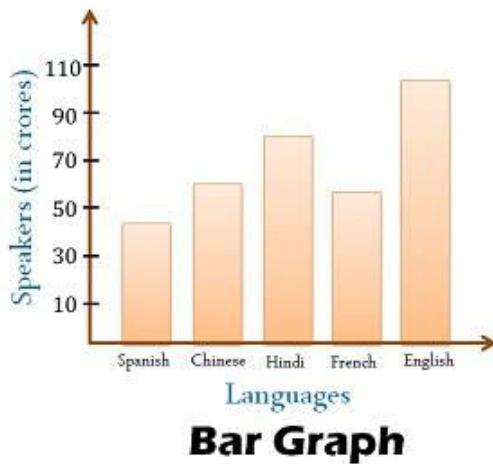
Max: 70

Any possible outliers here?

Graphic Displays of Basic Statistical Descriptions

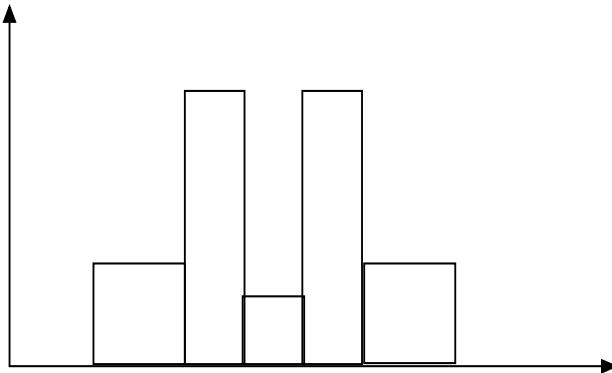
- **Boxplot:** graphic display of five-number summary
- **Histogram:** x-axis are values, y-axis repres. frequencies
- **Quantile plot:** each value x_i is paired with f_i indicating that approximately $100 f_i \%$ of data are $\leq x_i$
- **Quantile-quantile (q-q) plot:** graphs the quantiles of one 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

Histogram Analysis

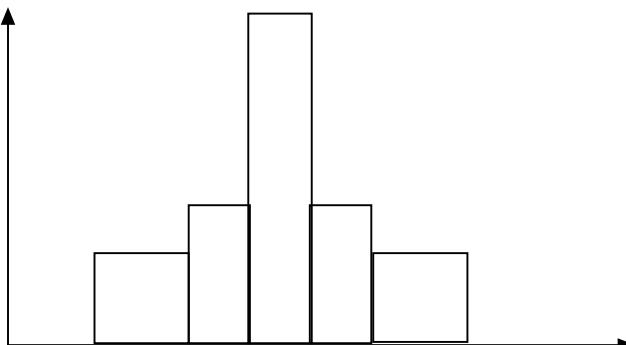


- Histogram: Graph display of tabulated frequencies, shown as bars
- It shows what proportion of cases fall into each of several categories
- Differs from a bar chart in that it is the *area* of the bar that denotes the value, not the height as in bar charts, a crucial distinction when the categories 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

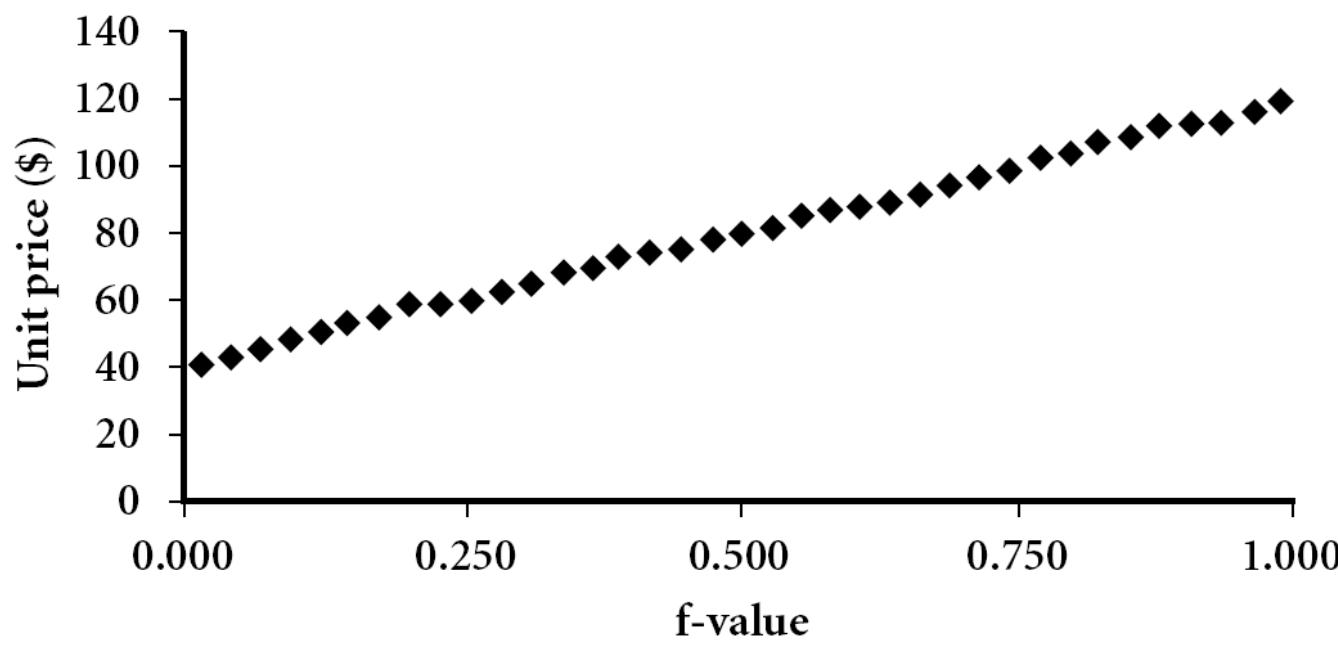


- The two histograms shown in the left may have the same boxplot representation
 - The same values for: min, Q1, median, Q3, max
- But they have rather different data distributions



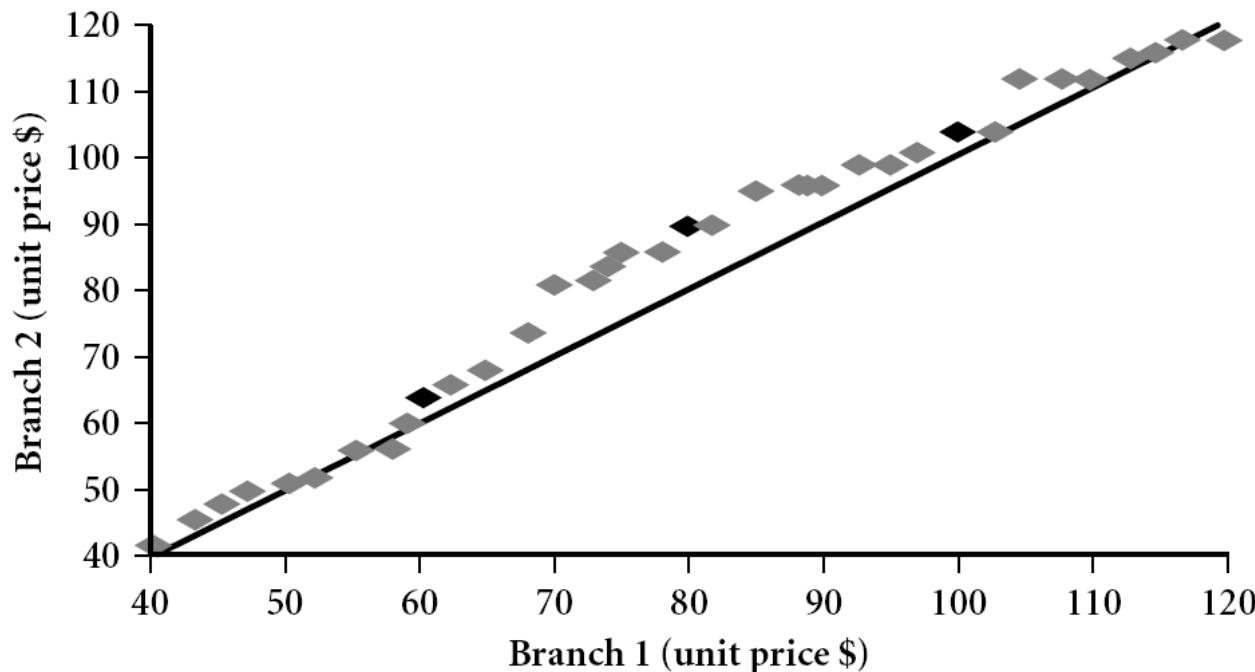
Quantile Plot

- 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 , data sorted in increasing order, f_i indicates that approximately $100 f_i\%$ of the data are below or equal to the value x_i



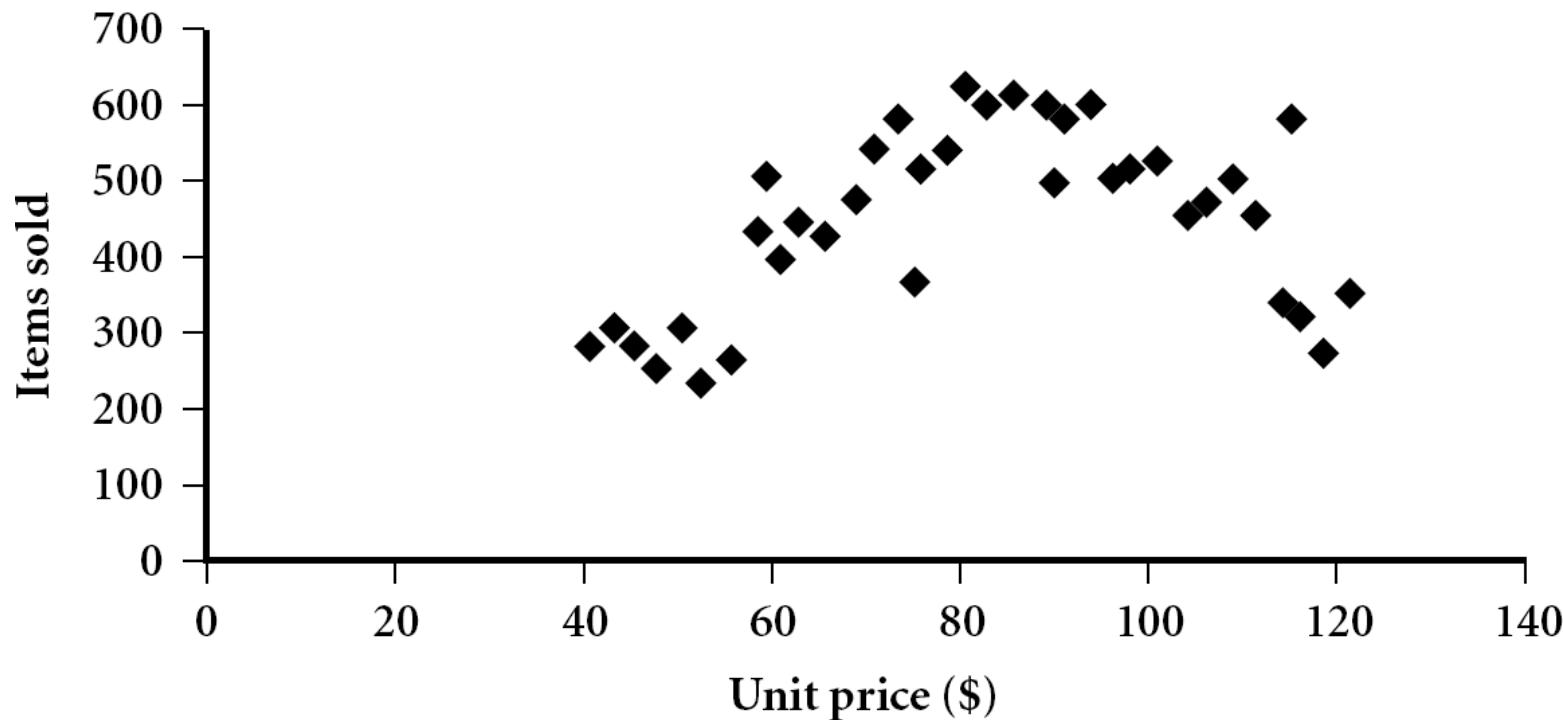
Quantile-Quantile (Q-Q) Plot

- Graphs the quantiles of one univariate distribution against the corresponding quantiles of another
- View: Is there 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.

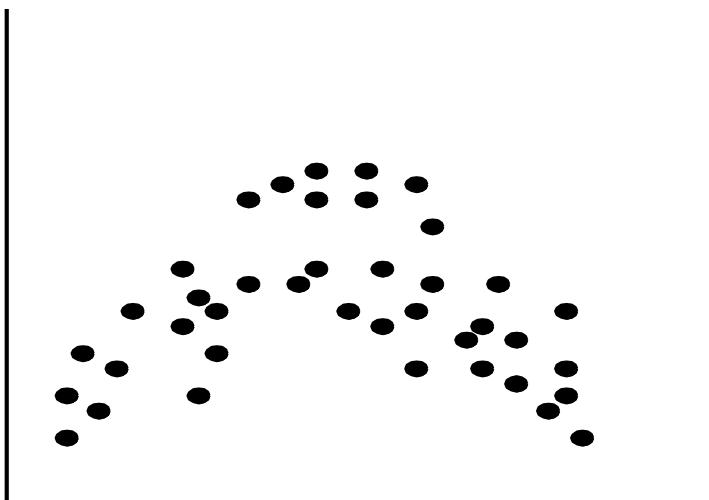
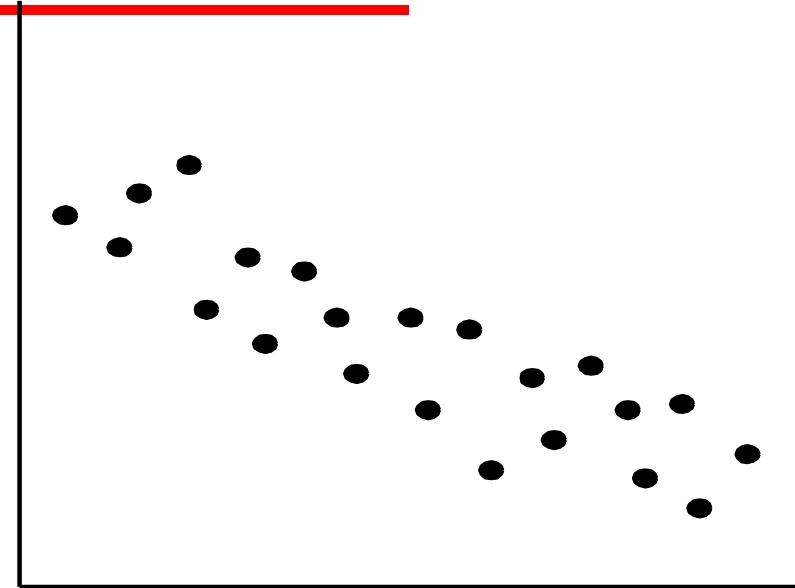
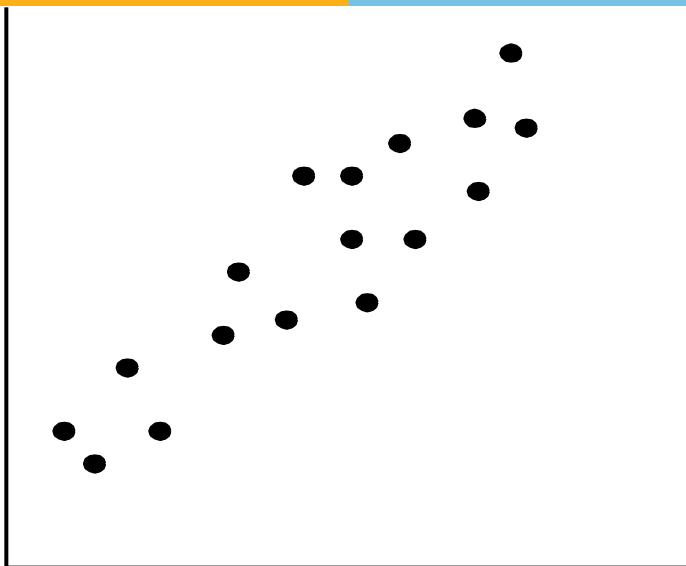


Scatter plot

- 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

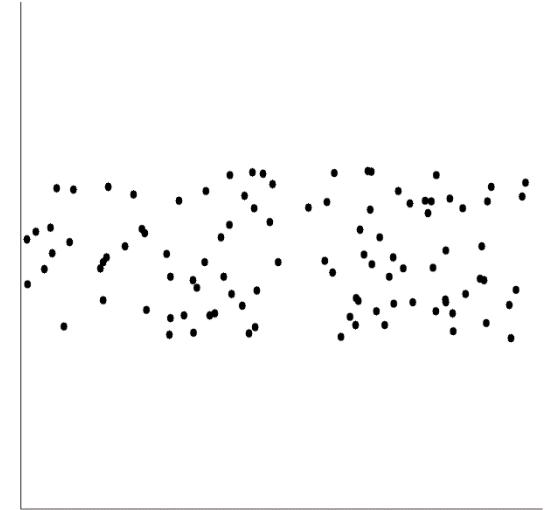
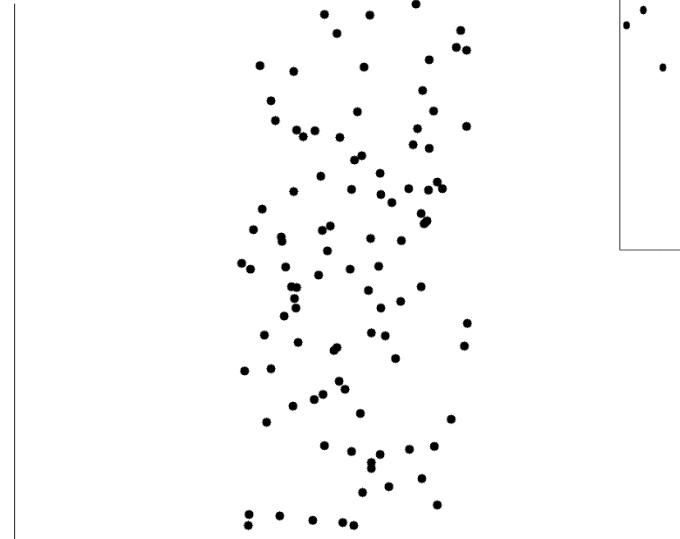


Positively and Negatively Correlated Data



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

Uncorrelated Data



Data Similarity/Dissimilarity

Similarity and Dissimilarity

- **Similarity**
 - Numerical measure of how alike two data objects are
 - Value is higher when objects are more alike
 - Often falls in the range [0,1]
- **Dissimilarity** (e.g., distance)
 - Numerical measure of how different two data objects are
 - Lower when objects are more alike
 - Minimum dissimilarity is often 0
 - Upper limit varies
- **Proximity** refers to a similarity or dissimilarity

Data Matrix and Dissimilarity Matrix

- Data matrix
 - n data points with p dimensions
 - Two modes

- Dissimilarity matrix
 - n data points, but registers only the distance
 - A triangular matrix
 - Single mode

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Proximity Measure for Nominal Attributes

- Can take 2 or more states, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Method 1: Simple matching
 - m : # of matches, p : total # of variables
- Method 2: Use a large number of binary attributes
 - creating a new binary attribute for each of the M nominal states

$$d(i, j) = \frac{P - m}{P}$$

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>

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

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

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

- Distance measure for symmetric binary variables:
- Distance measure for asymmetric binary variables:
- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):
 - 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 (to match contingency table of prev slide)
- Following are distances based on asymmetric binary variables:

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

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

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

Standardizing Numeric Data

- Z-score:

$$z = \frac{x - \mu}{\sigma}$$

- X: raw score to be standardized, μ : mean of the population, σ : standard deviation
- the distance between the raw score and the population mean in units of the standard deviation
- negative when the raw score is below the mean, “+” when above
- An alternative way: Calculate the mean absolute deviation

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|)$$

where

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \dots + x_{nf}) \quad z_{if} = \frac{x_{if} - m_f}{s_f}$$

- standardized measure (*z-score*):
- Using mean absolute deviation is more robust than using standard deviation

Distance on Numeric Data: Minkowski Distance

- *Minkowski distance*: A popular distance measure

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

where $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ are two p -dimensional data objects, and h is the order (the distance so defined is also called L- h norm)

- Properties
 - $d(i, j) > 0$ if $i \neq j$, and $d(i, i) = 0$ (Positive definiteness)
 - $d(i, j) = d(j, i)$ (Symmetry)
 - $d(i, j) \leq d(i, k) + d(k, j)$ (Triangle Inequality)
- A distance that satisfies these properties is a **metric**

Special Cases of Minkowski Distance

- $h = 1$: Manhattan (city block, L_1 norm) distance
 - E.g., the Hamming distance: the number of bits that are different between two binary vectors

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

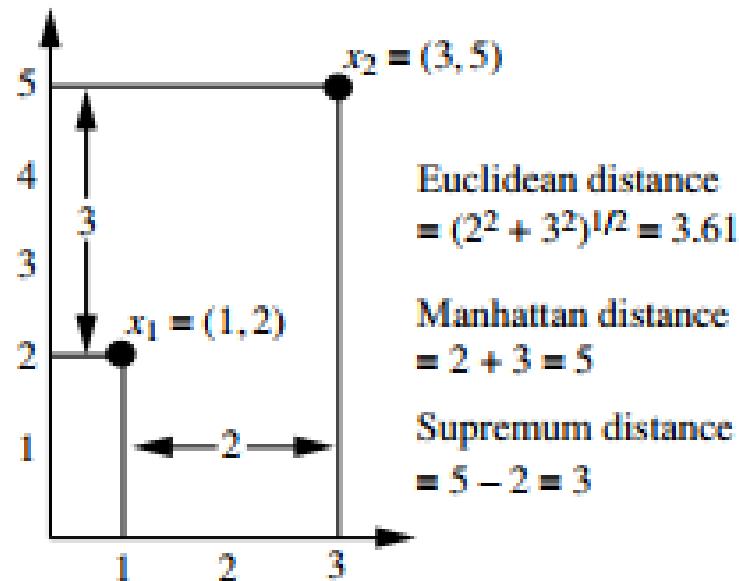
- $h = 2$: (L_2 norm) Euclidean distance

$$d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)}$$

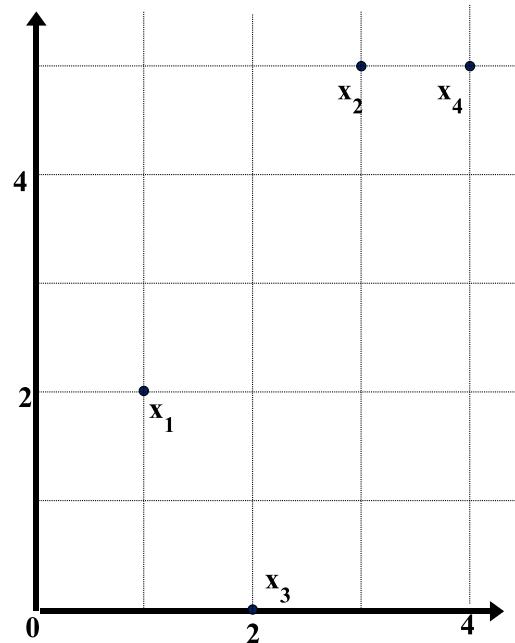
- $h \rightarrow \infty$. “supremum” (L_{\max} norm, L_{∞} norm) distance.
 - This is the maximum difference between any component (attribute) of the vectors

$$d(i, j) = \lim_{h \rightarrow \infty} \left(\sum_{f=1}^p |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_f^p |x_{if} - x_{jf}|$$

Euclidean, Manhattan, and supremum distances between two objects



Example: Data Matrix and Dissimilarity Matrix



Data Matrix

point	attribute1	attribute2
$x1$	1	2
$x2$	3	5
$x3$	2	0
$x4$	4	5

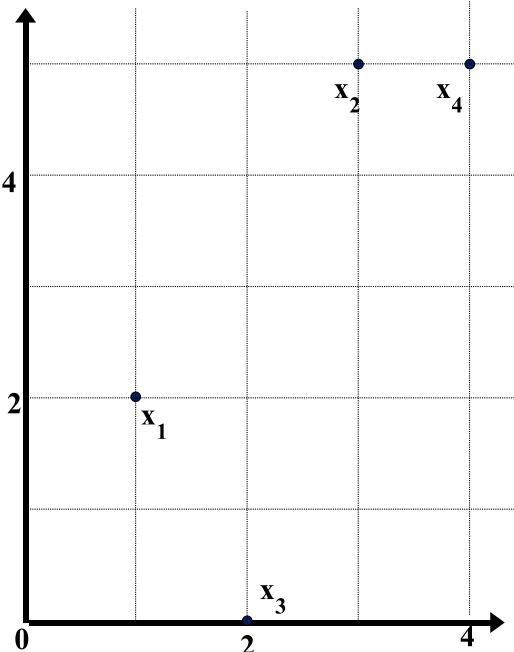
Dissimilarity Matrix

(with Euclidean Distance)

	$x1$	$x2$	$x3$	$x4$
$x1$	0			
$x2$	3.61	0		
$x3$	2.24	5.1	0	
$x4$	4.24	1	5.39	0

Example: Minkowski Distance

point	attribute 1	attribute 2
x1	1	2
x2	3	5
x3	2	0
x4	4	5



(Dissimilarity Matrices)					
Manhattan (L_1)					
L_1	x1	x2	x3	x4	
x1	0				
x2	5	0			
x3	3	6	0		
x4	6	1	7	0	
Euclidean (L_2)					
L_2	x1	x2	x3	x4	
x1	0				
x2	3.61	0			
x3	2.24	5.1	0		
x4	4.24	1	5.39	0	
Supremum					
L_∞	x1	x2	x3	x4	
x1	0				
x2	3	0			
x3	2	5	0		
x4	3	1	5	0	

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank
- Can be treated like interval-scaled
 - replace x_{if} by their rank $r_{if} \in \{1, \dots, M_f\}$
 - map the range of each variable onto [0, 1] by replacing i -th object in the f -th variable by
$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$
- compute the dissimilarity using methods for interval-scaled variables

Attributes of Mixed Type

- A database may contain all attribute types
 - Nominal, symmetric binary, asymmetric binary, numeric, ordinal
- One may use a weighted formula to combine their effects

$$d(i, j) = \frac{\sum_{f=1}^p \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^p \delta_{ij}^{(f)}}$$

- f is binary or nominal:
 $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise
- f is numeric: use the normalized distance

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max(x) - \min(x)}$$

- f is ordinal
 - Compute ranks r_{if} and
 - Treat r_{if} as interval-scaled

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

Object Identifier	test-1 (nominal)	test-2 (ordinal)	test-3 (numeric)
1	code A	excellent	45
2	code B	fair	22
3	code C	good	64
4	code A	excellent	28

$$\begin{bmatrix} 0 \\ 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.0 \\ 1.0 & 0.0 \\ 0.5 & 0.5 & 0.0 \\ 0.0 & 1.0 & 0.5 & 0.0 \end{bmatrix} \begin{bmatrix} 0.0 & 0.0 \\ 0.55 & 0.0 \\ 0.45 & 1.0 & 0.0 \\ 0.40 & 0.14 & 0.86 & 0.0 \end{bmatrix}$$

$$\begin{bmatrix} 0.0 \\ 0.85 & 0.0 \\ 0.65 & 0.83 & 0.0 \\ 0.13 & 0.71 & 0.79 & 0.0 \end{bmatrix}$$

Objects 1 and 2 are most dissimilar while 1 and 4 are most similar.

Example

Based on the information given in the table below, find most similar and most dissimilar persons among them. Apply min-max normalization on income to obtain [0,1] range. Consider profession and mother tongue as nominal. Consider native place as ordinal variable with ranking order of [Village, Small Town, Suburban, Metropolitan]. Give equal weight to each attribute.

Name	Income	Profession	Mother tongue	Native Place
Ram	70000	Doctor	Bengali	Village
Balram	50000	Data Scientist	Hindi	Small Town
Bharat	60000	Carpenter	Hindi	Suburban
Kishan	80000	Doctor	Bhojpuri	Metropolitan

Solution

After normalizing income and quantifying native place, we get

Name	Income	Profession	Mother tongue	Native Place
Ram	0.67	Doctor	Bengali	1
Balram	0	Data Scientist	Hindi	2
Bharat	0.33	Carpenter	Hindi	3
Kishan	1	Doctor	Bhojpuri	4

$$d(\text{Ram}, \text{Balram}) = 0.67+1+1+(2-1)/(4-1)=3 \quad d(\text{Ram}, \text{Bharat}) = 0.33+1+1+(3-1)/(4-1)=3$$

$$d(\text{Ram}, \text{Kishan}) = 0.33+0+1+(4-1)/(4-1) = 2.33 \quad d(\text{Balram}, \text{Bharat}) = 0.33+1+0+(3-2)/(4-1)=1.67$$

$$d(\text{Balram}, \text{Kishan}) = 1+1+1+(4-2)/(4-1) = 3.67 \quad d(\text{Bharat}, \text{Kishan}) = 0.67+1+1+(4-3)/(4-1) = 3$$

Most similar – Balram and Bharat; Most dissimilar – Balram and Kishan

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	<i>team</i>	<i>coach</i>	<i>hockey</i>	<i>baseball</i>	<i>soccer</i>	<i>penalty</i>	<i>score</i>	<i>win</i>	<i>loss</i>	<i>season</i>
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

- Other vector objects: gene features in micro-arrays, ...
- 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.

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

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

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

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

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

$$\cos(d_1, d_2) = 0.94$$

Thank You



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

S2-21_DSECLZC415

Classification and Prediction



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

Classification

Classification

- Classification involves dividing up objects so that each is assigned to one of a number of mutually exhaustive and exclusive categories known as *classes*
- Many practical decision-making tasks can be formulated as classification problems
 - customers who are likely to buy or not buy a particular product in a supermarket
 - people who are at high, medium or low risk of acquiring a certain illness
 - student projects worthy of a distinction, merit, pass or fail grade
 - objects on a radar display which correspond to vehicles, people, buildings or trees
 - people who closely resemble, slightly resemble or do not resemble someone seen committing a crime
 - houses that are likely to rise in value, fall in value or have an unchanged value in 12 months' time
 - people who are at high, medium or low risk of a car accident in the next 12 months
 - people who are likely to vote for each of a number of political parties (or none)
 - the likelihood of rain the next day for a weather forecast (very likely, likely, unlikely, very unlikely).

Classification vs. Prediction

- Classification
 - predicts categorical class labels (discrete or nominal)
 - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Prediction
 - models continuous-valued functions, i.e., predicts unknown or missing values

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
 - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
 - New data is classified based on the training set
- Unsupervised learning (clustering)
 - The class labels of training data is unknown
 - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

People also talk about more forms of machine learning

<https://www.gartner.com/smarterwithgartner/understand-3-key-types-of-machine-learning/>

https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQk6wDdKff6owv5L902n1AMs6UB_dn4Kiwc6w&usqp=CAU

Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
 - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
 - The set of tuples used for model construction is training set
 - The model is represented as classification rules, decision trees, or mathematical formulae
- Model usage: for classifying future or unknown objects
 - Estimate accuracy of the model
 - The known label of test sample is compared with the classified result from the model
 - Accuracy rate is the percentage of test set samples that are correctly classified by the model
 - Test set is independent of training set, otherwise over-fitting will occur
 - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

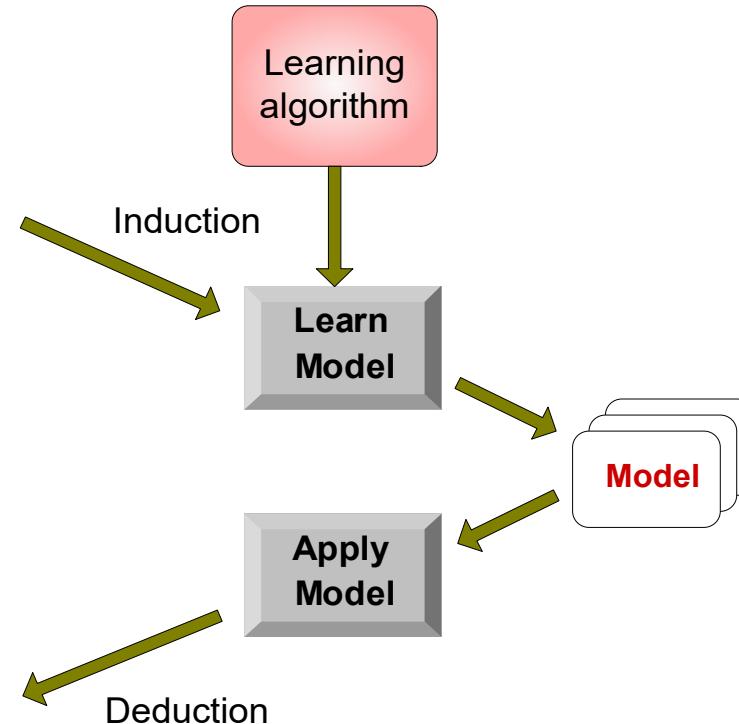
Illustrating Classification Task

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



Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Neural Networks
 - computational networks that simulate the decision process in neurons (networks of nerve cell)
- Naïve Bayes and Bayesian Belief Networks
 - uses the *probability theory* to find the most likely of the possible classifications
- Support Vector Machines
 - fits a boundary to a region of points that are all alike; uses the boundary to classify a new point

Lazy vs. Eager Learning

- Lazy vs. eager learning
 - Lazy learning (e.g., instance-based learning): Simply stores training data (or only minor processing) and waits until it is given a test tuple
 - Eager learning (the above discussed methods): Given a set of training set, constructs a classification model before receiving new (e.g., test) data to classify
- Lazy: less time in training but more time in predicting
- Accuracy
 - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
 - Eager: must commit to a single hypothesis that covers the entire instance space

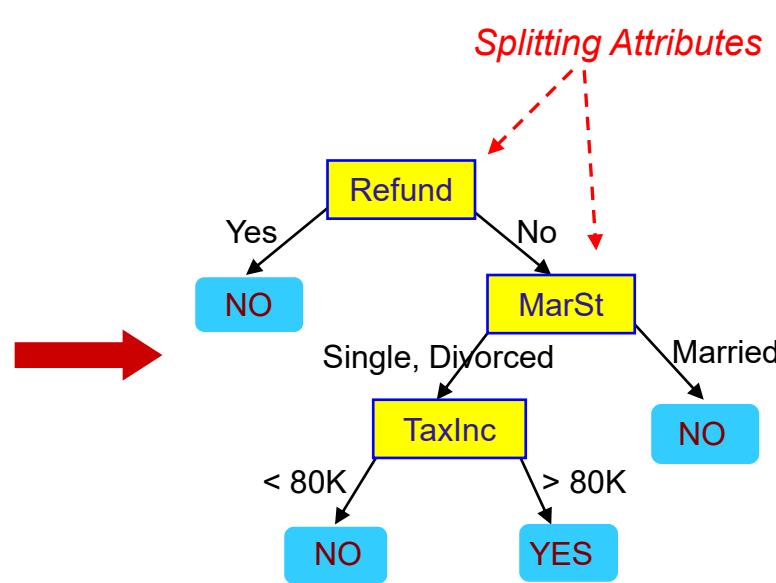
Lazy Learner: Instance-Based Methods

- Instance-based learning:
 - Store training examples and delay the processing (“lazy evaluation”) until a new instance must be classified
- Typical approaches
 - [k-nearest neighbor approach](#)
 - Instances represented as points in a Euclidean space.
 - [Locally weighted regression](#)
 - Constructs local approximation
 - [Case-based reasoning](#)
 - Uses symbolic representations and knowledge-based inference

Example of a Decision Tree

<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

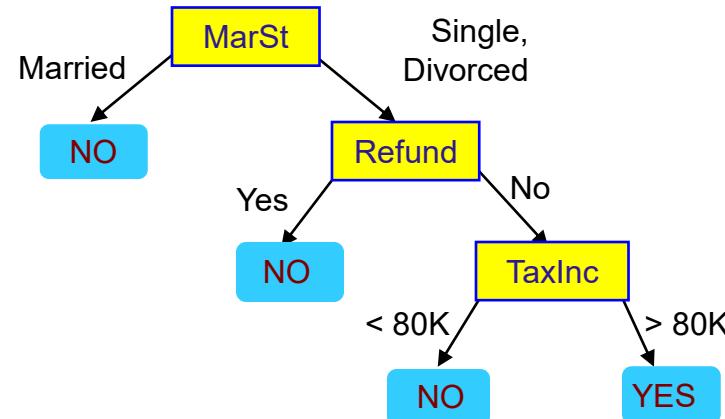
Training Data



Model: Decision Tree

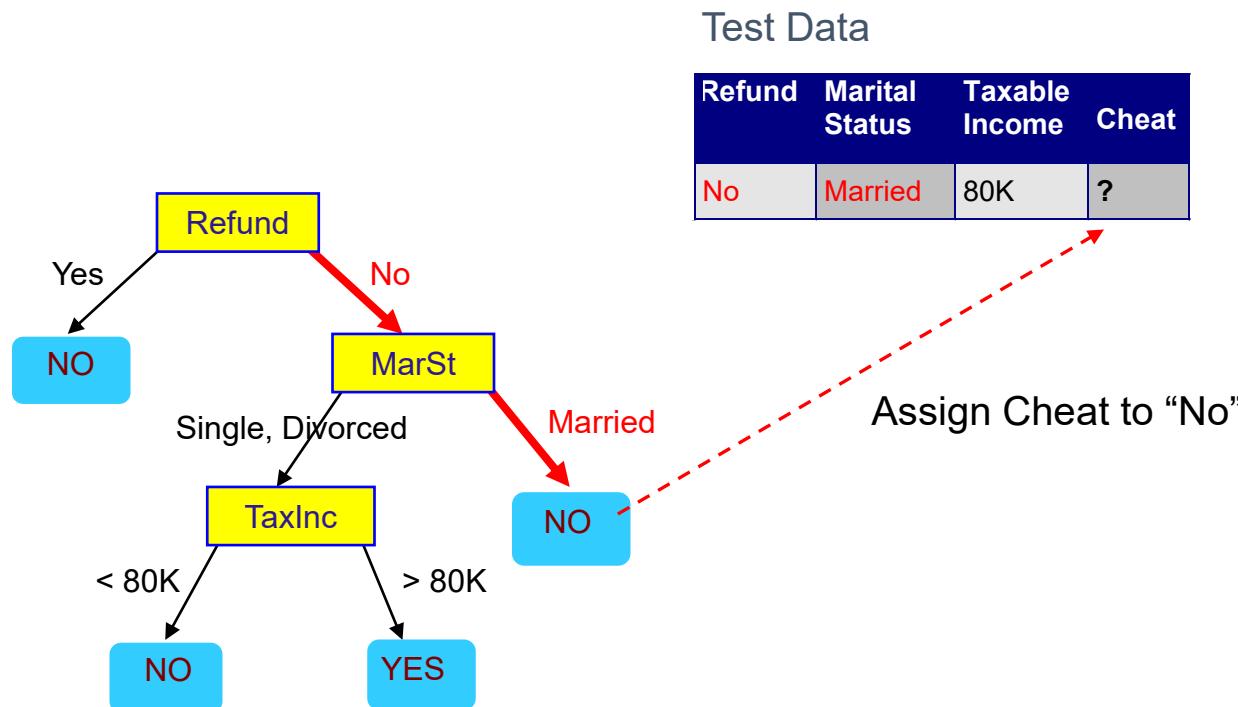
Another Example of Decision Tree

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!

Apply Model to Test Data



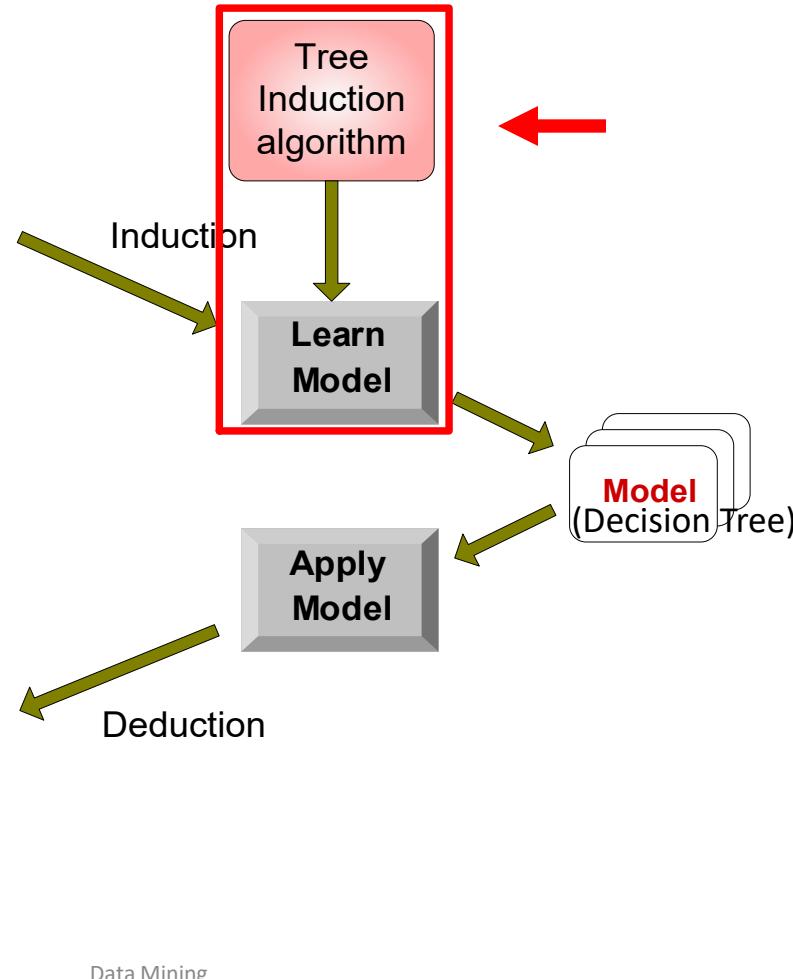
Decision Tree Classification Task

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	?

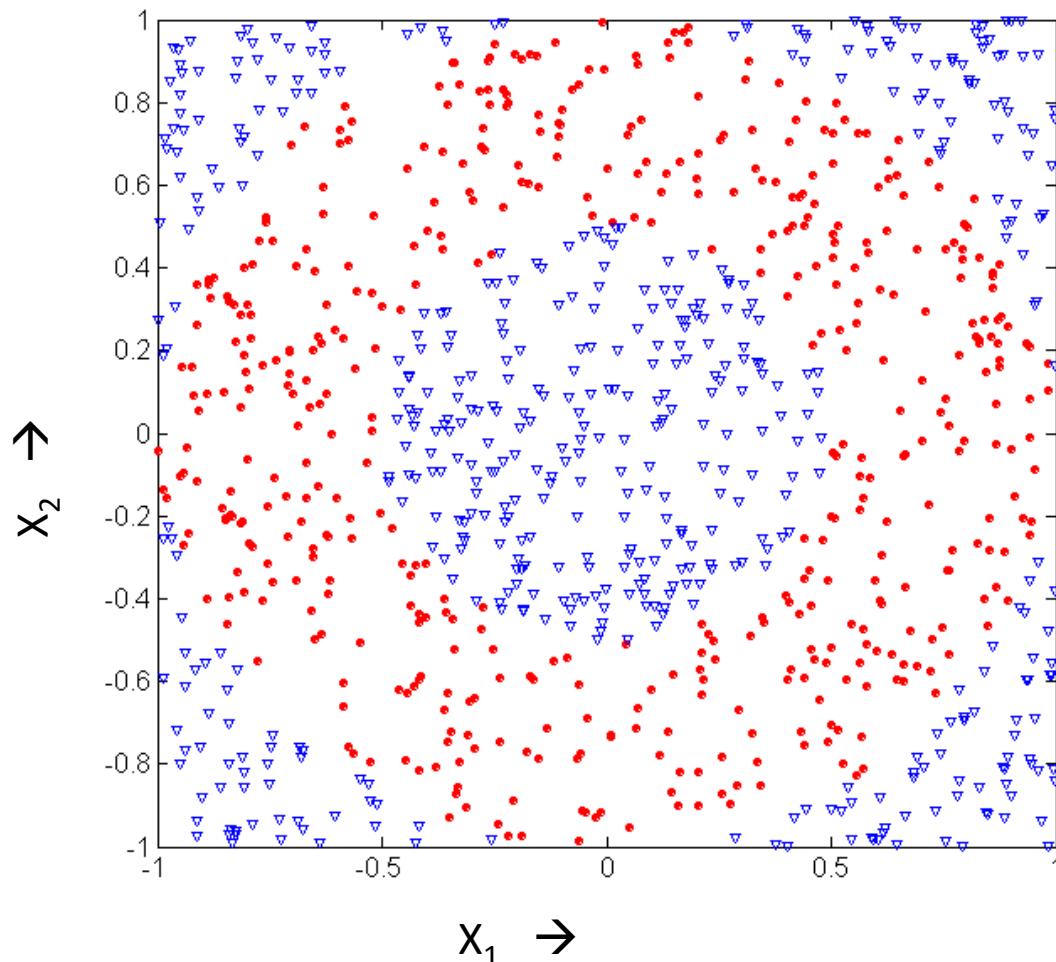
Test Set



Issues: Evaluating Classification Methods

- Accuracy
 - classifier accuracy: predicting class label
 - predictor accuracy: guessing value of predicted attributes
- Speed
 - time to construct the model (training time)
 - time to use the model (classification/prediction time)
- Robustness: handling noise and missing values
- Scalability: efficiency in disk-resident databases
- Interpretability
 - understanding and insight provided by the model
- Other measures, e.g., goodness of rules, such as decision tree size or compactness of classification rules

Underfitting and Overfitting (Example)



500 circular and 500 triangular data points.

Circular points:

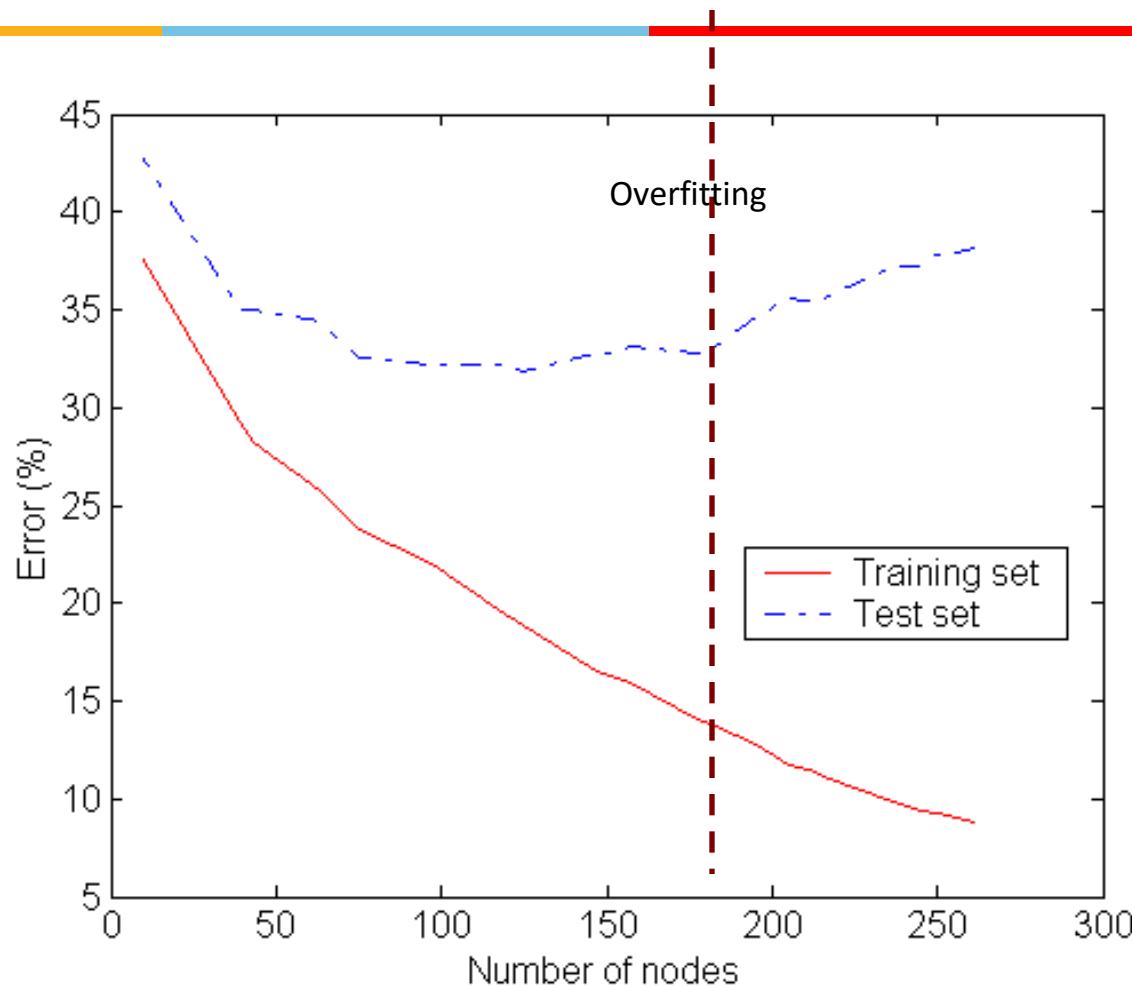
$$0.5 \leq \sqrt{x_1^2 + x_2^2} \leq 1$$

Triangular points:

$$\sqrt{x_1^2 + x_2^2} > 0.5 \text{ or}$$

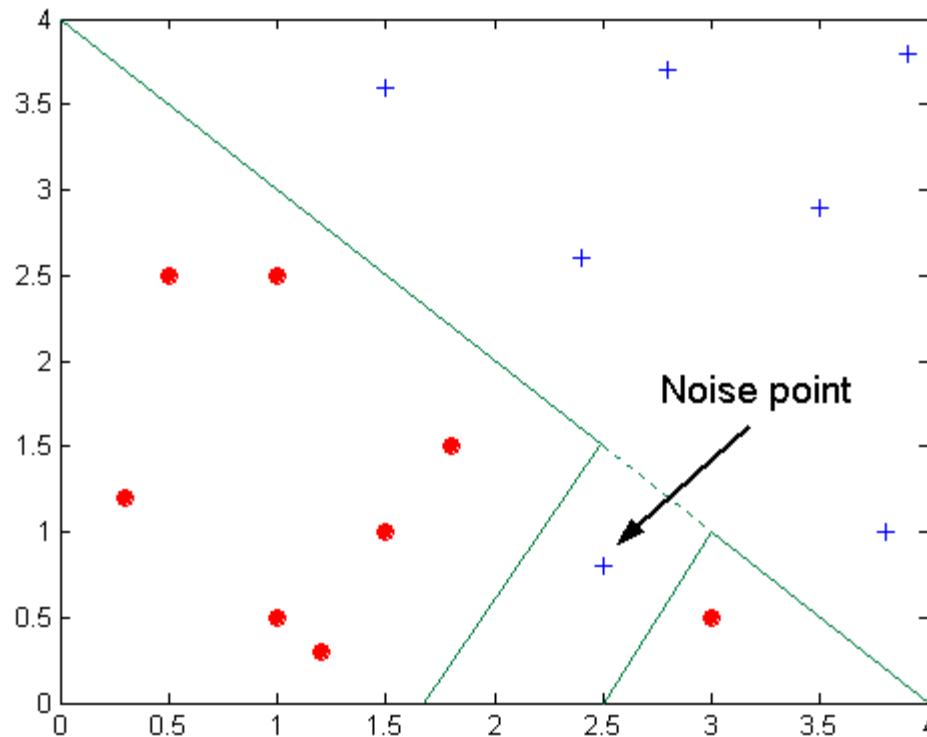
$$\sqrt{x_1^2 + x_2^2} < 1$$

Underfitting and Overfitting



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

Overfitting due to Noise



Decision boundary is distorted by noise point

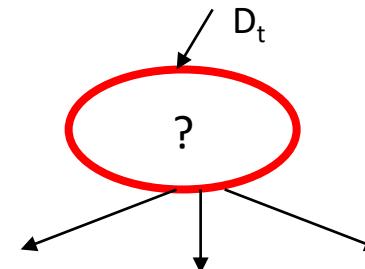
Decision Tree Based Classification

- Decision trees are intuitive and frequently used data mining technique for Classification
- For an analyst, they are easy to set up and for a business user they are easy to interpret.
- A decision tree model is a decision flowchart where an attribute is tested in each node and ends in a leaf node where a prediction is made.
- There are many algorithms for decision tree induction such as Hunt's Algorithm, CART, ID3, C4.5, SLIQ, SPRINT

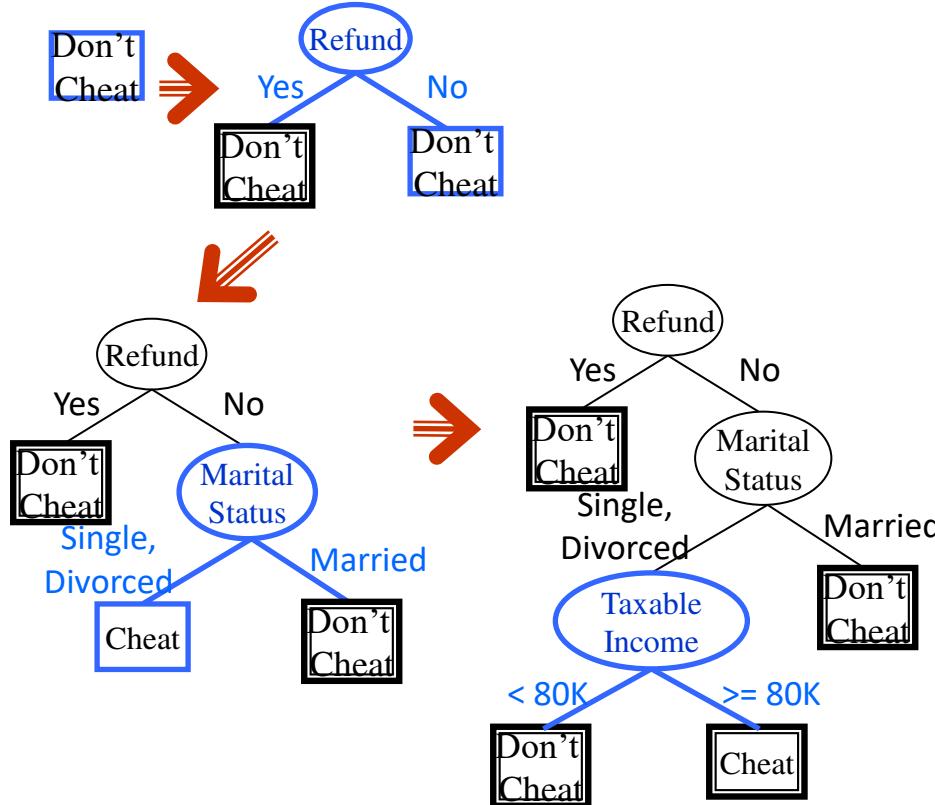
Hunt's Algorithm - Structure

- Hunt's algorithm is among the earliest. More complex algorithms were built upon it.
- It grows a decision tree in a recursive fashion by partitioning the training records into successively purer subsets
- Let D_t be the set of training records that reach a node t
- General Procedure:
 - If D_t contains records that belong to the same class y_t , then t is a leaf node labeled as y_t
 - If D_t is an empty set, then t is a leaf node labeled by the default class, y_d
 - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

<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



Hunt's Algorithm - 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

Tree Induction

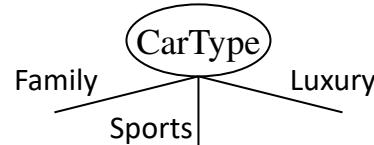
- Greedy strategy.
 - Split the records based on an attribute test that optimizes certain criterion.
- Issues
 - Determine how to split the records
 - How to specify the attribute test condition?
 - How to determine the best split?
 - Determine when to stop splitting

How to Specify Test Condition?

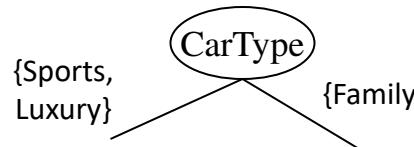
- Depends on attribute types
 - Nominal
 - Ordinal
 - Continuous
- Depends on number of ways to split
 - 2-way split
 - Multi-way split

Splitting Based on Nominal Attributes

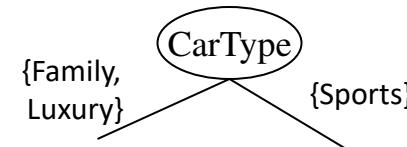
- **Multi-way split:** Use as many partitions as distinct values.



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

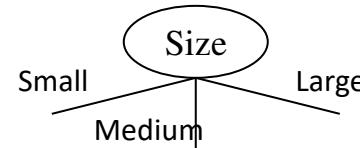


OR

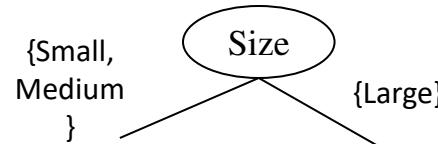


Splitting Based on Ordinal Attributes

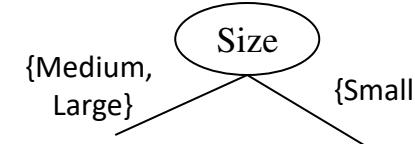
- **Multi-way split:** Use as many partitions as distinct values.



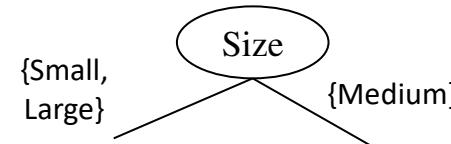
- **Binary split:** Divides values into two subsets.
Need to find optimal partitioning.



OR



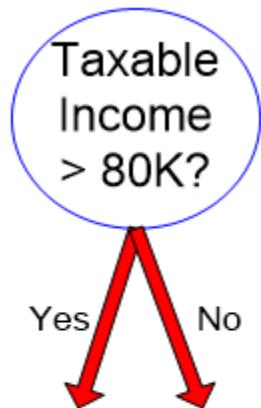
- What about this split?



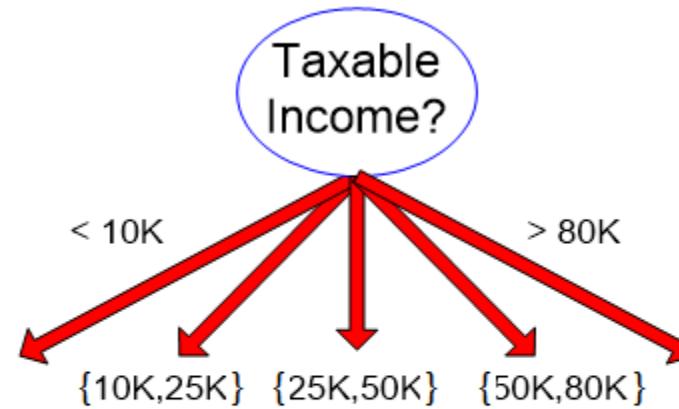
Splitting Based on Continuous Attributes

- Different ways of handling
 - **Discretization** to form an ordinal categorical attribute
 - Static – discretize once at the beginning
 - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
 - **Binary Decision:** $(A < v)$ or $(A \geq v)$
 - consider all possible splits and finds the best cut
 - can be more compute intensive

Splitting Based on Continuous Attributes



(i) Binary split

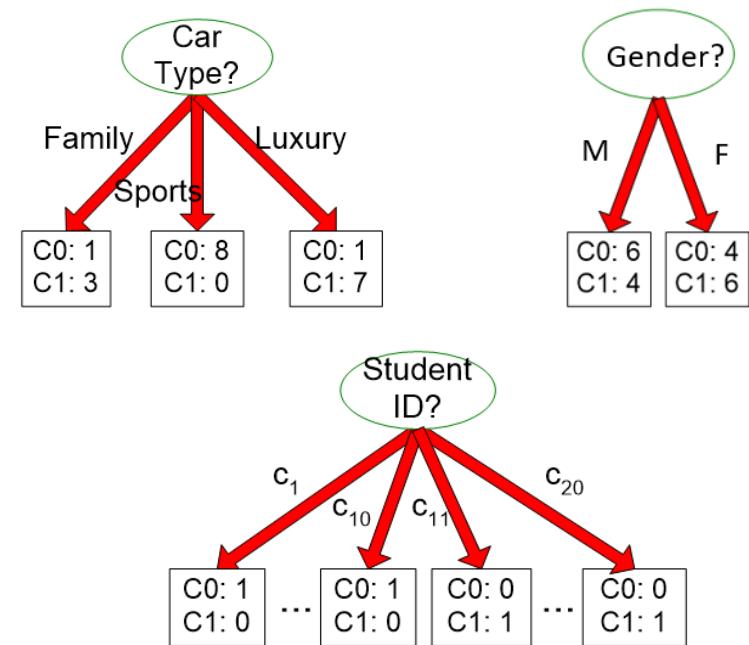


(ii) Multi-way split

How to determine the Best Split

Customer Id	Gender	Car Type	Shirt Size	Class
1	M	Family	Small	C0
2	M	Sports	Medium	C0
3	M	Sports	Medium	C0
4	M	Sports	Large	C0
5	M	Sports	Extra Large	C0
6	M	Sports	Extra Large	C0
7	F	Sports	Small	C0
8	F	Sports	Small	C0
9	F	Sports	Medium	C0
10	F	Luxury	Large	C0
11	M	Family	Large	C1
12	M	Family	Extra Large	C1
13	M	Family	Medium	C1
14	M	Luxury	Extra Large	C1
15	F	Luxury	Small	C1
16	F	Luxury	Small	C1
17	F	Luxury	Medium	C1
18	F	Luxury	Medium	C1
19	F	Luxury	Medium	C1
20	F	Luxury	Large	C1

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



Which test condition is the best?

How to determine the Best Split

- Greedy approach:
 - 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

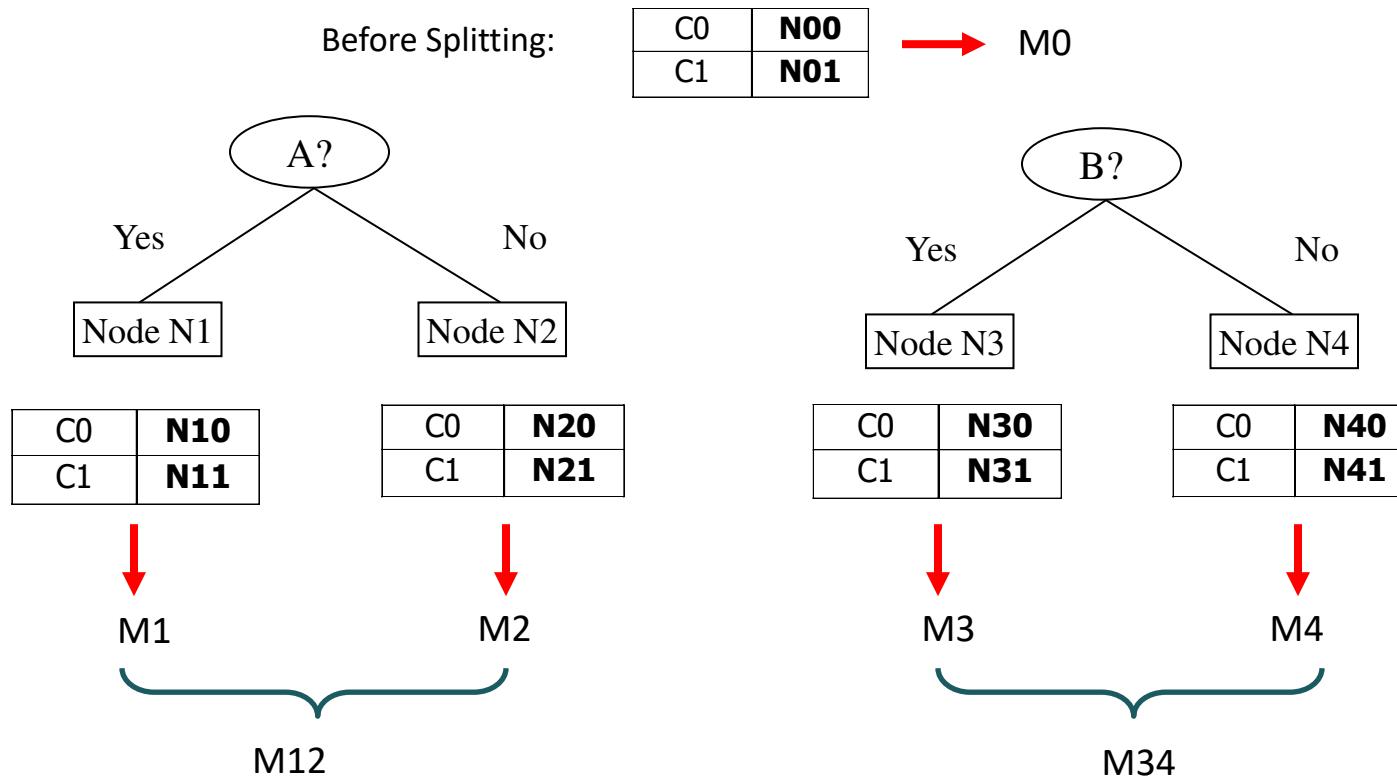
Finding the Best Split

1. Compute impurity measure (M_0) before splitting
2. Compute impurity measure (M) after splitting
 - Compute impurity measure of each child node
 - M is the weighted impurity of children
3. Choose the attribute test condition that produces the highest gain

$$\text{Gain} = M_0 - M$$

or equivalently, lowest impurity measure after splitting (M)

How to Find the Best Split



$$\text{Gain} = M0 - M12 \text{ vs } M0 - M34$$

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

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

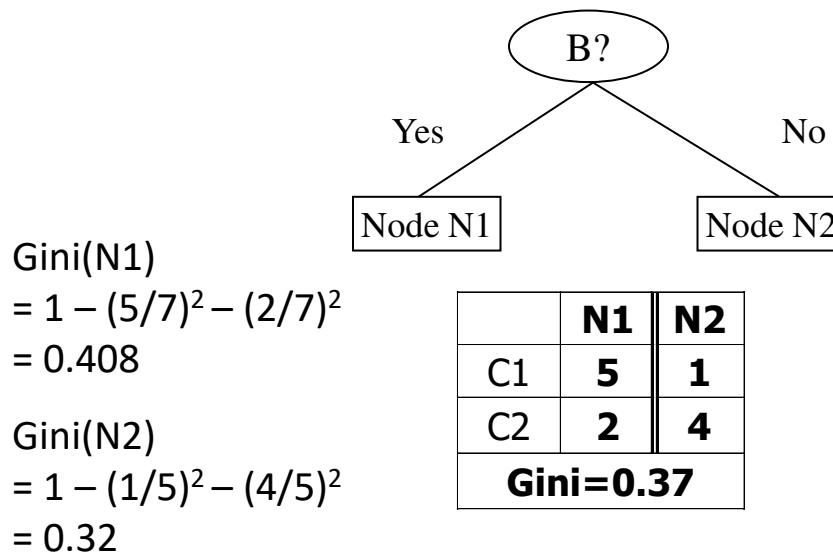
- Used in CART, SLIQ, SPRINT.
- When a node p is split into k partitions (children), the quality of split is computed as,

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

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

Binary Attributes: Computing GINI Index

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



	Parent
C1	6
C2	6
Gini = 0.500	

Gini(Children)

$$= 7/12 * 0.408 +$$

$$5/12 * 0.32$$

$$= 0.37$$

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)

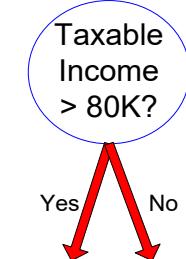
	CarType	
	{Sports, Luxury}	{Family}
C1	3	1
C2	2	4
Gini	0.400	

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

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...

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

→ → → → → → → → → → → → →

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No	No
Taxable Income											
	60	70	75	85	90	95	100	120	125	172	220
	55	65	72	80	87	92	97	110	122	172	230
	<=	>	<=	>	<=	>	<=	>	<=	>	<=
Yes	0	3	0	3	0	3	1	2	2	1	3
No	0	7	1	6	2	5	3	4	3	4	3
Gini	0.420	0.400	0.375	0.343	0.417	0.400	0.300	0.343	0.375	0.400	0.420

→ → → → → → → → → → → →

Sorted Values
Split Positions

Gini Index Example 1

The Gini index to compute the impurity of D:

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459.$$

Consider each attribute and all possible split. Ex. Let's consider attribute **income**. Find Gini index of the split into subset {low, medium} and {high}

$$\begin{aligned} Gini_{income \in \{low,medium\}}(D) &= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2) \\ &= \frac{10}{14} \left(1 - \left(\frac{6}{10}\right)^2 - \left(\frac{4}{10}\right)^2\right) + \frac{4}{14} \left(1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right)^2\right) \\ &= 0.450 \\ &= Gini_{income \in \{high\}}(D). \end{aligned}$$

Similarly, the Gini index values for splits on the remaining subsets are **0.458** (for the subsets {low, high} and {medium}) and **0.450** (for the subsets {medium, high} and {low}).

Therefore, the best binary split for attribute **income** is on {low, medium} (or {high}) because it minimizes the Gini index.

Evaluating **age**, we obtain {youth, senior} (or {middle aged}) as the best split for age with a Gini index of **0.357**. The attributes **student** and **credit rating** are both **binary**, with Gini index values of **0.367** and **0.429**, respectively.

The attribute **age** and splitting subset {youth, senior} therefore give the minimum Gini index overall, with a reduction in impurity of **0.459 - 0.357 = 0.102**.

<i>RID</i>	<i>age</i>	<i>income</i>	<i>student</i>	<i>credit_rating</i>	<i>Class: buys_computer</i>
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Alternative Splitting Criteria

- 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

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

$$\text{Entropy} = -0 * \log_2 0 - 1 * \log_2 1 = -0 - 0 = 0$$

C1	1
C2	5

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

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

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 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 ($1 - 1/n_c$) 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

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

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

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

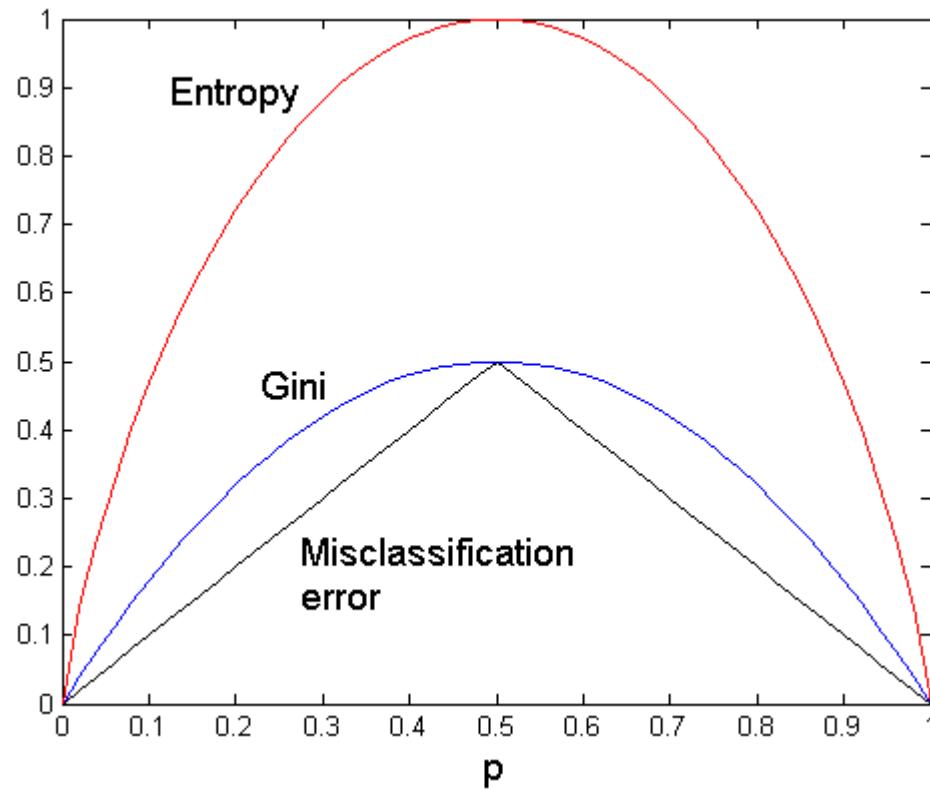
C1	2
C2	4

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

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

Comparison among Splitting Criteria

For a 2-class problem:



Gain Ratio

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary tuple in D belongs to class $C_{i,D}$, estimated by $|C_{i,D}|/|D|$
- **Expected information** (entropy) needed to classify a tuple in D:

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i)$$

- **Information** needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j)$$

- **Information gained** by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Information Gain: Example

RID	age	income	student	credit_rating	Class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j)$$

$$Info(D) = -\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) = 0.940 \text{ bits.}$$

$$\begin{aligned} Info_{age}(D) &= \frac{5}{14} \times \left(-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) \\ &\quad + \frac{4}{14} \times \left(-\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} \right) \\ &\quad + \frac{5}{14} \times \left(-\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right) \\ &= 0.694 \text{ bits.} \end{aligned}$$

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits.}$$

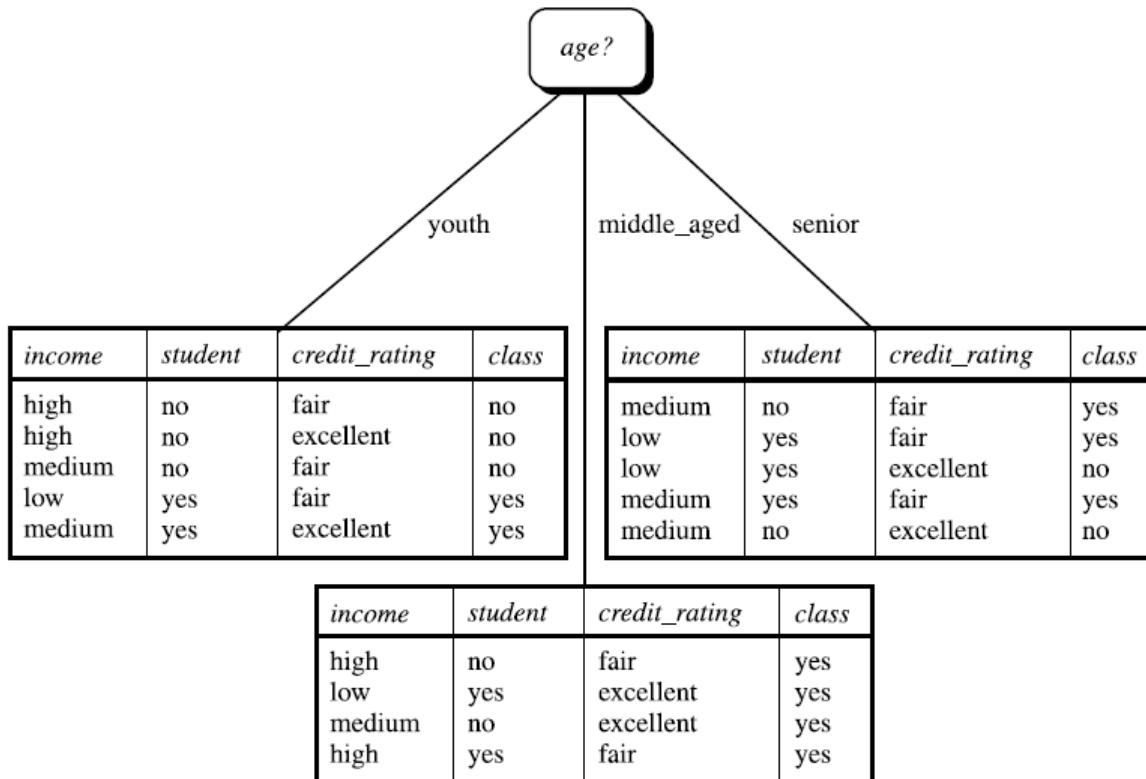
$$Gain(income) = 0.029 \text{ bits.}$$

$$Gain(student) = 0.151 \text{ bits}$$

$$Gain(credit_rating) = 0.048 \text{ bits.}$$

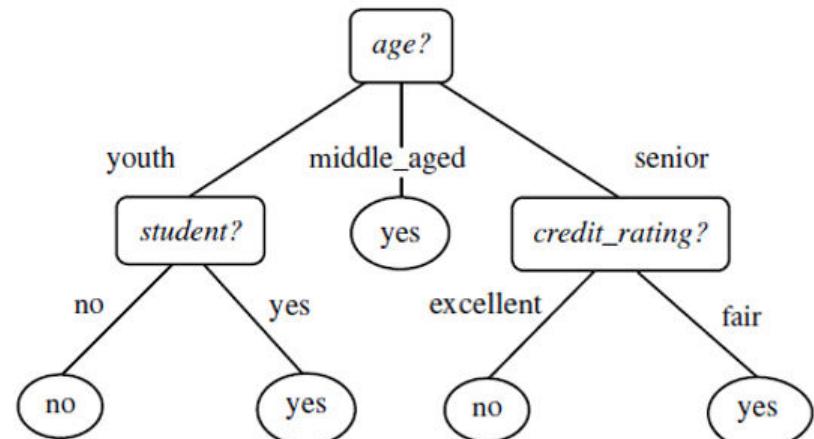
The attribute with the highest information gain is chosen as the splitting attribute at node.

Decision Tree: Step 1



Final Decision Tree

<i>RID</i>	<i>age</i>	<i>income</i>	<i>student</i>	<i>credit_rating</i>	<i>Class: buys_computer</i>
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no



Refining Decision Tree Model

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

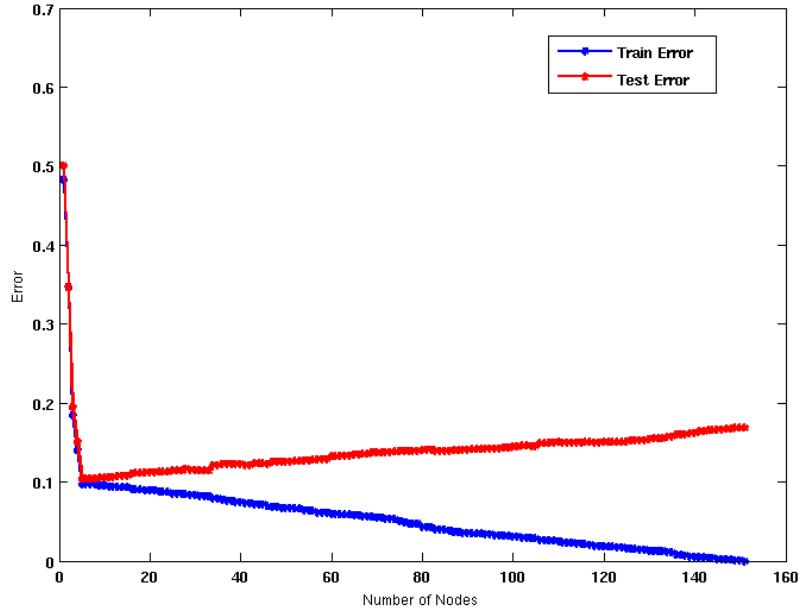
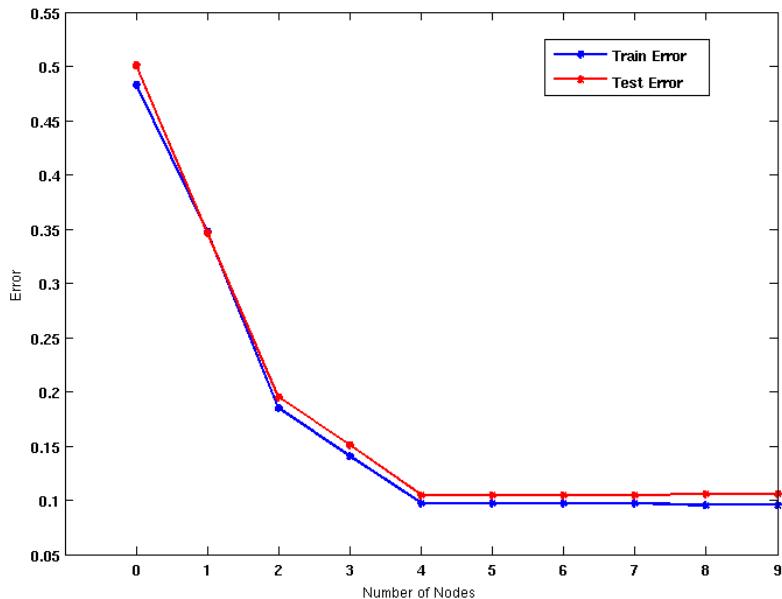
Practical Issues of Classification

- Underfitting and Overfitting
- Missing Values
- Costs of Classification

Underfitting vs. Overfitting

- Underfitting results in decision trees that are too simple to solve the problem. They may offer superior interpretability.
- Overfitting results in decision trees that are more complex than necessary
 - Training error no longer provides a good estimate of how well the tree will perform on previously unseen records
 - Need new ways for estimating errors

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

How to Address Overfitting

- Pre-Pruning (Early Stopping Rule)
 - Stop the algorithm before it becomes a fully-grown tree
 - General stopping conditions for a node:
 - Stop if all instances belong to the same class
 - Stop if all the attribute values are the same
 - More restrictive conditions (for pre-pruning) :
 - Stop if number of instances is less than some user-specified threshold
 - 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
 - Grow decision tree to its entirety
 - Trim the nodes of the decision tree in a bottom-up fashion
 - If generalization error(i.e. expected error of the model on previously unseen records) improves after trimming, replace sub-tree by a leaf node.
 - Class label of leaf node is determined from majority class of instances in the sub-tree

Decision Tree Based Classification

- Advantages:
 - Inexpensive to construct
 - Extremely fast at classifying unknown records
 - Easy to interpret for small-sized trees
 - Accuracy is comparable to other classification techniques for many simple data sets

Prescribed Text Books

Author(s), Title, Edition, Publishing House	
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
R2	Principles of Data Mining, Second Edition by Max Bramer Springer © 2013
R1	Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner by Vijay Kotu and Bala Deshpande Morgan Kaufmann Publishers

Thank You



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

S2-21_DSECLZC415

Classification and Prediction



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

Rule-based Classification

Rule-Based Classifier

- Classify records by using a collection of “if...then...” rules
- Rule: (Condition) → y where
 - Condition is a conjunctions of attributes
 - y is the class label
 - LHS: rule antecedent or condition
 - RHS: rule consequent
 - Examples of classification rules:
 - (Blood Type=Warm) ∧ (Lay Eggs=Yes) → Birds
 - (Taxable Income < 50K) ∧ (Refund=Yes) → Evade=No

Rule-based Classifier (Example)

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
human	warm	yes	no	no	mammals
python	cold	no	no	no	reptiles
salmon	cold	no	no	yes	fishes
whale	warm	yes	no	yes	mammals
frog	cold	no	no	sometimes	amphibians
komodo	cold	no	no	no	reptiles
bat	warm	yes	yes	no	mammals
pigeon	warm	no	yes	no	birds
cat	warm	yes	no	no	mammals
leopard shark	cold	yes	no	yes	fishes
turtle	cold	no	no	sometimes	reptiles
penguin	warm	no	no	sometimes	birds
porcupine	warm	yes	no	no	mammals
eel	cold	no	no	yes	fishes
salamander	cold	no	no	sometimes	amphibians
gila monster	cold	no	no	no	reptiles
platypus	warm	no	no	no	mammals
owl	warm	no	yes	no	birds
dolphin	warm	yes	no	yes	mammals
eagle	warm	no	yes	no	birds

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians

Application of Rule-Based Classifier

- A rule r **covers** an instance x if the attributes of the instance satisfy the condition of the rule

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
hawk	warm	no	yes	no	?
grizzly bear	warm	yes	no	no	?

The rule R1 covers a hawk => Bird

The rule R3 covers the grizzly bear => Mammal

Rule Coverage and Accuracy

- Coverage of a rule:
 - Fraction of records that satisfy the antecedent of a rule
- Accuracy of a rule:
 - Fraction of records that satisfy both the antecedent and consequent of a rule

(Status=Single) → No

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	No	Single	90K	Yes

Coverage = 40%, Accuracy = 50%

How does Rule-based Classifier Work?

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians

Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
lemur	warm	yes	no	no	?
turtle	cold	no	no	sometimes	?
dogfish shark	cold	yes	no	yes	?

A lemur triggers rule R3, so it is classified as a mammal

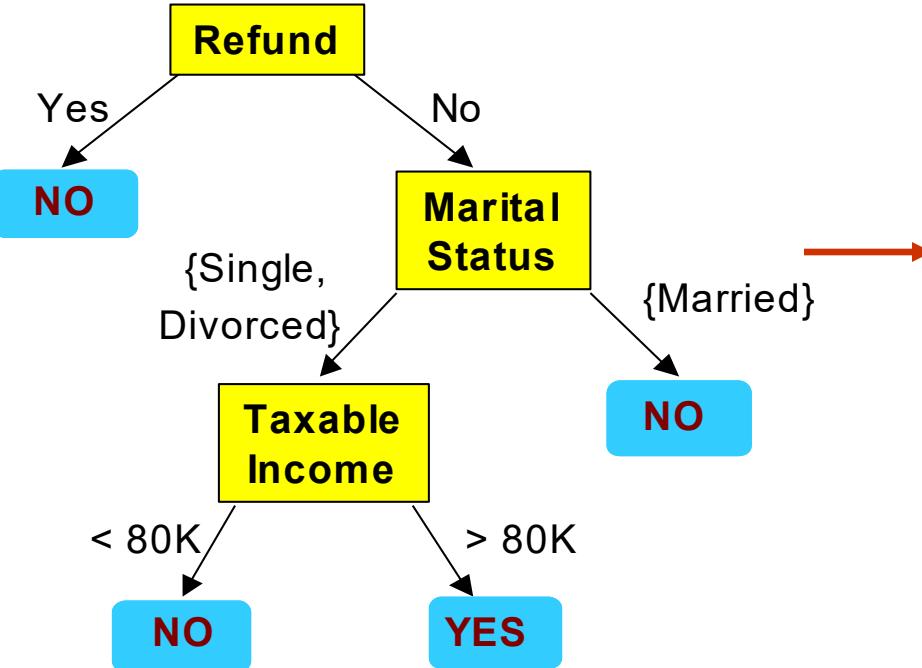
A turtle triggers both R4 and R5

A dogfish shark triggers none of the rules

Characteristics of Rule-Based Classifier

- Mutually exclusive rules
 - Classifier contains mutually exclusive rules if the rules are independent of each other
 - Every record is covered by at most one rule
- Exhaustive rules
 - Classifier has exhaustive coverage if it accounts for every possible combination of attribute values
 - Each record is covered by at least one rule

From Decision Trees To Rules



Classification Rules

$(\text{Refund}=\text{Yes}) \implies \text{No}$

$(\text{Refund}=\text{No}, \text{Marital Status}=\{\text{Single}, \text{Divorced}\}, \text{Taxable Income} < 80\text{K}) \implies \text{No}$

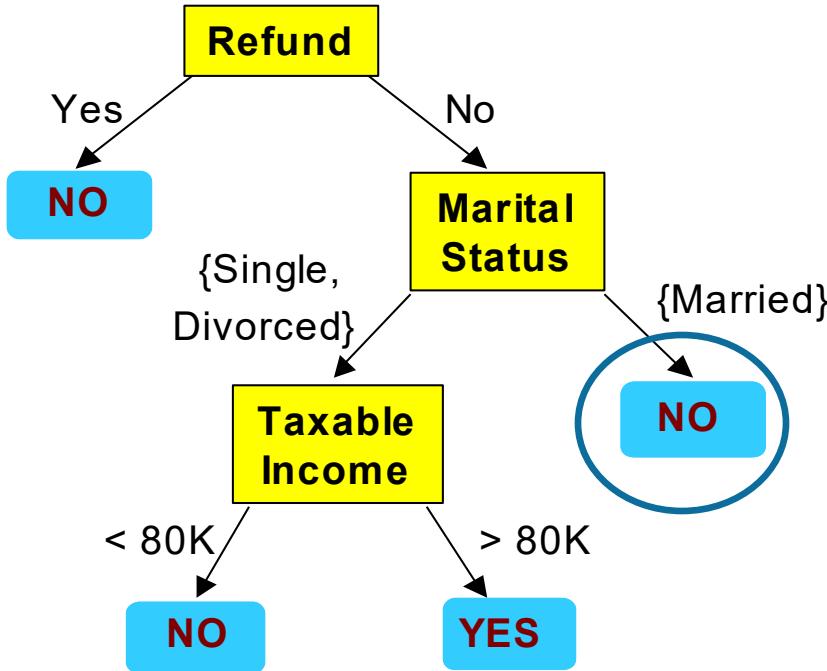
$(\text{Refund}=\text{No}, \text{Marital Status}=\{\text{Single}, \text{Divorced}\}, \text{Taxable Income} > 80\text{K}) \implies \text{Yes}$

$(\text{Refund}=\text{No}, \text{Marital Status}=\{\text{Married}\}) \implies \text{No}$

Rules are mutually exclusive and exhaustive

Rule set contains as much information as the tree

Rules Can Be Simplified



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

Initial Rule: $(\text{Refund}=\text{No}) \wedge (\text{Status}=\text{Married}) \rightarrow \text{No}$

Simplified Rule: $(\text{Status}=\text{Married}) \rightarrow \text{No}$

Further Characterizing Rules

- Rules are not mutually exclusive
 - A record may trigger more than one rule
 - Solution?
 - Ordered rule set
 - Unordered rule set – use voting schemes
- Rules are not exhaustive
 - A record may not trigger any rules
 - Solution?
 - Use a default class

Ordered Rule Set

- Rules are rank ordered according to their priority
 - An ordered rule set is known as a decision list
- When a test record is presented to the classifier
 - It is assigned to the class label of the highest ranked rule it has triggered
 - If none of the rules fired, it is assigned to the default class

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds

R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes

R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals

R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles

R5: (Live in Water = sometimes) \rightarrow Amphibians



Name	Blood Type	Give Birth	Can Fly	Live in Water	Class
turtle	cold	no	no	sometimes	?

Rule Ordering Schemes

- Rule-based ordering
 - Individual rules are ranked based on their quality
- Class-based ordering
 - Rules that belong to the same class appear together

Rule-based Ordering

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced},
Taxable Income<80K) ==> No

(Refund=No, Marital Status={Single,Divorced},
Taxable Income>80K) ==> Yes

(Refund=No, Marital Status={Married}) ==> No

Class-based Ordering

(Refund=Yes) ==> No

(Refund=No, Marital Status={Single,Divorced},
Taxable Income<80K) ==> No

(Refund=No, Marital Status={Married}) ==> No

(Refund=No, Marital Status={Single,Divorced},
Taxable Income>80K) ==> Yes

How to Evaluate Learnt Rule?

- Start with the most general rule possible: condition = empty
- Adding new attributes by adopting a greedy depth-first strategy
 - Picks the one that most improves the rule quality
- Rule-Quality measures: consider both coverage and accuracy
 - Foil-gain (in FOIL & RIPPER): assesses info_gain by extending condition
 - favors rules that have high accuracy and cover many positive tuples

$$FOIL_Gain = pos' \times (\log_2 \frac{pos'}{pos'+neg'} - \log_2 \frac{pos}{pos+neg})$$

- Rule pruning based on an independent set of test tuples
 - Pos/neg are # of positive/negative tuples covered by R.
 - If FOIL_Prune is higher for the pruned version of R, prune R

$$FOIL_Prune(R) = \frac{pos - neg}{pos + neg}$$

FOIL Example

There are total 44 '+' class records and 6 '-' class records in a data set of 50 records. '-' records are those which are not '+' class.

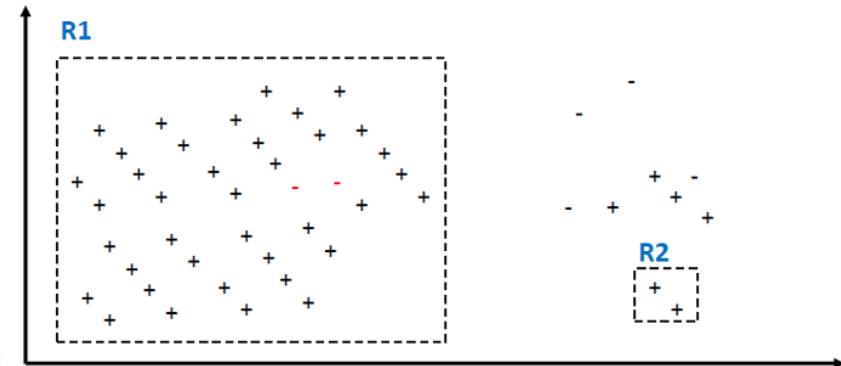
Rule R1 covers 40 records and correctly classifies 38 as +.

$$\text{Accuracy (R1)} = 38/40 = 95\%$$

Rule R2 covers 2 records and correctly classifies all 2 as +.

$$\text{Accuracy (R2)} = 2/2 = 100\%$$

R1 has more coverage but may have classified a few wrong. R2 is more accurate but its coverage is poor.



$$\begin{aligned}\text{FOIL Gain}(R_0, R_1) &= p_x \cdot [\log_2(p_x/(p_x+n_x)) - \log_2(p_0/(p_0+n_0))] \\ &= 38 \cdot [\log_2(38/(38+2)) - \log_2(44/(44+6))] \\ &= 38 \cdot [-0.074 + 0.18] = 4.0\end{aligned}$$

$$\begin{aligned}\text{FOIL Gain}(R_0, R_2) &= p_x \cdot [\log_2(p_x/(p_x+n_x)) - \log_2(p_0/(p_0+n_0))] \\ &= 2 \cdot [\log_2(2/(2+0)) - \log_2(44/(44+6))] \\ &= 2 \cdot [0 + 0.18] = 0.36\end{aligned}$$

FOIL Gain suggests R1 is better!

How to Evaluate Learnt Rule?

- We can use Likelihood Ratio Statistic, which confirms that effect of rule is not attributed to chance, but represents correlation between attribute value and classes.

$$\text{Likelihood_Ratio} = 2 * \sum_{j=1}^m f_i \log_2 \left(\frac{f_i}{e_i} \right)$$

- m is the number of classes
- For tuples satisfying the rule, f_i is the observed frequency of each class among tuples, e_i is the expected frequency if the rule made random predictions
- Higher the Likelihood Ratio, better the rule is.
- Used by CN2

Building Classification Rules

- Direct Method:
 - Extract rules directly from data
 - e.g.: RIPPER, CN2, Holte's 1R
- Indirect Method:
 - Extract rules from other classification models (e.g. decision trees, neural networks, etc).
 - e.g: C4.5rules

Direct Method: Sequential Covering

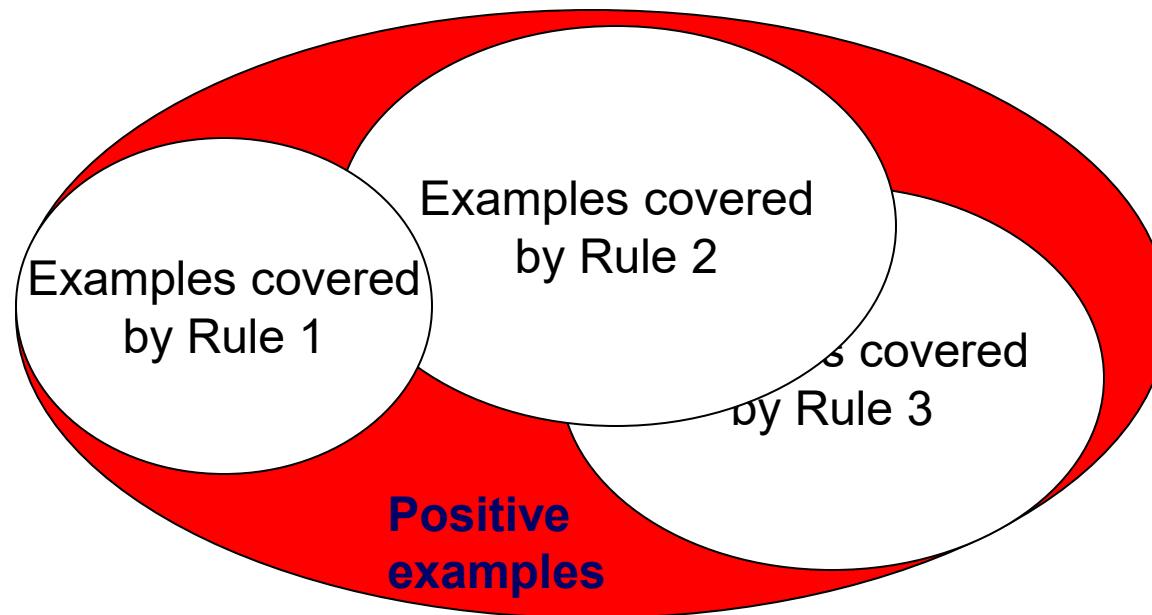
- Sequential covering algorithm: Extracts rules directly from training data
- Typical sequential covering algorithms: FOIL, AQ, CN2, RIPPER
- Rules are learned *sequentially*, each for a given class C_i will cover many tuples of C_i but none (or few) of the tuples of other classes

Rule Induction: Sequential Covering Method

- Start with an empty rule set
- Steps:
 - Rules are learned one at a time
 - Each time a rule is learned, the tuples covered by the rules are removed
 - Repeat the process(above steps) on the remaining tuples
 - until *termination condition*, e.g., when no more training examples or when the quality of a rule returned is below a user-specified threshold
- Comparison with decision-tree induction: learning a set of rules *simultaneously*

Sequential Covering Algorithm

```
while (enough target tuples left)
    generate a rule
    remove positive target tuples satisfying this rule
```



Rule Generation

- To generate a rule

while(true)

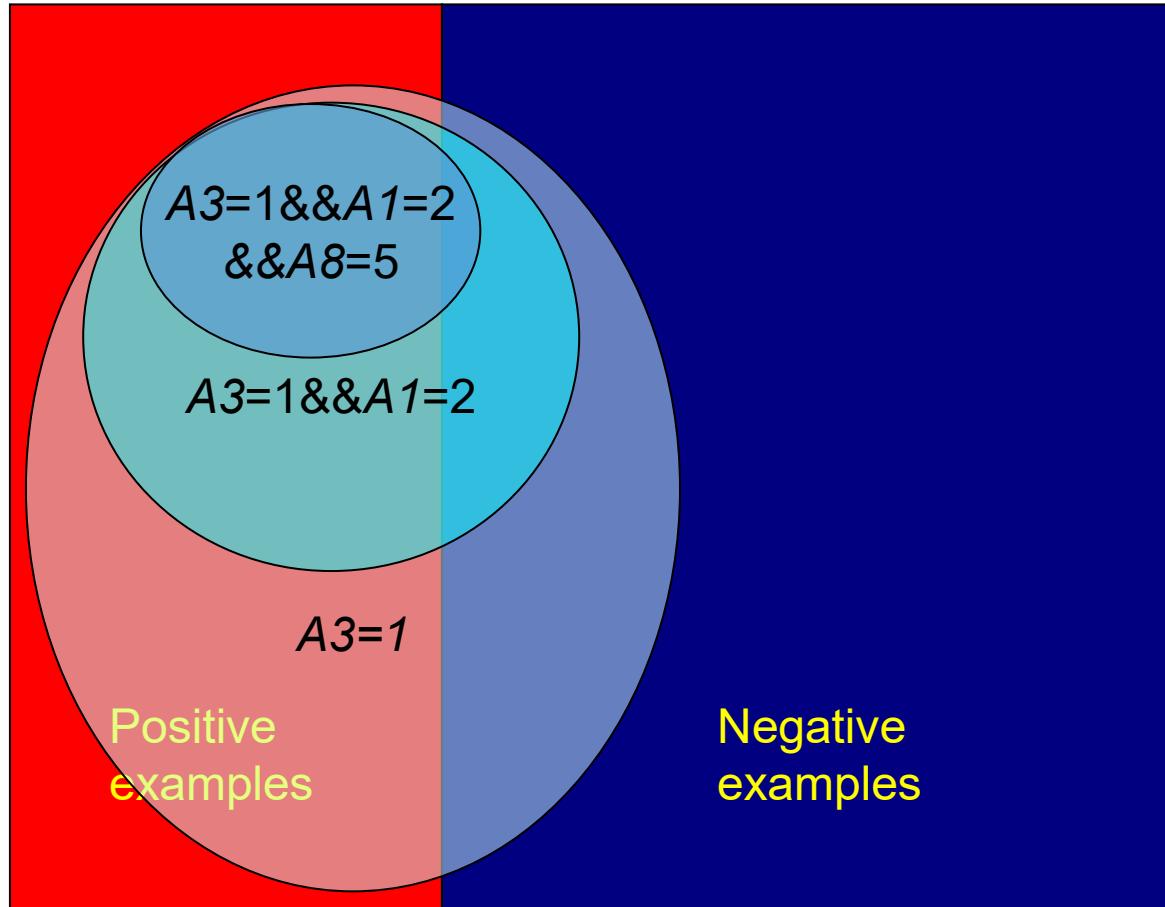
 find the best predicate p

if foil-gain(p) > threshold **then** add p to current rule

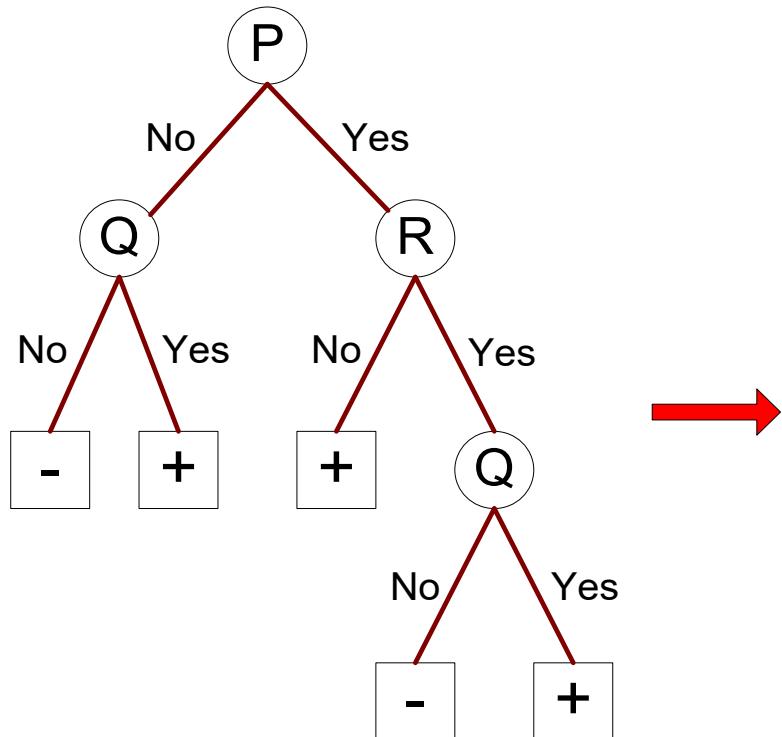
else break

Predicates considered may be independent of each other (as in previous slide) or progressively restrictive (as in the next slide)

Rule Generation



Indirect Methods

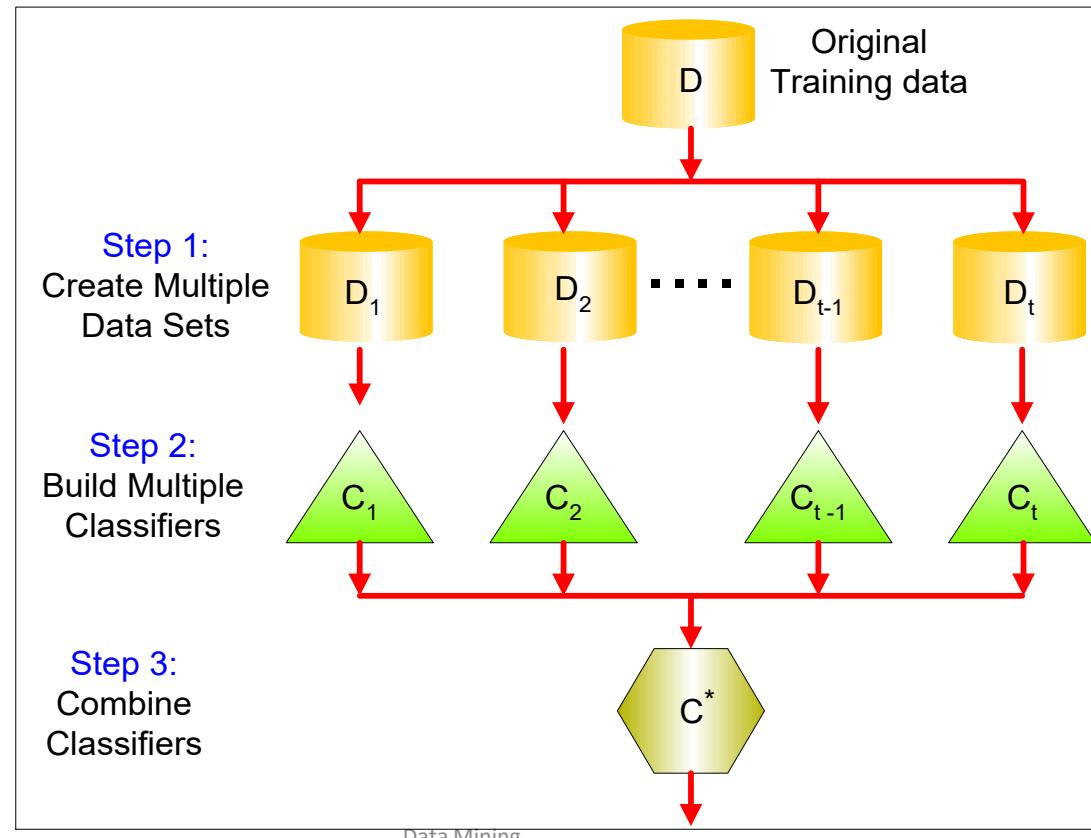


Rule Set

- r1: $(P=\text{No}, Q=\text{No}) \Rightarrow -$
- r2: $(P=\text{No}, Q=\text{Yes}) \Rightarrow +$
- r3: $(P=\text{Yes}, R=\text{No}) \Rightarrow +$
- r4: $(P=\text{Yes}, R=\text{Yes}, Q=\text{No}) \Rightarrow -$
- r5: $(P=\text{Yes}, R=\text{Yes}, Q=\text{Yes}) \Rightarrow +$

Ensemble Methods

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers
- **General Idea**



Why does it work?

- Suppose there are 25 base classifiers
 - Each classifier has error rate, $\varepsilon = 0.35$
 - Assume classifiers are independent
 - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

When error rate differs...

- Suppose there are k base classifiers
 - Each classifier has different error rate, ε_i
 - Again, assume classifiers are independent
 - Probability that the ensemble classifier makes a wrong prediction:
 - Majority of classifiers have to make wrong prediction
 - Compute the probability for each combination that can make wrong prediction (brute force method)
 - Sum up for all possible combinations

Prescribed Text Books

	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers

Thank You



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

S2-21_DSECLZC415

Classification & Prediction



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

Model Evaluation and Selection

Model Evaluation and Selection

Evaluation metrics: How can we measure accuracy? Other metrics to consider?

Use **validation test set** of class-labeled tuples instead of training set when assessing accuracy

Methods for estimating a classifier's accuracy:

- Holdout method, random subsampling
- Cross-validation
- Bootstrap

Comparing classifiers:

- Cost-benefit analysis and ROC Curves

Classifier Evaluation Metrics: Confusion Matrix

Confusion Matrix:

Given m classes, an entry, $\text{CM}_{i,j}$, in a **confusion matrix** indicates # of tuples in class i that were labeled by the classifier as class j

May have extra rows/columns to provide totals

Predicted class ->	C_1	$\neg C_1$
Actual class ↓		
C_1	True Positives (TP)	False Negatives (FN)
$\neg C_1$	False Positives (FP)	True Negatives (TN)

Classifier Evaluation Metrics: Confusion Matrix

Example of Confusion Matrix:

Predicted class ->	buy_computer = yes	buy_computer = no	Total
Actual class ↓			
buy_computer = yes	6954	46	7000
buy_computer = no	412	2588	3000
Total	7366	2634	10000

Classifier Evaluation Metrics: Accuracy, Error Rate, Sensitivity and Specificity

Classifier Accuracy, or recognition rate: percentage of test set tuples that are correctly classified

$$\text{Accuracy} = (\text{TP} + \text{TN})/\text{All}$$

Error rate: $1 - \text{accuracy}$, or

$$\text{Error rate} = (\text{FP} + \text{FN})/\text{All}$$

A\P	C	$\neg C$	
C	TP	FN	P
$\neg C$	FP	TN	N
	P'	N'	All

- **Class Imbalance Problem:**
 - One class may be *rare*, e.g. fraud, or HIV-positive
 - Significant *majority of the negative class* and minority of the positive class
 - **Sensitivity**: True Positive recognition rate
 - **Sensitivity** = TP/P
 - **Specificity**: True Negative recognition rate
 - **Specificity** = TN/N

Classifier Evaluation Metrics: Precision and Recall, and F-measures

Precision: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

Recall: completeness – what % of positive tuples did the classifier label as positive?

Perfect score is 1.0

$$recall = \frac{TP}{TP + FN} = \frac{TP}{P}$$

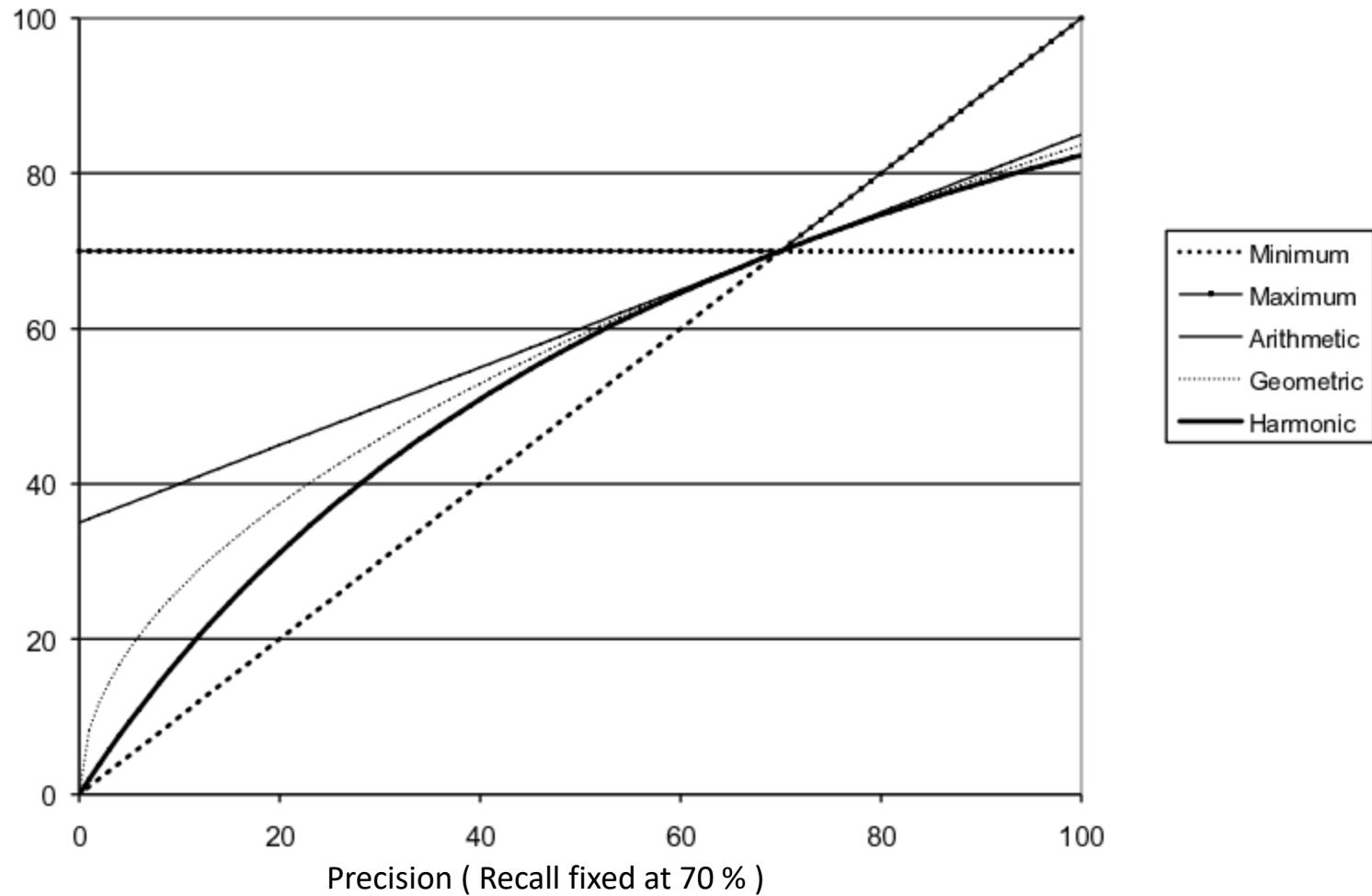
Inverse relationship between precision & recall

F measure (F_1 or F-score): harmonic mean of precision and recall,

$$F = \frac{2 \times precision \times recall}{precision + recall}$$

Why a harmonic mean, but not arithmetic or geometric mean?

Classifier Evaluation Metrics: Precision and Recall, and F-measures



Classifier Evaluation Metrics: Precision and Recall, and F-measures

Harmonic mean can be recomputed by applying weights to precision and recall

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$$

By substituting $\beta^2 = (1 - \alpha)/\alpha$,

We get

F_β : weighted measure of precision and recall
 – assigns β^2 times as much weight to recall as to precision

$$F_\beta = \frac{(1 + \beta^2) \times \text{precision} \times \text{recall}}{\beta^2 \times \text{precision} + \text{recall}},$$

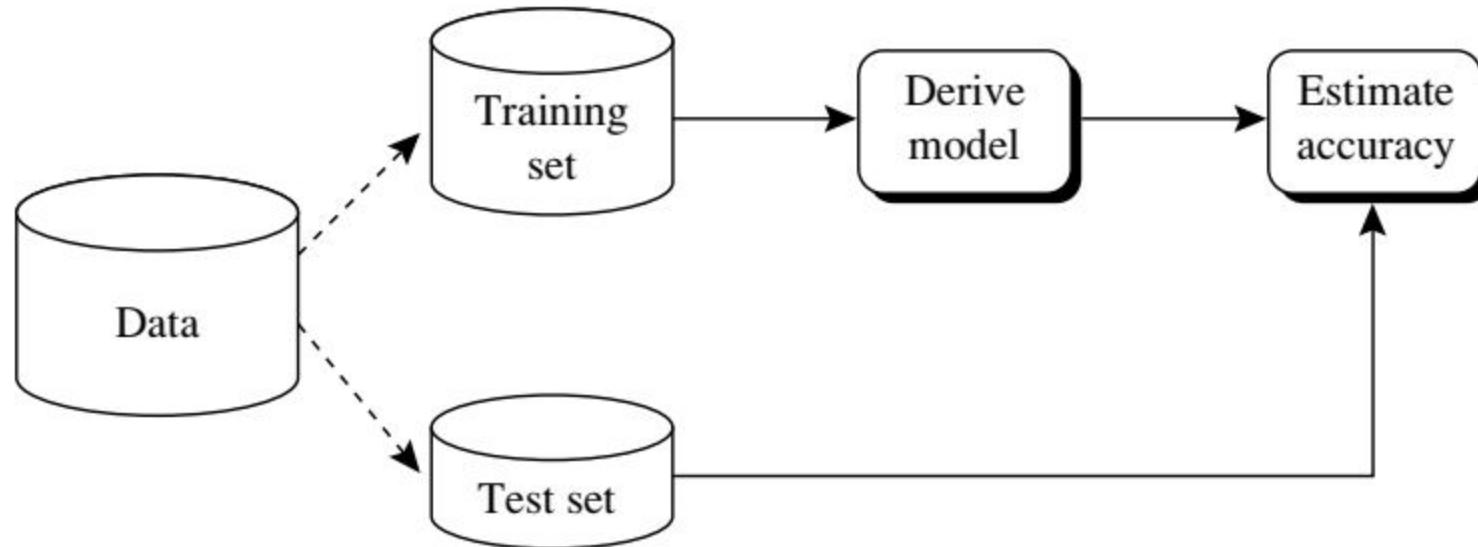
Classifier Evaluation Metrics: Example

Precision = $90/230 = 39.13\%$

Recall = $90/300 = 30.00\%$

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	90	210	300	30.00 <i>(sensitivity)</i>
cancer = no	140	9560	9700	98.56 <i>(specificity)</i>
Total	230	9770	10000	96.40 <i>(accuracy)</i>

Evaluating Classifier Accuracy: Holdout Method



Holdout method

- Given data is randomly partitioned into two independent sets
 - Training set (e.g., 2/3) for model construction
 - Test set (e.g., 1/3) for accuracy estimation
- Random sampling: a variation of holdout
 - Repeat holdout k times, accuracy = avg. of the accuracies obtained

Evaluating Classifier Accuracy: Cross-Validation Methods

Cross-validation (k -fold, where $k = 10$ is most popular)

- Randomly partition the data into k *mutually exclusive* subsets, each approximately equal size
- At i -th iteration, use D_i as test set and others as training set
- Leave-one-out: k folds where $k = \#$ of tuples, for small sized data
- ***Stratified cross-validation***: folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

Evaluating Classifier Accuracy: Bootstrap

Bootstrap

- Works well with small data sets
- Samples the given training tuples uniformly *with replacement*
 - i.e., each time a tuple is selected, it is equally likely to be selected again and re-added to the training set

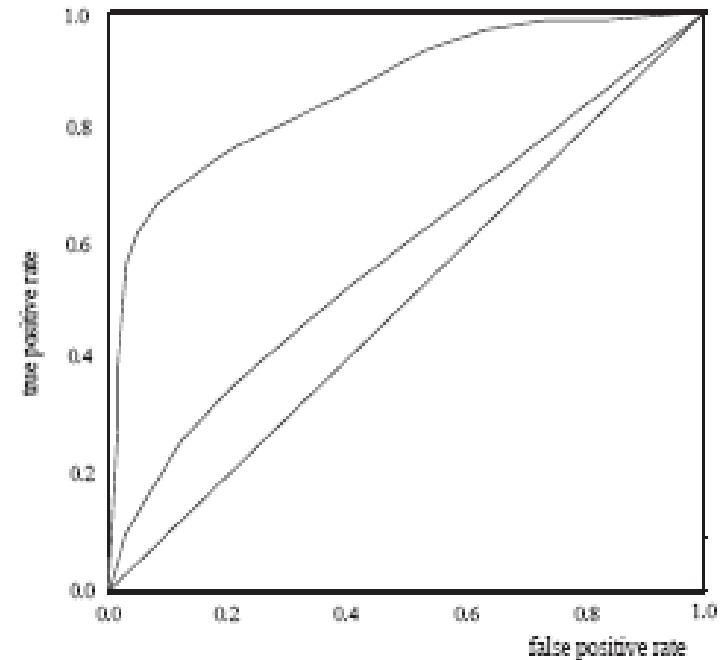
Several bootstrap methods, and a common one is **.632 bootstrap**

- A data set with d tuples is sampled d times, with replacement, resulting in a training set of d samples. The data tuples that did not make it into the training set end up forming the test set. About 63.2% of the original data end up in the bootstrap, and the remaining 36.8% form the test set (since $(1 - 1/d)^d \approx e^{-1} = 0.368$)
- Repeat the sampling procedure k times, overall accuracy of the model:

$$Acc(M) = \frac{1}{k} \sum_{i=1}^k (0.632 \times Acc(M_i)_{test_set} + 0.368 \times Acc(M_i)_{train_set})$$

Model Selection: ROC Curves

- ROC (Receiver Operating Characteristics) curves: for visual comparison of classification models
- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate
- The area under the ROC curve is a measure of the accuracy of the model
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model



- Vertical axis represents the true positive rate
- Horizontal axis rep. the false positive rate
- The plot also shows a diagonal line
- A model with perfect accuracy will have an area of 1.0

Prediction

Source Courtesy: Some of the contents of this PPT are sourced from materials provided by publishers of prescribed books

Prediction vs. Classification

- How is (Numerical) prediction similar to classification?
 - construct a model
 - use model to predict continuous or ordered value for a given input
- Difference between Prediction and classification
 - Classification refers to predict categorical class label
 - Prediction models continuous-valued functions
- Major method for prediction: regression
 - model the relationship between one or more independent or predictor variables and a dependent or response variable
- Profit, sales, mortgage rates, house values, square footage, temperature, or distance could all be predicted using regression techniques. For example, a regression model could be used to predict the value of a house based on location, number of rooms, lot size, and other factors.

Regression for Prediction

- A regression task begins with a data set in which the target values are known, e.g.
 - A regression model that predicts house values could be developed based on observed data for many houses over a period of time.
 - The data might track the age of the house, square footage, number of rooms, taxes, school district, proximity to shopping centers, and so on.
 - House value would be the target, the other attributes would be the predictors, and the data for each house would constitute a case.
- In the model build (training) process, a regression algorithm estimates the value of the target as a function of the predictors for each case in the build data.
 - These relationships between predictors and target are summarized in a model, which can then be applied to a different data set in which the target values are unknown

Prediction Techniques

- Regression analysis
 - Linear and multiple regression
 - Non-linear regression
 - Other regression methods:
 - Log-linear models,
 - Regression trees
 - etc.

Regression Analysis

- Regression analysis seeks to determine the values of parameters for a function that cause the function to best fit a set of data observations that you provide.
- The following equation expresses these relationships in symbols.

$$y = F(x, w) + e$$

- Regression is the process of estimating the value of a continuous target (y) as a function (F) of one or more predictors (x_1, x_2, \dots, x_n), a set of parameters ($w_0, w_1, w_2, \dots, w_n$), and a measure of error (e).

Regression Analysis

- In the equation

$$y = F(x, w) + e$$

- The predictors(x_1, x_2, \dots, x_n) can be understood as independent variables
- The target (y) is the dependent variable.
- The error (e), also called the residual, is the difference between the expected and predicted value of the dependent variable.
- The regression parameters are also known as regression coefficients.
- The process of training a regression model involves finding the parameter values that minimize a measure of the error, for example, the sum of squared errors.

Simple Linear Regression

- Simple Linear regression: involves a response variable y and a single predictor variable x

$$y = w_0 + w_1 x$$

where w_0 (y-intercept) and w_1 (slope) are regression coefficients

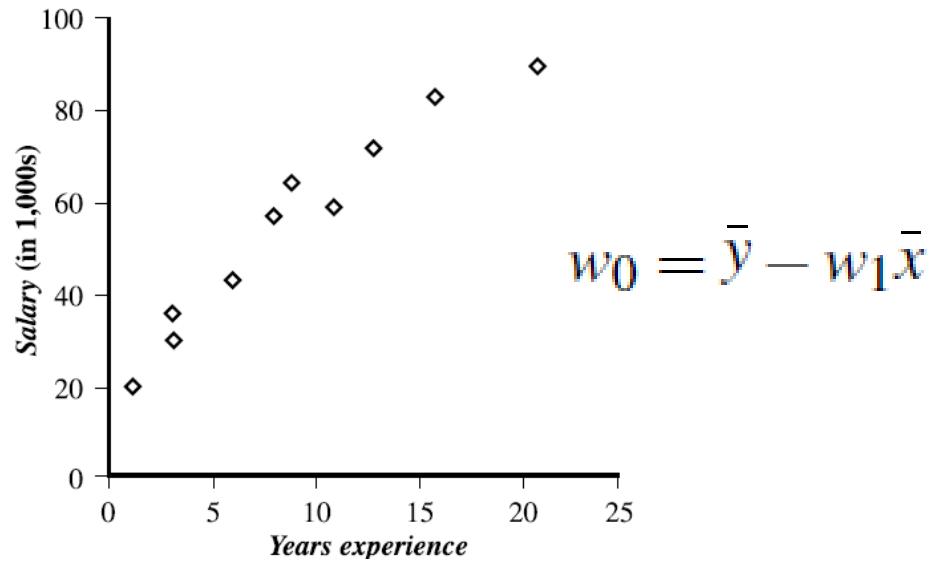
- Method of least squares: estimates the best-fitting straight line

$$w_1 = \frac{\sum_{i=1}^{|D|} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|} (x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - w_1 \bar{x}$$

Linear Regression Example (Method of least squares)

<i>x</i> years experience	<i>y</i> salary (in \$1000s)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83



$$w_0 = \bar{y} - w_1 \bar{x}$$

$$\bar{x} = 9.1 \quad \bar{y} = 55.4$$

What would be a salary of a person with 10 years experience?

Solution

we compute $\bar{x} = 9.1$ and $\bar{y} = 55.4$. Substituting these values into Equations

$$w_1 = \frac{\sum_{i=1}^{|D|} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|} (x_i - \bar{x})^2}$$

$$w_1 = \frac{(3 - 9.1)(30 - 55.4) + (8 - 9.1)(57 - 55.4) + \dots + (16 - 9.1)(83 - 55.4)}{(3 - 9.1)^2 + (8 - 9.1)^2 + \dots + (16 - 9.1)^2} = 3.5$$

$$w_0 = \bar{y} - w_1 \bar{x} \quad w_0 = 55.4 - (3.5)(9.1) = 23.6$$

Thus, the equation of the least squares line is estimated by $y = 23.6 + 3.5x$

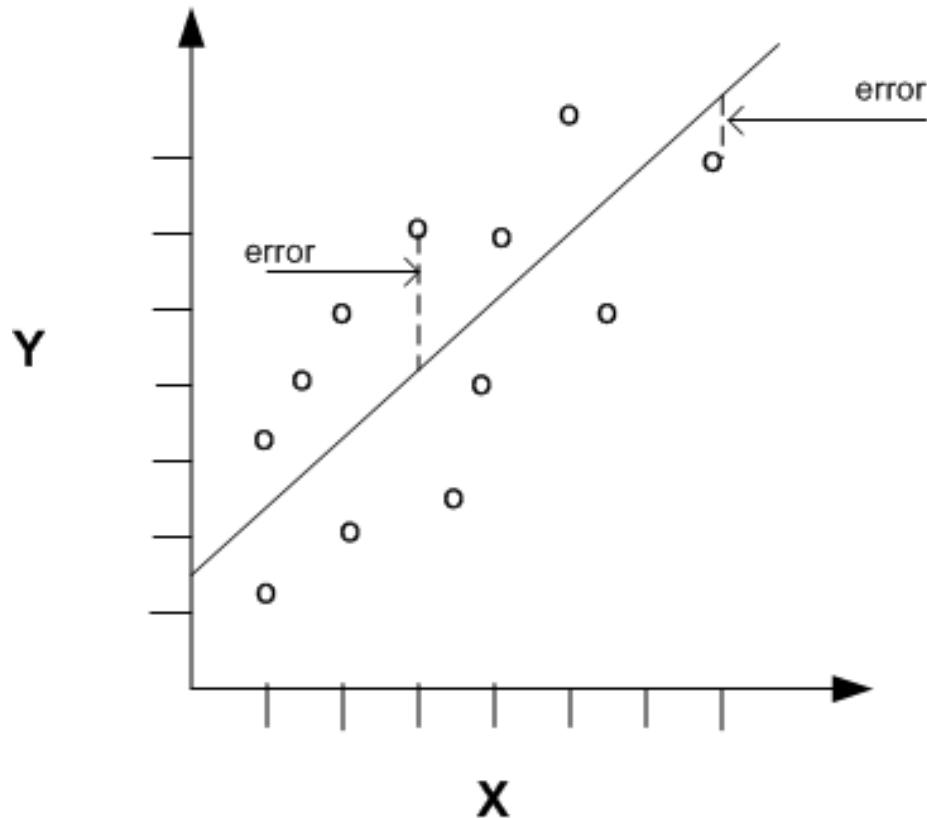
To calculate the salary of a person with 10 yrs experience, put the values in the equation. i.e., $x=10$

The salary $y = 23.6 + (3.5 * 10) = 58.6$ or \$58, 600

x years experience	y salary (in \$1000s)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

$$y = w_0 + w_1 x.$$

Linear Regression With a Single Predictor



Multiple Linear Regression

Multiple linear regression: involves more than one predictor variable

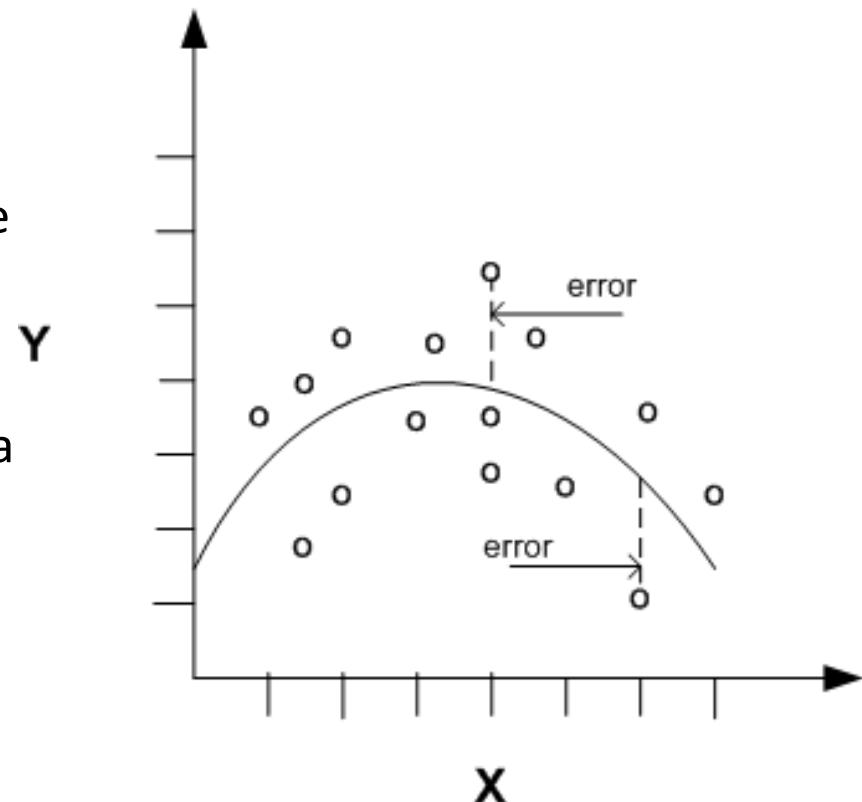
- Training data is of the form $(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \dots, (\mathbf{X}_{|\mathcal{D}|}, y_{|\mathcal{D}|})$
- e.g. For 2-D data, we may have:

$$y = w_0 + w_1 x_1 + w_2 x_2$$

- Solvable by extension of least square method or using SAS, S-Plus
- Many nonlinear functions can be transformed into the above

Nonlinear Regression

- Often the relationship between x and y cannot be approximated with a straight line. In this case, a nonlinear regression technique may be used. Alternatively, the data could be preprocessed to make the relationship linear.
- Nonlinear regression models define y as a function of x using an equation that is more complicated than the linear regression equation



Nonlinear Regression

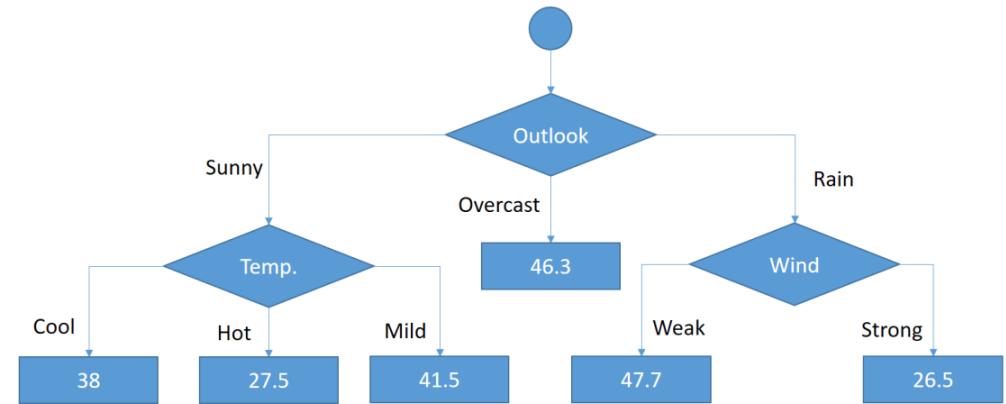
- Some nonlinear models can be modeled by a polynomial function
- A polynomial regression model can be transformed into linear regression model. For example,
 - $y = w_0 + w_1 x + w_2 x^2 + w_3 x^3$
 - convertible to linear with new variables: $x_2 = x^2, x_3 = x^3$
 - $y = w_0 + w_1 x + w_2 x_2 + w_3 x_3$
- Other functions, such as power function, can also be transformed to linear model
- Some models are intractable nonlinear (e.g., sum of exponential terms)
 - possible to obtain least square estimates through extensive calculation on more complex formulae

Regression Trees and Model Trees

- Regression tree: proposed in CART system (Breiman et al. 1984)
 - CART: Classification And Regression Trees
 - Each leaf stores a continuous-valued prediction
 - It is the average value of the predicted attribute for the training tuples that reach the leaf
- Model tree: proposed by Quinlan (1992)
 - Each leaf holds a regression model—a multivariate linear equation for the predicted attribute
 - A more general case than regression tree
- Regression and model trees tend to be more accurate than linear regression when the data are not represented well by a simple linear model

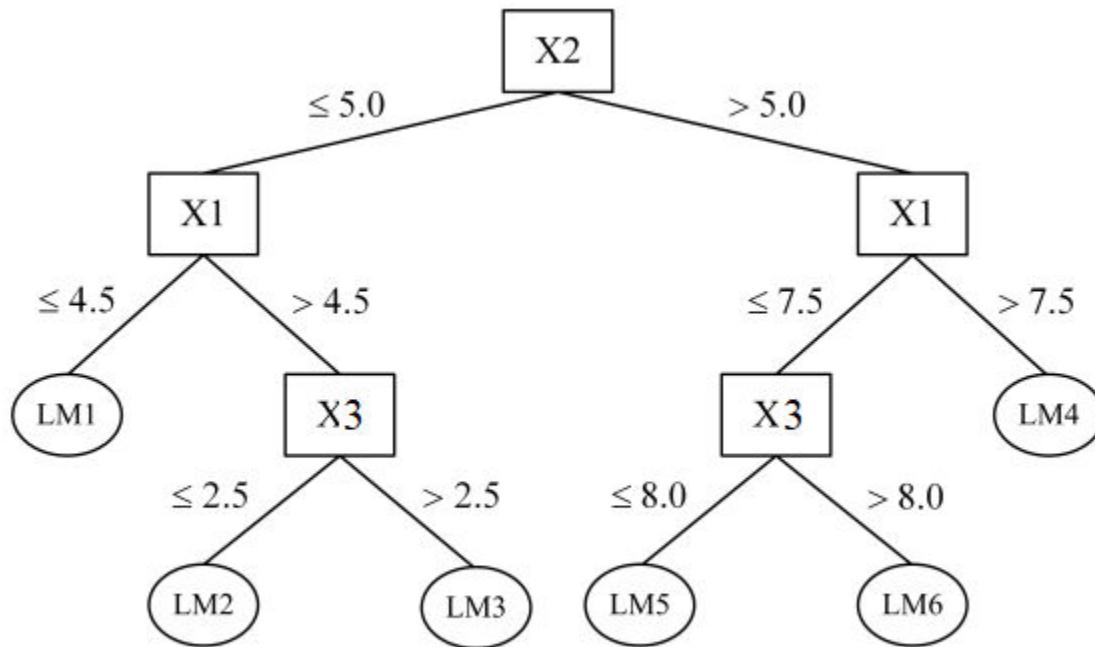
Regression Trees

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44
14	Rain	Mild	High	Strong	30



<https://sefiks.com/2018/08/28/a-step-by-step-regression-decision-tree-example/>

Model Tree Sample



$LM1, LM2, \dots, LM6$ are distinct linear models

https://www.researchgate.net/publication/289078955_Estimation_of_Pan_Coefficient_using_M5_Model_Tree/link/593fde58458515a62183cd38/download

Prescribed Text Books

	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
R2	Principles of Data Mining, Second Edition by Max Bramer Springer © 2013
R1	Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner by Vijay Kotu and Bala Deshpande Morgan Kaufmann Publishers



S2-21_DSECLZC415

Association Analysis

BITS Pilani

Pilani|Dubai|Goa|Hyderabad



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

Association Analysis Basics

Source Courtesy: Some of the contents of this PPT are sourced from materials provided by publishers of prescribed books

3

Association Analysis

Association analysis measures the strength of co-occurrence between one item and another.

- The objective of this class of data mining algorithms is not to predict an occurrence of an item, like classification or regression do, but to find usable patterns in the co-occurrences of the items.
- Association rules learning is a branch of an unsupervised learning process that discovers hidden patterns in data, in the form of easily recognizable *rules*

Association algorithms are widely used in retail analysis of transactions, recommendation engines, and online clickstream analysis across web pages.

- One of the popular applications of this technique is called *market basket analysis*, which finds co-occurrences of one retail item with another item within the same retail purchase transaction

Retailer can take advantage of this association for bundle pricing, product placement, and even shelf space optimization within the store layout.

Association Rule Mining

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

Market-Basket transactions

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

Example of Association Rules

$\{\text{Diaper}\} \rightarrow \{\text{Butter}\}$,
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Beans, Coke}\}$,
 $\{\text{Butter, Bread}\} \rightarrow \{\text{Milk}\}$,

Implication means co-occurrence,
not causality!

Definition: Frequent Itemset

Itemset

- A collection of one or more items
 - Example: {Milk, Bread, Diaper}
- k-itemset
 - An itemset that contains k items

Support count (σ)

- Frequency of occurrence of an itemset
- E.g. $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$

Support

- Fraction of transactions that contain an itemset
- E.g. $s(\{\text{Milk, Bread, Diaper}\}) = 2/5$

Frequent Itemset

- An itemset whose support is greater than or equal to a *minsup* threshold

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

Definition: Association Rule

- Association Rule
 - An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
 - Example:
 $\{Milk, Diaper\} \rightarrow \{Butter\}$

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

- Rule Evaluation Metrics
 - Support (s)
 - ◆ Fraction of transactions that contain both X and Y
 - Confidence (c)
 - ◆ Measures how often items in Y appear in transactions that contain X

Example:

$$\{Milk, Diaper\} \Rightarrow Butter$$

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

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

Association Rule Mining Task

Given a set of transactions T, the goal of association rule mining is to find all rules having

- support $\geq \text{minsup}$ threshold
- confidence $\geq \text{minconf}$ threshold

Brute-force approach:

- List all possible association rules
 - Compute the support and confidence for each rule
 - Prune rules that fail the minsup and minconf thresholds
- ⇒ Computationally prohibitive!

Mining Association Rules

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

Example of Rules:

$\{\text{Milk}, \text{Diaper}\} \rightarrow \{\text{Butter}\}$ ($s=0.4, c=0.67$)
 $\{\text{Milk}, \text{Butter}\} \rightarrow \{\text{Diaper}\}$ ($s=0.4, c=1.0$)
 $\{\text{Diaper}, \text{Butter}\} \rightarrow \{\text{Milk}\}$ ($s=0.4, c=0.67$)
 $\{\text{Butter}\} \rightarrow \{\text{Milk}, \text{Diaper}\}$ ($s=0.4, c=0.67$)
 $\{\text{Diaper}\} \rightarrow \{\text{Milk}, \text{Butter}\}$ ($s=0.4, c=0.5$)
 $\{\text{Milk}\} \rightarrow \{\text{Diaper}, \text{Butter}\}$ ($s=0.4, c=0.5$)

Observations:

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

Mining Association Rules

Two-step approach:

1. Frequent Itemset Generation

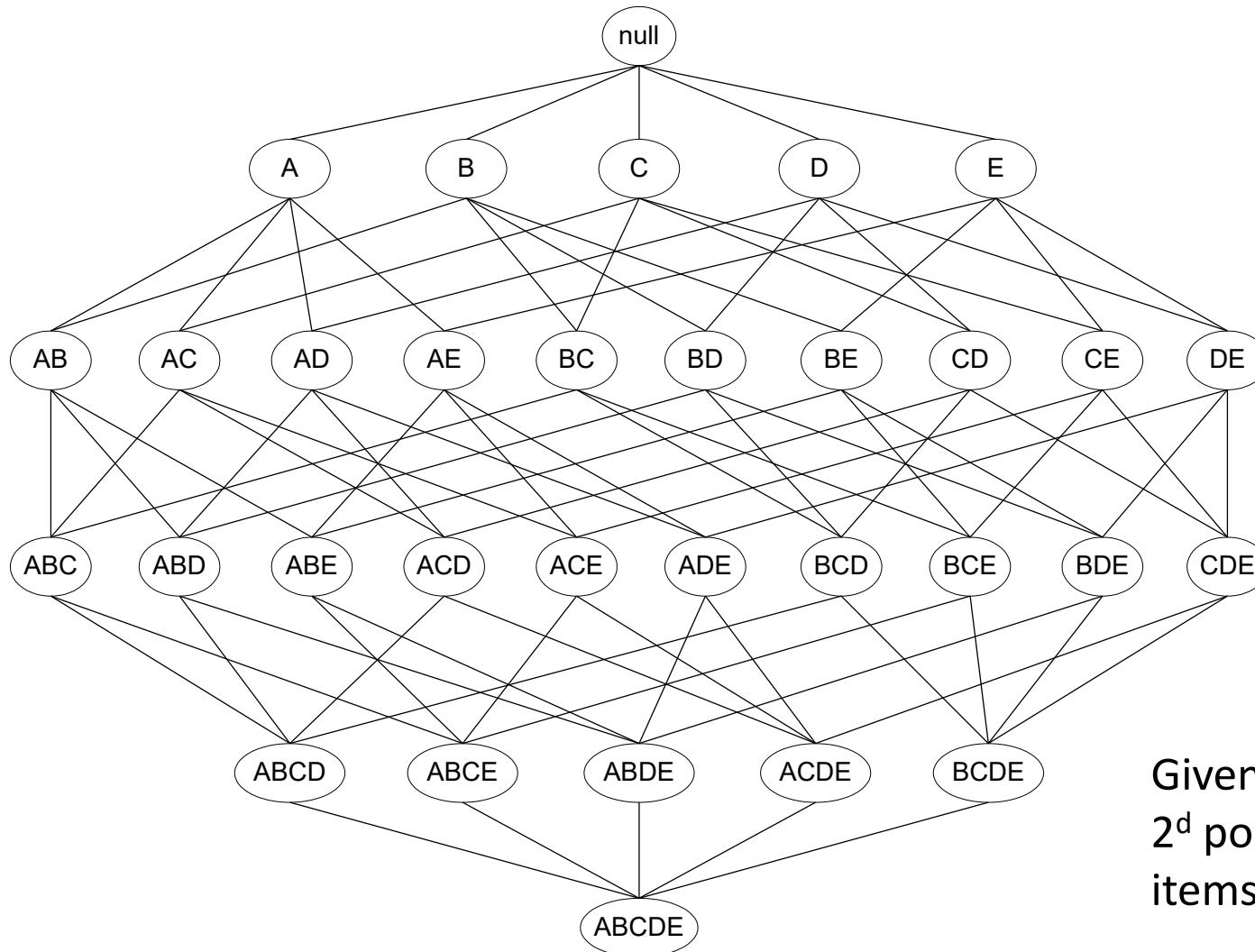
- Generate all itemsets whose support $\geq \text{minsup}$

2. Rule Generation

- Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

Frequent itemset generation is still computationally expensive

Frequent Itemset Generation

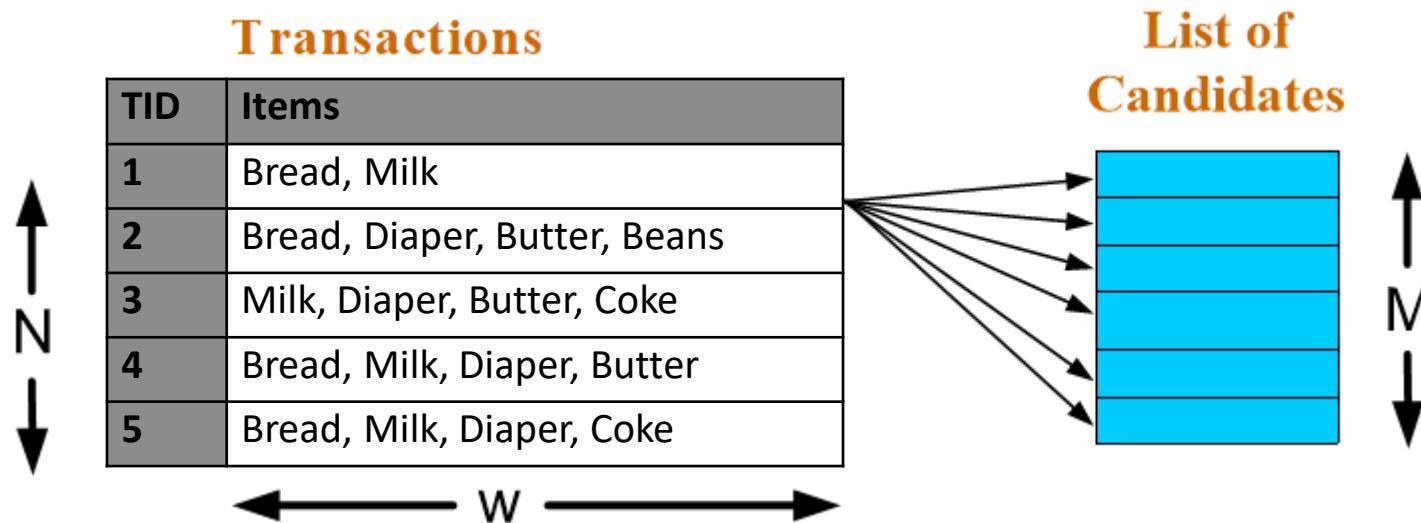


Given d items, there are 2^d possible candidate itemsets

Frequent Itemset Generation

Brute-force approach:

- Each itemset in the lattice is a **candidate** frequent itemset
- Count the support of each candidate by scanning the database



- Match each transaction against every candidate
- Complexity $\sim O(NMw)$ => **Expensive since $M = 2^d$!!!**

Frequent Itemset Generation Strategies

Reduce the **number of candidates** (M)

- Complete search: $M=2^d$
- Use pruning techniques to reduce M

Reduce the **number of transactions** (N)

- Reduce size of N as the size of itemset increases
- Used by DHP(Direct Hashing & Pruning) and vertical-based mining algorithms

Reduce the **number of comparisons** (NM)

- Use efficient data structures to store the candidates or transactions
- No need to match every candidate against every transaction

Apriori Algorithm

Mining Association Rules

Two-step approach:

1. Frequent Itemset Generation

- Generate all itemsets whose support $\geq \text{minsup}$

2. Rule Generation

- Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

Frequent itemset generation is still computationally expensive

- *Apriori* principle can be used to reduce computations

Reducing Number of Candidates

Apriori principle:

- If an itemset is frequent, then all of its subsets must also be frequent

Apriori principle holds due to the following property of the support measure:

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

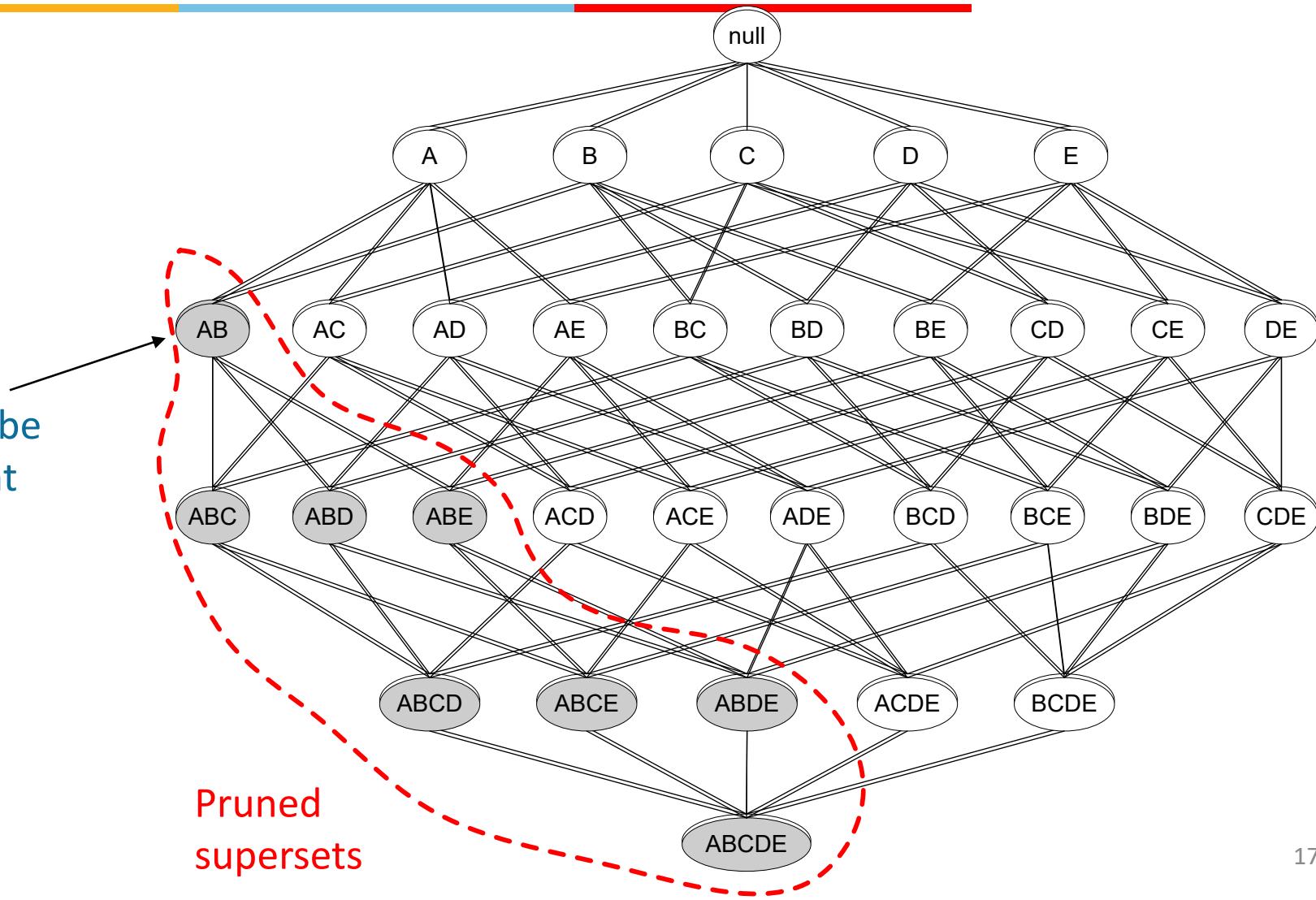
- Support of an itemset never exceeds the support of its subsets
- This is known as the anti-monotone property of support

The Apriori algorithm was proposed by Rakesh Agrawal and Ramakrishnan Srikant in 1994

Illustrating Apriori Principle

Found to be
Infrequent

Pruned
supersets



Apriori: A Candidate Generation-and-Test Approach

Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested!

Method:

Initially, scan DB once to get frequent 1-itemset

Generate length (k+1) candidate itemsets from length k frequent itemsets

Test the candidates against DB

Terminate when no frequent or candidate set can be generated

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

Apriori Algorithm

Method:

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

Important Details of Apriori

How to generate candidates?

- Step 1: self-joining L_k
- Step 2: pruning

Example of Candidate-generation

- $L_3 = \{abc, abd, acd, ace, bcd\}$
- Self-joining: $L_3 * L_3$
 - $abcd$ from abc and abd
 - $acde$ from acd and ace
- Pruning:
 - $acde$ is removed because ade is not in L_3
- $C_4 = \{abcd\}$

How to count supports of candidates?

Illustrating Apriori Principle

Item	Count
Bread	4
Coke	2
Milk	4
Butter	3
Diaper	4
Beans	1

Minimum Support = 3

Items (1-itemsets)



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

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

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

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Beans)



Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	2

Factors Affecting Complexity

Choice of minimum support threshold

- lowering support threshold results in more frequent itemsets
- this may increase number of candidates and max length of frequent itemsets

Dimensionality (number of items) of the data set

- more space is needed to store support count of each item
- if number of frequent items also increases, both computation and I/O costs may also increase

Size of database

- since Apriori makes multiple passes, run time of algorithm may increase with number of transactions

Average transaction width

- transaction width increases with denser data sets
- This may increase max length of frequent itemsets and traversals of hash tree (number of subsets in a transaction increases with its width)

Can we improve Apriori Efficiency?

Hash-based technique

Transaction reduction

Partitioning

Sampling

Dynamic itemset counting



Hash-based technique

Create hash table using hash function

$$h(x, y) = ((\text{order of } x) * 10 + (\text{order of } y)) \bmod 7$$

TID	List of item IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

bucket address	0	1	2	3	4	5	6
bucket count	2	2	4	2	2	4	4
bucket contents	{I1, I4}	{I1, I5}	{I2, I3}	{I2, I4}	{I2, I5}	{I1, I2}	{I1, I3}
	{I3, I5}	{I1, I5}	{I2, I3}	{I2, I4}	{I2, I5}	{I1, I2}	{I1, I3}

A 2-itemset with a corresponding bucket count in the hash table that is below the support threshold cannot be frequent

Efficiency Techniques

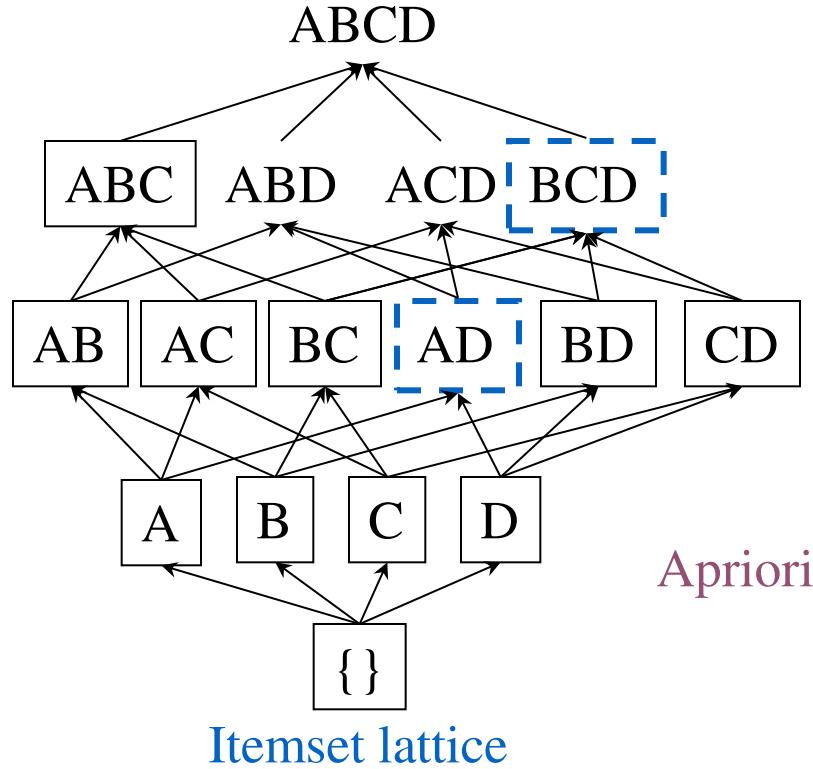
Transaction reduction

- A transaction that does not contain any frequent k -itemsets cannot contain any frequent $(k + 1)$ -itemsets. Such a transaction can be removed from further consideration

Dynamic Itemset Counting

- Instead of counting for the entire database, promote a candidate to frequent itemset if it passes a (lower) threshold after partial counting. Afterwards, generate larger patterns using the itemset, so promoted.

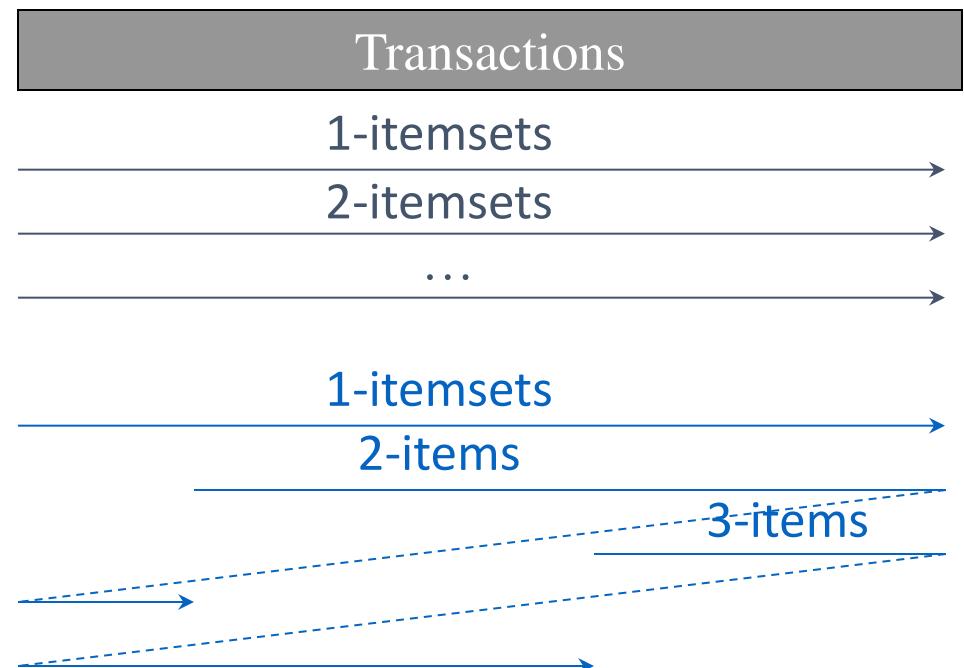
Efficiency Techniques - Dynamic Itemset Counting



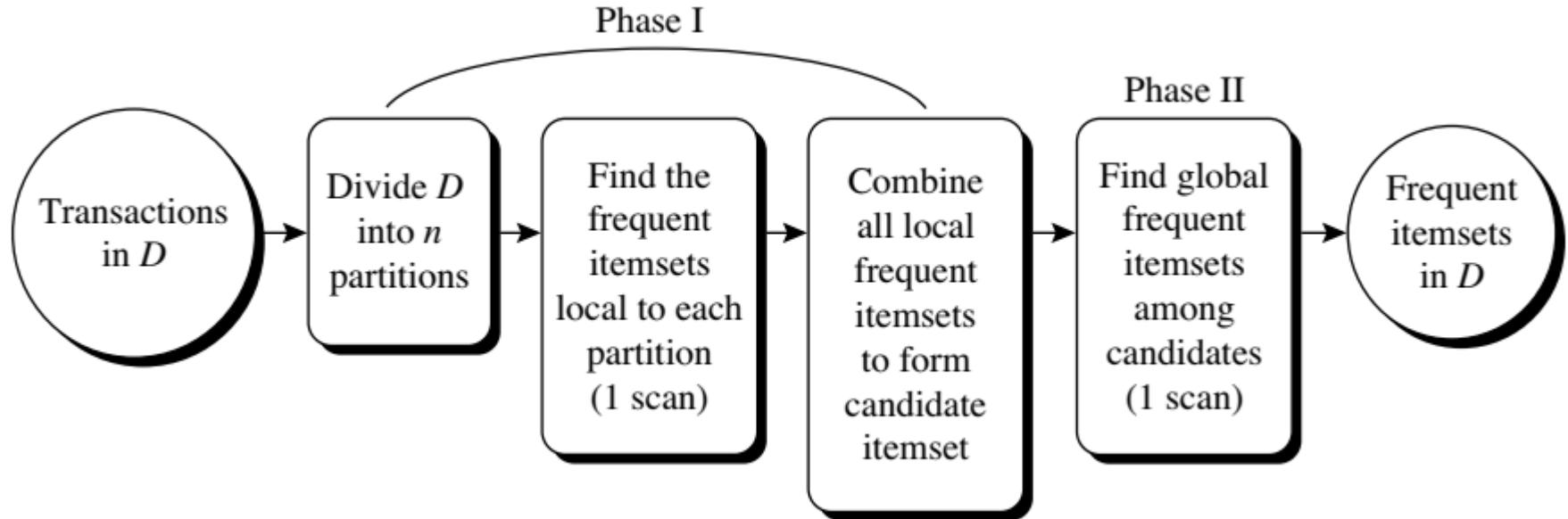
S. Brin R. Motwani, J. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. *SIGMOD'97*

Once both A and D are determined frequent, the counting of AD begins

Once all length-2 subsets of BCD are determined frequent, the counting of BCD begins, thus reducing effective number of scans



Efficiency Techniques



Mining by partitioning the data

- It has two phases. In phase I, divide the transactions of D into n partitions. Each partition has proportionally lower threshold. For each partition, all the *local frequent itemsets* are found.
- Any itemset that is potentially frequent with respect to D must be a frequent itemset in at least one of the partitions. Therefore, all local frequent itemsets are candidate itemsets with respect to D

Mining with Vertical format

Vertical format: $t(AB) = \{T_{11}, T_{25}, \dots\}$

- tid-list: list of trans.-ids containing an itemset

Deriving frequent patterns based on vertical intersections

- $t(X) = t(Y)$: X and Y always happen together
- $t(X) \subset t(Y)$: transaction having X always has Y

Using **diffset** to accelerate mining

- Only keep track of differences of tids
- $t(X) = \{T_1, T_2, T_3\}$, $t(XY) = \{T_1, T_3\}$
- Diffset $(XY, X) = \{T_2\}$
 - Diffset can reduce space complexity

Mining with Vertical format

Minimum support = 2

<i>TID</i>	<i>List of item_IDs</i>
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3



<i>itemset</i>	<i>TID_set</i>
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}

Mining with Vertical format

<i>itemset</i>	<i>TID_set</i>
I1	{T100, T400, T500, T700, T800, T900}
I2	{T100, T200, T300, T400, T600, T800, T900}
I3	{T300, T500, T600, T700, T800, T900}
I4	{T200, T400}
I5	{T100, T800}



<i>itemset</i>	<i>TID_set</i>
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I5}	{T800}

Notice that because the itemsets {I1, I4} and {I3, I5} each contain only one transaction, they do not belong to the set of frequent 2-itemsets.

Mining with Vertical format

<i>itemset</i>	<i>TID_set</i>
{I1, I2}	{T100, T400, T800, T900}
{I1, I3}	{T500, T700, T800, T900}
{I1, I4}	{T400}
{I1, I5}	{T100, T800}
{I2, I3}	{T300, T600, T800, T900}
{I2, I4}	{T200, T400}
{I2, I5}	{T100, T800}
{I3, I5}	{T800}



<i>itemset</i>	<i>TID_set</i>
{I1, I2, I3}	{T800, T900}
{I1, I2, I5}	{T100, T800}

Closed Patterns and Max-Patterns

A long pattern contains a combinatorial number of sub-patterns, e.g., $\{a_1, \dots, a_{100}\}$ contains

$${}^{100}C_1 + {}^{100}C_2 + \dots + {}^{100}C_{100} = 2^{100} - 1 = 1.27 * 10^{30} \text{ sub-patterns!}$$

Solution: *Mine closed frequent patterns and maximal frequent patterns instead*

- An itemset X is closed if X is *frequent* and there exists *no super-pattern Y ⊃ X, with the same support as X*
- An itemset X is a maximal pattern if X is frequent and there exists no frequent super-pattern Y ⊃ X

Closed pattern is a lossless compression of freq. patterns

- Reducing the # of patterns and rules

Closed Patterns and Max-Patterns

Example

- DB = { $\langle a_1 \dots, a_{100} \rangle$, $\langle a_1 \dots, a_{100} \rangle$, $\langle a_1, \dots, a_{50} \rangle$ }
- Min_sup = 2

What is the set of closed itemset?

- $\langle a_1, \dots, a_{100} \rangle$: 2
- $\langle a_1, \dots, a_{50} \rangle$: 3

What is the set of maximal pattern?

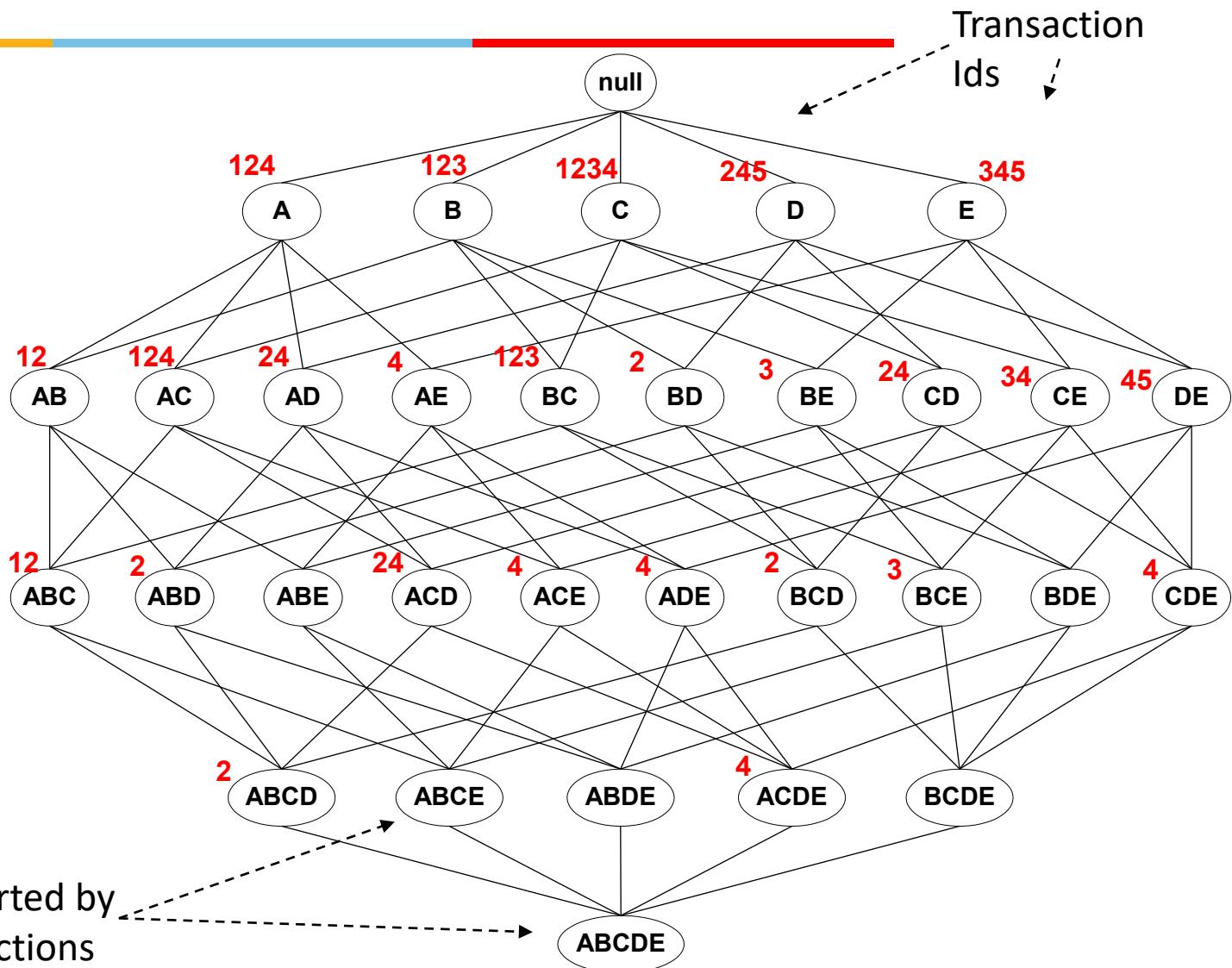
- $\langle a_1, \dots, a_{100} \rangle$: 2

What is the set of all patterns?

- 1.27×10^{30}

Maximal vs Closed Itemsets

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE

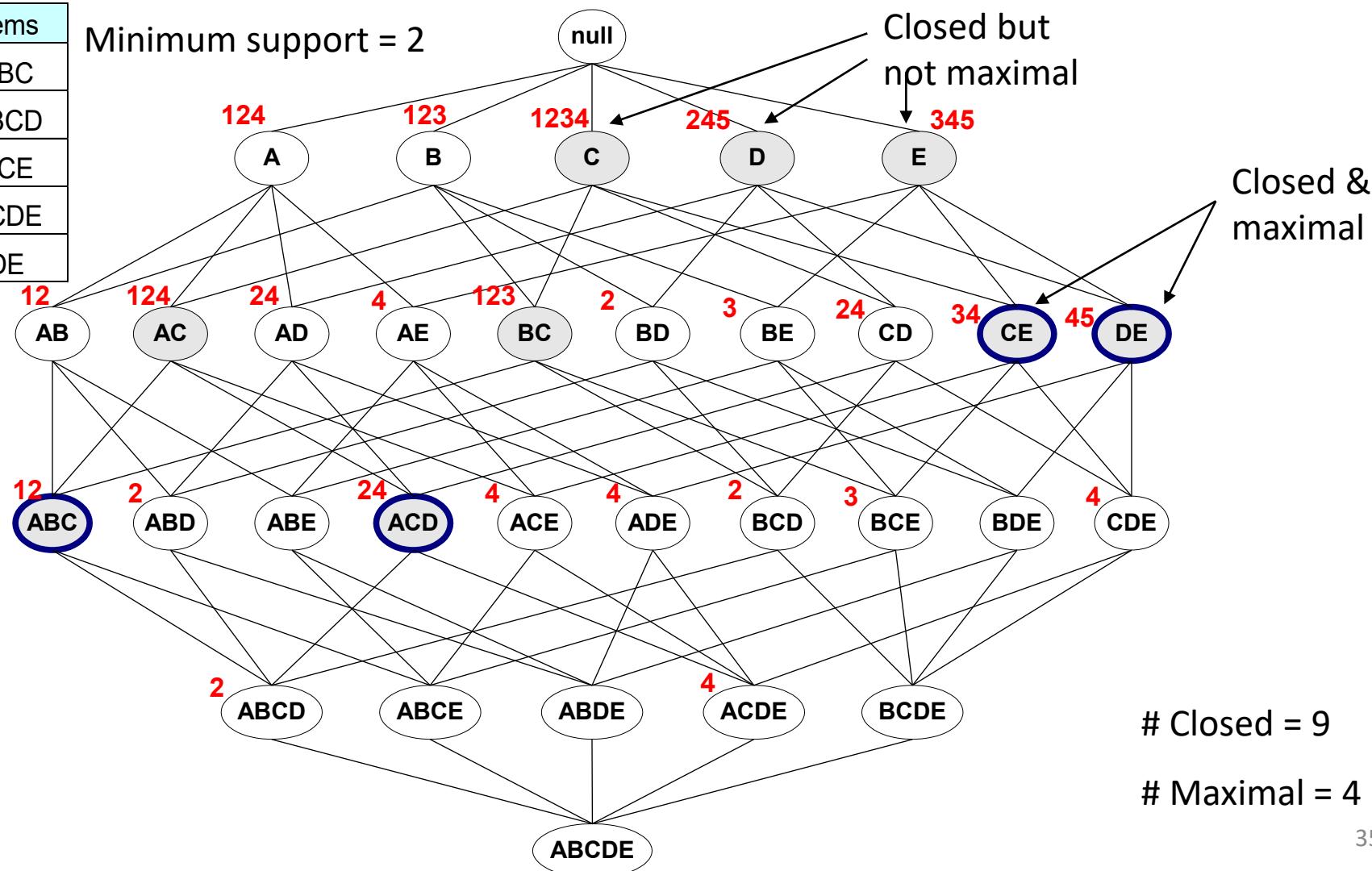


Not supported by
any transactions

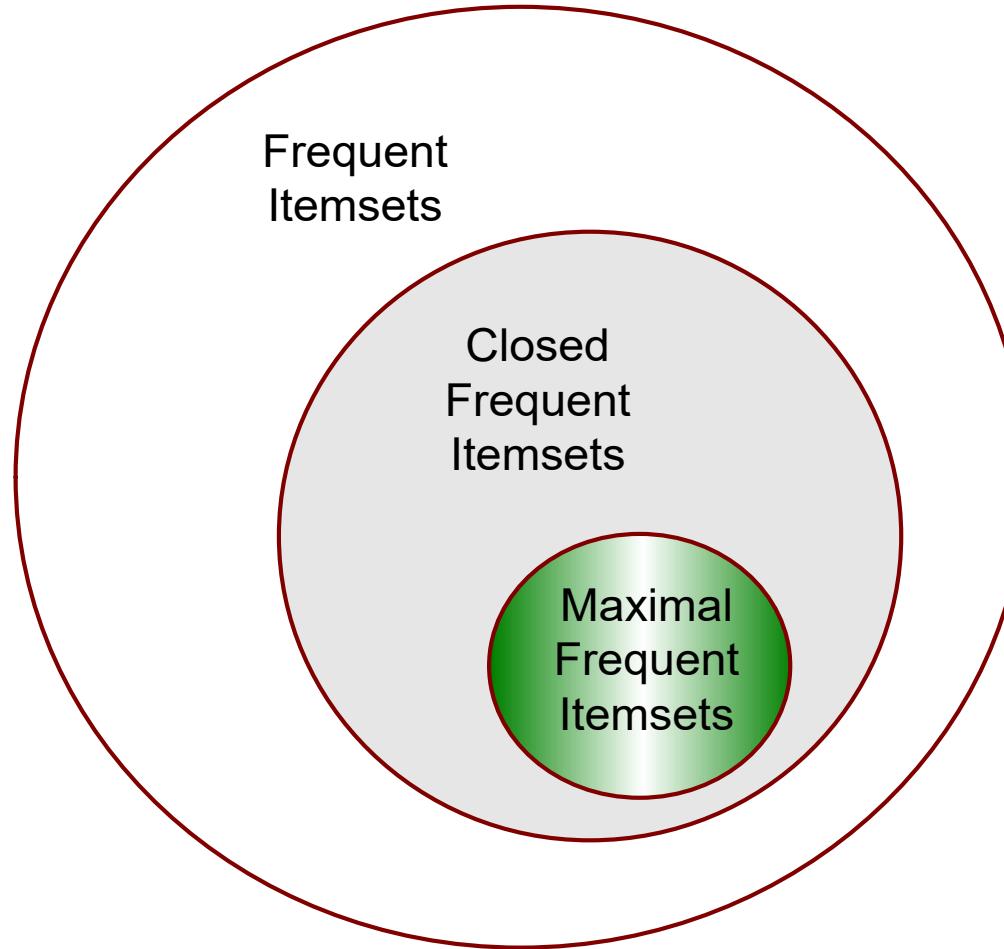
Maximal vs Closed Frequent Itemsets

TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE

Minimum support = 2



Maximal vs Closed Itemsets



Prescribed Text Books

	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
R1	Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner by Vijay Kotu and Bala Deshpande Morgan Kaufmann Publishers

Thank You



BITS Pilani

Pilani | Dubai | Goa | Hyderabad

S2-21_DSECFZC415: Data Mining (Lecture #9 – Association Analysis)



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

Association Analysis (Review)

Association analysis measures the strength of co-occurrence between one item and another.

- The objective of this class of data mining algorithms is not to predict an occurrence of an item, like classification or regression do, but to find usable patterns in the co-occurrences of the items.
- Association rules learning is a branch of an unsupervised learning process that discovers hidden patterns in data, in the form of easily recognizable *rules*

Association algorithms are widely used in retail analysis of transactions, recommendation engines, and online clickstream analysis across web pages.

- One of the popular applications of this technique is called *market basket analysis*, which finds co-occurrences of one retail item with another item within the same retail purchase transaction

Retailer can take advantage of this association for bundle pricing, product placement, and even shelf space optimization within the store layout.

Association Rule Mining (*Review*)

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

Market-Basket transactions

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

Example of Association Rules

$\{\text{Diaper}\} \rightarrow \{\text{Butter}\}$,
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Beans, Coke}\}$,
 $\{\text{Butter, Bread}\} \rightarrow \{\text{Milk}\}$,

Implication means co-occurrence,
not causality!

Definition: Frequent Itemset (Review)

Itemset

- A collection of one or more items
 - Example: {Milk, Bread, Diaper}
- k-itemset
 - An itemset that contains k items

Support count (σ)

- Frequency of occurrence of an itemset
- E.g. $\sigma(\{\text{Milk, Bread, Diaper}\}) = 2$

Support

- Fraction of transactions that contain an itemset
- E.g. $s(\{\text{Milk, Bread, Diaper}\}) = 2/5$

Frequent Itemset

- An itemset whose support is greater than or equal to a *minsup* threshold

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

Definition: Association Rule (Review)

- **Association Rule**

- An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
- Example:
 $\{Milk, Diaper\} \rightarrow \{Butter\}$

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

- **Rule Evaluation Metrics**

- Support (s)
 - ◆ Fraction of transactions that contain both X and Y
- Confidence (c)
 - ◆ Measures how often items in Y appear in transactions that contain X

Example:

$$\{Milk, Diaper\} \Rightarrow Butter$$

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

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

Mining Association Rules

Two-step approach:

1. Frequent Itemset Generation
 - Generate all itemsets whose support $\geq \text{minsup}$
2. Rule Generation
 - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

Frequent itemset generation is computationally expensive

- Can we avoid it?

Bottleneck of Frequent-pattern Mining

Multiple database scans are costly

Mining long patterns needs many passes of scanning and generates lots of candidates

- To find frequent itemset $i_1 i_2 \dots i_{100}$
 - # of scans: 100
 - # of Candidates: ${}^{100}C_1 + {}^{100}C_2 + \dots + {}^{100}C_{100} = 2^{100} - 1 = 1.27 * 10^{30}$!

Bottleneck: candidate-generation-and-test

- Can we avoid candidate generation?



FP-growth Algorithm

Motivation

- Apriori algorithm needs a lot of candidate itemset generation that is computationally prohibitive
 - Moreover, it need to scan the database multiple times
- An algorithm for mining Frequent Itemsets without Candidate Generation – frequent pattern growth, or simply FP-growth
- Bottlenecks of the Apriori approach
 - Breadth-first (i.e., level-wise) search
 - Candidate generation and test
 - Often generates a huge number of candidates
- FP-growth adopts a divide-and-conquer strategy as follows.
 - First, it compresses the database representing frequent items into a frequent pattern tree, or FP-tree, which retains the itemset association information.
 - It then divides the compressed database into a set of conditional databases (a special kind of projected database), each associated with one frequent item or “pattern fragment,” and mines each database separately.
 - For each “pattern fragment,” only its associated data sets need to be examined.
- The FP-Growth Approach (J. Han, J. Pei, and Y. Yin, SIGMOD’ 00)
 - Depth-first search
 - Avoid explicit candidate generation
- Therefore, this approach may substantially reduce the size of the data sets to be searched, along with the “growth” of patterns being examined.

FP-growth Algorithm

Two step approach:

- Step 1: Build a compact data structure called the FP-tree (Use a compressed representation of the database using an FP-tree). Built using 2 passes over the data-set.
 - Scan DB once to determine the support count of each item, find frequent 1-itemset (single item pattern).
 - Sort frequent items in frequency descending order, f-list
 - Infrequent items are discarded, while the frequent items are sorted in decreasing support counts.
 - Scan DB again, construct FP-tree
 - As the transactions are read, before being processed, their items are sorted according to the above order.
- Step 2: Once an FP-tree has been constructed, use a recursive divide-and-conquer approach to mine the frequent itemsets. Extract frequent itemsets directly from the FP-tree
 - Mining Frequent Patterns Without Candidate Generation: Grow long patterns from short ones using local frequent items
 - “abc” is a frequent pattern
 - Get all transactions having “abc”: DB|abc
 - “d” is a local frequent item in DB|abc → abcd is a frequent pattern

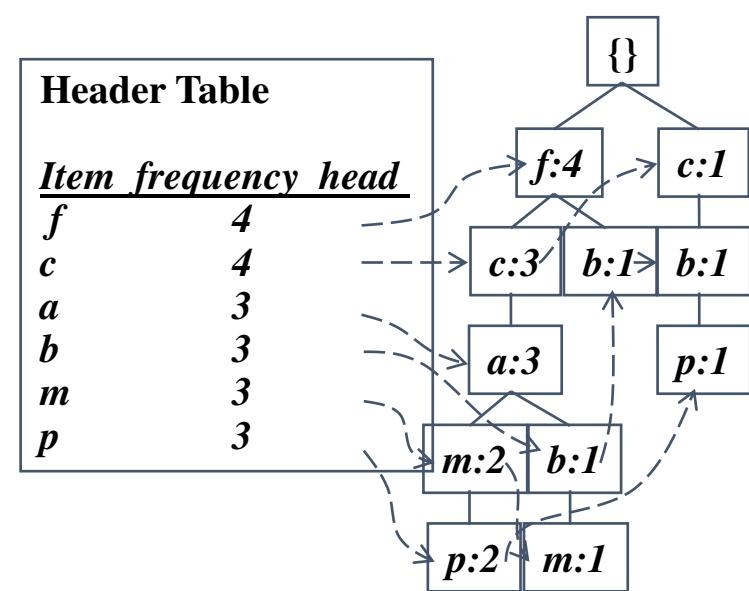
Construct FP-tree from a Transaction Database (Step 1)

min_support = 3

<i>TID</i>	<i>Items bought</i>	<i>(ordered) frequent items</i>
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

Item	Support Count
f	4
c	4
a	3
b	3
m	3
p	3

F-list = f-c-a-b-m-p



Partition Patterns and Databases

Frequent patterns can be partitioned into subsets according to f-list

- F-list=f-c-a-b-m-p
- Patterns containing p
- Patterns having m but no p
- ...
- Patterns having c but no a nor b, m, p
- Pattern f

Completeness and non-redundancy

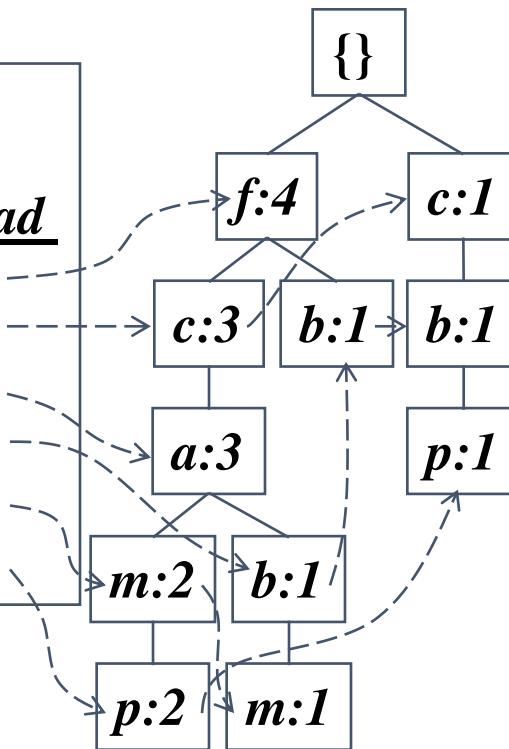
Find Patterns Having P From P-conditional Database (Step 2)

Starting at the frequent item header table in the FP-tree

Traverse the FP-tree by following the link of each frequent item p

Accumulate all of *transformed prefix paths* of item p to form p 's conditional pattern base

Header Table		
<u>Item frequency head</u>		
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



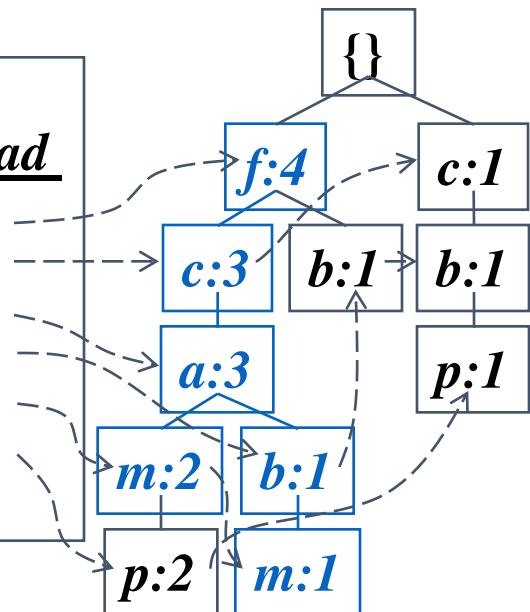
Conditional pattern base of p :
fcam:2, cb:1

From Conditional Pattern-bases to Conditional FP-trees

For each pattern-base

- Accumulate the count for each item in the base
- Construct the FP-tree for the frequent items of the pattern base

Header Table		
	<u>Item frequency</u>	<u>head</u>
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



m-conditional pattern base:
fca:2, fcab:1

All frequent patterns relate to *m*

{}	→	
		m,
f:3	→	fm, cm, am,
		fcm, fam, cam,
c:3	→	fcam
a:3		

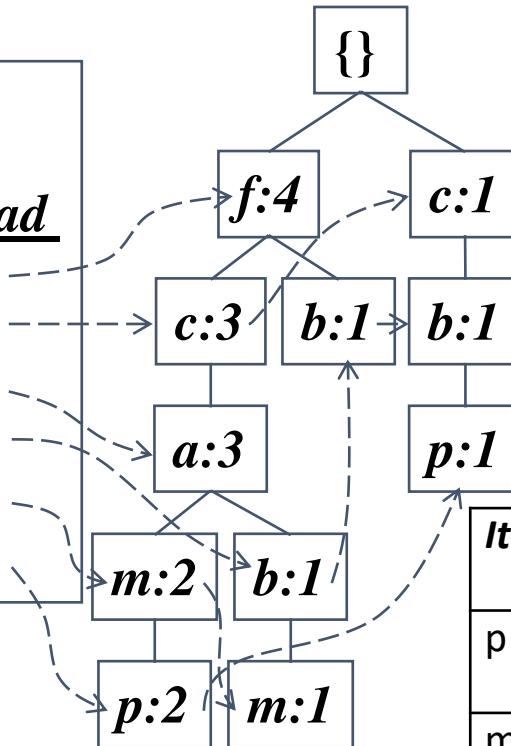
m-conditional FP-tree

Find Patterns Having x From x-conditional Database

Header Table

Item frequency head

f	4
c	4
a	3
b	3
m	3
p	3



Item	Conditional pattern base	Conditional fp-tree	Frequent patterns
p	fcam: 2, cb:1	c:3	cp:3
m	fca:2, fcab:1	fca:3	fcam:3 (and all its subsets)
b	fca:1, f:1, c:1	None	
a	fc:3	fc:3	
c	f:3	f:3	

Benefits of the FP-tree Structure

Completeness

- Preserve complete information for frequent pattern mining
- Never break a long pattern of any transaction

Compactness

- Reduce irrelevant info—infrequent items are gone
- Items in frequency descending order: the more frequently occurring, the more likely to be shared
- Never be larger than the original database (not count node-links and the *count* field)

Mining Frequent Patterns With FP-trees

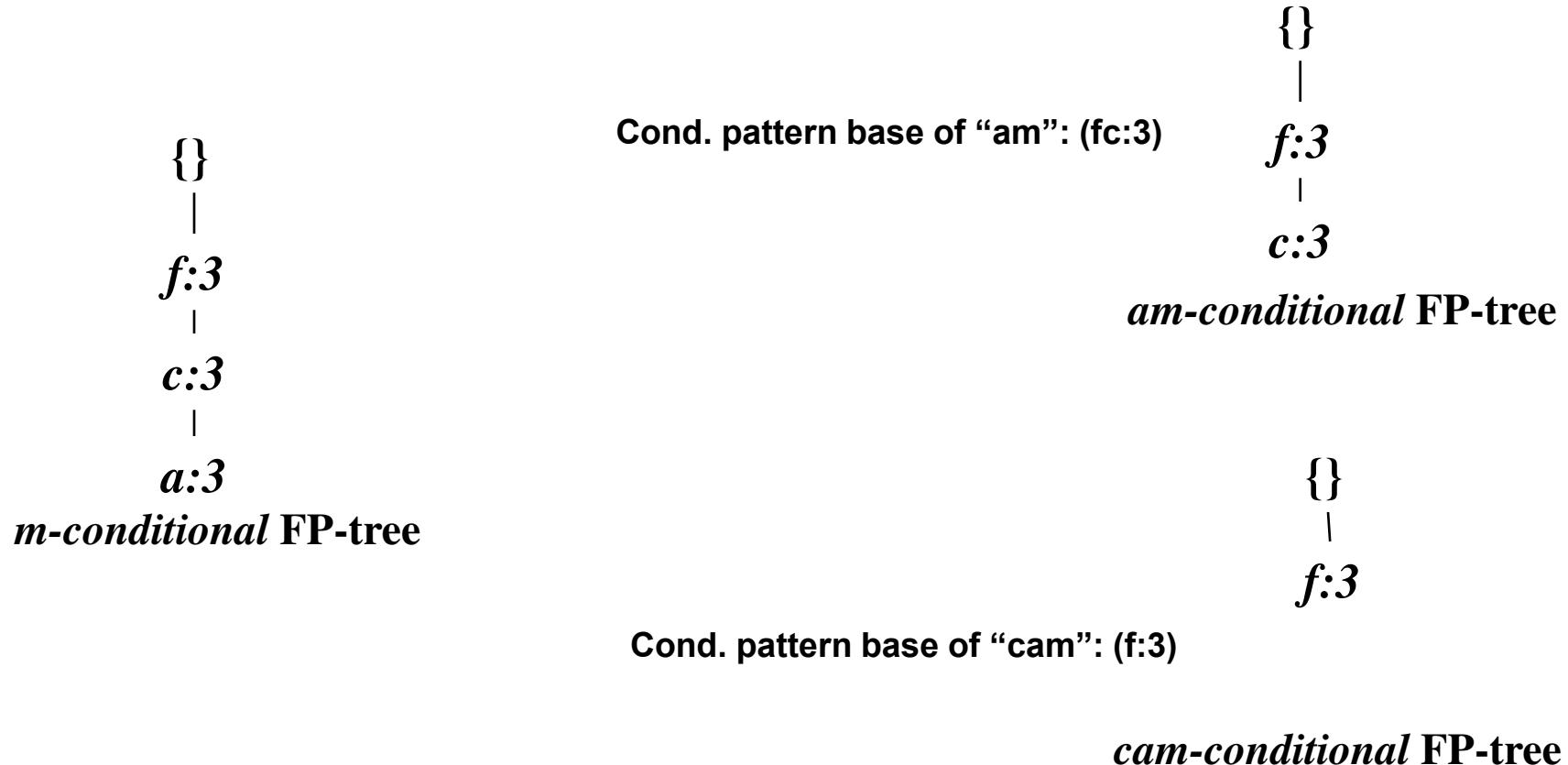
Idea: Frequent pattern growth

- Recursively grow frequent patterns by pattern and database partition

Method

- For each frequent item, construct its conditional pattern-base, and then its conditional FP-tree
- Repeat the process on each newly created conditional FP-tree
- Until the resulting FP-tree is empty, or it contains only one path—single path will generate all the combinations of its sub-paths, each of which is a frequent pattern

Recursion: Mining Each Conditional FP-tree



A Special Case: Single Prefix Path in FP-tree

Suppose a (conditional) FP-tree T has a shared single prefix-path P

Mining can be decomposed into two parts

$\{\}$

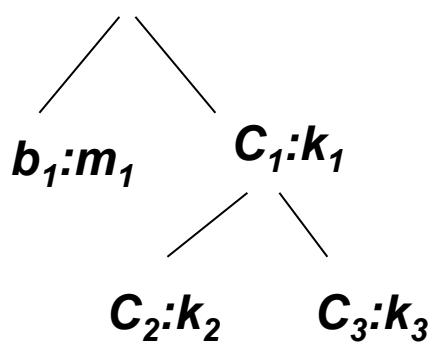
- Reduction of the single prefix path into one node

$a_1:n_1$

- Concatenation of the mining results of the two parts

$a_2:n_2$

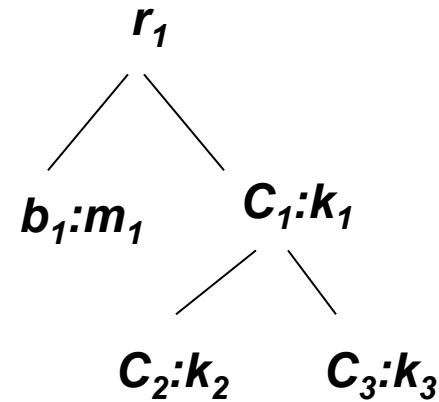
$a_3:n_3$



$r_1 =$

$\{\}$
 $a_1:n_1$
 $a_2:n_2$
 $a_3:n_3$

$+$



Why Is FP-Growth the Winner?

Divide-and-conquer:

- decompose both the mining task and DB according to the frequent patterns obtained so far
- leads to focused search of smaller databases

Other factors

- no candidate generation, no candidate test
- compressed database: FP-tree structure
- no repeated scan of entire database
- basic ops—counting local freq items and building sub FP-tree, no pattern search and matching



Mining association rules

Rule Generation

- Given a frequent itemset L , find all non-empty subsets $f \subset L$ such that $f \rightarrow L - f$ satisfies the minimum confidence requirement
 - If $\{A, B, C, D\}$ is a frequent itemset, candidate rules:

$ABC \rightarrow D, \quad ABD \rightarrow C, \quad ACD \rightarrow B, \quad BCD \rightarrow A,$
 $AB \rightarrow CD, \quad AC \rightarrow BD, \quad AD \rightarrow BC, \quad BC \rightarrow AD,$
 $BD \rightarrow AC, \quad CD \rightarrow AB,$
 $A \rightarrow BCD, \quad B \rightarrow ACD, \quad C \rightarrow ABD, \quad D \rightarrow ABC$

- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

Rule Generation

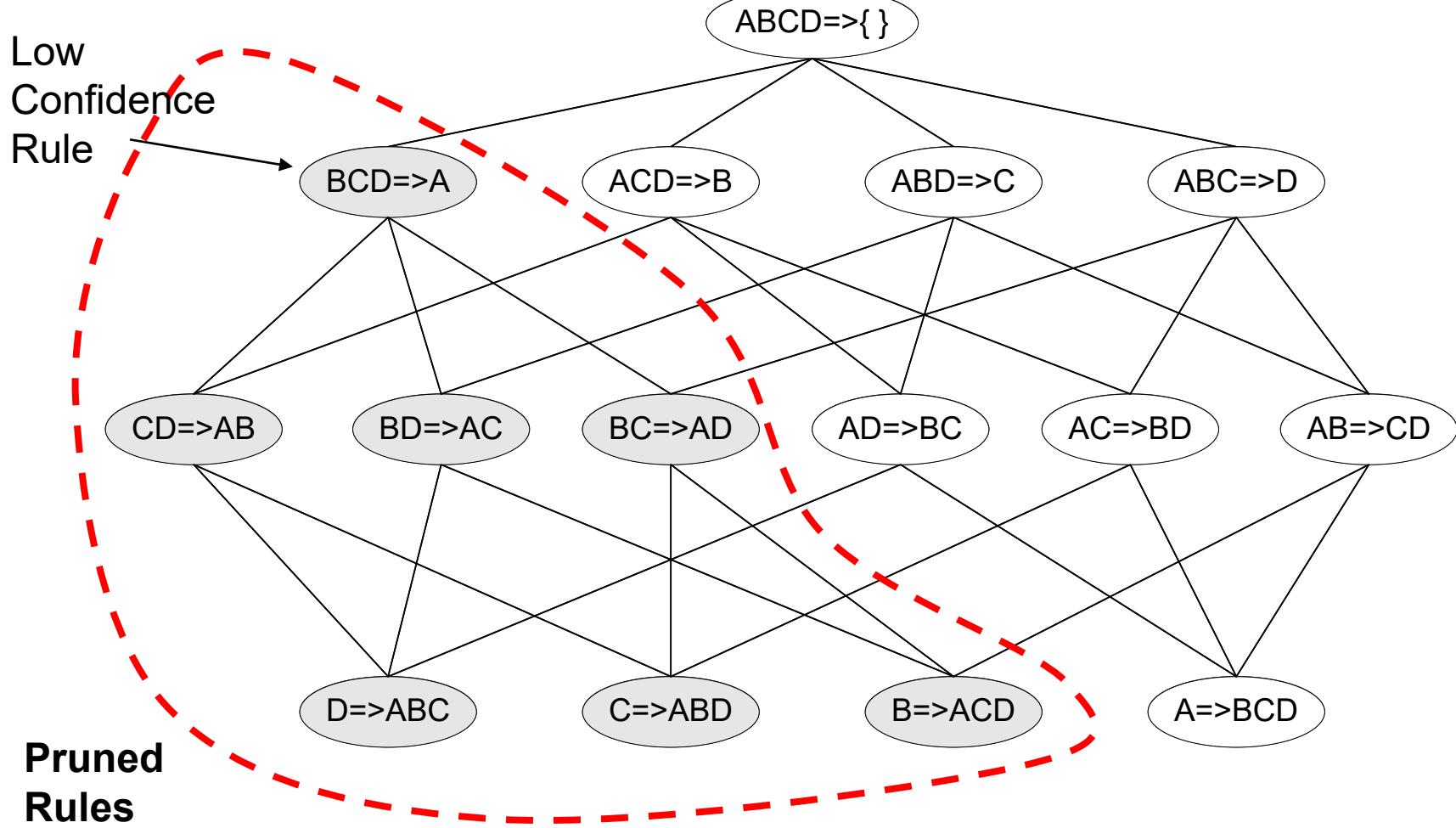
- How to efficiently generate rules from frequent itemsets?
 - In general, confidence does not have an anti-monotone property
 $c(ABC \rightarrow D)$ can be larger or smaller than $c(AB \rightarrow D)$
 - But confidence of rules generated from the same itemset has an anti-monotone property
 - e.g., $L = \{A, B, C, D\}$:

$$c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$$

- Confidence is anti-monotone in this case

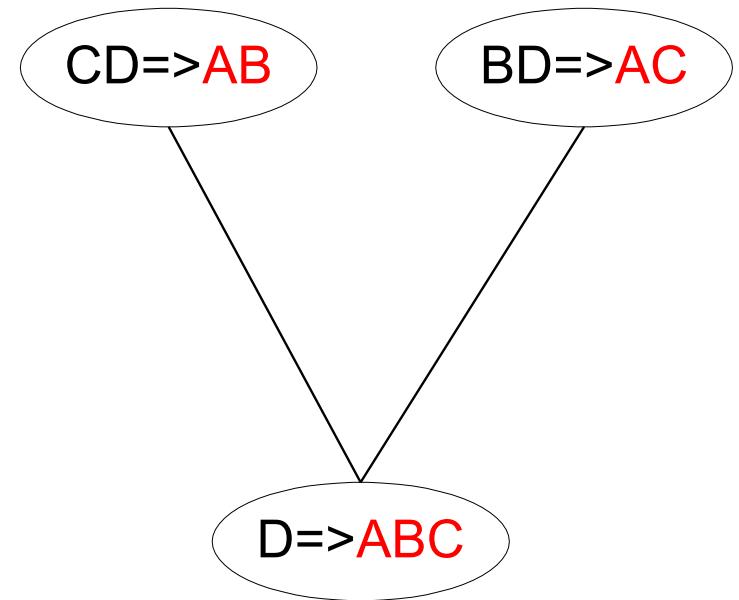
Rule Generation for Apriori Algorithm

Lattice of rules



Rule Generation with Apriori Algorithm

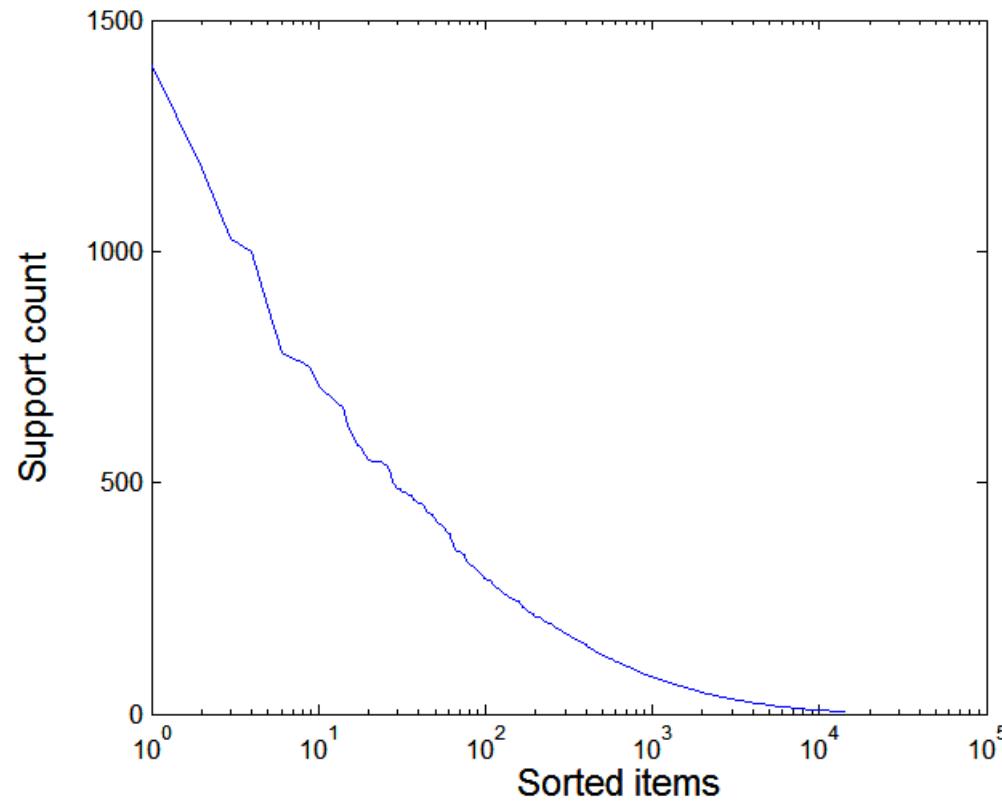
- Candidate rule is generated by merging two rules that share the same prefix in the rule consequent
- $\text{join}(\text{CD} \Rightarrow \text{AB}, \text{BD} \Rightarrow \text{AC})$
would produce the candidate rule $\text{D} \Rightarrow \text{ABC}$
- Prune rule $\text{D} \Rightarrow \text{ABC}$ if its subset $\text{AD} \Rightarrow \text{BC}$ does not have high confidence



Effect of Support Distribution

- Many real data sets have skewed support distribution

**Support
distribution of
a retail data set**



Effect of Support Distribution

- How to set the appropriate *minsup* threshold?
 - If *minsup* is set too high, we could miss itemsets involving interesting rare items (e.g., expensive products)
 - If *minsup* is set too low, it is computationally expensive and the number of itemsets is very large
- Using a single minimum support threshold may not be effective

Multiple Minimum Support

- How to apply multiple minimum supports?
 - $MS(i)$: minimum support for item i
 - e.g.: $MS(\text{Milk})=5\%$, $MS(\text{Coke}) = 3\%$,
 $MS(\text{Broccoli})=0.1\%$, $MS(\text{Salmon})=0.5\%$
 - $MS(\{\text{Milk, Broccoli}\}) = \min (MS(\text{Milk}), MS(\text{Broccoli}))$
 $= 0.1\%$
- Challenge: Support is no longer anti-monotone
 - Suppose: $\text{Support}(\text{Milk, Coke}) = 1.5\%$ and
 $\text{Support}(\text{Milk, Coke, Broccoli}) = 0.5\%$
 - $\{\text{Milk,Coke}\}$ is infrequent but $\{\text{Milk,Coke,Broccoli}\}$ is frequent

Pattern Evaluation

- Association rule algorithms tend to produce too many rules
 - many of them are uninteresting or redundant
 - Redundant if $\{A,B,C\} \rightarrow \{D\}$ and $\{A,B\} \rightarrow \{D\}$ have same support & confidence
- Interestingness measures can be used to prune/rank the derived patterns
- In the original formulation of association rules, support & confidence are the only measures used

Computing Interestingness Measure

- Given a rule $X \rightarrow Y$, information needed to compute rule interestingness can be obtained from a contingency table

Contingency table for $X \rightarrow Y$

	Y	\bar{Y}	
X	f_{11}	f_{10}	f_{1+}
\bar{X}	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	$ T $

f_{11} : support of X and Y

f_{10} : support of X and \bar{Y}

f_{01} : support of \bar{X} and Y

f_{00} : support of \bar{X} and \bar{Y}

Used to define various measures

- ◆ support, confidence, lift, Gini, J-measure, etc.

Drawback of Confidence

	Coffee	<u>Coffee</u>	
Tea	15	5	20
<u>Tea</u>	75	5	80
	90	10	100

Association Rule: Tea → Coffee

$$\text{Confidence} = P(\text{Coffee} | \text{Tea}) = 0.75$$

$$\text{but } P(\text{Coffee}) = 0.9$$

⇒ Although confidence is high, rule is misleading

$$\Rightarrow P(\text{Coffee} | \overline{\text{Tea}}) = 0.9375$$

Statistical Independence

- Population of 1000 students
 - 600 students know how to swim (S)
 - 700 students know how to bike (B)
 - 420 students know how to swim and bike (S,B)
- $P(S \wedge B) = 420/1000 = 0.42$
- $P(S) \times P(B) = 0.6 \times 0.7 = 0.42$
- $P(S \wedge B) = P(S) \times P(B) \Rightarrow$ Statistical independence
- $P(S \wedge B) > P(S) \times P(B) \Rightarrow$ Positively correlated
- $P(S \wedge B) < P(S) \times P(B) \Rightarrow$ Negatively correlated

Statistical-based Measures

- Measures that take into account statistical dependence

$$Lift = \frac{P(Y | X)}{P(Y)}$$

$$Interest = \frac{P(X, Y)}{P(X)P(Y)}$$

Example: Lift/Interest

	Coffee	<hr/> Coffee	
Tea	15	5	20
<hr/> Tea	75	5	80
	90	10	100

Association Rule: Tea → Coffee

$$\text{Confidence} = P(\text{Coffee} | \text{Tea}) = 0.75$$

$$\text{but } P(\text{Coffee}) = 0.9$$

$$\Rightarrow \text{Lift} = 0.75/0.9 = 0.8333 (< 1, \text{ therefore is negatively associated})$$

Drawback of Lift & Interest

	Y	\bar{Y}	
X	10	0	10
\bar{X}	0	90	90
	10	90	100

	Y	\bar{Y}	
X	90	0	90
\bar{X}	0	10	10
	90	10	100

$$Lift = \frac{0.1}{(0.1)(0.1)} = 10$$

$$Lift = \frac{0.9}{(0.9)(0.9)} = 1.11$$

Statistical independence:

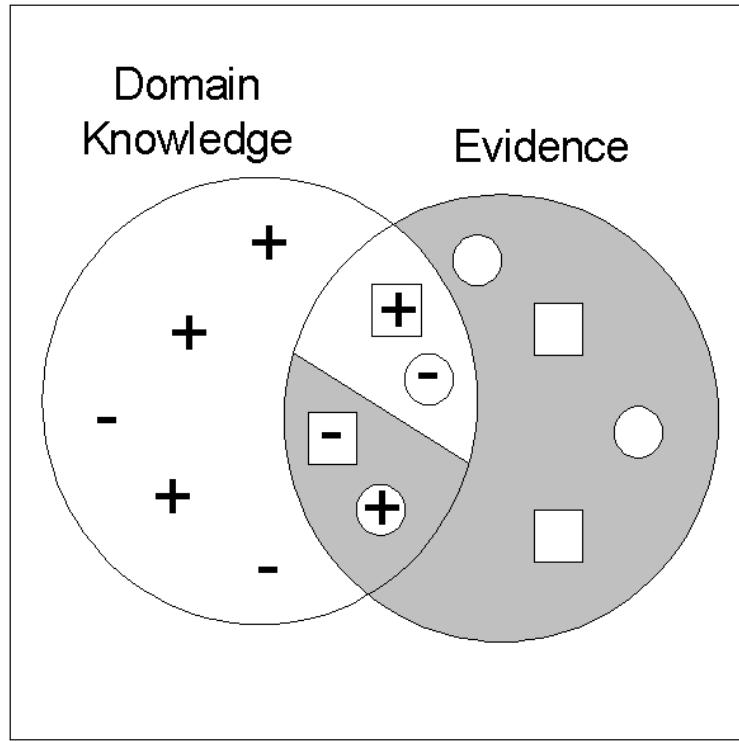
If $P(X,Y) = P(X)P(Y)$ \Rightarrow Lift = 1

Subjective Interestingness Measure

- Objective measure:
 - Rank patterns based on statistics computed from data
 - e.g., many measures of association (support, confidence, Laplace, Gini, mutual information, Jaccard, etc).
- Subjective measure:
 - Rank patterns according to user's interpretation
 - A pattern is subjectively interesting if it contradicts the expectation of a user
 - A pattern is subjectively interesting if it is actionable

Interestingness via Unexpectedness

- Need to model expectation of users (domain knowledge)



- + Pattern expected to be frequent
- Pattern expected to be infrequent
- Pattern found to be frequent
- Pattern found to be infrequent
- + Expected Patterns
- Unexpected Patterns

- Need to combine expectation of users with evidence from data (i.e., extracted patterns)

Prescribed Text Books

	Author(s), Title, Edition, Publishing House
T1	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
R1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
R2	Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner by Vijay Kotu and Bala Deshpande Morgan Kaufmann Publishers

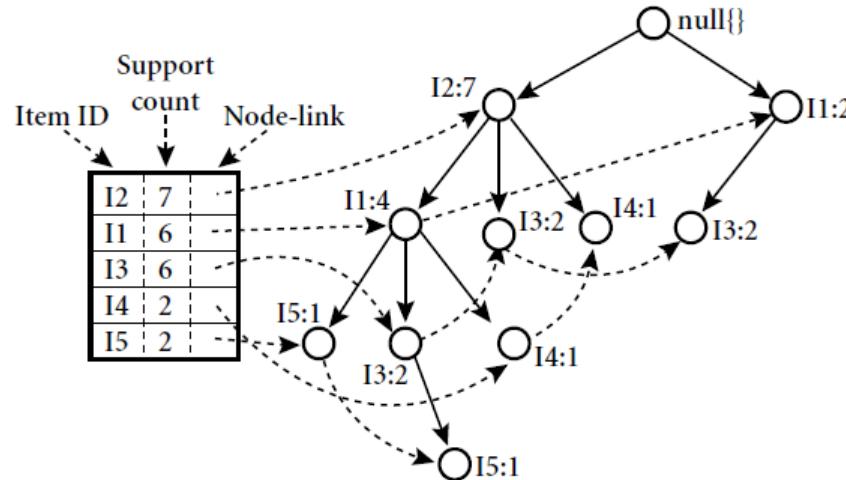
Example #2: Creation of FP-tree

TID	List of item IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Minimum support count = 2

The first scan of the database is the same as Apriori, which derives the set of frequent items (1-itemsets) and their support counts (frequencies). Let the minimum support count be 2. The set of frequent items is sorted in the order of descending support count. This resulting set or *list* is denoted L . Thus, we have $L = \{\{I2: 7\}, \{I1: 6\}, \{I3: 6\}, \{I4: 2\}, \{I5: 2\}\}$.

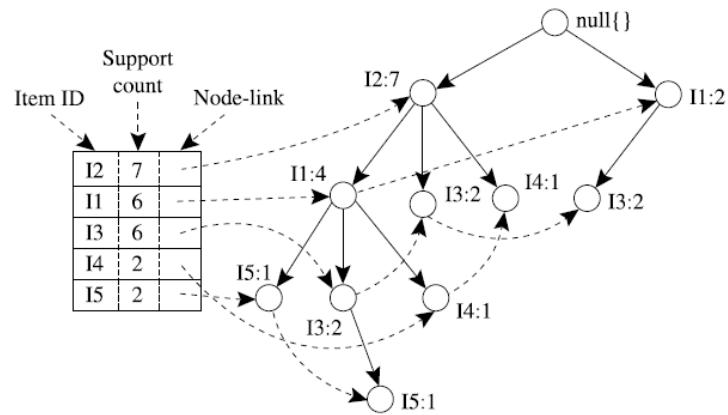
(I2, I1, I5 in L order)



FP-Growth and identifying frequent patterns

Mining the FP-Tree by Creating Conditional (Sub-)Pattern Bases

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
I5	{I2, I1: 1}, {I2, I1, I3: 1}	(I2: 2, I1: 2)	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{I2, I1: 1}, {I2: 1}	(I2: 2)	{I2, I4: 2}
I3	{I2, I1: 2}, {I2: 2}, {I1: 2}	(I2: 4, I1: 2), (I1: 2)	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{I2: 4}	(I2: 4)	{I2, I1: 4}





BITS Pilani

Pilani | Dubai | Goa | Hyderabad

S2-21_DSECLZC415: Data Mining (Lecture #10 – Cluster Analysis)



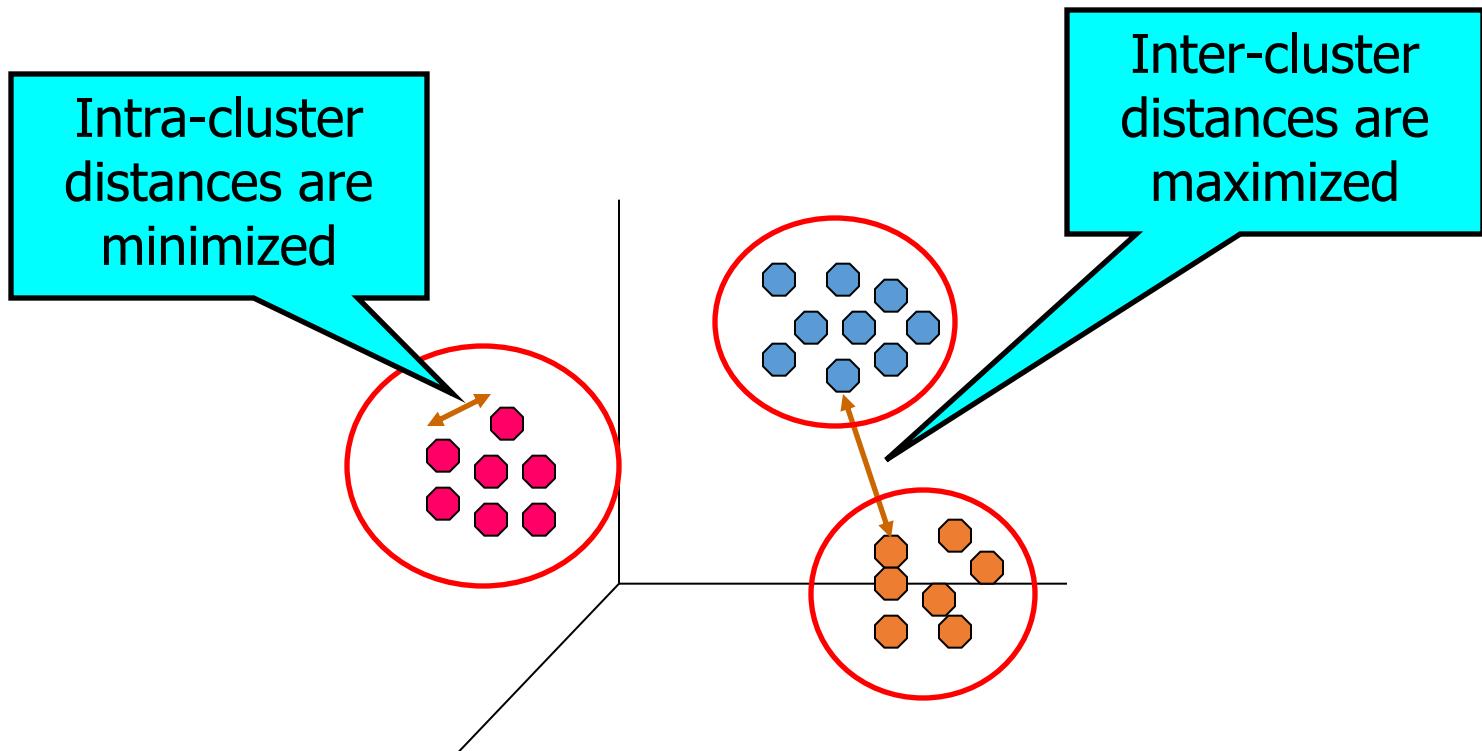
- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*



Clustering Concepts

What is Cluster Analysis?

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



Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earthquake studies: Observed earth quake epicenters should be clustered along continent faults

Applications of Cluster Analysis

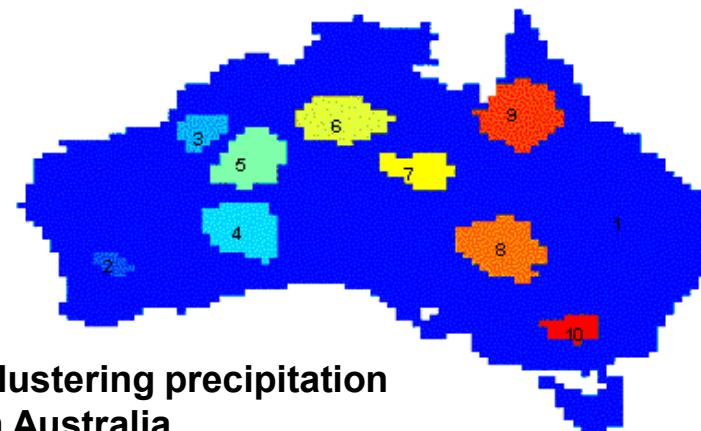
- **Understanding**

- **Group related documents for browsing, group genes and proteins that have similar functionality, or group stocks with similar price fluctuations**

	<i>Discovered Clusters</i>	<i>Industry Group</i>
1	Applied-Matl-DOWN,Bay-Network-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,Oracl-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-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, 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

- **Summarization**

- **Reduce the size of large data sets**



What is not Cluster Analysis?

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

Quality: What Is Good Clustering?

- A good clustering method will produce high quality clusters with
 - high intra-class similarity (cohesive within clusters)
 - low inter-class similarity (distinctive between clusters)
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns

Measure the Quality of Clustering

- **Dissimilarity/Similarity metric:** Similarity is expressed in terms of a distance function, typically metric: $d(i, j)$
- There is a separate “quality” function that measures the “goodness” of a cluster.
- The definitions of **distance functions** are usually very different for interval-scaled, boolean, categorical, ordinal ratio, and vector variables.
- Weights should be associated with different variables based on applications and data semantics.
- It is hard to define “similar enough” or “good enough”
 - the answer is typically highly subjective.

Requirements of Clustering in Data Mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Able to deal with noise and outliers
- Insensitive to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Type of data in clustering analysis

- Interval-scaled variables
- Binary variables
- Nominal, ordinal, and ratio variables
- Variables of mixed types

Interval-valued variables(standardization)

Review

Z-score:

$$z = \frac{x - \mu}{\sigma}$$

X: raw score to be standardized,

μ : mean of the population, σ : standard deviation

- the distance between the raw score and the population mean in units of the standard deviation
- negative when the raw score is below the mean, “+” when above
- Alternately
 - Calculate the mean absolute deviation:

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|)$$

where $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \dots + x_{nf})$.

- Calculate the standardized measurement (z-score)

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

- Using mean absolute deviation is more robust than using standard deviation

Similarity and Dissimilarity Between Objects

Review

- Distances are normally used to measure the similarity or dissimilarity between two data objects
- Some popular ones include: *Minkowski distance*:

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q)}$$

where $i = (x_{i1}, x_{i2}, \dots, x_{ip})$ and $j = (x_{j1}, x_{j2}, \dots, x_{jp})$ are two p -dimensional data objects, and q is a positive integer

- If $q = 1$, d is Manhattan distance

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

Similarity and Dissimilarity Between Objects (Cont.)

Review

- If $q = 2$, d is Euclidean distance:

$$d(i, j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \dots + |x_{i_p} - x_{j_p}|^2)}$$

- Properties
 - $d(i, j) \geq 0$
 - $d(i, i) = 0$
 - $d(i, j) = d(j, i)$
 - $d(i, j) \leq d(i, k) + d(k, j)$
- Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures

Binary Variables

- A contingency table for binary data
- Distance measure for symmetric binary variables:
- Distance measure for asymmetric binary variables:
- Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

		Object <i>j</i>		
		1	0	<i>sum</i>
Object <i>i</i>	1	<i>a</i>	<i>b</i>	<i>a+b</i>
	0	<i>c</i>	<i>d</i>	<i>c+d</i>
	<i>sum</i>	<i>a+c</i>	<i>b+d</i>	<i>p</i>

$$d(i, j) = \frac{b+c}{a+b+c+d}$$

$$d(i, j) = \frac{b+c}{a+b+c}$$

$$sim_{Jaccard}(i, j) = \frac{a}{a+b+c}$$

Nominal Variables

Review

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching
 - m : # of matches, p : total # of variables

$$d(i, j) = \frac{p - m}{p}$$

- Method 2: use a large number of binary variables
 - creating a new binary variable for each of the M nominal states

Review

Ordinal Variables

- An ordinal variable can be discrete or continuous
- Order is important, e.g., rank
- Can be treated like interval-scaled
 - replace x_{if} by their rank $r_{if} \in \{1, \dots, M_f\}$
 - map the range of each variable onto $[0, 1]$ by replacing i -th object in the f -th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- compute the dissimilarity using methods for interval-scaled variables

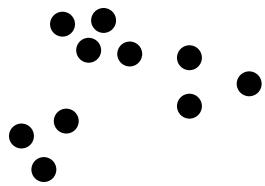
Review

Attributes of Mixed Type

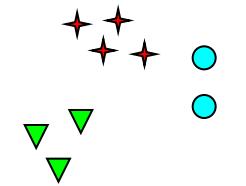
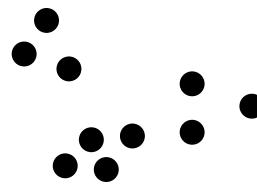
- A database may contain all attribute types
 - Nominal, symmetric binary, asymmetric binary, numeric, ordinal
- One may use a weighted formula to combine their effects

$$d(i, j) = \frac{\sum_{f=1}^p \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^p \delta_{ij}^{(f)}}$$

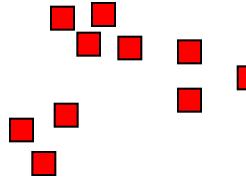
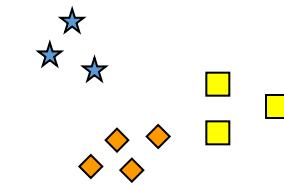
Notion of a Cluster can be Ambiguous



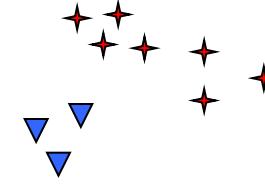
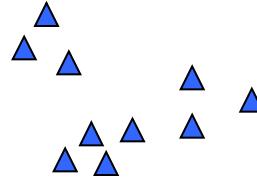
How many clusters?



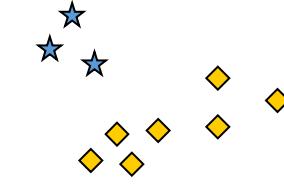
Six Clusters



Two Clusters



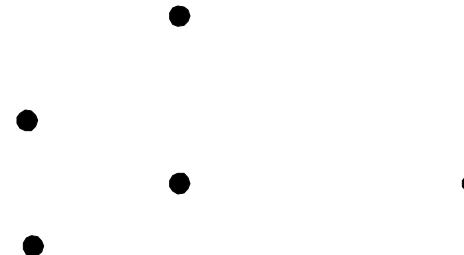
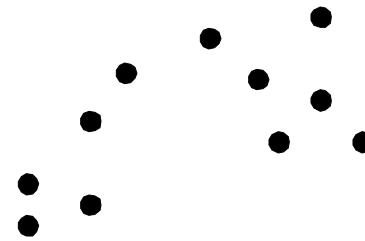
Four Clusters



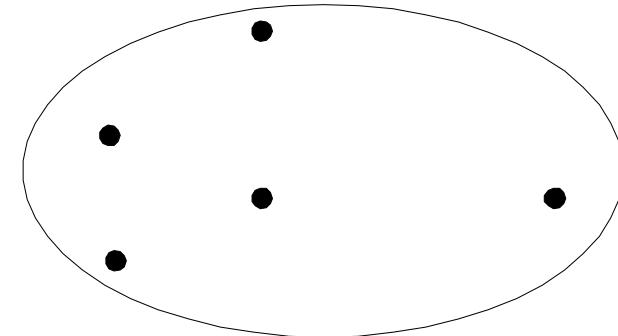
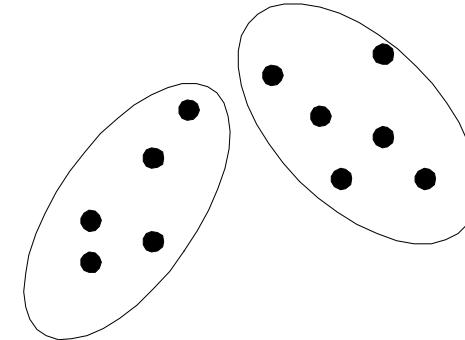
Types of Clusterings

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

Partitional Clustering

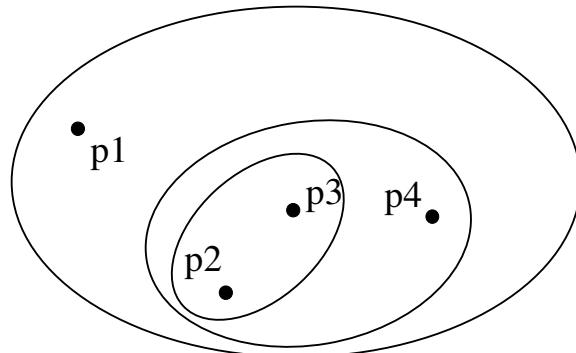


Original Points

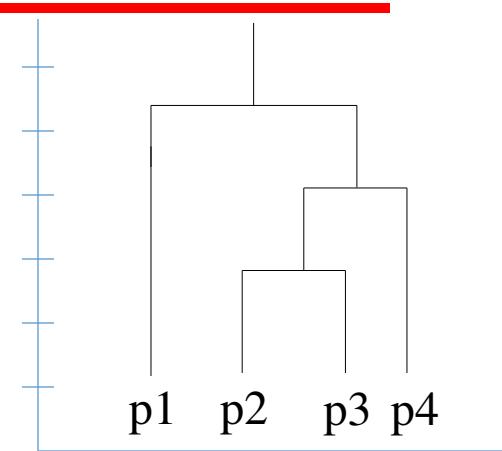


A Partitional Clustering

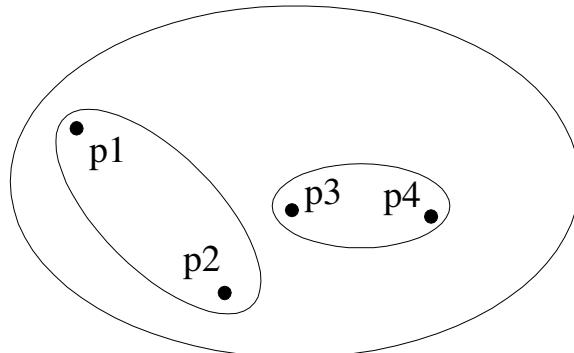
Hierarchical Clustering



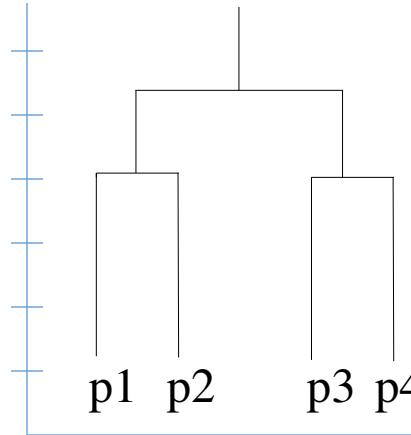
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
 - In non-exclusive clustering, points may belong to multiple clusters.
 - Can represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
 - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
 - Weights must sum to 1
 - Probabilistic clustering has similar characteristics
- Partial versus complete
 - In some cases, we only want to cluster some of the data
- Heterogeneous versus homogeneous
 - Cluster of widely different sizes, shapes, and densities

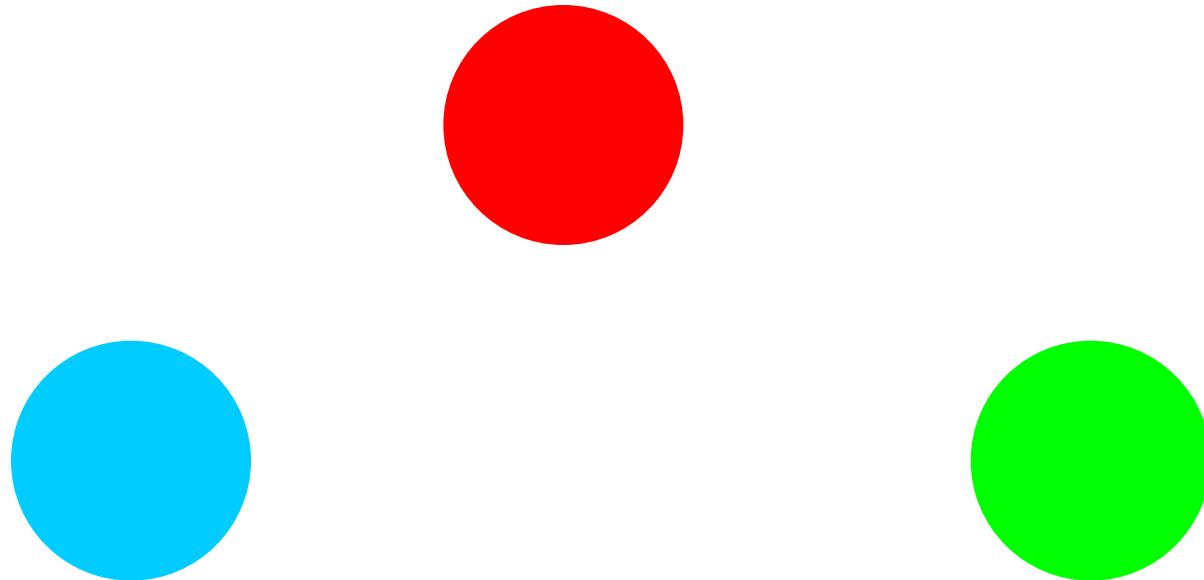
Types of Clusters

Clusters can be of many types:

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

Types of Clusters: Well-Separated

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



3 well-separated clusters

Types of Clusters: Center-Based

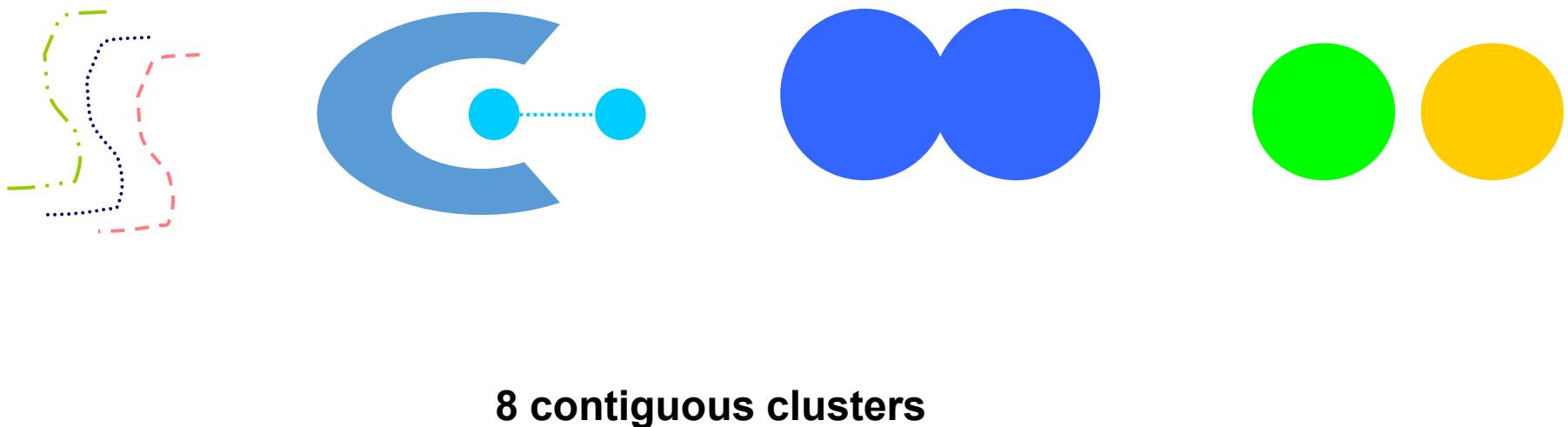
- Center-based
 - A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
 - The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**, the most “representative” point of a cluster



4 center-based clusters

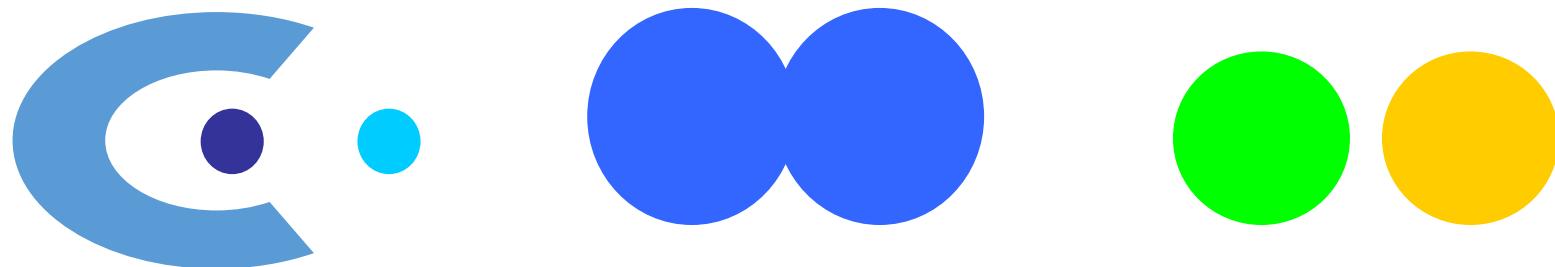
Types of Clusters: Contiguity-Based

- Contiguous Cluster (Nearest neighbor or Transitive)
 - A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.



Types of Clusters: Density-Based

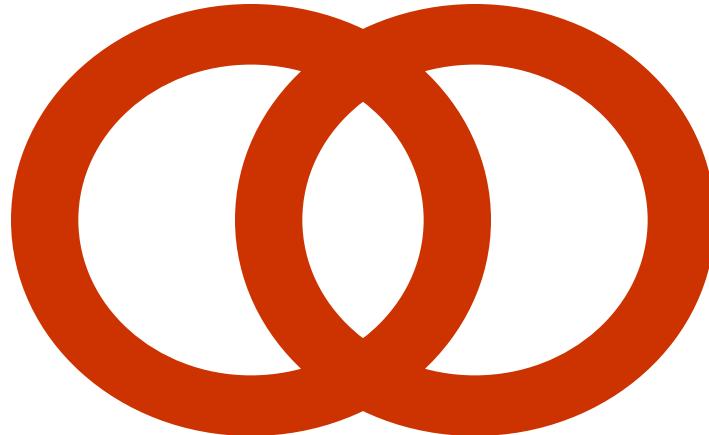
- Density-based
 - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
 - Used when the clusters are irregular or intertwined, and when noise and outliers are present.



6 density-based clusters

Types of Clusters: Conceptual Clusters

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



2 Overlapping Circles

Types of Clusters: Objective Function

- Clusters Defined by an Objective Function
 - Finds clusters that minimize or maximize an objective function.
 - Enumerate all possible ways of dividing the points into clusters and evaluate the 'goodness' of each potential set of clusters by using the given objective function. (NP Hard)
 - Can have global or local objectives.
 - Hierarchical clustering algorithms typically have local objectives
 - Partitional algorithms typically have global objectives
 - A variation of the global objective function approach is to fit the data to a parameterized model.
 - Parameters for the model are determined from the data.
 - Mixture models assume that the data is a 'mixture' of a number of statistical distributions.

Types of Clusters: Objective Function ...

- Map the clustering problem to a different domain and solve a related problem in that domain
 - Proximity matrix defines a weighted graph, where the nodes are the points being clustered, and the weighted edges represent the proximities between points
 - Clustering is equivalent to breaking the graph into connected components, one for each cluster.
 - Want to minimize the edge weight between clusters and maximize the edge weight within clusters

Important Characteristics of the Input Data

- Type of proximity or density measure
 - This is a derived measure, but central to clustering
 - Sparseness
 - Dictates type of similarity
 - Adds to efficiency
 - Attribute type
 - Dictates type of similarity
 - Type of Data
 - Dictates type of similarity
 - Other characteristics, e.g., autocorrelation
 - Dimensionality
 - Noise and Outliers
 - Type of Distribution
-



Partitioning Methods

Partitioning Algorithms: Basic Concept

- Partitioning method: Partitioning a database D of n objects into a set of k clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

$$E = \sum_{i=1}^k \sum_{p \in C_i} (p - c_i)^2$$

- Given k , find a partition of k *clusters* that optimizes the chosen partitioning criterion
 - Global optimal: exhaustively enumerate all partitions
 - Heuristic methods: *k-means* and *k-medoids* algorithms
 - *k-means* : Each cluster is represented by the center of the cluster
 - *k-medoids* or PAM (Partition around medoids): Each cluster is represented by one of the objects in the cluster

K-means Clustering

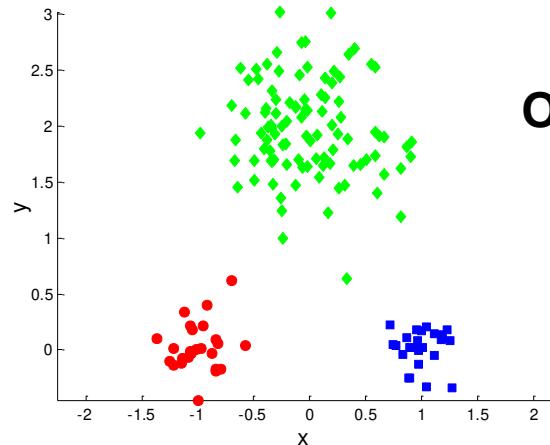
- Partitional clustering approach
- Each cluster is associated with a *centroid* (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K , must be specified
- The basic algorithm is very simple

-
- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-

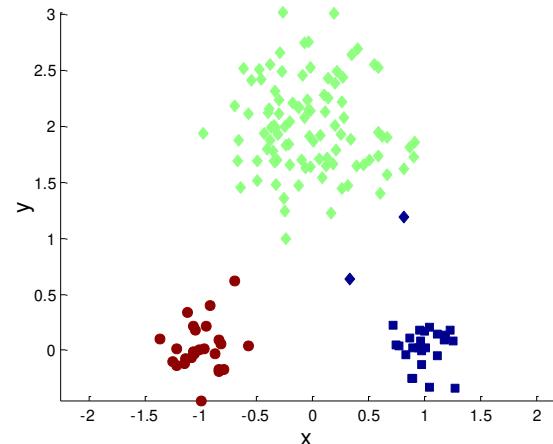
K-means Clustering – Details

- Initial centroids are often chosen randomly.
 - Clusters produced vary from one run to another.
- The centroid is (typically) the mean of the points in the cluster.
- ‘Closeness’ is measured by Euclidean distance, cosine similarity, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
 - Often the stopping condition is changed to ‘Until relatively few points change clusters’
- Complexity is $O(n * K * I * d)$
 - n = number of points, K = number of clusters,
 I = number of iterations, d = number of attributes

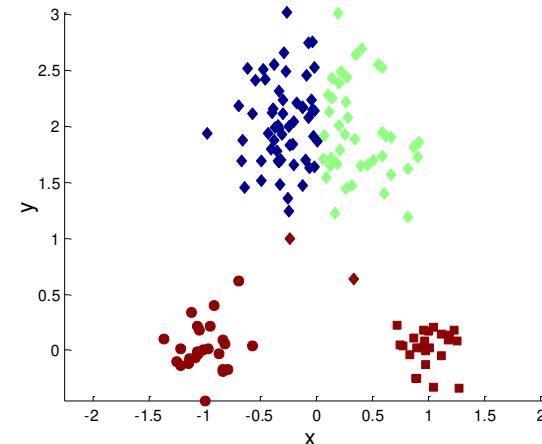
Evaluating K-means Clusters



Original Points



Optimal Clustering



Sub-optimal Clustering

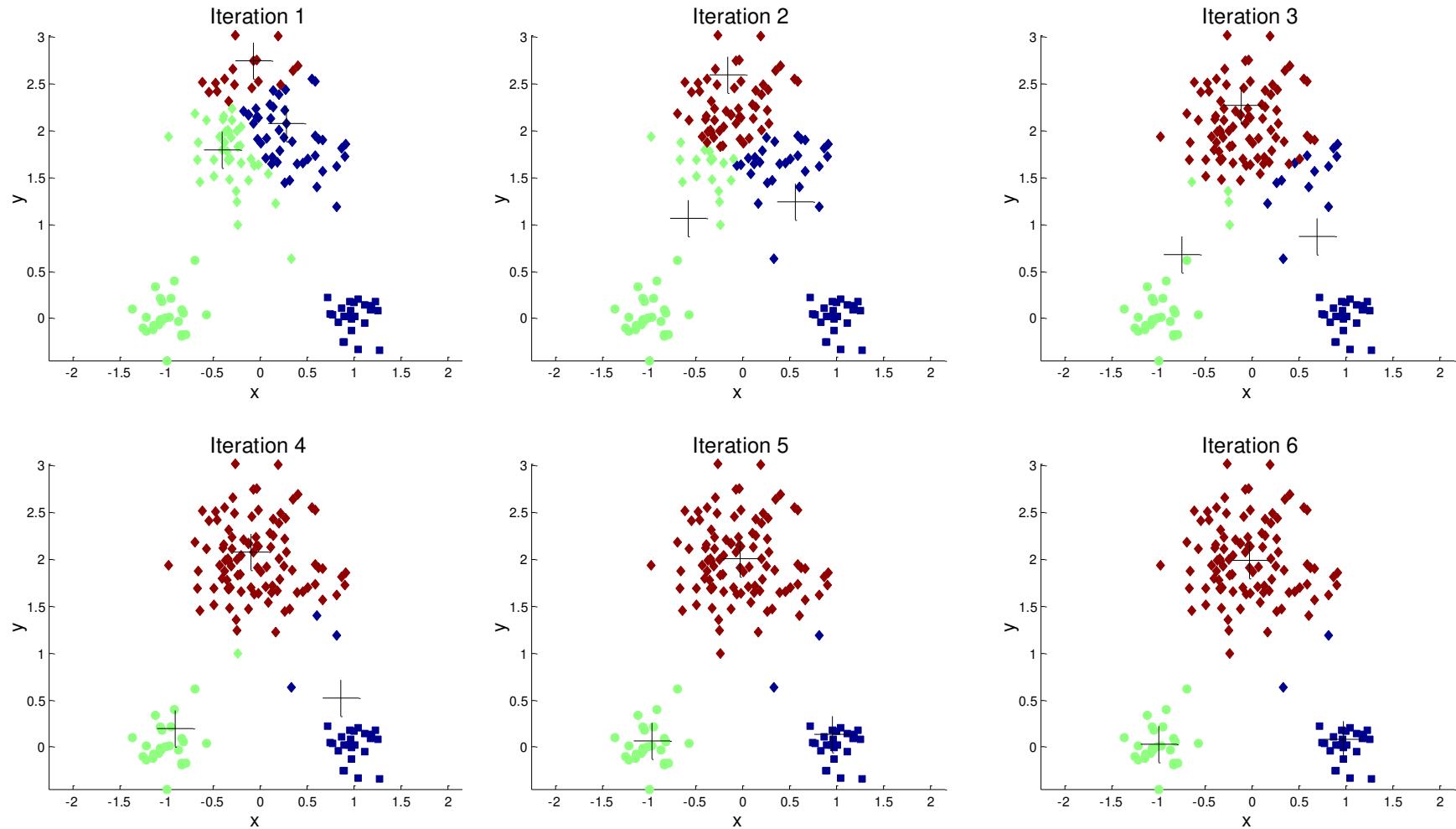
Evaluating K-means Clusters

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

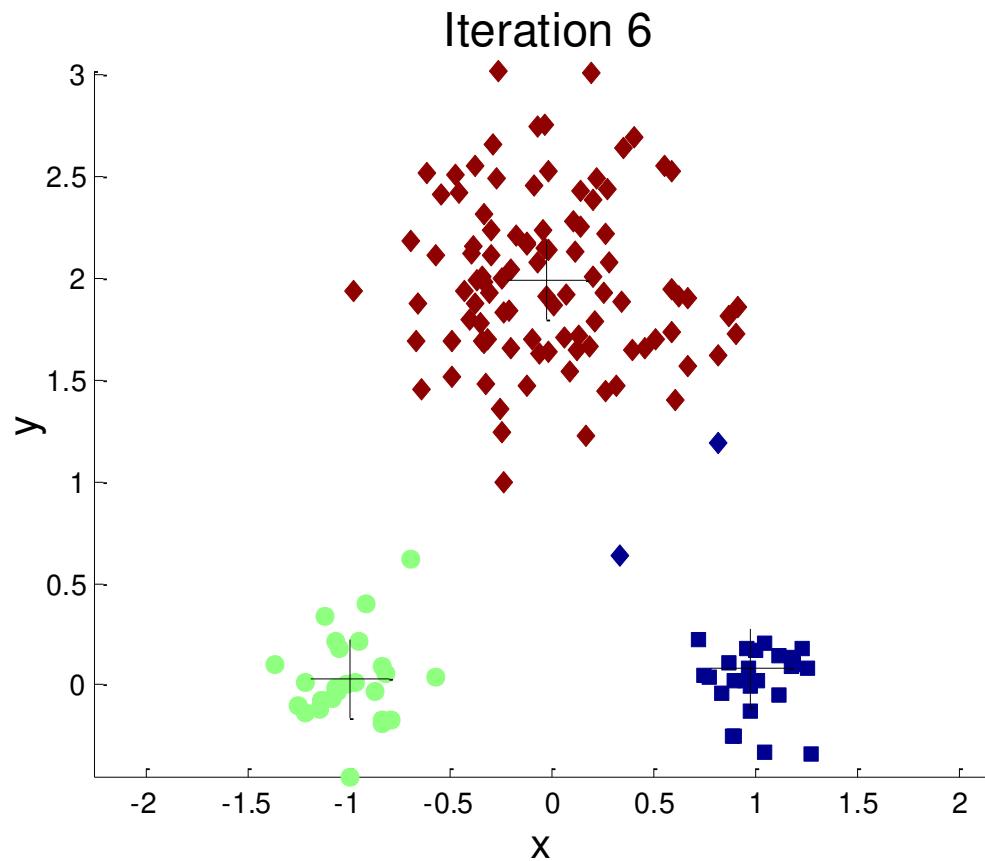
$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x)$$

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

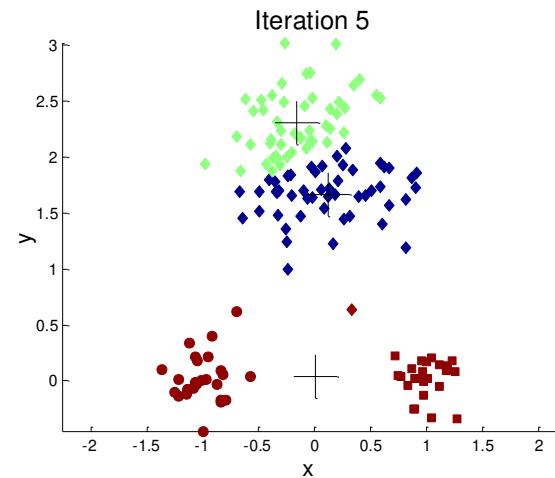
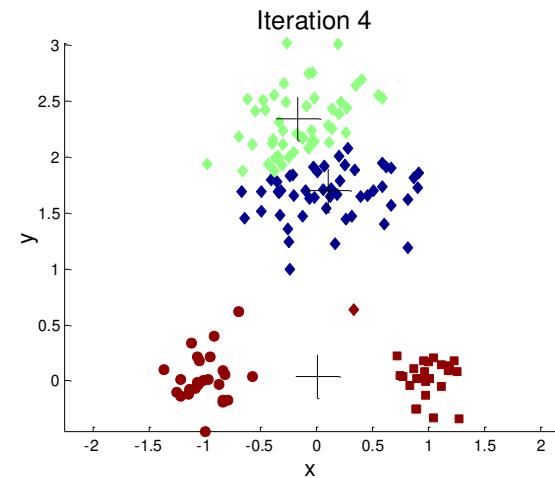
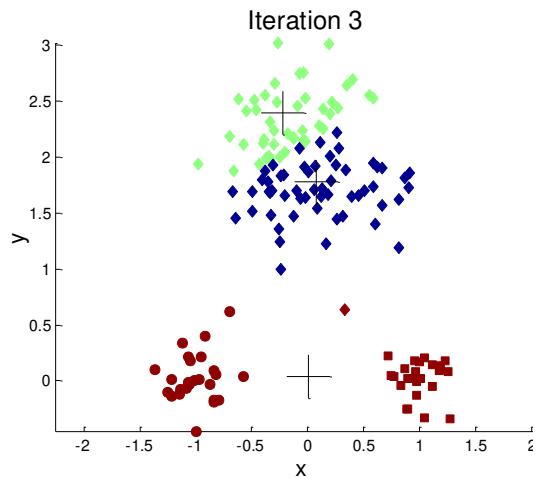
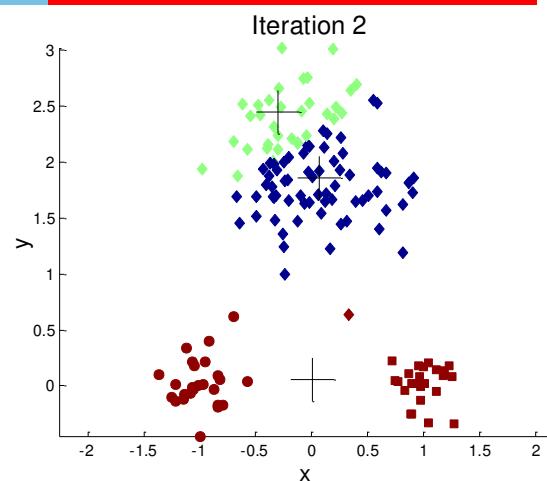
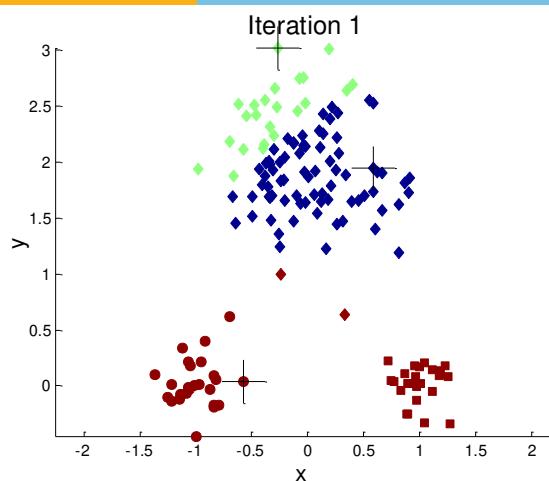
Importance of Choosing Initial Centroids



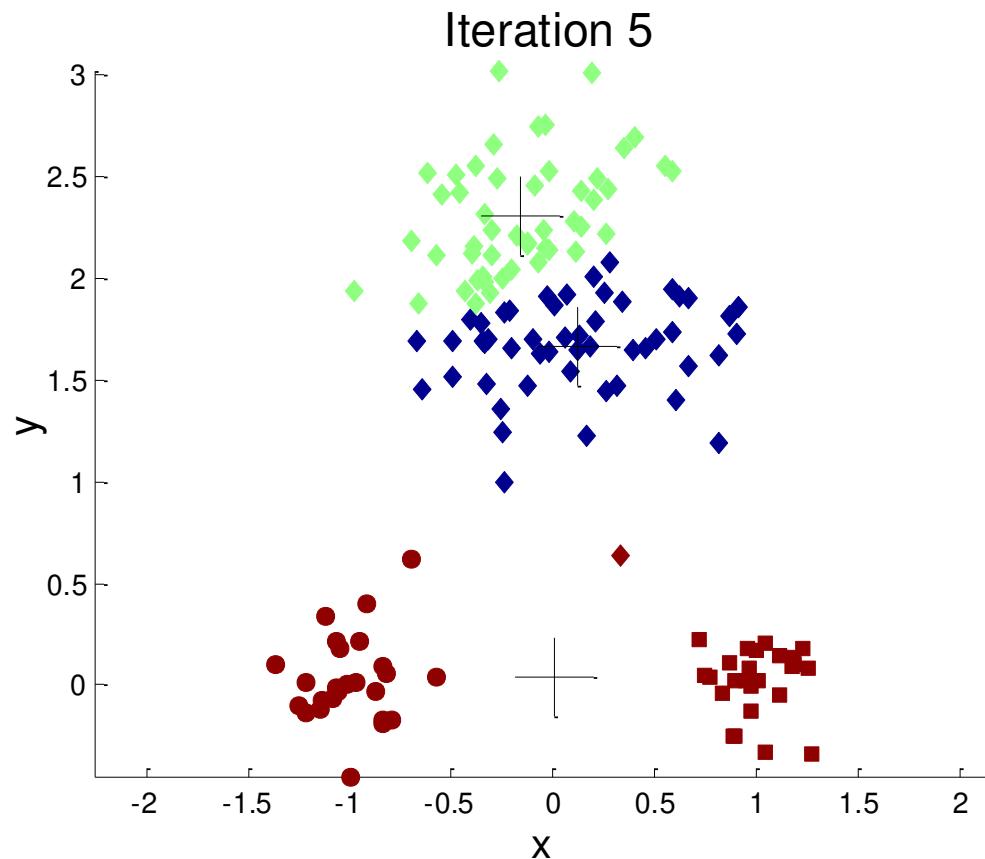
Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids ...



Importance of Choosing Initial Centroids ...



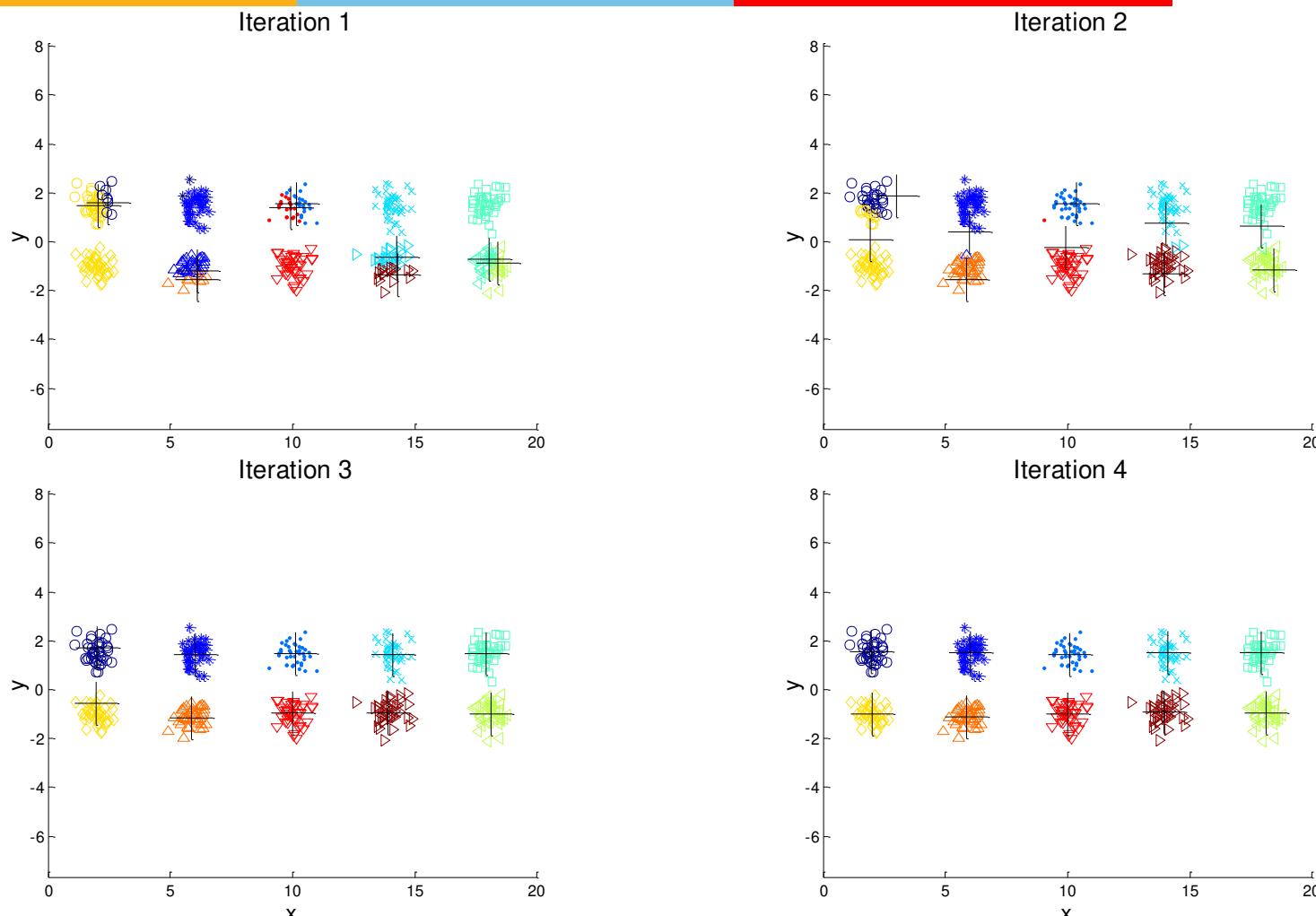
Problems with Selecting Initial Points

- If there are K ‘real’ clusters then the chance of selecting one centroid from each cluster is small.
 - Chance is relatively small when K is large
 - If clusters are the same size, n , then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

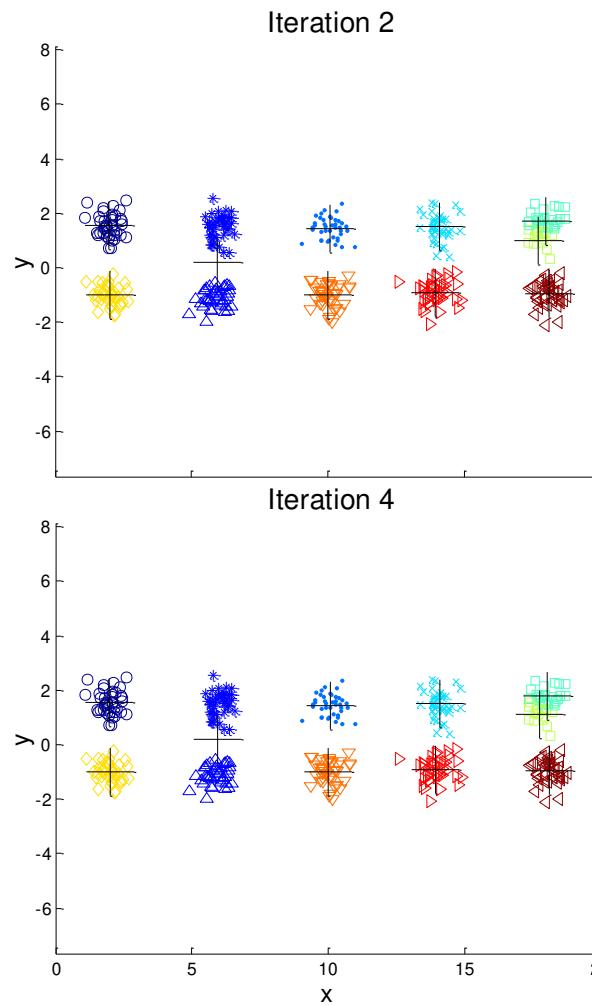
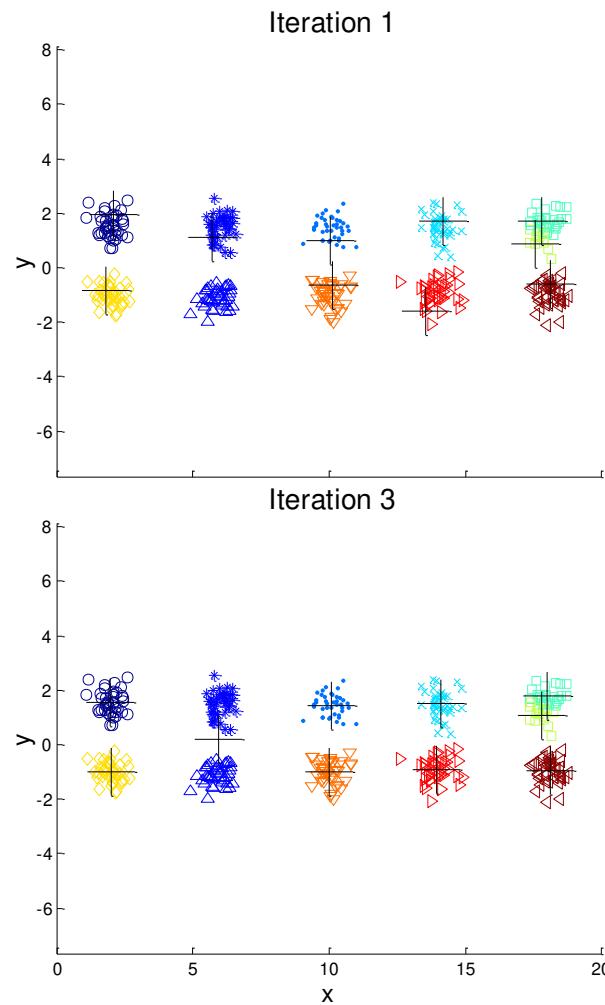
- For example, if $K = 10$, then probability = $10!/10^{10} = 0.00036$
- Sometimes the initial centroids will readjust themselves in ‘right’ way, and sometimes they don’t
- Consider an example of five pairs of clusters

10 Clusters Example



Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example



Starting with some pairs of clusters having three initial centroids, while other have only one.

Solutions to Initial Centroids Problem

- Multiple runs
 - Helps, but probability is not favorable
- Sample and use hierarchical clustering to determine initial centroids
- Select more than k initial centroids and then select among these initial centroids
 - Select most widely separated
- Postprocessing

Pre-processing and Post-processing

- Pre-processing
 - Normalize the data
 - Eliminate outliers
- Post-processing
 - Eliminate small clusters that may represent outliers
 - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
 - Merge clusters that are ‘close’ and that have relatively low SSE

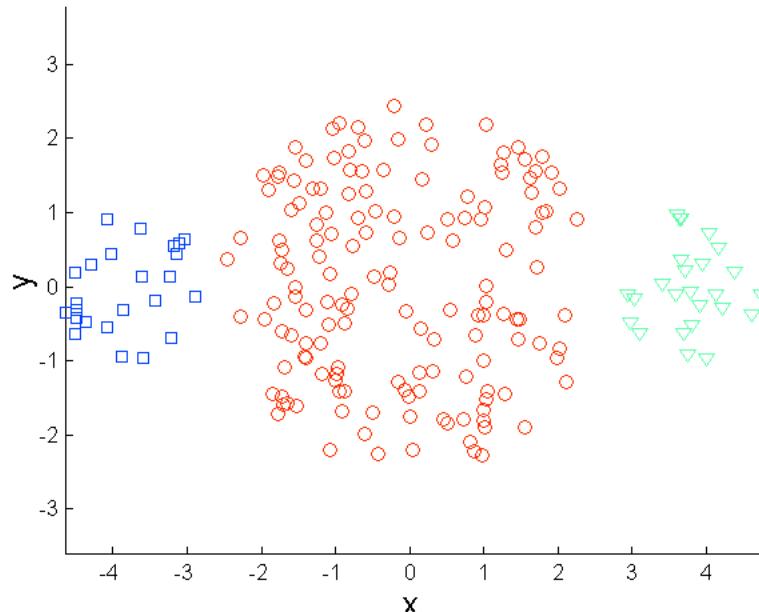
Variations of the *K-Means* Method

- Most of the variants of the *k-means* which differ in
 - Selection of the initial *k* means
 - Dissimilarity calculations
 - Strategies to calculate cluster means
- Handling categorical data: *k-modes*
 - Replacing means of clusters with modes
 - Using new dissimilarity measures to deal with categorical objects
 - Using a frequency-based method to update modes of clusters

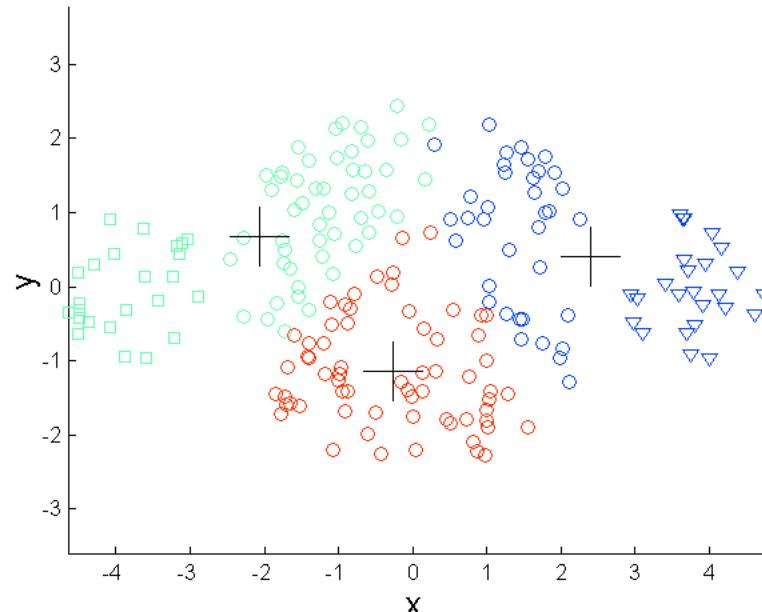
Limitations of K-means

- K-means has problems when clusters are of differing
 - Sizes
 - Densities
 - Non-globular shapes
- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

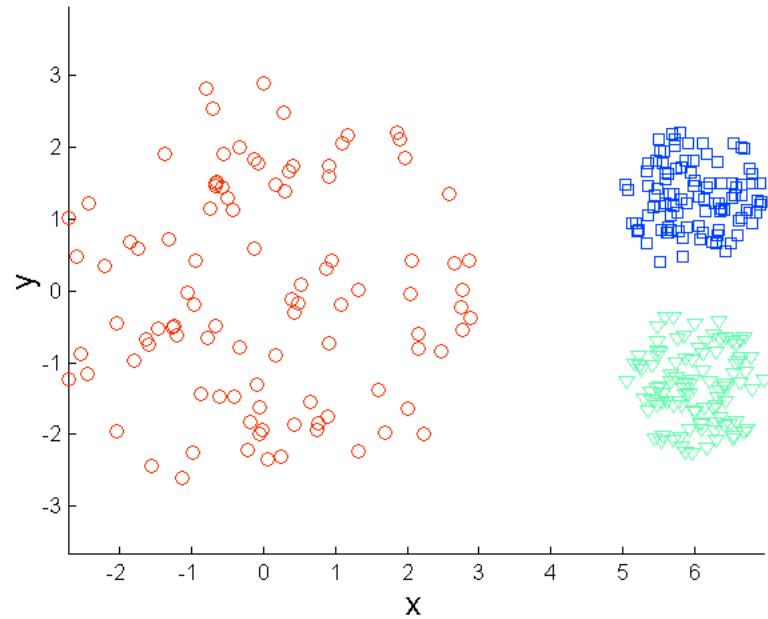


Original Points

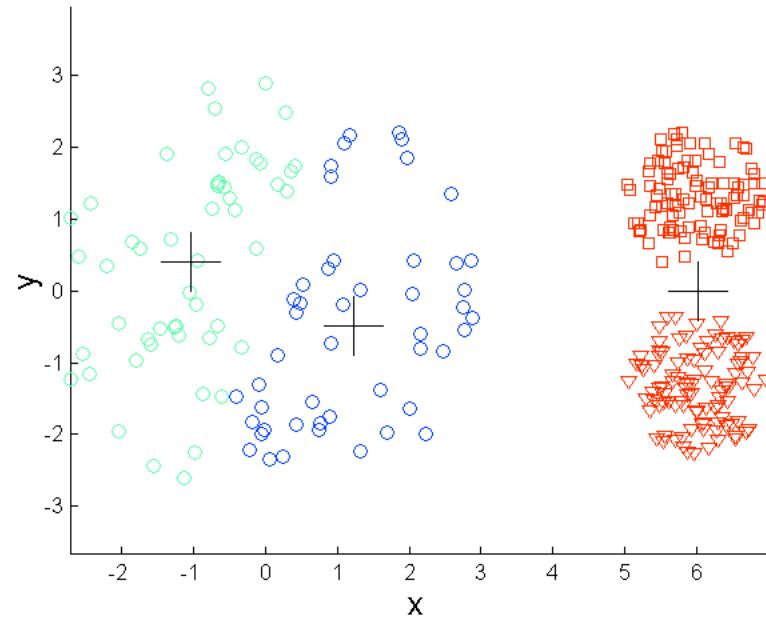


K-means (3 Clusters)

Limitations of K-means: Differing Density

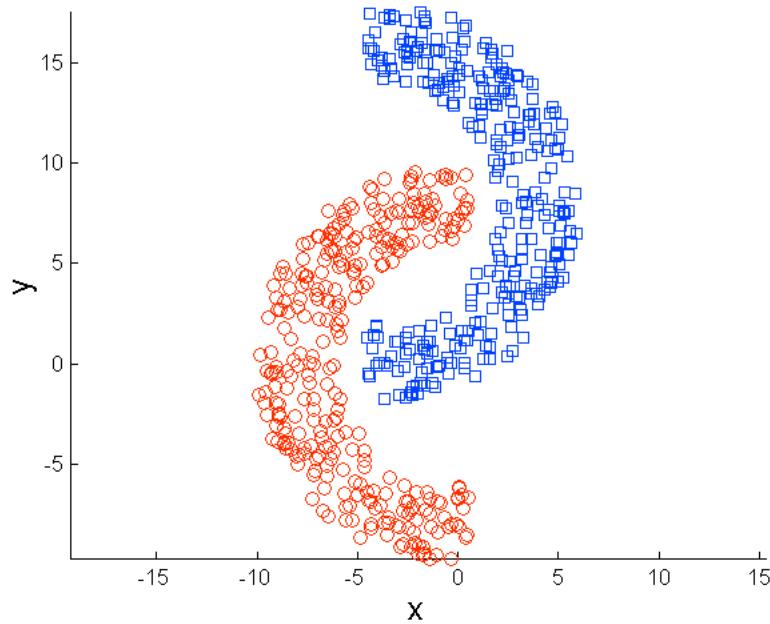


Original Points

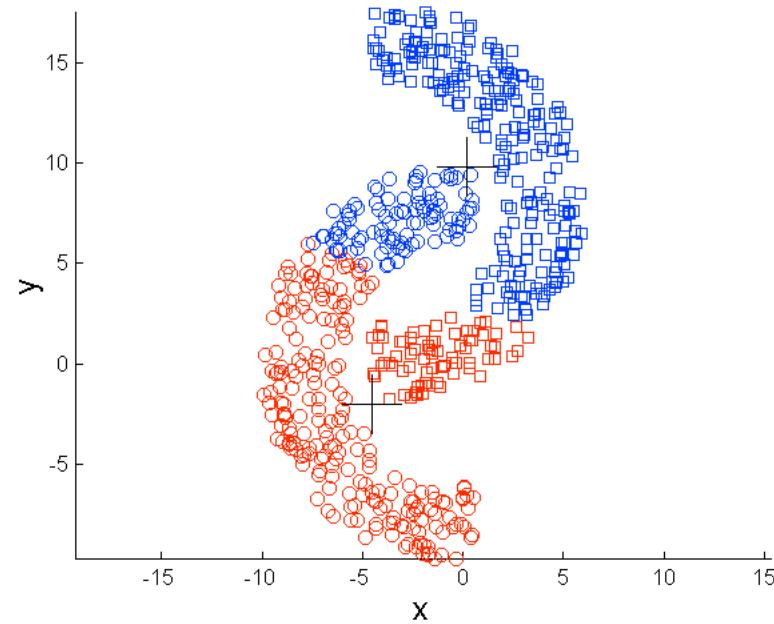


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

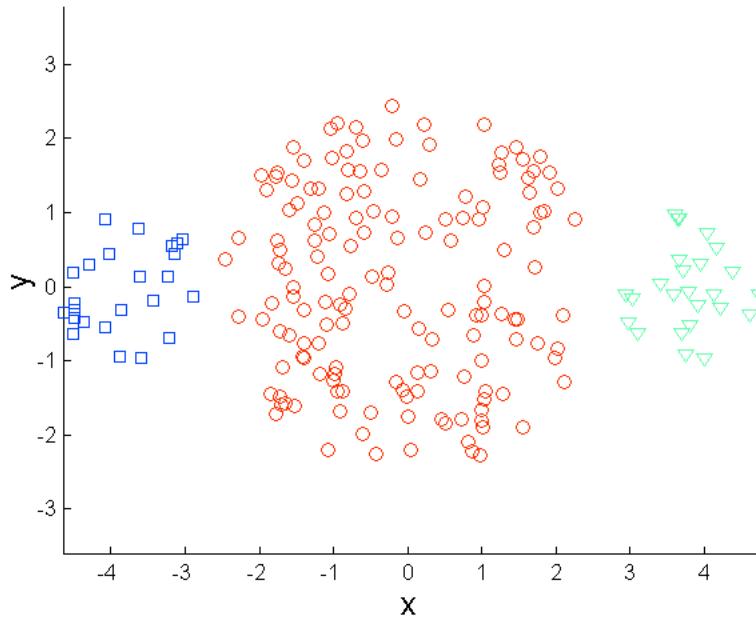


Original Points

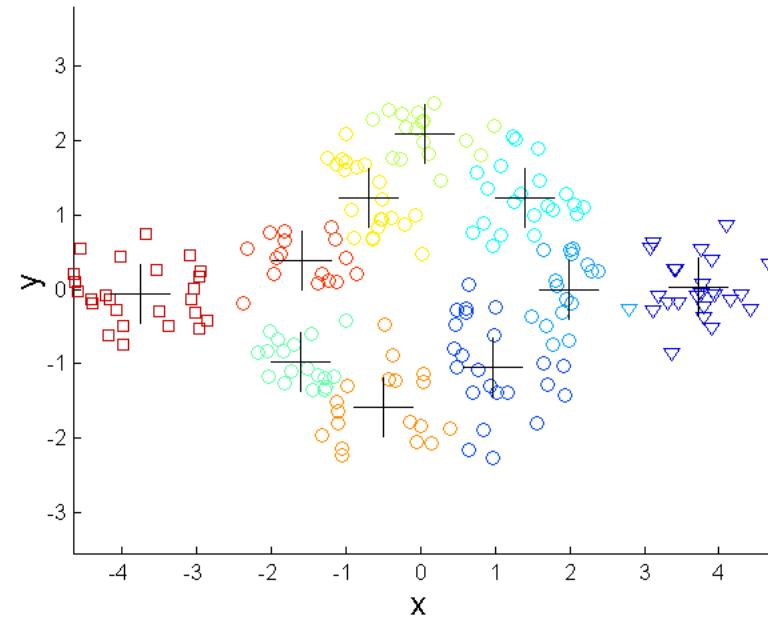


K-means (2 Clusters)

Overcoming K-means Limitations



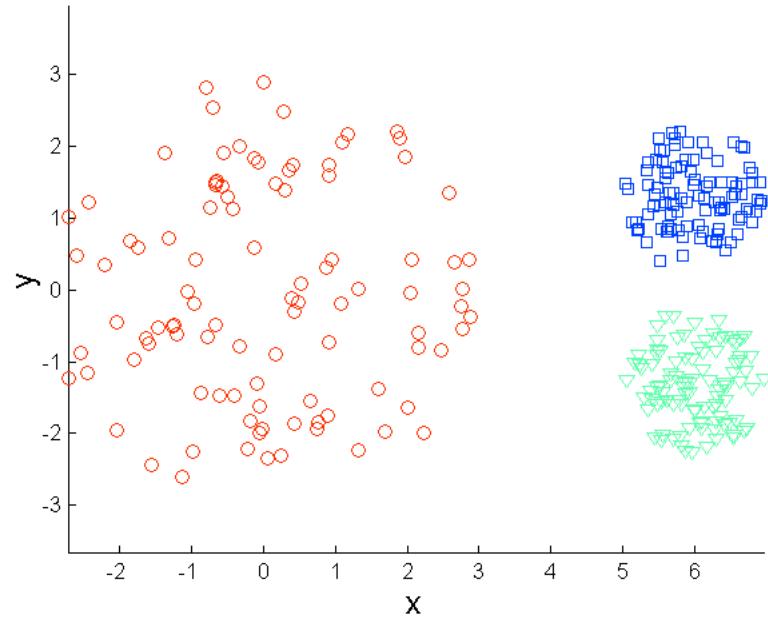
Original Points



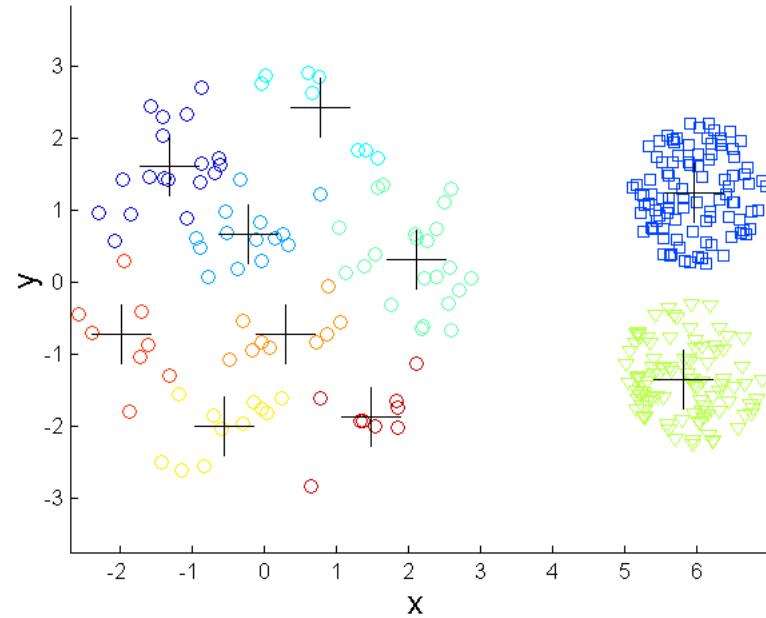
K-means Clusters

One solution is to use many clusters.
Find parts of clusters, but need to put together.

Overcoming K-means Limitations

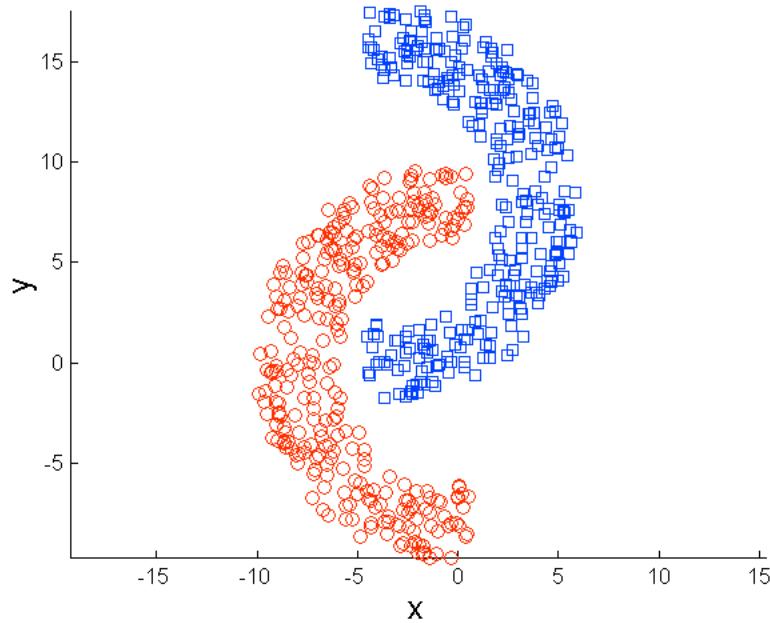


Original Points

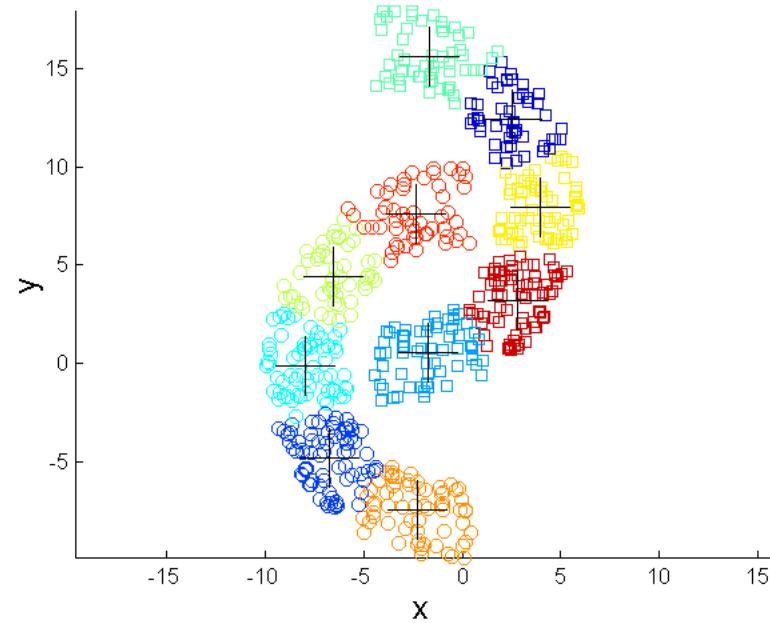


K-means Clusters

Overcoming K-means Limitations



Original Points



K-means Clusters

Comments on the *K-Means* Method

- Strength: *Efficient*: $O(tkn)$, where n is # objects, k is # clusters, and t is # iterations. Normally, $k, t \ll n$.
 - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*.
- Weakness
 - Applicable only to objects in a continuous n-dimensional space
 - Using the k-modes method for categorical data
 - In comparison, k-medoids can be applied to a wide range of data
 - Need to specify k , the *number* of clusters, in advance
 - Sensitive to noisy data and *outliers*
 - Not suitable to discover clusters with *non-convex shapes*

The K-Medoid Clustering Method

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
 - *PAM* (Partitioning Around Medoids)
 - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
 - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
 - Efficiency improvement on PAM
 - *CLARA* : PAM on samples
 - *CLARANS* : Randomized re-sampling

Prescribed Text Books

	Author(s), Title, Edition, Publishing House
T1	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
T2	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education



BITS Pilani

Pilani | Dubai | Goa | Hyderabad

S2-21_DSECLZC415 : Data Mining Lecture #11 – Cluster Analysis



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Data Mining

Cluster Analysis



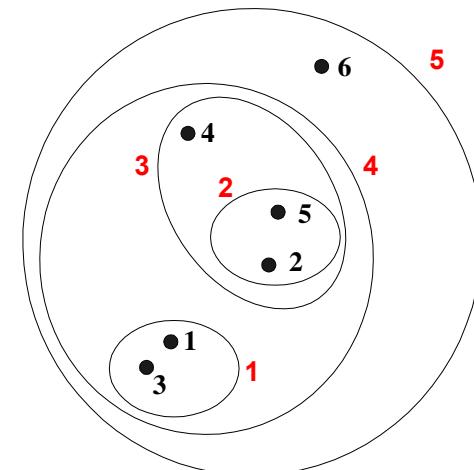
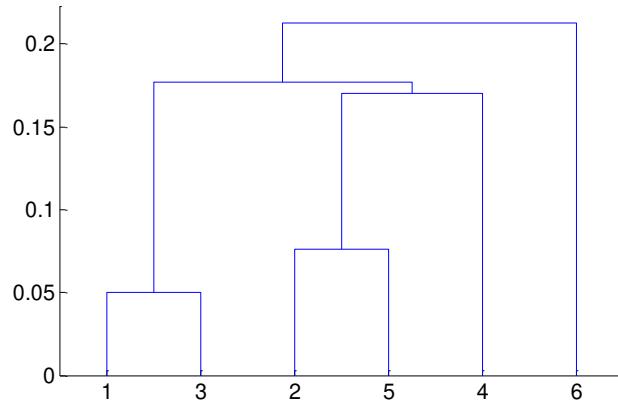
Hierarchical Methods

Hierarchical Clustering

Produces a set of nested clusters organized as a hierarchical tree

Can be visualized as a dendrogram

- A tree like diagram that records the sequences of merges or splits



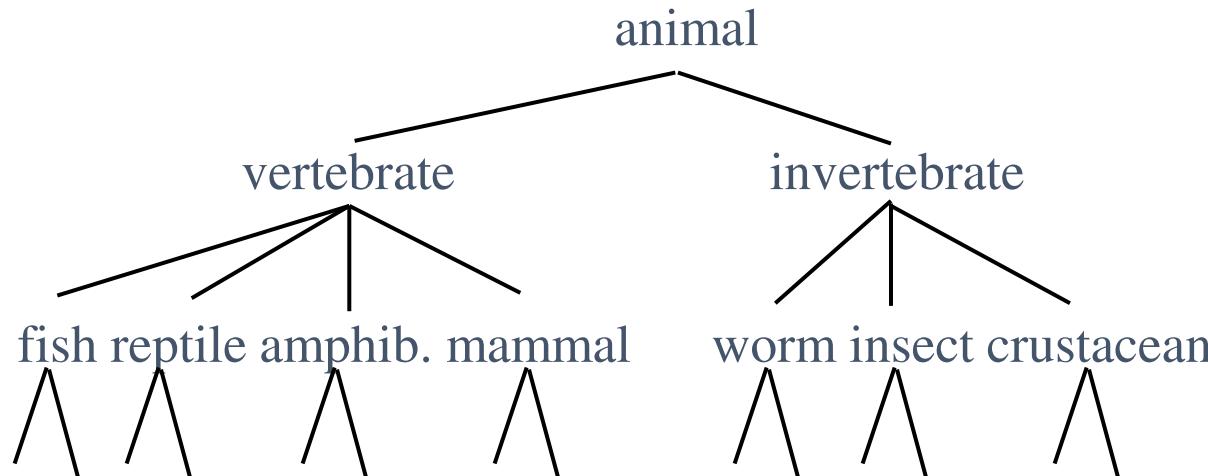
Strengths of Hierarchical Clustering

Do not have to assume any particular number of clusters

- Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level

They may correspond to meaningful taxonomies

- Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)



Hierarchical Clustering

Two main types of hierarchical clustering

- Agglomerative:
 - Start with the points as individual clusters
 - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
- Divisive:
 - Start with one, all-inclusive cluster
 - At each step, split a cluster until each cluster contains a point (or there are k clusters)

Traditional hierarchical algorithms use a similarity or distance matrix

- Merge or split one cluster at a time

Agglomerative Clustering Algorithm

More popular hierarchical clustering technique

Basic algorithm is straightforward

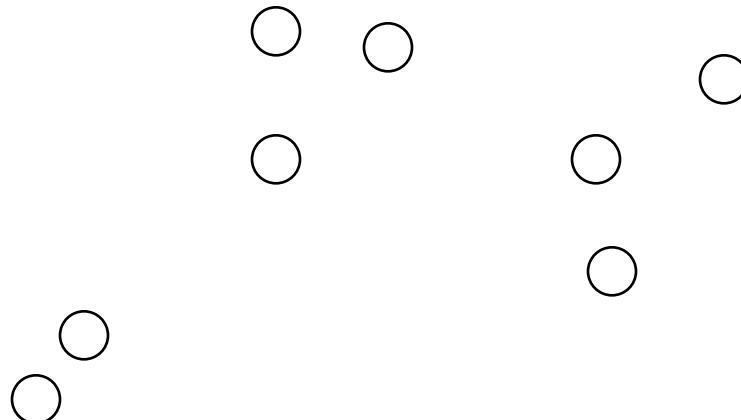
1. Compute the proximity matrix
2. Let each data point be a cluster
- 3. Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
- 6. Until** only a single cluster remains

Key operation is the computation of the proximity of two clusters

- Different approaches to defining the distance between clusters distinguish the different algorithms

Starting Situation

- Start with clusters of individual points and a proximity matrix



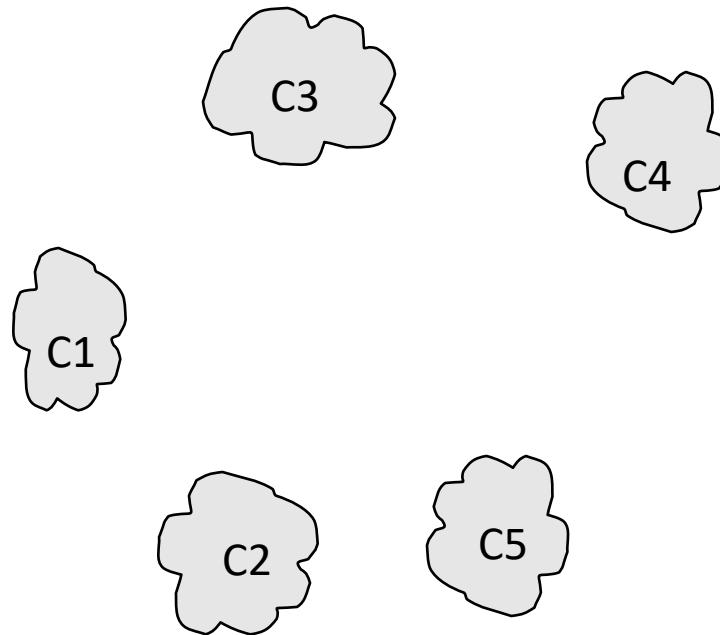
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

p1 **p2** **p3** **p4** ■ ■ ■ **p9** **p10** **p11** **p12**

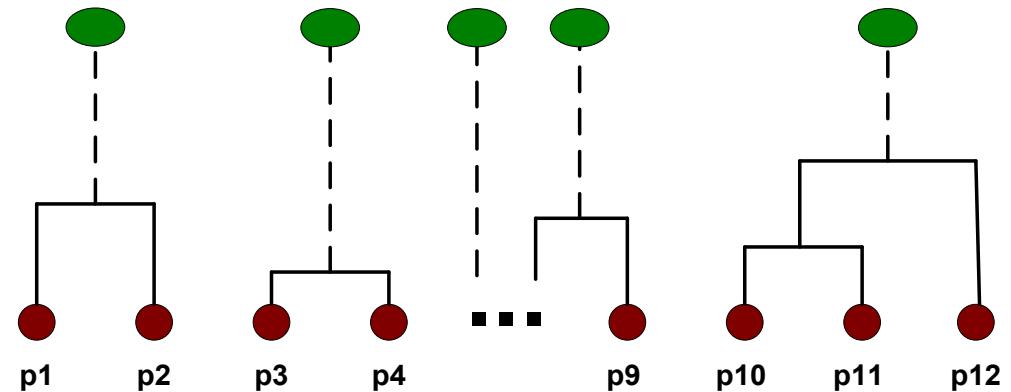
Intermediate Situation

- After some merging steps, we have some clusters



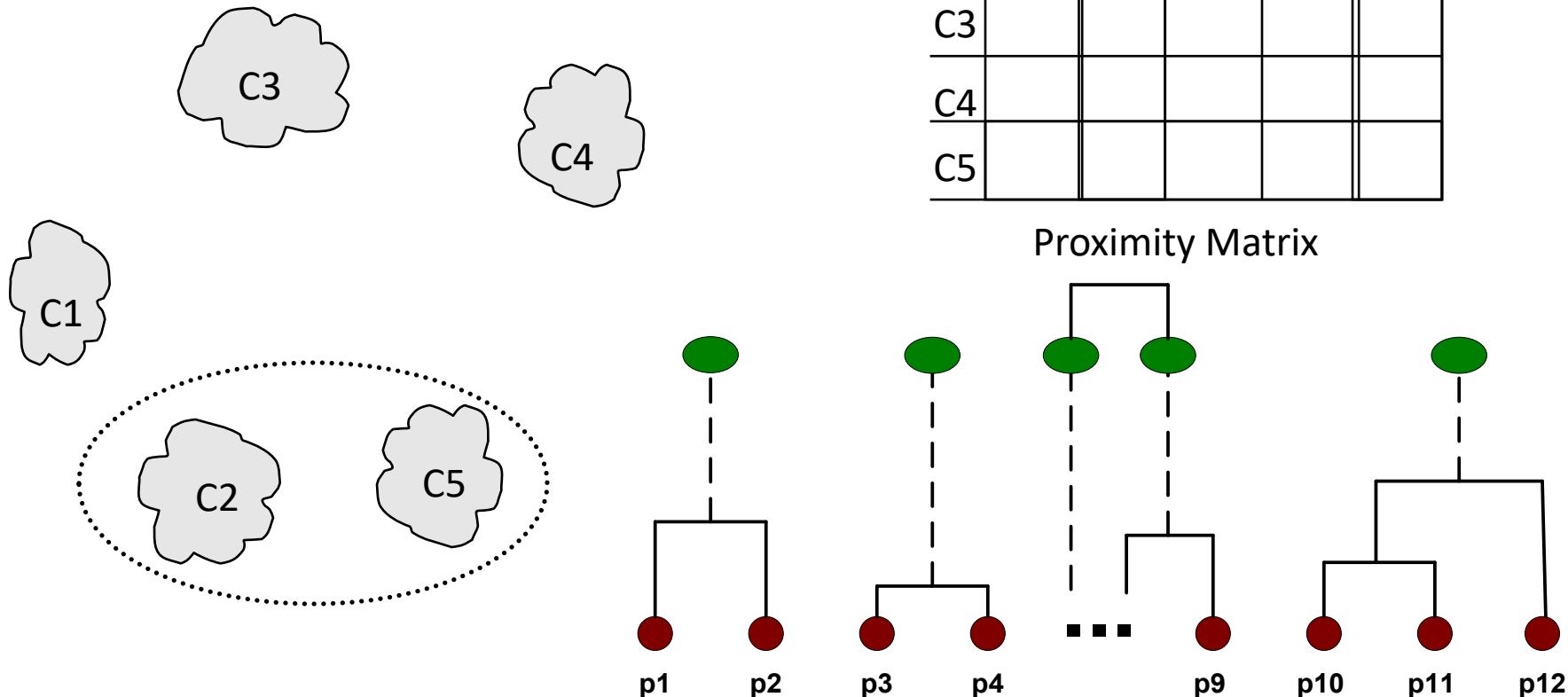
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix

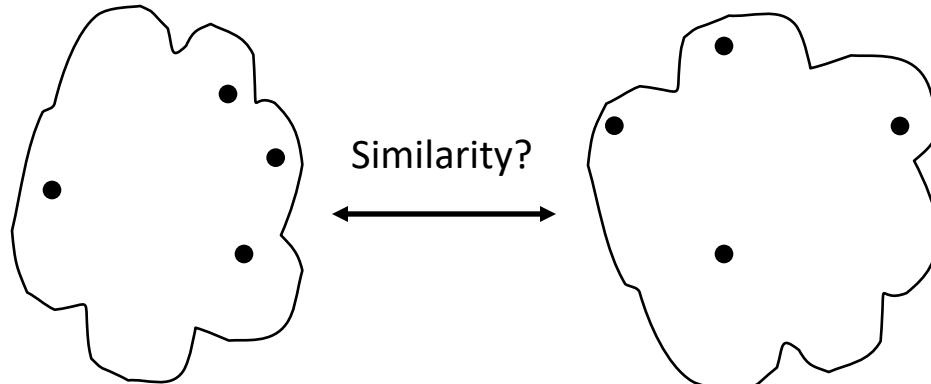


Intermediate Situation

- We want to merge the two closest clusters (C_2 and C_5) and update the proximity matrix.



How to Define Inter-Cluster Similarity

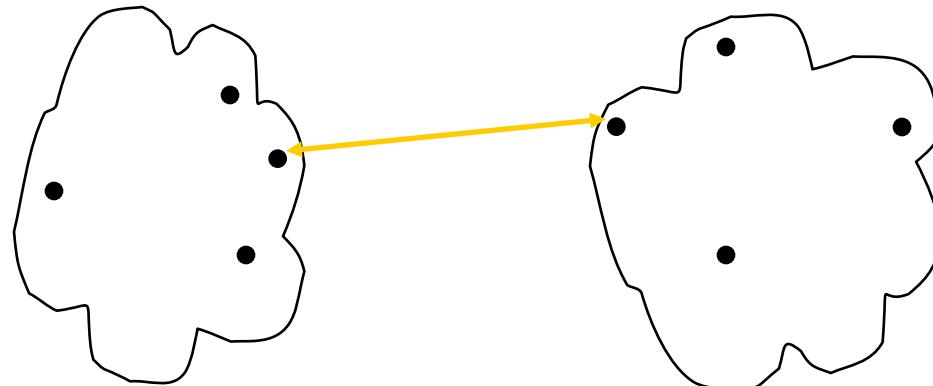


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						

Proximity Matrix

How to Define Inter-Cluster Similarity

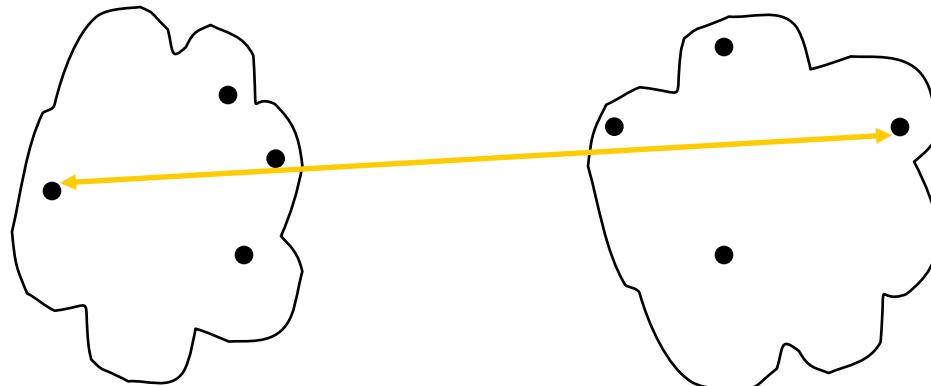


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.

Proximity Matrix

How to Define Inter-Cluster Similarity

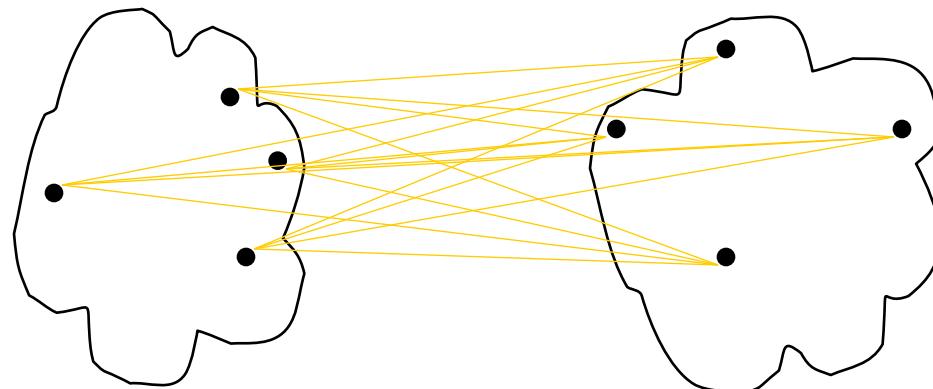


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.

Proximity Matrix

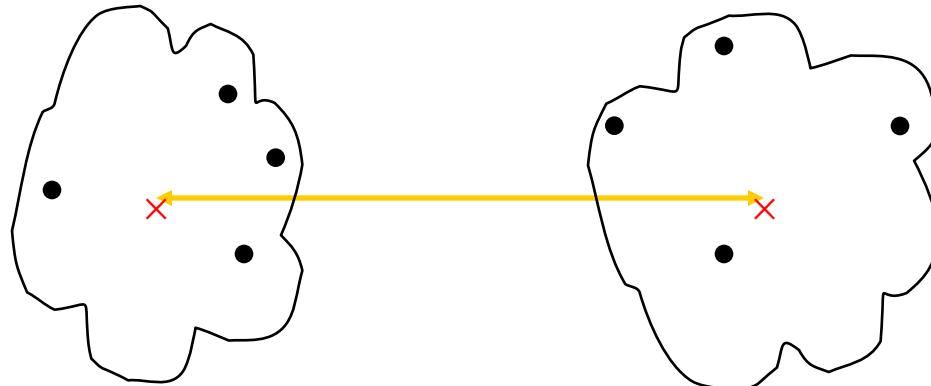
How to Define Inter-Cluster Similarity



- MIN
 - MAX
 - **Group Average**
 - Distance Between Centroids
 - Other methods driven by an objective function
- . . . Proximity Matrix

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.

How to Define Inter-Cluster Similarity

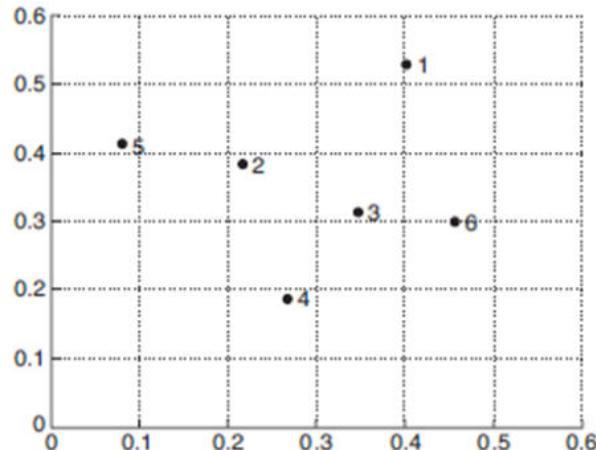


- MIN
- MAX
- Group Average
- **Distance Between Centroids**
- Other methods driven by an objective function

	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.

Proximity Matrix

Clustering Example



Set of 6 two-dimensional points.

Point	<i>x</i> Coordinate	<i>y</i> Coordinate
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30

xy coordinates of 6 points.

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Euclidean distance matrix for 6 points.

	P1	P2	P3, P6	P4	P5
P1	0	.24	.22	.37	.34
P2		0	.15	.2	.14
P3, P6			0	.155	.28
P4				0	.29
P5					0

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Euclidean distance matrix for 6 points.

	P1	P2,P5	P3,P6	P4
P1	0	.24	.22	.37
P2,P5		0	.15	.2
P3,P6			0	.155
P4				0



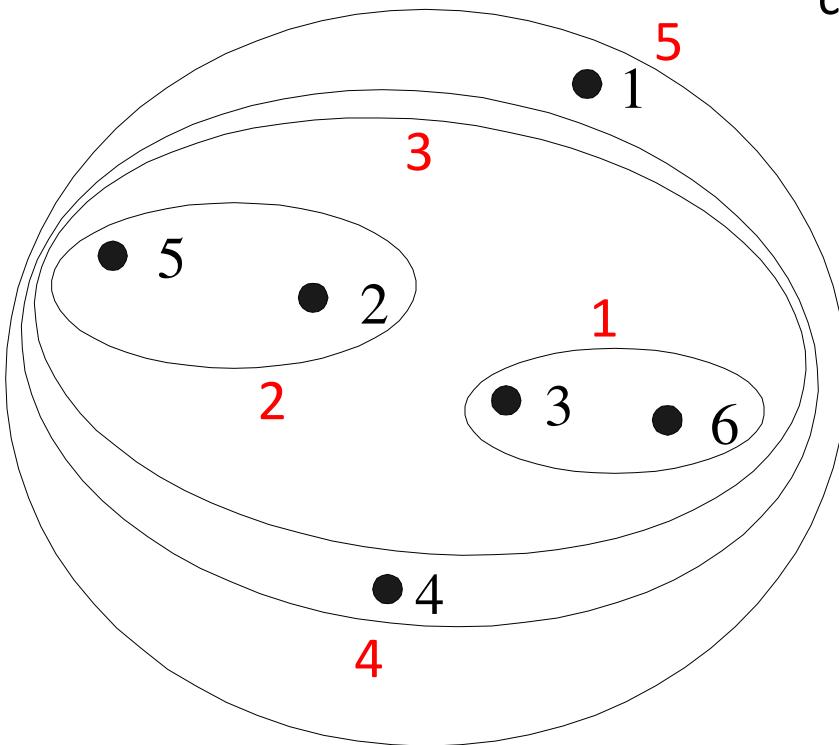
	P1	P2,P5, P3,P6	P4
P1	0	.22	.37
P2,P5, P3,P6		0	.155
P4			0

Hierarchical Clustering example - min

	P1	P2,P5,P3, P6,P4
P1	0	0.22
P2,P5,P3, P6,P4		0



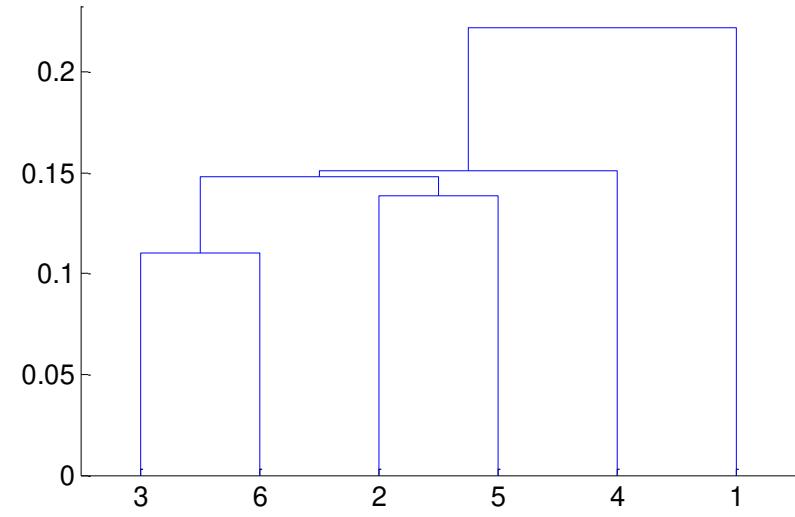
Hierarchical Clustering: MIN



Nested Clusters

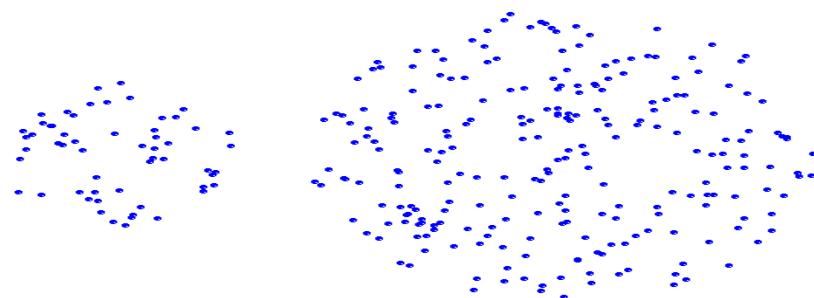
Similarity of two clusters is based on the two most similar (closest) points in the different clusters

Determined by one pair of points, i.e., by one link in the proximity graph.

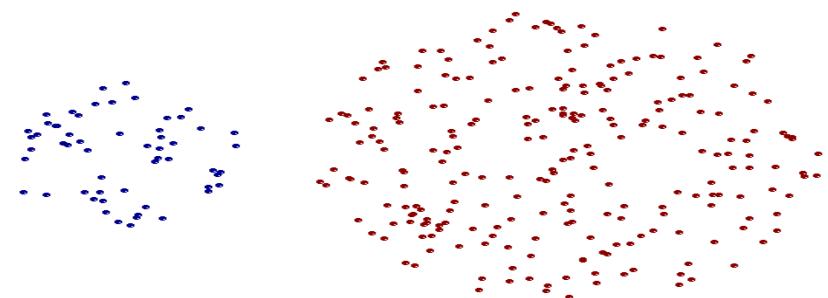


Dendrogram

Strength of MIN



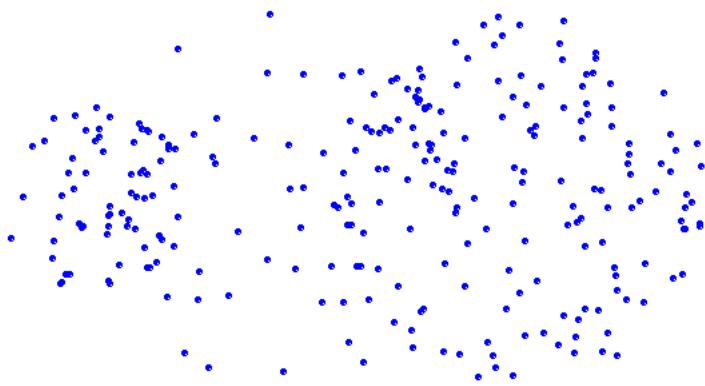
Original Points



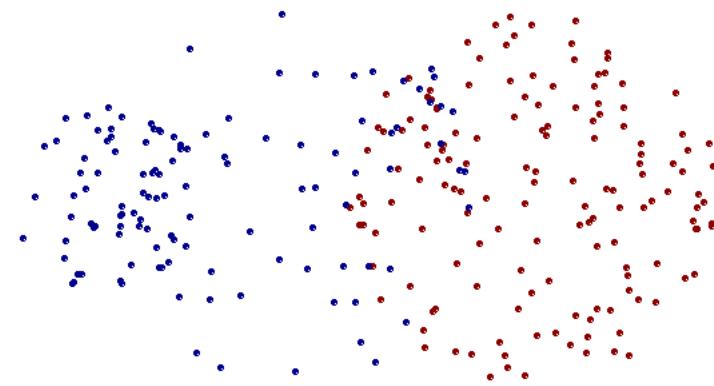
Two Clusters

- Can handle non-elliptical shapes

Limitations of MIN



Original Points



Two Clusters

- Sensitive to noise and outliers

	P1	P2	P3, P6	P4	P5
P1	0	.24	.23	.37	.34
P2		0	.25	.2	.14
P3, P6			0	.22	.39
P4				0	.29
P5					0

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Euclidean distance matrix for 6 points.

	P1	P2,P5	P3,P6	P4
P1	0	.34	.23	.37
P2,P5		0	.39	.29
P3,P6			0	.22
P4				0



	P1	P2,P5	P3,P6, P4
P1	0	.34	.37
P2,P5		0	.39
P3,P6, P4			0

Hierarchical Clustering example - max

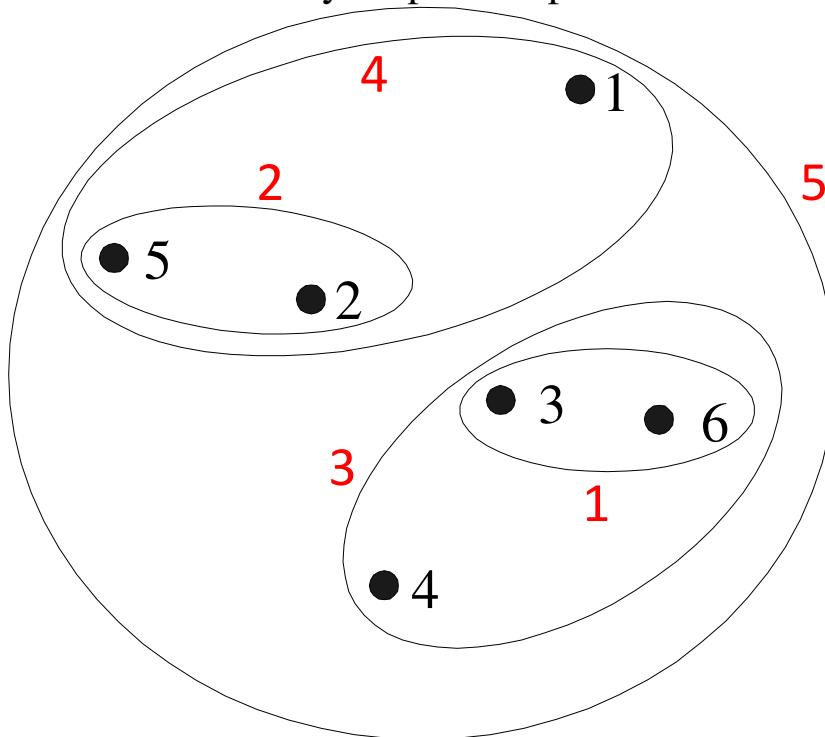
	P1, P2,P5	P3, P6,P4
P1, P2,P5	0	0.39
P3, P6,P4		0



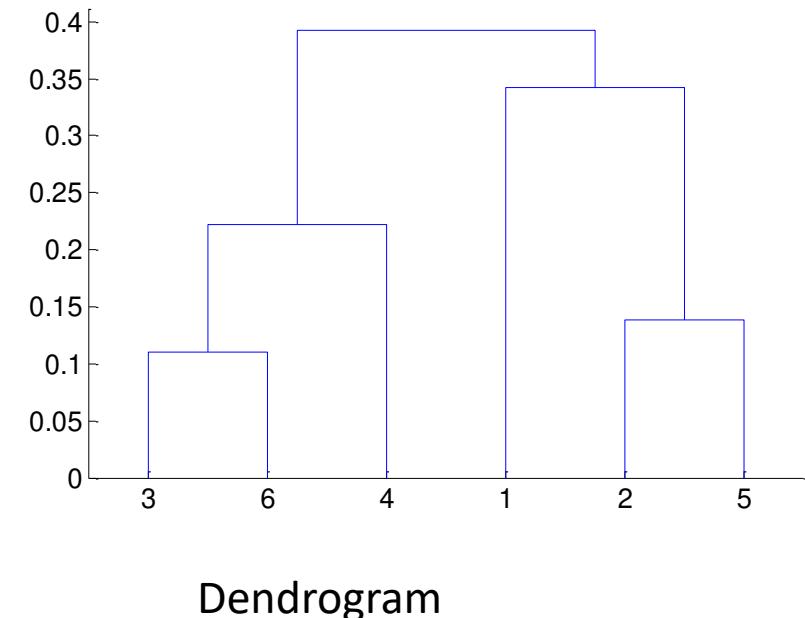
Hierarchical Clustering: MAX

Similarity of two clusters is based on the two least similar (most distant) points in the different clusters

Determined by all pairs of points in the two clusters

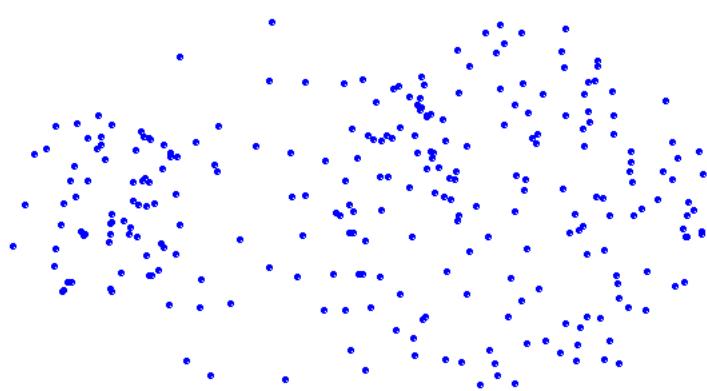


Nested Clusters

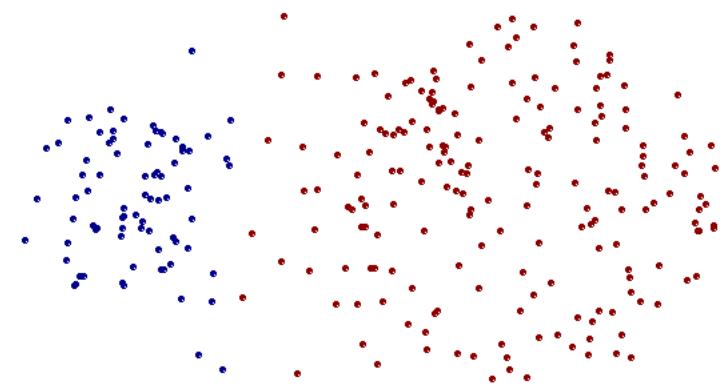


Dendrogram

Strength of MAX



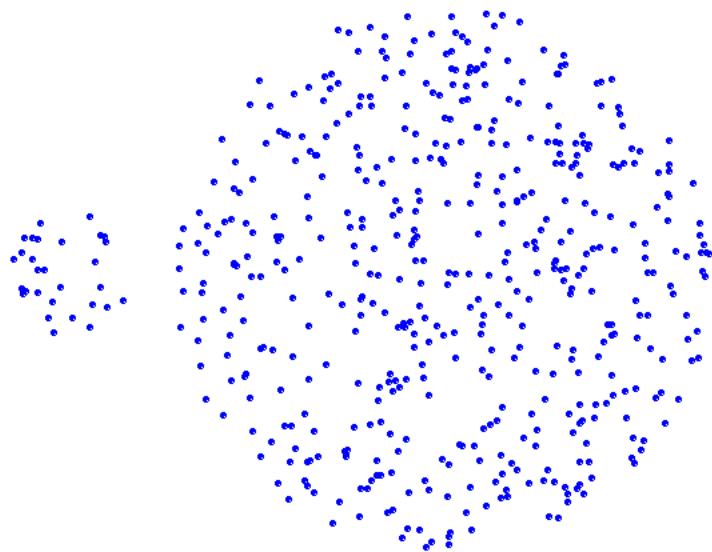
Original Points



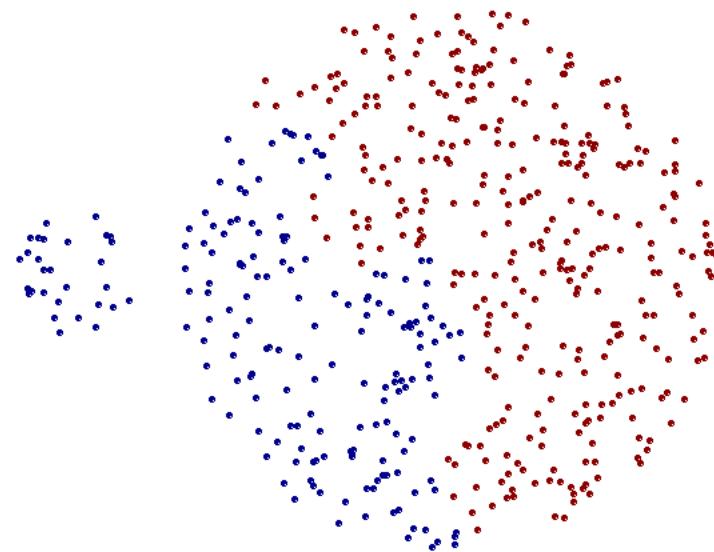
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points

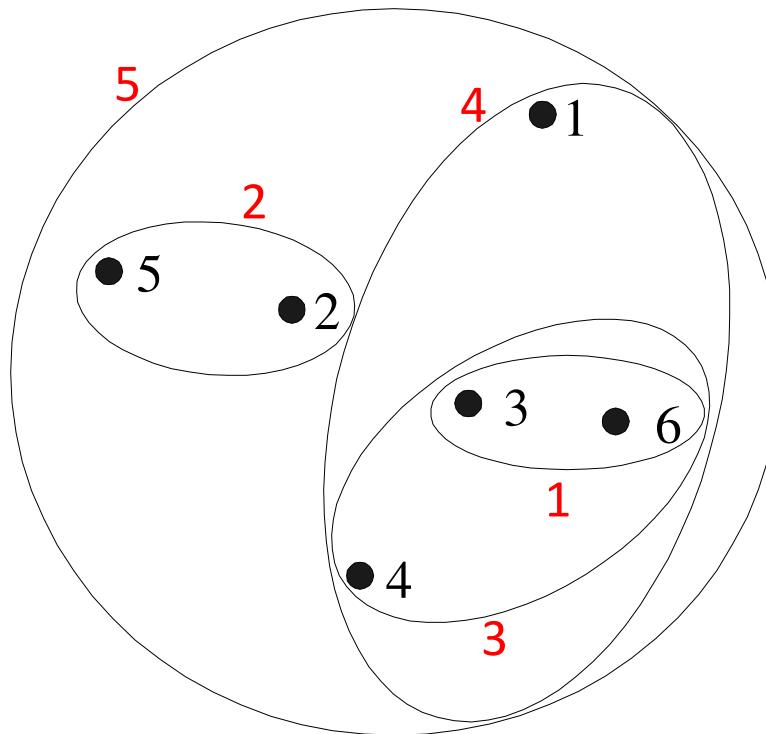


Two Clusters

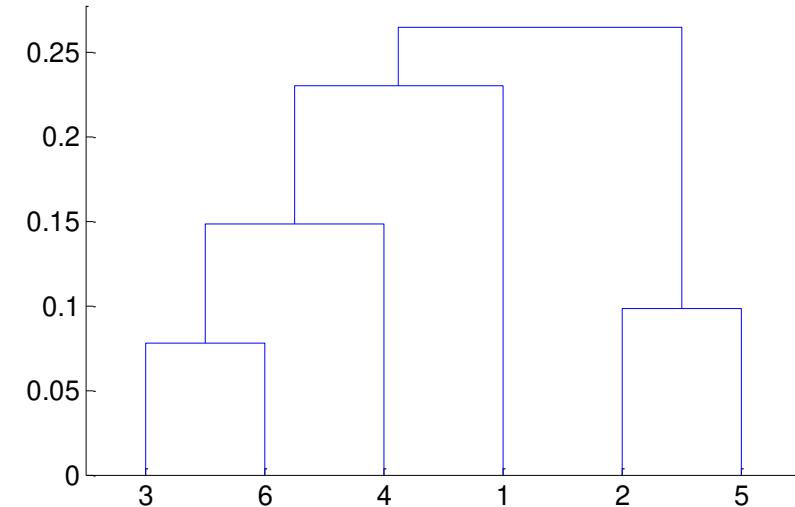
- Tends to break large clusters
- Biased towards globular clusters

Hierarchical Clustering: Group Average

Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

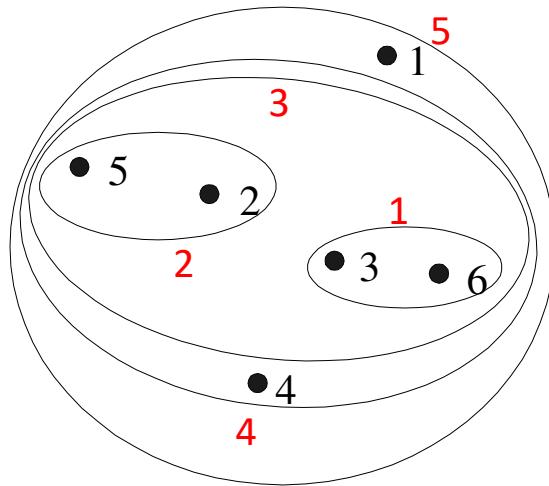


Nested Clusters

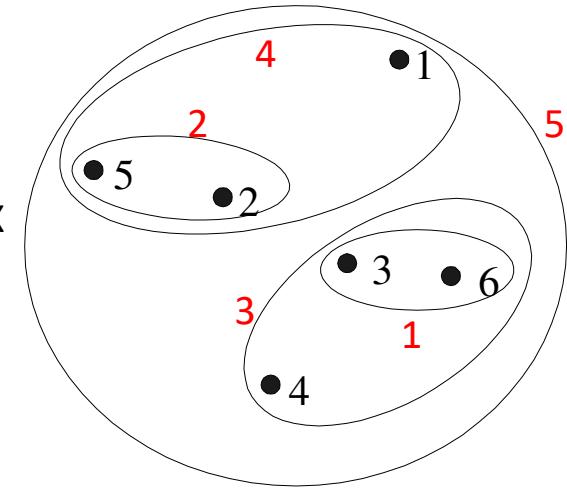


Dendrogram

Hierarchical Clustering: Comparison

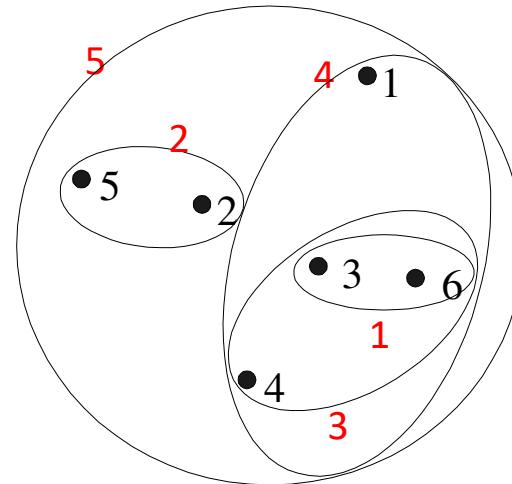


MIN



MAX

Group Average

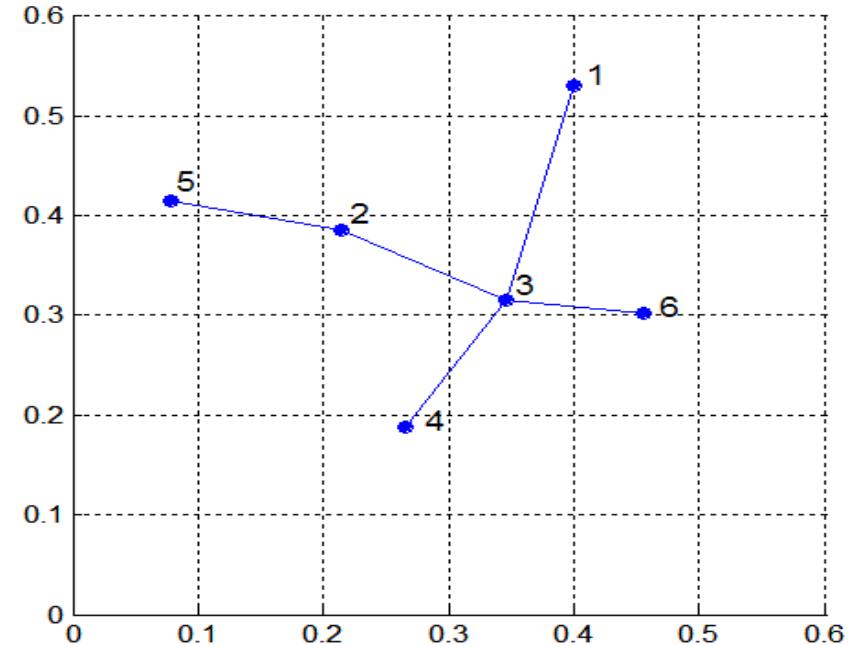
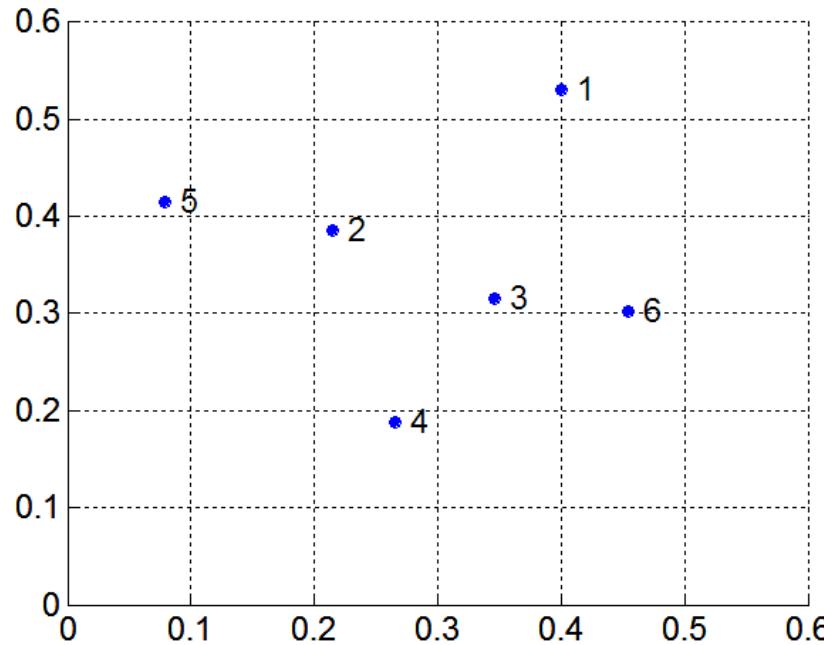


MST: Divisive Hierarchical Clustering

- Build MST (Minimum Spanning Tree)
 - Start with a tree that consists of any point
 - In successive steps, look for the closest pair of points (p, q) such that one point (p) is in the current tree but the other (q) is not
 - Add q to the tree and put an edge between p and q

	p1	p2	p3	p4	p5	p6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
p6	0.23	0.25	0.11	0.22	0.39	0.00

Euclidean distance matrix for 6 points.



MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

- 1: Compute a minimum spanning tree for the proximity graph.
 - 2: **repeat**
 - 3: Create a new cluster by breaking the link corresponding to the largest distance
 (smallest similarity).
 - 4: **until** Only singleton clusters remain
-

Hierarchical Clustering: Time and Space requirements

$O(N^2)$ space since it uses the proximity matrix.

- N is the number of points.

$O(N^3)$ time in many cases

- There are N steps and at each step the size, N^2 , proximity matrix must be updated and searched
- Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

Hierarchical Clustering: Problems and Limitations

Once a decision is made to combine two clusters, it cannot be undone

No objective function is directly minimized

Different schemes have problems with one or more of the following:

- Sensitivity to noise and outliers
- Difficulty handling different sized clusters and convex shapes
- Breaking large clusters



Density based Cluster Analysis

Density-Based Clustering Methods

Clustering based on density (local cluster criterion), such as density-connected points

Major features:

- Discover clusters of arbitrary shape
- Handle noise
- One scan
- Need density parameters as termination condition

Several interesting studies:

- DBSCAN: Ester, et al. (KDD'96)
- OPTICS: Ankerst, et al (SIGMOD'99).
- DENCLUE: Hinneburg & D. Keim (KDD'98)
- CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based)

Density-Based Clustering: Basic Concepts

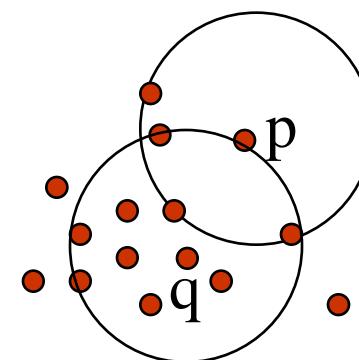
Two parameters:

- *Eps*: Maximum radius of the neighbourhood
- *MinPts*: Minimum number of points in an *Eps*-neighbourhood of that point

$N_{Eps}(p)$: {q belongs to D | $\text{dist}(p,q) \leq Eps$ }

Directly density-reachable: A point p is directly density-reachable from a point q w.r.t. $Eps, MinPts$ if

- p belongs to $N_{Eps}(q)$
- core point condition:
 $|N_{Eps}(q)| \geq MinPts$



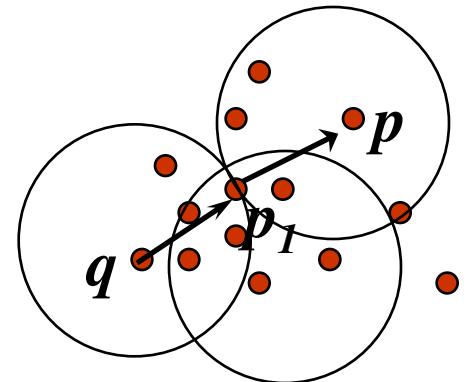
$MinPts = 5$

$Eps = 1 \text{ cm}$

Density-Reachable and Density-Connected

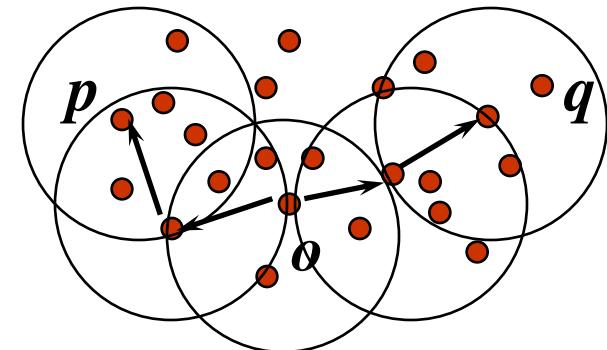
Density-reachable:

- A point p is **density-reachable** from a point q w.r.t. $Eps, MinPts$ if there is a chain of points $p_1, \dots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i



Density-connected

- A point p is **density-connected** to a point q w.r.t. $Eps, MinPts$ if there is a point o such that both, p and q are density-reachable from o w.r.t. Eps and $MinPts$



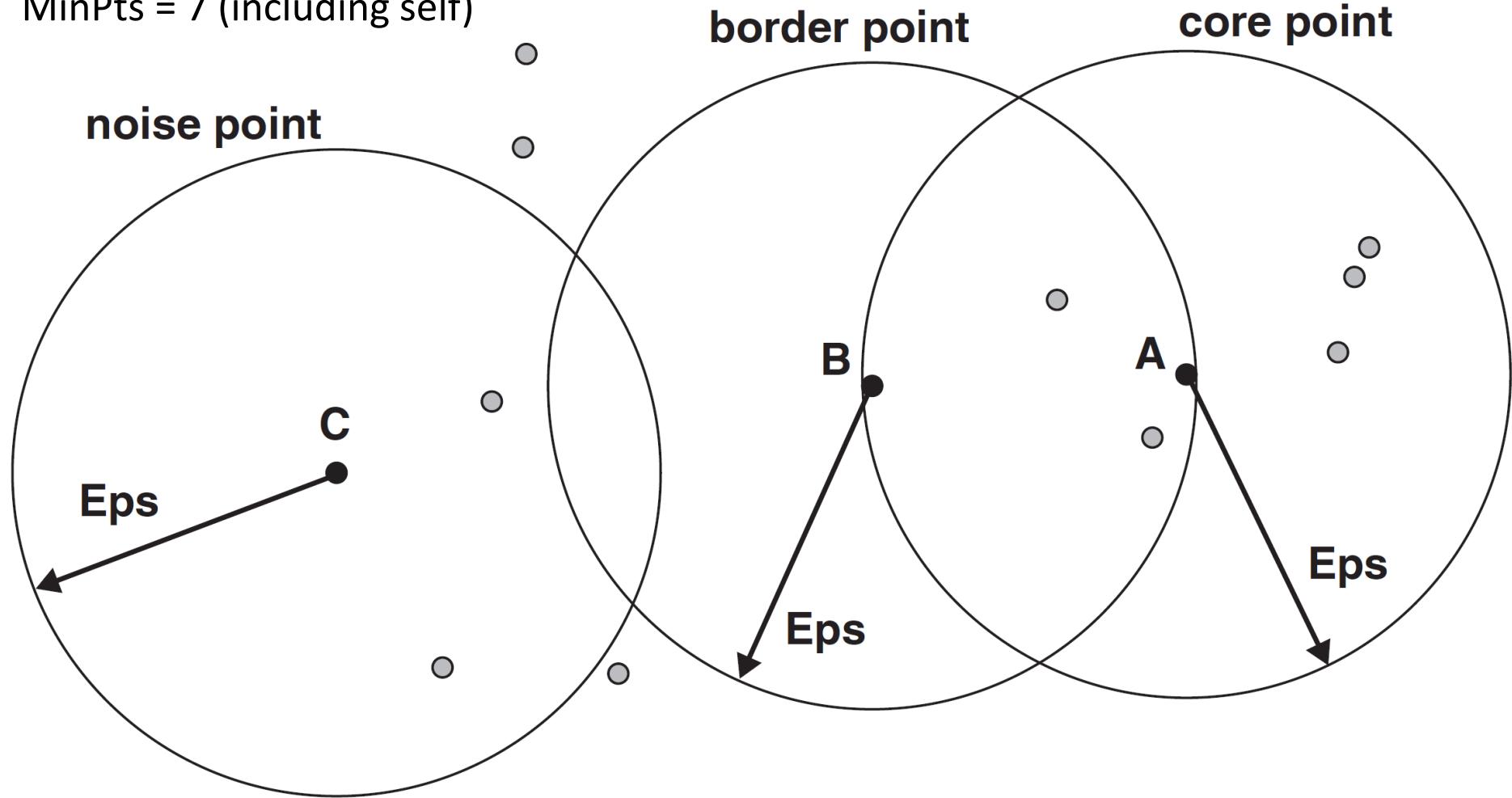
DBSCAN

DBSCAN is a density-based algorithm.

- Density = number of points within a specified radius (Eps)
- A point is a **core point** if it has more than a specified number of points (MinPts) within Eps
 - These are points that are at the interior of a cluster
- A **border point** has fewer than MinPts within Eps, but is in the neighborhood of a core point
- A **noise point** is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points

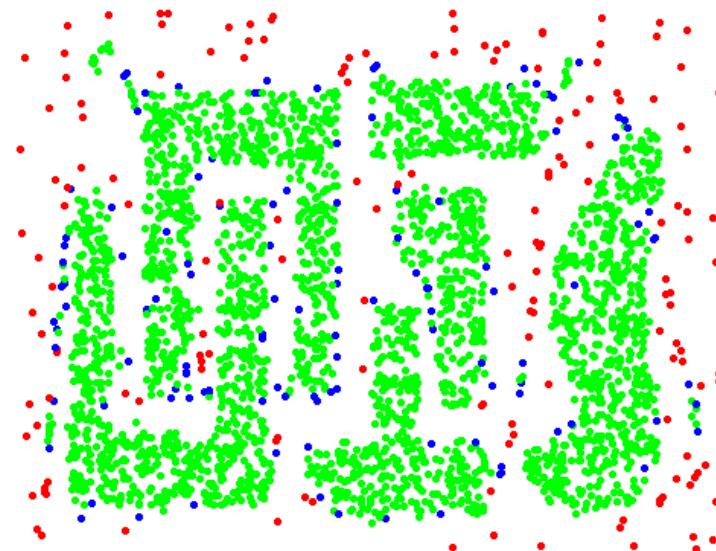
MinPts = 7 (including self)



DBSCAN: Core, Border and Noise Points



Original Points



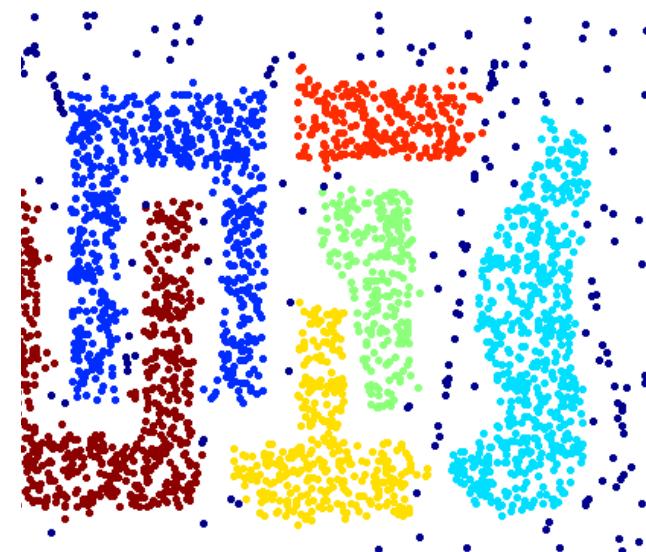
Point types: **core**, **border**
and **noise**

Eps = 10, MinPts = 4

When DBSCAN Works Well



Original Points



Clusters

- Resistant to Noise
- Can handle clusters of different shapes and sizes

DBSCAN: Sensitive to Parameters

Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.

DBSCAN does not work well for
Varying densities
High-dimensional data

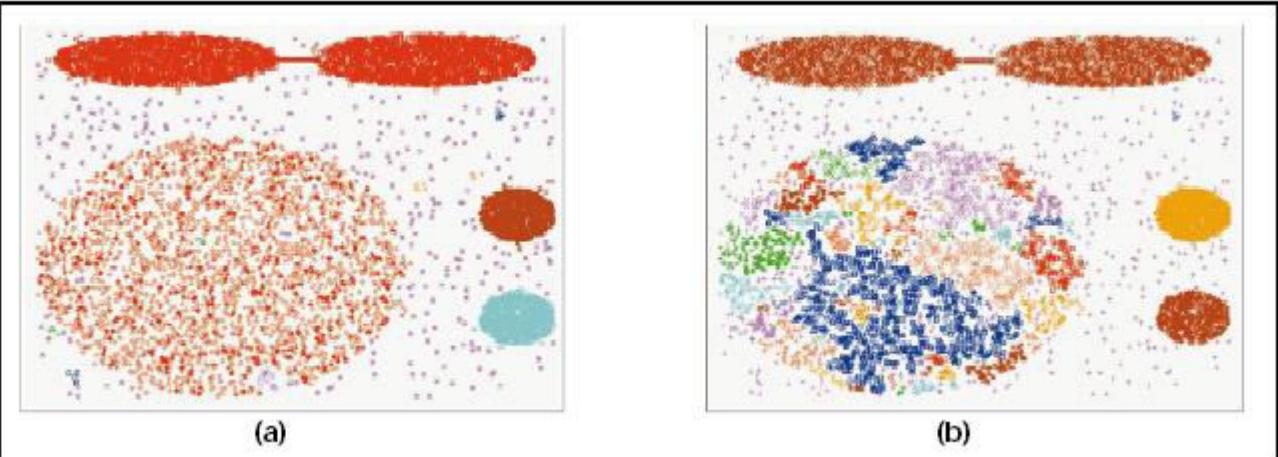
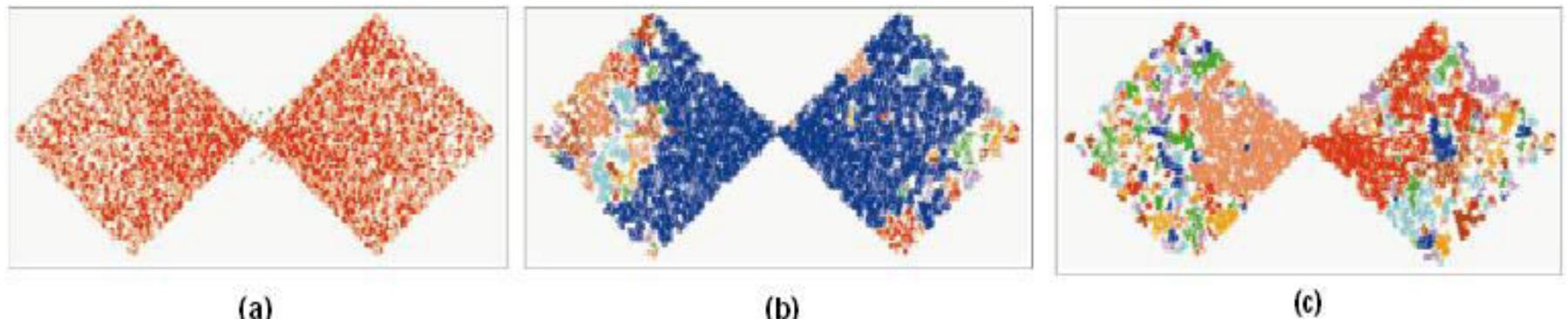


Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.

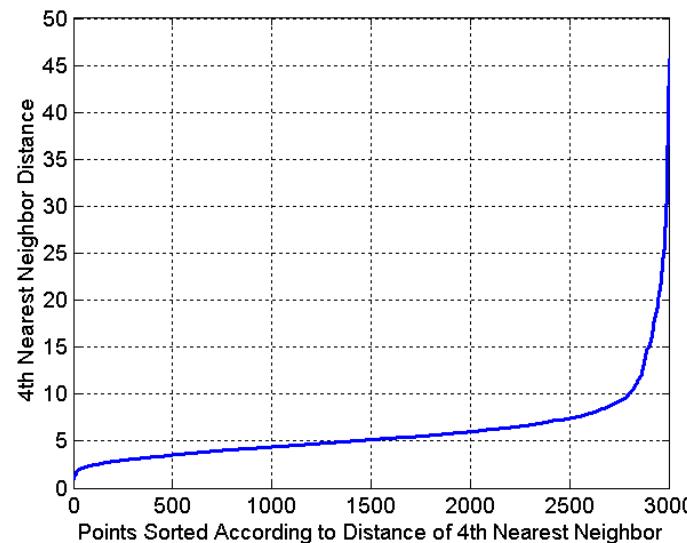


DBSCAN: Determining EPS and MinPts

Idea is that for points in a cluster, their k^{th} nearest neighbors are at roughly the same distance

Noise points have the k^{th} nearest neighbor at farther distance

So, plot sorted distance of every point to its k^{th} nearest neighbor



Prescribed Text Books

	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
R1	Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner by Vijay Kotu and Bala Deshpande Morgan Kaufmann Publishers

CMPUT 690 – Topics in Databases

Knowledge Discovery in Databases



*Additional Slides for Clustering II:
Animation of the OPTICS Algorithm*

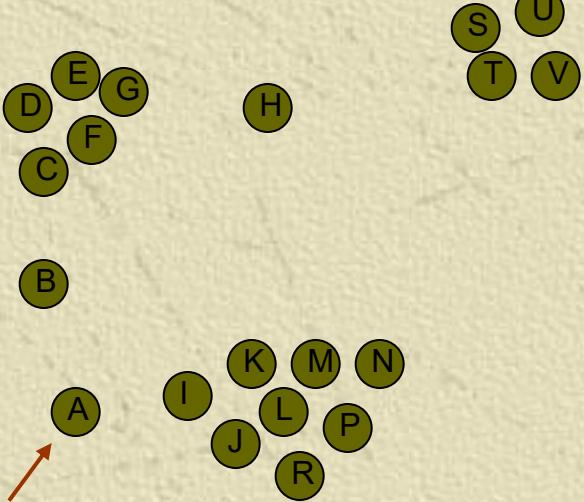
Dr. Jörg Sander

Department of Computing Science
University of Alberta

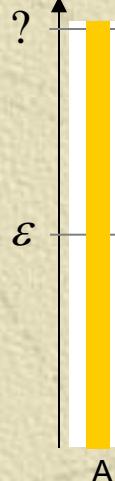


Computing a Cluster Ordering - Example

- Example Database (2-dimensional, 20 points)
 $\varepsilon = 44$, $MinPts = 3$

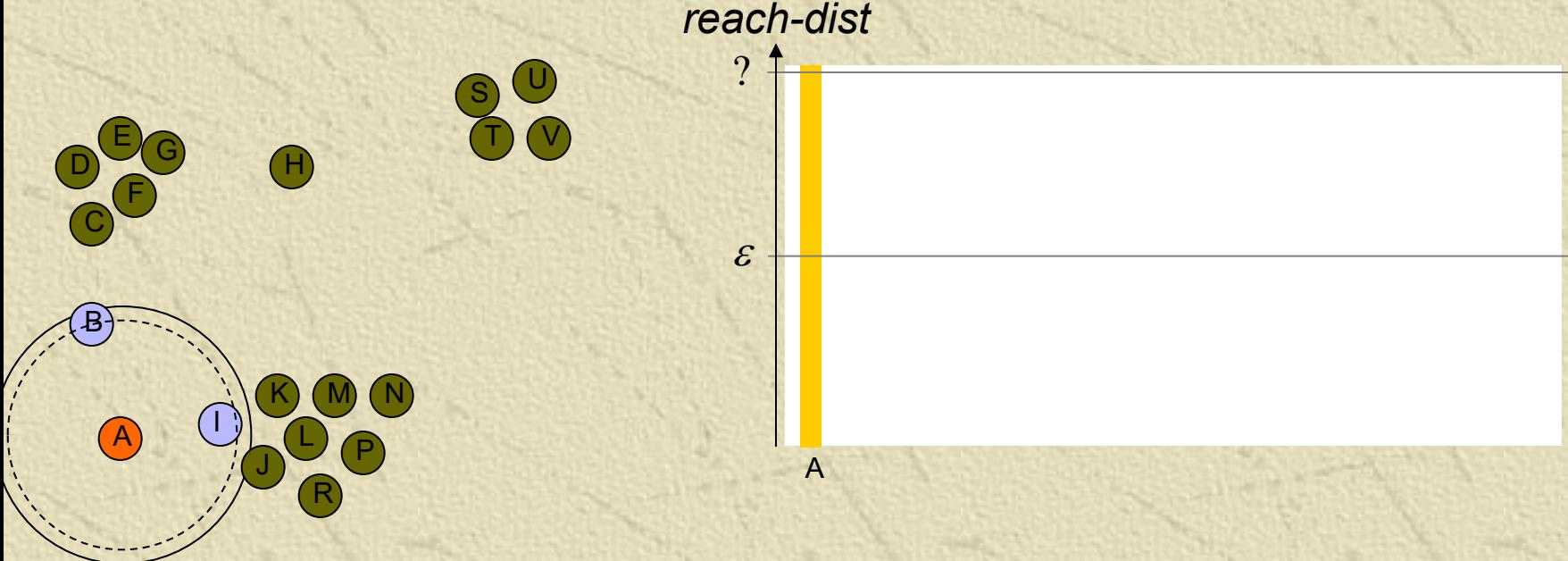


reach-dist



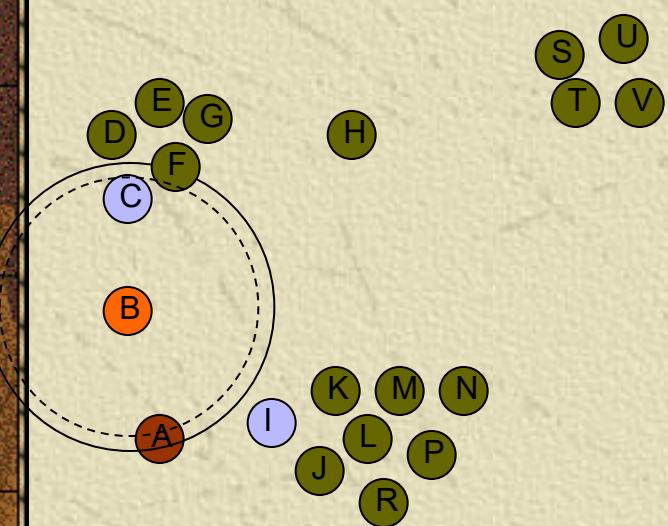
controlList: (A, ?)

Computing a Cluster Ordering - Example

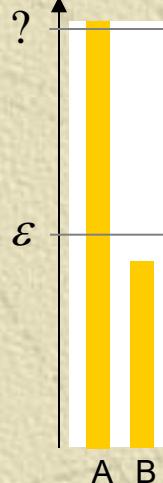


controlList: (B,40) (I, 40)

Computing a Cluster Ordering - Example

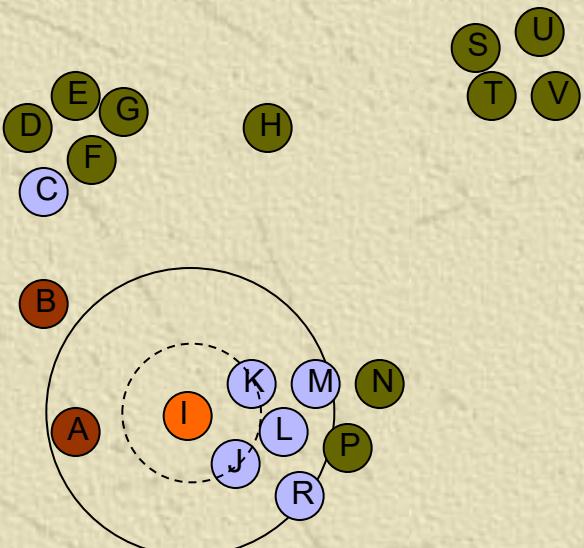


reach-dist

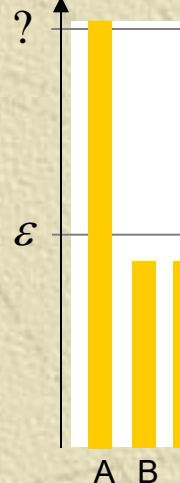


controlList: (I, 40) (C, 40)

Computing a Cluster Ordering - Example

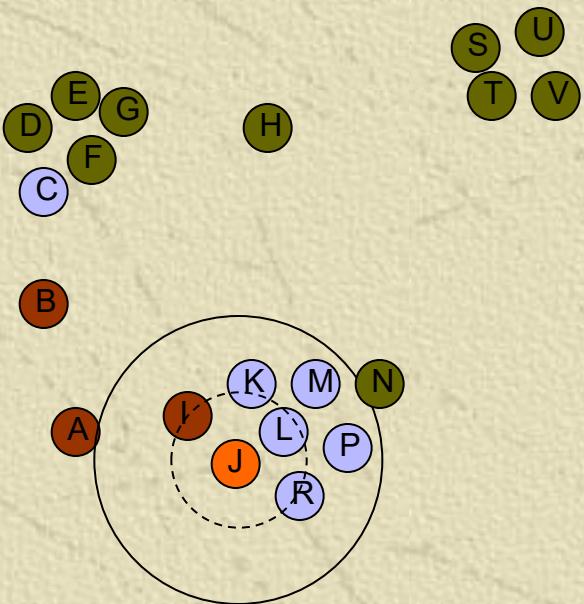


reach-dist

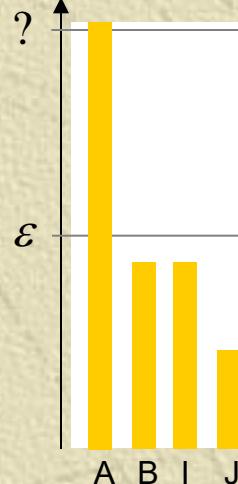


controlList: (J, 20) (K, 20) (L, 31) (C, 40) (M, 40) (R, 43)

Computing a Cluster Ordering - Example

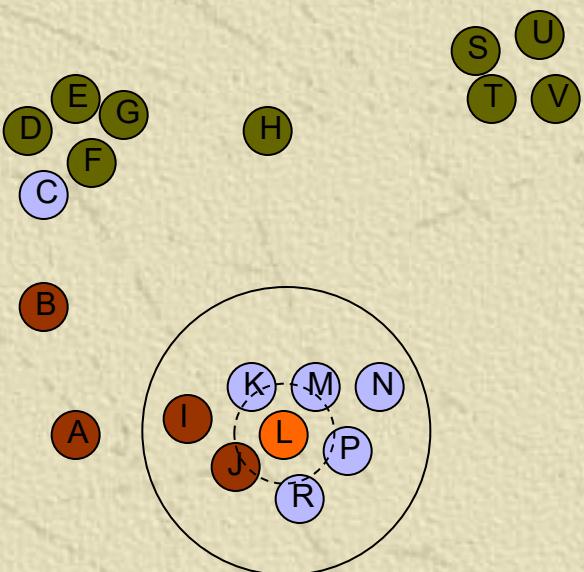


reach-dist

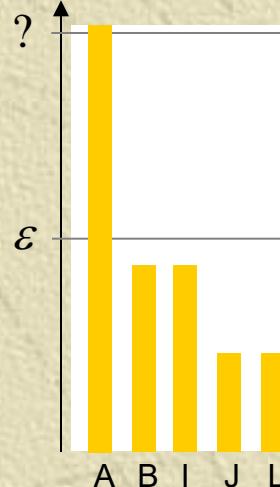


controlList: (L, 19) (K, 20) (R, 21) (M, 30) (P, 31) (C, 40)

Computing a Cluster Ordering - Example

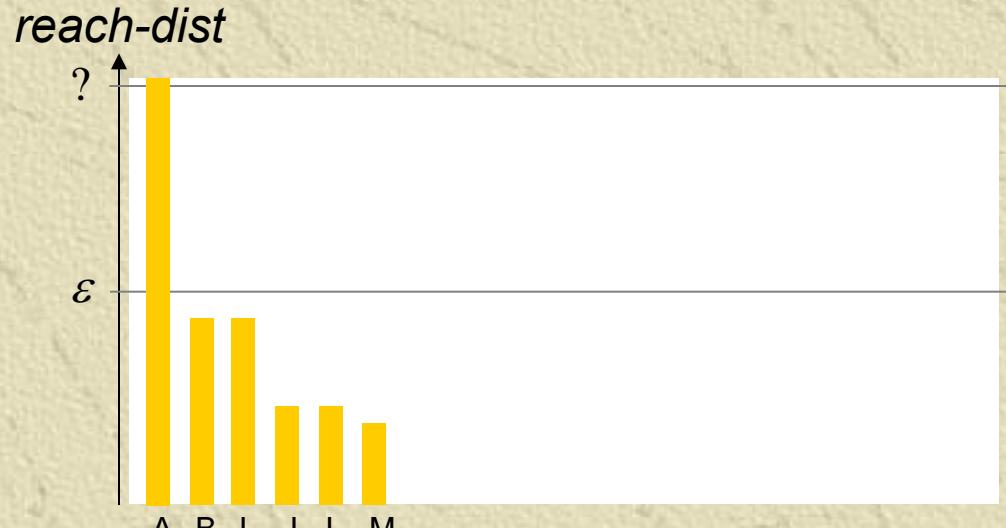
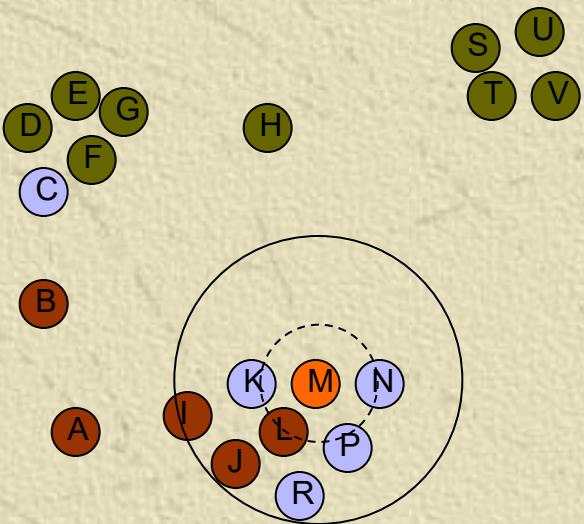


reach-dist



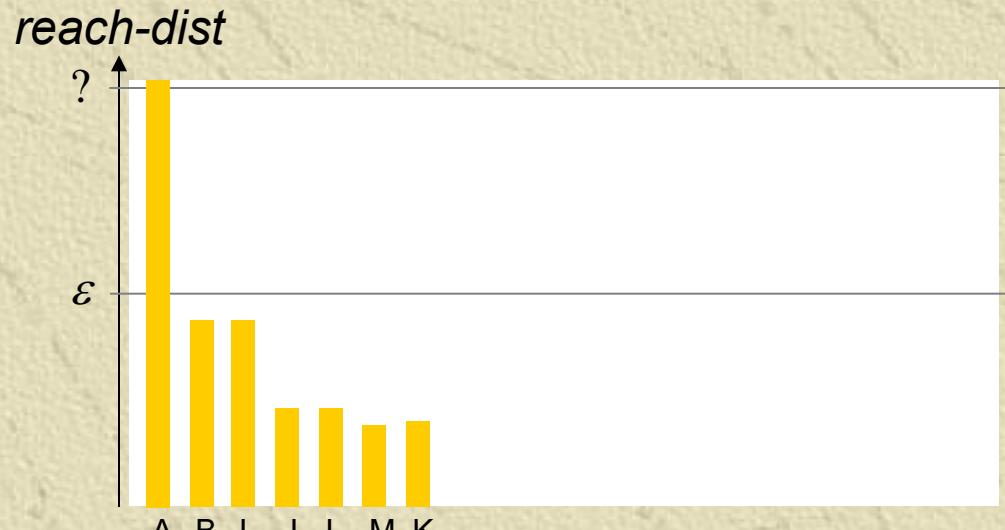
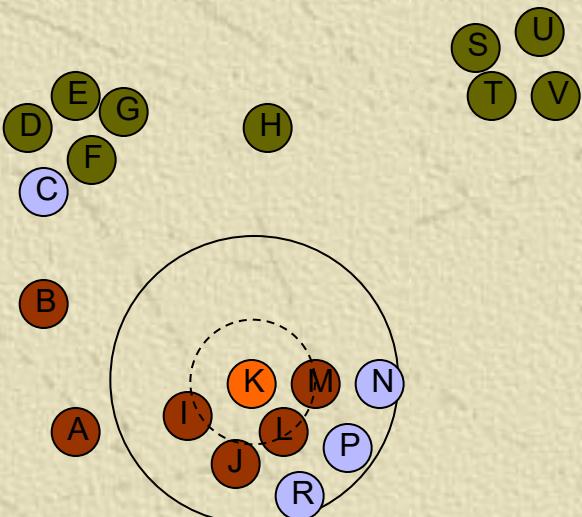
controlList: (M, 18) (K, 18) (R, 20) (P, 21) (N, 35) (C, 40)

Computing a Cluster Ordering - Example



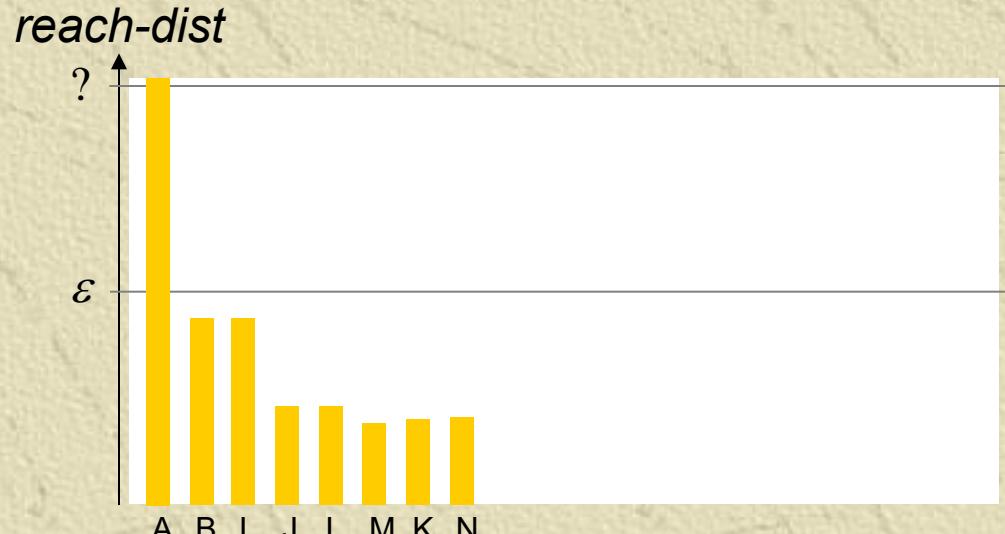
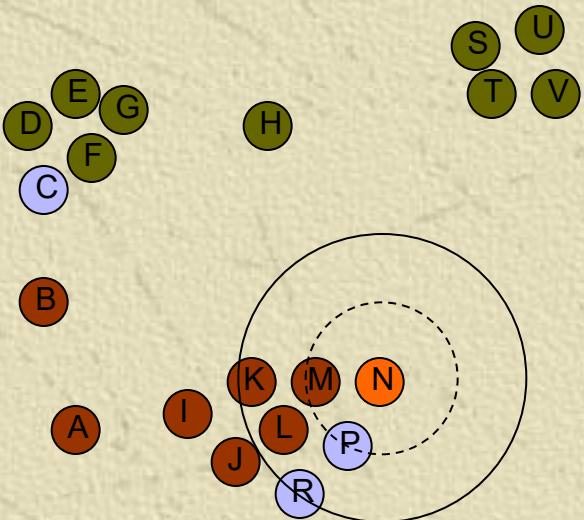
controlList: (K, 18) (N, 19) (R, 20) (P, 21) (C, 40)

Computing a Cluster Ordering - Example



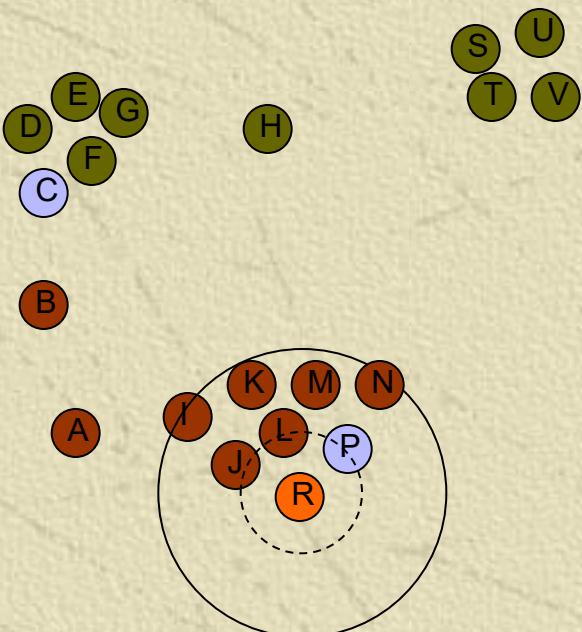
controlList: (N, 19) (R, 20) (P, 21) (C, 40)

Computing a Cluster Ordering - Example

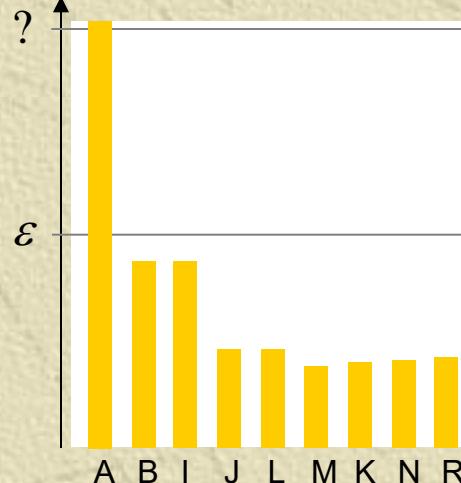


controlList: (R, 20) (P, 21) (C, 40)

Computing a Cluster Ordering - Example

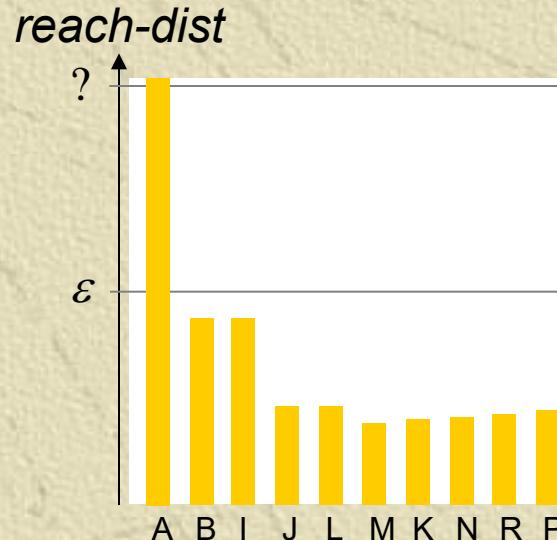
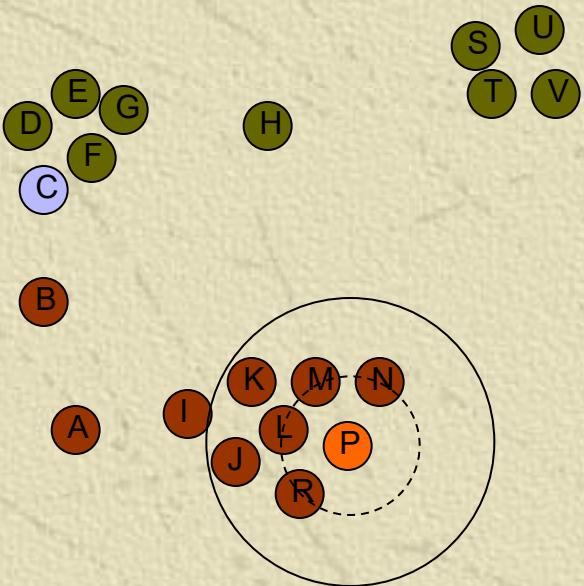


reach-dist



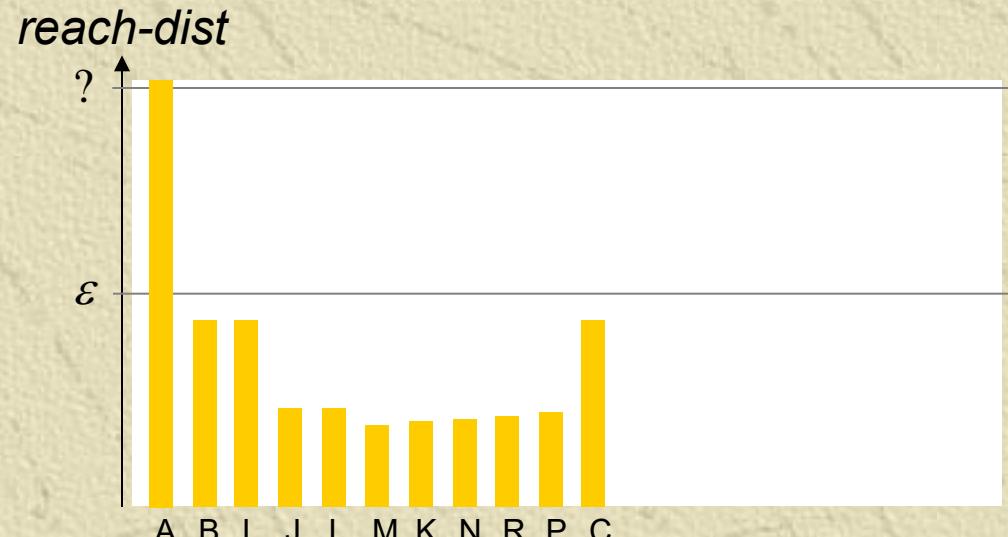
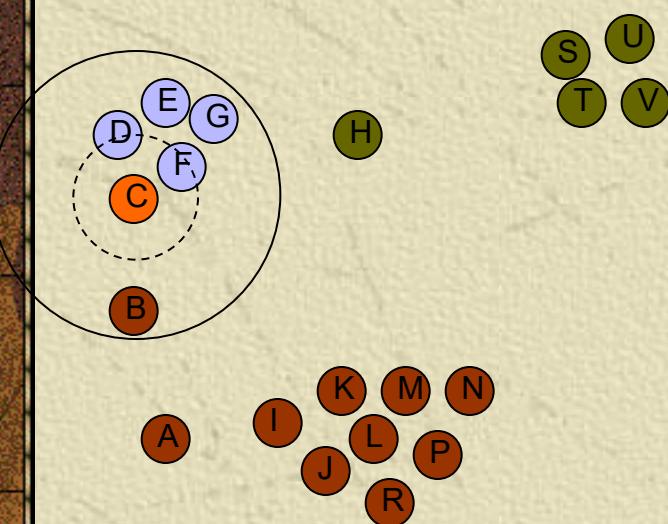
controlList: (P, 21) (C, 40)

Computing a Cluster Ordering - Example



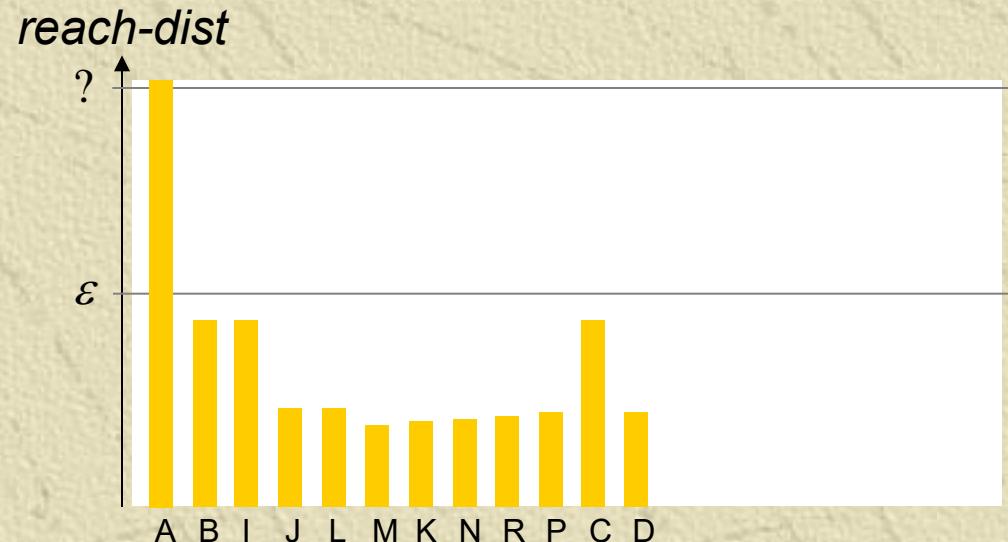
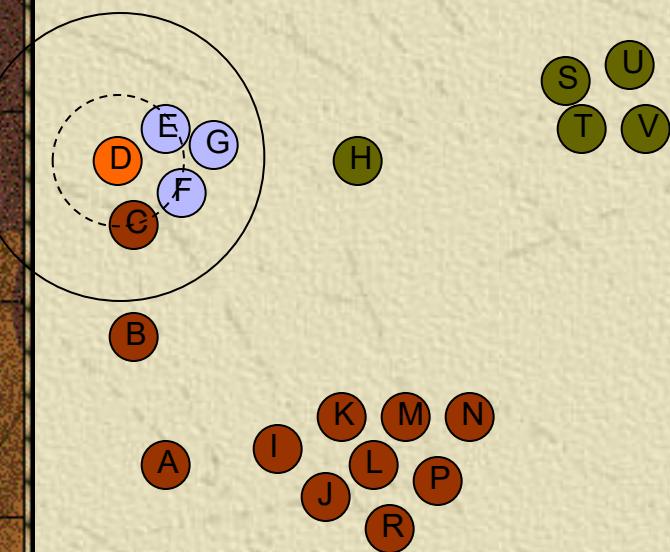
controlList: (C, 40)

Computing a Cluster Ordering - Example



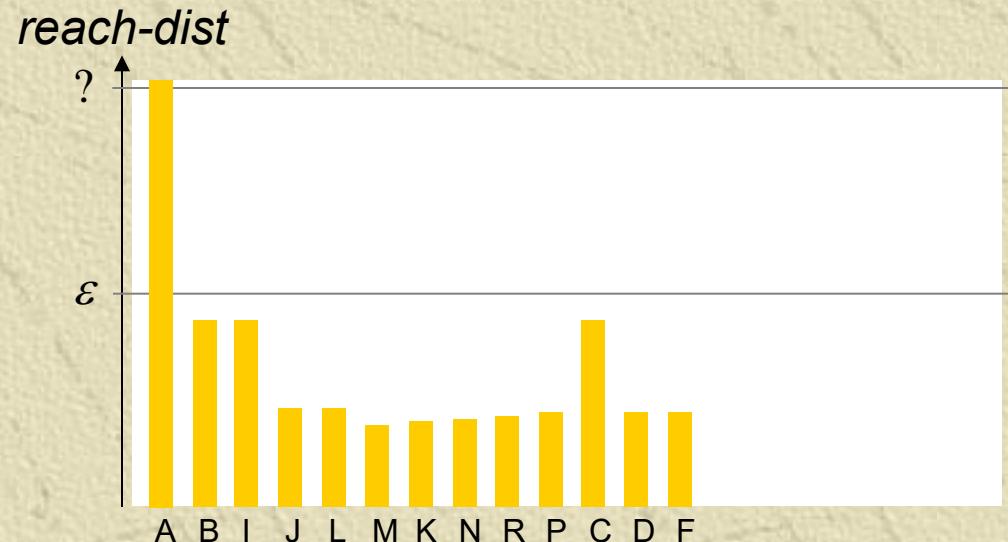
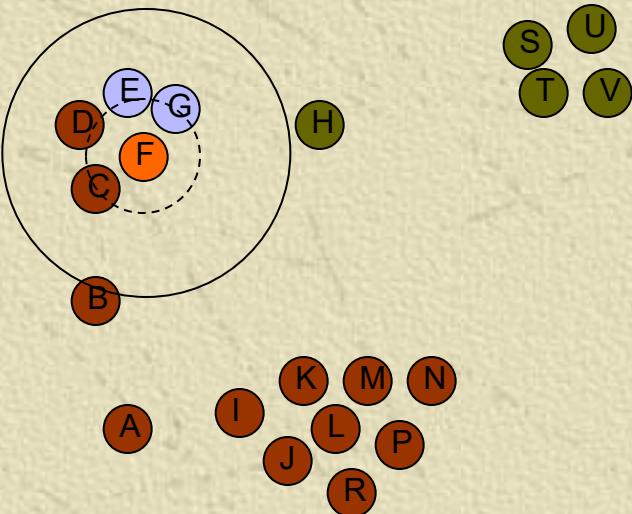
controlList: (D, 22) (F, 22) (E, 30) (G, 35)

Computing a Cluster Ordering - Example



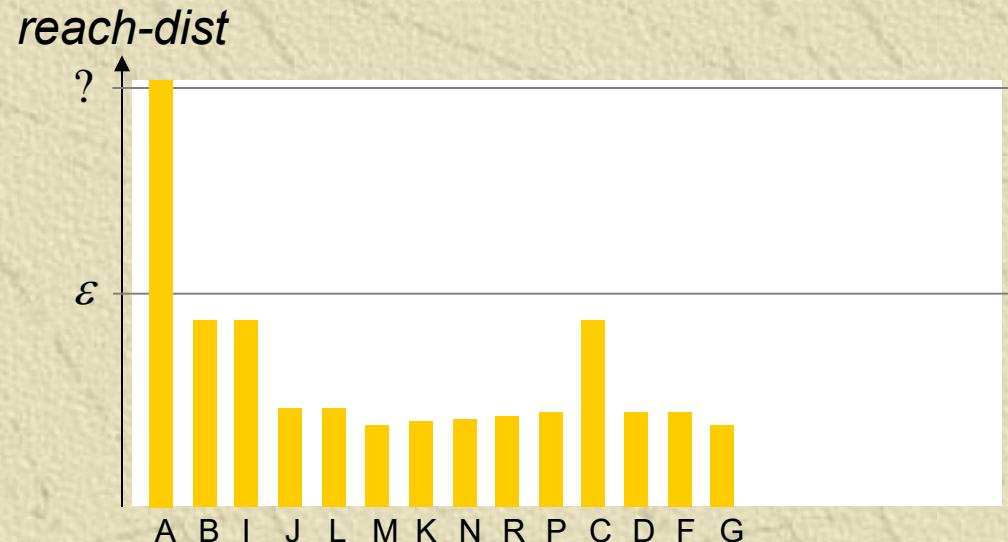
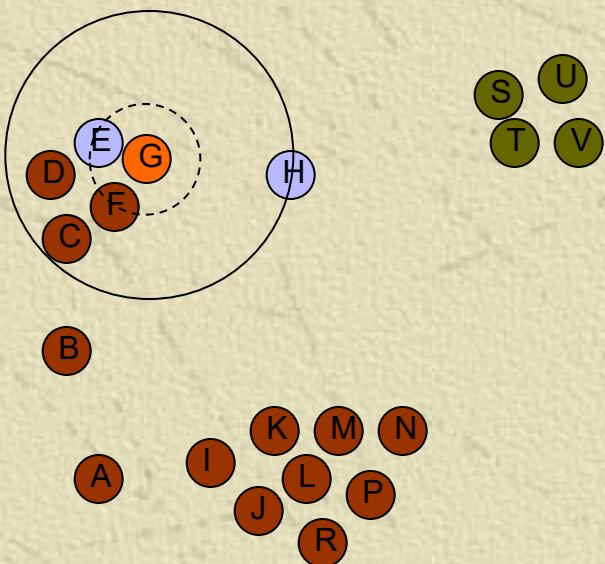
controlList: (F, 22) (E, 22) (G, 32)

Computing a Cluster Ordering - Example



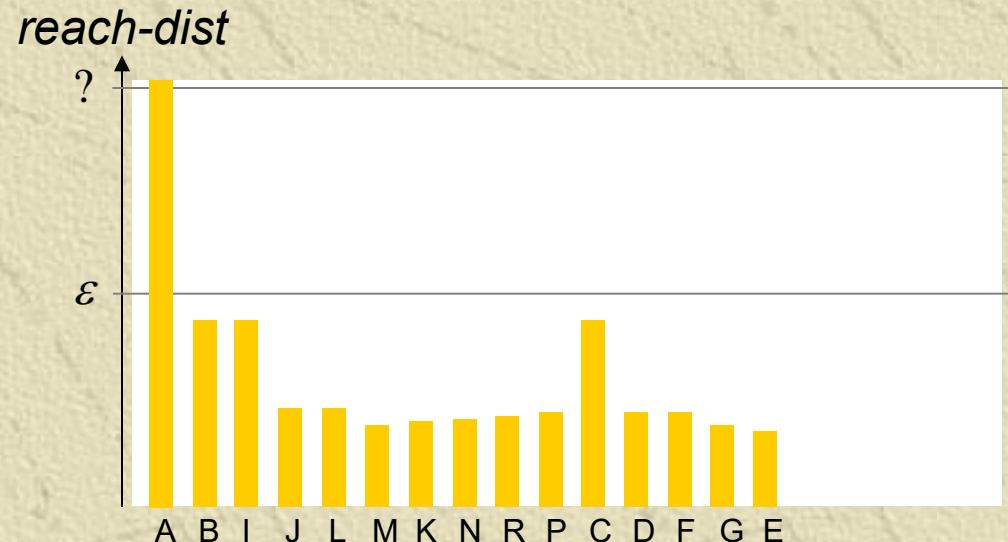
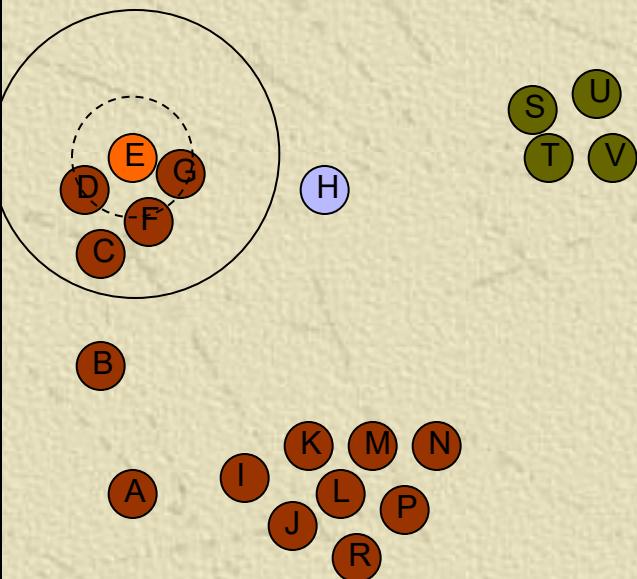
controlList: (G, 17) (E, 22)

Computing a Cluster Ordering - Example



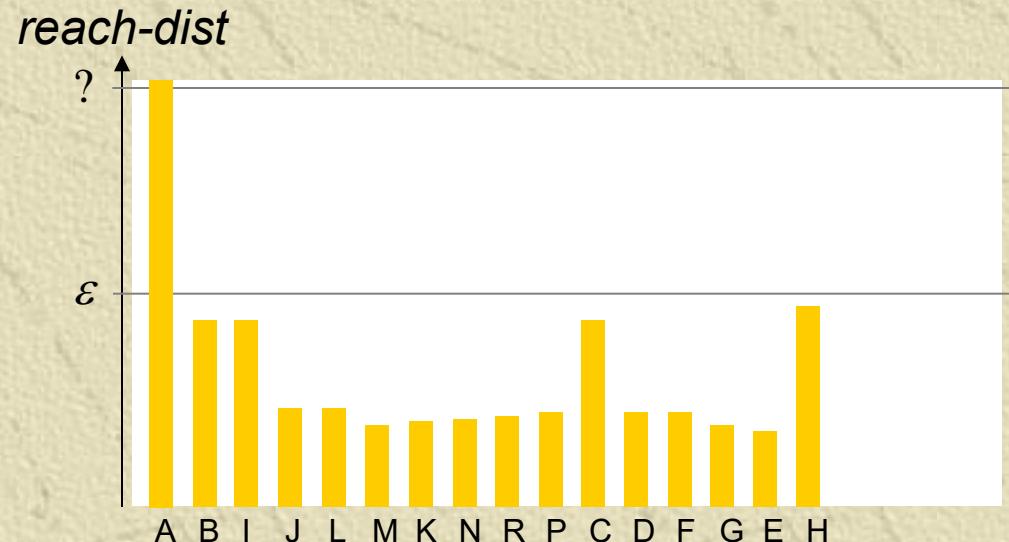
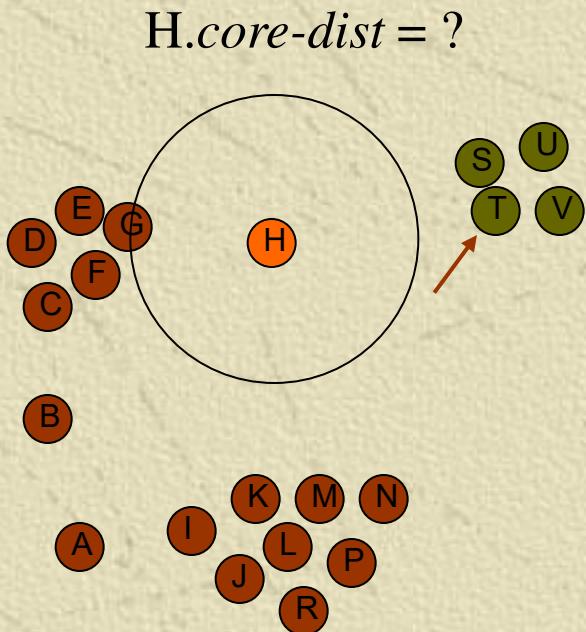
controlList: (E, 15) (H, 43)

Computing a Cluster Ordering - Example



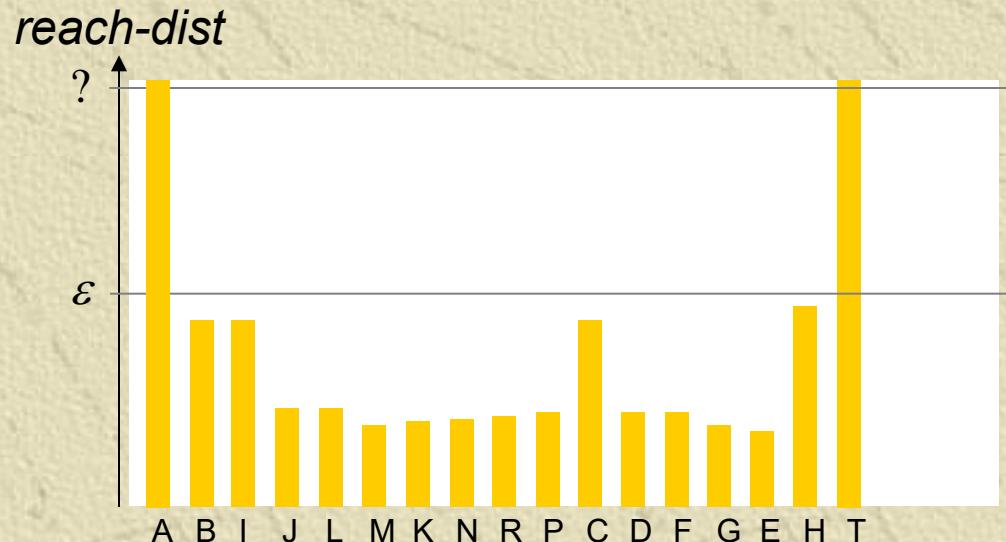
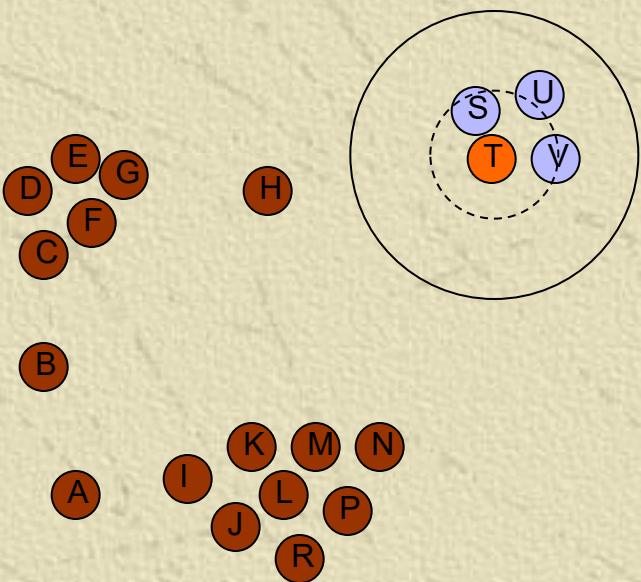
controlList: (H, 43)

Computing a Cluster Ordering - Example



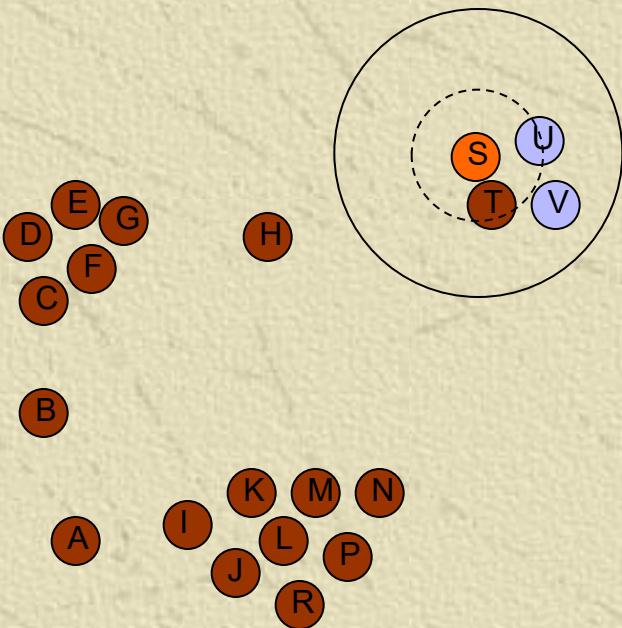
controlList: (T, ?)

Computing a Cluster Ordering - Example

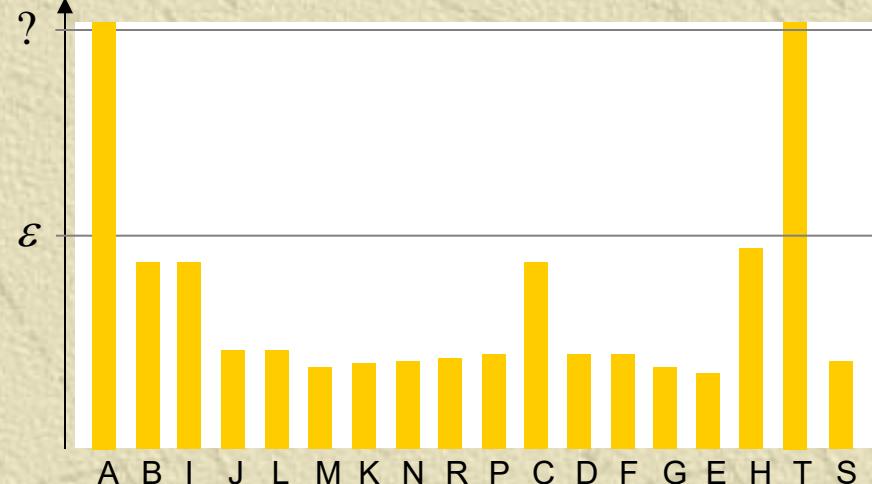


controlList: (S, 18) (V, 18) (U, 25)

Computing a Cluster Ordering - Example

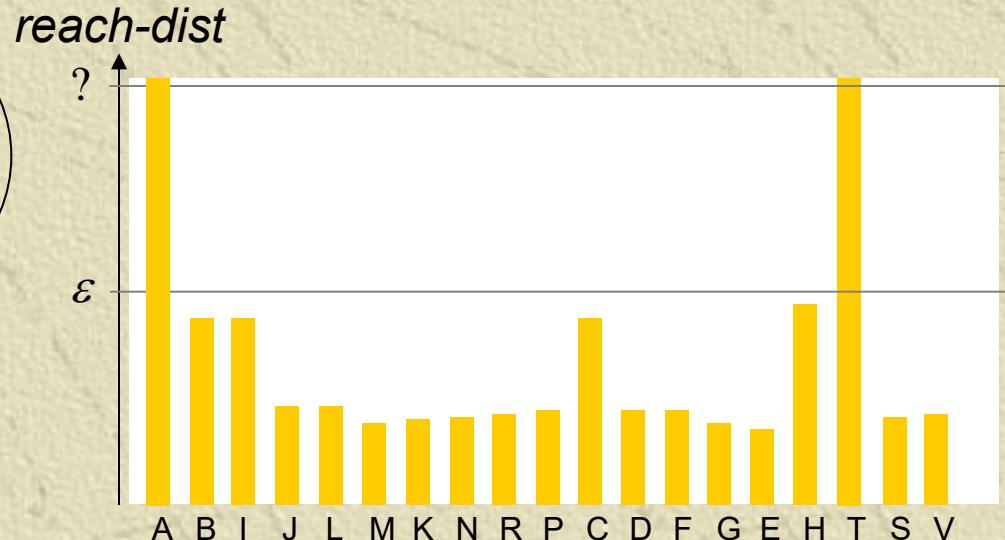
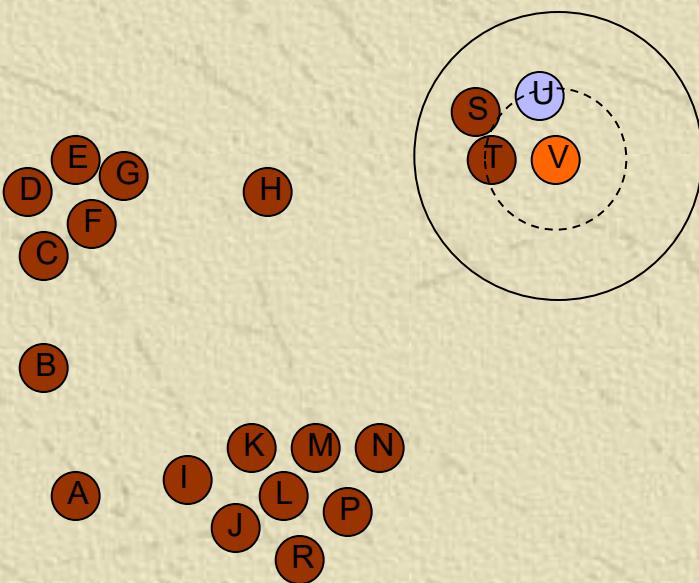


reach-dist



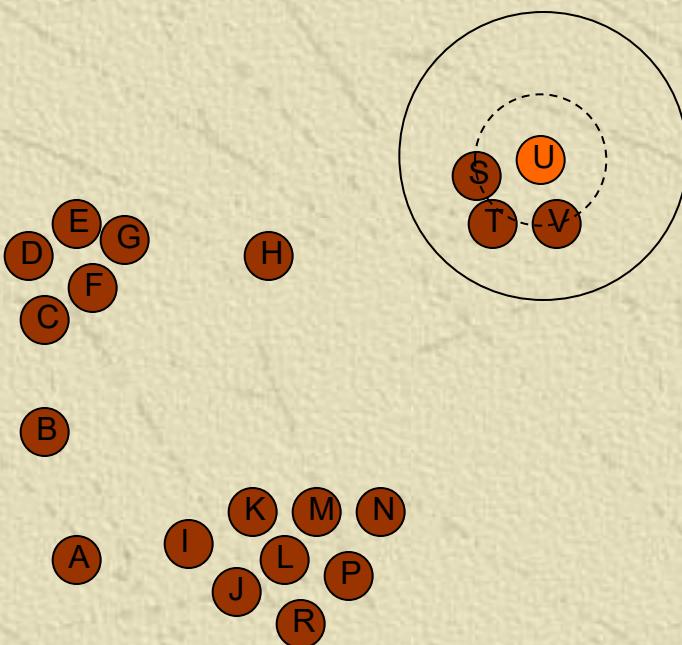
controlList: (V, 18) (U, 20)

Computing a Cluster Ordering - Example



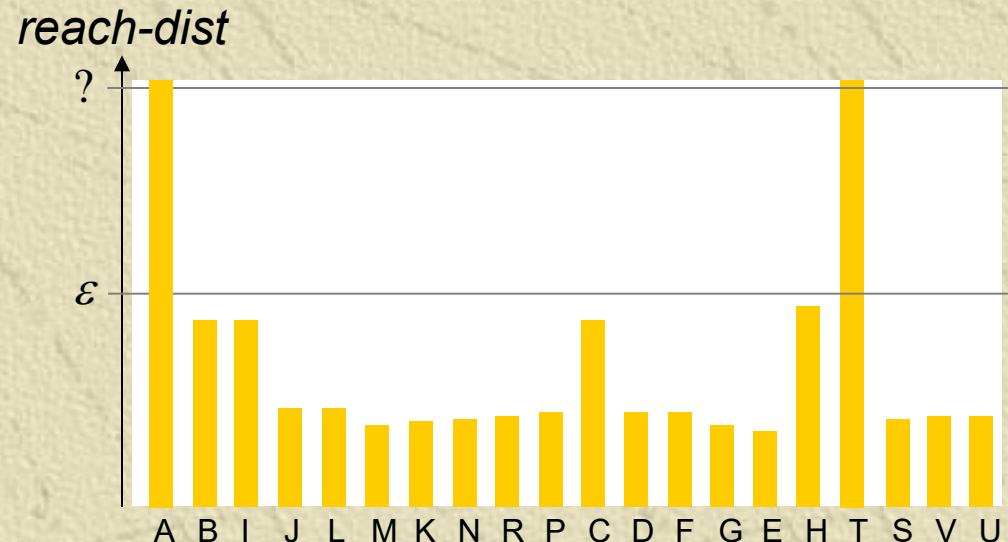
controlList: (U, 19)

Computing a Cluster Ordering - Example



controlList: -

Computing a Cluster Ordering - Example



controlList: -



BITS Pilani

Pilani | Dubai | Goa | Hyderabad

S2-21_DSECLZC415: Data Mining (Lecture #12 – Cluster Analysis)



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Data Mining

Cluster Analysis

Types of Clusterings

Partitional Clustering

- A division of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Divisions can be
 - Distance based
 - Density based

Hierarchical clustering

- A set of nested clusters organized as a hierarchical tree

Comparison of DBSCAN and K-means

Both are partitional.

K-means is complete; DBSCAN is not.

K-means has a prototype-based notion of a cluster; DBSCAN uses a density-based notion.

K-means can find clusters that are not well-separated. DBSCAN will merge clusters that touch.

DBSCAN handles clusters of different shapes and sizes; K-means prefers globular clusters.

DBSCAN can handle noise and outliers; K-means performs poorly in the presence of outliers

Comparison of DBSCAN and K-means

K-means can only be applied to data for which a centroid is meaningful; DBSCAN requires a meaningful definition of density

Both techniques were designed for Euclidean data, but extended to other types of data

K-means has an $O(n)$ time complexity; DBSCAN is $O(n^2)$

Because of random initialization, the clusters found by K-means can vary from one run to another; DBSCAN always produces the same clusters

DBSCAN automatically determines the number of clusters; K-means does not

K-means has only one parameter, DBSCAN has two.



OPTICS

Challenges with DBSCAN

- DBSCAN requires users with the responsibility of selecting parameter values for the discovery of acceptable clusters.
 - Eps (the maximum radius of a neighborhood) and
 - MinPts (the minimum number of points required in the neighborhood of a core object)
- The parameters are empirically set and difficult to determine especially for real-world, high dimensional data sets.
- Sensitive to these parameter values: Slightly different settings may lead to very different clustering of the data.
- The real-world, high-dimensional data can have skewed distributions such that their intrinsic clustering structure may not be well characterized by a single set of global density parameters.

OPTICS: A Cluster-Ordering Method (1999)

OPTICS: Ordering Points To Identify the Clustering Structure

- Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)
- Produces a special order of the database wrt its density-based clustering structure
- This cluster-ordering contains info equivalent to the density-based clusterings corresponding to a broad range of parameter settings
- Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
- Can be represented graphically or using visualization techniques

OPTICS algorithm

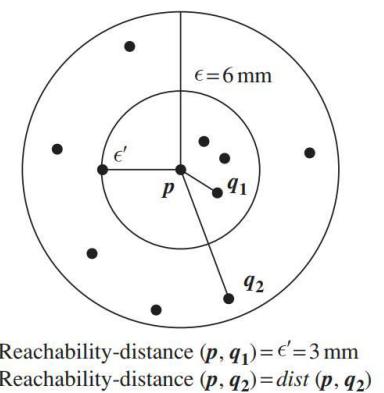
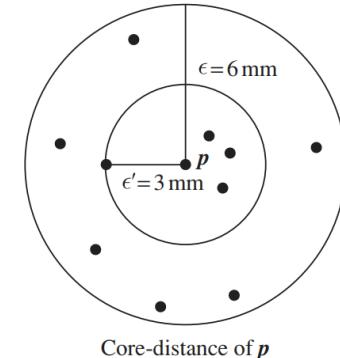
Ordering points to identify the clustering structure (OPTICS) is an algorithm for finding density-based clusters

OPTICS overcomes the (DBSCAN) problem of detecting meaningful clusters in data of varying density

Points of the dataset are (linearly) ordered such that spatially closest points become neighbors. A special distance is plotted for each point

OPTICS: Some Extension from DBSCAN

- Index-based:
 - k: # of dimensions
 - N: # of points
 - Complexity: $O(N * \log N)$
- OPTICS adds two more terms to the concepts of DBSCAN clustering. They are:-
 - **Core Distance:** It is the minimum value of radius required to classify a given point as a core point. If the given point is not a Core point, then its Core Distance is undefined.
 - **Reachability Distance:** It is defined with respect to another data point q. The Reachability distance between a point p and q is the maximum of the Core Distance of p and the Euclidean Distance(or some other distance metric) between p and q. Note that the Reachability Distance is not defined if q is not a Core point.



OPTICS algorithm

- The core-distance of an object p is the smallest value Eps' such that
 - Eps'-neighborhood of p has at least MinPts objects.
- That is, Eps' is the minimum distance threshold that makes p a core object.
- If p is not a core object with respect to Eps and MinPts, the core-distance of p is undefined.
- The reachability-distance to object p from q is the minimum radius value that makes p density-reachable from q.
 - According to the definition of density-reachability, q has to be a core object and p must be in the neighborhood of q.
 - Therefore, the reachability-distance from q to p is $\max(\text{core-distance}(q), \text{dist}(p, q))$. If q is not a core object with respect to Eps and MinPts, the reachability-distance to p from q is undefined.

OPTICS: Some Extension from DBSCAN

Complexity: $O(N \log N)$

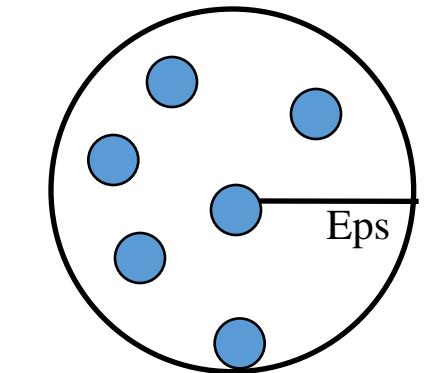
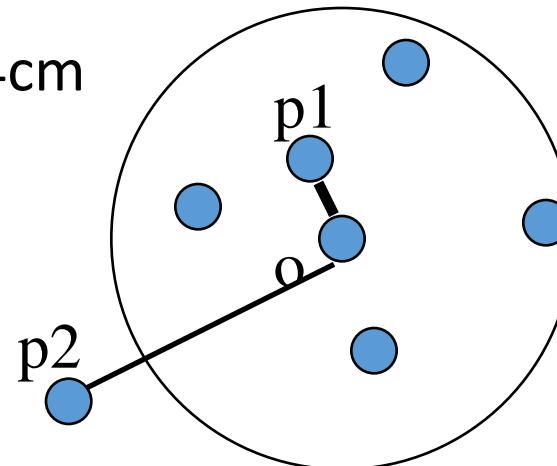
Core Distance:

- min ϵ s.t. point is core

Reachability Distance

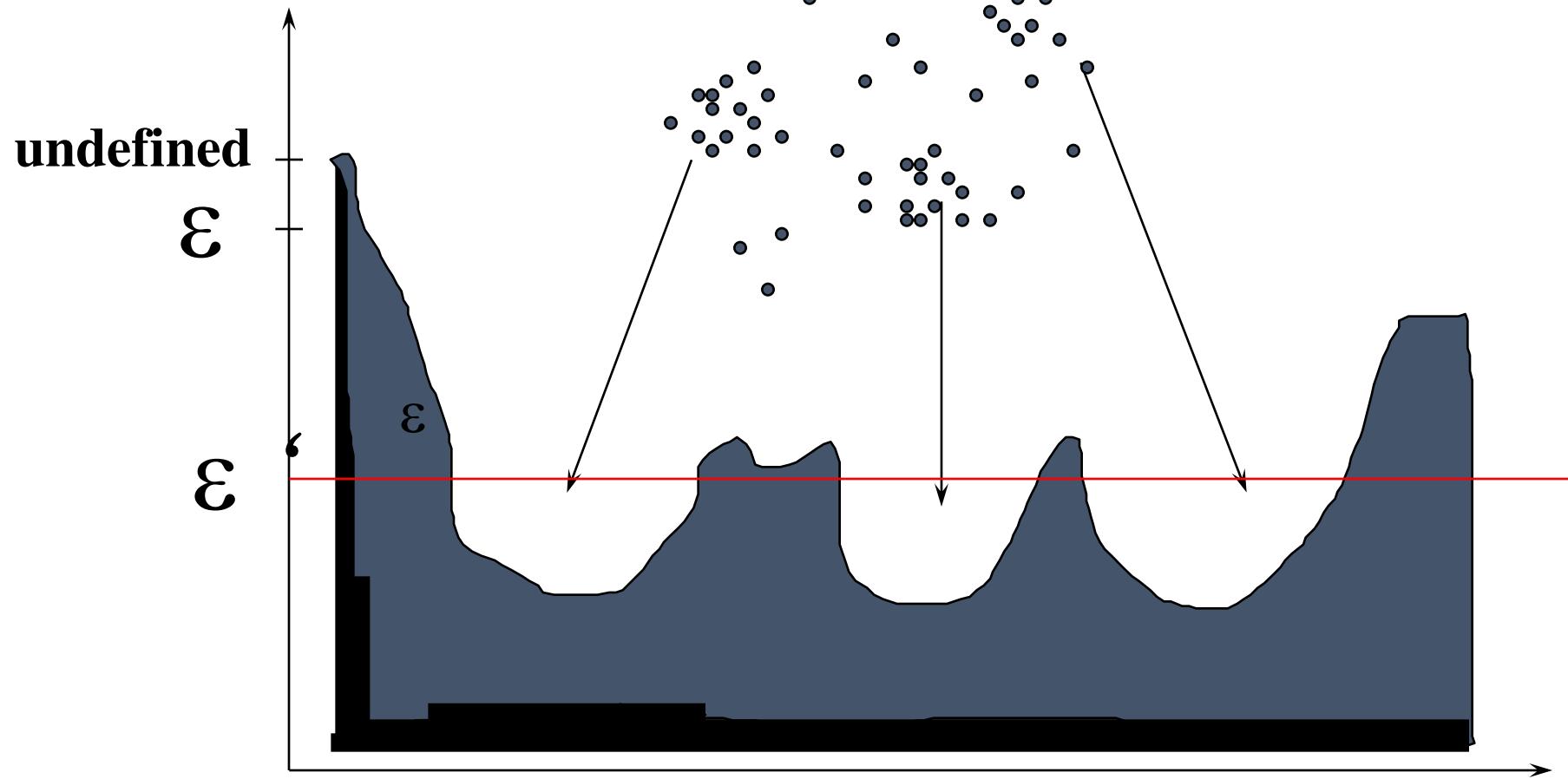
$\text{Max}(\text{core-distance}(o), d(o, p))$

$$r(p_1, o) = 3\text{cm. } r(p_2, o) = 4\text{cm}$$



$$\begin{aligned} \text{MinPts} &= 5 \\ \epsilon &= 3 \text{ cm} \end{aligned}$$

Reachability-distance



Cluster-order of the objects



Extensions to Hierarchical Clustering

Limitations to Hierarchical Clustering

- Major weakness of agglomerative clustering methods
 - Can never undo what was done previously
 - Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
- Integration of hierarchical & distance-based clustering
 - BIRCH: uses CF-tree and incrementally adjusts the quality of sub-clusters

BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies)

- Zhang, Ramakrishnan & Livny, SIGMOD'96
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
 - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
 - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness*: handles only numeric data, and sensitive to the order of the data record

Clustering Feature Vector in BIRCH

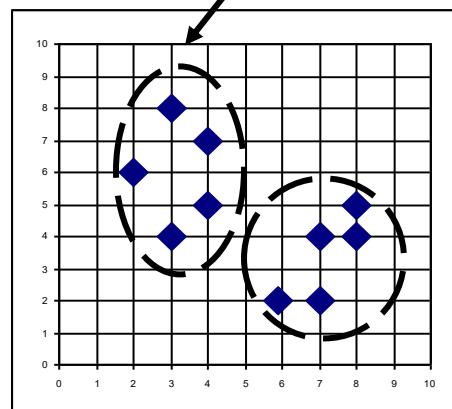
Clustering Feature (CF): $CF = (N, LS, SS)$

N : Number of data points

LS : linear sum of N points: $\sum_{i=1}^N X_i$

SS : square sum of N points: $\sum_{i=1}^N X_i^2$

$$CF1 = (5, (16,30),(54,190))$$



(3,4)

(2,6)

(4,5)

(4,7)

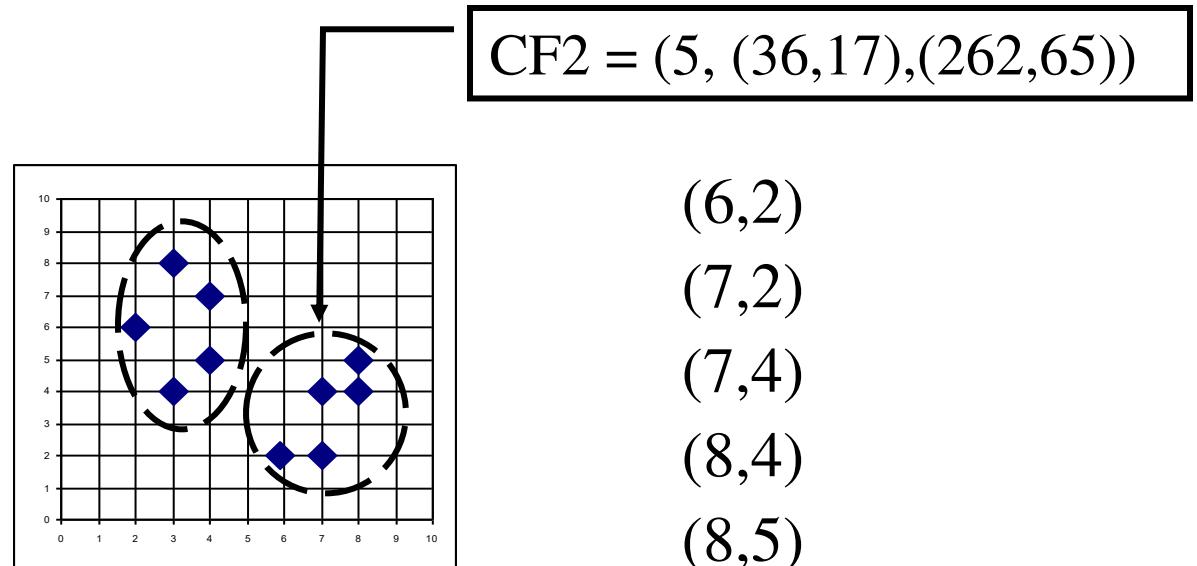
(3,8)

Clustering Feature Vector in BIRCH

$CF1$ (previous slide) = $(5, (16,30), (54, 190))$

If we create CF_{12} by combining $CF1$ and $CF2$,

$$CF_{12} = (5+5, (16+36, 30+17), (54+262, 190+65))$$



CF-Tree in BIRCH

Clustering feature:

- Summary of the statistics for a given subcluster: the 0-th, 1st, and 2nd moments of the subcluster from the statistical point of view
- Registers crucial measurements for computing cluster and utilizes storage efficiently

A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering

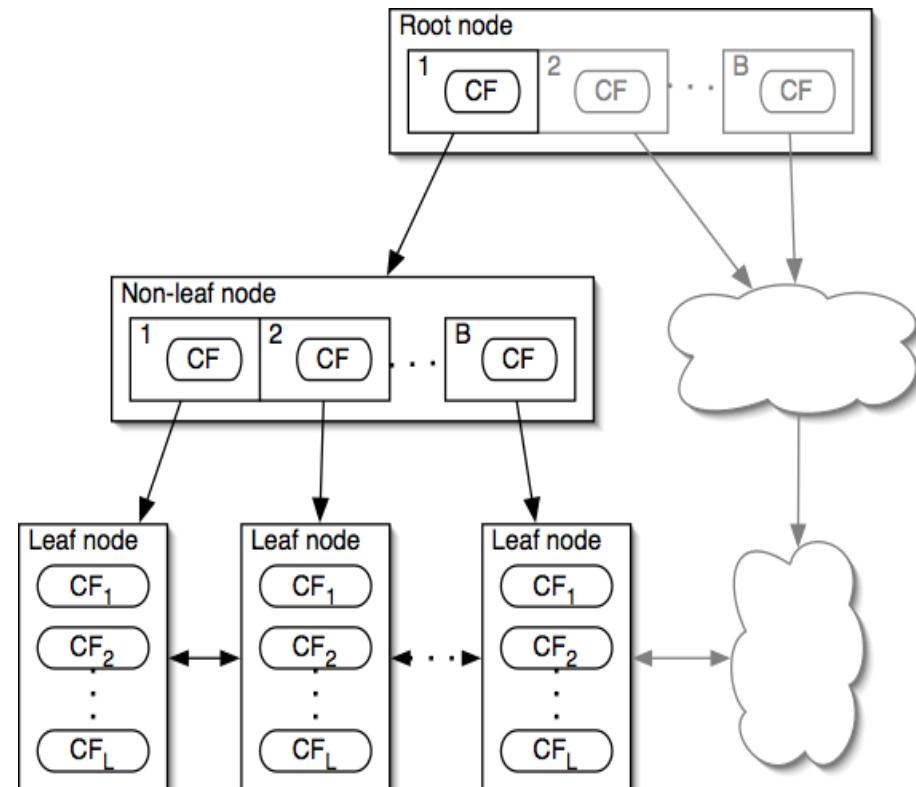
- A nonleaf node in a tree has descendants or “children”
- The nonleaf nodes store sums of the CFs of their children

A CF tree has two parameters

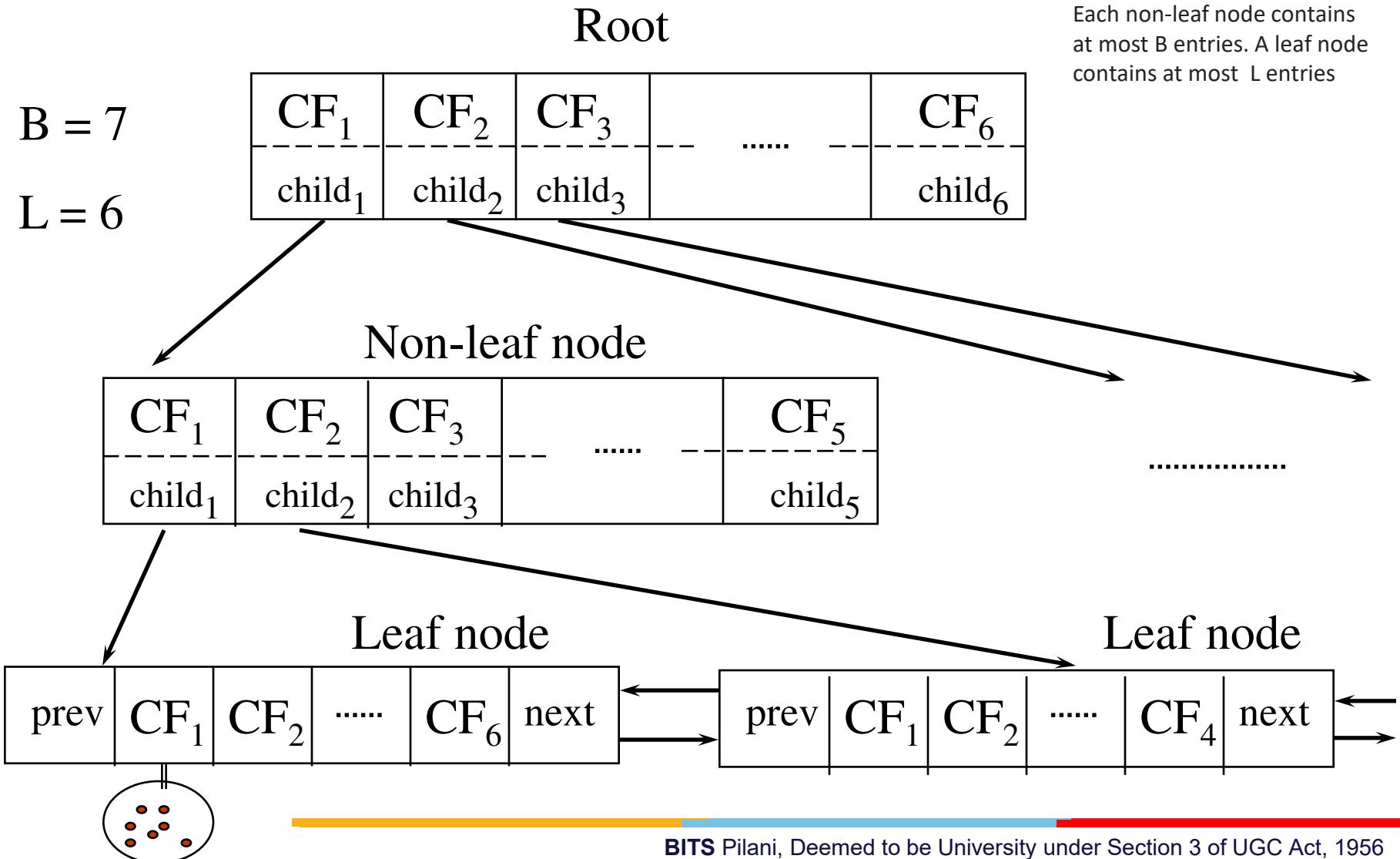
- Branching factor: max # of children
- Threshold: max diameter of sub-clusters stored at the leaf nodes

The CF Tree Structure

- Each non-leaf node has at most **B** entries
- Each leaf node has at most **L** CF entries which each satisfy threshold **T**
- Node size is determined by dimensionality of data space and input parameter **P** (page size)



The CF Tree Structure



BIRCH Steps

The first phase builds a CF tree out of the data points, a height-balanced tree data structure

In the second step, the algorithm scans all the leaf entries in the initial CF tree to rebuild a smaller CF tree, while removing outliers and grouping crowded subclusters into larger ones.

In step three an existing clustering algorithm is used to cluster all leaf entries. Here algorithm is applied directly to the subclusters represented by their CF vectors.

In (optional) step 4 the centroids of the clusters produced in step 3 are used as seeds and redistribute the data points to its closest seeds to obtain a new set of clusters

The Birch Algorithm

Cluster Diameter

$$\sqrt{\frac{1}{n(n-1)} \sum (x_i - x_j)^2}$$

In first phase, For each point in the input

- Find closest leaf entry
- Add point to leaf entry and update CF
- If entry diameter > max_diameter, then split leaf, and possibly parents

Algorithm is O(n)

Concerns

- Sensitive to insertion order of data points
- Since we fix the size of leaf nodes, so clusters may not be so natural
- Clusters tend to be spherical given the radius and diameter measures

Centroid, Radius and Diameter of a Cluster

- Centroid: the “middle” of a cluster

$$C = \frac{\sum_{i=1}^N (x_i)}{N} = \frac{LS}{n}$$

- Radius: square root of average distance from any point of the cluster to its centroid

$$R = \sqrt{\frac{\sum_{i=1}^N (x_i - c)^2}{N}}$$

$$R = \sqrt{\frac{nSS - LS^2}{n^2}}$$

- Diameter: square root of average mean squared distance between all pairs of points in the cluster

$$D = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^N (x_i - x_j)^2}{N(N - 1)}} = \sqrt{\frac{2nSS - 2LS^2}{n(n - 1)}}$$



Cluster Validation

Cluster Validity

For supervised classification we have a variety of measures to evaluate how good our model is

- Accuracy, precision, recall

For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

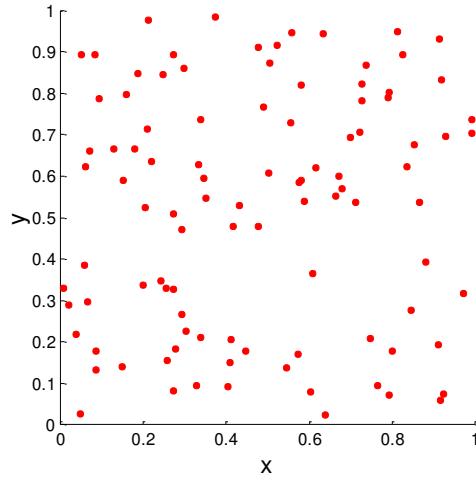
But “clusters are in the eye of the beholder”!

Then why do we want to evaluate them?

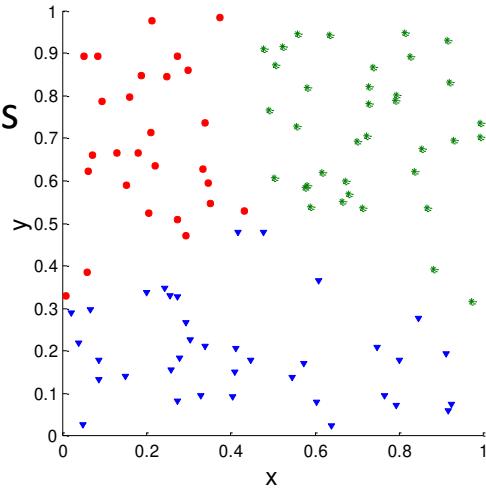
- To avoid finding patterns in noise
- To compare clustering algorithms
- To compare two sets of clusters
- To compare two clusters

Clusters found in Random Data

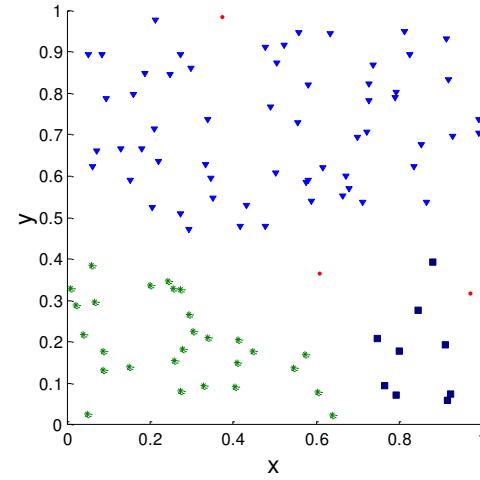
Random Points



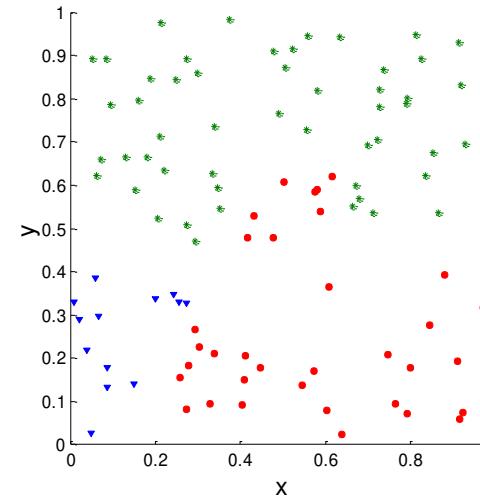
K-means



DBSCAN



Complete Link



Different Aspects of Cluster Validation

1. Determining the **clustering tendency** of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.
 - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the ‘correct’ number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Clustering Tendency

All clustering algorithms find some clusters, whether data has natural clusters or not.

We need a mechanism to check if at least some clusters are of good quality

Alternatively, we can directly check the data for clustering tendency. A common approach is to use statistical tests for spatial randomness (in Euclidean space) among data points.

Hopkins Statistic: Use p points from data and generate a set of p random points in data space. Let u_i be nearest neighbor distances in generated data and w_i be nearest neighbor distances in supplied data. Hopkin statistic H is defined as

$$H = \frac{\sum_{i=1}^p w_i}{\sum_{i=1}^p u_i + \sum_{i=1}^p w_i}$$

H is closer to 0 => data is highly clustered;

H is closer to 0.5 => data is uniformly distributed in data space

Evaluation measures for Cluster Validity

Unsupervised Measures without resort to external information, also referred *internal indices* e.g. SSE.

Measures can be

- Cluster Cohesion (compactness, tightness) - how closely related the objects in a cluster are?
- Cluster Separation (isolation) - how distinct or well-separated a cluster is from other clusters

Supervised Measures match the clustering output to some external structure, also referred *external indices* e.g. entropy

- Measures how well cluster labels match externally supplied class labels.

Relative - compare different clustering outcomes. can be based on supervised or unsupervised measures, e.g. two K-means clustering outputs can be compared using either SSE or entropy.

Internal Measures: SSE

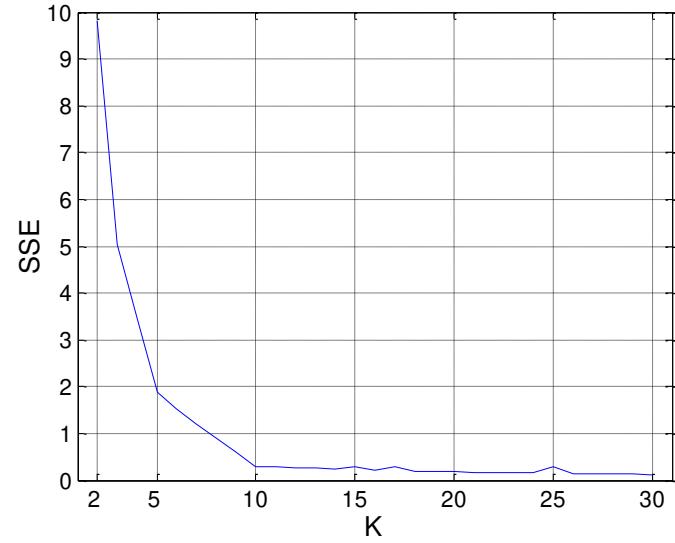
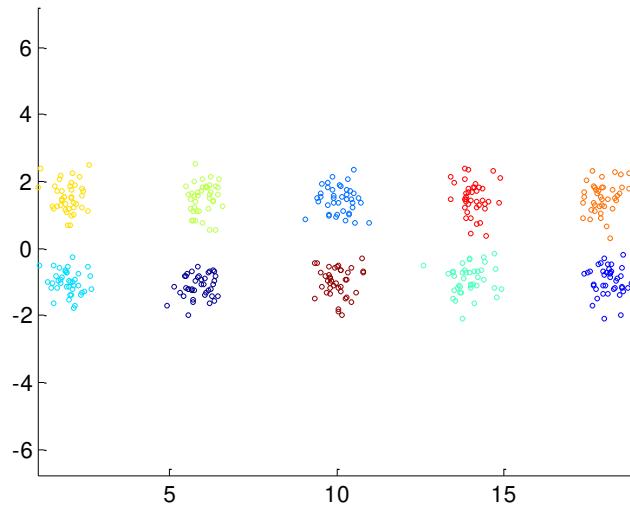
Clusters in more complicated figures aren't well separated

Internal Index: Used to measure the goodness of a clustering structure without respect to external information

- SSE

SSE is good for comparing two clusterings or two clusters (average SSE).

Can also be used to estimate the number of clusters



Internal Measures: Cohesion and Separation

Cluster Cohesion: Measures how closely related are objects in a cluster

- Example: SSE

Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters

- Example: Squared Error
- Cohesion is measured by the within cluster sum of squares (SSE)

$$WSS = \sum_i \sum_{x \in C_i} (x - c_i)^2$$

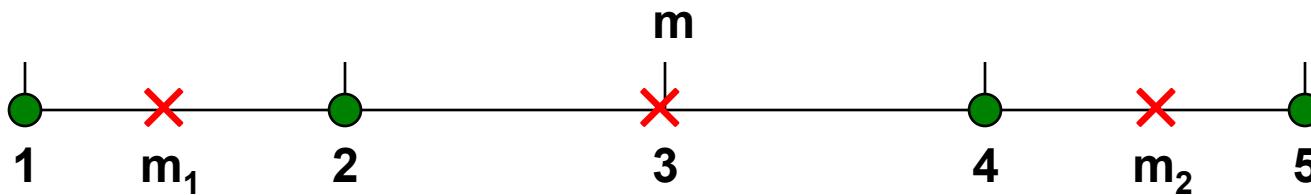
- Separation is measured by the between cluster sum of squares

$$BSS = \sum_i |C_i| (c - c_i)^2$$

- Where $|C_i|$ is the size of cluster i

Internal Measures: Cohesion and Separation

- Example: SSE
 - BSS + WSS = constant



K=1 cluster:

$$SSE = WSS = (1 - 3)^2 + (2 - 3)^2 + (4 - 3)^2 + (5 - 3)^2 = 10$$

$$BSS = 4 \times (3 - 3)^2 = 0$$

$$Total = 10 + 0 = 10$$

K=2 clusters:

$$SSE = WSS = (1 - 1.5)^2 + (2 - 1.5)^2 + (4 - 4.5)^2 + (5 - 4.5)^2 = 1$$

$$BSS = 2 \times (3 - 1.5)^2 + 2 \times (4.5 - 3)^2 = 9$$

$$Total = 1 + 9 = 10$$

Measuring Clustering Quality: Extrinsic Methods

Clustering quality measure: $Q(C, C_g)$, for a clustering C given the ground truth C_g .

Q is good if it satisfies the following **4** essential criteria

- Cluster homogeneity: the purer, the better
- Cluster completeness: should assign objects belong to the same category in the ground truth to the same cluster
- Rag bag: putting a heterogeneous object into a pure cluster should be penalized more than putting it into a *rag bag* (i.e., “miscellaneous” or “other” category)
- Small cluster preservation: splitting a small category into pieces is more harmful than splitting a large category into pieces

Measuring Clustering Quality

Two methods: extrinsic vs. intrinsic

Extrinsic: supervised, i.e., the ground truth is available

- Compare a clustering against the ground truth using certain clustering quality measure
- Ex. precision and recall metrics (averaged over all classes)

Intrinsic: unsupervised, i.e., the ground truth is unavailable

- Evaluate the goodness of a clustering by considering how well the clusters are separated, and how compact the clusters are
- Ex. Silhouette coefficient

Classification-based Measures of Cluster Validity

Precision: The fraction of a cluster that consists of objects of a specified class. The precision of cluster i with respect to a class j is
 $\text{precision}(i,j) = p_{ij}$

Recall: The extent to which a cluster contains all objects of a specified class. The recall of cluster i with respect to class j is $\text{recall}(i, j) = m_{ij}/m_j$ where m_j is the number of objects in class j.

F-measure: A combination of both precision and recall that measures the extent to which a cluster contains only objects of a particular class and all objects of that class. The F-measure of cluster i with respect to class j is

$$F(i,j) = (2 * \text{precision}(i,j) * \text{recall}(i,j)) / (\text{precision}(i,j) + \text{recall}(i,j))$$

External Measures of Cluster Validity: Precision/Recall

Cluster	Entertainm ent	Financial	Foreign	Metro	National	Sports	Totals
1	3	5	40	506	96	27	677
2	4	7	280	29	39	2	361
3	1	1	1	7	4	671	685
4	10	162	3	119	73	2	369
5	331	22	5	70	13	23	464
6	5	358	12	212	48	13	648
Total	354	555	341	943	273	738	

K-means clustering result for a newspaper articles document data set

For Cluster 1 and Metro class,
 Precision = $506/677 = 0.75$
 Recall = $506/943 = 0.54$
 F-value = 0.63

For Cluster 3 and Sports class,
 Precision = $671/685 = 0.98$
 Recall = $671/738 = 0.91$
 F-value = 0.94

Measuring Cluster Validity Via Correlation

- Two matrices
 - Proximity Matrix
 - Ideal Similarity Matrix
 - ◆ One row and one column for each data point
 - ◆ An entry is 1 if the associated pair of points belong to the same cluster
 - ◆ An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
 - Since the matrices are symmetric, only the correlation between $n(n-1) / 2$ entries needs to be calculated.
- High correlation indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Determine the Number of Clusters

Empirical method

- # of clusters $\approx \sqrt{(n/2)}$ for a dataset of n points

Elbow method

- Use the turning point in the curve of sum of within cluster variance w.r.t the # of clusters

Cross validation method

- Divide a given data set into m parts
- Use $m - 1$ parts to obtain a clustering model
- Use the remaining part to test the quality of the clustering
 - E.g., For each point in the test set, find the closest centroid, and use the sum of squared distance between all points in the test set and the closest centroids to measure how well the model fits the test set
- For any $k > 0$, repeat it m times, compare the overall quality measure w.r.t. different k 's, and find # of clusters that fits the data the best

Final Comment on Cluster Validity

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

Algorithms for Clustering Data, Jain and Dubes

Prescribed Text Books

Author(s), Title, Edition, Publishing House	
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers



BITS Pilani

Pilani | Dubai | Goa | Hyderabad

S2-21_DSECLZC415: Data Mining (Lecture #13 – Outlier Analysis)



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Data Mining

Outlier Analysis

What Are Outliers/Anomalies?

In anomaly detection, the goal is to find objects that are different from most other objects.

Outlier: A data object that **deviates significantly** from the normal objects as if it were **generated by a different mechanism**

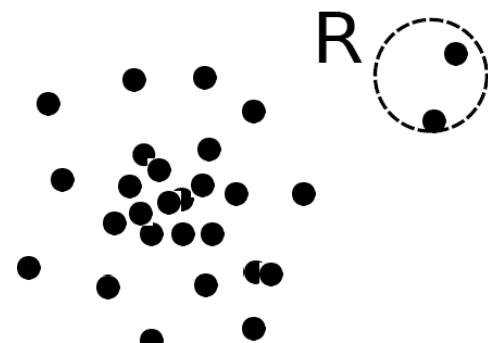
Outliers are different from the noise data

- Noise is random error or variance in a measured variable
- Noise should be removed before outlier detection

Outliers are interesting: It violates the mechanism that generates the normal data

Applications:

- Credit card fraud detection
- Telecom fraud detection
- Customer segmentation
- Medical analysis



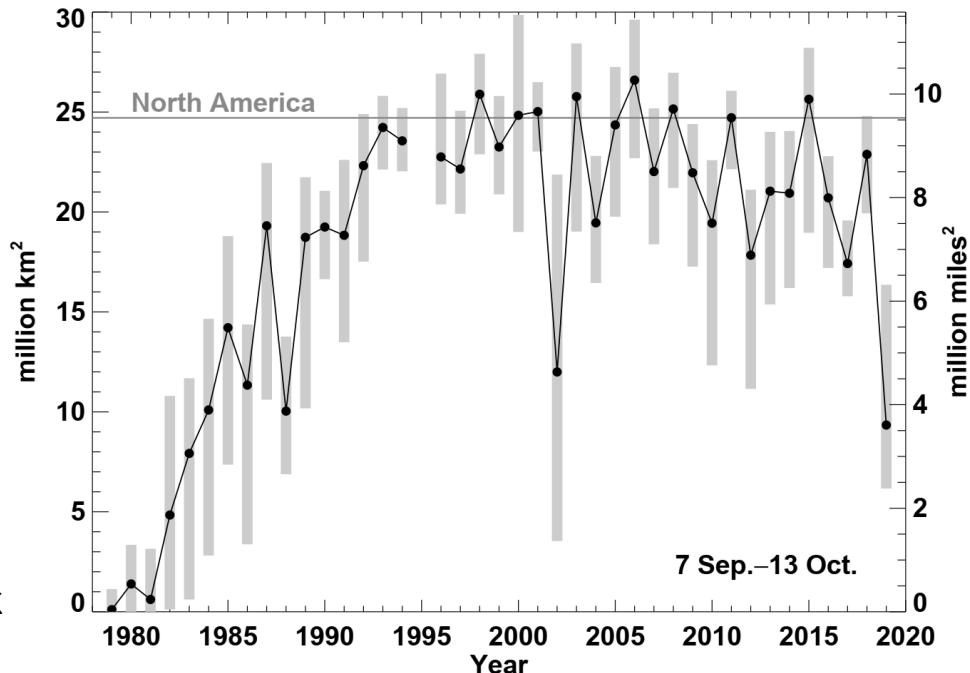
Importance of Anomaly Detection

Ozone Depletion History

In 1977 three researchers (Farman, Gardinar and Shanklin) were puzzled by data gathered by the British Antarctic Survey showing that ozone levels for Antarctica had dropped 10% below normal levels

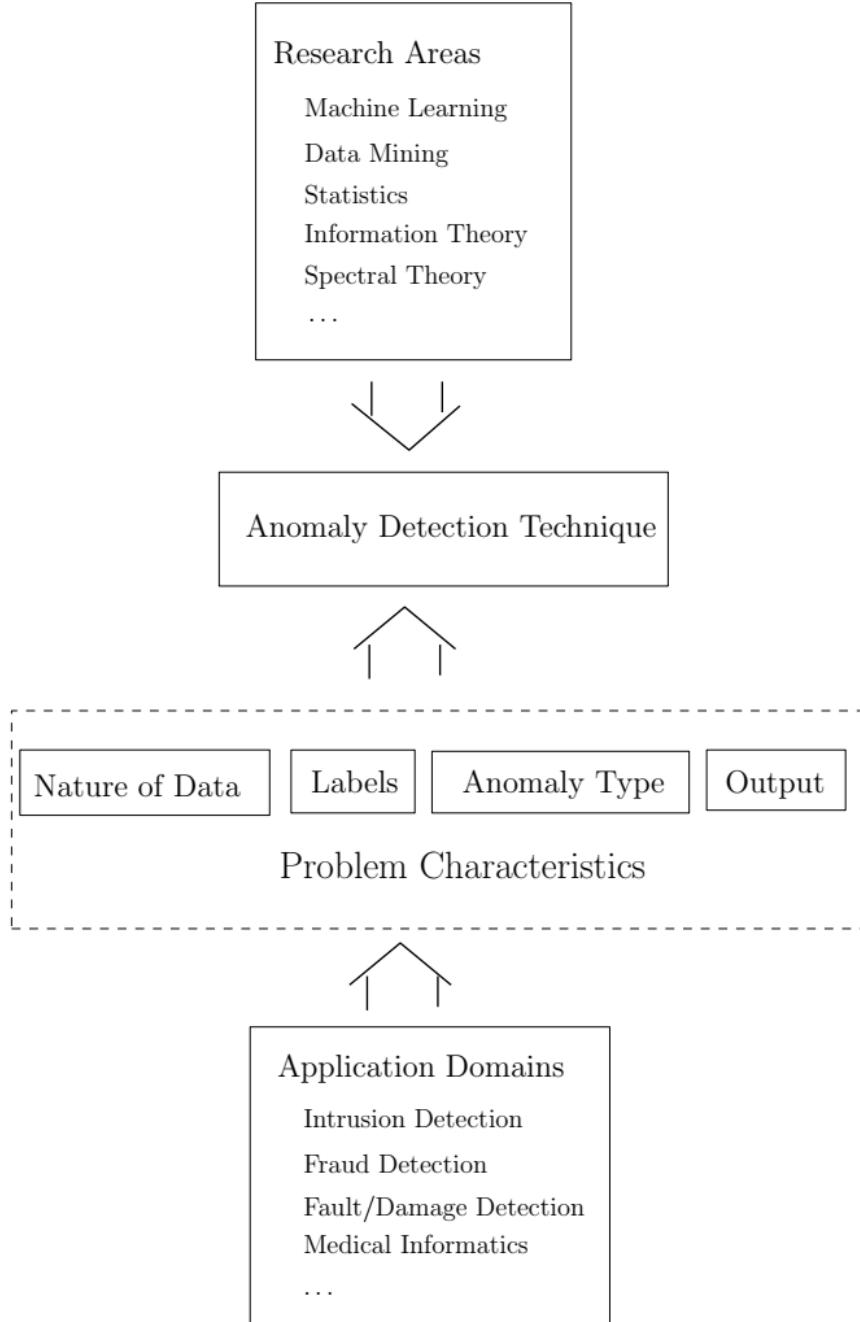
Why did the Nimbus 7 satellite, which had instruments aboard for recording ozone levels, not record similarly low ozone concentrations? The researchers held back publishing their work for nearly a decade.

The ozone concentrations recorded by the satellite were so low they were being treated as outliers by a computer program and discarded!



Sources:

"Cosmic Imagery: Key Images in the History of Science" By John D. Barrow
<http://www3.epa.gov/ozone/science>
<https://ozonewatch.gsfc.nasa.gov/>



Key components associated with an anomaly detection technique

Anomaly Detection : A Survey by
 Varun Chandola, Arindam Banerjee
 And Vipin Kumar University of Minnesota
ACM Computing Surveys, September 2009

More on Outlier/Anomaly Detection

Challenges

- How many outliers are there in the data?
- Method is unsupervised (sometimes supervised methods are used)
 - Validation can be quite challenging (just like for clustering)
- Finding needle in a haystack

Working assumption:

- There are considerably more “normal” observations than “abnormal” observations (outliers/anomalies) in the data

Outlier detection vs. *novelty detection* (identify new topics and trends in a timely manner in social media): early stage, outlier; but later merged into the model

Types of Outliers

Three kinds: *global*, *contextual* and *collective* outliers

Global outlier (or point anomaly)

- Object is O_g if it significantly deviates from the rest of the data set
- Ex. Intrusion detection in computer networks
- Issue: Find an appropriate measurement of deviation

Contextual outlier (or *conditional outlier*)

- Object is O_c if it deviates significantly based on a selected context
- Ex. 40° C in Bengaluru: outlier? (depending on summer or winter?)
- Attributes of data objects should be divided into two groups
 - Contextual attributes: defines the context, e.g., time & location
 - Behavioral attributes: characteristics of the object, used in outlier evaluation, e.g., temperature
- Can be viewed as a generalization of *local outliers*—whose density significantly deviates from its local area
- Issue: How to define or formulate meaningful context?

Types of Outliers (Contd.)

Collective Outliers

- A subset of data objects *collectively* deviate significantly from the whole data set, even if the individual data objects may not be outliers
- Applications: E.g., *intrusion detection*:
 - When a number of computers keep sending denial-of-service packages to each other
- Detection of collective outliers
 - Consider not only behavior of individual objects, but also that of groups of objects
 - Need to have the background knowledge on the relationship among data objects, such as a distance or similarity measure on objects.
- A data set may have multiple types of outlier
- One object may belong to more than one type of outlier



Outlier Detection Approaches

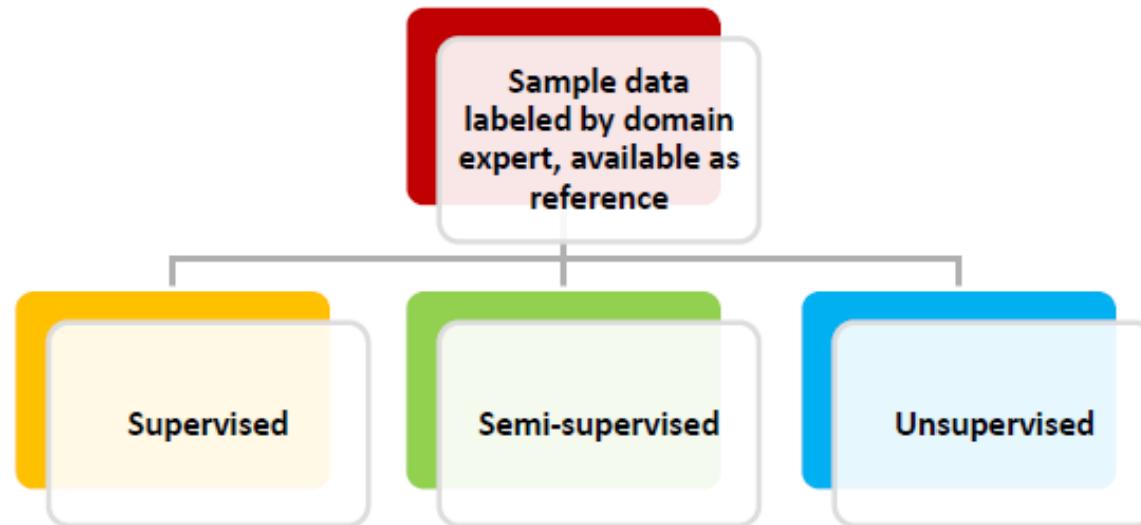
Outlier Detection Methods

Two ways to categorize outlier detection methods:

- Based on whether user-labeled examples of outliers can be obtained:
 - Supervised, semi-supervised vs. unsupervised methods
- Based on assumptions about normal data and outliers:
 - Statistical, proximity-based, and clustering-based methods

Outlier Detection Methods

The First Approach



<ul style="list-style-type: none"> ▪ Outlier detection works as classifier. ▪ Classification with two labels: normal and outliers. ▪ Challenges like: class imbalance, lack of representative outliers. ▪ Sensitivity (Recalls) of outlier detection is an important measure (TP/P). 	<ul style="list-style-type: none"> ▪ Only few labels are available. ▪ Classification model is prepared using available labeled data. Then unlabeled data is labeled using the model. ▪ Final model is used to detect outliers. 	<ul style="list-style-type: none"> ▪ Labels are not available. Classifiers cannot be built. ▪ Clustering can be used to detect outliers as normal object tend to form clusters. ▪ Expensive so not appealing. ▪ Difficult to isolate noise from outliers.
--	---	---

Outlier Detection I: Supervised Methods

Outlier Detection I: Supervised Methods

- Modeling outlier detection as a classification problem
 - Samples examined by domain experts used for training & testing
- Methods for Learning a classifier for outlier detection effectively:
 - Model normal objects & report those not matching the model as outliers, or
 - Model outliers and treat those not matching the model as normal
- Challenges
 - Imbalanced classes, i.e., outliers are rare: Boost the outlier class and make up some artificial outliers
 - Catch as many outliers as possible, i.e., recall is more important than accuracy (i.e., not mislabeling normal objects as outliers)

Outlier Detection II: Unsupervised Methods

Assume the normal objects are somewhat "clustered" into multiple groups, each having some distinct features

An outlier is expected to be far away from any groups of normal objects

Weakness: Cannot detect collective outlier effectively

- Normal objects may not share any strong patterns, but the collective outliers may share high similarity in a small area

Ex. In some intrusion or virus detection, normal activities are diverse

- Unsupervised methods may have a high false positive rate but still miss many real outliers.
- Supervised methods can be more effective, e.g., identify attacking some key resources

Many clustering methods can be adapted for unsupervised methods

- Find clusters, then outliers: not belonging to any cluster
- Problem 1: Hard to distinguish noise from outliers
- Problem 2: Costly since first clustering: but far less outliers than normal objects
 - Newer methods: tackle outliers directly

Outlier Detection III: Semi-Supervised Methods

Situation: In many applications, the number of labeled data is often small:
Labels could be on outliers only, normal objects only, or both

Semi-supervised outlier detection: Regarded as applications of semi-supervised learning

If some labeled normal objects are available

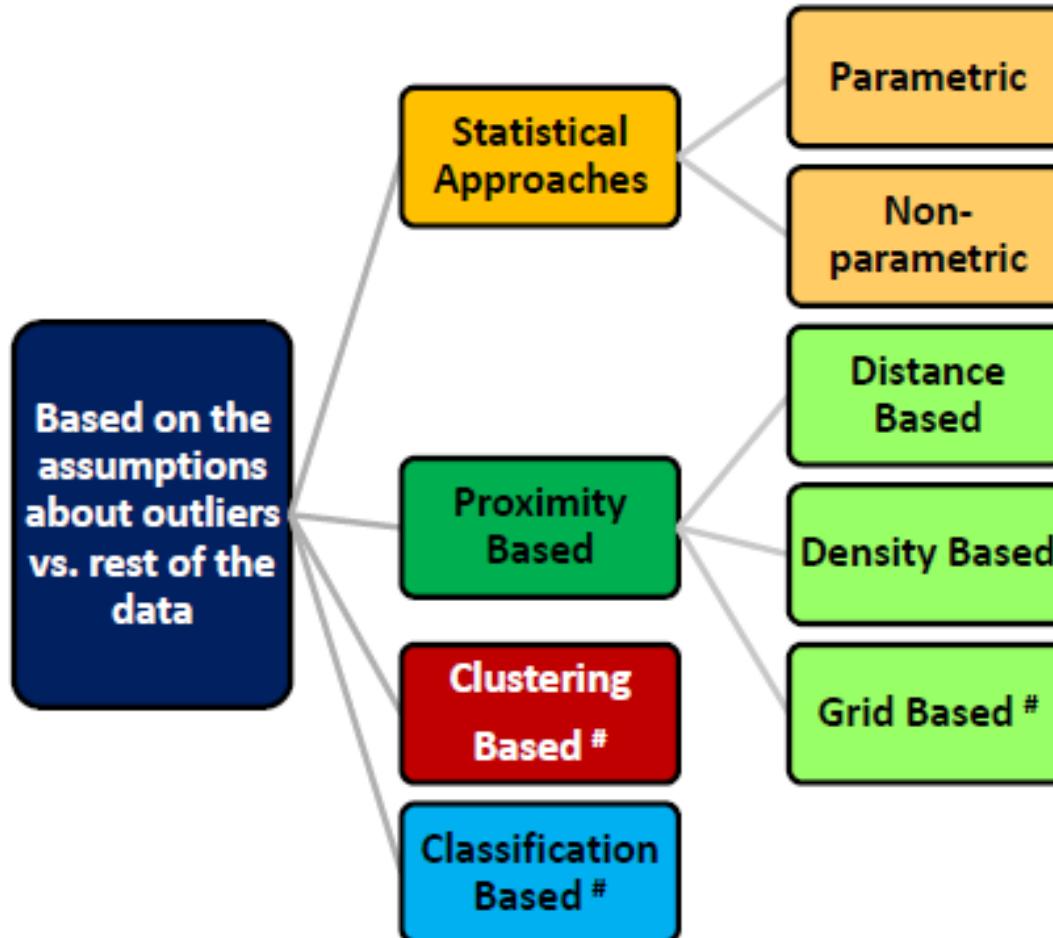
- Use the labeled examples and the proximate unlabeled objects to train a model for normal objects
- Those not fitting the model of normal objects are detected as outliers

If only some labeled outliers are available, a small number of labeled outliers may not cover the possible outliers well

- To improve the quality of outlier detection, one can get help from models for normal objects learned from unsupervised methods

Outlier Detection Methods

The Second Approach



- Not covered in this syllabus

Outlier Detection Methods: Approaches to Anomaly Detection

- Model-Based Techniques:

- Many anomaly detection techniques first build a model of the data. Anomalies are objects that do not fit the model very well.
- An object does not fit the model very well; i.e., it is an anomaly, if it is not very likely under the distribution. If the model is a set of clusters, then an anomaly is an object that does not strongly belong to any cluster.
- When a regression model is used, an anomaly is an object that is relatively far from its predicted value.
- Statistical methods (also known as model-based methods) make assumptions of data normality. They assume that normal data objects are generated by a statistical (stochastic) model, and that data not following the model are outliers.

- Proximity-Based Techniques:

- It is often possible to define a proximity measure between objects.
- Anomalous objects are those that are distant from most of the other objects.
- Many of the techniques in this area are based on distances and are referred to as distance-based outlier detection techniques.
- Proximity-based methods assume that an object is an outlier if the nearest neighbors of the object are far away in feature space, that is, the proximity of the object to its neighbors significantly deviates from the proximity of most of the other objects to their neighbors in the same data set.

Outlier Detection Methods: Approaches to Anomaly Detection

- Density-Based Techniques:

- Objects that are in regions of low density are relatively distant from their neighbors, and can be considered anomalous.
- The data sets can have regions of widely differing densities, and classifies a point as an outlier only if it has a local density significantly less than that of most of its neighbors.

- Clustering-Based Methods:

- Clustering-based methods assume that the normal data objects belong to large and dense clusters, whereas outliers belong to small or sparse clusters, or do not belong to any clusters.

Mining Contextual Outliers: Transform into Conventional Outlier Detection

If the contexts can be clearly identified, transform it to conventional outlier detection

1. Identify the context of the object using the contextual attributes
2. Calculate the outlier score for the object in the context using a conventional outlier detection method

Ex. Detect outlier customers in the context of customer groups

- Contextual attributes: *age group, postal code*
- Behavioral attributes: *# of trans/yr, annual total trans. amount*

Steps:

- (1) locate c's context,
- (2) compare c with the other customers in the same group, and
- (3) use a conventional outlier detection method

Mining Contextual Outliers: Modeling Normal Behavior with Respect to Contexts

In some applications, one cannot clearly partition the data into contexts

- Ex. if a customer suddenly purchased a product that is unrelated to those she recently browsed, it is unclear how many products browsed earlier should be considered as the context

Model the “normal” behavior with respect to contexts

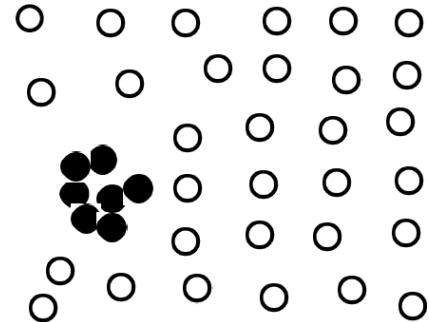
- Using a training data set, train a model that predicts the expected behavior attribute values with respect to the contextual attribute values
- An object is a contextual outlier if its behavior attribute values significantly deviate from the values predicted by the model

Using a prediction model that links the contexts and behavior, these methods avoid the explicit identification of specific contexts

Methods: A number of classification and prediction techniques can be used to build such models, such as regression, Markov Models, and Finite State Automaton

Mining Collective Outliers : On the Set of “Structured Objects”

- Collective outlier - objects as a group deviate from the entire data
- Need to examine the *structure* of the data set,
 - i.e, the relationships between multiple data objects
- Each of these structures is inherent to its respective type of data
 - For temporal data (such as time series and sequences)
 - explore the structures formed by time, which occur in segments of the time series or subsequences
 - For spatial data, explore local areas
 - For graph and network data, we explore subgraphs



Mining Collective Outliers : On the Set of “Structured Objects”

- Difference from the contextual outlier detection: the structures are often not explicitly defined, and have to be discovered as part of the outlier detection process.
- Collective outlier detection methods: two categories
 - Reduce the problem to conventional outlier detection
 - Identify *structure units*, treat each structure unit (e.g., subsequence, time series segment, local area, or subgraph) as a data object, and extract features
 - Then outlier detection on the set of “structured objects” constructed as such using the extracted features
 - e.g. Detect collective outliers in online social network of customers
 - Treat each possible subgraph of the network as a structure unit
 - Collective outlier: An outlier subgraph in the social network
 - Small subgraphs that are of very low frequency
 - Large subgraphs that are surprisingly frequent

Mining Collective Outliers II: Direct Modeling of the Expected Behavior of Structure Units

Model the expected behavior of structure units directly

- e.g. Detect collective outliers in temporal sequences
 - Learn a Markov model from the sequences
 - A subsequence can then be declared as a collective outlier if it significantly deviates from the model

Collective outlier detection is subtle due to the challenge of exploring the structures in data

- The exploration typically uses heuristics, and thus may be application dependent
- The computational cost is often high due to the sophisticated mining process

Challenges of Outlier Detection

- Modeling normal objects and outliers properly
 - Hard to enumerate all possible normal behaviors in an application
 - The border between normal and outlier objects is often a gray area
- Application-specific outlier detection
 - Choice of distance measure among objects and the model of relationship among objects are often application-dependent
 - E.g., clinic data: a small deviation could be an outlier; while in marketing analysis, larger fluctuations
- Handling noise in outlier detection
 - Noise may distort the normal objects and blur the distinction between normal objects and outliers. It may help hide outliers and reduce the effectiveness of outlier detection
- Understandability
 - Understand why these are outliers: Justification of the detection
 - Specify the degree of an outlier: the unlikelihood of the object being generated by a normal mechanism



Statistical Outliers

Statistical Approaches

Statistical approaches assume that the objects in a data set are generated by a stochastic process (a generative model)

- The effectiveness of statistical methods highly depends on whether the assumptions made for the statistical model hold true for the given data

Statistic models used in the methods may be parametric or nonparametric.

- A **parametric method** assumes that the normal data objects are generated by a parametric distribution
- A **nonparametric method** does not assume an a priori statistical model. Instead, a nonparametric method tries to determine the model from the input data.

Discordancy test

The statistical distribution-based approach identifies outliers with respect to the model using a *discordancy test*.

A statistical discordancy test examines first a *working hypothesis*. A **working hypothesis**, H , is a statement that the entire data set of n objects comes from an initial distribution model, F , that is,

$$H : o_i \in F, \text{ where } i=1,2,\dots,n$$

The hypothesis is retained if there is no statistically significant evidence supporting its rejection

A **discordancy test** verifies whether an object, o_i , is significantly large (or small) in relation to the distribution F

The result is very much dependent on which model F is chosen because o_i may be an outlier under one model and a perfectly valid value under another.

Statistical Approaches – Parametric Methods

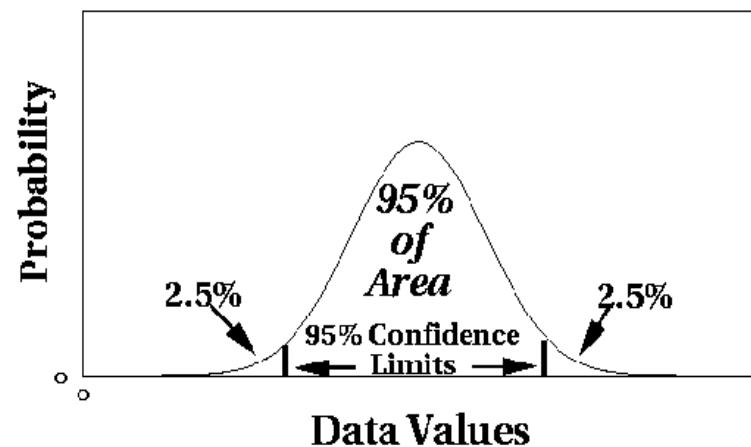
Assumes that the normal data is generated by a parametric distribution with parameter θ

The probability density function of the parametric distribution $f(x, \theta)$ gives the probability that object x is generated by the distribution

The smaller this value, the more likely x is an outlier

The parametric distribution can be normal distribution with a mean and variance.

Outliers are points where probability of occurrence is below a threshold.



Parametric Methods: Univariate Outliers

- Univariate data: A data set involving only one attribute or variable
- Often assume that data are generated from a normal distribution, learn the parameters from the input data, and identify the points with low probability as outliers
- Ex: Avg. temp.: {24.0, 28.9, 28.9, 29.0, 29.1, 29.1, 29.2, 29.2, 29.3, 29.4}

- Use the **maximum likelihood method** to estimate μ and σ

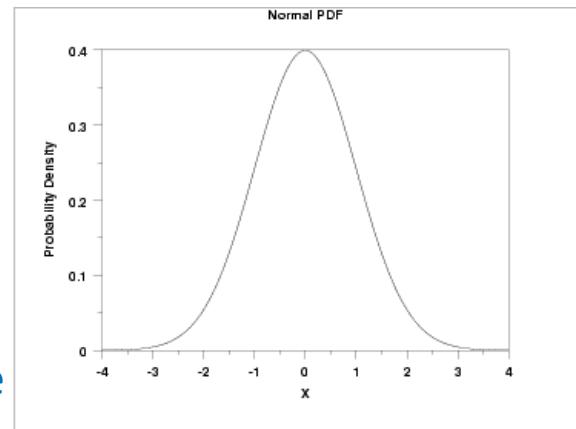
$$\hat{\mu} = \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

- For the above data with $n = 10$, we have

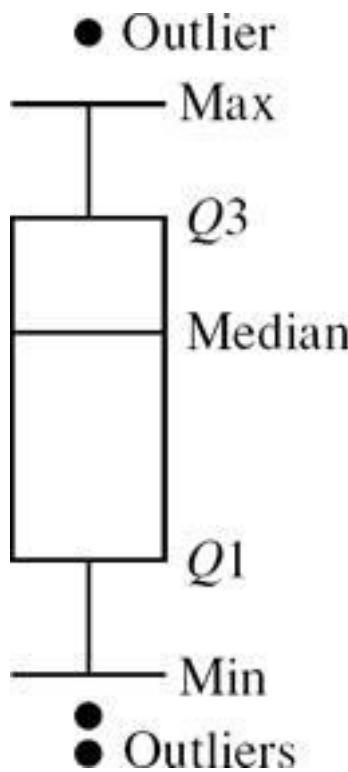
$$\hat{\mu} = 28.61 \quad \hat{\sigma} = \sqrt{2.29} = 1.51$$

- Then $(24 - 28.61) / 1.51 = -3.04 < -3$, 24 is an outlier since

$\mu \pm 3\sigma$ region contains 99.7% data



Visual Approach



A straightforward method for statistical outlier detection can also be used in visualization, e.g., the *boxplot method* plots the univariate input data using a five-number summary

- the smallest nonoutlier value (Min),
- the lower quartile (Q_1),
- the median (Q_2),
- the upper quartile (Q_3), and
- the largest nonoutlier value (Max).

The *interquartile range (IQR)* is defined as $Q_3 - Q_1$. Any object that is more than $1.5 \times IQR$ smaller than Q_1 or $1.5 \times IQR$ larger than Q_3 is treated as an outlier because the region between $Q_1 - 1.5 \times IQR$ and $Q_3 + 1.5 \times IQR$ contains 99.3% of the objects. The rationale is similar to using 3σ as the threshold for normal distribution

Parametric Methods: Detection of Multivariate Outliers

Multivariate data: A data set involving two or more attributes or variables

Transform the multivariate outlier detection task into a univariate outlier detection problem

Method 1. Compute Mahalanobis distance

- Mahalanobis distance is a measure of the distance between a point P and a distribution D.
- This distance is zero if P is at the mean of D, and grows as P moves away from the mean: along each principal component axis, it measures the number of standard deviations from P to the mean of D

Method 2. Use χ^2 –statistic:

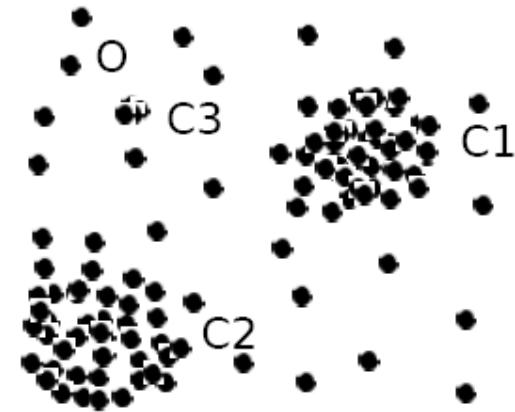
- where E_i is the mean of the i -dimension among all objects, and n is the dimensionality
- If χ^2 –statistic is large, then object o_i is an outlier

$$\chi^2 = \sum_{i=1}^n \frac{(o_i - E_i)^2}{E_i}$$

Parametric Methods: Using Mixture of Parametric Distributions

Assuming data generated by a normal distribution could be sometimes overly simplified

Example (right figure): The objects between the two clusters cannot be captured as outliers since they are close to the estimated mean



- To overcome this problem, assume the normal data is generated by two normal distributions. For any object o in the data set, the probability that o is generated by the mixture of the two distributions is given by

$$Pr(o|\Theta_1, \Theta_2) = f_{\Theta_1}(o) + f_{\Theta_2}(o)$$

- where f_{θ_1} and f_{θ_2} are the probability density functions of θ_1 and θ_2
- Then use EM algorithm to learn the parameters $\mu_1, \sigma_1, \mu_2, \sigma_2$ from data
- An object o is an outlier if it does not belong to any cluster

Detecting outliers

There are two basic types of procedures for detecting outliers:

Block procedures: In this case, either all of the suspect objects are treated as outliers or all of them are accepted as consistent.

Consecutive (or sequential) procedures: e.g. *inside-out* procedure. The idea is that the object that is least "likely" to be an outlier is tested first. If it is found to be an outlier, then all of the more extreme values are also considered outliers; otherwise, the next most extreme object is tested, and so on.

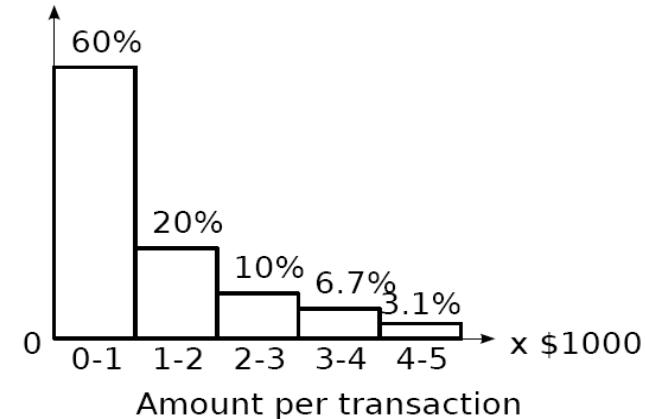
(This procedure tends to be more effective than block procedures.)

Non-Parametric Methods: Detection Using Histogram

The model of normal data is learned from the input data without any *a priori* structure.

Often makes fewer assumptions about the data, and thus can be applicable in more scenarios

Outlier detection using histogram:



- Figure shows the histogram of purchase amounts in transactions
 - A transaction in the amount of \$7,500 is an outlier, since only 0.2% transactions have an amount higher than \$5,000
- Problem: Hard to choose an appropriate bin size for histogram
 - Too small bin size → normal objects in empty/rare bins, false positive
 - Too big bin size → outliers in some frequent bins, false negative



Information Theoretic Outlier Detection

Information Theoretic Approach

Intuition: Key idea is to measure how much information decreases when you delete an observation

Let x be the observation in dataset D to be deleted

$$\text{Gain}(x) = \text{Info}(D) - \text{Info}(D - x)$$

- Anomalies should show higher gain
- Normal points should have less gain

Information Theoretic Example

Survey of height and weight for 100 participants

Weight	Height	Count
Low	Low	20
Low	Medium	15
Medium	Medium	40
High	High	20
High	Low	5

Entropy of the data = 2.084

Deleted Entity	Change in Entropy
{Low, Low}	0.004
{Low, Medium}	0.008
{Medium, Medium}	0.008
{High, High}	0.004
{High, Low}	0.024

Clearly, deletion of entry from the last category impacted entropy the most.

Strengths and Weaknesses

Solid theoretical foundation

Theoretically applicable to all kinds of data

Difficult and computationally expensive to implement in practice



Proximity Based Outliers

Proximity-Based Approaches: Distance-Based vs. Density-Based Outlier Detection

Intuition: Objects that are far away from the others are outliers

Assumption of proximity-based approach: The proximity of an outlier deviates significantly from that of most of the others in the data set

Two types of proximity-based outlier detection methods

- Distance-based outlier detection: An object o is an outlier if its neighborhood does not have enough other points
- Density-based outlier detection: An object o is an outlier if its density is relatively much lower than that of its neighbors

Distance-Based Outlier Detection

For each object o , examine the # of other objects in the r -neighborhood of o , where r is a user-specified **distance threshold**

An object o is an outlier if most (taking π as a **fraction threshold**) of the objects in D are far away from o , i.e., not in the r -neighborhood of o

$$\frac{\|\{o' | dist(o, o') \leq r\}\|}{\|D\|} \leq \pi$$

An object o is a $DB(r, \pi)$ outlier if

Equivalently, one can check the distance between o and its k -th nearest neighbor o_k , where

$$k = \lceil \pi \|D\| \rceil$$

o is an outlier if $dist(o, o_k) > r$

Distance-Based Outlier Detection

Algorithm: Distance-based outlier detection.

Input:

- a set of objects $D = \{o_1, \dots, o_n\}$, threshold r ($r > 0$) and π ($0 < \pi \leq 1$);

Output: $DB(r, \pi)$ outliers in D .

Method:

```

for  $i = 1$  to  $n$  do
   $count \leftarrow 0$ 
  for  $j = 1$  to  $n$  do
    if  $i \neq j$  and  $dist(o_i, o_j) \leq r$  then
       $count \leftarrow count + 1$ 
    if  $count \geq \pi \cdot n$  then
      exit  $\{o_i \text{ cannot be a } DB(r, \pi) \text{ outlier}\}$ 
    endif
  endif
endfor
print  $o_i \{o_i \text{ is a } DB(r, \pi) \text{ outlier according to (Eq. 12.10)}\}$ 
endfor;
  
```

Efficient computation: Nested loop algorithm

- For any object o_i , calculate its distance from other objects, and count the # of other objects in the r -neighborhood.
- If $\pi \cdot n$ other objects are within r distance, terminate the inner loop
- Otherwise, o_i is a $DB(r, \pi)$ outlier

Efficiency: Actually CPU time is not $O(n^2)$ but linear to the data set size since for most non-outlier objects, the inner loop terminates early

Distance-Based Outlier Detection: Improving Algorithm

Why efficiency is still a concern? When the complete set of objects cannot be held into main memory, cost of I/O swapping will be high

The major cost:

- (1) each object tests against the whole data set, why not only its close neighbor?
- (2) instead of checking objects one by one, why not group by group?

Grid-based method (CELL): Data space is partitioned into a multi-D grid. Only adjoining cells are checked for determining if object is an outlier

Distance-Based Outlier Detection: Limitations

Distance-based outliers, such as $DB(r, \pi)$ -outliers, are just one type of outlier

Distance-based outlier detection takes a global view of the data set

$DB(r, \pi)$ -outlier, for example, is far (as quantified by parameter r) from at least $(1 - \pi) \times 100\%$ of the objects in the data set. In other words, an outlier as such is remote from the majority of the data.

To detect distance-based outliers, we need two global parameters, r and π , which are applied to every outlier object.

Many real-world data sets demonstrate a more complex structure, where objects may be considered outliers with respect to their local neighborhoods, rather than with respect to the global data distribution.



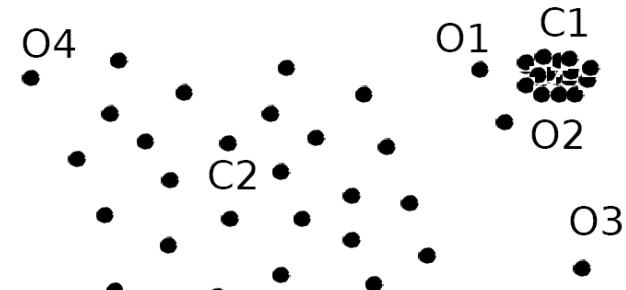
Density-Based Outlier

Density-Based Outlier Detection

Local outliers: Outliers comparing to their local neighborhoods, instead of the global data distribution

In Fig., o_1 and o_2 are local outliers to C_1 , o_3 is a global outlier, but o_4 is not an outlier. However, proximity-based clustering cannot find o_1 and o_2 are outlier (e.g., comparing with O_4).

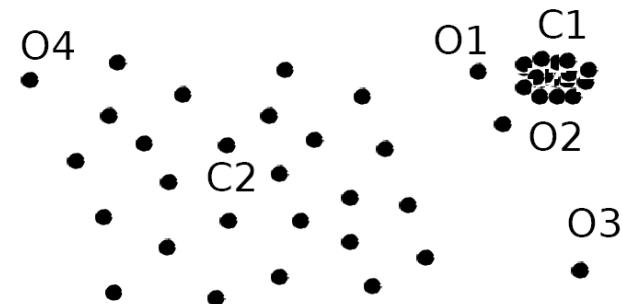
- Intuition (density-based outlier detection):
The density around an outlier object is significantly different from the density around its neighbors
- Method: Use the relative density of an object against its neighbors as the indicator of the degree of the object being outliers



Density-Based Outlier Detection

Local outliers: Outliers comparing to their local neighborhoods, instead of the global data distribution

In Fig., o_1 and o_2 are local outliers to C_1 , o_3 is a global outlier, but o_4 is not an outlier. However, proximity-based clustering cannot find o_1 and o_2 are outlier (e.g., comparing with O_4).



- Intuition (density-based outlier detection): The density around an outlier object is significantly different from the density around its neighbors
- Method: Use the relative density of an object against its neighbors as the indicator of the degree of the object being outliers

Local Reachability Density

k-distance of an object o , $\text{dist}_k(o)$: distance between o and its k-th NN

k-distance neighborhood of o , $N_k(o) = \{o' \mid o' \text{ in } D, \text{dist}(o, o') \leq \text{dist}_k(o)\}$

- $N_k(o)$ could be bigger than k since multiple objects may have identical distance to o

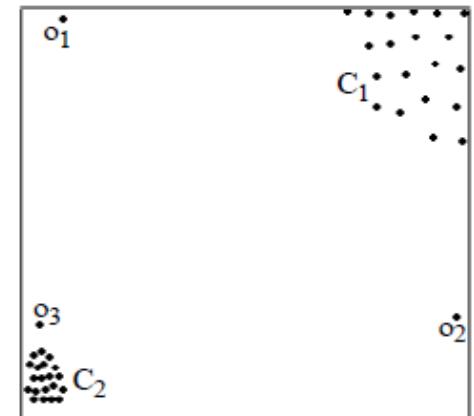
Reachability distance from o' to o :

$$\text{reachdist}_k(o \leftarrow o') = \max\{\text{dist}_k(o), \text{dist}(o, o')\}$$

where k is a user-specified parameter
that controls the smoothing effect

Local reachability density of o :

$$\text{lrd}_k(o) = \frac{\|N_k(o)\|}{\sum_{o' \in N_k(o)} \text{reachdist}_k(o' \leftarrow o)}$$



Local Outlier Factor: LOF

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

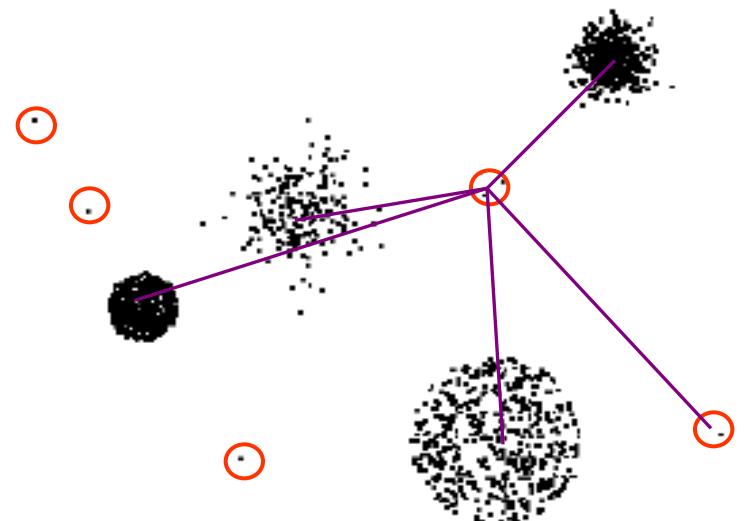
$$LOF_k(o) = \frac{\sum_{o' \in N_k(o)} \frac{lrk(o')}{lrk(o)}}{\|N_k(o)\|}$$

- the local outlier factor is the average of the ratio of the local reachability density of o and those of o 's k -nearest neighbors
- The lower the local reachability density of o , and the higher the local reachability density of the k -NN of o , the higher LOF
- This captures a local outlier whose local density is relatively low comparing to the local densities of its k -NN

Clustering-Based

Basic idea:

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





Base Rate Fallacy – An outlier challenge

Base Rate Fallacy

- Bayes theorem:

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}$$

- More generally:

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{\sum_{i=1}^n P(A_i) \cdot P(B|A_i)}$$

Base Rate Fallacy (Axelsson, 1999)

The base-rate fallacy is best described through example:

Suppose a doctor performs on a patient a diagnostic test that is 99% accurate symmetrically (i.e. both for presence and absence of disease).

The doctor may report the test result as bad news + good news

- Bad News – the patient is tested positive
- Good News – Out of entire population, rate of incidence is only 1 in 10000.

Now what is the probability that the patient has the disease?

Base Rate Fallacy

$$P(S|P) = \frac{P(S) \cdot P(P|S)}{P(S) \cdot P(P|S) + P(\neg S) \cdot P(P|\neg S)}$$

$$\begin{aligned} P(S|P) &= \frac{1/10000 \cdot 0.99}{1/10000 \cdot 0.99 + (1 - 1/10000) \cdot 0.01} = \\ &= 0.00980\dots \approx 1\% \end{aligned}$$

- Even though the test is 99% certain, your chance of having the disease is 1/100, because the population of healthy people is much larger than sick people

Base Rate Fallacy in Intrusion Detection

I: intrusive behavior,

$\neg I$: non-intrusive behavior

A: alarm

$\neg A$: no alarm

Detection rate (true positive rate): $P(A|I)$

False alarm rate: $P(A|\neg I)$

Goal is to maximize both

- Bayesian detection rate, $P(I|A)$
- $P(\neg I|\neg A)$

Detection Rate vs False Alarm Rate

$$P(I|A) = \frac{P(I) \cdot P(A|I)}{P(I) \cdot P(A|I) + P(\neg I) \cdot P(A|\neg I)}$$

Suppose there are about 20 intrusions Per million

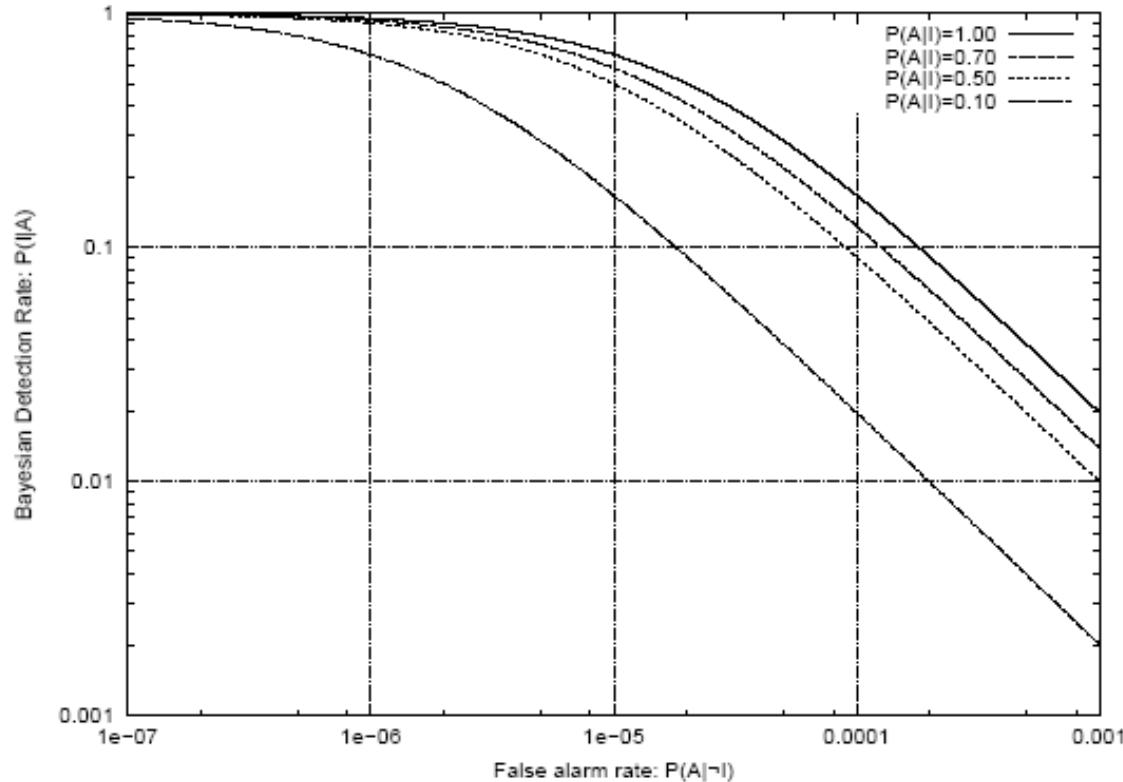
$$P(I) = 1 / \frac{1 \cdot 10^6}{2 \cdot 10} = 2 \cdot 10^{-5};$$

$$P(\neg I) = 1 - P(I) = 0.99998$$

$$P(I|A) = \frac{2 \cdot 10^{-5} \cdot P(A|I)}{2 \cdot 10^{-5} \cdot P(A|I) + 0.99998 \cdot P(A|\neg I)}$$

- False alarm rate becomes more dominant if $P(I)$ is very low

Detection Rate vs False Alarm Rate



- Axelsson: We need an extremely low false alarm rate to achieve a reasonable Bayesian detection rate

Prescribed Text Books

Author(s), Title, Edition, Publishing House	
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
	The Base-Rate Fallacy and the Difficulty of Intrusion Detection by Stefan Axelsson ACM Transactions on Information and System Security, Vol. 3, No. 3, August 2000, Pages 186–205



BITS Pilani

Pilani | Dubai | Goa | Hyderabad

S2-21_DSECLZC415 : Data Mining (Lecture #15 - Mining Unstructured Data)



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Data Mining

Mining Unstructured Data

Structured data vs. Unstructured data

Structured data

comprised of clearly defined data types whose pattern makes them easily searchable

Unstructured data – “everything else”

comprised of data that is usually not as easily searchable, including formats like audio, video, and social media postings

Unstructured Data

Can be

- Text
- www
- Multimedia
- Graph
- Spatial data

.....



Mining Text Data – NLP Challenges

Mining Text Data: An Introduction



Data Mining / Knowledge Discovery

Structured Data

```
HomeLoan (
  Loanee: Frank Rizzo
  Lender: MWF
  Agency: Lake View
  Amount: $200,000
  Term: 15 years
)
```

Multimedia



Free Text

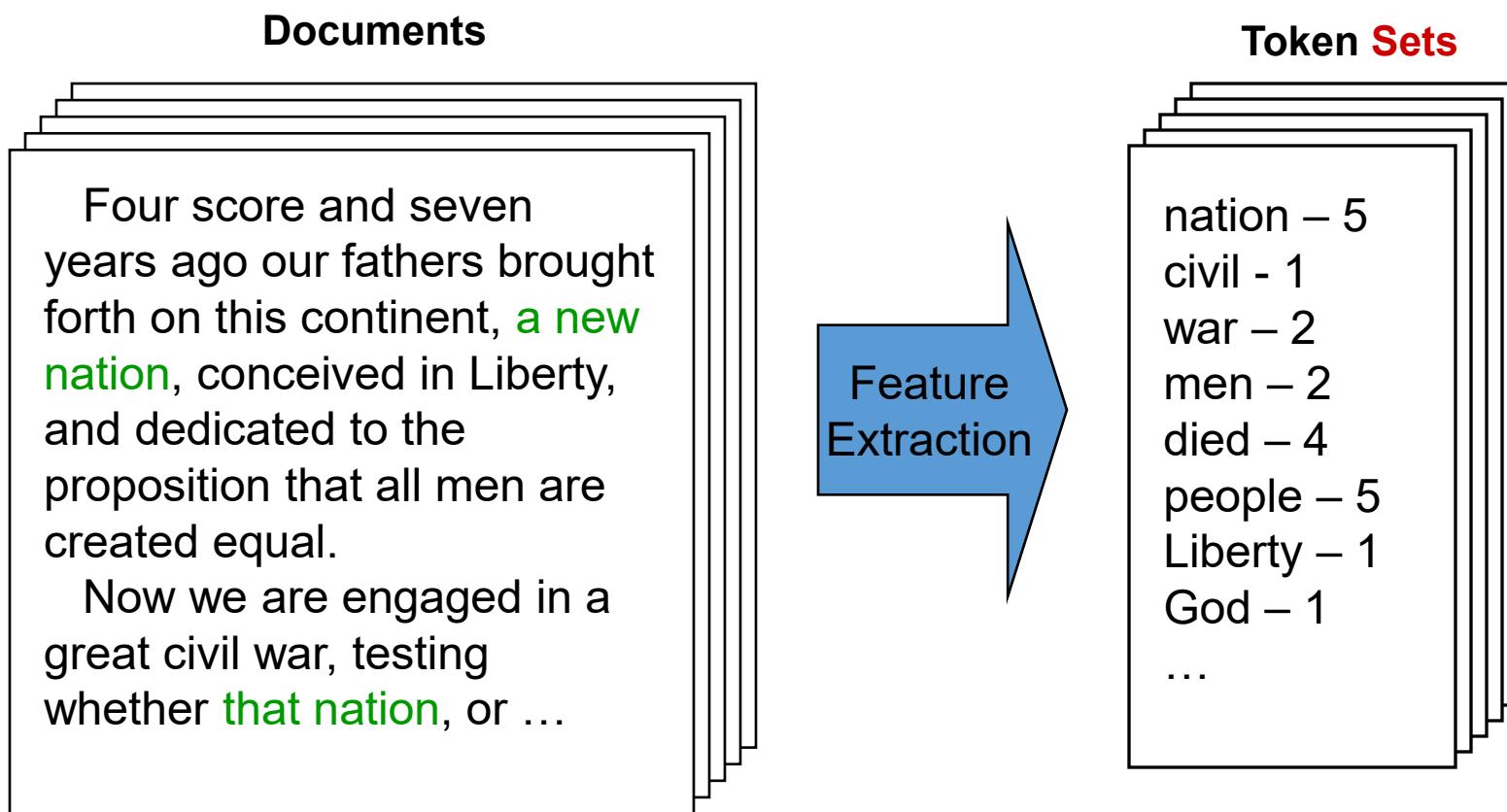
Frank Rizzo bought his home from Lake View Real Estate in 1992. He paid \$200,000 under a 15-year loan from MW Financial.

Hypertext

[Frank Rizzo](#) Bought
[this home](#) from [Lake View Real Estate](#) In **1992**.

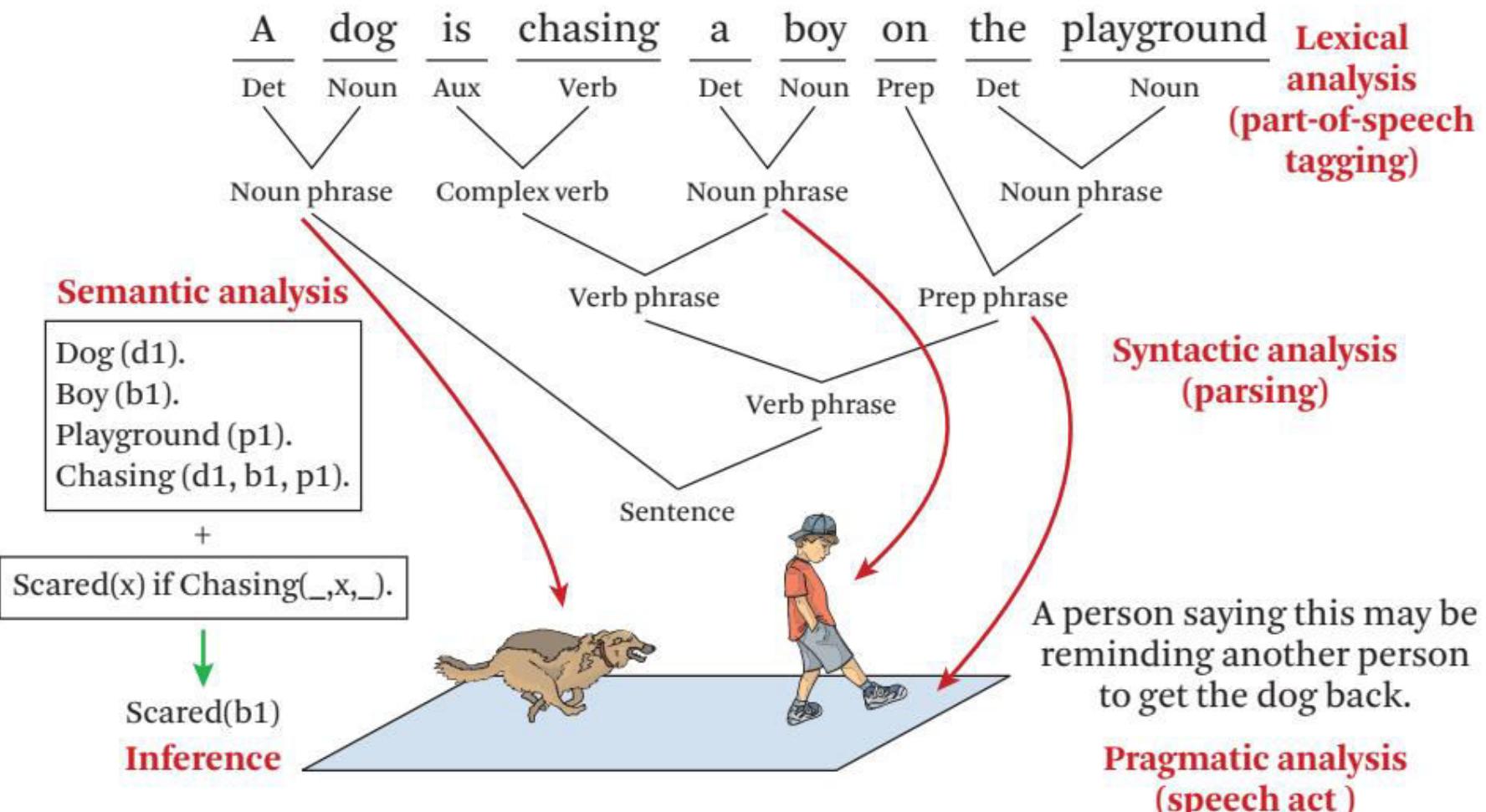
...

Bag-of-Tokens Approaches



**Loses all order-specific information!
Severely limits context!**

Natural Language Processing



General NLP—Too Difficult!

Word-level ambiguity

- “**design**” can be a noun or a verb (Ambiguous POS)
- “**root**” has multiple meanings (Ambiguous sense)

Syntactic ambiguity

- “**natural language processing**” (Modification)
- “**A man saw a boy with a telescope.**” (PP Attachment)

Anaphora resolution

- “**John persuaded Bill to buy a TV for himself.**”
(himself = John or Bill?)

Presupposition

- “**He has quit smoking.**” implies that he smoked before.

**Humans rely on context to interpret (when possible).
This context may extend beyond a given document!**



Information Retrieval

Text Databases and IR

Text databases (document databases)

- Large collections of documents from various sources: news articles, research papers, books, digital libraries, e-mail messages, and Web pages, library database, etc.
- Data stored is usually *semi-structured*
- Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data

Information retrieval

- A field developed in parallel with database systems
- Information is organized into (a large number of) documents
- Information retrieval problem: locating relevant documents based on user input, such as keywords or example documents

Information Retrieval

Typical IR systems

- Online library catalogs
- Online document management systems

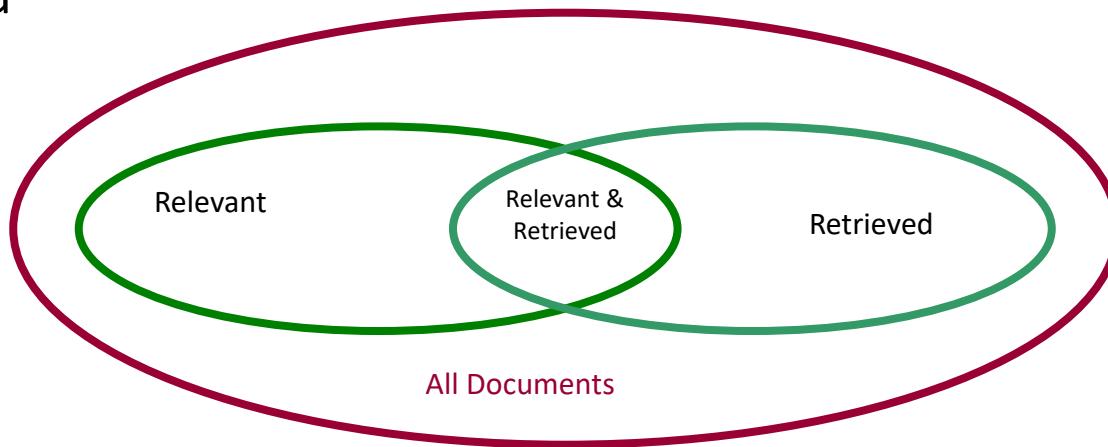
Information retrieval vs. database systems

- Some DB problems are not present in IR, e.g., update, transaction management, complex objects
- Some IR problems are not addressed well in DBMS, e.g., unstructured documents, approximate search using keywords and relevance

Basic Measures for Text Retrieval

Precision: the percentage of retrieved documents that are in fact relevant to the query (i.e., “correct” responses)

Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved



$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|} \quad recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}$$

Information Retrieval Techniques

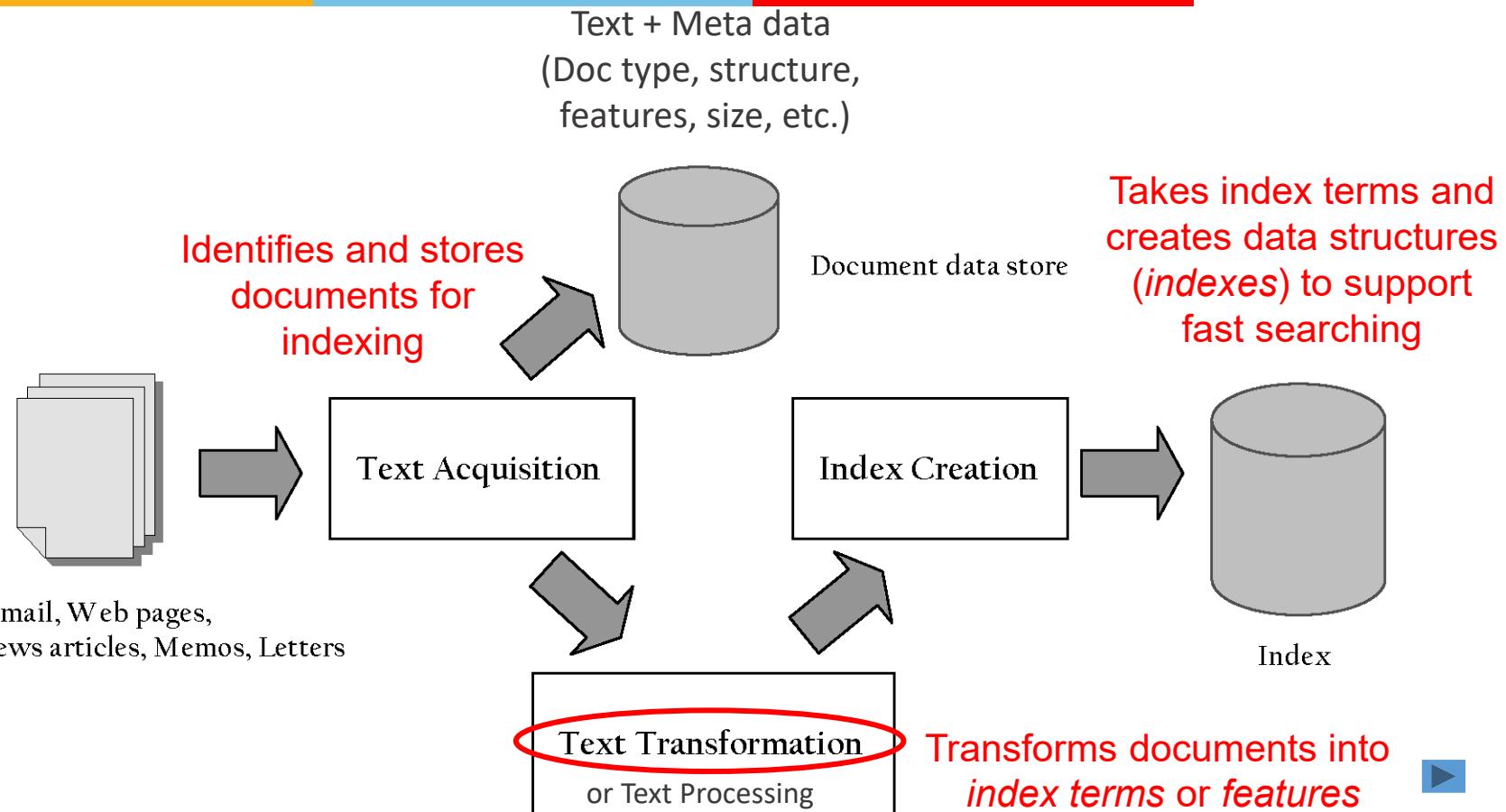
Basic Concepts

- A document can be described by a set of representative keywords called **index terms**.
- Different index terms have varying relevance when used to describe document contents.
- This effect is captured through the **assignment of numerical weights to each index term** of a document.
(e.g.: frequency, tf-idf)

DBMS Analogy

- Index Terms → **Attributes**
- Weights → **Attribute Values**

Indexing Process



Top 50 Words from AP89

<i>Word</i>	<i>Freq.</i>	<i>r</i>	<i>P_r(%)</i>	<i>r.P_r</i>	<i>Word</i>	<i>Freq</i>	<i>r</i>	<i>P_r(%)</i>	<i>r.P_r</i>
the	2,420,778	1	6.49	0.065	has	136,007	26	0.37	0.095
of	1,045,733	2	2.80	0.056	are	130,322	27	0.35	0.094
to	968,882	3	2.60	0.078	not	127,493	28	0.34	0.096
a	892,429	4	2.39	0.096	who	116,364	29	0.31	0.090
and	865,644	5	2.32	0.120	they	111,024	30	0.30	0.089
in	847,825	6	2.27	0.140	its	111,021	31	0.30	0.092
said	504,593	7	1.35	0.095	had	103,943	32	0.28	0.089
for	363,865	8	0.98	0.078	will	102,949	33	0.28	0.091
that	347,072	9	0.93	0.084	would	99,503	34	0.27	0.091
was	293,027	10	0.79	0.079	about	92,983	35	0.25	0.087
on	291,947	11	0.78	0.086	i	92,005	36	0.25	0.089
he	250,919	12	0.67	0.081	been	88,786	37	0.24	0.088
is	245,843	13	0.65	0.086	this	87,286	38	0.23	0.089
with	223,846	14	0.60	0.084	their	84,638	39	0.23	0.089
at	210,064	15	0.56	0.085	new	83,449	40	0.22	0.090
by	209,586	16	0.56	0.090	or	81,796	41	0.22	0.090
it	195,621	17	0.52	0.089	which	80,385	42	0.22	0.091
from	189,451	18	0.51	0.091	we	80,245	43	0.22	0.093
as	181,714	19	0.49	0.093	more	76,388	44	0.21	0.090
be	157,300	20	0.42	0.084	after	75,165	45	0.20	0.091
were	153,913	21	0.41	0.087	us	72,045	46	0.19	0.089
an	152,576	22	0.41	0.090	percent	71,956	47	0.19	0.091
have	149,749	23	0.40	0.092	up	71,082	48	0.19	0.092
his	142,285	24	0.38	0.092	one	70,266	49	0.19	0.092
but	140,880	25	0.38	0.094	people	68,988	50	0.19	0.093

Associated Press collection of news stories from 1989 (called AP89)

Vocabulary Growth

Heaps' Law, another prediction of word occurrence

As corpus grows, so does vocabulary size. However, fewer new words when corpus is already large

Observed relationship (**Heaps' Law**):

$$v = k \times n^\beta$$

where

v is the *vocabulary size* (number of *unique words*)

n is the *total number of words* in corpus

k, β are parameters that vary for each corpus

(typical values given are $10 \leq k \leq 100$ and $\beta \approx 0.5$)

- Predicting that the number of new words increases very rapidly when the corpus is small



Information Retrieval Techniques

Information Retrieval Techniques

Index Terms (Attribute) Selection:

- Stop list
- Word stem
- Index terms weighting methods

Terms \times Documents Frequency Matrices

Information Retrieval Models:

- Boolean Model (simplistic)
- Vector Space Model (Document Ranking considered)
- Probabilistic Retrieval Models (Ranked as per probability of relevance)

Text Retrieval Model 1: Boolean Model

Consider that index terms are either present or absent in a document

As a result, the index term weights are assumed to be all binaries

A query is composed of index terms linked by three connectives:
not, **and**, and **or**

- e.g.: car *and* repair, plane *or* airplane

The Boolean model predicts that each document is either relevant or non-relevant based on the match of a document to the query

Keyword-Based Retrieval

A document is represented by a string, which can be identified by a set of keywords

Queries may use **expressions** of keywords

- E.g., car *and* repair shop, tea *or* coffee, DBMS *but not* Oracle
- Queries and retrieval should consider **synonyms**, e.g., repair and maintenance

Major difficulties of the model

- **Synonymy**: A keyword T does not appear anywhere in the document, even though the document is closely related to T , e.g., data mining
- **Polysemy**: The same keyword may mean different things in different contexts, e.g., mining

Similarity-Based Retrieval in Text Data

Finds similar documents based on a set of common keywords

- Answer should be based on the degree of relevance based on the nearness of the keywords, relative frequency of the keywords, etc.

Basic techniques

-Stop list

- Set of words that are deemed “irrelevant”, even though they may appear frequently
- E.g., *a, the, of, for, to, with*, etc.
- Stop lists may vary when document set varies

Similarity-Based Retrieval in Text Data (contd)

- Word stem
 - Several words are small syntactic variants of each other since they share a common word stem
 - E.g., *drug, drugs, drugged*
- A term frequency table
 - Each entry $frequent_table(i, j) = \#$ of occurrences of the word t_i in document d_j
 - Usually, the *ratio* instead of the absolute number of occurrences is used
- Similarity metrics: measure the closeness of a document to a query (a set of keywords)
 - Relative term occurrences
 - Cosine distance:

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$

Text Retrieval Model 2: Vector Space Model

Represent a doc by a term vector

- Term: basic concept, e.g., word or phrase
- Each term defines one dimension
- N terms define a N-dimensional space
- Element of vector corresponds to term weight
- E.g., $d = (x_1, \dots, x_N)$, x_i is “importance” of term i

New document is assigned to the most likely category based on vector similarity.

What Vector Space Model Does Not Specify

How to select terms to capture “basic concepts”

- Word stopping
 - e.g. “a”, “the”, “always”, “along”
- Word stemming
 - e.g. “computer”, “computing”, “computerize” => “compute”
- Latent semantic indexing

How to assign weights

- Not all words are equally important: Some are more indicative than others
 - e.g. “algebra” vs. “science”

How to measure the similarity

Text Retrieval Model 3: How to Assign Weights

Two-fold heuristics based on frequency

- TF (Term frequency)
 - More frequent ***within*** a document → more relevant to semantics
- IDF (Inverse document frequency)
 - Less frequent ***among*** documents → more discriminative

Cornell SMART TF-IDF Model

TF is computed using the equation:

$$TF(d, t) = \begin{cases} 0 & \text{if } freq(d, t) = 0 \\ 1 + \log(1 + \log(freq(d, t))) & \text{otherwise.} \end{cases}$$

IDF is computed using the equation:

$$IDF(t) = \log \frac{1 + |d|}{|d_t|}$$

where d is the document collection, and d_t is the set of documents containing term t. If $|d_t| \ll |d|$, the term t will have a large IDF scaling factor

Alternate TF-IDF Weighting

TF Weighting:

- More frequent => more relevant to topic
 - Raw TF = $f(t, d)$: how many times term t appears in doc d

Normalization:

- Document length varies => relative frequency preferred
 - e.g., Maximum frequency normalization

$$TF(t, d) = 0.5 + \frac{0.5 * f(t, d)}{MaxFreq(d)}$$

IDF Idea:

- Less frequent *among* documents → more discriminative

Formula:

$$IDF(t) = 1 + \log\left(\frac{n}{k}\right)$$

n — total number of docs
 k — # docs with term t appearing

(IDF – inverse document frequency)

TF-IDF Weighting

TF-IDF weighting : $\text{weight}(t, d) = \text{TF}(t, d) * \text{IDF}(t)$

- Frequent within doc \rightarrow high tf \rightarrow high weight
- Selective among docs \rightarrow high idf \rightarrow high weight

Recall VS model

- Each selected term represents one dimension
- Each doc is represented by a feature vector
- Its t -term coordinate of document d is the TF-IDF weight
- This is more reasonable

Just for illustration ...

- Many complex and more effective weighting variants exist in practice

How to Measure Similarity?

Given two documents

$$D_i = (w_{i1}, w_{i2}, \dots, w_{iN}) \quad D_j = (w_{j1}, w_{j2}, \dots, w_{jN})$$

Similarity definition

- dot product
- normalized dot product (or cosine)

$$\text{Sim}(D_i, D_j) = \sum_{t=i}^N w_{it} * w_{jt}$$

$$\text{Sim}(D_i, D_j) = \frac{\sum_{t=1}^N w_{it} * w_{jt}}{\sqrt{\sum_{t=1}^N (w_{it})^2 * \sum_{t=1}^N (w_{jt})^2}}$$

Illustrative Example

doc1

text
mining
search
engine
text

$$\text{Sim}(\text{newdoc}, \text{doc1}) = (4.8 * 2.4 + 4.5 * 4.5) / \|v_1\| * \|v_n\|$$

doc2

travel
text
map
travel

$$\text{Sim}(\text{newdoc}, \text{doc2}) = 2.4 * 2.4 / \|v_2\| * \|v_n\|$$

To whom is newdoc more similar?

doc3

government
president
congress

	text	mining	travel	map	search	engine	govern	president	congress
IDF(faked)	2.4	4.5	2.8	3.3	2.1	5.4	2.2	3.2	4.3
doc1	2 (4.8)	1 (4.5)			1 (2.1)	1 (5.4)			
doc2	1 (2.4)		2 (5.6)	1 (3.3)					
doc3							1 (2.2)	1(3.2)	1(4.3)
newdoc	1(2.4)	1(4.5)							

.....

VS Model-Based Classifiers

What do we have so far?

- A feature space with similarity measure
- This is a classic supervised learning problem
 - Search for an approximation to classification hyper plane

VS model based classifiers

- K-NN
- Decision tree based
- Neural networks
- Support vector machine



Text Data Mining

Types of Text Data Mining

- Keyword-based association analysis
- Automatic document classification
- Similarity detection
 - Cluster documents by a common author
 - Cluster documents containing information from a common source
- Link analysis: unusual correlation between entities
- Sequence analysis: predicting a recurring event
- Anomaly detection: find information that violates usual patterns
- Hypertext analysis
 - Patterns in anchors/links
 - Anchor text correlations with linked objects

Keyword-Based Association Analysis

- Motivation
 - Collect sets of keywords or terms that occur frequently together and then find the **association** or **correlation** relationships among them
- Association Analysis Process
 - Preprocess the text data by parsing, stemming, removing stop words, etc.
 - Evoke association mining algorithms
 - Consider each document as a transaction
 - View a set of keywords in the document as a set of items in the transaction
 - Term level association mining
 - No need for human effort in tagging documents
 - The number of meaningless results and the execution time is greatly reduced

Text Classification

- Motivation
 - Automatic classification for the large number of on-line text documents (Web pages, e-mails, corporate intranets, etc.)
- Classification Process
 - Data preprocessing
 - Definition of training set and test sets
 - Creation of the classification model using the selected classification algorithm
 - Classification model validation
 - Classification of new/unknown text documents
- Text document classification differs from the classification of relational data
 - Document databases are not structured according to attribute-value pairs

Document Clustering

- Motivation
 - Automatically group related documents based on their contents
 - No predetermined training sets or taxonomies
 - Generate a taxonomy at runtime
- Clustering Process
 - Data preprocessing: remove stop words, stem, feature extraction, lexical analysis, etc.
 - Hierarchical clustering: compute similarities applying clustering algorithms.
 - Model-Based clustering: clusters are represented by “exemplars”. (e.g.: Self-Organizing Maps)

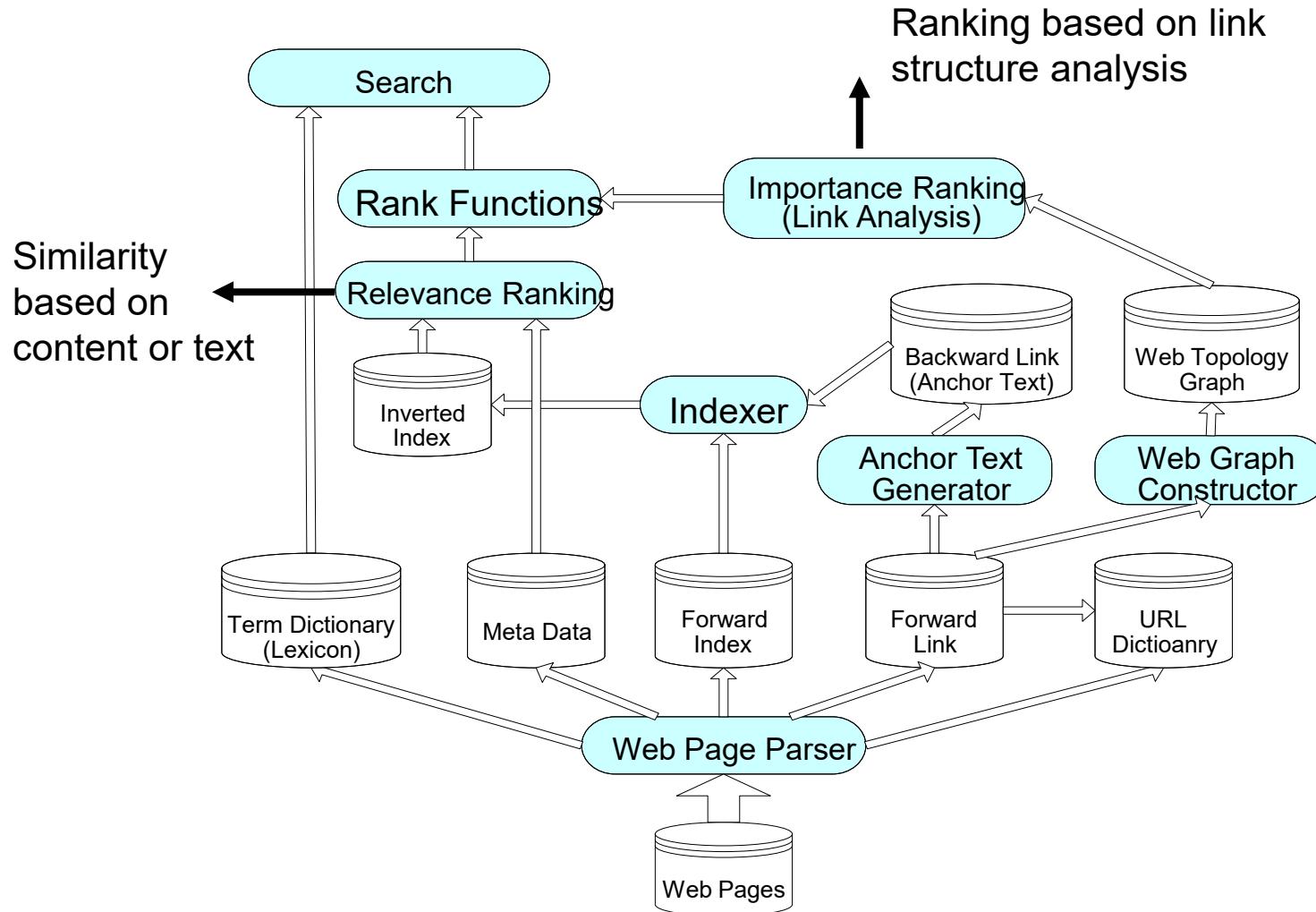


Mining WWW

Challenges in Mining WWW

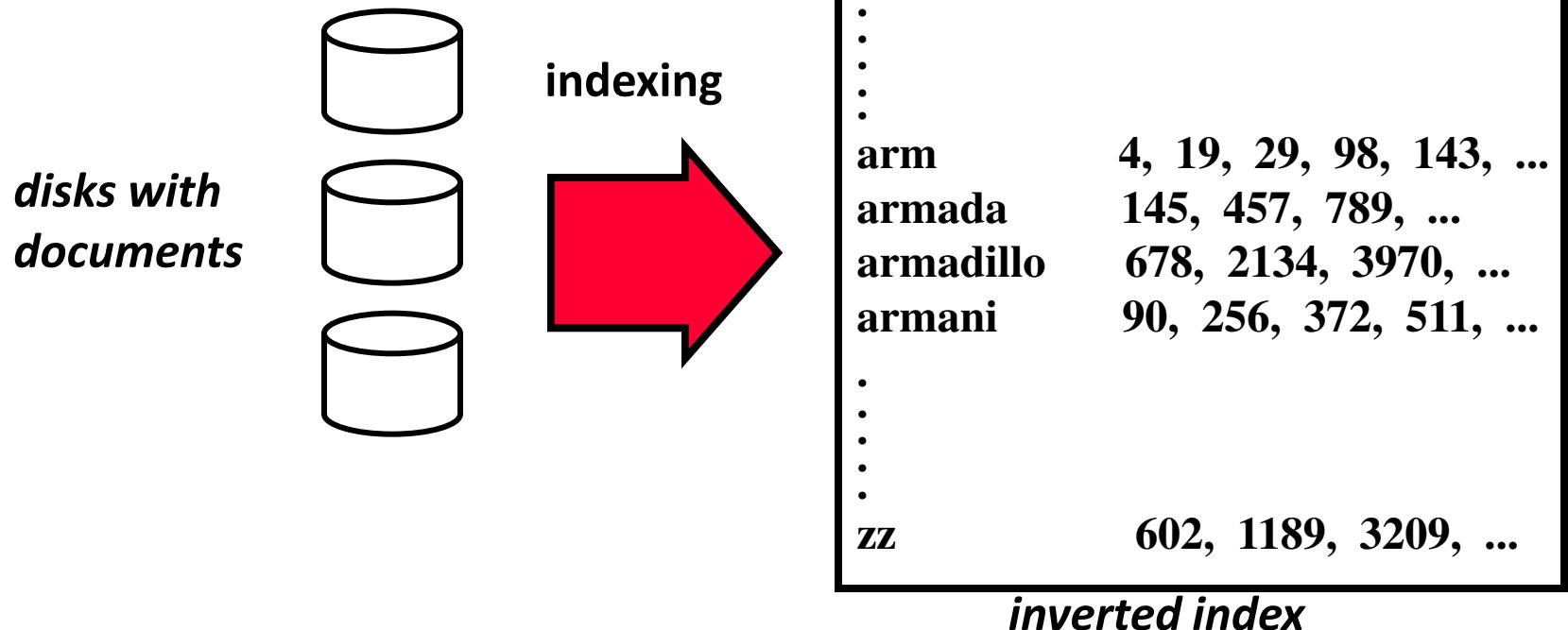
- The Web is too huge for effective data warehousing and data mining. It is barely possible to set up a data warehouse to replicate, store, or integrate all of the data on the Web
- The complexity of Web pages is far greater than that of any traditional text document collection. There is no index by category, nor by title, author, cover page, table of contents, etc.
- The Web is a highly dynamic information source. Not only does the Web grow rapidly, but is also constantly updated. Linkage information and access records are also updated frequently
- The Web serves a broad diversity of user communities. Users may have very different backgrounds, interests, and usage purposes.
- Only a small portion of the information on the Web is truly relevant or useful. It is said that 99% of the Web information is useless to 99% of Web users. How can the portion of the Web that is truly relevant to your interest be determined? How can we find high quality Web pages on a specified topic?

Search Engine – Two Rank Functions



Relevance Ranking

- Inverted index
 - A data structure for supporting text queries
 - like index in a book



Introduction on PageRank

PageRank is a link analysis algorithm ... with the purpose of "measuring" its (Webpage) relative importance within the set.

– From Wikipedia, the free encyclopedia

Developed by Larry Page as his PhD research topic
3 years later, he quit Stanford and founded Google with Brin

Apparently Larry Page had lost his PhD qualification.

PageRank: the intuitive idea

PageRank relies on the democratic nature of the Web by using its vast link structure as an indicator of an individual page's value or quality.

PageRank interprets a hyperlink from page x to page y as a vote, by page x , for page y .

However, PageRank looks at more than the sheer number of votes; it also analyzes the page that casts the vote.

- Votes casted by “important” pages weigh more heavily and help to make other pages more "important."

This is exactly the idea of **rank prestige** in social network.

More specifically ...

A hyperlink from a page to another page is an implicit conveyance of authority to the target page.

- The more in-links that a page i receives, the more prestige the page i has.

Pages that point to page i also have their own prestige scores.

- A page of a higher prestige pointing to i is more important than a page of a lower prestige pointing to i .
- In other words, a page is important if it is pointed to by other important pages.

PageRank algorithm

According to **rank prestige**, the importance of page i (i 's PageRank score) is the sum of the PageRank scores of all pages that point to i .

Since a page may point to many other pages, its prestige score should be shared.

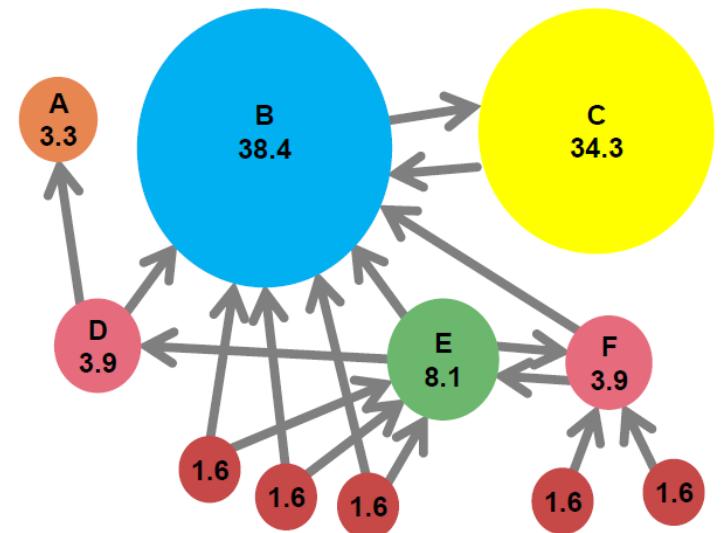
The Web as a directed graph $G = (V, E)$. Let the total number of pages be n . The PageRank score of the page i (denoted by $P(i)$) is defined by:

$$P(i) = \sum_{(j,i) \in E} \frac{P(j)}{O_j},$$

O_j is the number
of out-link of j

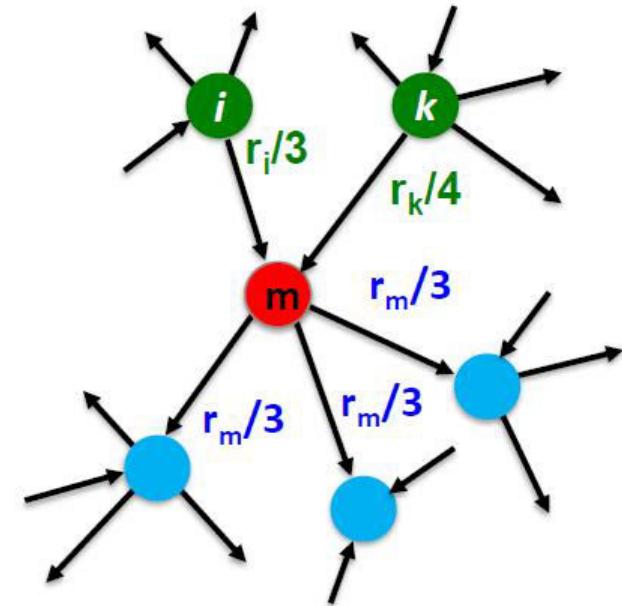
PageRank Score

- For depiction, bigger the circle, larger the PageRank.
- Page-B is pointed to by several pages. Therefore the highest PageRank.
- Page-C is not pointed to by several pages but pointed to by an important page-B having a high PageRank. Therefore a higher PageRank than several others.
- The same logic applies elsewhere.



PageRank: Simple Formulation Example

- Users of the Web “vote” through their web pages. On their page, they keep the links to those pages which they think are good.
- Let for the page m the PageRank is r_m .
- If page m has n outlinks, each link will get r_m/n votes.
- The own PageRank of page m will be the sum of votes on its inlinks.
- So, $r_m = r_i/3 + r_k/4$.
- In the shown figure, the outlinks of page m will each have $r_m/3$ votes because there are only 3 outlinks from page m .



Matrix notation

We have a system of n linear equations with n unknowns.

We can use a matrix to represent them.

Let \mathbf{P} be a n -dimensional column vector of PageRank values,
i.e., $\mathbf{P} = (P(1), P(2), \dots, P(n))^T$.

Let \mathbf{A} be the adjacency matrix of our graph with

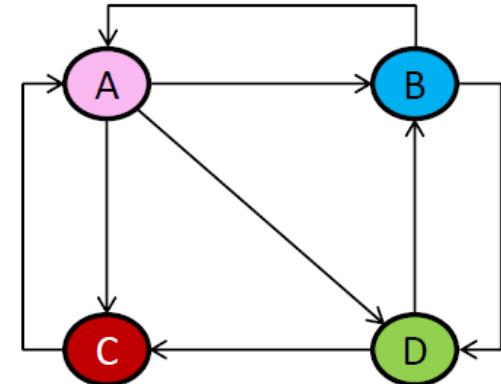
$$A_{ij} = \begin{cases} \frac{1}{O_i} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

We can write the n equations with (PageRank)

$$\mathbf{P} = \mathbf{A}^T \mathbf{P}$$

Transition Matrix - Example

- In a tiny version of the Web let us say there are only 4 Web pages – A, B, C and D.
- A random surfer at A can go to B, C or D with equal probability of $1/3$ and a 0 probability to A because there is no self loop.
- Similarly a random surfer at B can go to A or D with probability $1/2$.
- In general, what happens to the random surfers after one step can be captured in the transition matrix M.
- Each element m_{pq} of M will have a value of $1/x$, if page q has x outlinks where one of the outlinks is to page p otherwise $m_{pq} = 0$.



$$\begin{bmatrix}
 & A & B & C & D \\
 A & 0 & 1/2 & 1 & 0 \\
 B & 1/3 & 0 & 0 & 1/2 \\
 C & 1/3 & 0 & 0 & 1/2 \\
 D & 1/3 & 1/2 & 0 & 0
 \end{bmatrix}$$

Calculation of PageRank Vector - Using Matrix Multiplication Method

- Let us assume a random surfer can start his surfing from any page with equal probability. So the initial vector matrix $v_0 = [1/4 \ 1/4 \ 1/4 \ 1/4]^T$.
- Let M is the transition matrix for the given graph and v_t is the PageRank vector at iteration t .
- So, the PageRank vector at iteration $t+1$ is given by: $v_{t+1} = M.v_t$
- Two matrices A and B can be multiplied if the count of columns in A is equal to count of rows in B . Therefore:

$$\begin{array}{c}
 \textcolor{blue}{M} \\
 \left[\begin{array}{cccc}
 0 & 1/2 & 1 & 0 \\
 1/3 & 0 & 0 & 1/2 \\
 1/3 & 0 & 0 & 1/2 \\
 1/3 & 1/2 & 0 & 0
 \end{array} \right]
 \end{array}
 \times
 \begin{array}{c}
 \textcolor{blue}{v}_0 \\
 \left[\begin{array}{c}
 1/4 \\
 1/4 \\
 1/4 \\
 1/4
 \end{array} \right]
 \end{array}
 =
 \begin{array}{c}
 \textcolor{blue}{v}_1 \\
 \left[\begin{array}{c}
 9/24 \\
 5/24 \\
 5/24 \\
 5/24
 \end{array} \right]
 \end{array}$$

$$\begin{array}{c}
 \textcolor{blue}{M} \\
 \left[\begin{array}{cccc}
 0 & 1/2 & 1 & 0 \\
 1/3 & 0 & 0 & 1/2 \\
 1/3 & 0 & 0 & 1/2 \\
 1/3 & 1/2 & 0 & 0
 \end{array} \right]
 \end{array}
 \times
 \begin{array}{c}
 \textcolor{blue}{v}_1 \\
 \left[\begin{array}{c}
 9/24 \\
 5/24 \\
 5/24 \\
 5/24
 \end{array} \right]
 \end{array}
 =
 \begin{array}{c}
 \textcolor{blue}{v}_2 \\
 \left[\begin{array}{c}
 15/48 \\
 11/48 \\
 11/48 \\
 11/48
 \end{array} \right]
 \end{array}
 \quad \text{and so on.....}$$

Signals for Google Search Engine

- Google apparently deals with the 15 percent of queries a day it gets which its systems have never seen before.
- For last 7 years, Google has been using RankBrain, which uses AI/ML to make a guess as to what words or phrases might have a similar meaning
- Google Executives state that three major search ranking factors are
 - Content (of query)
 - Links
 - RankBrain

<https://searchengineland.com/now-know-googles-top-three-search-ranking-factors-245882>

Clark, Jack. "Google Turning Its Lucrative Web Search Over to AI Machines". Bloomberg Business. 2015

Thank You

References

	Author(s), Title, Edition, Publishing House
T1	Tan P. N., Steinbach M & Kumar V. "Introduction to Data Mining" Pearson Education
T2	Data Mining: Concepts and Techniques, Third Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
	Data Mining: Concepts and Techniques, Second Edition by Jiawei Han, Micheline Kamber Morgan Kaufmann Publishers
	Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining ChengXiang Zhai, Sean Massung, ACM Books 2016
	Web data Mining - Exploring Hyperlinks, Contents and Usage Data, By Bing Liu, Second Edition, Springer, July 2011
	Search Engines: Information Retrieval in Practice, Croft, Metzler, and Strohman, 2010



BITS Pilani

Pilani | Dubai | Goa | Hyderabad

S2-21_DSECLZC415 : Data Mining (Lecture #15 - Data Mining Applications)



- *The slides presented here are obtained from the authors of the books and from various other contributors. I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*



BITS Pilani

Pilani|Dubai|Goa|Hyderabad

Data Mining

Data Mining Applications



9.1 Recommendation Systems

Recommendation System

- System that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options

Burke (2002)

- A personalized information filtering technology used to either predict whether a particular user will like a particular item (prediction problem) or to identify a set of N items that will be of interest to a certain user (top- N recommendation problem).

Deshpande and Karypis (2004)

- Recommender systems suggest items of interest to users based on their explicit and implicit preferences, the preferences of other users, and user and item attributes

Schein et al. (2005)

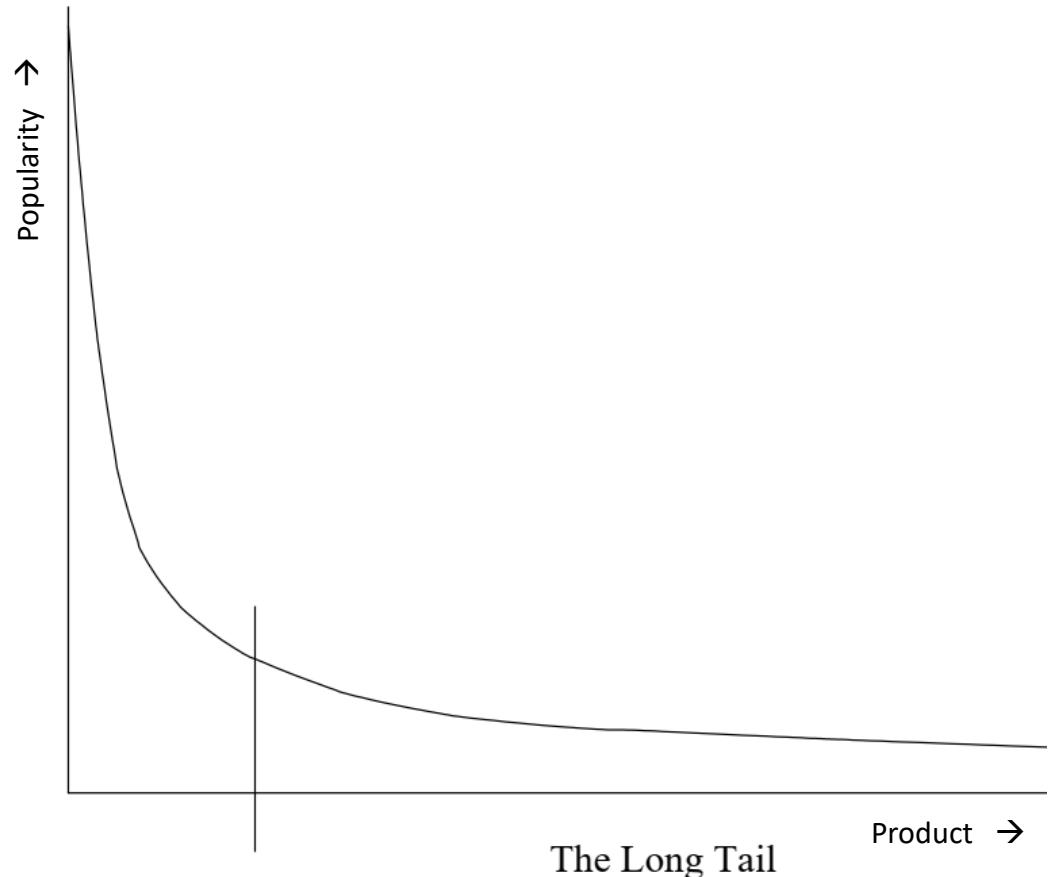
Why Recommender Systems?

- Physical delivery systems are characterized by a scarcity of resources. Brick-and-mortar stores have limited shelf space, and can show the customer only a small fraction of all the choices that exist.
 - A physical bookstore may have several thousand books on its shelves
 - It is not possible to tailor the store to each individual customer
 - The choice of what is made available is governed only by the aggregate numbers
 - Typically, a bookstore will display only the books that are most popular, and a newspaper will print only the articles it believes the most people will be interested in.
- On-line stores can make anything that exists available to the customer.
 - Largest ecommerce stores offer millions of books.
 - It is not possible to present all available items to the user, the way physical institution can. Neither can we expect users to have heard of each of the items they might like.

Why Recommender Systems?

The long tail: physical institutions can only provide what is popular, while on-line institutions can make everything available

The long-tail phenomenon forces on-line institutions to recommend items to individual users.



Popular Recommendation Engines

Some popular recommendation engines which have been proved as highly profitable:

- the Amazon.com,
- the Netflix.com and
- the Google.com recommendation engines

Amazon.com claims that 35 % of products sales result from recommendations.

About 66 % of movies rented in Netflix.com are recommended.

Google News Recommendations generate 38 % more click-throughs .

Into Thin Air and Touching the Void

An extreme example of how the long tail, together with a well designed recommendation system can influence events is the story told by Chris Anderson about a book called *Touching the Void*. This mountain-climbing book was not a big seller in its day, but many years after it was published, another book on the same topic, called *Into Thin Air* was published. Amazon's recommendation system noticed a few people who bought both books, and started recommending *Touching the Void* to people who bought, or were considering, *Into Thin Air*. Had there been no on-line bookseller, *Touching the Void* might never have been seen by potential buyers, but in the on-line world, *Touching the Void* eventually became very popular in its own right, in fact, more so than *Into Thin Air*.

Mining of Massive Datasets
By Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman

Recommendation Approaches

The approaches of recommender systems: collaborative filtering (CF), content-based Filtering (CB) and hybrid methods:

- ***Collaborative filtering*** algorithms recommend those items to the target user, that have been rated highly by other users with similar preferences and tastes.
 - "Customers who bought item i also bought item y "
- ***Content-based filtering*** uses the information derived from documents or item features (eg. terms or attributes). It uses a set of attributes, which describes the items and recommends other items similar to those that exist in the user's profile.
 - The cold start problem for new items and new users are alleviated, provided that features of users and items are known.
 - The pitfall is that there is no diversity in the recommendations. That is, the user gets recommendations that are very familiar to her, since the recommended items are similar to those in her item profile
 - TF-IDF is widely used
- ***Hybrid algorithms*** attempt to combine Collaborative filtering with Content-based filtering. The combination of content with rating data helps capture more effective correlations between users or items, which yields more accurate recommendations.

Recommendation System Types

Other types of recommender systems proposed (by Burke) in the literature :

- ***Demographic recommendation***, which classifies the users according to the attributes of their personal profile, and makes recommendations based on demographic classes
- ***Utility-based recommendation***, which makes suggestions based on a computation of the utility of each item for a user, for whom a utility function has to be stored
- ***Knowledge-based recommendation***, which suggests items based on logical inferences about user preferences. A knowledge representation (e.g. rules) about how an item meets a particular user need is necessary.

Utility matrix sample

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

The movie names are HP1, HP2, and HP3 for Harry Potter I, II, and III, TW for Twilight, and SW1, SW2, and SW3 for Star Wars episodes 1, 2, and 3. The users are represented by capital letters A through D

A utility matrix representing ratings of movies on a 1–5 scale

Blanks represent the situation where the user has not rated the movie. In practice, the matrix would be even sparser

Recommendation Problem

- *Object of the decision.* That is, defining the object upon which the decision has to be made and the rationale of the recommendation decision
- *Family of criteria.* That is, the identification and modelling of a set of criteria that affect the recommendation decision, and which are exhaustive and non-redundant
- *Global preference model.* That is, the definition of the function that aggregates the marginal preferences upon each criterion into the global preference of the decision maker about each item
- *Decision support process.* That is, the study of the various categories and types of recommender systems that may be used to support the recommendation decision maker, in accordance to the results of the previous steps

Recommendation Capabilities

- *Choice*, which involves choosing one item from a set of candidates
- *Sorting*, which involves classifying items into pre-defined categories
- *Ranking*, which involves ranking items from the best one to the worst one
- *Description*, which involves describing all the items in terms of performance upon each criterion

Recommendation technique used in TEL

Collaborative filtering (CF) techniques				
Name	Short description	Advantages	Disadvantages	Usefulness for TEL
1. User-based CF	Users that rated the same item similarly probably have the same taste. Based on this assumption, this technique recommends unseen items already rated by similar users.	<ul style="list-style-type: none"> – No content analysis – Domain-independent – Quality improves over time – Bottom-up approach 	<ul style="list-style-type: none"> – New user problem – New item problem – Popular taste – Scalability – Sparsity – Cold-start problem 	<ul style="list-style-type: none"> – Benefits from experience – Allocates learners to groups (based on similar ratings)
2. Item-based CF	Focus on items, assuming that items rated similarly are probably similar. It recommends items with highest correlation (based on ratings to the items).	<ul style="list-style-type: none"> – No content analysis – Domain-independent – Quality improves over time – Bottom-up approach 	<ul style="list-style-type: none"> – New item problem – Popular taste – Sparsity – Cold-start problem 	<ul style="list-style-type: none"> – Benefits from experience
3. Stereotypes or demographics CF	Users with similar attributes are matched, then recommends items that are preferred by similar users (based on user data instead of ratings).	<ul style="list-style-type: none"> – No cold-start problem – Domain-independent 	<ul style="list-style-type: none"> – Obtaining information – Insufficient information – Only popular taste – Obtaining metadata information – Maintenance ontology 	<ul style="list-style-type: none"> – Allocates learners to groups – Benefits from experience – Recommendation from the beginning of the RS

Recommendation technique used in TEL

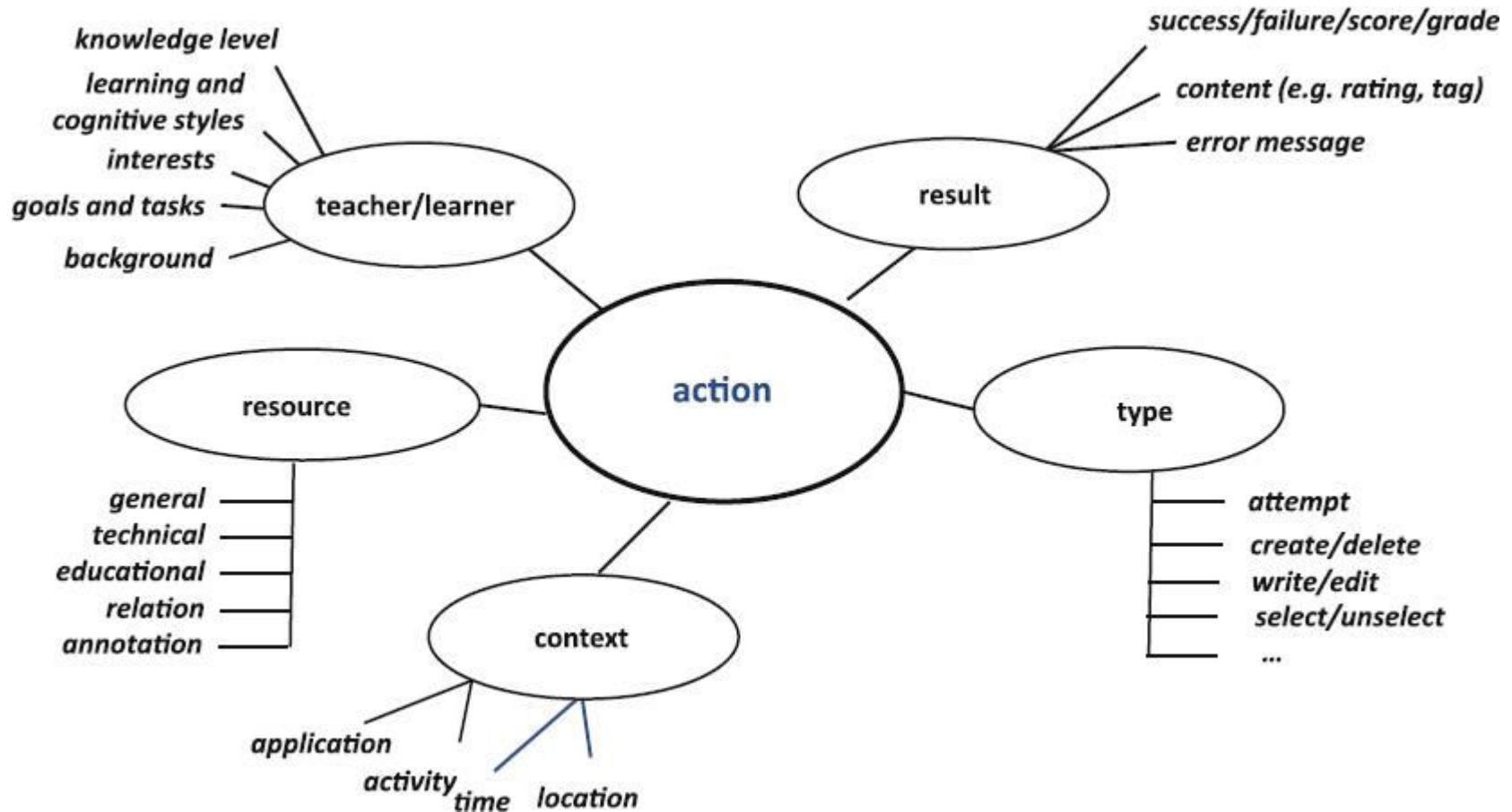
Content-based (CB) techniques				
Name	Short description	Advantages	Disadvantages	Usefulness for TEL
4. Case-based reasoning	<p>Assumes that if a user likes a certain item, (s)he will probably also like similar items.</p> <p>Recommends new but similar items.</p>	<ul style="list-style-type: none"> – No content analysis – Domain-independent – Quality improves over time 	<ul style="list-style-type: none"> – New user problem – Overspecialization – Sparsity – Cold-start problem 	<ul style="list-style-type: none"> – Keeps learner informed about learning goal – Useful for hybrid RS
5. Attribute-based techniques	<p>Recommends items based on the matching of their attributes to the user profile. Attributes could be weighted for their importance to the user.</p>	<ul style="list-style-type: none"> – No cold-start problem – No new user / new item problem – Sensitive to changes of preferences – Can include non-item related features – Can map from user needs to items 	<ul style="list-style-type: none"> – Does not learn – Only works with categories – Ontology modeling and maintenance is required – Overspecialization 	<ul style="list-style-type: none"> – Useful for hybrid RS – Recommendation from the beginning

Recommendation technique used in TEL

Data-Mining (DM) techniques

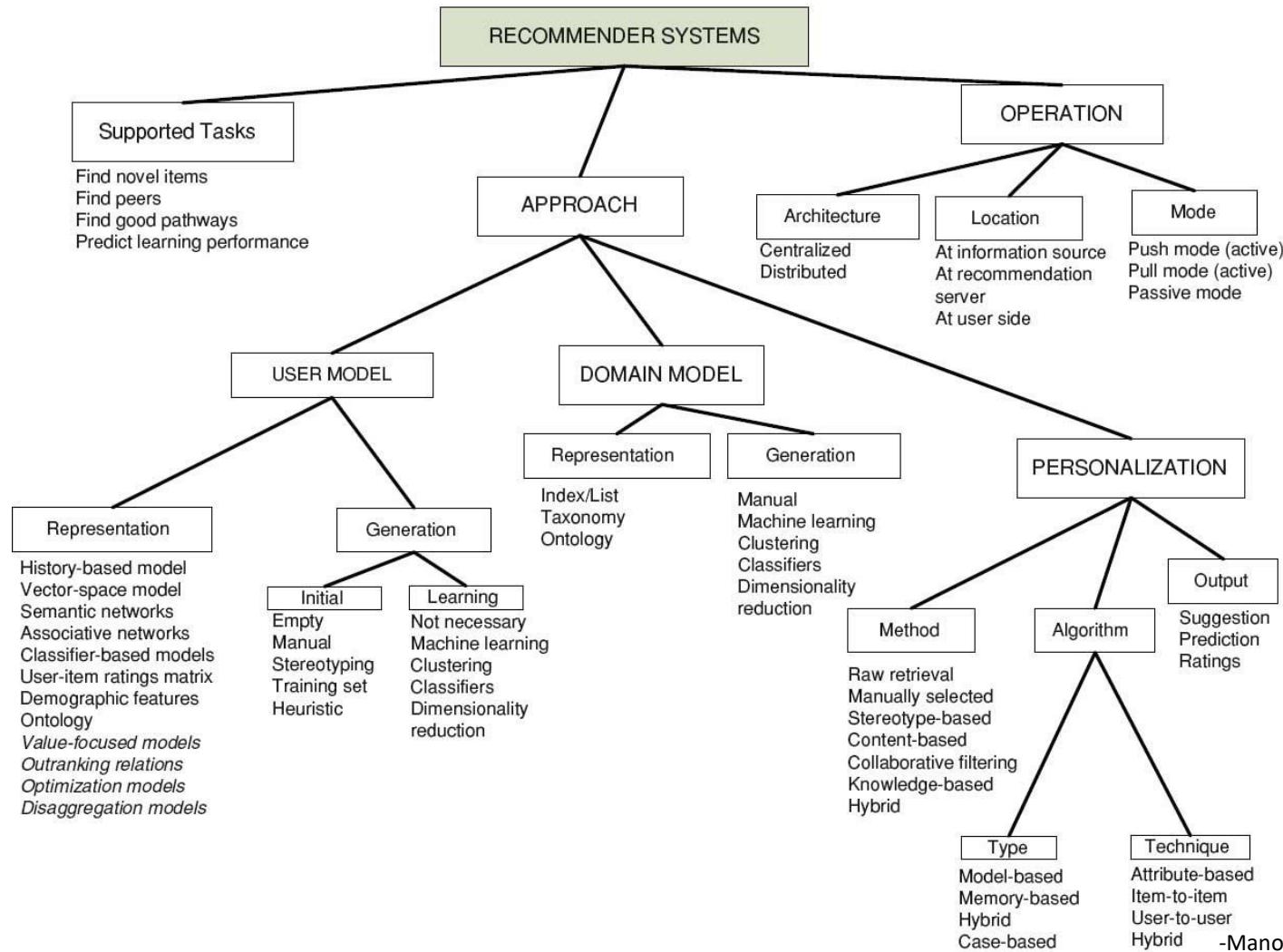
6. Decision Trees (C4.5, ID3)	A decision tree represents a set of classifications created from a set of rules. They start from a single classification and branch out based on classification rules mined from the data.	<ul style="list-style-type: none"> – Easy to understand – High representation power 	<ul style="list-style-type: none"> – Overspecialisation in small datasets – Can become very broad 	<ul style="list-style-type: none"> – Visualize differences of learners from the data – Alternative approach to expert driven ontologies
7.K-Nearest Neighbor (Isodata, Forgy)	Does not build an explicit model instead exams the categories of the K -most similar data points. K -means is often used in TEL recommenders to compute similarity of vector-based approaches.	<ul style="list-style-type: none"> – Simple approach only – Two parameters to select – Robust to noise – High representation power 	<ul style="list-style-type: none"> – Difficult to select distance function d – Irrelevant data needs to be removed – Slower than model-based 	<ul style="list-style-type: none"> – Recommend similar peers, or contents to learners – Cluster learners in groups
8. Vector-based models (TF-IDF, Singular value decomposition, Matrix Factorisation)	Vector-based approaches characterise items and users as vectors of factors in a 3D space. A high correlation between an item and a user can be used as recommendation but also predictions can be created.	<ul style="list-style-type: none"> – Suitable for sparse datasets – Can take temporal differences into account – Can take various implicit information into account does not need explicit ratings 	<ul style="list-style-type: none"> – Content depended (Items with same context but different terms are not matched) – User keywords have to match semantic space 	<ul style="list-style-type: none"> – Useful to monitor and predict learner performance – Can adapt to increased knowledge level of learners – Can mark learning resources that are not popular anymore

TEL variables



- K. Verbert, N. Manouselis, H. Drachsler, E. Duval

Framework for the analysis of Recommender Systems



Prescribed Text Books

	Author(s), Title, Edition, Publishing House
	Recommender Systems for Learning by Nikos Manouselis, Hendrik Drachsler, Katrien Verbert and Erik Duval Springer © 2013
	Recommender Systems for Location-based Social Networks by Panagiotis Symeonidis, Dimitrios Ntempos and Yannis Manolopoulos Springer © 2014



9.2 Fraud Detection

Why study Fraud?

- Fraud can be defined as a criminal activity, involving false representations to gain an unjust advantage (*Concise Oxford Dictionary*)
- The Association of Certified Fraud Examiners estimates that globally organizations lose about 5% of their revenues to fraud. If this were to hold true for all organizations contributing to the Gross World Product of about \$90.52 trillion for 2019 (IMF estimate), fraud losses could be as high as \$4.5 trillion

<https://www.acfe.com/surveys-and-statistics.aspx>

Fraud in Aspects of Business

- Fraud can occur in many aspects of business, for e.g.:
 - **Credit card fraud:** Stealing or counterfeiting credit card numbers, or nonpayment of accounts
 - **Application fraud:** Untrue statements on a credit application, leading to assignment of an artificially low credit risk
 - **Claim fraud:** Submitting inflated or false claims
 - **Life insurance:** False or "engineered" death claims
 - **Health care fraud:** False billings by health care providers
- etc.

Issues with Fraud Detection

- Fraud is usually a rare event. Identifying fraud is difficult because of its rarity and because its very nature is stealthy.
- We need accurate models to make effective detection.
 - The vast majority of the records (i.e., 99.9%) may be legitimate. Only 0.1% of the records may be fraudulent. Here a 99% accurate model will lead to too many false alarms.
 - Say we have million transactions. As per above, 1000 are fraudulent. With 99% accuracy, i.e. 1% inaccuracy (false positives, false negatives), total alarms will be 1% of 0.999 million false alarms and 990 (99% of 1000) true alarms.
Total alarms = $9990 + 990 = 10980$, out of which more than 90% are false alarms.
- Often, the extra accuracy is associated with higher cost, but the cost of *not* doing so may be much higher

Issues with Fraud Detection

- **Fraud is Evolving**

- Fraudsters may adapt quickly to many fraud detection methods, by devising novel and increasingly subtle ways to get away with it. Also, fraud detection schemes must evolve also to try to keep up with (and get ahead of) fraudsters

- **Large Data Set Processing Needed**

- Large credit card issuers like Capital One may process billions of transactions per year. Even a very small percentage of fraud among these billions of transactions can result in proportionately large losses.
- Telecom companies handle billions of calls in a month

- **The Fact of Fraud is Not Always Known during Modeling**

- We need to use both supervised and unsupervised methods to detect fraud.

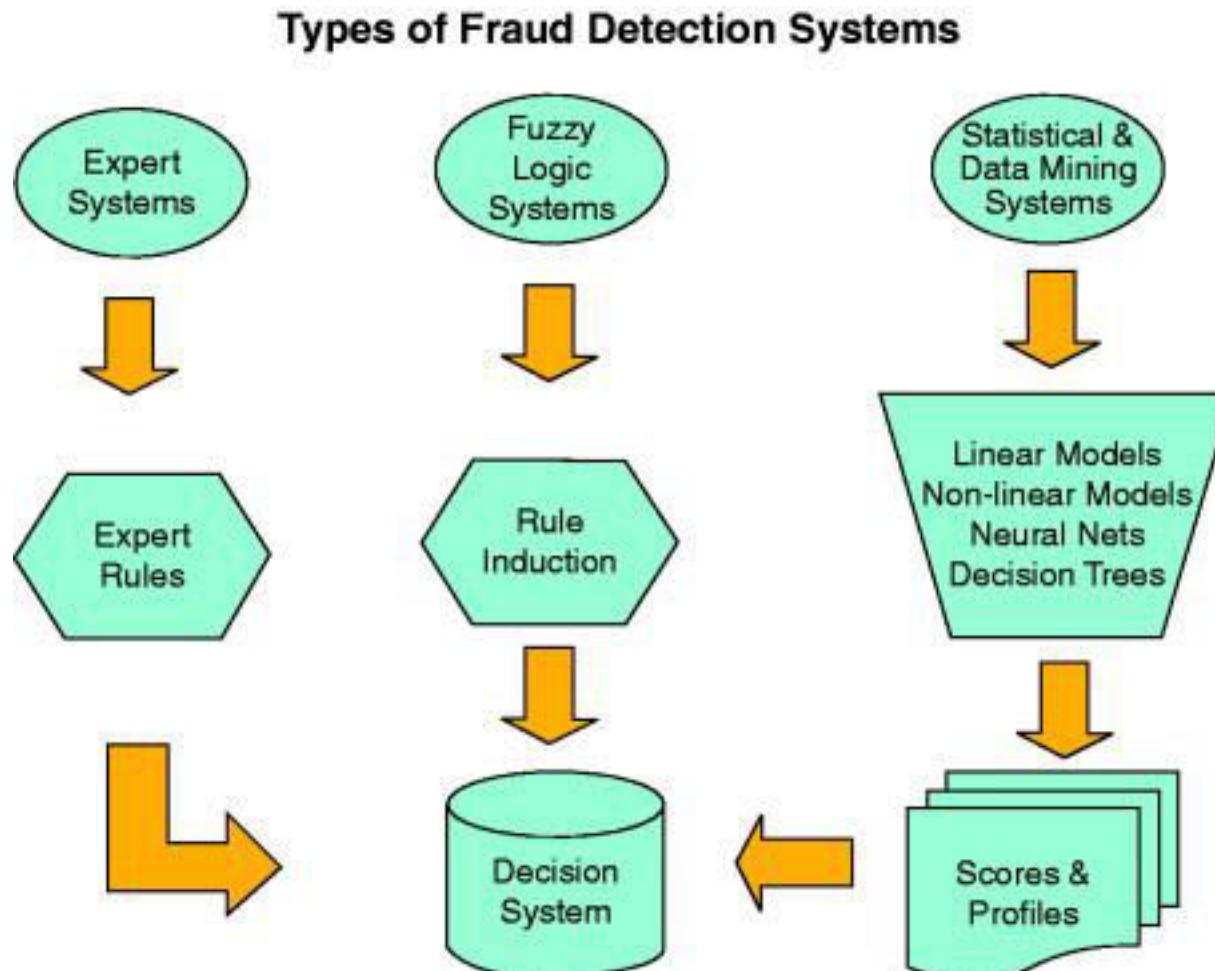
- **Fraud is Very Complex**

- The complexity is partly due to the fraudster's need for stealth and secrecy, and partly due to the intentional obfuscation of the trail of evidence indicating fraud

Issues with Fraud Detection

- **Fraud Detection May Require the Formulation of Rules Based on General Principles, "Red Flags," Alerts, and Profiles**
 - **General principle:** The incidence of fraud is more likely when the opportunity is high and the potential gains are large.
 - **A "red flag":** A large number of accidents or claims is made by one individual
 - **An alert:** A new product is introduced before fraud management systems are put in place
- **Fraud Detection Requires Both Internal and External Business Data**
 - Internal data describing their business events (selling things or providing services)
 - External data such as demographic data, firmographic data (profile of businesses), psychographic data (people with various attitudinal and philosophical views)
- **Very Few Data Sets and Modeling Details are Available**
 - Fraud data sets and modeling methodologies are tightly kept secrets. Companies do not share with anyone.

Types of fraud models



Types of Fraud Models

- Early fraud models employed expert systems to detect fraudulent events. An expert system is a collection of expert opinions on a number of decision criteria. These systems induced rules from the responses of a group of experts in the field. These rules can be coordinated into a flow chart leading to a decision.
 - The problem with expert systems is that they are based on subjective inputs that may be contradictory
- Subsequent fraud detection systems used automated rule induction engines, based decision tree technology, and fuzzy logic. Some of these fraud detection systems are still marketed today (iPrevent by Brighterion).

Types of Fraud Models

- The Fair Isaac fraud detection systems Falcon Fraud Manager, eFalcon, and LiquidCredit Fraud Solution are built around a sophisticated system of predictive variables derived from extensive historical customer data.
 - These predictors have been selected by many years of modeling fraud in many companies. The variables are submitted to a powerful backpropagation neural net.

Supervised Methods for Fraud Detection

Several elements are crucial to the successful supervised fraud model

- The fraud event and the relationship of that event to specific transactions or responses of the fraudster must be accurately identified
- Historical data of past transactions or responses must be available to derive powerfully predictive variables
- Profiles of the past behavior and actions of both the fraudsters and the nonfraudsters must be built and employed in the modeling methodology
- Predictive variables need to be identified for each type of fraud.

Detection of money laundering and other financial crimes

To detect money laundering and other financial crimes, it is necessary to integrate information from multiple databases such as bank transaction databases, and federal or state crime history

Multiple data analysis tools can then be used :

- Data visualization tools to display transaction activities using graphs by time and by groups of customers
- Linkage analysis tools to identify links among different customers and activities
- Classification tools to filter unrelated attributes and rank the related ones
- Clustering tools to group different cases
- Outlier analysis tools to detect unusual amounts of fund transfers or other activities, and
- Sequential pattern analysis tools to characterize unusual access sequences

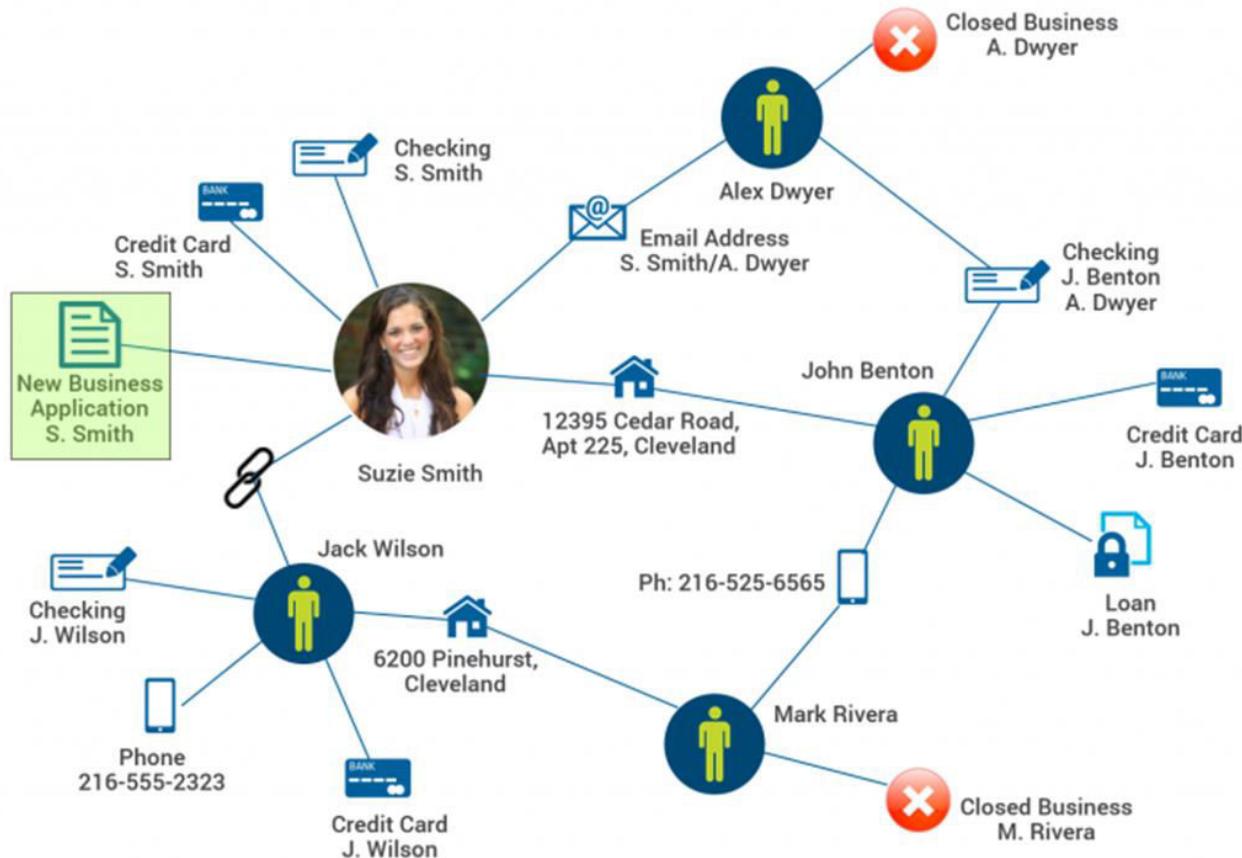
Identifying Tax Fraud through Social Network Analysis

- SNA(Social Network Analysis) is an analytic approach of correlating people, entities and relationships to determine how tightly an individual or business is related to others who have known compliance issues
- These relationships can be from shared
 - phone numbers,
 - physical addresses,
 - bank accounts,
 - credit cards, or
 - any other connection
- Graph analytics can give insights into members of the network

Opportunities for Tax Agencies in SNA

- Tax and revenue agencies to take advantage of SNA tools, across registration, audit and collections business areas
- **Improper registrations.** A tax evader closes a business and the business re-opens (typically owned by a relative of the original owner) at the same or a nearby location. i.e. the owner stays in business by opening a similar (or identical) business with a different legal name and a different legal owner (e.g., a spouse, parent or another relative).
 - Very difficult to catch by manual efforts. The challenge for the tax agency is the owner will have changed and won't be an exact match to the previous business.

Opportunities for Tax Agencies in SNA



- In the example, Suzie Smith has a direct relationship with one closed business and a second degree relationship with another closed business.
- The state can deny the new registration or conduct additional investigations in a cost effective manner.

Opportunities for Tax Agencies in SNA

- **Identify Fraud Rings.** The SNA can identify related businesses. i.e. businesses with related addresses, bank accounts, phone numbers, email addresses, or other identifying characteristics.
 - SNA can help a revenue agent can identify additional individuals and/or businesses that may be related to the same fraud ring, saving the investigator time and effort.
- **Assisting with Locating a Delinquent Debtor.** SNA's are frequently used during the debt collection process to identify related individuals. SNA can greatly enhance and automate this effort, by finding people who share the same physical address, phone number, email, etc
 - Earlier collectors have utilized manual tools along these lines for years, contacting next-door neighbors (for example) in an attempt to locate a debtor.

Opportunities for Tax Agencies in SNA

- **Finding Successor Businesses.** When a business ceases operation, the business can re-open in a new location or under new ownership. If the original business owes money, the government can in many cases pursue that debt if there is a successor business
 - This can save the collector a significant amount of time for what otherwise would require significant manual research.
- Tax agencies are data rich organizations, and analytics solutions like Social Network Analysis will allow them to identify more fraud and potential non-compliance situations well before a liability occurs

Prescribed Text Books

	Author(s), Title, Edition, Publishing House
	Fraud Detection; Handbook of Statistical Analysis and Data Mining Applications by Robert Nisbet, John Elder and Gary Miner Academic Press 2009
	Data Mining: Concepts and Techniques, Second Edition by Jiawei Han, Micheline Kamber and Jian Pei Morgan Kaufmann Publishers
	www.fico.com (Fair Issac Corporation)



9.3 Sentiment Analysis

Sentiment Analysis

- Monitoring what consumers are saying about a company's brands and products and how they are expressing their opinions and sentiments to others has always been important to businesses.
 - Until the last century, businesses typically used surveys and focus groups from time to time to gauge and track consumer sentiments.
 - With the widespread adoption of the Internet, the proliferation of social media channels (such as Twitter, Facebook, and others), and the abundant opportunity for consumers to express their opinions and sentiments, monitoring sentiment continuously has become more critical
-
- "Conventional marketing wisdom long held that a dissatisfied customer tells ten people; but in the age of new social media, he or she has the tools to tell millions,"
 - Paul Gillin, author of *The New Influencers: A Marketer's Guide to the New Social Media*

Sentiment Analysis

- The basic task involved in sentiment analysis is identifying and quantifying the polarity or valence of sentiments (such as positive, negative, neutral, or mixed) expressed typically in written opinions, expressions, reviews, comments, and so on
- It involves many of the text analytics steps such as
 - tokenization,
 - sentence identification,
 - part-of-speech tagging,
 - and so on

Sentiment Analysis

- Need to identify statements that convey sentiment
 - "It is an amazing TV" – conveys sentiment
 - "Do not buy this TV" – conveys sentiment
 - "Which is the best TV?" – conveys no sentiment
- Depending on the context, sentences can be non-comparative (where opinion is restricted to one thing) or comparative (where multiple things might be compared).

Example of a Review

Consider the following example of a review by a customer for a TV:

The TV is **wonderful**. Great size, great picture, easy interface. It makes a **cute** little song when you **boot** it up and when you shut it off. I just want to point out that the **43"** **does not** in fact play videos from the USB. This is really **annoying** because that was one of the major perks. I wanted from a new **TV** Looking at the product description now. I realize that the feature list applies to the **X758** series as a whole, and that each model's capabilities are listed below. Kind of a **dumb** oversight on my part, but it's equally **stupid** to put a description that does not apply on the listing for a very specific model

Granularity of Sentiment

Sentiment analysis starts with determining whether a text contains an opinion (sentiment). If it does contain sentiment, at what granularity level does the sentiment exist?

- *Document level:* At this level, the task is to figure out whether the entire document can be classified as positive or negative
 - This is possible only if the document involves a single entity (such as the TV in the previous example).
- *Sentence level:* At this level, the task is to classify each sentence in a document as a positive, negative, or mixed sentiment sentence.
 - In the previous example, the first sentence, expresses positive sentiment. The third statement, expresses negative sentiment

Granularity of Sentiment

Sentiment granularity can also be looked at from the object side:

- *Entity (or Object) and Attribute (or Aspect or Feature) level:* An entity is typically the target of the opinion. However, in many sentences, the sentiments reflect the reviewer's opinions about attributes (or aspects or features) of the entity
 - "Great size, great picture, easy interface", express positive sentiment for three specific attributes "size, picture, and interface" of the entity, the TV

Challenges in Sentiment Analysis - NLP

Sentiment analysis starts with text data. So it has all of the typical natural language processing (NLP) problems associated with text analytics, viz.

- identifying part-of-speech tags,
- disambiguating terms and lexicons,
- correcting spelling errors, etc.

In addition to words, there are idiom lexicons— e.g. "costs an arm and a leg" that embody sentiments.

In general, the difficulty with correctly identifying sentiments increases as you move from general to context-dependent to idiom lexicons in texts

Challenges in Sentiment Analysis – Opinion Words

Sentiment analysis needs to correctly identifying opinion words that express positive or negative sentiments.

- There are opinion words whose polarity is always the same, e.g. the word "beautiful," always expresses a positive sentiment.
- But, there are also context-dependent lexicons in which the polarity of the word depends on the domain or context, e.g. the word "small" can be positive or negative depending on the context. The sentence, "The size seems *small*." can be positive for a USB flash drive with 1 TB capacity. But, the same sentence can be interpreted as negative if the context is an LED big-screen TV.

Challenges in Sentiment Analysis – Type of Text

Further challenges in conducting sentiment analysis come from the nature of the text.

- For example, tweets are short, and they are typically focused on one topic only. In that sense, they are easier to analyze. But, tweets often contain a lot of special meaning characters, such as RT (retweets), hashtags (#), emoticons (such as smiley faces), that need to be handled carefully.
- Customer reviews are typically on one entity or object. Therefore, there is less ambiguity in the entity detection task when analyzing reviews.
- Analysis of discussions, free-flow comments, and blog postings is often the hardest because they typically cover multiple entities, make comparisons instead of expressing direct opinions, use a lot of sarcasm, etc.

Unsupervised versus Supervised Sentiment Analysis

- The sentiment analysis can be formulated as a supervised or an unsupervised mining problem, depending on whether there are known examples of documents belonging to positive or negative sentiments.
- Unsupervised sentiment analysis involves the application of a sentiment lexicon of opinion-related positive or negative terms to evaluate text in the document.
- Supervised approach involves machine-learning algorithms (such as support vector machines (SVMs) and neural networks) to textual feature representations to derive the relationships between features of the text segment and the opinions expressed in the document.
 - In many situations, known class examples are created by experts who read the documents or use rules, e.g. if a text review's numeric rating is four or more stars, then the review is positive.
 - If no known class examples are possible, then analysts have to use an unsupervised classification of sentiments.

Unsupervised versus Supervised Sentiment Analysis

Supervised classification is typically performed at the document level.

- If enough labeled examples are available, commonly used classification models can be trained, validated, and tested to check their performances.
- A good candidate is product review data, which typically has a text review and an overall numeric rating on a scale of one to five stars. Often, a review rating of four to five stars is considered a positive rating, and a review rating of one to two stars is considered a negative rating.
- The main challenge for modelers is to select the inputs from text features such as terms and their frequencies (often weighted or normalized), part-of-speech tags, opinion lexicons (general, context-specific, and idiom), syntactic dependency (from parsing trees), and the handling of negation words (such as "not").

Unsupervised versus Supervised Sentiment Analysis

Unsupervised method is typically applied at the sentence level. There are two types of unsupervised methods: lexicon-based and syntactic-pattern based.

- The *lexicon-based approach* can be used for sentence- and aspect-level sentiment classification. The relationships between opinion words and attributes are identified via dependency relationships obtained through parsing.
 - For example, in the sentence, "The picture quality is outstanding," the opinion word "outstanding" and the attribute "picture quality" share the same dependency relationship with the verb "is."
 - If a clear dependency is not observed between an opinion word and an attribute, then how close an opinion word is to an attribute in a sentence can be used to judge the polarity of the attribute.
 - This process can get very complex, depending on how long the sentence is, how many attributes are being mentioned in the same sentence, whether both positive and negative polarity words are used in the same sentence, whether negation is used, and so on.
 - Once sentiment values are computed for each word-attribute combination, they are typically combined using appropriate normalization or weights to come up with an overall sentiment score.
- The *syntactic pattern-based approach* involves defining part-of-speech tags and the keywords AND, NOR, OR, NOT, BUT, etc.
 - Primarily useful in contextual analysis when performing phrase -level analysis, this method can be used to develop a variety of rules for better accuracy. For example, a simple pattern such as <subject> <NOT> <verb> can be used to extract negative phrases like, "*This <feature> does <not> < work> as advertised.*"

Prescribed Text Books

	Author(s), Title, Edition, Publishing House
	Tutorial BB - Mining Twitter for Airline Consumer Sentiment Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications by Gary Miner et al. Academic Press © 2012
	Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS SAS Institute © 201

Thank You