

Data Mining: Introduction

Lecture Notes for Chapter 1

Introduction to Data Mining

by

Tan, Steinbach, Kumar

Why Mine Data? Commercial Viewpoint

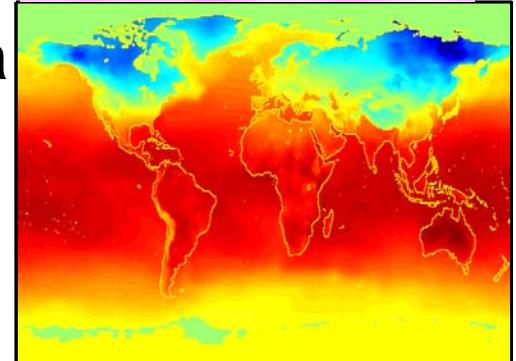
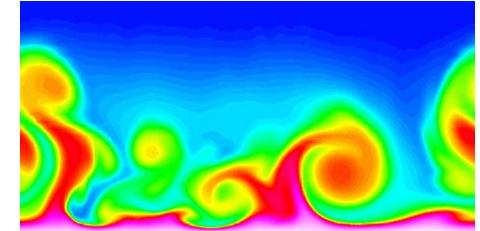
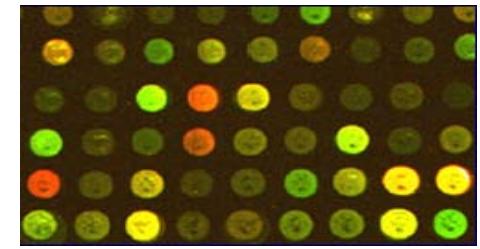
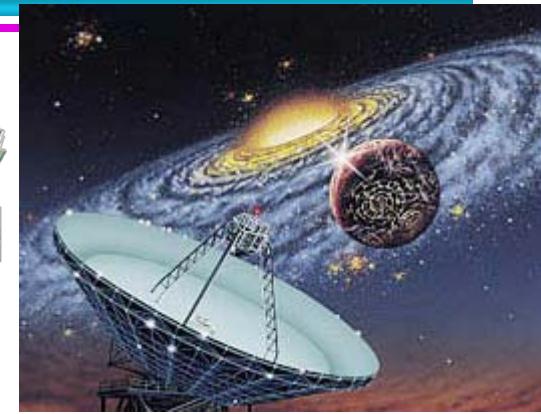
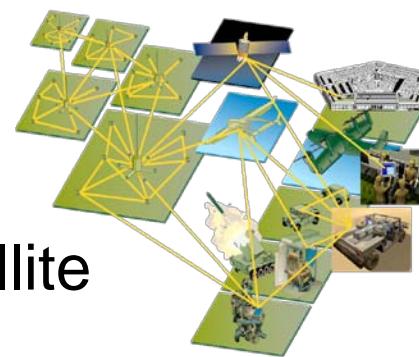
- Lots of data is being collected and warehoused
 - Web data, e-commerce
 - purchases at department/grocery stores
 - Bank/Credit Card transactions



- Computers have become cheaper and more powerful
- Competitive Pressure is Strong
 - Provide better, customized services for an edge (e.g. in Customer Relationship Management)

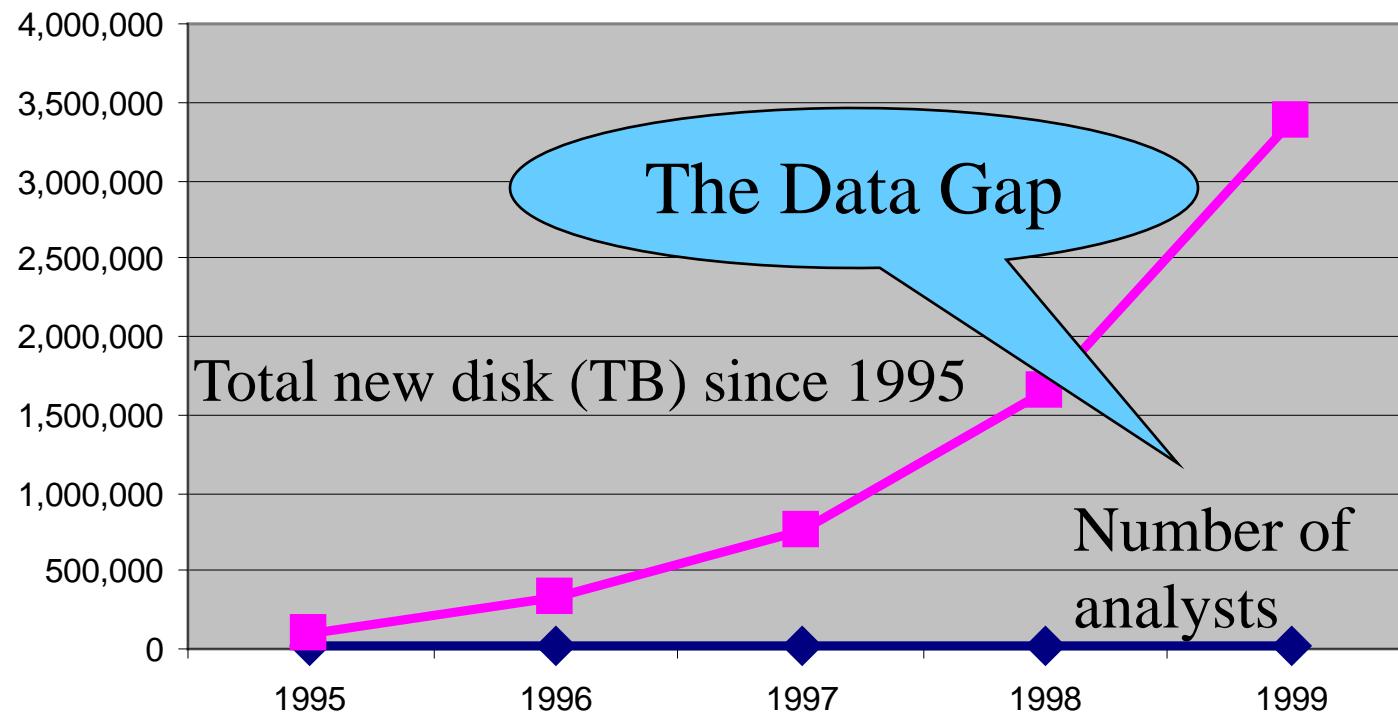
Why Mine Data? Scientific Viewpoint

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



Mining Large Data Sets - Motivation

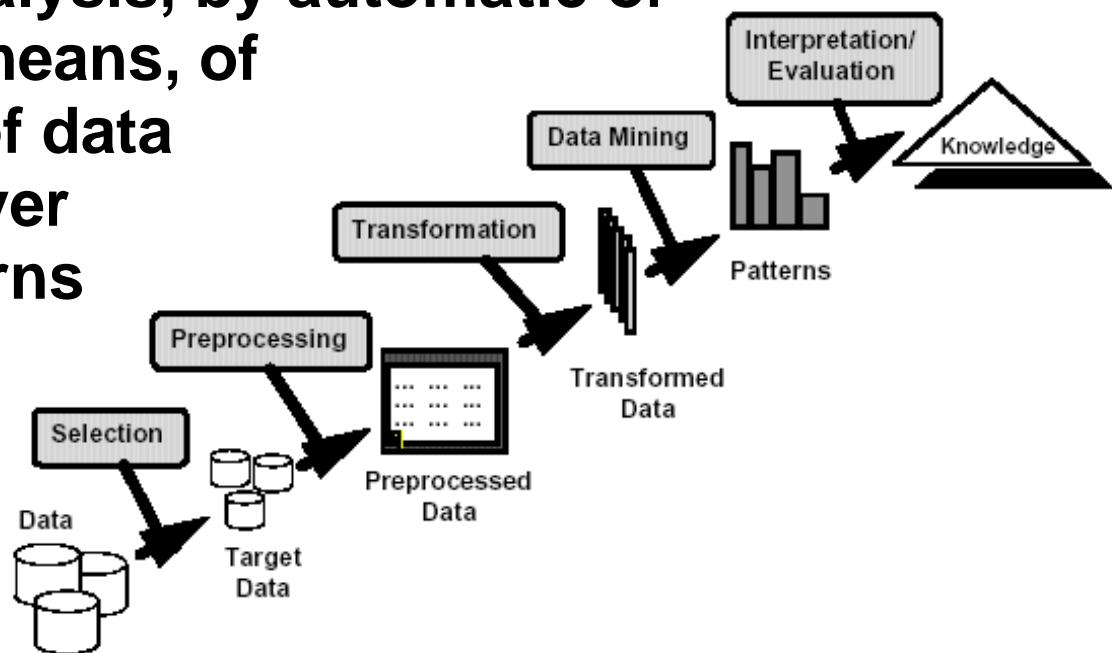
- There is often information “hidden” in the data that is not readily evident
- Human analysts may take weeks to discover useful information
- Much of the data is never analyzed at all



What is Data Mining?

• Many Definitions

- Non-trivial extraction of implicit, previously unknown and potentially useful information from data
- Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns



What is (not) Data Mining?

- **What is not Data Mining?**

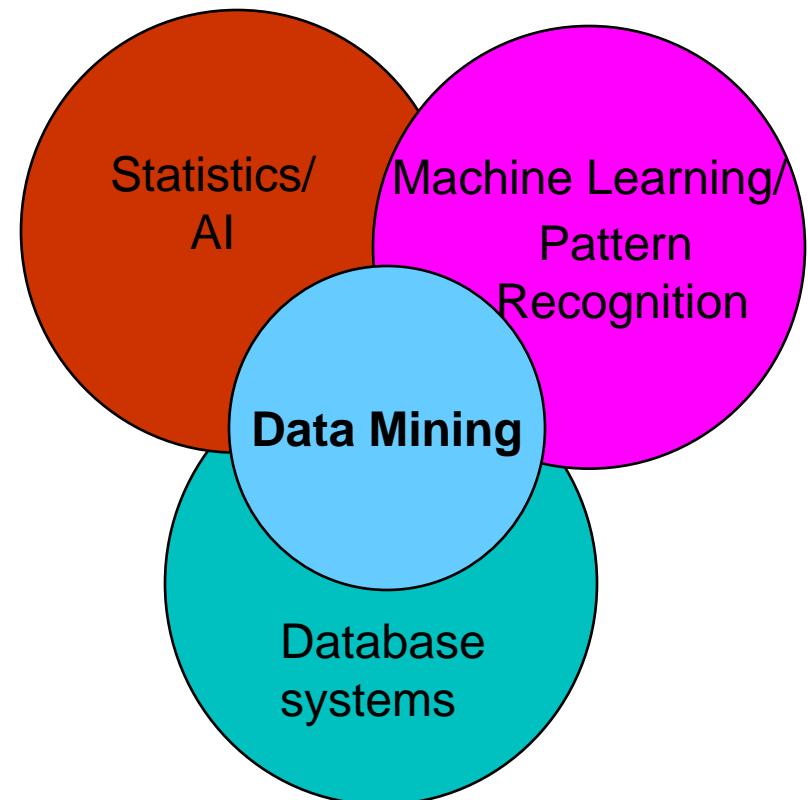
- Look up phone number in phone directory
- Query a Web search engine for information about “Amazon”

- **What is Data Mining?**

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

Origins of Data Mining

- 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 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

Data Mining Tasks...

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

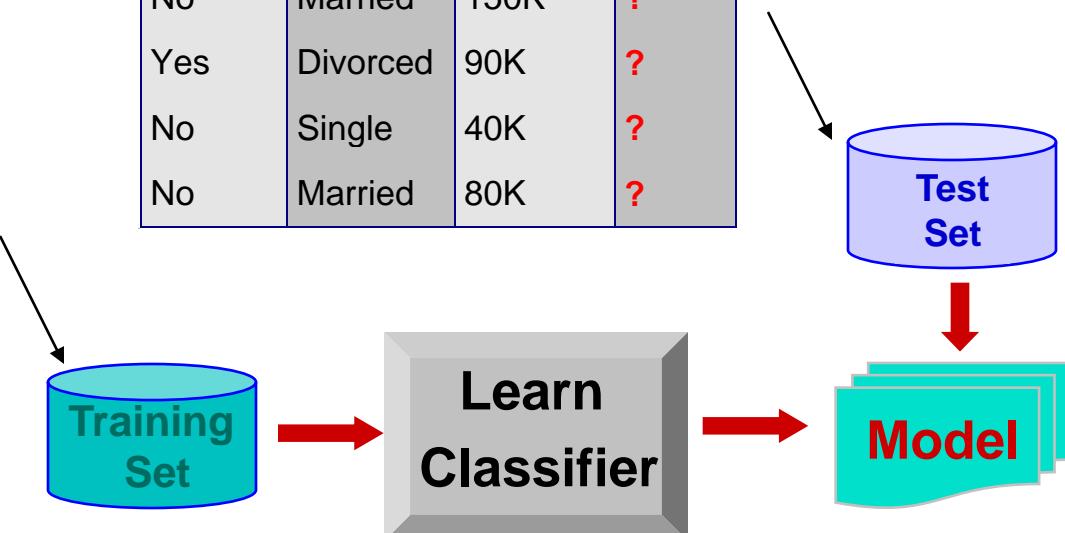
Classification: Definition

- 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

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

Classification: Application 4

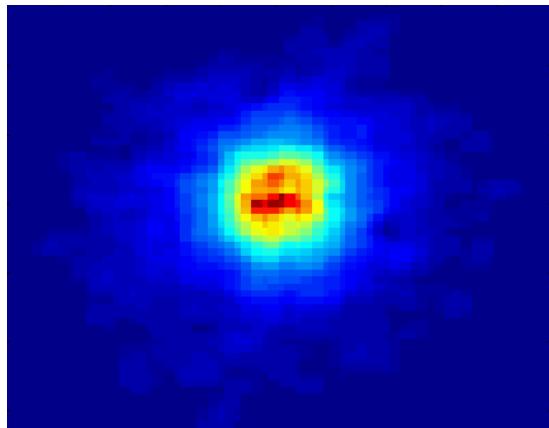
- Sky Survey Cataloging
 - Goal: To predict class (star or galaxy) of sky objects, especially visually faint ones, based on the telescopic survey images (from Palomar Observatory).
 - 3000 images with $23,040 \times 23,040$ pixels per image.
 - Approach:
 - ◆ Segment the image.
 - ◆ Measure image attributes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

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

Classifying Galaxies

Courtesy: <http://aps.umn.edu>

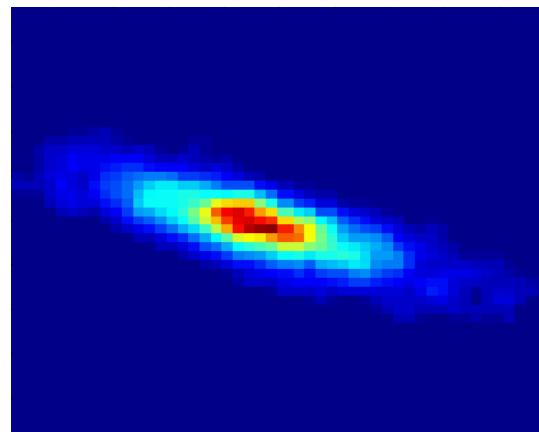
Early



Class:

- Stages of Formation

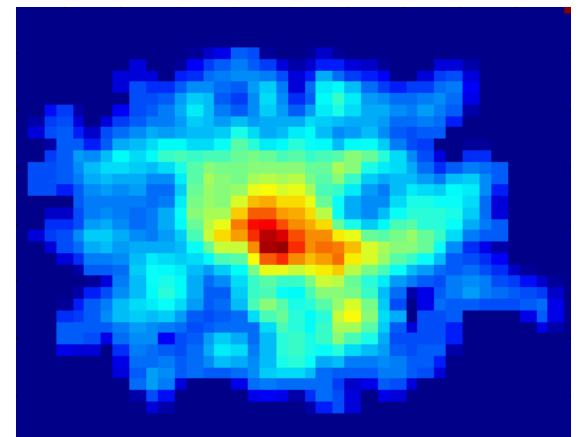
Intermediate



Attributes:

- Image features,
- Characteristics of light waves received, etc.

Late



Data Size:

- 72 million stars, 20 million galaxies
- Object Catalog: 9 GB
- Image Database: 150 GB

Clustering Definition

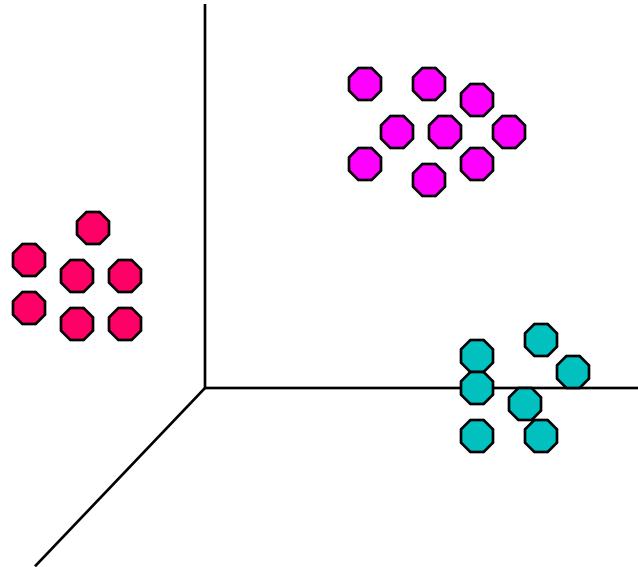
- Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that
 - Data points in one cluster are more similar to one another.
 - Data points in separate clusters are less similar to one another.
- Similarity Measures:
 - Euclidean Distance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

Illustrating Clustering

- Euclidean Distance Based Clustering in 3-D space.

Intracluster distances
are minimized

Intercluster distances
are maximized



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.

Illustrating Document Clustering

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

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

Clustering of S&P 500 Stock Data

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

	<i>Discovered Clusters</i>	<i>Industry Group</i>
1	Applied-Matl-DOWN,Bay-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,Oracle-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

Association Rule Discovery: Definition

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

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

Rules Discovered:

$\{\text{Milk}\} \rightarrow \{\text{Coke}\}$

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

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 2

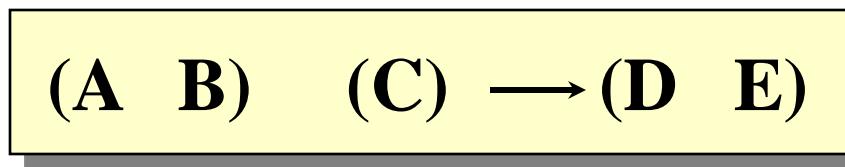
- Supermarket shelf management.
 - Goal: To identify items that are bought together by sufficiently many customers.
 - Approach: Process the point-of-sale data collected with barcode scanners to find dependencies among items.
 - A classic rule --
 - ◆ If a customer buys diaper and milk, then he is very likely to buy beer.
 - ◆ So, don't be surprised if you find six-packs stacked next to diapers!

Association Rule Discovery: Application 3

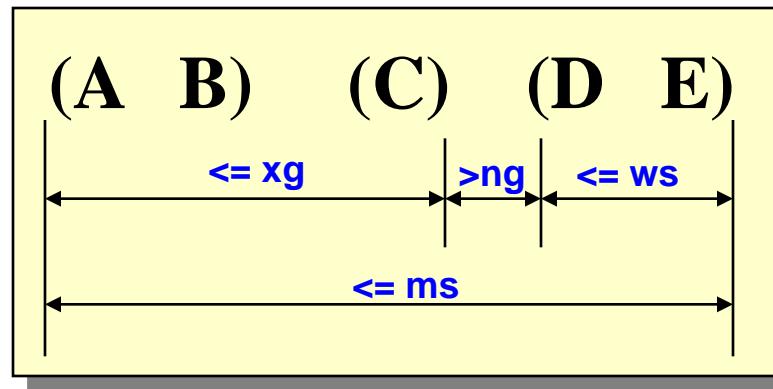
- Inventory Management:
 - Goal: A consumer appliance repair company wants to anticipate the nature of repairs on its consumer products and keep the service vehicles equipped with right parts to reduce on number of visits to consumer households.
 - Approach: Process the data on tools and parts required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

Sequential Pattern Discovery: Definition

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



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



Sequential Pattern Discovery: Examples

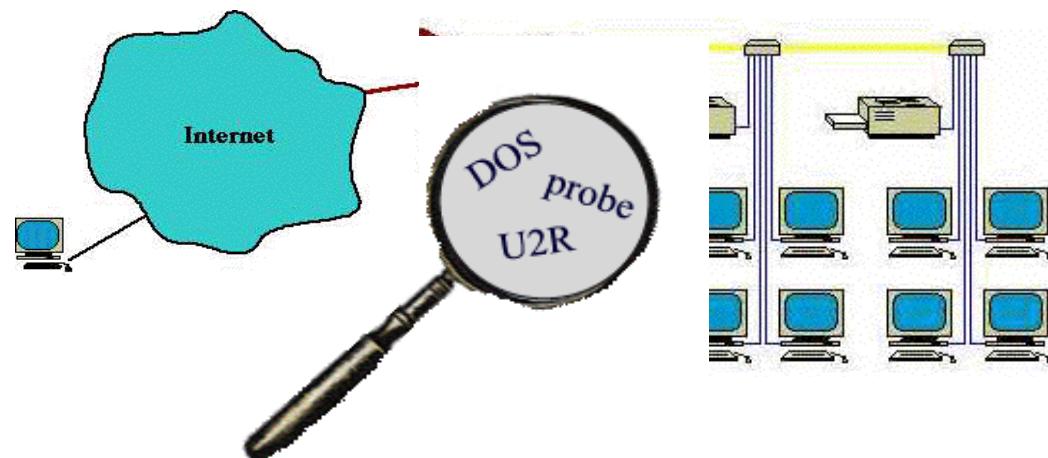
- In telecommunications alarm logs,
 - (Inverter_Problem Excessive_Line_Current)
(Rectifier_Alarm) --> (Fire_Alarm)
- In point-of-sale transaction sequences,
 - Computer Bookstore:
(Intro_To_Visual_C) (C++_Primer) -->
(Perl_for_dummies,Tcl_Tk)
 - Athletic Apparel Store:
(Shoes) (Racket, Racketball) -->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

Regression

- Predict a value of a 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



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

Challenges of Data Mining

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

Data Mining: Data

Lecture Notes for Chapter 2

Introduction to Data Mining

by

Tan, Steinbach, Kumar

What is Data?

- Collection of data objects and their attributes
- An attribute is a property or characteristic of an object
 - Examples: eye color of a person, temperature, etc.
 - Attribute is also known as variable, field, characteristic, or feature
- A collection of attributes describe an object
 - Object is also known as record, point, case, sample, entity, or instance

Attributes

Objects

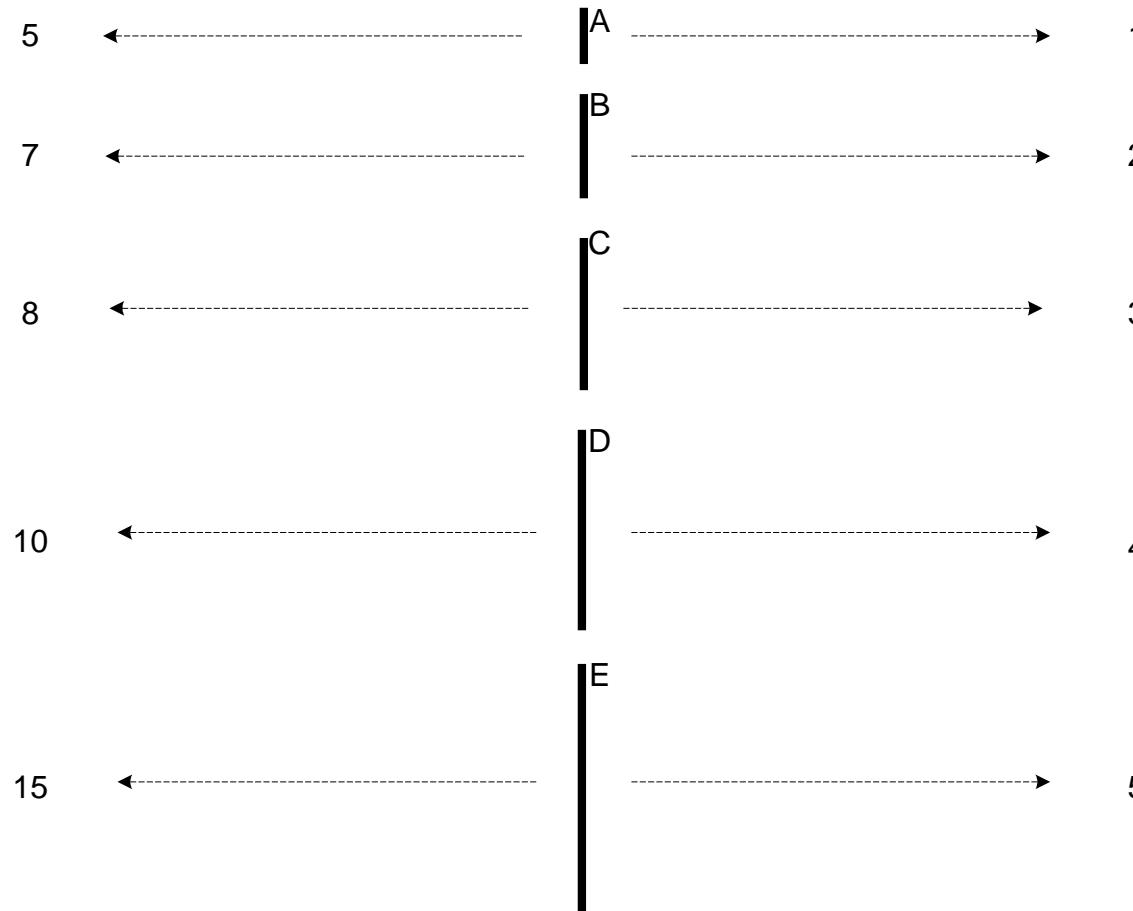
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Attribute Values

- Attribute values are numbers or symbols assigned to an attribute
- Distinction between attributes and attribute values
 - Same attribute can be mapped to different attribute values
 - ◆ Example: height can be measured in feet or meters
 - Different attributes can be mapped to the same set of values
 - ◆ Example: Attribute values for ID and age are integers
 - ◆ But properties of attribute values can be different
 - ID has no limit but age has a maximum and minimum value

Measurement of Length

- The way you measure an attribute is somewhat may not match the attributes properties.



Types of Attributes

- There are different types of attributes
 - Nominal
 - ◆ Examples: ID numbers, eye color, zip codes
 - Ordinal
 - ◆ Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
 - Interval
 - ◆ Examples: calendar dates, temperatures in Celsius or Fahrenheit.
 - Ratio
 - ◆ Examples: temperature in Kelvin, length, time, counts

Properties of Attribute Values

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

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

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

Discrete and Continuous Attributes

- Discrete Attribute
 - Has only a finite or countably infinite set of values
 - Examples: zip codes, counts, or the set of words in a collection of documents
 - Often represented as integer variables.
 - Note: binary attributes are a special case of discrete attributes
- Continuous Attribute
 - Has real numbers as attribute values
 - Examples: temperature, height, or weight.
 - Practically, real values can only be measured and represented using a finite number of digits.
 - Continuous attributes are typically represented as floating-point variables.

Types of data sets

- **Record**

- Data Matrix
- Document Data
- Transaction Data

- **Graph**

- World Wide Web
- Molecular Structures

- **Ordered**

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

Important Characteristics of Structured Data

- Dimensionality
 - ◆ Curse of Dimensionality
- Sparsity
 - ◆ Only presence counts
- Resolution
 - ◆ Patterns depend on the scale

Record Data

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

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

Data Matrix

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

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

Document Data

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

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

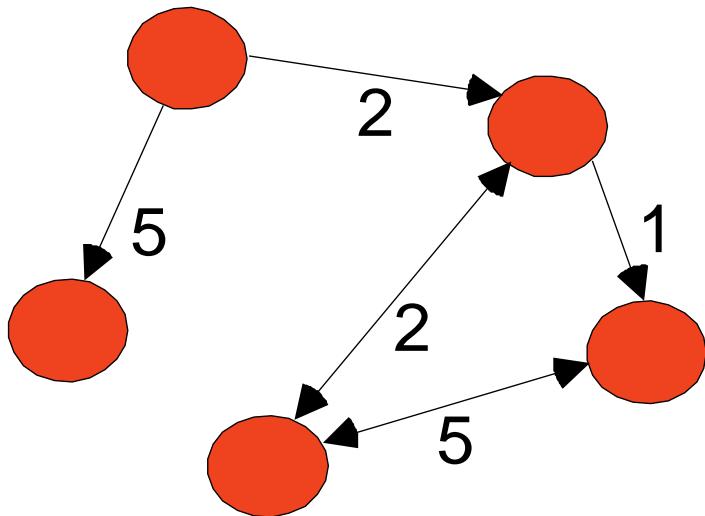
Transaction Data

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

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Graph Data

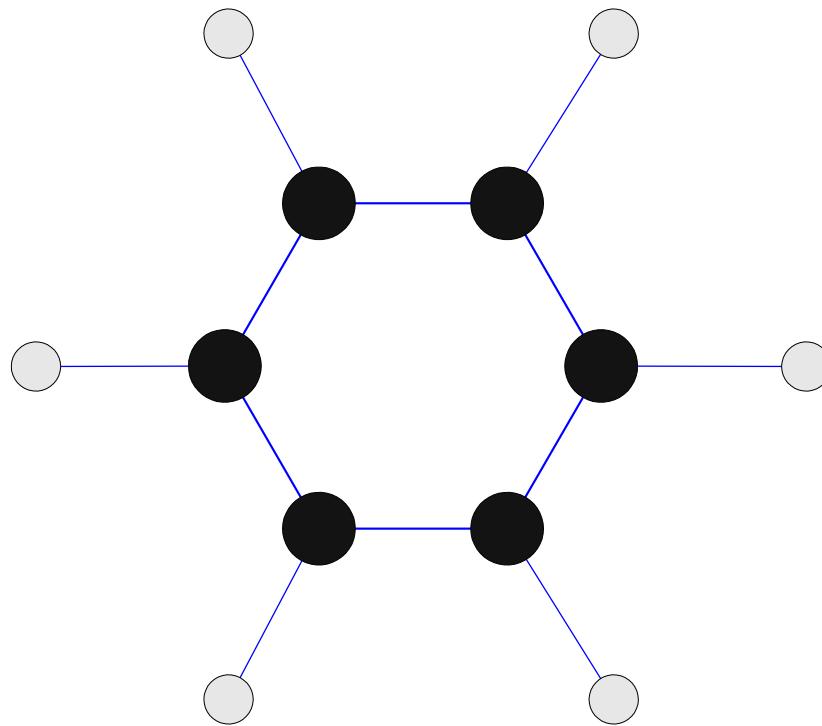
- Examples: Generic graph and HTML Links



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

Chemical Data

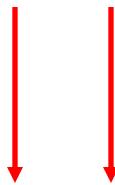
- Benzene Molecule: C_6H_6



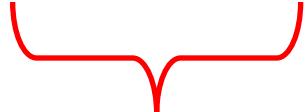
Ordered Data

- Sequences of transactions

Items/Events



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



**An element of
the sequence**

Ordered Data

- Genomic sequence data

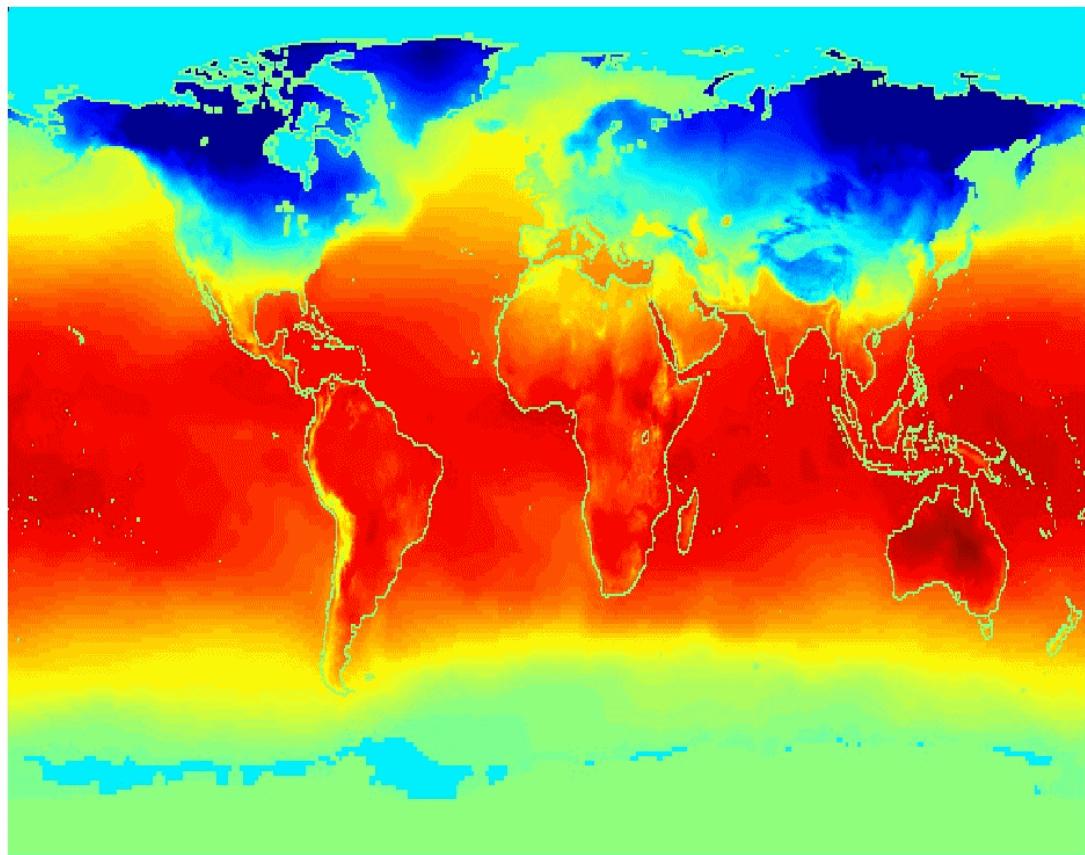
```
GGTTCCGCCTTCAGCCCCGCGCC  
CGCAGGGCCCGCCCCGGCGCCGTC  
GAGAAGGGCCCAGCTGGCGGGCG  
GGGGGAGGCAGGGCCGCCGAGC  
CCAACCGAGTCCGACCAAGGTGCC  
CCCTCTGCTCGGCCTAGACCTGA  
GCTCATTAGGCAGGGCAGCGGACAG  
GCCAAGTAGAACACCGCGAAGCGC  
TGGGCTGCCTGCTGCGACCAGGG
```

Ordered Data

- Spatio-Temporal Data

Average Monthly
Temperature of
land and ocean

Jan

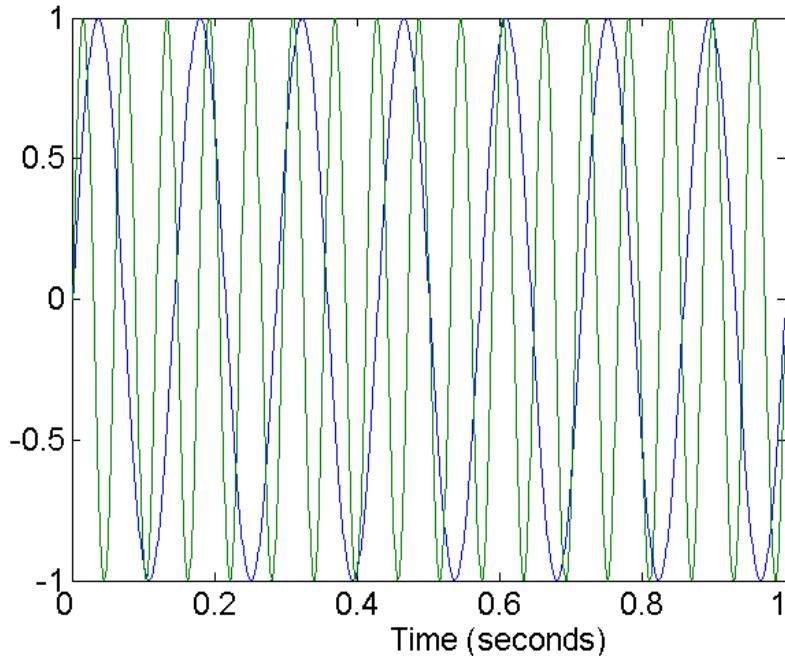


Data Quality

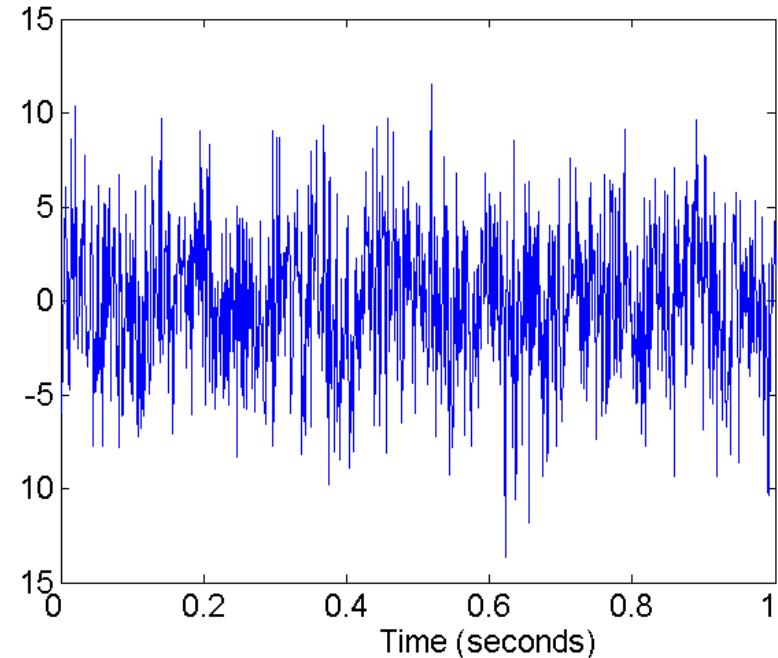
- What kinds of data quality problems?
 - How can we detect problems with the data?
 - What can we do about these problems?
-
- Examples of data quality problems:
 - Noise and outliers
 - missing values
 - duplicate data

Noise

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



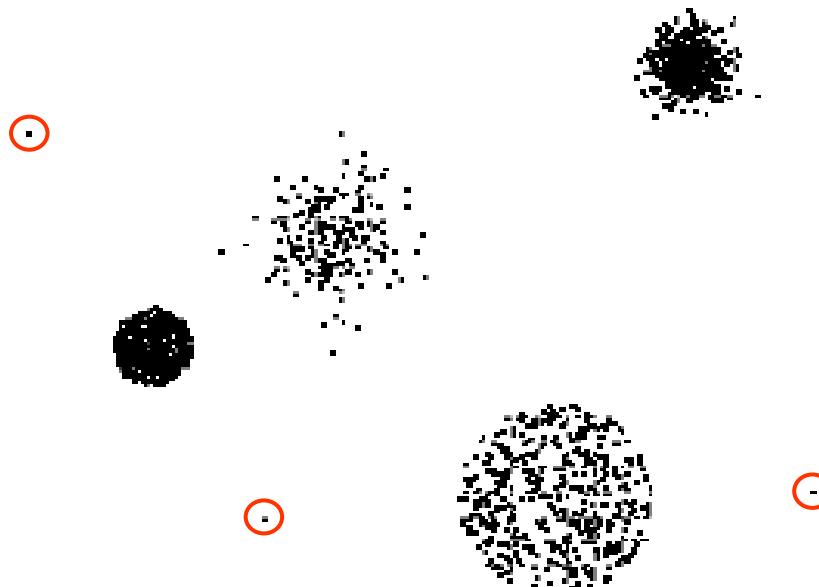
Two Sine Waves



Two Sine Waves + Noise

Outliers

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



Missing Values

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

Duplicate Data

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

Data Preprocessing

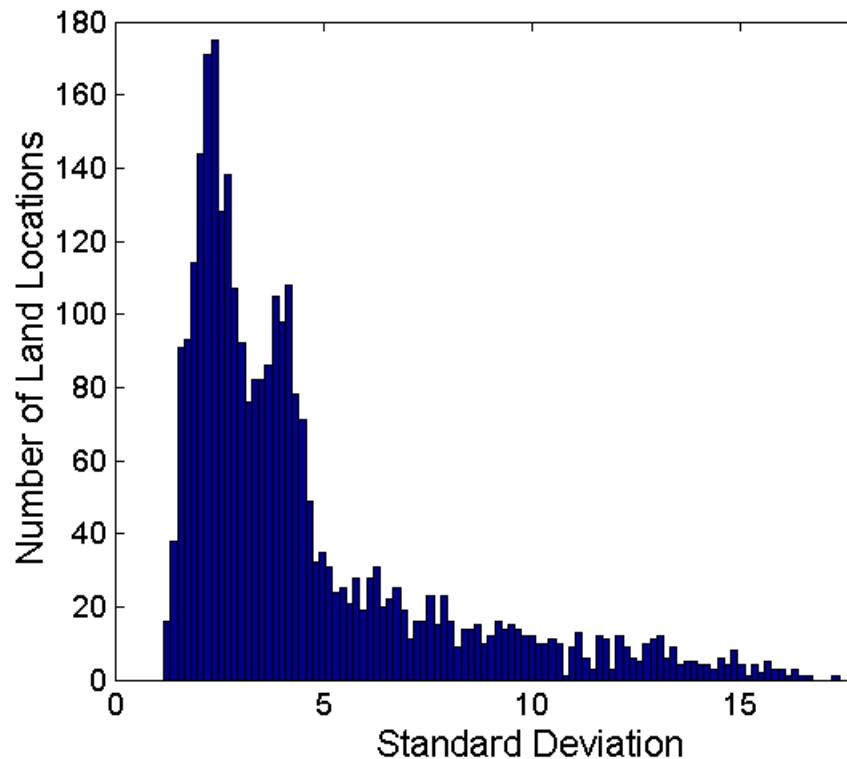
- Aggregation
- Sampling
- Dimensionality Reduction
- Feature subset selection
- Feature creation
- Discretization and Binarization
- Attribute Transformation

Aggregation

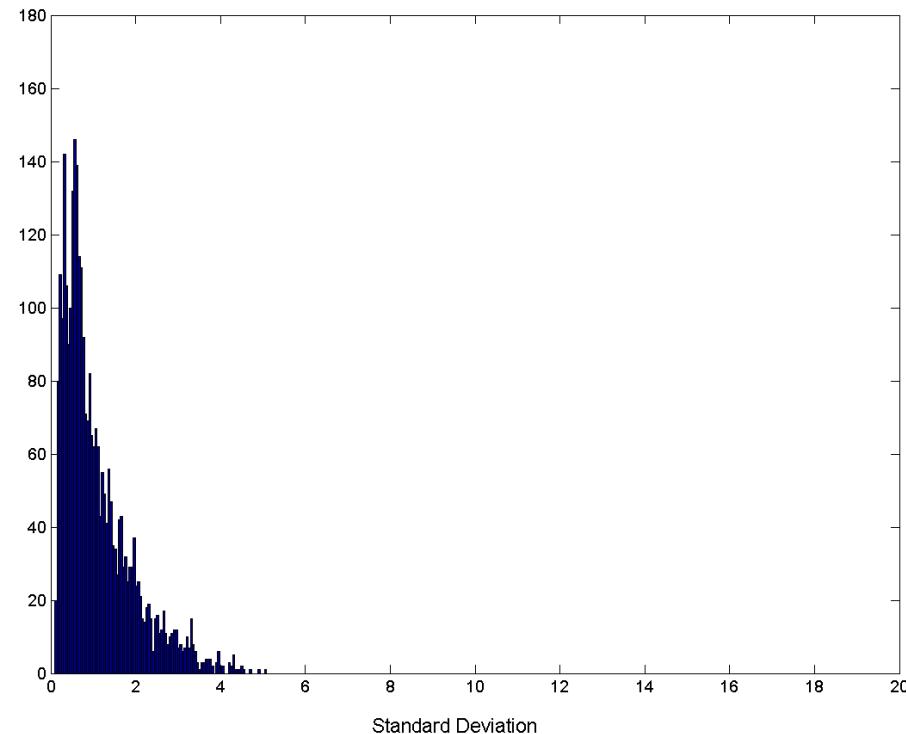
- Combining two or more attributes (or objects) into a single attribute (or object)
- Purpose
 - Data reduction
 - ◆ Reduce the number of attributes or objects
 - Change of scale
 - ◆ Cities aggregated into regions, states, countries, etc
 - More “stable” data
 - ◆ Aggregated data tends to have less variability

Aggregation

Variation of Precipitation in Australia



**Standard Deviation of Average
Monthly Precipitation**



**Standard Deviation of Average
Yearly Precipitation**

Sampling

- Sampling is the main technique employed for data selection.
 - It is often used for both the preliminary investigation of the data and the final data analysis.
- Statisticians sample because **obtaining** the entire set of data of interest is too expensive or time consuming.
- Sampling is used in data mining because **processing** the entire set of data of interest is too expensive or time consuming.

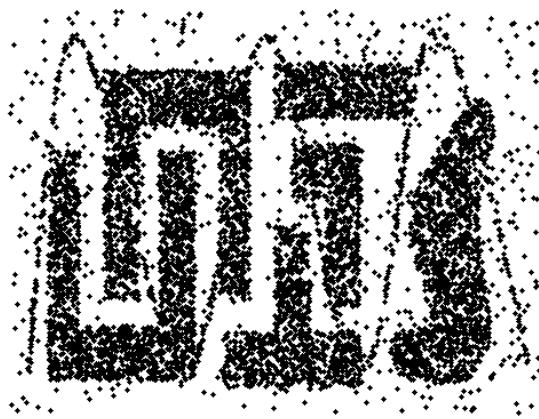
Sampling ...

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

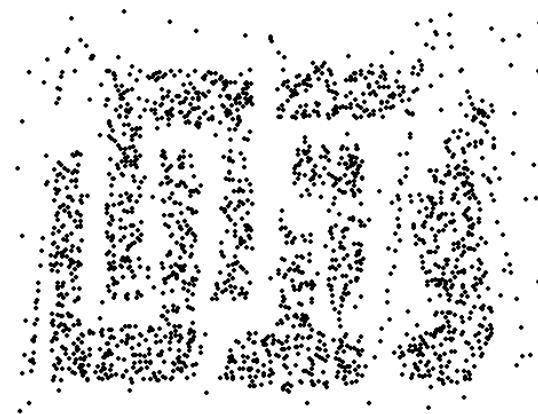
Types of Sampling

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

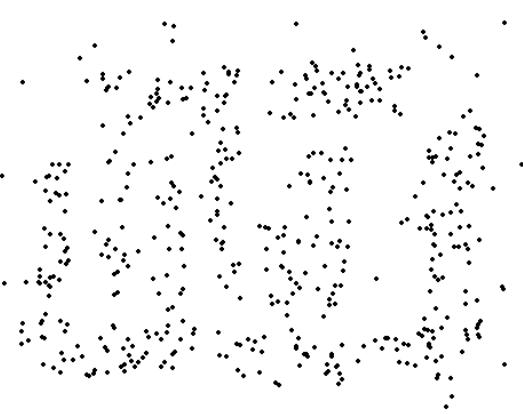
Sample Size



8000 points



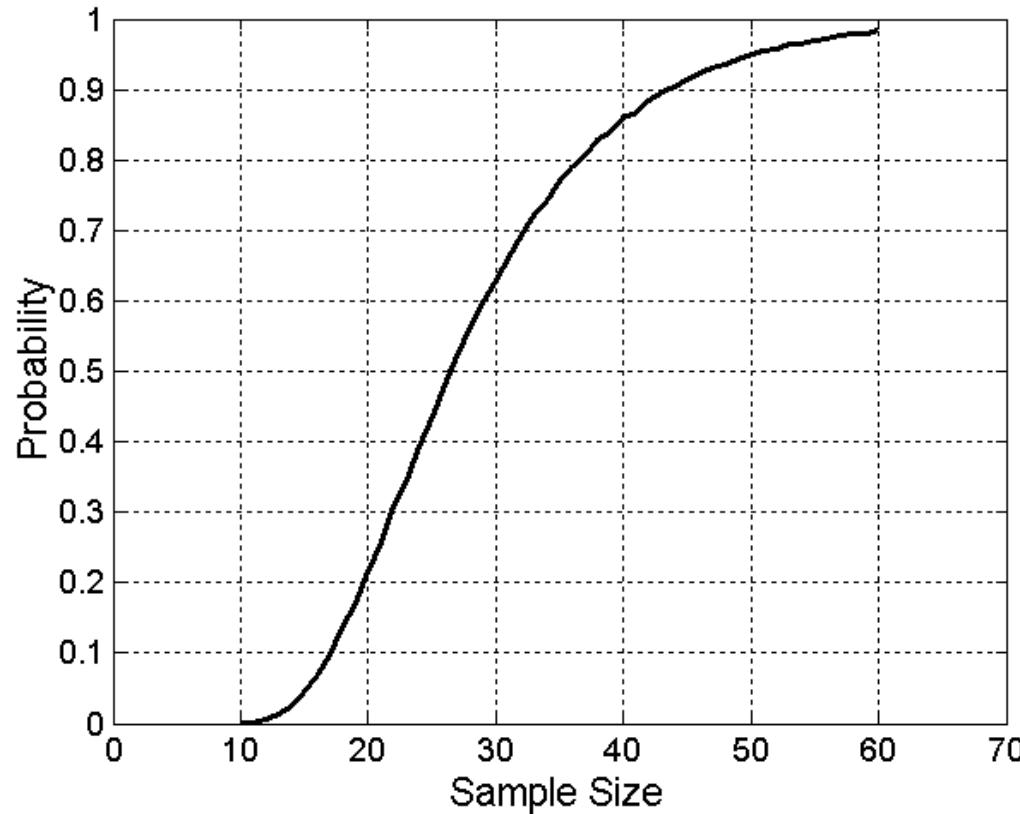
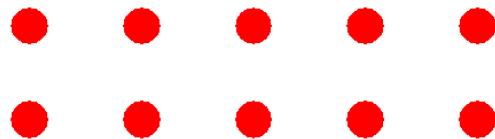
2000 Points



500 Points

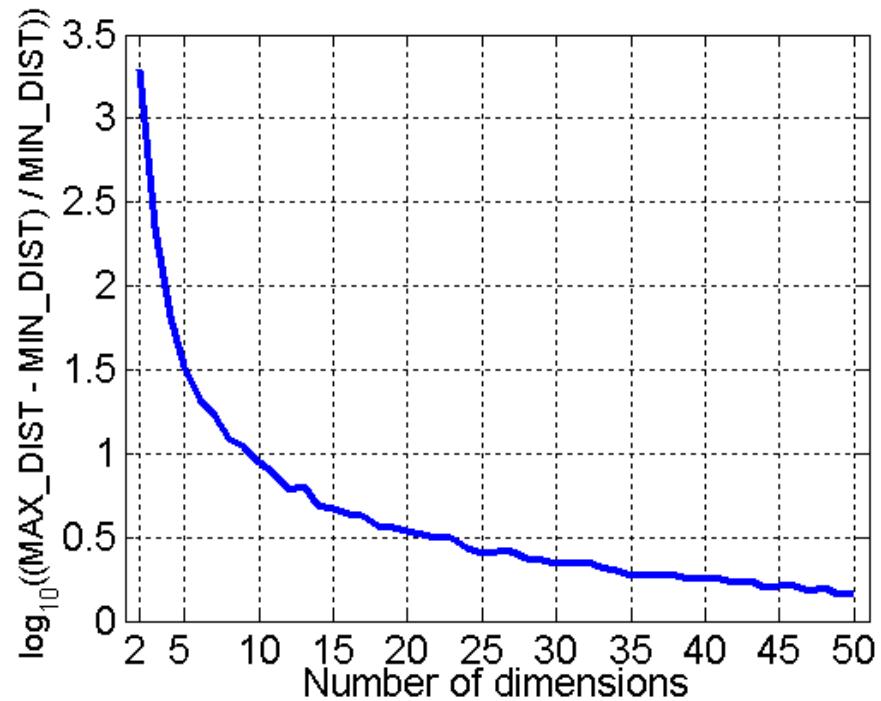
Sample Size

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



Curse of Dimensionality

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



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

Dimensionality Reduction

- Purpose:

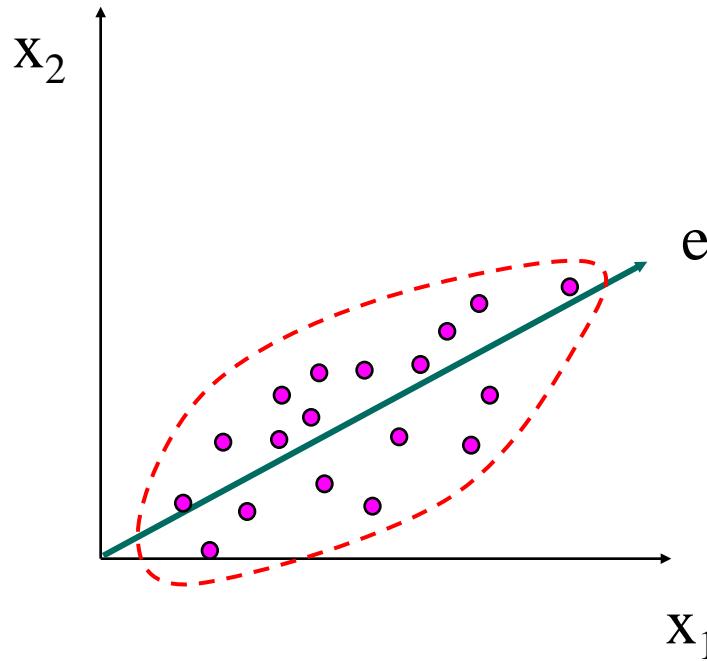
- Avoid curse of dimensionality
- Reduce amount of time and memory required by data mining algorithms
- Allow data to be more easily visualized
- May help to eliminate irrelevant features or reduce noise

- Techniques

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

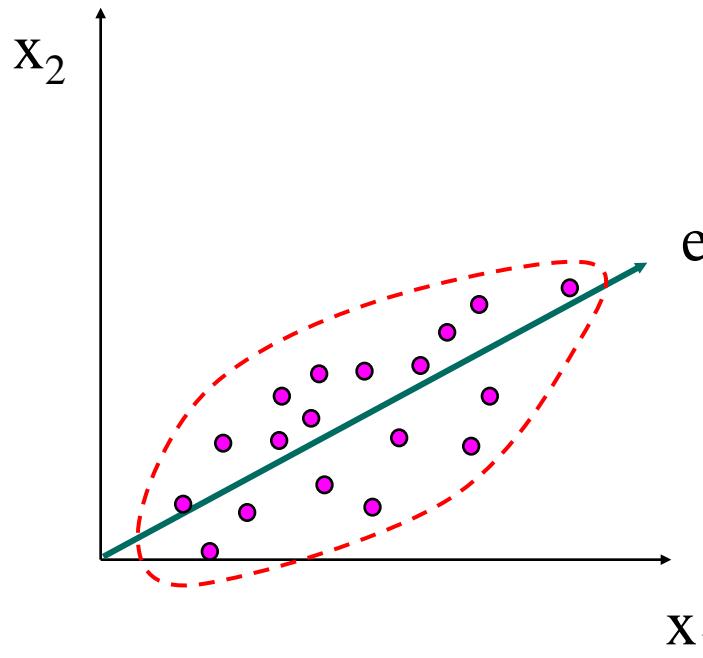
Dimensionality Reduction: PCA

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



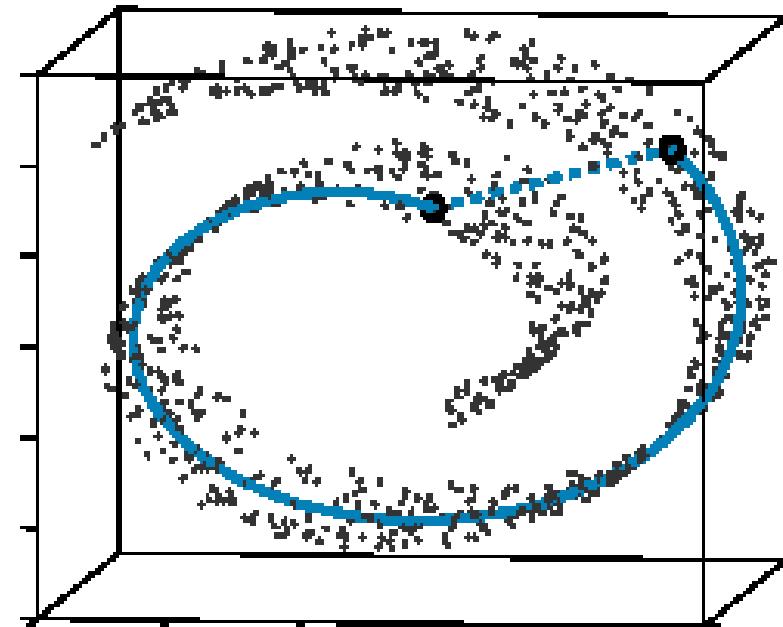
Dimensionality Reduction: PCA

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



Dimensionality Reduction: ISOMAP

By: Tenenbaum, de Silva,
Langford (2000)



- Construct a neighbourhood graph
- For each pair of points in the graph, compute the shortest path distances – geodesic distances

Dimensionality Reduction: PCA

Dimensions = 206



Feature Subset Selection

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

Feature Subset Selection

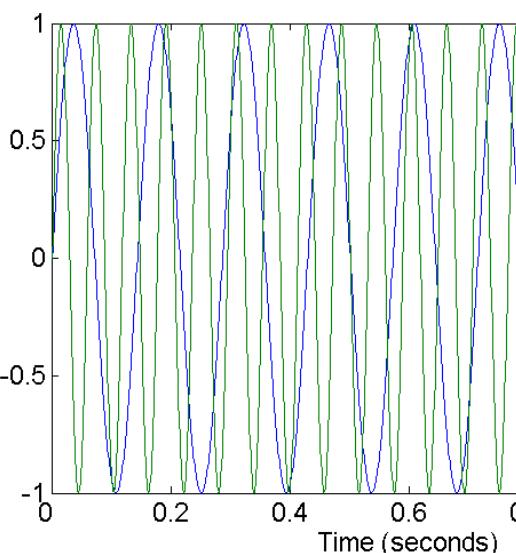
- Techniques:
 - Brute-force approach:
 - ◆ Try all possible feature subsets as input to data mining algorithm
 - Embedded approaches:
 - ◆ Feature selection occurs naturally as part of the data mining algorithm
 - Filter approaches:
 - ◆ Features are selected before data mining algorithm is run
 - Wrapper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

Feature Creation

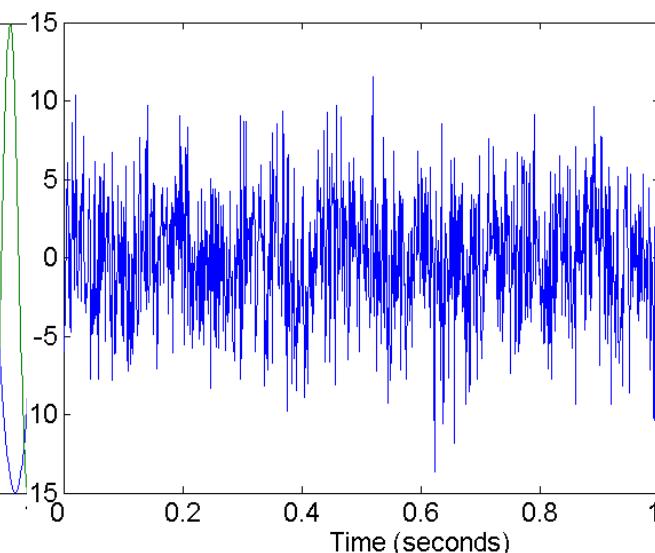
- Create new attributes that can capture the important information in a data set much more efficiently than the original attributes
- Three general methodologies:
 - Feature Extraction
 - ◆ domain-specific
 - Mapping Data to New Space
 - Feature Construction
 - ◆ combining features

Mapping Data to a New Space

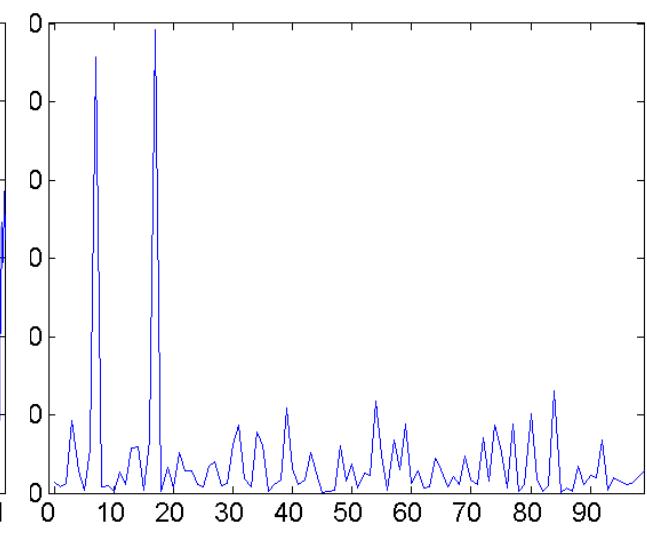
- Fourier transform
- Wavelet transform



Two Sine Waves



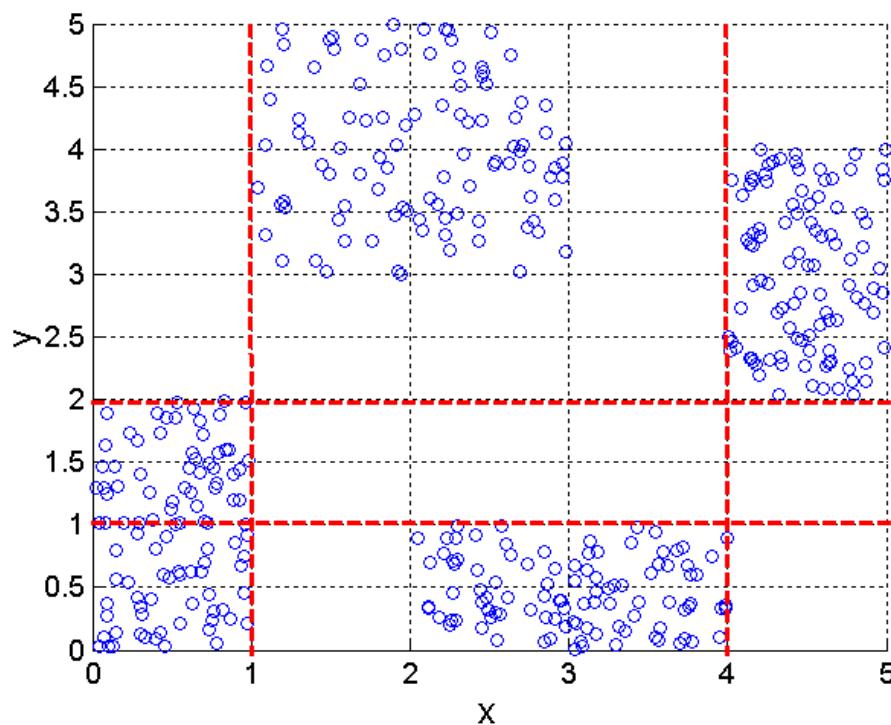
Two Sine Waves + Noise



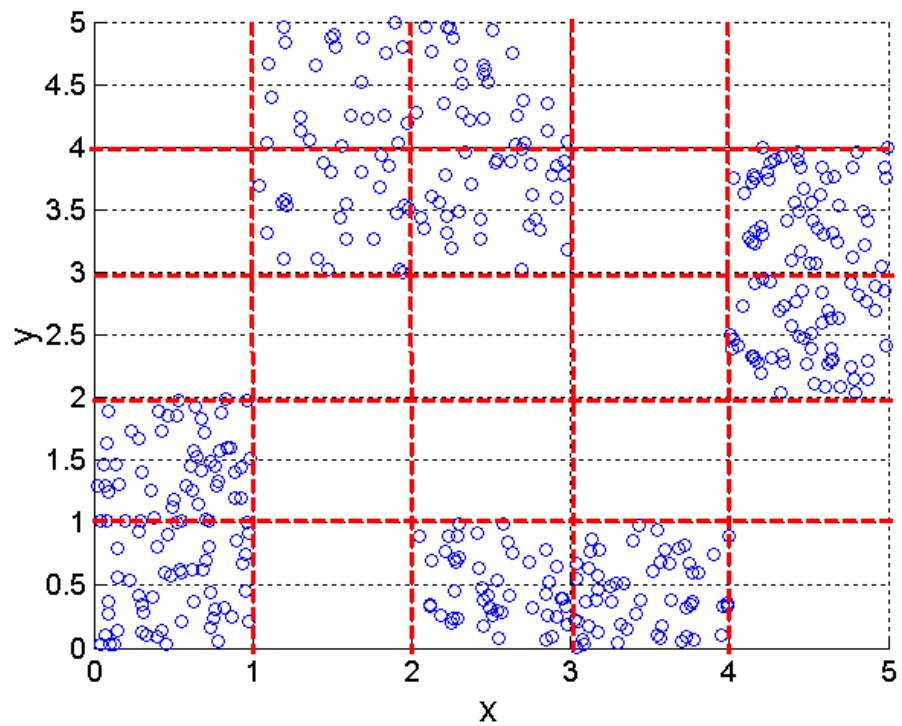
Frequency

Discretization Using Class Labels

- Entropy based approach

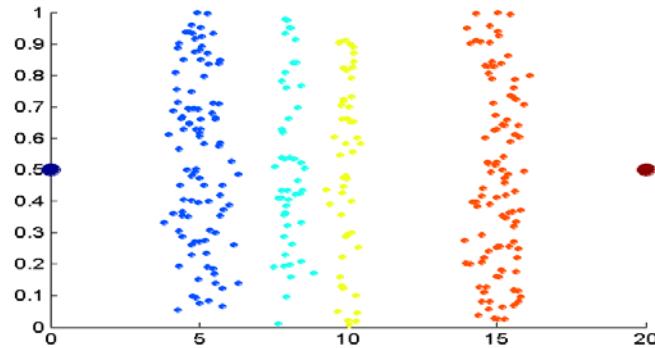


3 categories for both x and y

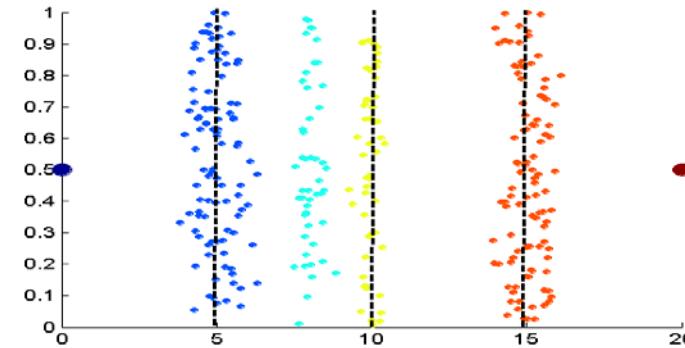


5 categories for both x and y

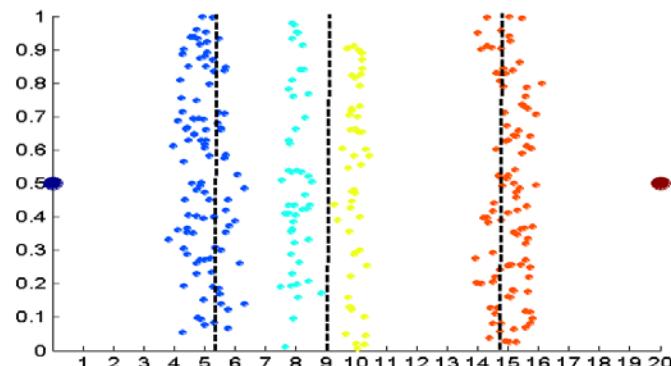
Discretization Without Using Class Labels



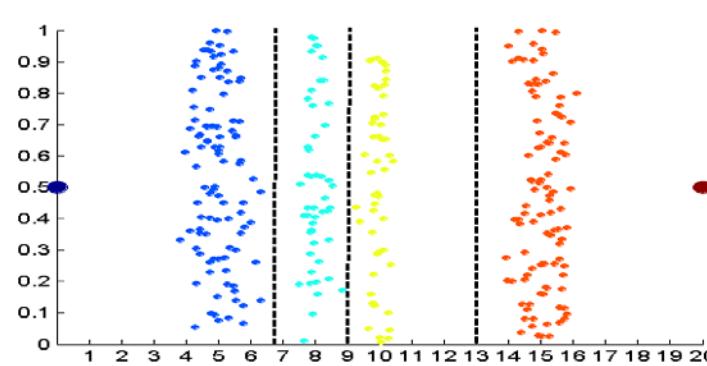
Data



Equal interval width



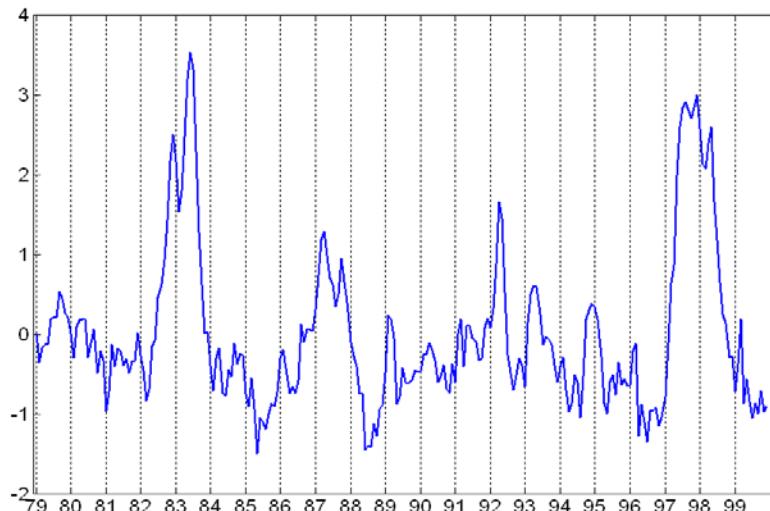
Equal frequency



K-means

Attribute Transformation

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



Similarity and Dissimilarity

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

Similarity/Dissimilarity for Simple Attributes

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

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

Table 5.1. Similarity and dissimilarity for simple attributes

Euclidean Distance

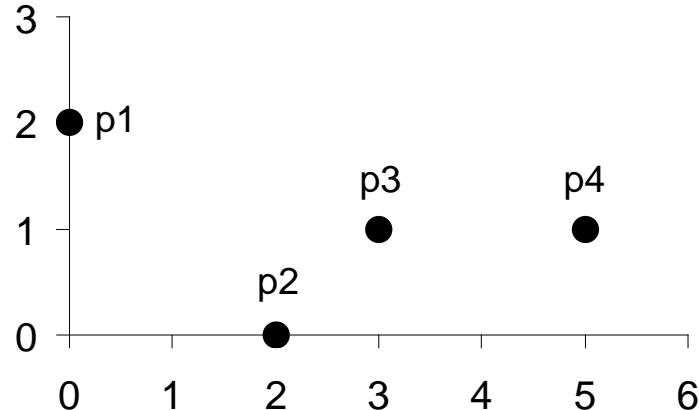
- Euclidean Distance

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

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

- Standardization is necessary, if scales differ.

Euclidean Distance



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

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

Distance Matrix

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$dist = \left(\sum_{k=1}^n |p_k - q_k|^r \right)^{\frac{1}{r}}$$

Where r is a parameter, n is the number of dimensions (attributes) and p_k and q_k are, respectively, the k th attributes (components) or data objects p and q .

Minkowski Distance: Examples

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

Minkowski Distance

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

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

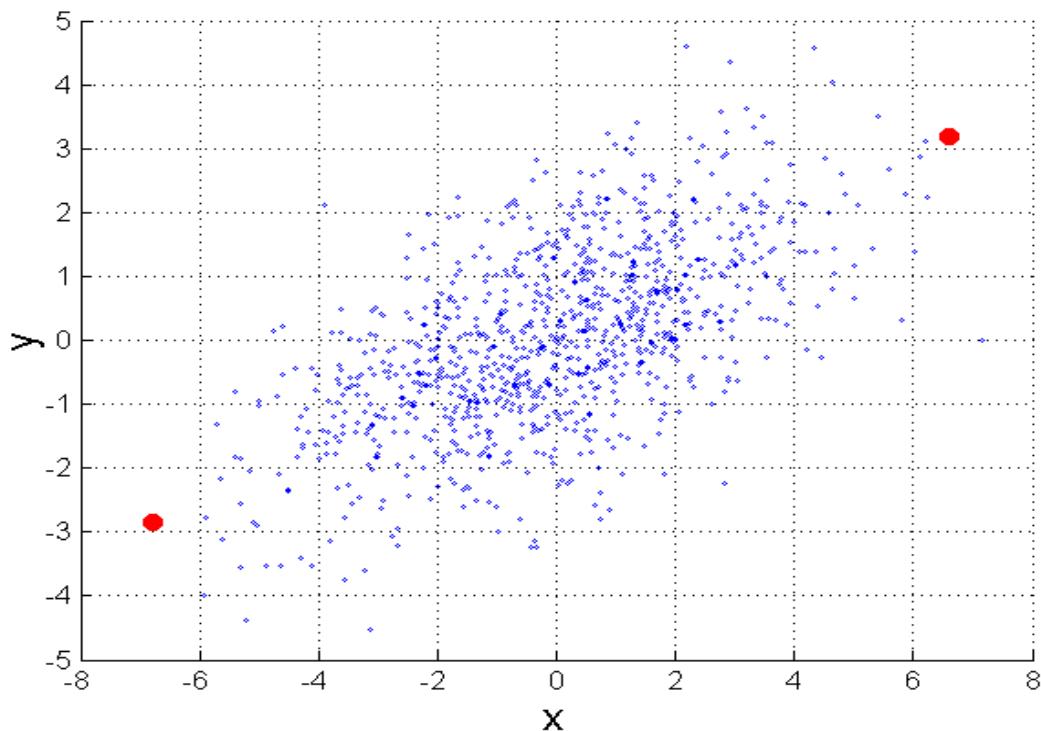
L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

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

Distance Matrix

Mahalanobis Distance

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

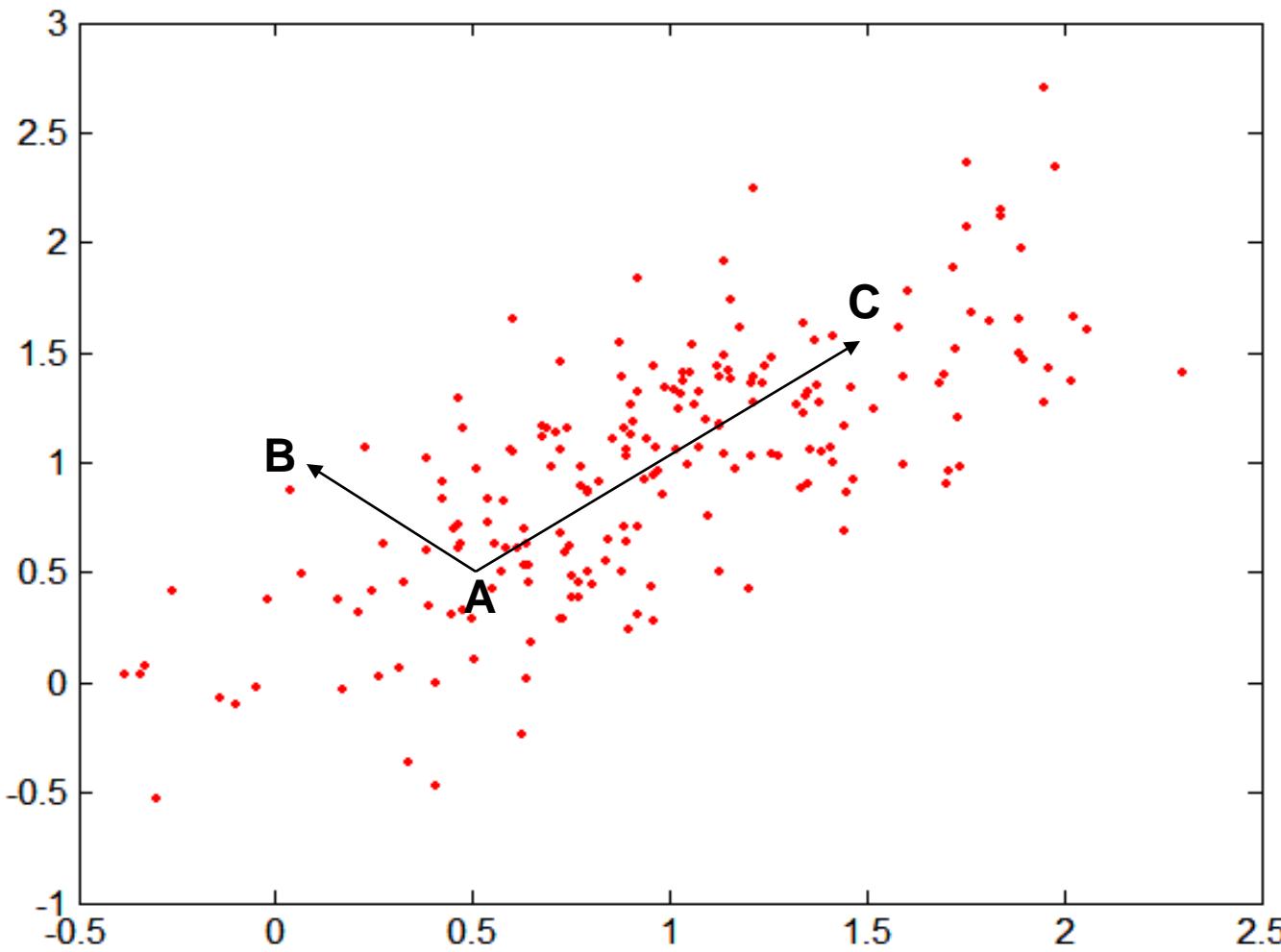


Σ is the covariance matrix of the input data X

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

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

Mahalanobis Distance



Covariance Matrix:

$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

Mahal(A,B) = 5

Mahal(A,C) = 4

Common Properties of a Distance

- Distances, such as the Euclidean distance, have some well known properties.
 1. $d(p, q) \geq 0$ for all p and q and $d(p, q) = 0$ only if $p = q$. (Positive definiteness)
 2. $d(p, q) = d(q, p)$ for all p and q . (Symmetry)
 3. $d(p, r) \leq d(p, q) + d(q, r)$ for all points p , q , and r . (Triangle Inequality)where $d(p, q)$ is the distance (dissimilarity) between points (data objects), p and q .
- A distance that satisfies these properties is a **metric**

Common Properties of a Similarity

- Similarities, also have some well known properties.
 1. $s(p, q) = 1$ (or maximum similarity) only if $p = q$.
 2. $s(p, q) = s(q, p)$ for all p and q . (Symmetry)

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

Similarity Between Binary Vectors

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

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

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

M_{00} = the number of attributes where p was 0 and q was 0

M_{11} = the number of attributes where p was 1 and q was 1

- Simple Matching and Jaccard Coefficients

SMC = number of matches / number of attributes

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

J = number of 11 matches / number of not-both-zero attributes values

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

SMC versus Jaccard: Example

$p = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$

$q = 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1$

$M_{01} = 2$ (the number of attributes where p was 0 and q was 1)

$M_{10} = 1$ (the number of attributes where p was 1 and q was 0)

$M_{00} = 7$ (the number of attributes where p was 0 and q was 0)

$M_{11} = 0$ (the number of attributes where p was 1 and q was 1)

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

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

Cosine Similarity

- If d_1 and d_2 are two document vectors, then

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

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

- Example:

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

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

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

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

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

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

Extended Jaccard Coefficient (Tanimoto)

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

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

Correlation

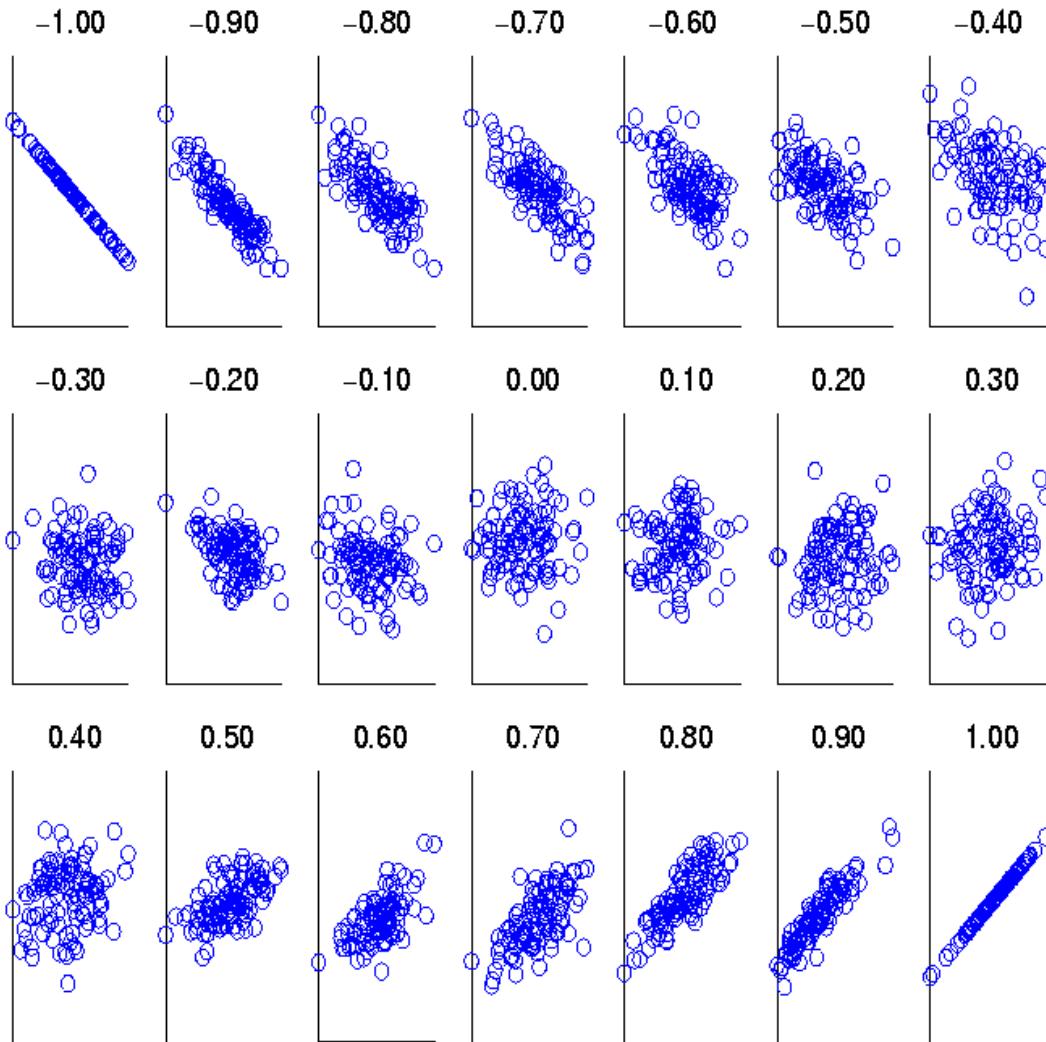
- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, p and q, and then take their dot product

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

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

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

Visually Evaluating Correlation



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

General Approach for Combining Similarities

- Sometimes attributes are of many different types, but an overall similarity is needed.
 1. For the k^{th} attribute, compute a similarity, s_k , in the range $[0, 1]$.
 2. Define an indicator variable, δ_k , for the k_{th} attribute as follows:
$$\delta_k = \begin{cases} 0 & \text{if the } k^{th} \text{ attribute is a binary asymmetric attribute and both objects have} \\ & \text{a value of 0, or if one of the objects has a missing values for the } k^{th} \text{ attribute} \\ 1 & \text{otherwise} \end{cases}$$
 3. Compute the overall similarity between the two objects using the following formula:

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

Using Weights to Combine Similarities

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

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

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

Density

- Density-based clustering require a notion of density
- Examples:
 - Euclidean density
 - ◆ Euclidean density = number of points per unit volume
 - Probability density
 - Graph-based density

Euclidean Density – Cell-based

- Simplest approach is to divide region into a number of rectangular cells of equal volume and define density as # of points the cell contains

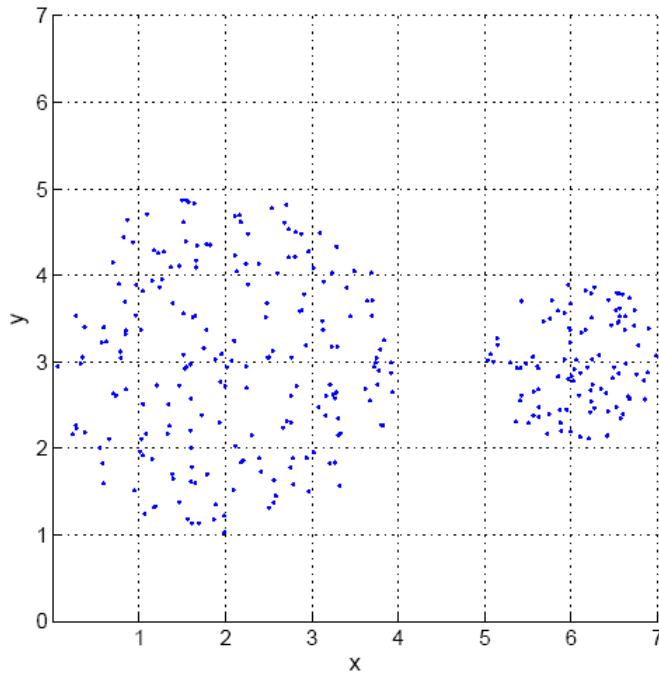


Figure 7.13. Cell-based density.

0	0	0	0	0	0	0
0	0	0	0	0	0	0
4	17	18	6	0	0	0
14	14	13	13	0	18	27
11	18	10	21	0	24	31
3	20	14	4	0	0	0
0	0	0	0	0	0	0

Table 7.6. Point counts for each grid cell.

Euclidean Density – Center-based

- Euclidean density is the number of points within a specified radius of the point

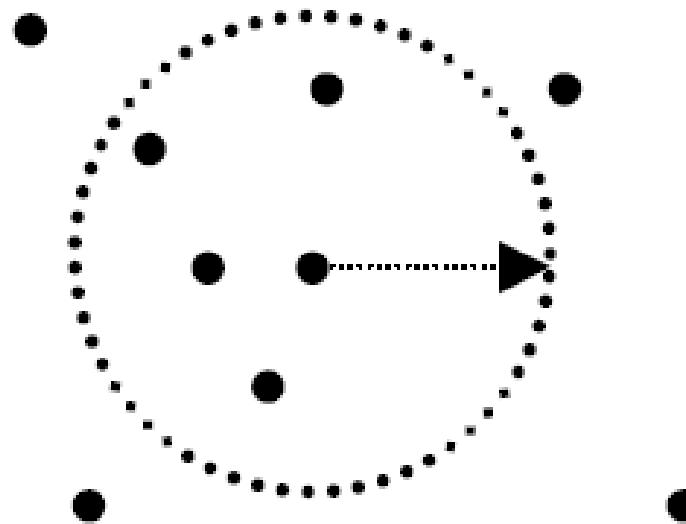


Figure 7.14. Illustration of center-based density.

Data Mining: Exploring Data

Lecture Notes for Chapter 3

Introduction to Data Mining

by

Tan, Steinbach, Kumar

What is data exploration?

A preliminary exploration of the data to better understand its characteristics.

- Key motivations of data exploration include
 - Helping to select the right tool for preprocessing or analysis
 - Making use of humans' abilities to recognize patterns
 - ◆ People can recognize patterns not captured by data analysis tools
- Related to the area of Exploratory Data Analysis (EDA)
 - Created by statistician John Tukey
 - Seminal book is Exploratory Data Analysis by Tukey
 - A nice online introduction can be found in Chapter 1 of the NIST Engineering Statistics Handbook

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

Techniques Used In Data Exploration

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

Iris Sample Data Set

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



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

Summary Statistics

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

Frequency and Mode

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

Percentiles

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

Given an ordinal or continuous attribute x and a number p between 0 and 100, the p th percentile is a value x_p of x such that $p\%$ of the observed values of x are less than x_p .

- For instance, the 50th percentile is the value $x_{50\%}$ such that 50% of all values of x are less than $x_{50\%}$.

Measures of Location: Mean and Median

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

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

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

Measures of Spread: Range and Variance

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

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

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

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

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

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

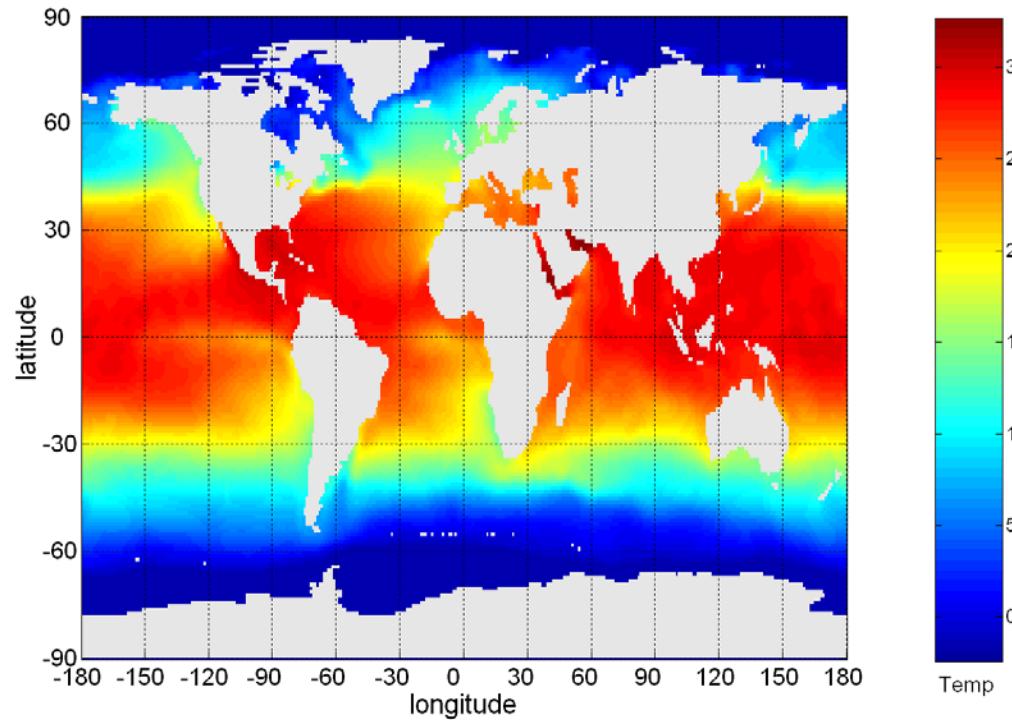
Visualization

Visualization is the conversion of data into a visual or tabular format so that the characteristics of the data and the relationships among data items or attributes can be analyzed or reported.

- Visualization of data is one of the most powerful and appealing techniques for data exploration.
 - Humans have a well developed ability to analyze large amounts of information that is presented visually
 - Can detect general patterns and trends
 - Can detect outliers and unusual patterns

Example: Sea Surface Temperature

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



Representation

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

Arrangement

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

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

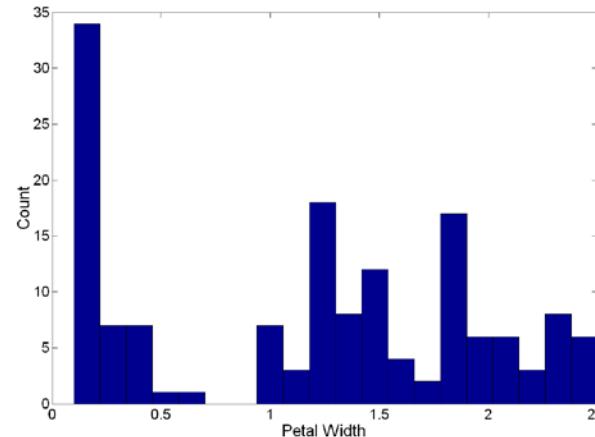
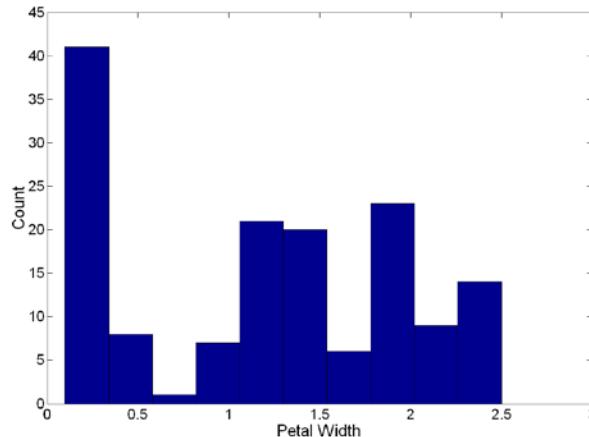
	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

Selection

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

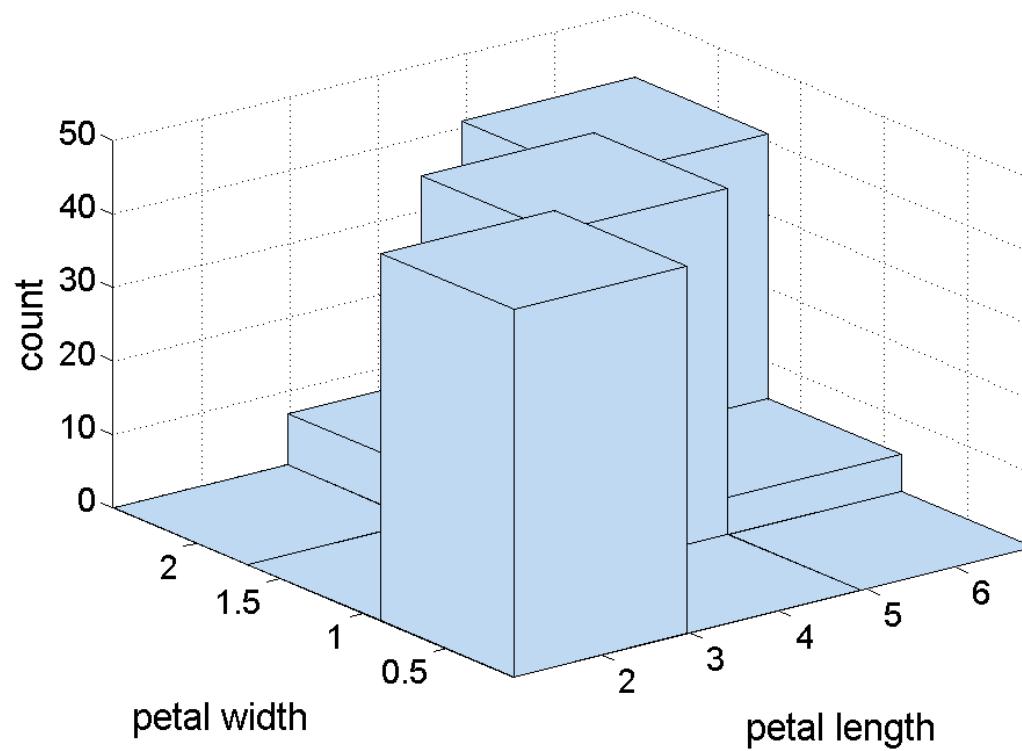
Visualization Techniques: Histograms

- Histogram
 - Usually shows the distribution of values of a single variable
 - Divide the values into bins and show a bar plot of the number of objects in each bin.
 - The height of each bar indicates the number of objects
 - Shape of histogram depends on the number of bins
- Example: Petal Width (10 and 20 bins, respectively)



Two-Dimensional Histograms

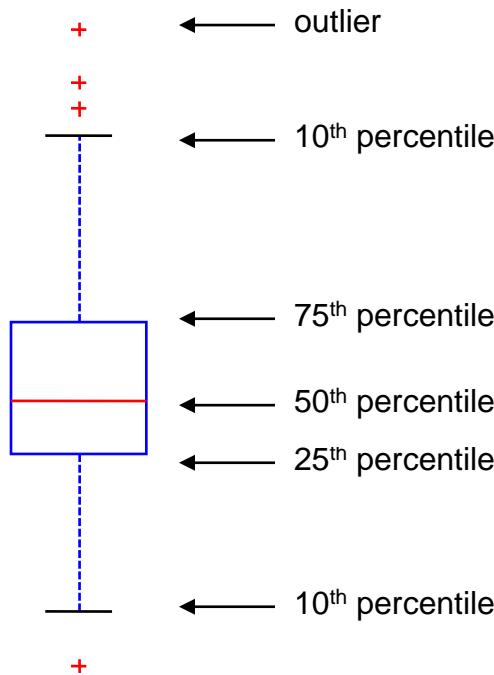
- Show the joint distribution of the values of two attributes
- Example: petal width and petal length
 - What does this tell us?



Visualization Techniques: Box Plots

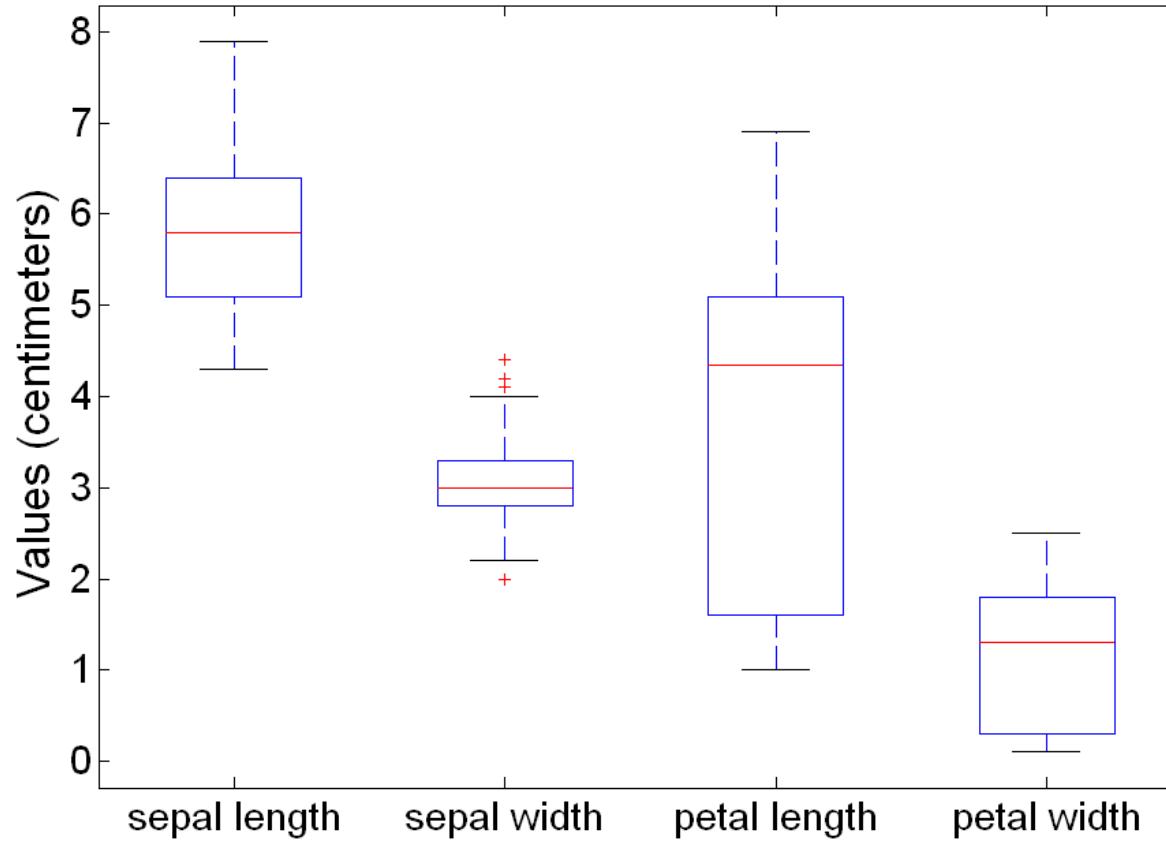
- Box Plots

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



Example of Box Plots

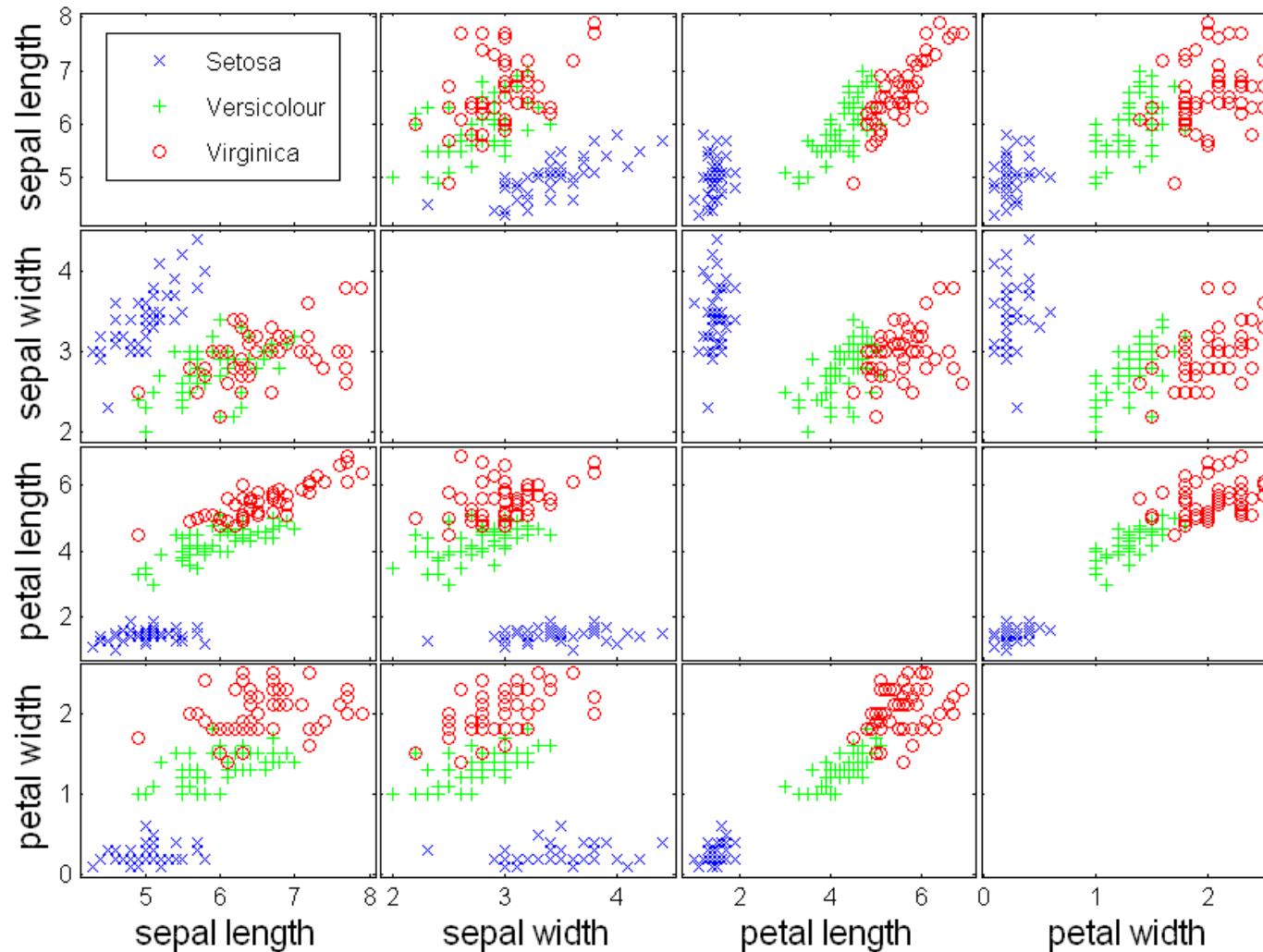
- Box plots can be used to compare attributes



Visualization Techniques: Scatter Plots

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

Scatter Plot Array of Iris Attributes

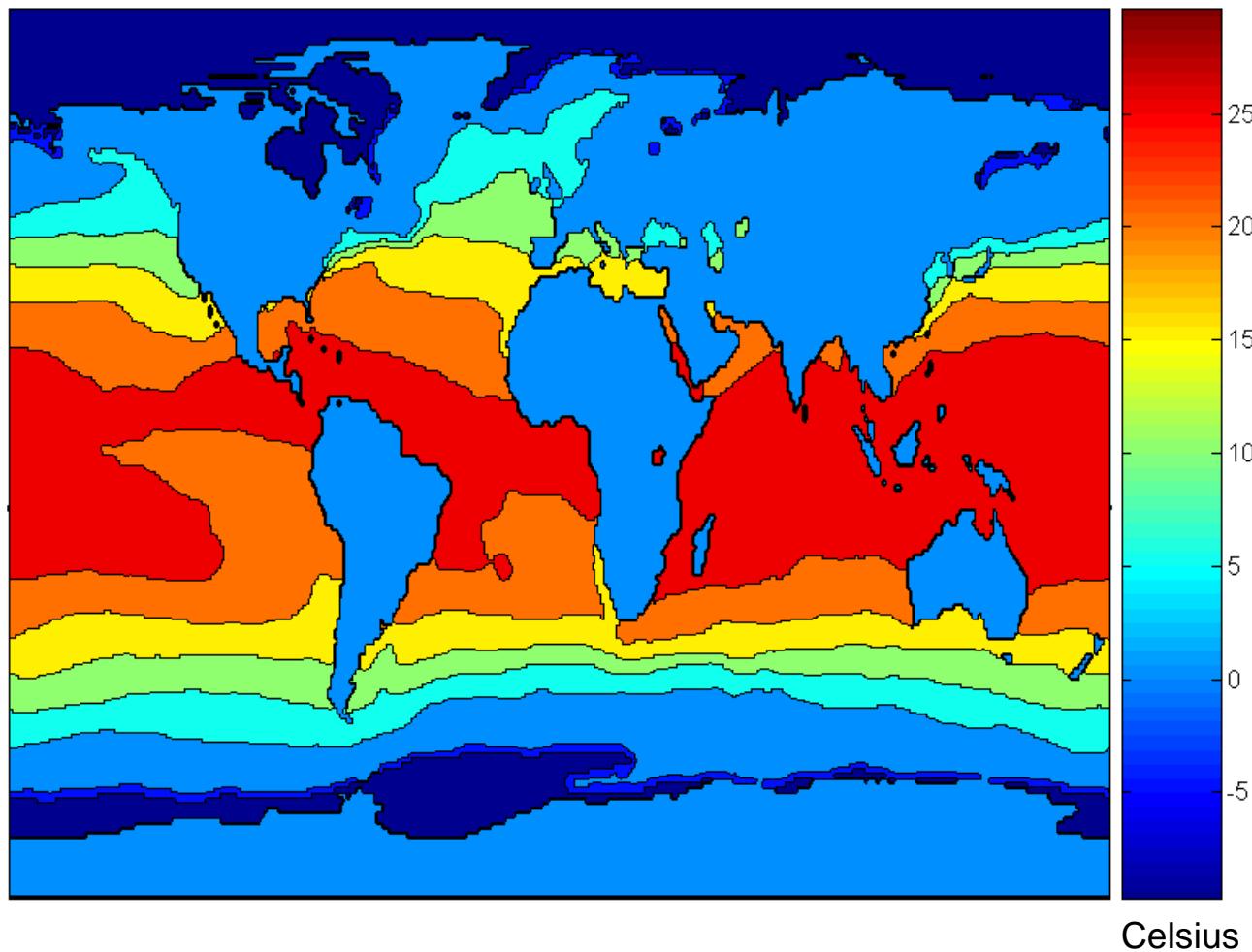


Visualization Techniques: Contour Plots

- Contour plots

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

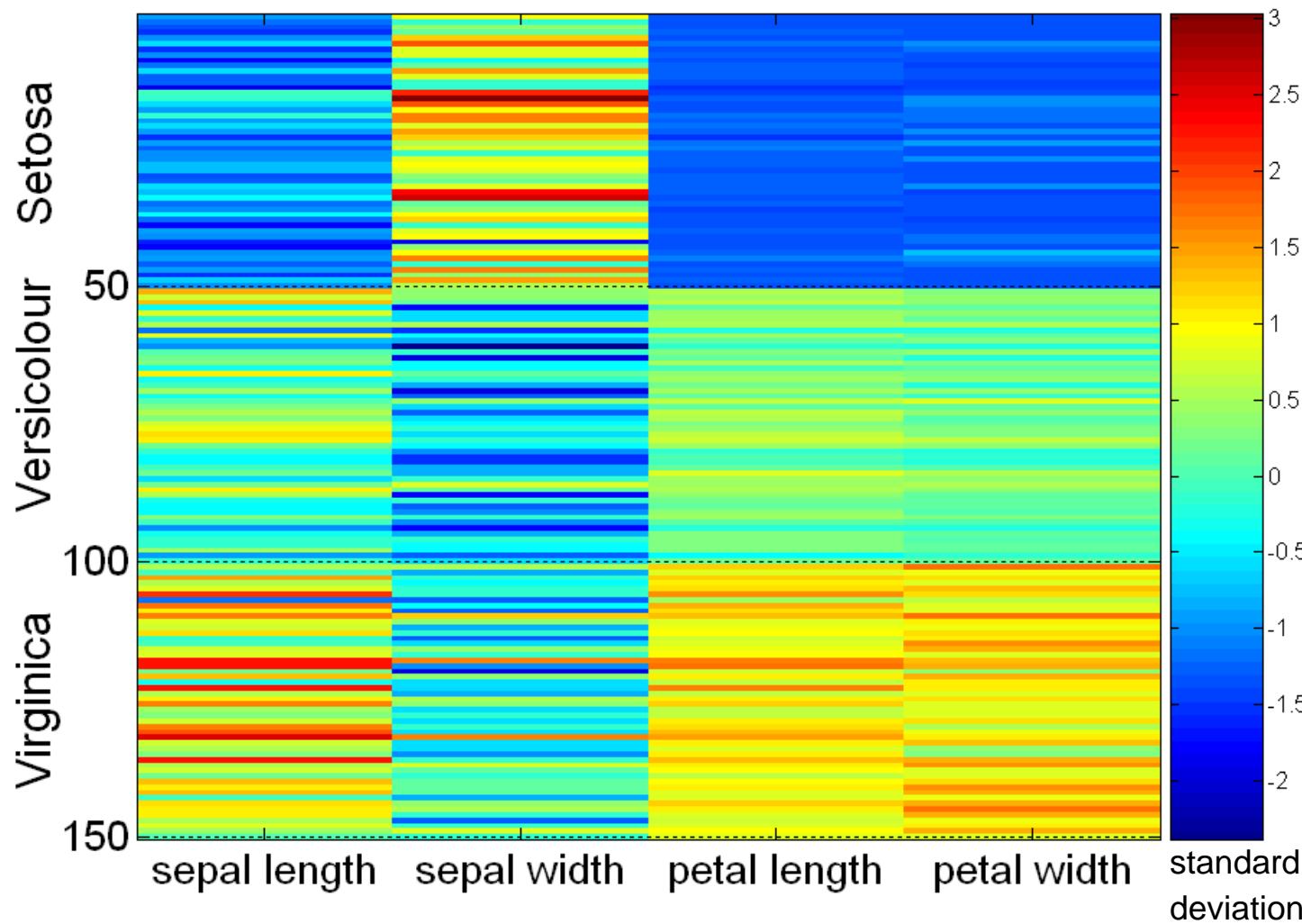
Contour Plot Example: SST Dec, 1998



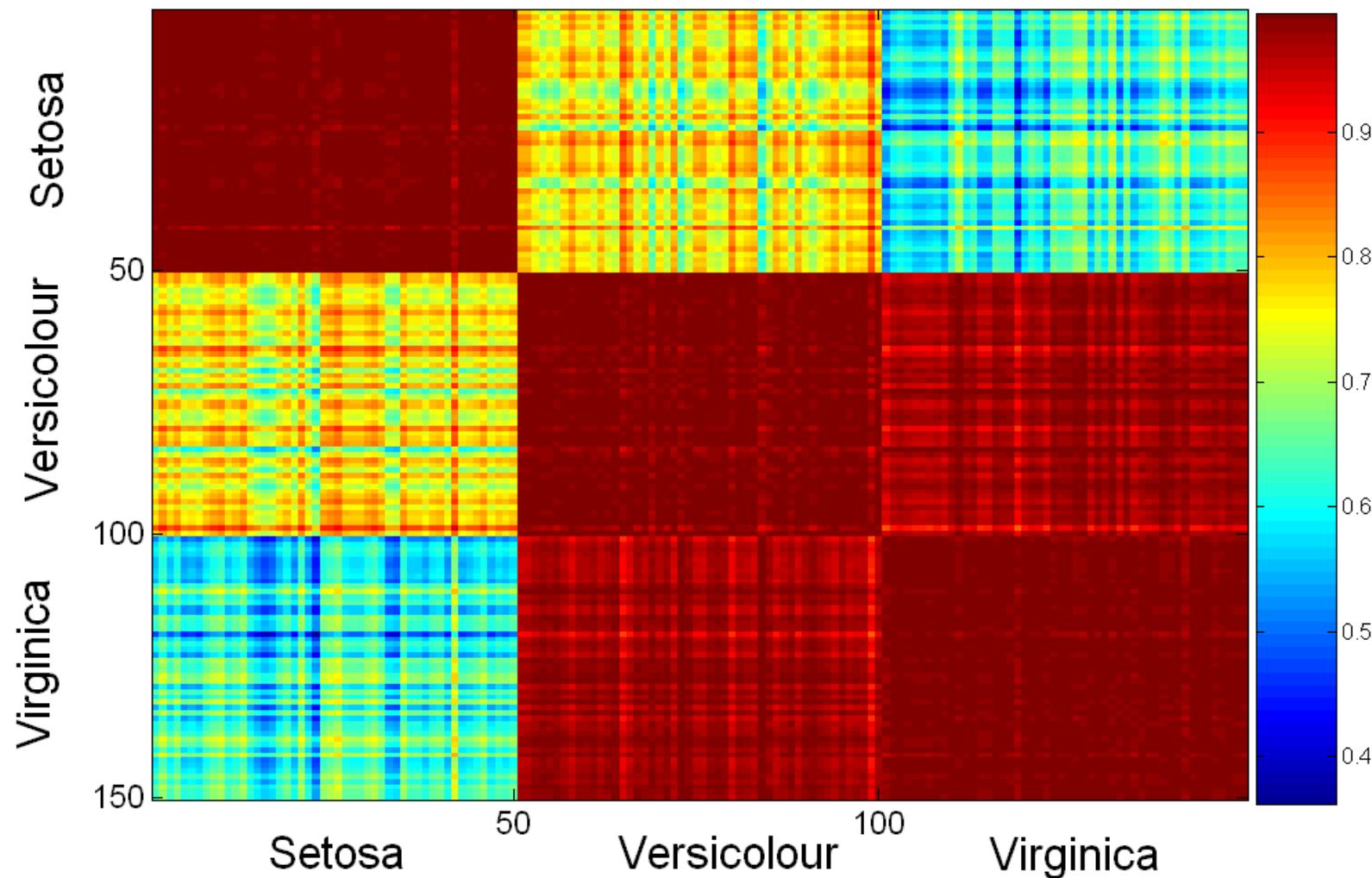
Visualization Techniques: Matrix Plots

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

Visualization of the Iris Data Matrix



Visualization of the Iris Correlation Matrix

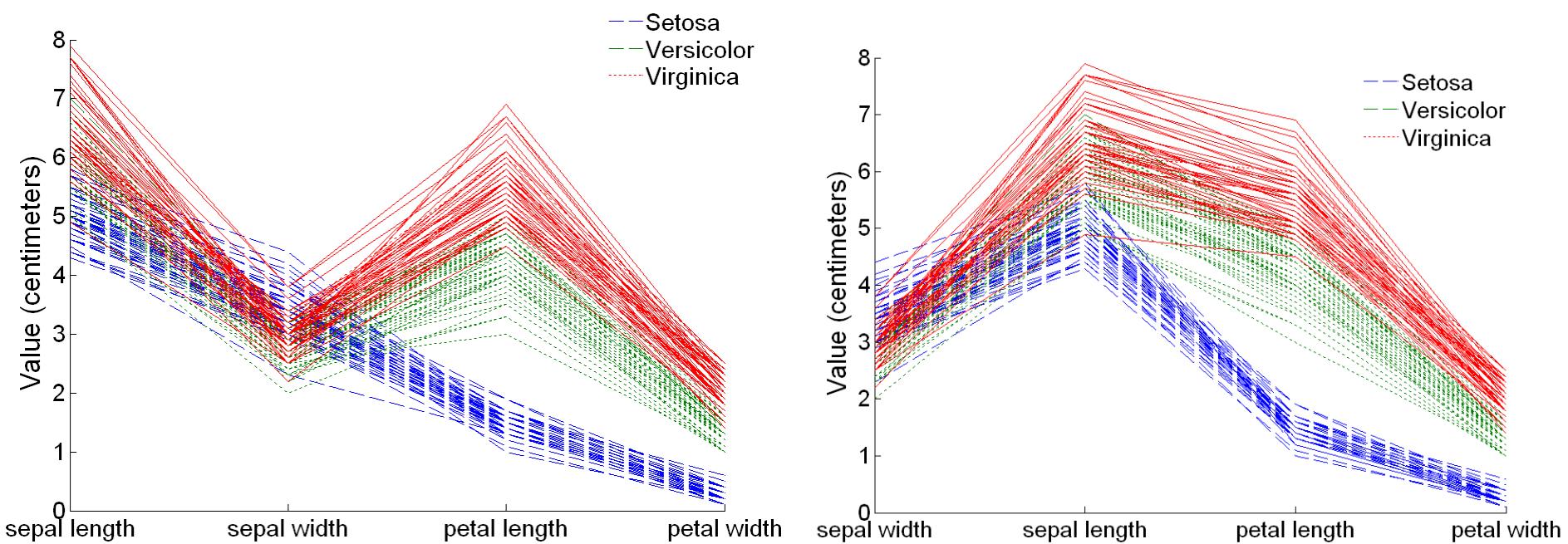


Visualization Techniques: Parallel Coordinates

- Parallel Coordinates

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

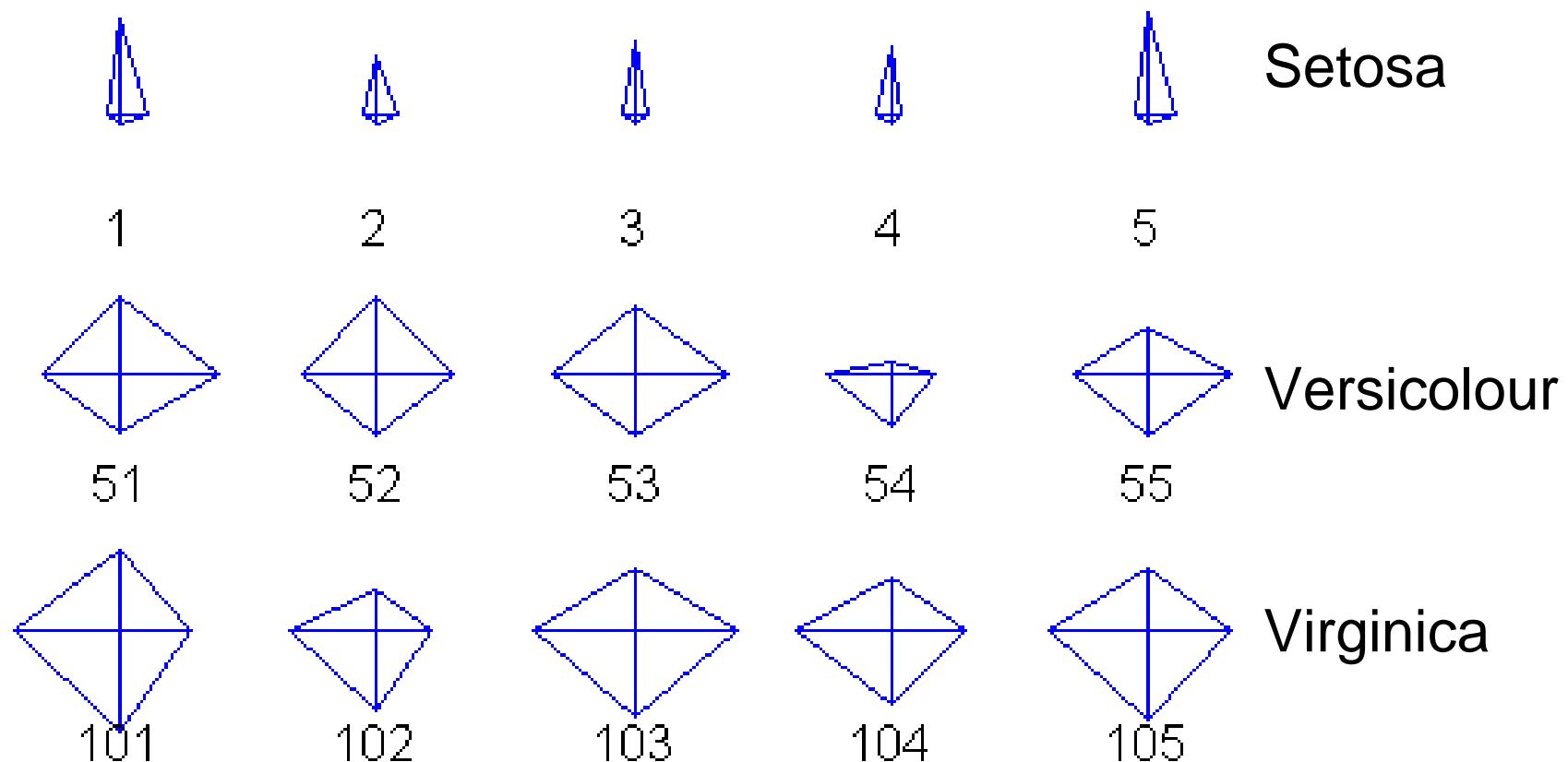
Parallel Coordinates Plots for Iris Data



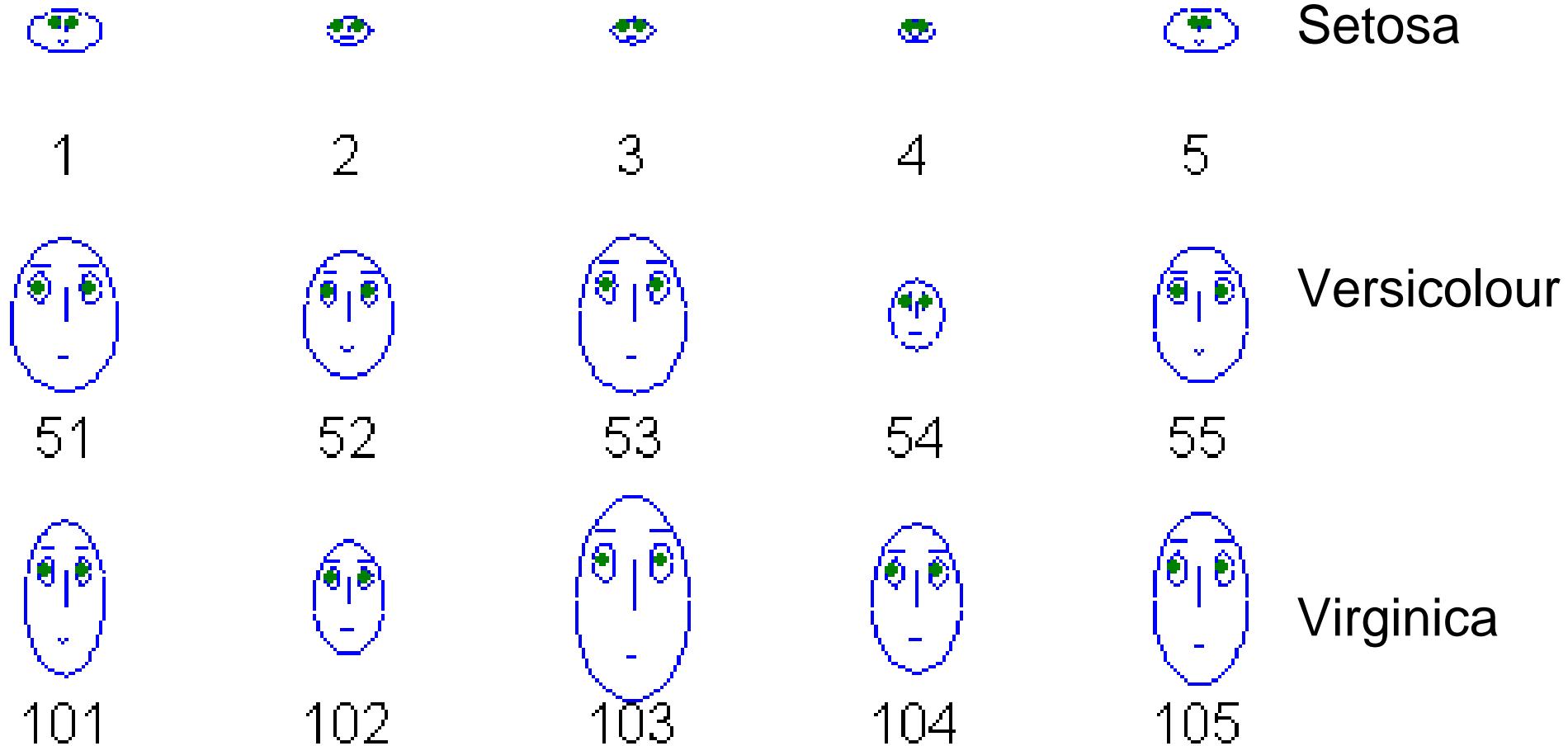
Other Visualization Techniques

- Star Plots
 - Similar approach to parallel coordinates, but axes radiate from a central point
 - The line connecting the values of an object is a polygon
- Chernoff Faces
 - Approach created by Herman Chernoff
 - This approach associates each attribute with a characteristic of a face
 - The values of each attribute determine the appearance of the corresponding facial characteristic
 - Each object becomes a separate face
 - Relies on human's ability to distinguish faces

Star Plots for Iris Data



Chernoff Faces for Iris Data



OLAP

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

Creating a Multidimensional Array

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

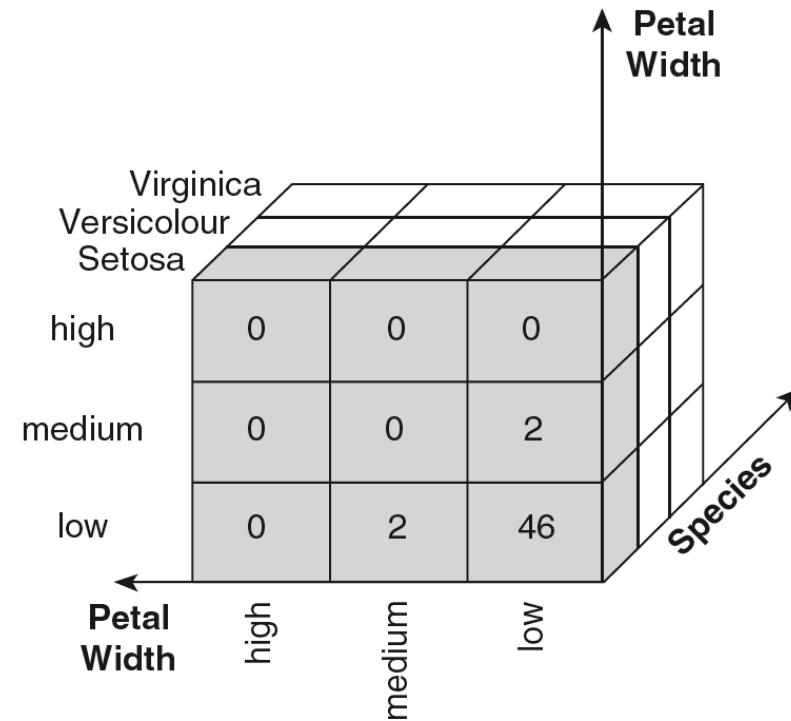
Example: Iris data

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

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

Example: Iris data (continued)

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



Example: Iris data (continued)

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

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

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

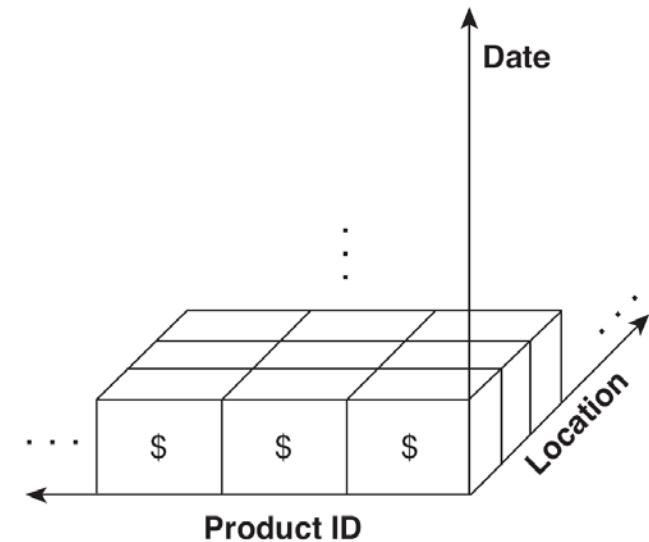
		Width		
		low	medium	high
Length	low	0	0	0
	medium	0	0	3
	high	0	3	44

OLAP Operations: Data Cube

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

Data Cube Example

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



Data Cube Example (continued)

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

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

OLAP Operations: Slicing and Dicing

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

OLAP Operations: Roll-up and Drill-down

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

OLAP Operations: Roll-up and Drill-down

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

Data Mining

Classification: Basic Concepts, Decision Trees, and Model Evaluation

Lecture Notes for Chapter 4

Introduction to Data Mining

by

Tan, Steinbach, Kumar

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.

Illustrating Classification Task

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	?

Training Set

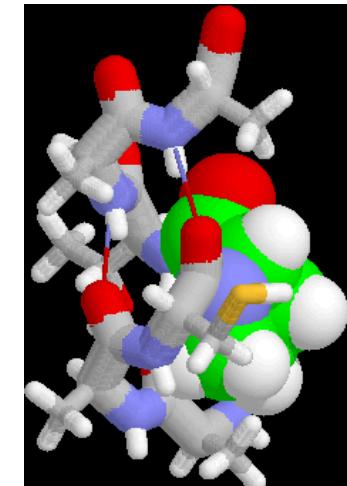
Test Set

Learning algorithm

Model

Examples of Classification Task

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil
- Categorizing news stories as finance, weather, entertainment, sports, etc



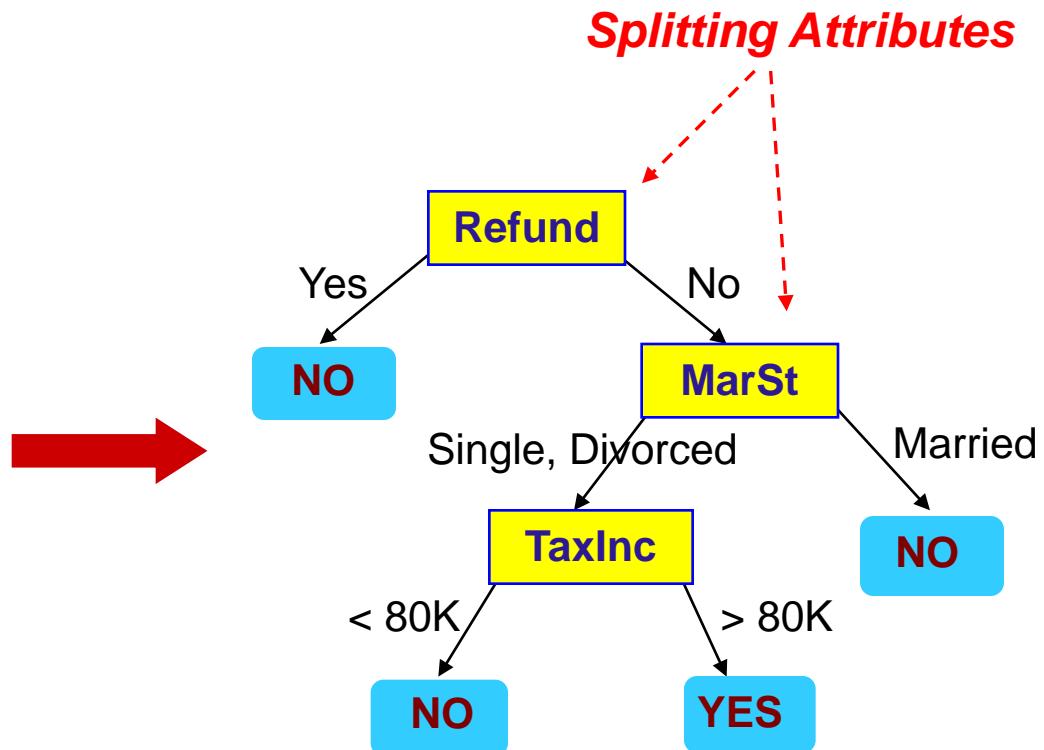
Classification Techniques

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

Example of a Decision Tree

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

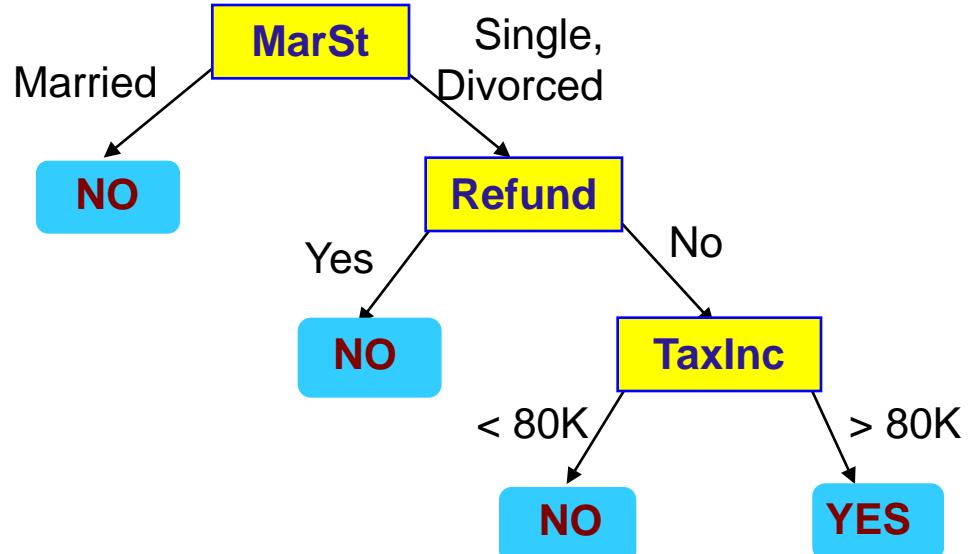
Training Data



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!

Decision Tree Classification Task

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	?

Training Set

Test Set

Decision Tree Induction algorithm

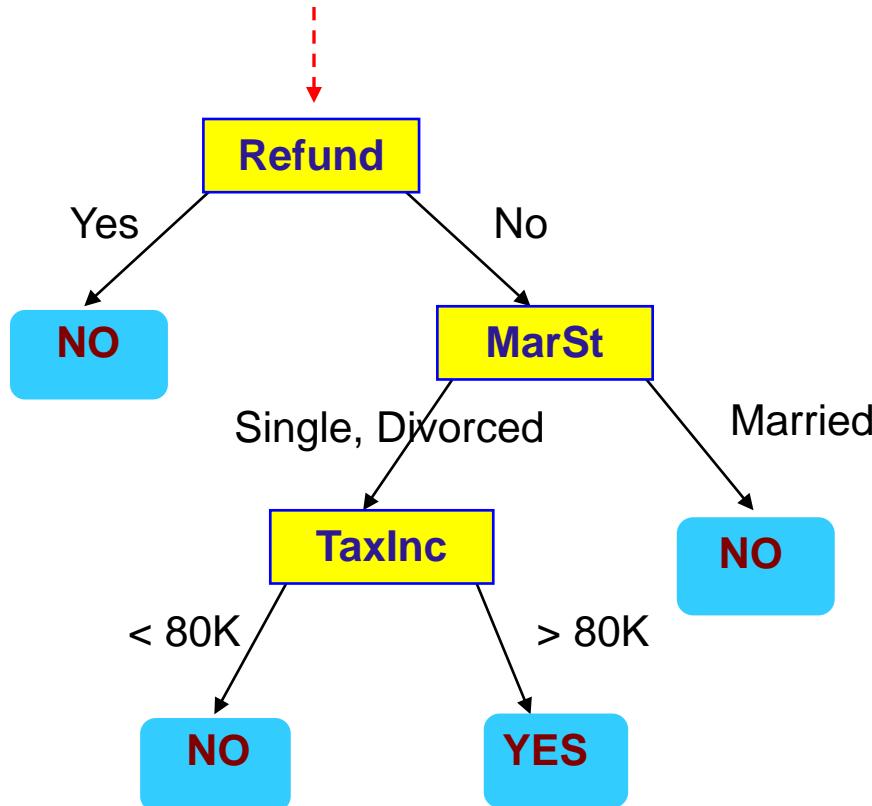
Model

Decision Tree

The diagram illustrates a decision tree classification task. A pink rounded rectangle labeled "Decision Tree Induction algorithm" contains a green downward arrow pointing to the "Attrib3" column of row 11. A red box highlights the value "110K" in row 13's "Attrib3" column. A red arrow points upwards from the bottom of this red box towards the question mark in the "Class" column of row 13. The rows are labeled "Training Set" and "Test Set".

Apply Model to Test Data

Start from the root of tree.



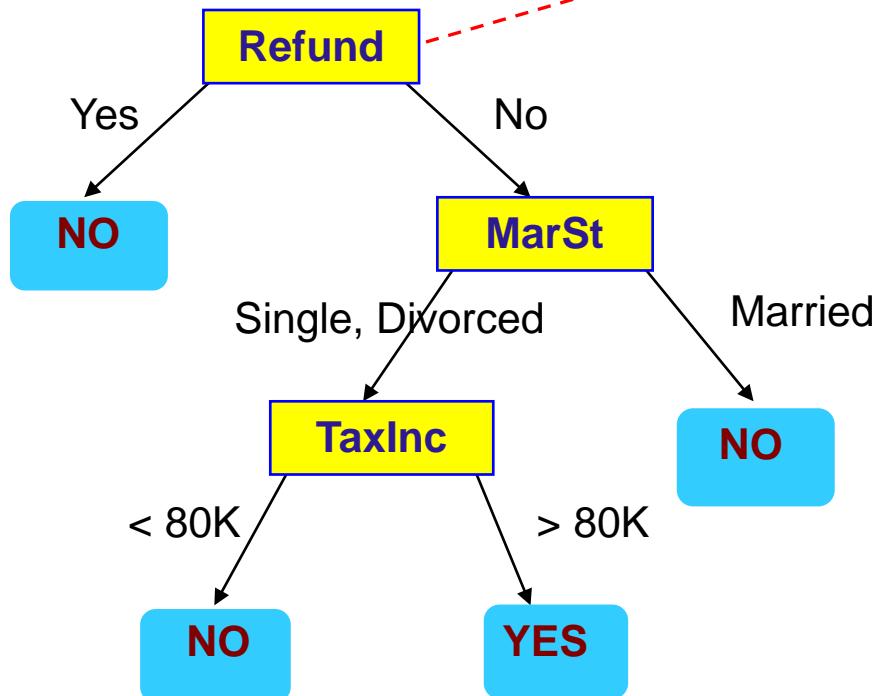
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

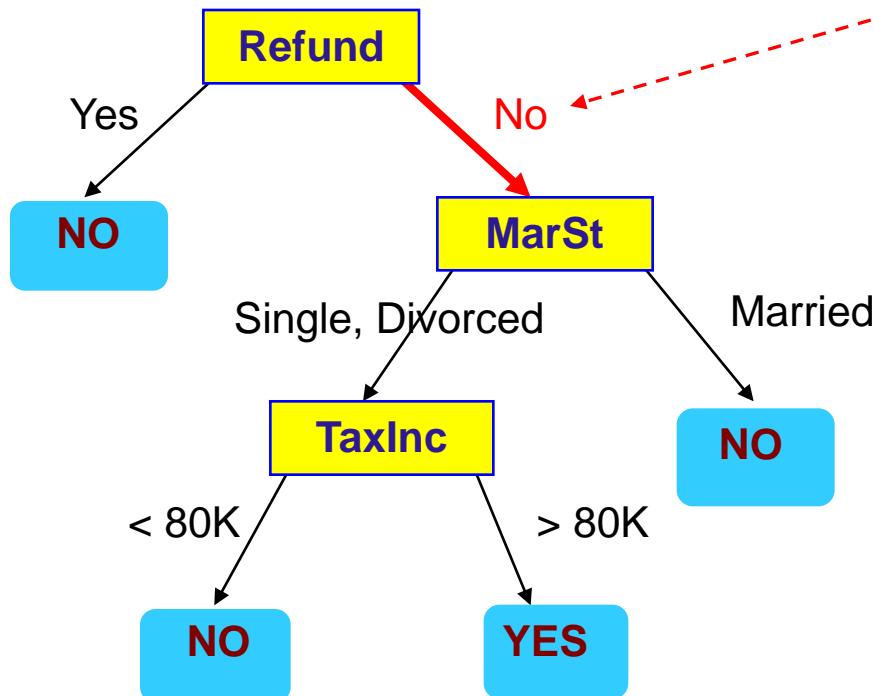
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

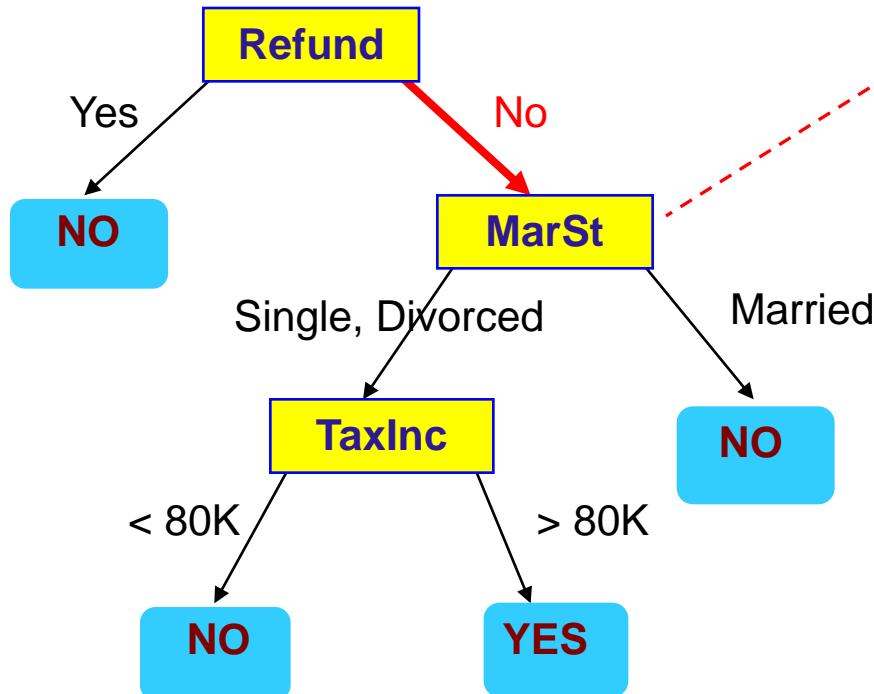
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

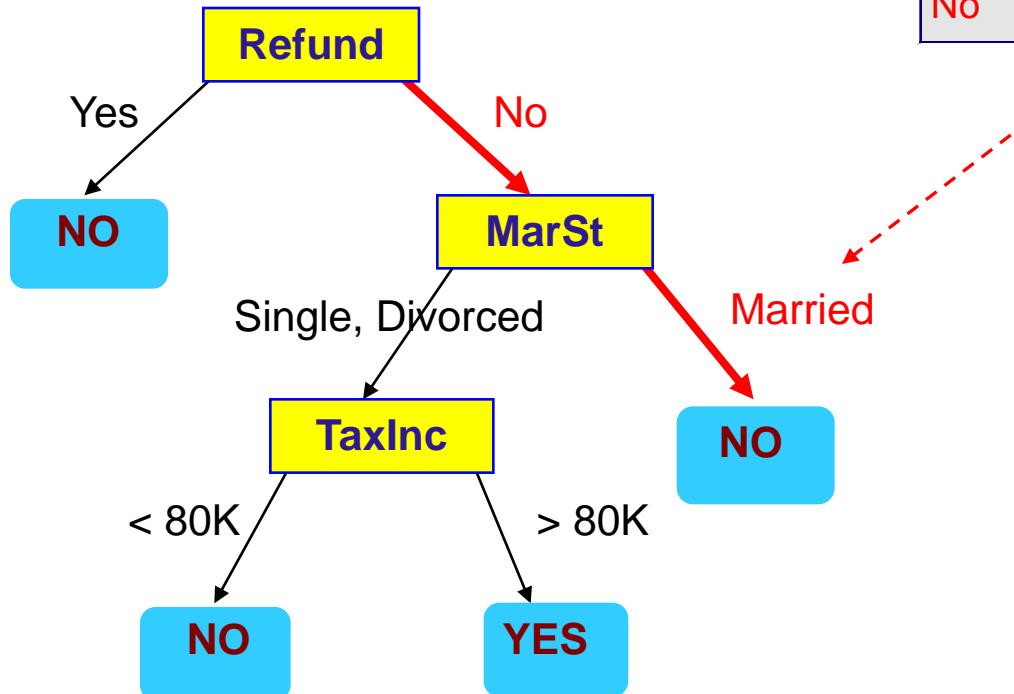
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

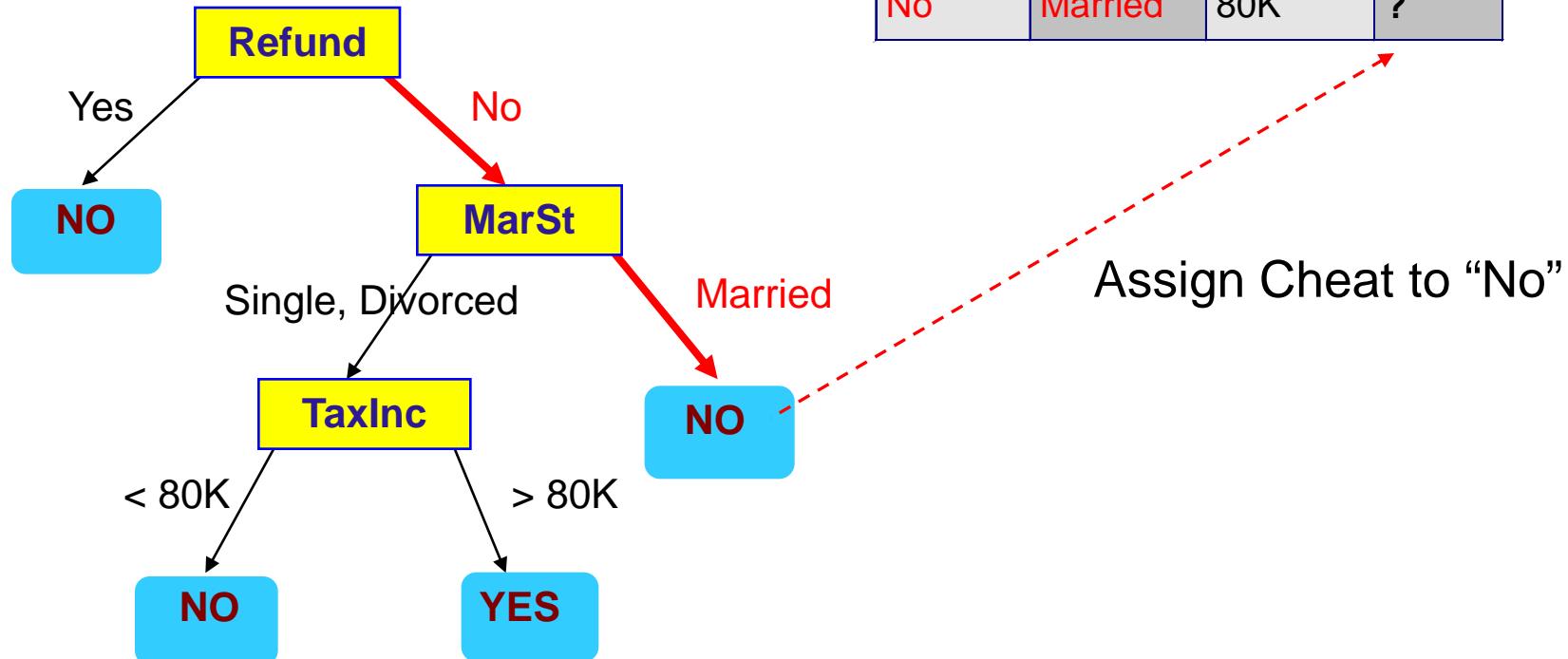
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Decision Tree Classification Task

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

A pink callout box labeled "Decision Tree Induction algorithm" has a green arrow pointing down to the value "55K". A red box highlights the row for Tid 12, and a red arrow points from the "Attrib3" column to the "Class" column. To the right of the table, there is a stack of three white cards labeled "Model", "Decision Tree", and "Data".

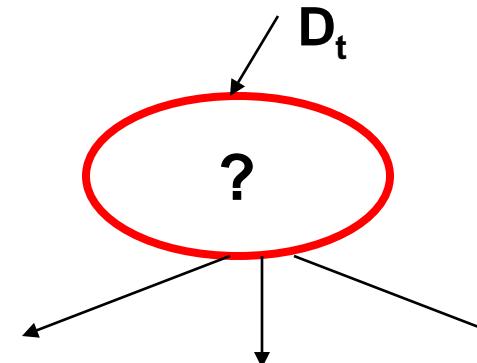
Decision Tree Induction

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

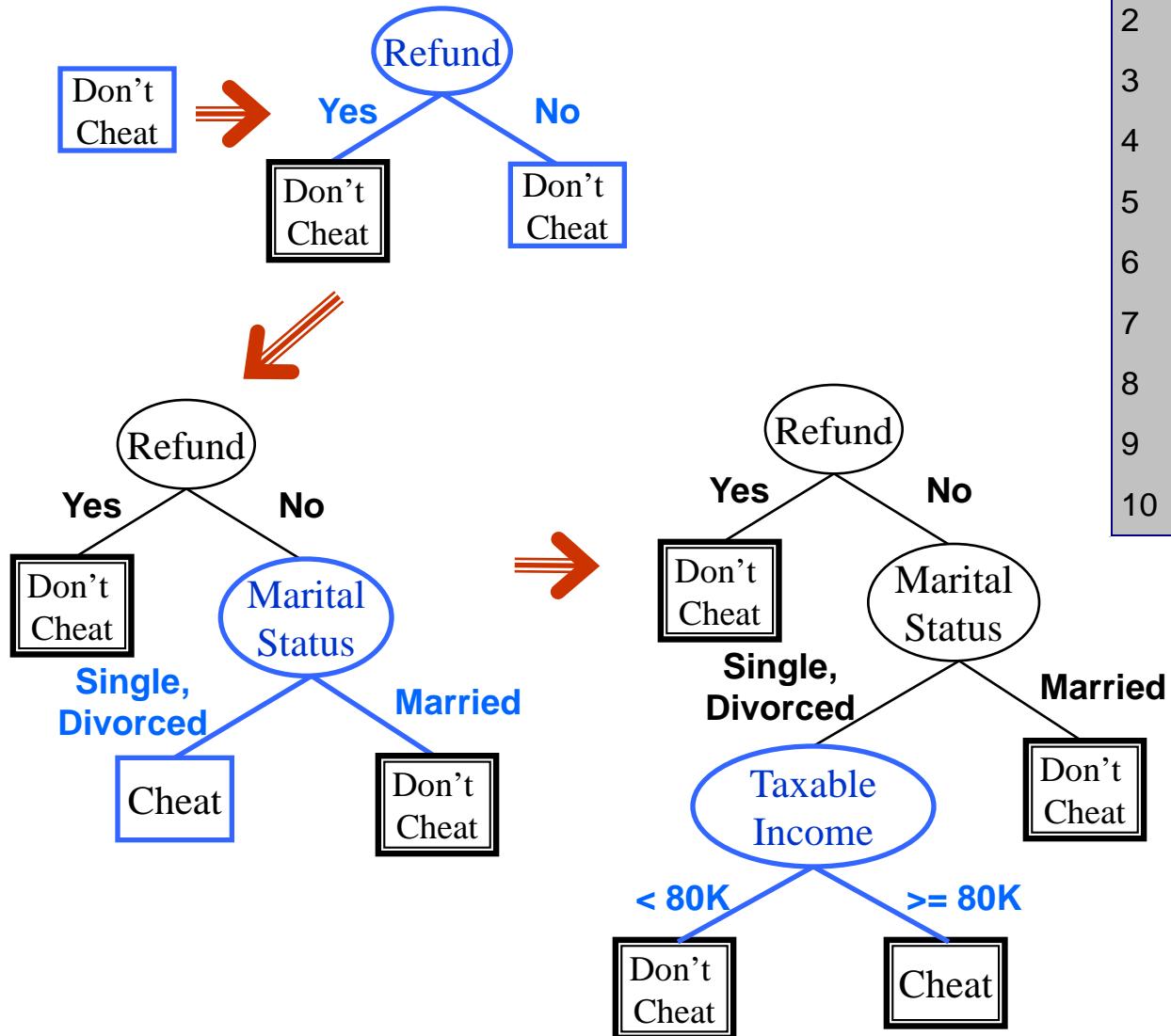
General Structure of Hunt's Algorithm

- Let D_t be the set of training records that reach a node t
- General Procedure:
 - If D_t contains records that belong 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.

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



Hunt's Algorithm



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

Tree Induction

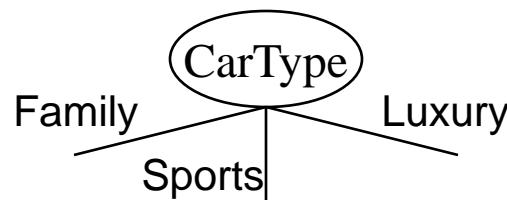
- Greedy strategy.
 - Split the records based on an attribute test that optimizes certain criterion.
- Issues
 - Determine how to split the records
 - ◆ How to specify the attribute test condition?
 - ◆ How to determine the best split?
 - Determine when to stop splitting

How to Specify Test Condition?

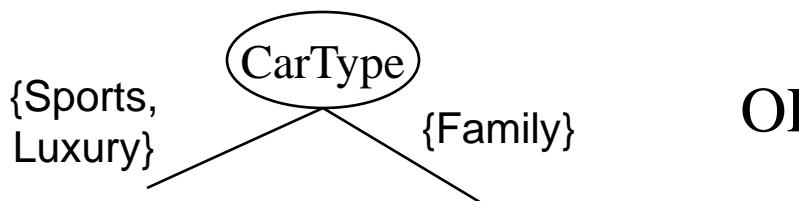
- Depends on attribute types
 - Nominal
 - Ordinal
 - Continuous
- Depends on number of ways to split
 - 2-way split
 - Multi-way split

Splitting Based on Nominal Attributes

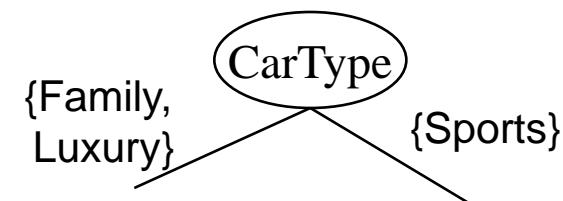
- **Multi-way split:** Use as many partitions as distinct values.



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

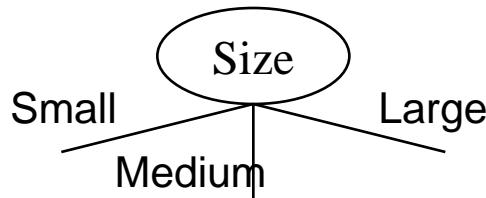


OR

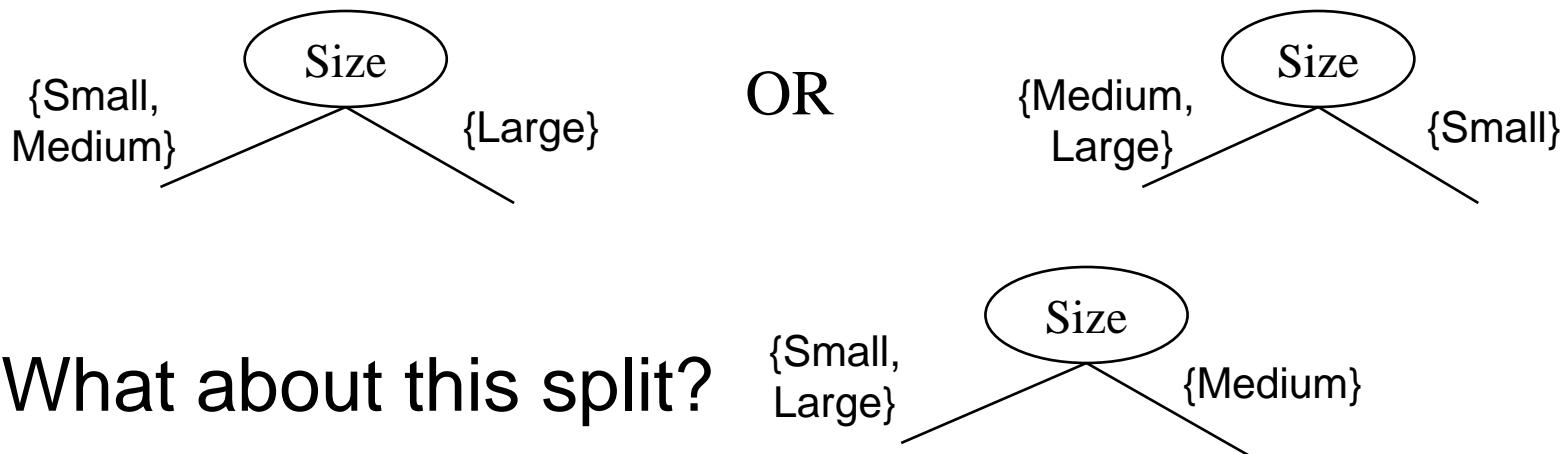


Splitting Based on Ordinal Attributes

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



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

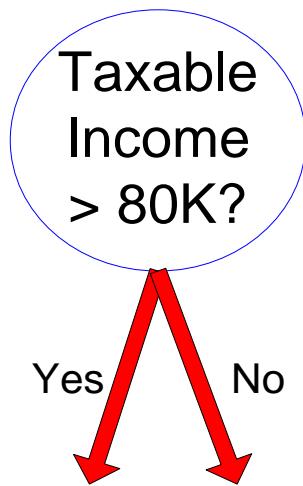


- What about this split?

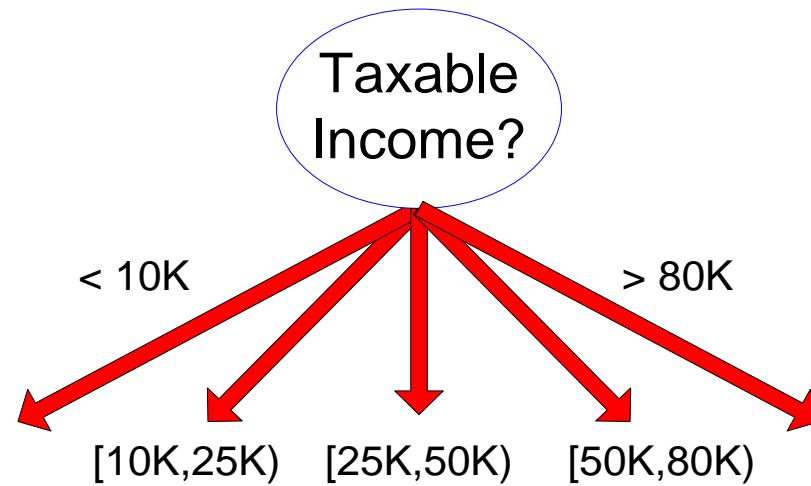
Splitting Based on Continuous Attributes

- 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



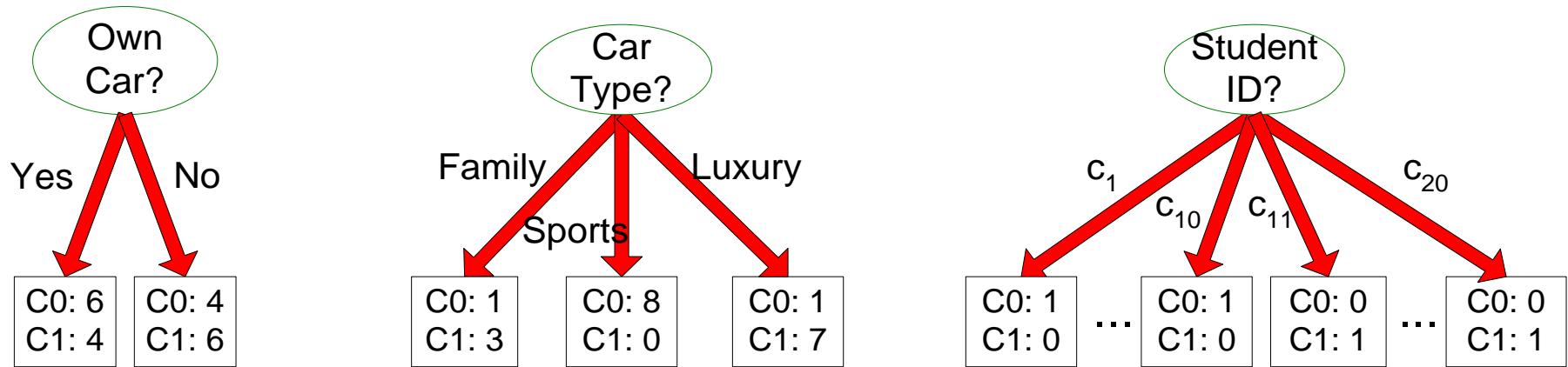
(ii) Multi-way split

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 determine the Best Split

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

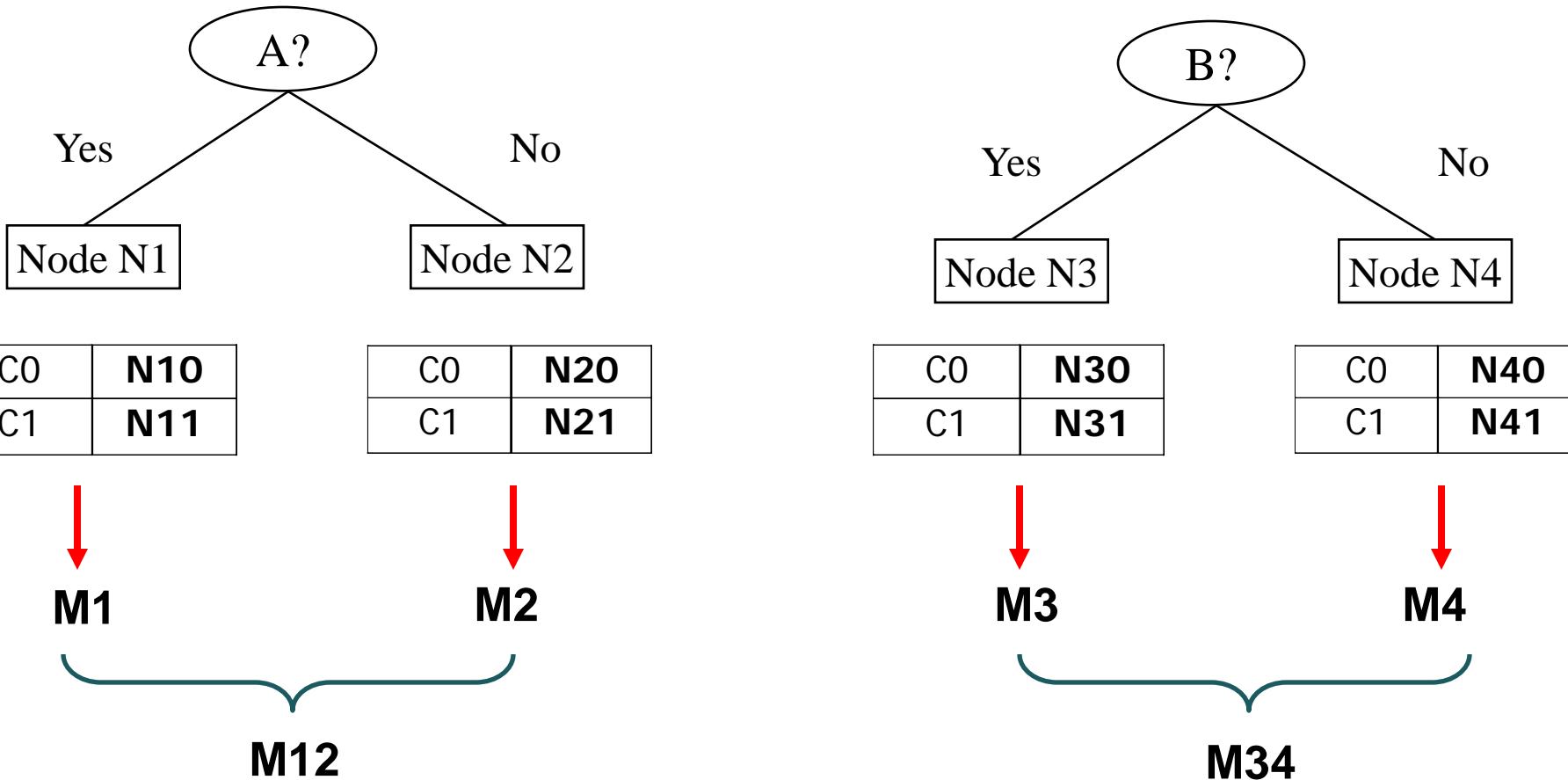
- Gini Index
- Entropy
- Misclassification error

How to Find the Best Split

Before Splitting:

C0	N00
C1	N01

→ M0



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

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

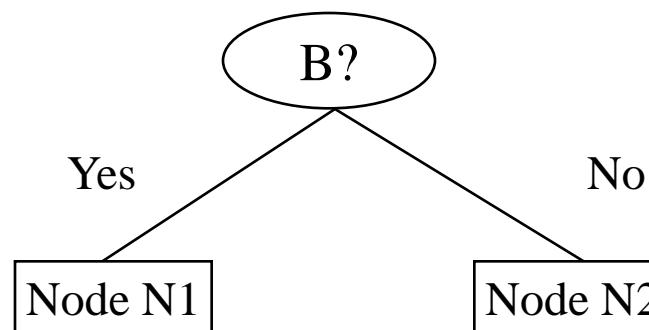
- Used in CART, SLIQ, SPRINT.
- When a node p is split into k partitions (children), the quality of split is computed as,

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

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.



Gini(N1)

$$\begin{aligned} &= 1 - (5/6)^2 - (2/6)^2 \\ &= 0.194 \end{aligned}$$

Gini(N2)

$$\begin{aligned} &= 1 - (1/6)^2 - (4/6)^2 \\ &= 0.528 \end{aligned}$$

	N1	N2
C1	5	1
C2	2	4
Gini = 0.333		

	Parent
C1	6
C2	6
Gini = 0.500	

Gini(Children)

$$\begin{aligned} &= 7/12 * 0.194 + \\ &\quad 5/12 * 0.528 \\ &= 0.333 \end{aligned}$$

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)

Two-way split (find best partition of values)

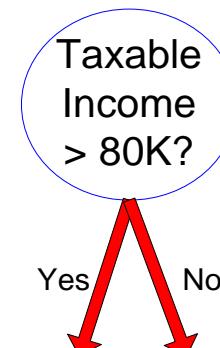
CarType			
	{Sports, Luxury}	{Family}	
C1	3	1	
C2	2	4	
Gini	0.400		

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

Alternative Splitting Criteria based on INFO

- Entropy at a given node t:

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

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

- Measures homogeneity of a node.
 - ◆ Maximum ($\log n_c$) when records are equally distributed among all classes implying least information
 - ◆ Minimum (0.0) when all records belong to one class, implying most information
- Entropy based computations are similar to the GINI index computations

Examples for computing Entropy

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

C1	0
C2	6

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

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

C1	1
C2	5

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

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

C1	2
C2	4

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

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

Splitting Based on INFO...

- Information Gain:

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

Parent Node, p is split into k partitions;

n_i is number of records in partition i

- Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
- Used in ID3 and C4.5
- Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

Splitting Based on INFO...

- Gain Ratio:

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

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

Parent Node, p is split into k partitions

n_i is the number of records in partition i

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

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

$$\begin{aligned}P(C1) &= 0/6 = 0 & P(C2) &= 6/6 = 1 \\Error &= 1 - \max(0, 1) = 1 - 1 = 0\end{aligned}$$

C1	1
C2	5

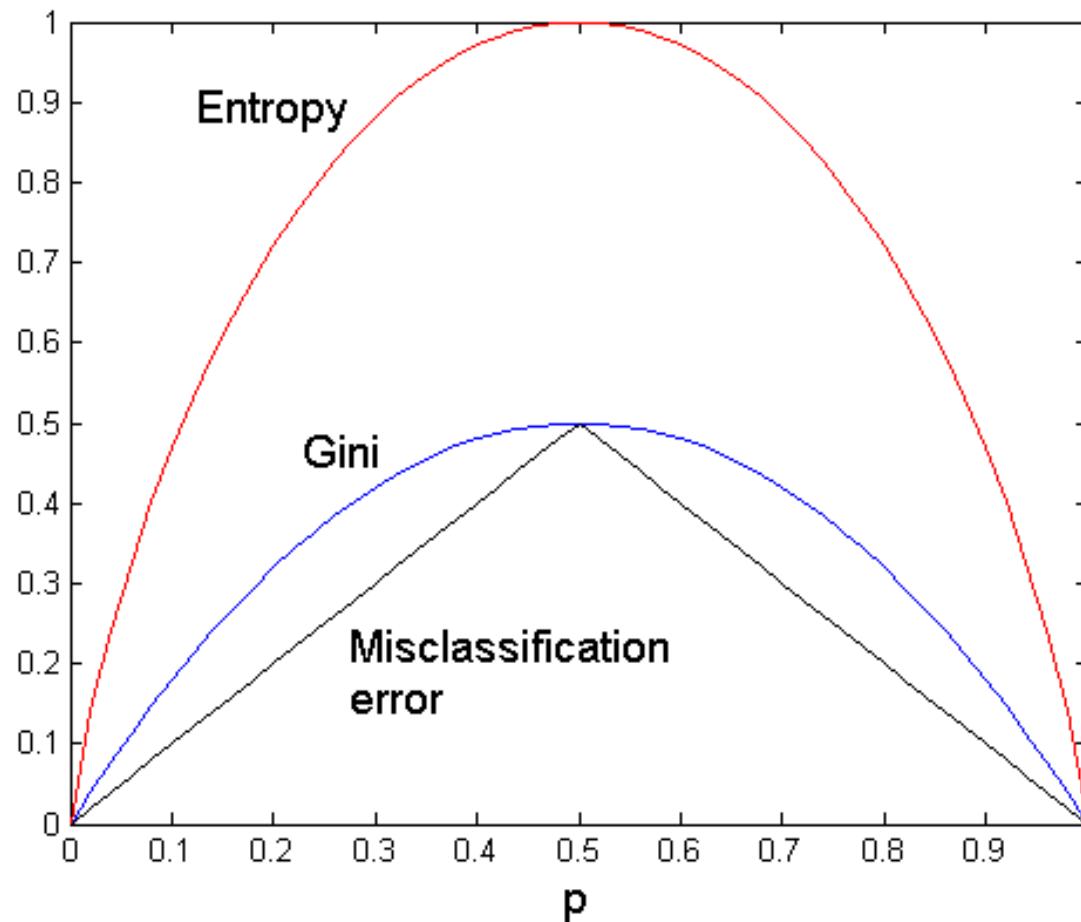
$$\begin{aligned}P(C1) &= 1/6 & P(C2) &= 5/6 \\Error &= 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6\end{aligned}$$

C1	2
C2	4

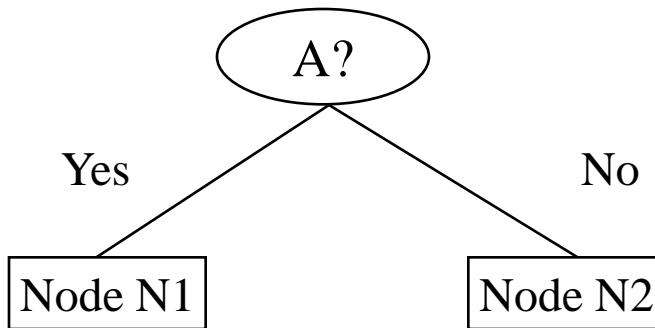
$$\begin{aligned}P(C1) &= 2/6 & P(C2) &= 4/6 \\Error &= 1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3\end{aligned}$$

Comparison among Splitting Criteria

For a 2-class problem:



Misclassification Error vs Gini



	Parent
C1	7
C2	3
Gini = 0.42	

Gini(N1)

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

$$= 0$$

Gini(N2)

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

$$= 0.489$$

	N1	N2
C1	3	4
C2	0	3
Gini=0.361		

Gini(Children)

$$= 3/10 * 0$$

$$+ 7/10 * 0.489$$

$$= 0.342$$

Gini improves !!

Tree Induction

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

Stopping Criteria for Tree Induction

- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination (to be discussed later)

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

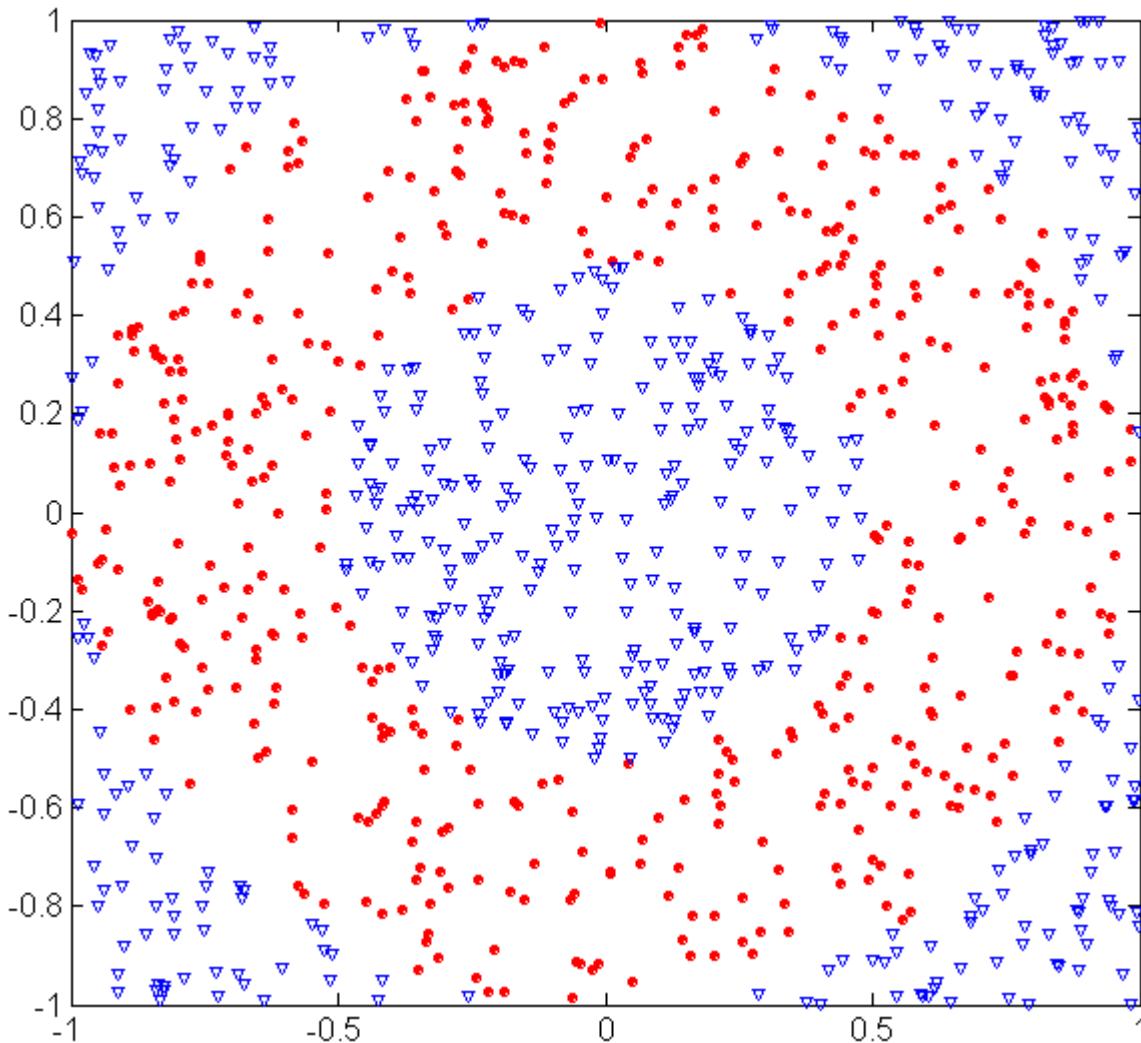
Example: C4.5

- Simple depth-first construction.
- Uses Information Gain
- Sorts Continuous Attributes at each node.
- Needs entire data to fit in memory.
- Unsuitable for Large Datasets.
 - Needs out-of-core sorting.
- You can download the software from:
<http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz>

Practical Issues of Classification

- Underfitting and Overfitting
- Missing Values
- Costs of Classification

Underfitting and Overfitting (Example)



500 circular and 500 triangular data points.

Circular points:

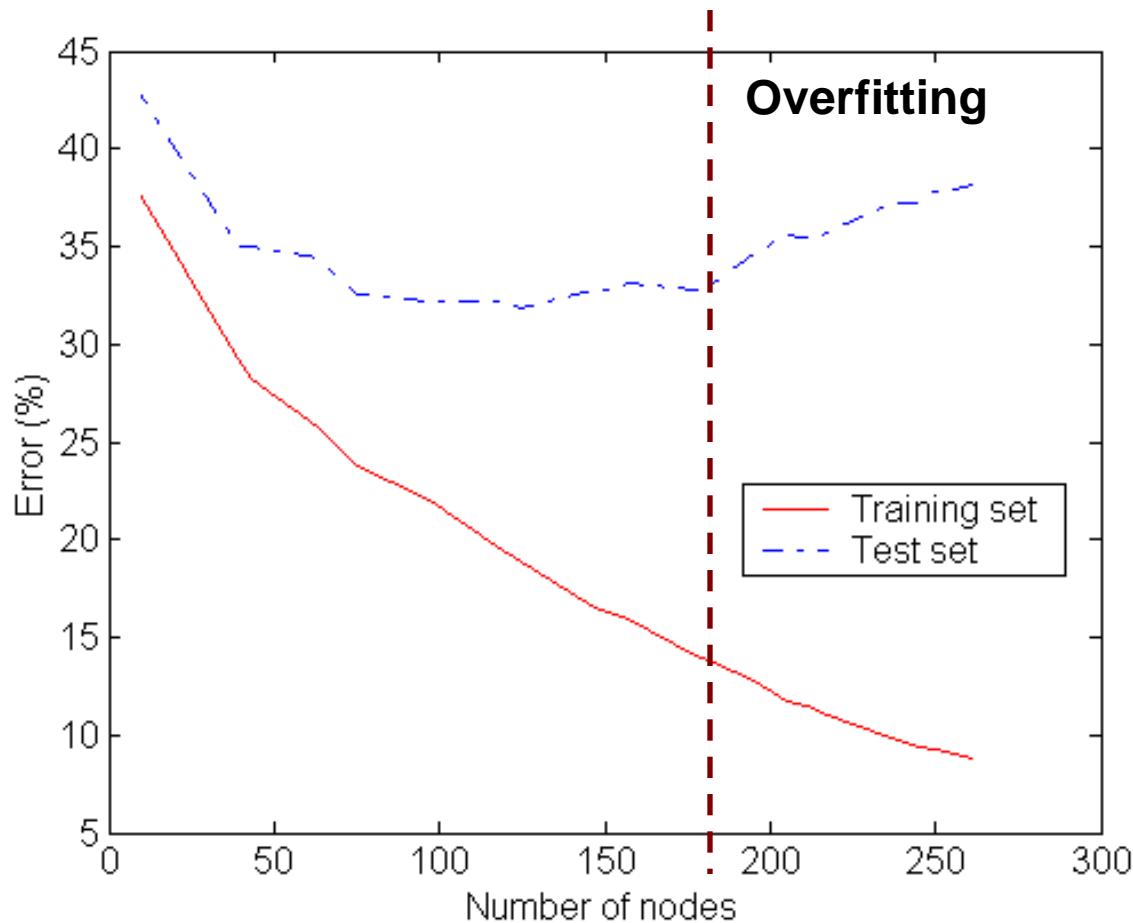
$$0.5 \leq \sqrt{x_1^2 + x_2^2} \leq 1$$

Triangular points:

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

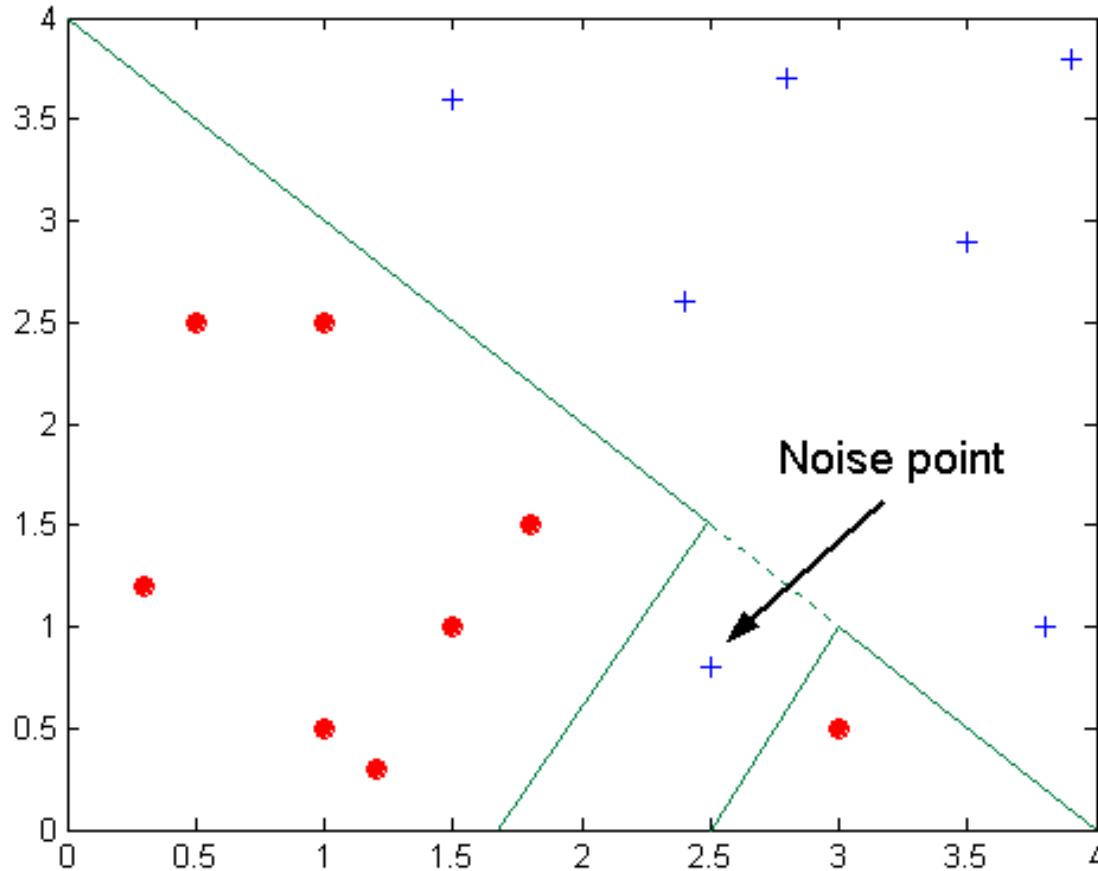
$$\sqrt{x_1^2 + x_2^2} < 1$$

Underfitting and Overfitting



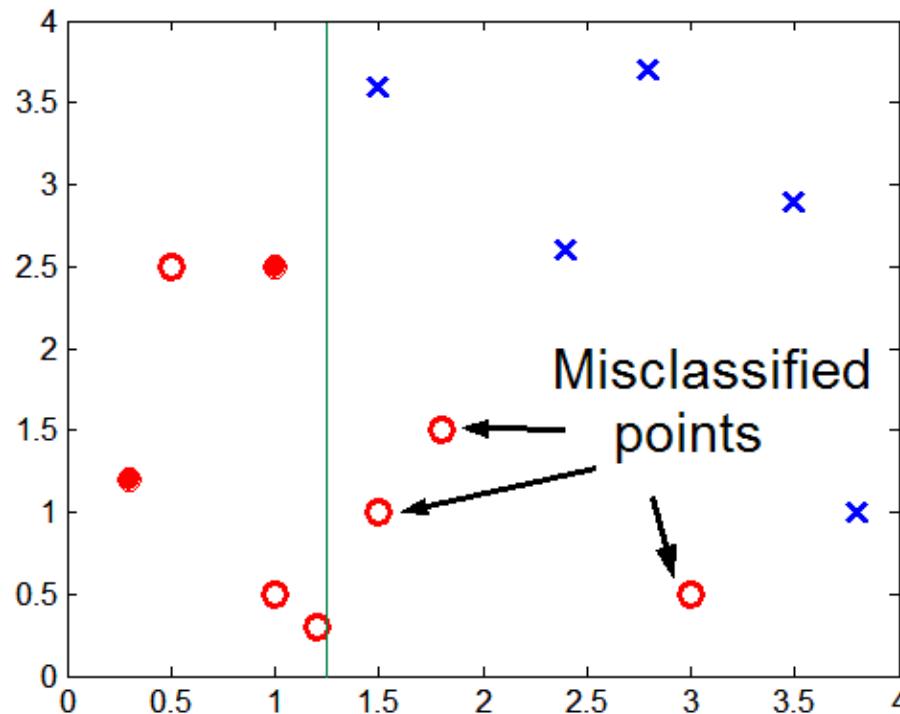
Underfitting: when model is too simple, both training and test errors are large

Overfitting due to Noise



Decision boundary is distorted by noise point

Overfitting due to Insufficient Examples



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

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

Notes on Overfitting

- Overfitting results in decision trees that are more complex than necessary
- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records
- Need new ways for estimating errors

Estimating Generalization Errors

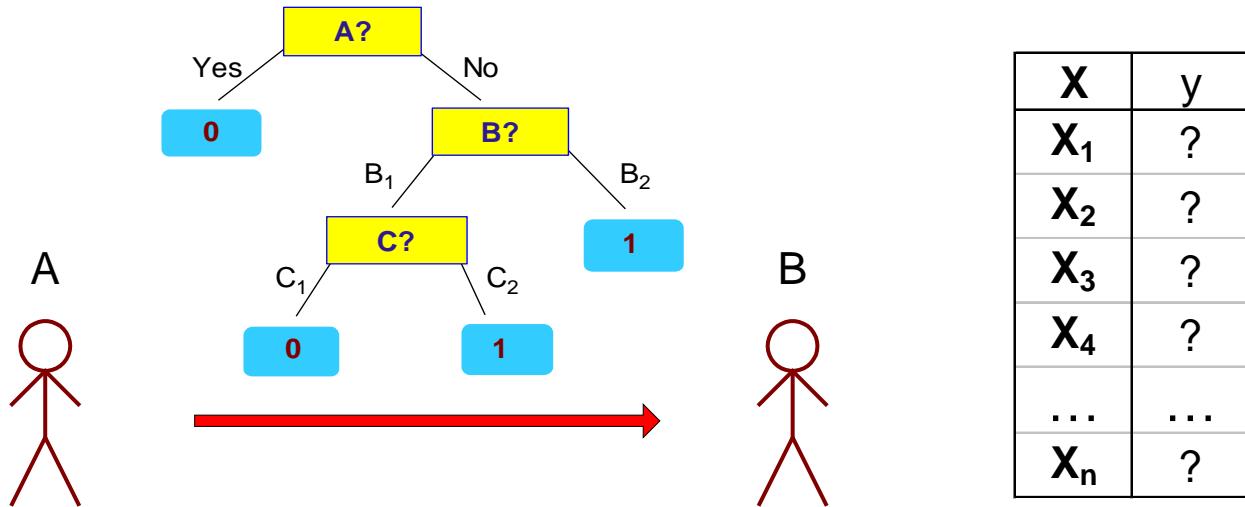
- Re-substitution errors: error on training ($\sum e(t)$)
- Generalization errors: error on testing ($\sum e'(t)$)
- Methods for estimating generalization errors:
 - Optimistic approach: $e'(t) = e(t)$
 - Pessimistic approach:
 - ◆ For each leaf node: $e'(t) = (e(t)+0.5)$
 - ◆ Total errors: $e'(T) = e(T) + N \times 0.5$ (N: number of leaf nodes)
 - ◆ For a tree with 30 leaf nodes and 10 errors on training (out of 1000 instances):
Training error = $10/1000 = 1\%$
Generalization error = $(10 + 30 \times 0.5)/1000 = 2.5\%$
 - Reduced error pruning (REP):
 - ◆ uses validation data set to estimate generalization error

Occam's Razor

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

Minimum Description Length (MDL)

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



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

How to Address Overfitting

- Pre-Pruning (Early Stopping Rule)
 - Stop the algorithm before it becomes a fully-grown tree
 - Typical 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:
 - ◆ 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 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
 - Can use MDL for post-pruning

Example of Post-Pruning

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

Training Error (Before splitting) = 10/30

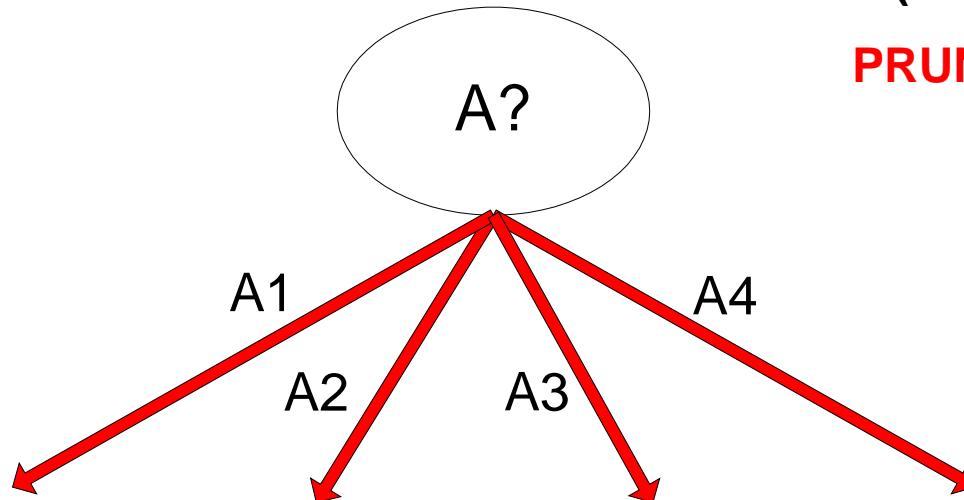
Pessimistic error = $(10 + 0.5)/30 = 10.5/30$

Training Error (After splitting) = 9/30

Pessimistic error (After splitting)

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

PRUNE!



Class = Yes	8
Class = No	4

Class = Yes	3
Class = No	4

Class = Yes	4
Class = No	1

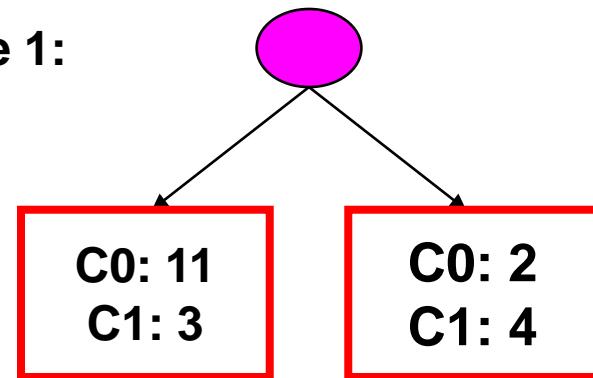
Class = Yes	5
Class = No	1

Examples of Post-pruning

- Optimistic error?

Don't prune for both cases

Case 1:



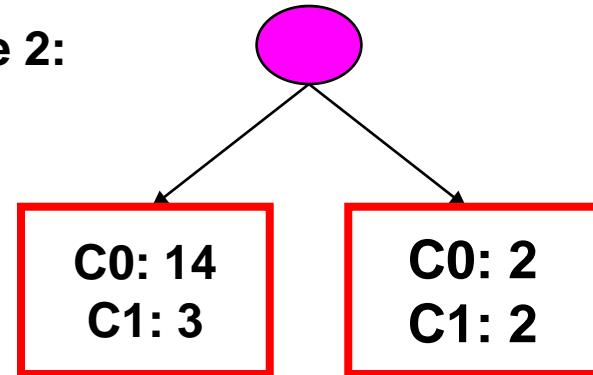
- Pessimistic error?

Don't prune case 1, prune case 2

- Reduced error pruning?

Depends on validation set

Case 2:



Handling Missing Attribute Values

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

Computing Impurity Measure

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

Missing value

Before Splitting:

Entropy(Parent)

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

	Class = Yes	Class = No
Refund=Yes	0	3
Refund=No	2	4
Refund=?	1	0

Split on Refund:

Entropy(Refund=Yes) = 0

Entropy(Refund=No)

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

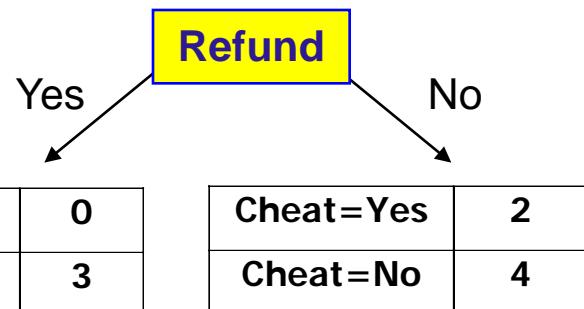
Entropy(Children)

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

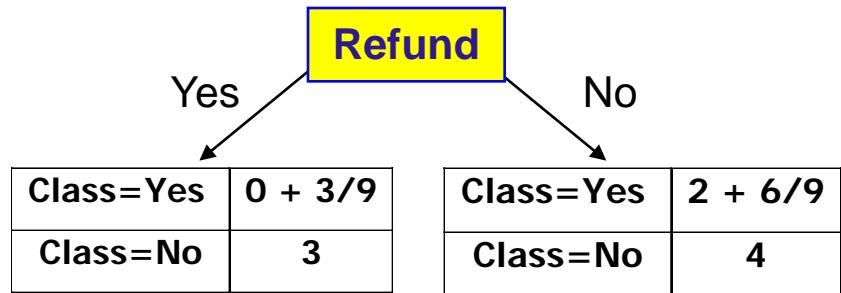
$$\text{Gain} = 0.9 \times (0.8813 - 0.551) = 0.3303$$

Distribute Instances

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



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



Probability that Refund=Yes is 3/9

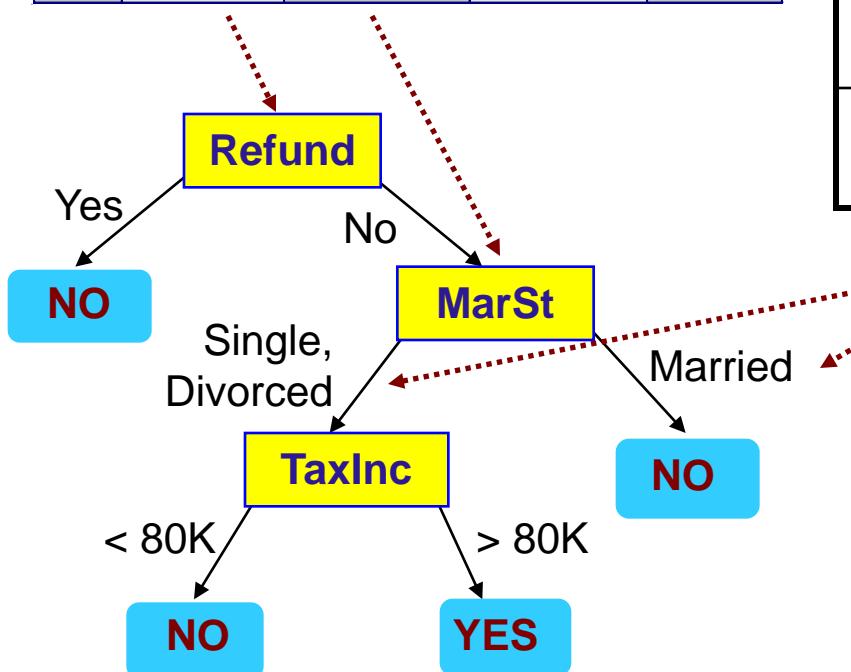
Probability that Refund=No is 6/9

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

Classify Instances

New record:

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



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

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

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

Other Issues

- Data Fragmentation
- Search Strategy
- Expressiveness
- Tree Replication

Data Fragmentation

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

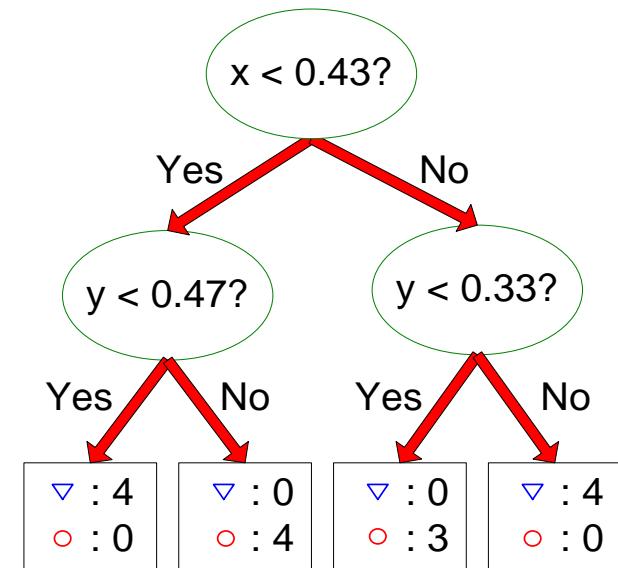
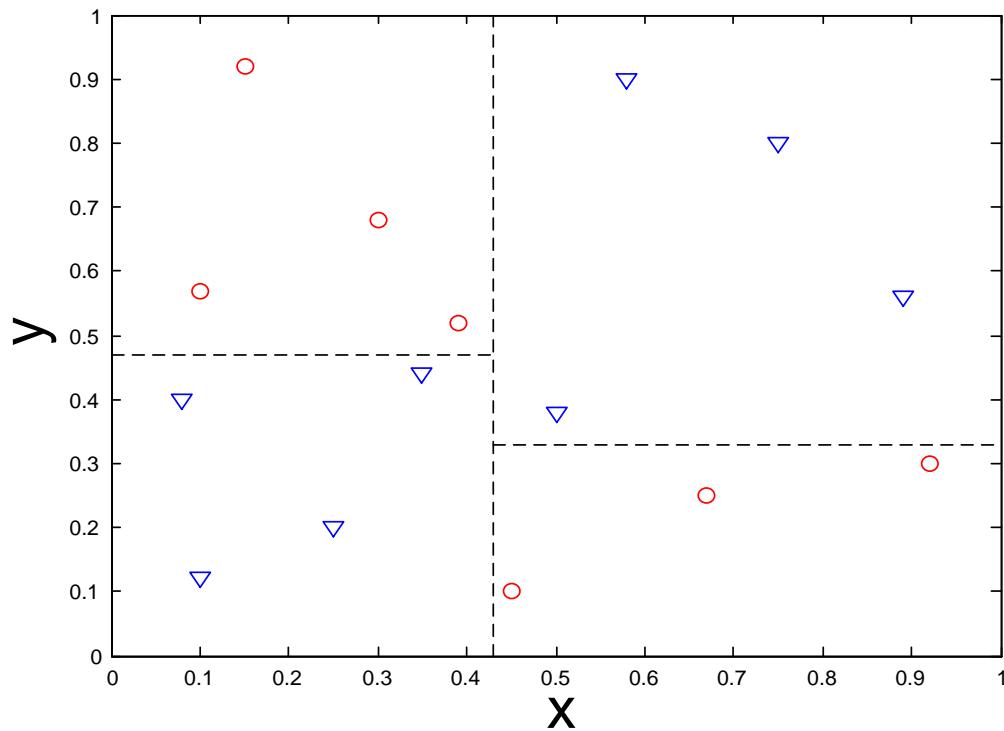
Search Strategy

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

Expressiveness

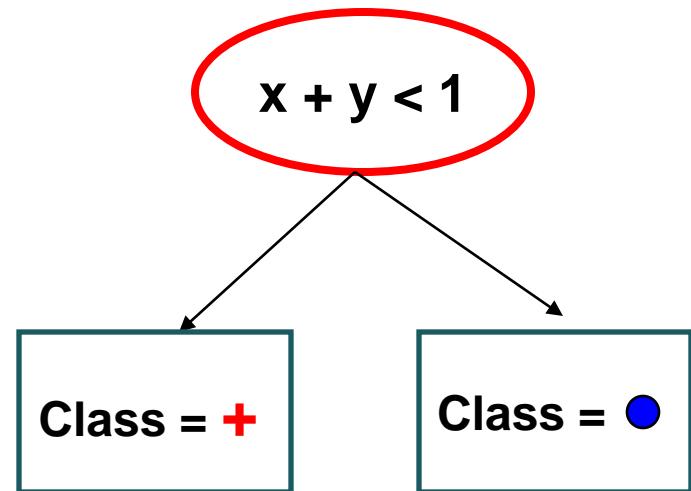
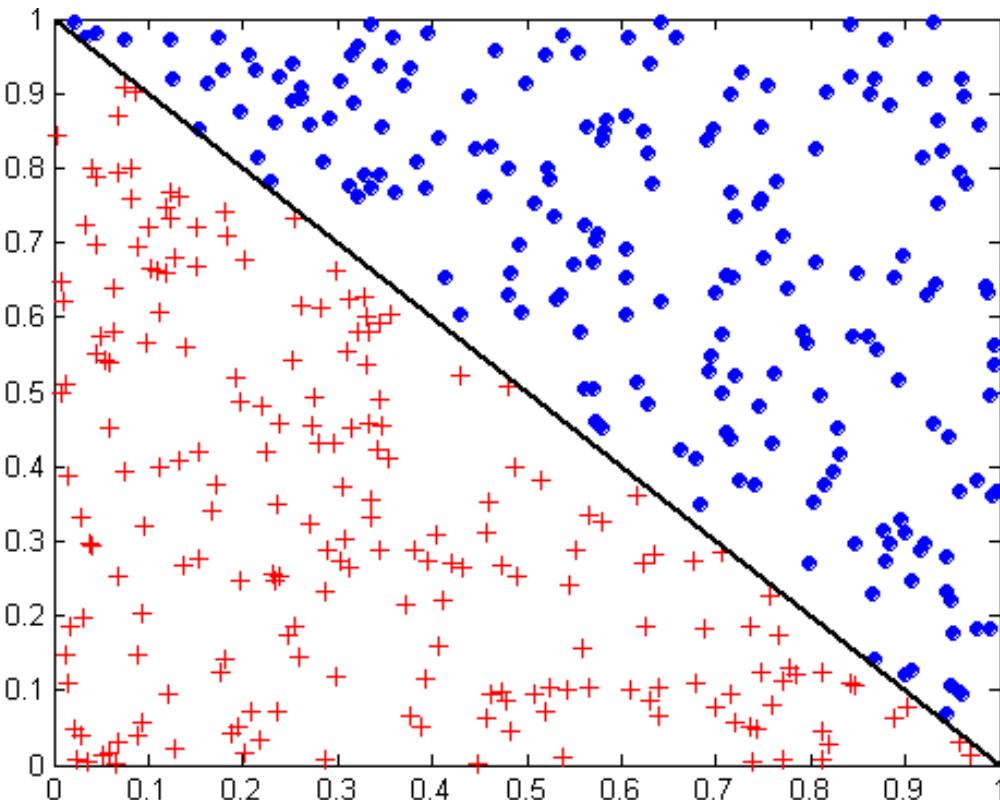
- Decision tree provides expressive representation for learning discrete-valued function
 - But they do not generalize well to certain types of Boolean functions
 - ◆ Example: parity function:
 - Class = 1 if there is an even number of Boolean attributes with truth value = True
 - Class = 0 if there is an odd number of Boolean attributes with truth value = True
 - ◆ For accurate modeling, must have a complete tree
- Not expressive enough for modeling continuous variables
 - Particularly when test condition involves only a single attribute at-a-time

Decision Boundary



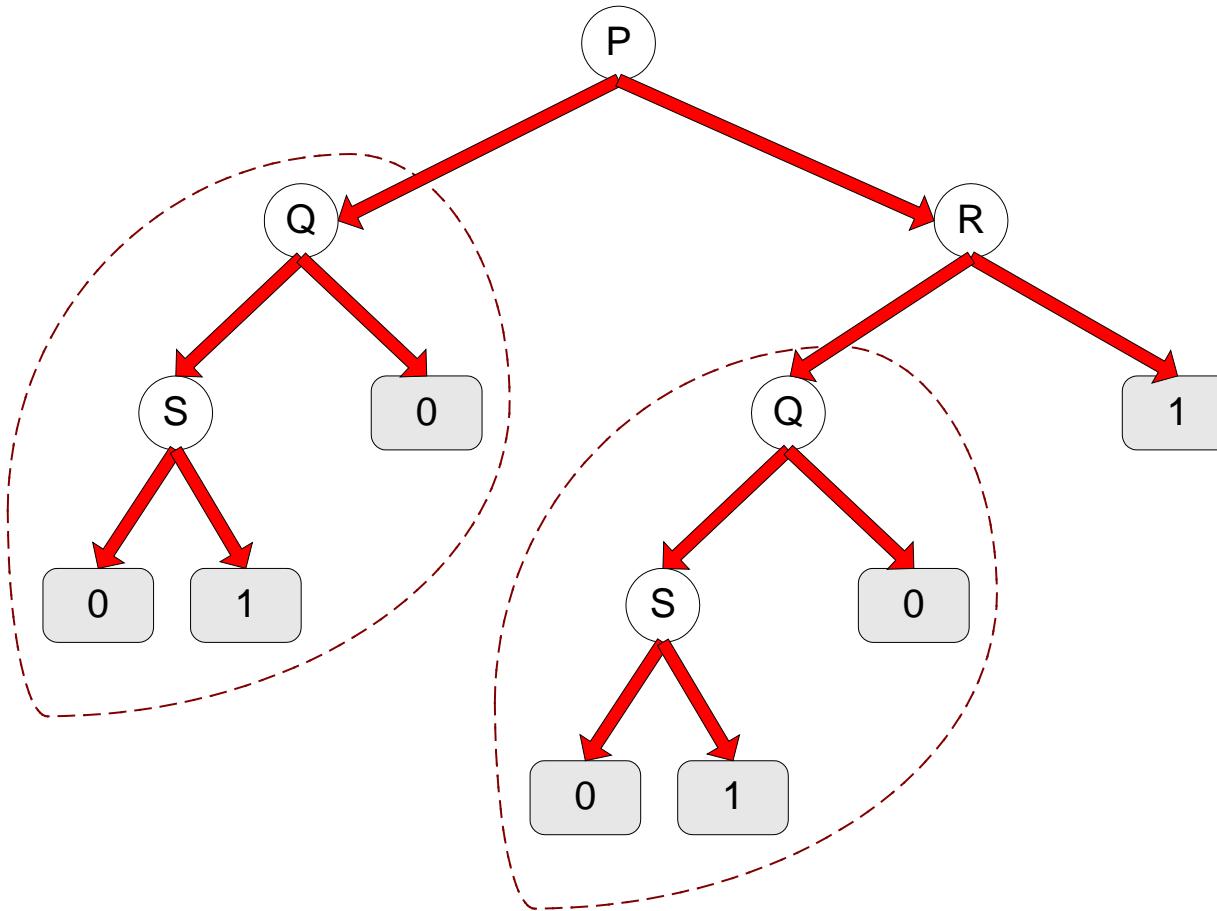
- Border line between two neighboring regions of different classes is known as **decision boundary**
- Decision boundary is parallel to axes because test condition involves a single attribute at-a-time

Oblique Decision Trees



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

Tree Replication



- Same subtree appears in multiple branches

Model Evaluation

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

Model Evaluation

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

Metrics for Performance Evaluation

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

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

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

Metrics for Performance Evaluation...

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

- Most widely-used metric:

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

Limitation of Accuracy

- Consider a 2-class problem
 - Number of Class 0 examples = 9990
 - Number of Class 1 examples = 10
- If model predicts everything to be class 0, accuracy is $9990/10000 = 99.9\%$
 - Accuracy is misleading because model does not detect any class 1 example

Cost Matrix

		PREDICTED CLASS		
		C(i j)	Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	C(Yes Yes)	C(No Yes)	
	Class>No	C(Yes No)	C(No No)	

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

Computing Cost of Classification

Cost Matrix		PREDICTED CLASS	
ACTUAL CLASS	C(i j)	+	-
	+	-1	100
	-	1	0

Model M ₁	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

Accuracy = 80%

Cost = 3910

Model M ₂	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	250	45
	-	5	200

Accuracy = 90%

Cost = 4255

Cost vs Accuracy

Count	PREDICTED CLASS	
ACTUAL CLASS	Class=Yes	Class>No
	Class=Yes	a
	Class>No	d

Accuracy is proportional to cost if

1. $C(Yes|No)=C(No|Yes) = q$
2. $C(Yes|Yes)=C(No|No) = p$

$$N = a + b + c + d$$

$$\text{Accuracy} = (a + d)/N$$

Cost	PREDICTED CLASS	
ACTUAL CLASS	Class=Yes	Class>No
	Class=Yes	p
	Class>No	q

$$\text{Cost} = p (a + d) + q (b + c)$$

$$= p (a + d) + q (N - a - d)$$

$$= q N - (q - p)(a + d)$$

$$= N [q - (q-p) \times \text{Accuracy}]$$

Cost-Sensitive Measures

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

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

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

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

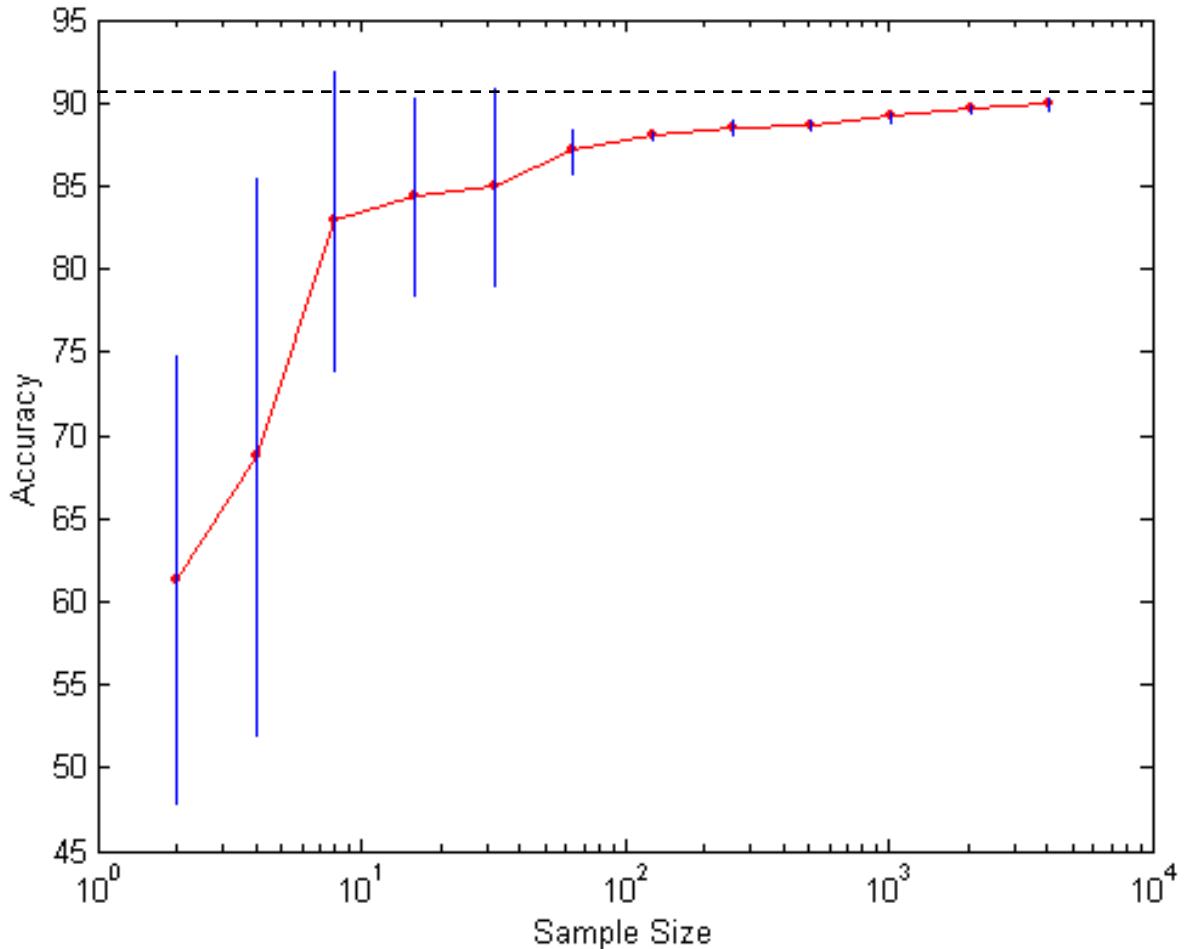
Model Evaluation

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

Methods for Performance Evaluation

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

Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve:
 - Arithmetic sampling (Langley, et al)
 - Geometric sampling (Provost et al)

Effect of small sample size:

- Bias in the estimate
- Variance of estimate

Methods of Estimation

- Holdout
 - Reserve 2/3 for training and 1/3 for testing
- Random subsampling
 - Repeated holdout
- Cross validation
 - Partition data into k disjoint subsets
 - k -fold: train on $k-1$ partitions, test on the remaining one
 - Leave-one-out: $k=n$
- Stratified sampling
 - oversampling vs undersampling
- Bootstrap
 - Sampling with replacement

Model Evaluation

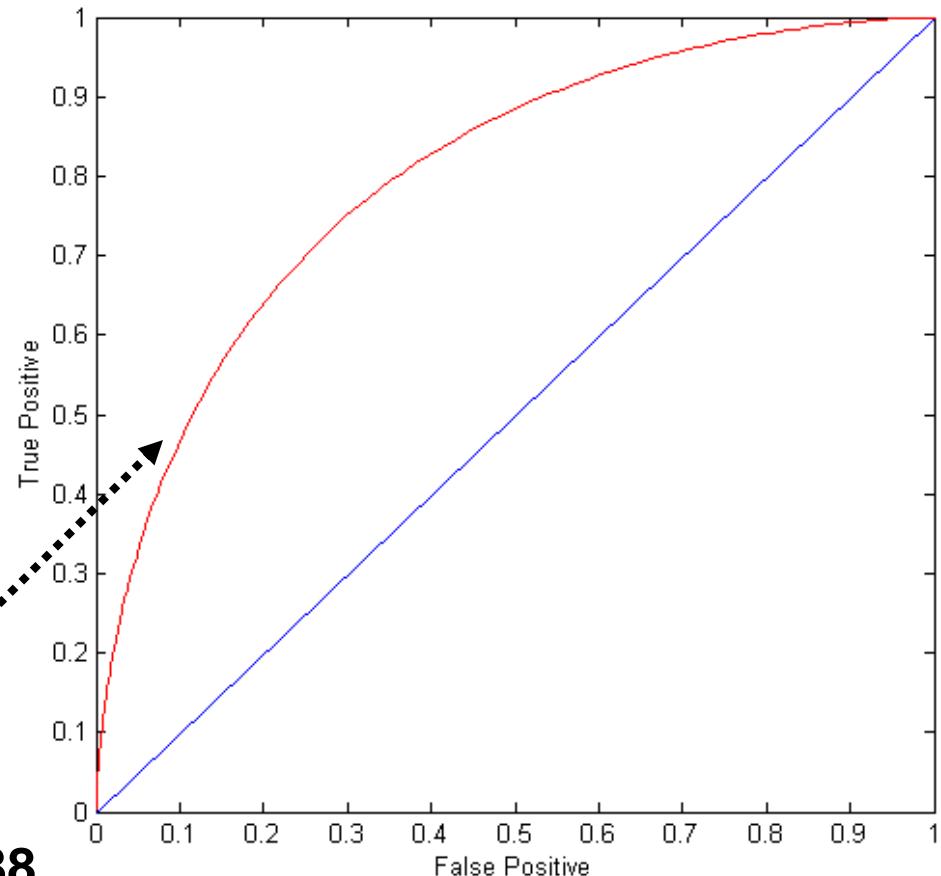
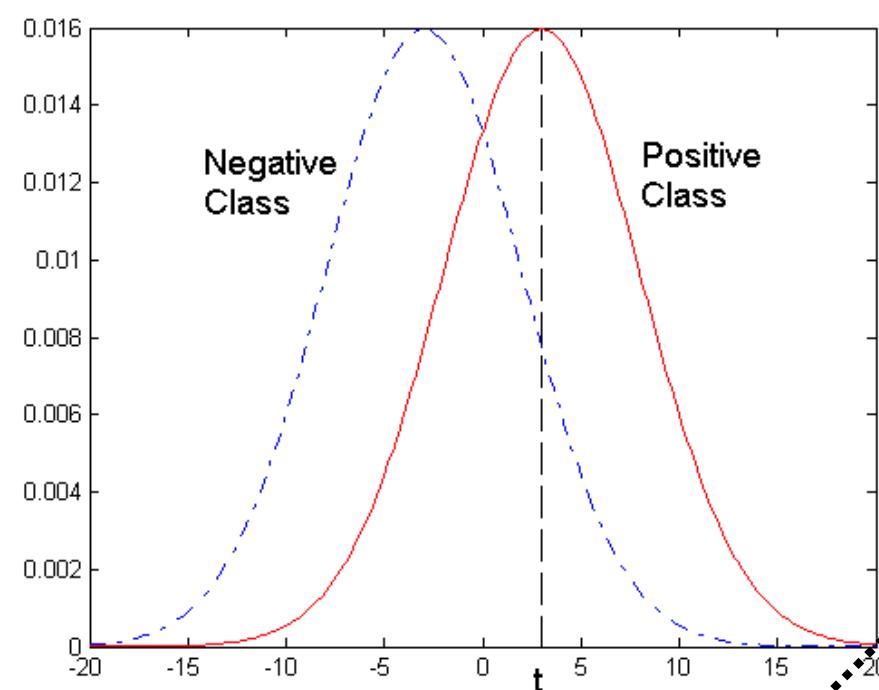
- Metrics for Performance Evaluation
 - How to evaluate the performance of a model?
- Methods for Performance Evaluation
 - How to obtain reliable estimates?
- Methods for Model Comparison
 - How to compare the relative performance among competing models?

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
 - Characterize the trade-off between positive hits and false alarms
- ROC curve plots TP (on the y-axis) against FP (on the x-axis)
- Performance of each classifier represented as a point on the ROC curve
 - changing the threshold of algorithm, sample distribution or cost matrix changes the location of the point

ROC Curve

- 1-dimensional data set containing 2 classes (positive and negative)
- any points located at $x > t$ is classified as positive



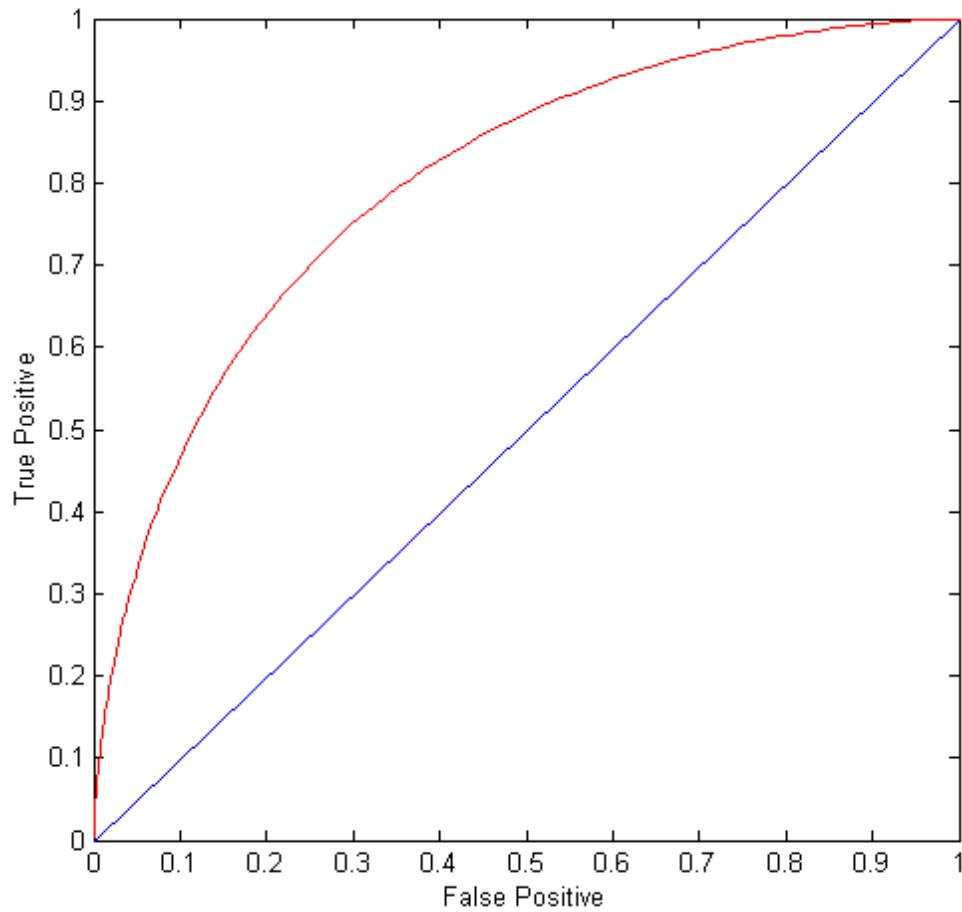
At threshold t :

TP=0.5, FN=0.5, FP=0.12, FN=0.88

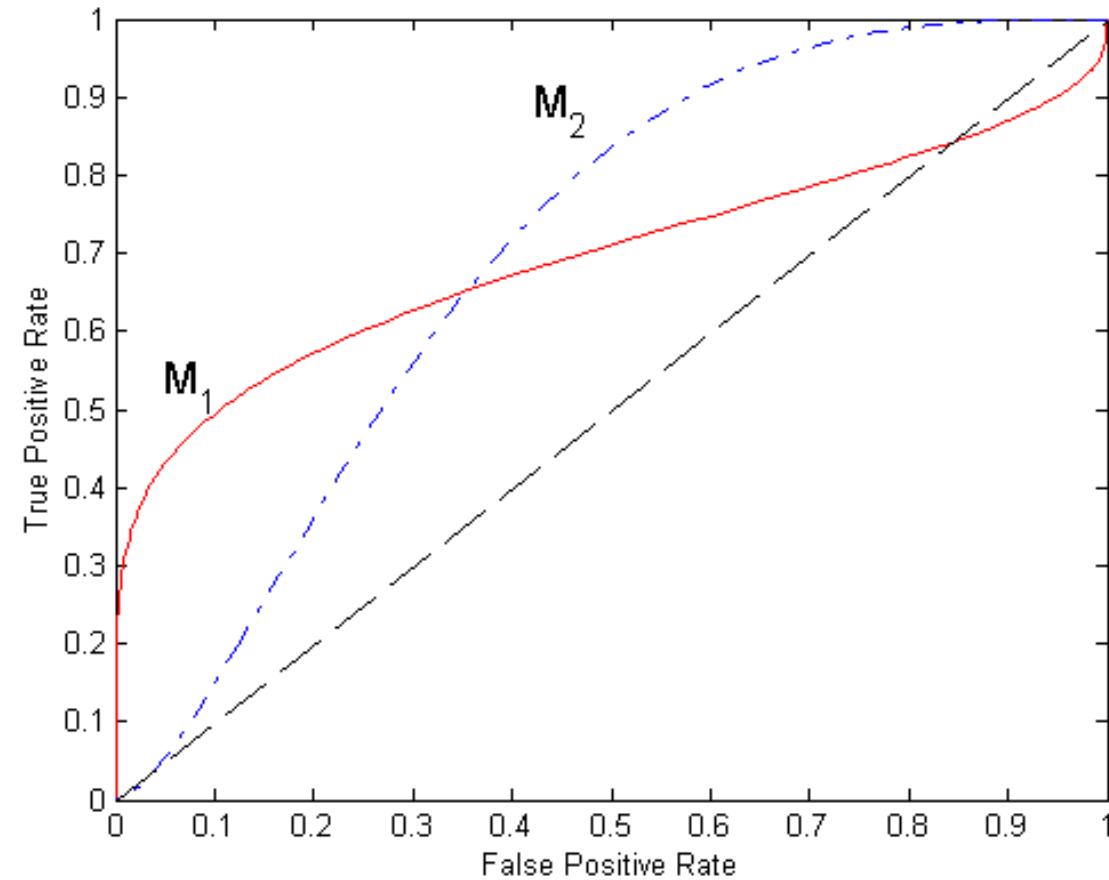
ROC Curve

(TP,FP):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal
- Diagonal line:
 - Random guessing
 - Below diagonal line:
 - ◆ prediction is opposite of the true class



Using ROC for Model Comparison



- No model consistently outperform the other
 - M_1 is better for small FPR
 - M_2 is better for large FPR
- Area Under the ROC curve
 - Ideal:
 - Area = 1
 - Random guess:
 - Area = 0.5

How to Construct an ROC curve

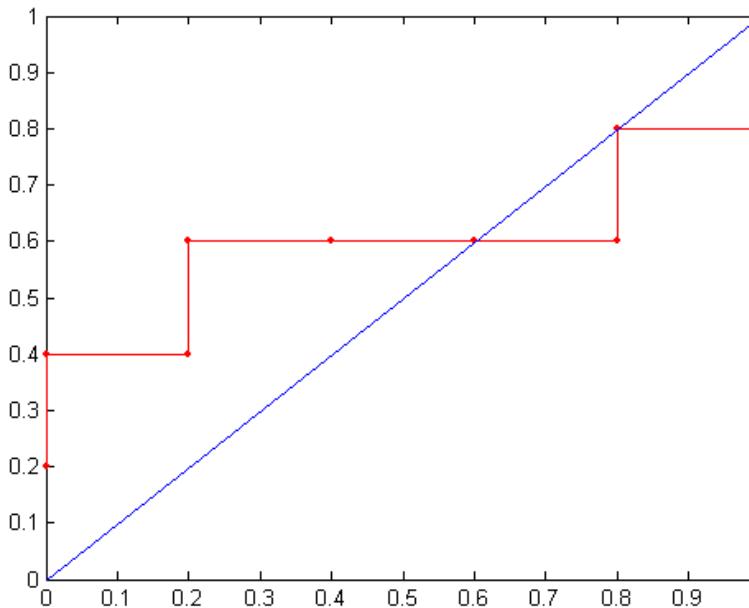
Instance	$P(+ A)$	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Use classifier that produces posterior probability for each test instance $P(+|A)$
- Sort the instances according to $P(+|A)$ in decreasing order
- Apply threshold at each unique value of $P(+|A)$
- Count the number of TP, FP, TN, FN at each threshold
- TP rate, $TPR = TP/(TP+FN)$
- FP rate, $FPR = FP/(FP + TN)$

How to construct an ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
→ TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
→ FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

ROC Curve:



Test of Significance

- Given two models:
 - Model M1: accuracy = 85%, tested on 30 instances
 - Model M2: accuracy = 75%, tested on 5000 instances
- Can we say M1 is better than M2?
 - How much confidence can we place on accuracy of M1 and M2?
 - Can the difference in performance measure be explained as a result of random fluctuations in the test set?

Confidence Interval for Accuracy

- Prediction can be regarded as a Bernoulli trial
 - A Bernoulli trial has 2 possible outcomes
 - Possible outcomes for prediction: correct or wrong
 - Collection of Bernoulli trials has a Binomial distribution:
 - ◆ $x \sim \text{Bin}(N, p)$ x : number of correct predictions
 - ◆ e.g: Toss a fair coin 50 times, how many heads would turn up?
Expected number of heads = $N \times p = 50 \times 0.5 = 25$
- Given x (# of correct predictions) or equivalently, $\text{acc} = x/N$, and N (# of test instances),

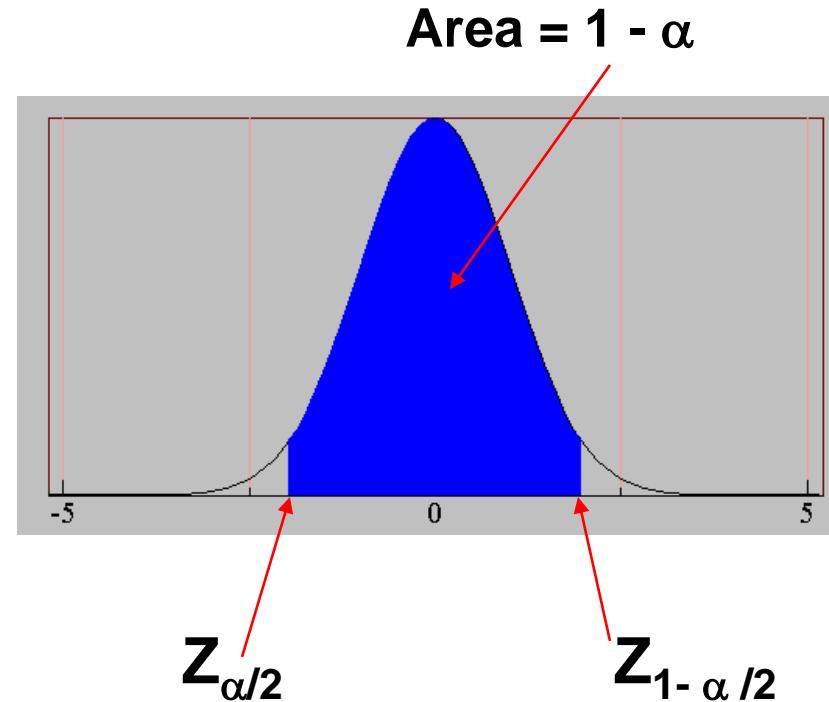
Can we predict p (true accuracy of model)?

Confidence Interval for Accuracy

- For large test sets ($N > 30$),
 - acc has a normal distribution with mean p and variance $p(1-p)/N$

$$P(Z_{\alpha/2} < \frac{acc - p}{\sqrt{p(1-p)/N}} < Z_{1-\alpha/2})$$

$$= 1 - \alpha$$



- Confidence Interval for p :

$$p = \frac{2 \times N \times acc + Z_{\alpha/2}^2 \pm \sqrt{Z_{\alpha/2}^2 + 4 \times N \times acc - 4 \times N \times acc^2}}{2(N + Z_{\alpha/2}^2)}$$

Confidence Interval for Accuracy

- Consider a model that produces an accuracy of 80% when evaluated on 100 test instances:
 - $N=100$, $acc = 0.8$
 - Let $1-\alpha = 0.95$ (95% confidence)
 - From probability table, $Z_{\alpha/2}=1.96$

N	50	100	500	1000	5000
p(lower)	0.670	0.711	0.763	0.774	0.789
p(upper)	0.888	0.866	0.833	0.824	0.811

1- α	Z
0.99	2.58
0.98	2.33
0.95	1.96
0.90	1.65

Comparing Performance of 2 Models

- Given two models, say M1 and M2, which is better?
 - M1 is tested on D1 (size=n1), found error rate = e_1
 - M2 is tested on D2 (size=n2), found error rate = e_2
 - Assume D1 and D2 are independent
 - If n1 and n2 are sufficiently large, then

$$e_1 \sim N(\mu_1, \sigma_1)$$

$$e_2 \sim N(\mu_2, \sigma_2)$$

- Approximate: $\hat{\sigma}_i = \frac{e_i(1-e_i)}{n_i}$

Comparing Performance of 2 Models

- To test if performance difference is statistically significant: $d = e_1 - e_2$
 - $d \sim N(d_t, \sigma_t)$ where d_t is the true difference
 - Since D_1 and D_2 are independent, their variance adds up:

$$\begin{aligned}\sigma_t^2 &= \sigma_1^2 + \sigma_2^2 \cong \hat{\sigma}_1^2 + \hat{\sigma}_2^2 \\ &= \frac{e_1(1-e_1)}{n_1} + \frac{e_2(1-e_2)}{n_2}\end{aligned}$$

- At $(1-\alpha)$ confidence level, $d_t = d \pm Z_{\alpha/2} \hat{\sigma}_t$

An Illustrative Example

- Given: M1: $n_1 = 30, e_1 = 0.15$
M2: $n_2 = 5000, e_2 = 0.25$
- $d = |e_2 - e_1| = 0.1$ (2-sided test)

$$\hat{\sigma}_d = \sqrt{\frac{0.15(1-0.15)}{30} + \frac{0.25(1-0.25)}{5000}} = 0.0043$$

- At 95% confidence level, $Z_{\alpha/2}=1.96$

$$d_t = 0.100 \pm 1.96 \times \sqrt{0.0043} = 0.100 \pm 0.128$$

=> Interval contains 0 => difference may not be statistically significant

Comparing Performance of 2 Algorithms

- Each learning algorithm may produce k models:
 - L1 may produce M₁₁ , M₁₂, ..., M_{1k}
 - L2 may produce M₂₁ , M₂₂, ..., M_{2k}
- If models are generated on the same test sets D₁,D₂, ..., D_k (e.g., via cross-validation)
 - For each set: compute $d_j = e_{1j} - e_{2j}$
 - d_j has mean d_t and variance σ_t
 - Estimate:

$$\hat{\sigma}_t^2 = \frac{\sum_{j=1}^k (d_j - \bar{d})^2}{k(k-1)}$$

$$d_t = d \pm t_{1-\alpha, k-1} \hat{\sigma}_t$$

Data Mining Classification: Alternative Techniques

Lecture Notes for Chapter 5

Introduction to Data Mining

by

Tan, Steinbach, Kumar

Rule-Based Classifier

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

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 \mathbf{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

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

$(\text{Status}=\text{Single}) \rightarrow \text{No}$

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

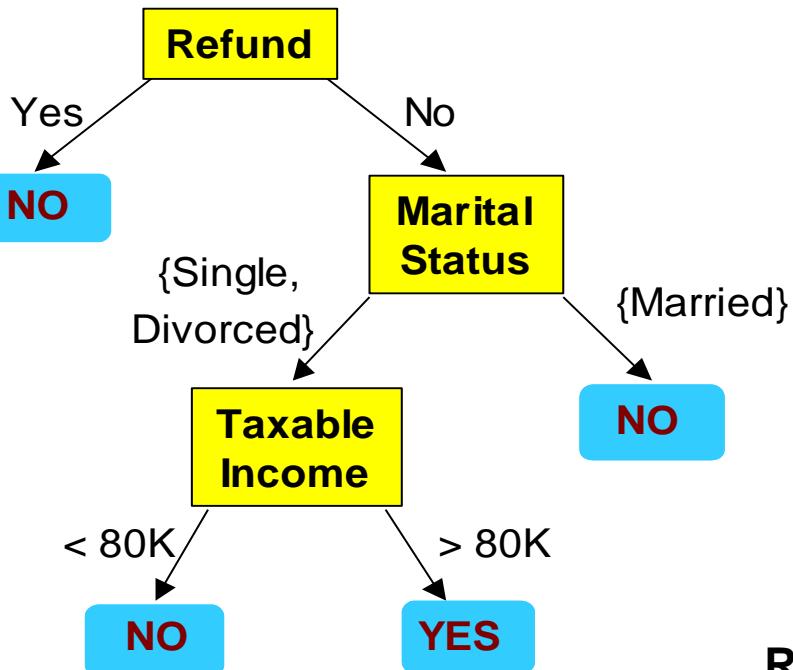
A turtle triggers both R4 and R5

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

(Refund=Yes) ==> No

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

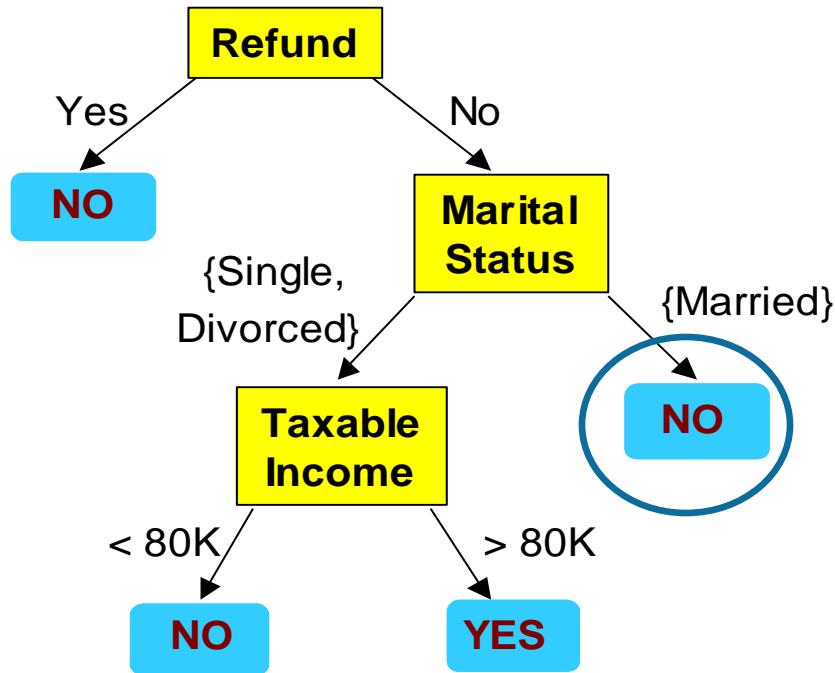
(Refund=No, Marital Status={Single,Divorced},
Taxable Income>80K) ==> Yes

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

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

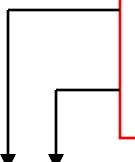
Effect of Rule Simplification

- Rules are no longer mutually exclusive
 - A record may trigger more than one rule
 - Solution?
 - ◆ Ordered rule set
 - ◆ Unordered rule set – use voting schemes
- Rules are no longer 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

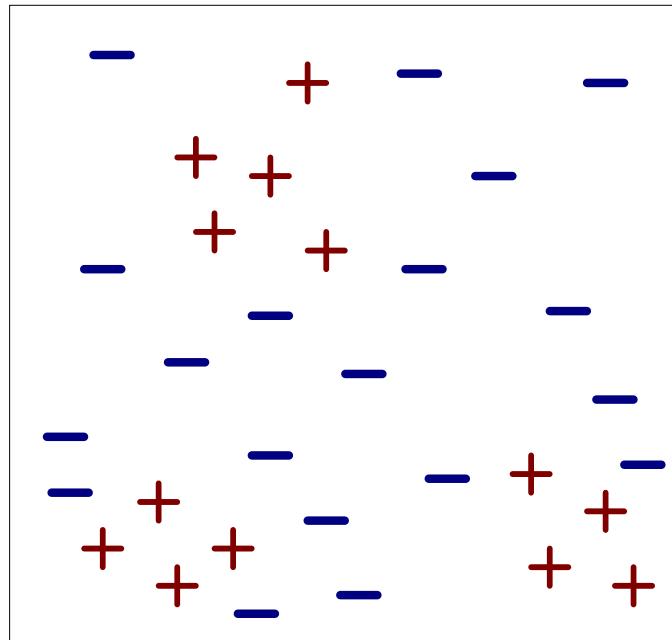
Building Classification Rules

- 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

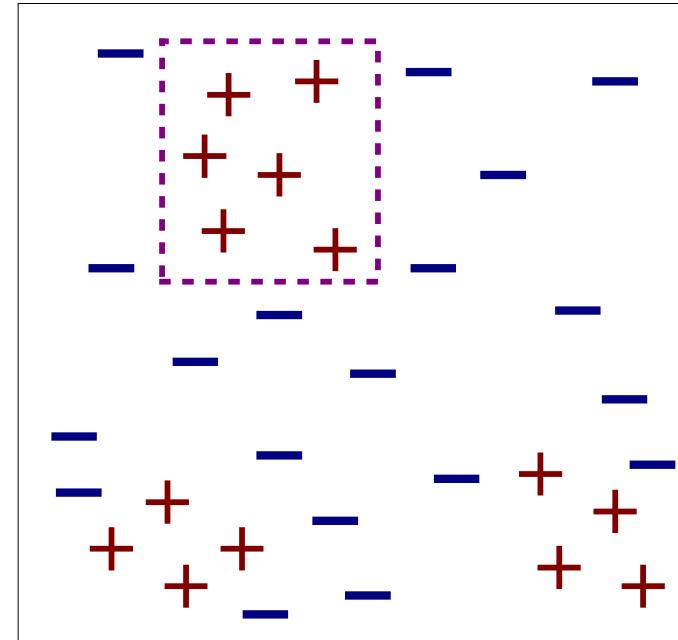
Direct Method: Sequential Covering

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

Example of Sequential Covering

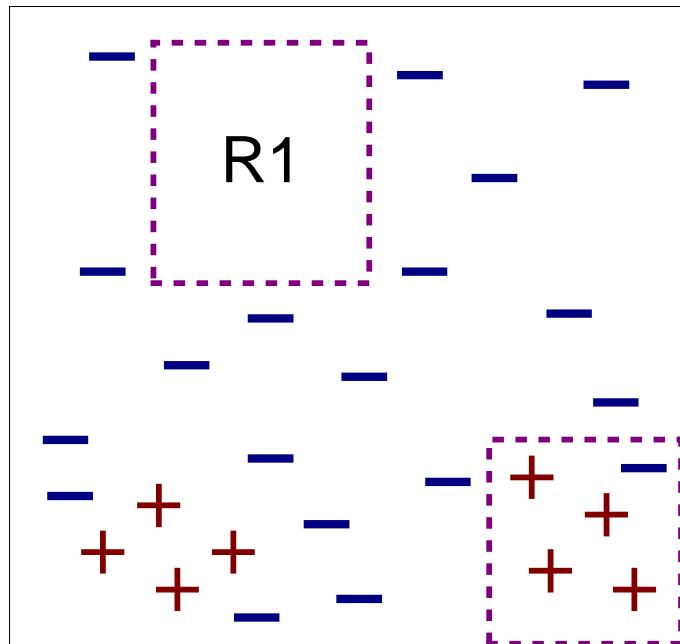


(i) Original Data

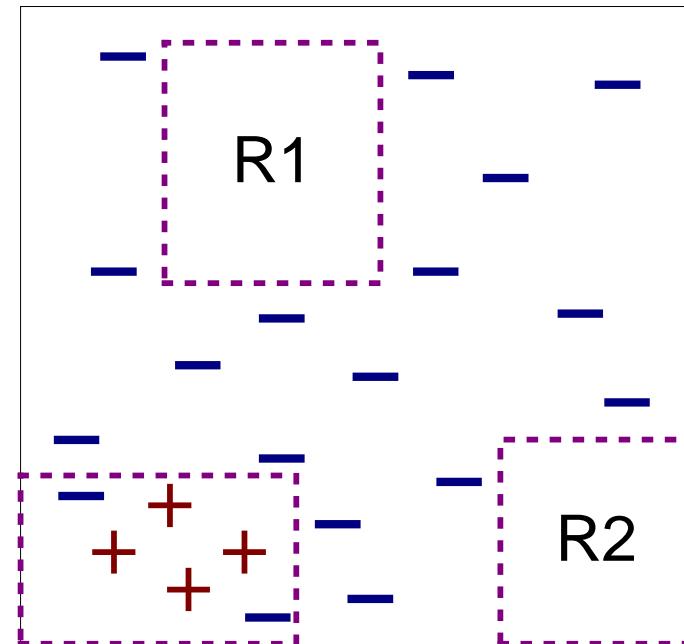


(ii) Step 1

Example of Sequential Covering...



(iii) Step 2



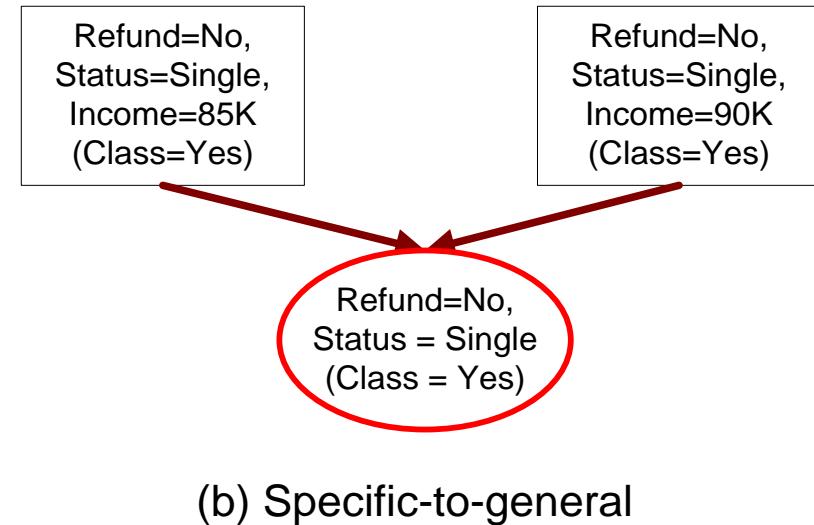
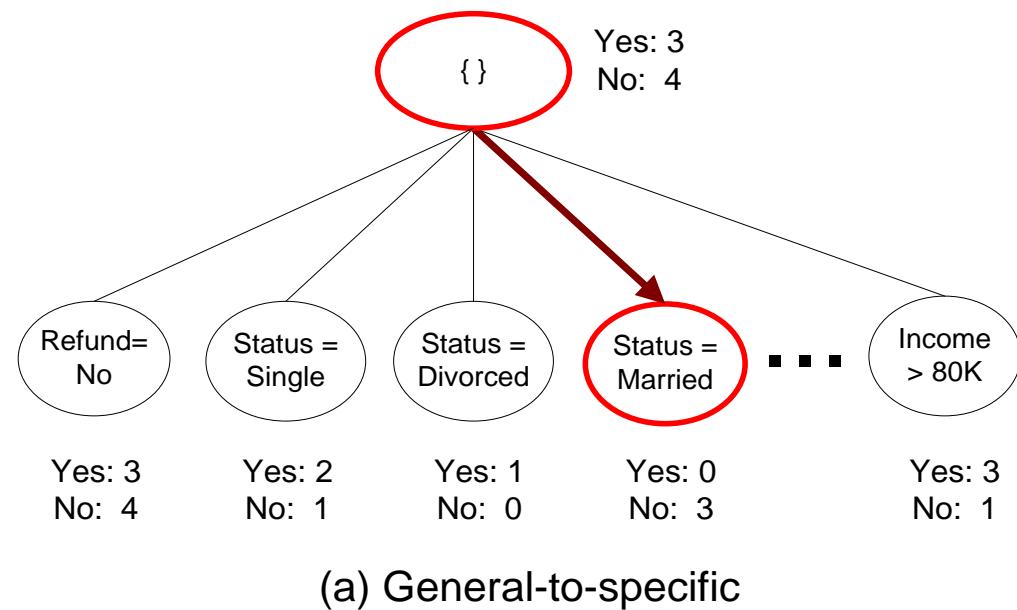
(iv) Step 3

Aspects of Sequential Covering

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

Rule Growing

- Two common strategies



Rule Growing (Examples)

- CN2 Algorithm:

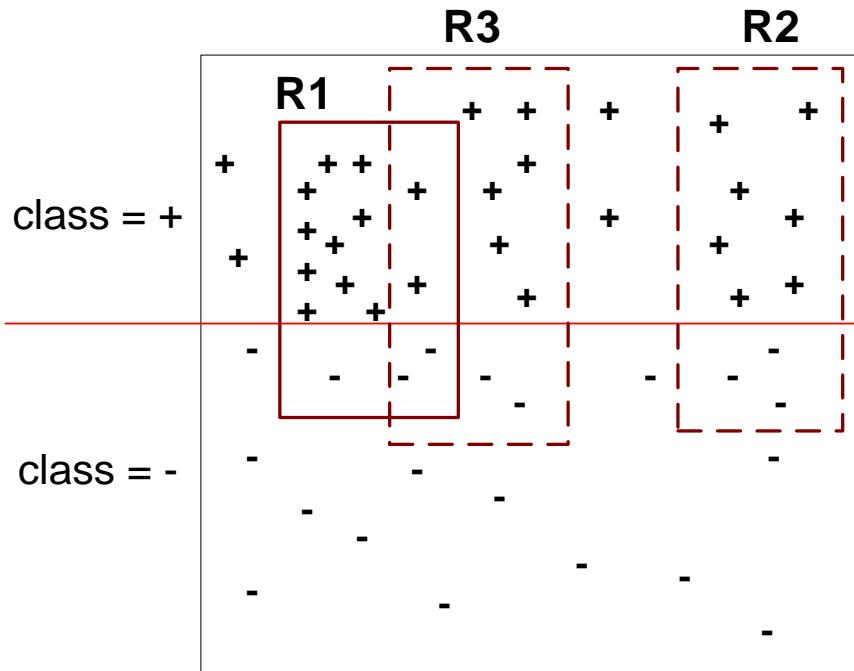
- Start from an empty conjunct: {}
- Add conjuncts that minimizes the entropy measure: {A}, {A,B}, ...
- Determine the rule consequent by taking majority class of instances covered by the rule

- RIPPER Algorithm:

- Start from an empty rule: {} => class
- Add conjuncts that maximizes FOIL's information gain measure:
 - ◆ R0: {} => class (initial rule)
 - ◆ R1: {A} => class (rule after adding conjunct)
 - ◆ $\text{Gain}(R0, R1) = t [\log(p1/(p1+n1)) - \log(p0/(p0 + n0))]$
 - ◆ where t: number of positive instances covered by both R0 and R1
p0: number of positive instances covered by R0
n0: number of negative instances covered by R0
p1: number of positive instances covered by R1
n1: number of negative instances covered by R1

Instance Elimination

- Why do we need to eliminate instances?
 - Otherwise, the next rule is identical to previous rule
- Why do we remove positive instances?
 - Ensure that the next rule is different
- Why do we remove negative instances?
 - Prevent underestimating accuracy of rule
 - Compare rules R2 and R3 in the diagram



Rule Evaluation

- Metrics:

- Accuracy = $\frac{n_c}{n}$

- Laplace = $\frac{n_c + 1}{n + k}$

- M-estimate = $\frac{n_c + kp}{n + k}$

n : Number of instances covered by rule

n_c : Number of instances covered by rule

k : Number of classes

p : Prior probability

Stopping Criterion and Rule Pruning

- Stopping criterion
 - Compute the gain
 - If gain is not significant, discard the new rule
- Rule Pruning
 - Similar to post-pruning of decision trees
 - Reduced Error Pruning:
 - ◆ Remove one of the conjuncts in the rule
 - ◆ Compare error rate on validation set before and after pruning
 - ◆ If error improves, prune the conjunct

Summary of Direct Method

- Grow a single rule
- Remove Instances from rule
- Prune the rule (if necessary)
- Add rule to Current Rule Set
- Repeat

Direct Method: RIPPER

- For 2-class problem, choose one of the classes as positive class, and the other as negative class
 - Learn rules for positive class
 - Negative class will be default class
- For multi-class problem
 - Order the classes according to increasing class prevalence (fraction of instances that belong to a particular class)
 - Learn the rule set for smallest class first, treat the rest as negative class
 - Repeat with next smallest class as positive class

Direct Method: RIPPER

- Growing a rule:
 - Start from empty rule
 - Add conjuncts as long as they improve FOIL's information gain
 - Stop when rule no longer covers negative examples
 - Prune the rule immediately using incremental reduced error pruning
 - Measure for pruning: $v = (p-n)/(p+n)$
 - ◆ p: number of positive examples covered by the rule in the validation set
 - ◆ n: number of negative examples covered by the rule in the validation set
 - Pruning method: delete any final sequence of conditions that maximizes v

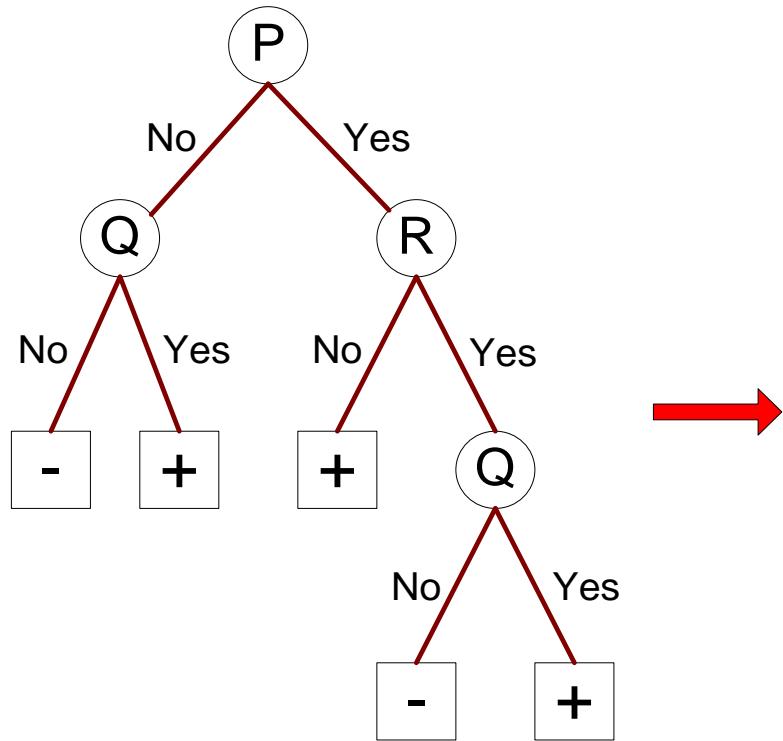
Direct Method: RIPPER

- Building a Rule Set:
 - Use sequential covering algorithm
 - ◆ Finds the best rule that covers the current set of positive examples
 - ◆ Eliminate both positive and negative examples covered by the rule
 - Each time a rule is added to the rule set, compute the new description length
 - ◆ stop adding new rules when the new description length is d bits longer than the smallest description length obtained so far

Direct Method: RIPPER

- Optimize the rule set:
 - For each rule r in the rule set R
 - ◆ Consider 2 alternative rules:
 - Replacement rule (r^*): grow new rule from scratch
 - Revised rule(r'): add conjuncts to extend the rule r
 - ◆ Compare the rule set for r against the rule set for r^* and r'
 - ◆ Choose rule set that minimizes MDL principle
 - Repeat rule generation and rule optimization for the remaining positive examples

Indirect Methods



Rule Set

- r1: (P=No,Q=No) ==> -
- r2: (P=No,Q=Yes) ==> +
- r3: (P=Yes,R=No) ==> +
- r4: (P=Yes,R=Yes,Q=No) ==> -
- r5: (P=Yes,R=Yes,Q=Yes) ==> +

Indirect Method: C4.5rules

- Extract rules from an unpruned decision tree
- For each rule, $r: A \rightarrow y$,
 - consider an alternative rule $r': A' \rightarrow y$ where A' is obtained by removing one of the conjuncts in A
 - Compare the pessimistic error rate for r against all r 's
 - Prune if one of the r 's has lower pessimistic error rate
 - Repeat until we can no longer improve generalization error

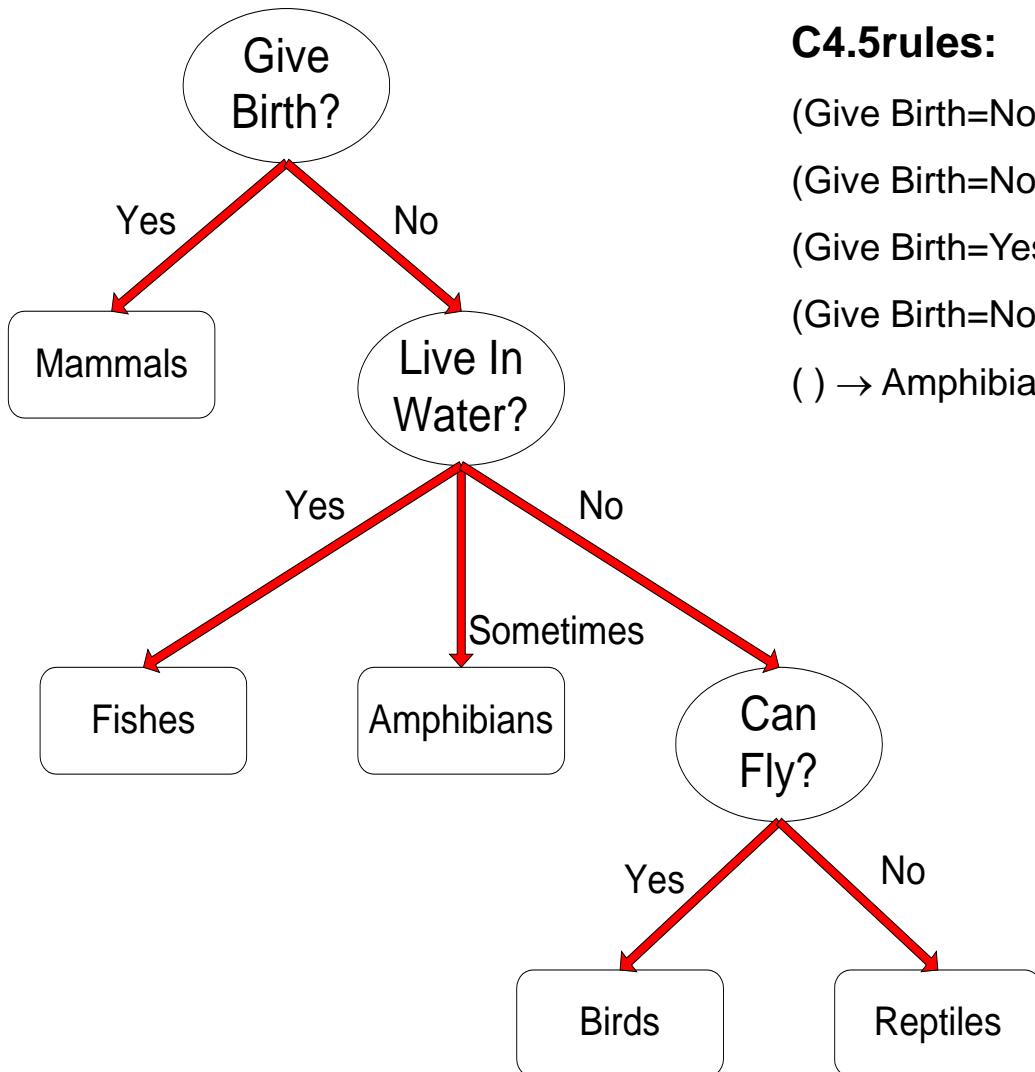
Indirect Method: C4.5rules

- Instead of ordering the rules, order subsets of rules (**class ordering**)
 - Each subset is a collection of rules with the same rule consequent (class)
 - Compute description length of each subset
 - ◆ Description length = $L(\text{error}) + g L(\text{model})$
 - ◆ g is a parameter that takes into account the presence of redundant attributes in a rule set (default value = 0.5)

Example

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

C4.5 versus C4.5rules versus RIPPER



C4.5rules:

- (Give Birth=No, Can Fly=Yes) → Birds
- (Give Birth=No, Live in Water=Yes) → Fishes
- (Give Birth=Yes) → Mammals
- (Give Birth=No, Can Fly>No, Live in Water>No) → Reptiles
- () → Amphibians

RIPPER:

- (Live in Water=Yes) → Fishes
- (Have Legs>No) → Reptiles
- (Give Birth=No, Can Fly>No, Live In Water>No)
→ Reptiles
- (Can Fly=Yes, Give Birth=No) → Birds
- () → Mammals

C4.5 versus C4.5rules versus RIPPER

C4.5 and C4.5rules:

		PREDICTED CLASS				
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	2	0	0	0	0
	Fishes	0	2	0	0	1
	Reptiles	1	0	3	0	0
	Birds	1	0	0	3	0
	Mammals	0	0	1	0	6

RIPPER:

		PREDICTED CLASS				
		Amphibians	Fishes	Reptiles	Birds	Mammals
ACTUAL CLASS	Amphibians	0	0	0	0	2
	Fishes	0	3	0	0	0
	Reptiles	0	0	3	0	1
	Birds	0	0	1	2	1
	Mammals	0	2	1	0	4

Advantages of Rule-Based Classifiers

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

Instance-Based Classifiers

Set of Stored Cases

Atr1	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

- Store the training records
- Use training records to predict the class label of unseen cases

Unseen Case

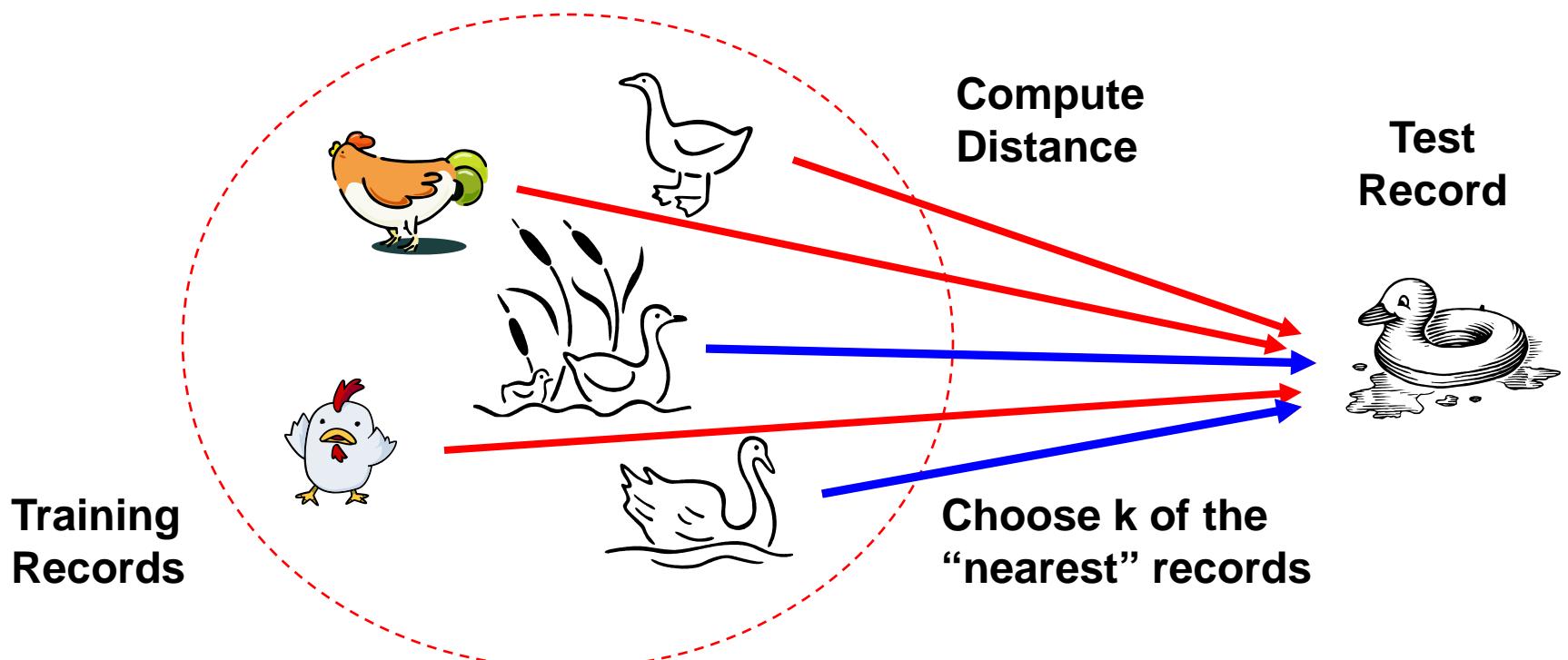
Atr1	AtrN

Instance Based Classifiers

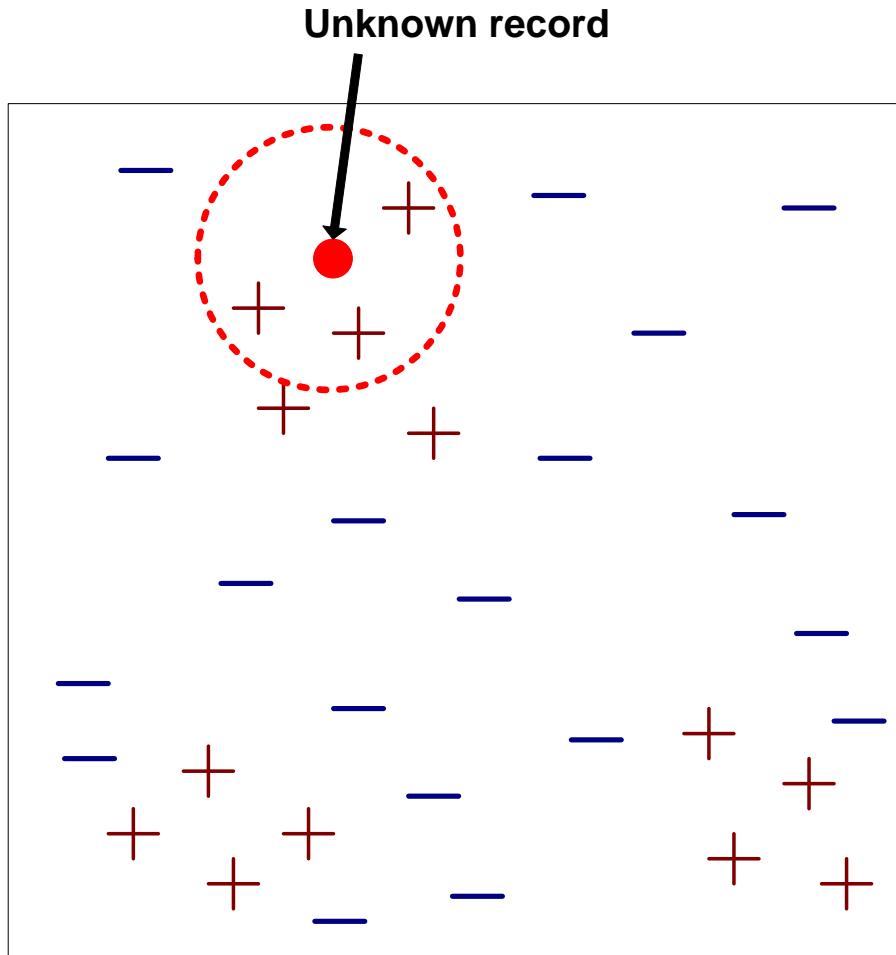
- Examples:
 - Rote-learner
 - ◆ Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
 - Nearest neighbor
 - ◆ Uses k “closest” points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

- Basic idea:
 - If it walks like a duck, quacks like a duck, then it's probably a duck

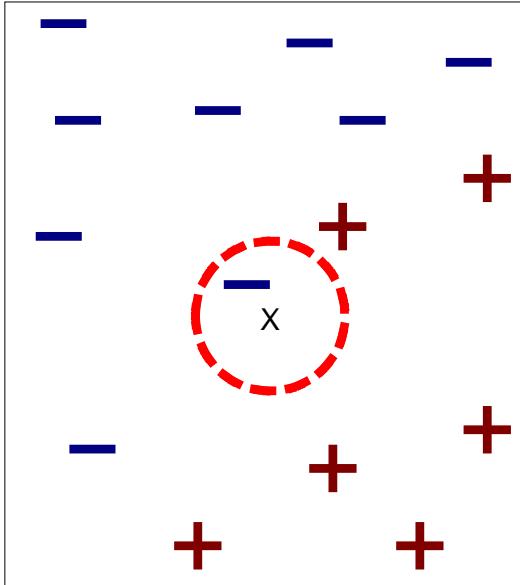


Nearest-Neighbor Classifiers

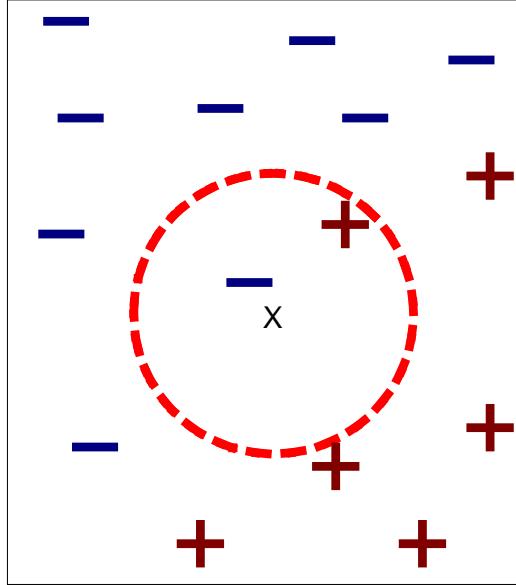


- Requires three things
 - The set of stored records
 - Distance Metric to compute distance between records
 - The value of k , the number of nearest neighbors to retrieve
- To classify an unknown record:
 - Compute distance to other training records
 - Identify k nearest neighbors
 - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

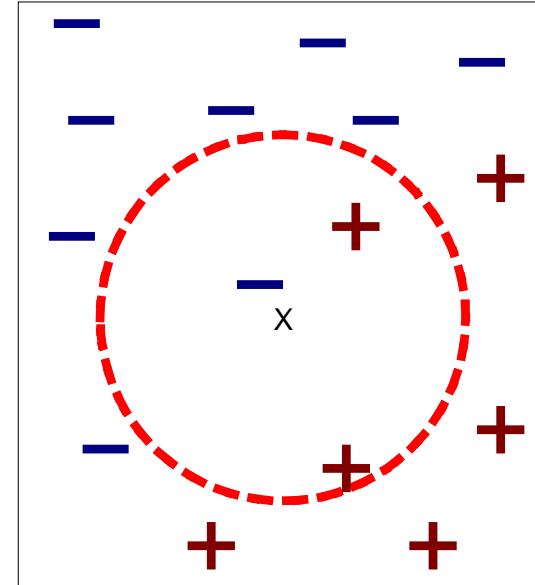
Definition of Nearest Neighbor



(a) 1-nearest neighbor



(b) 2-nearest neighbor

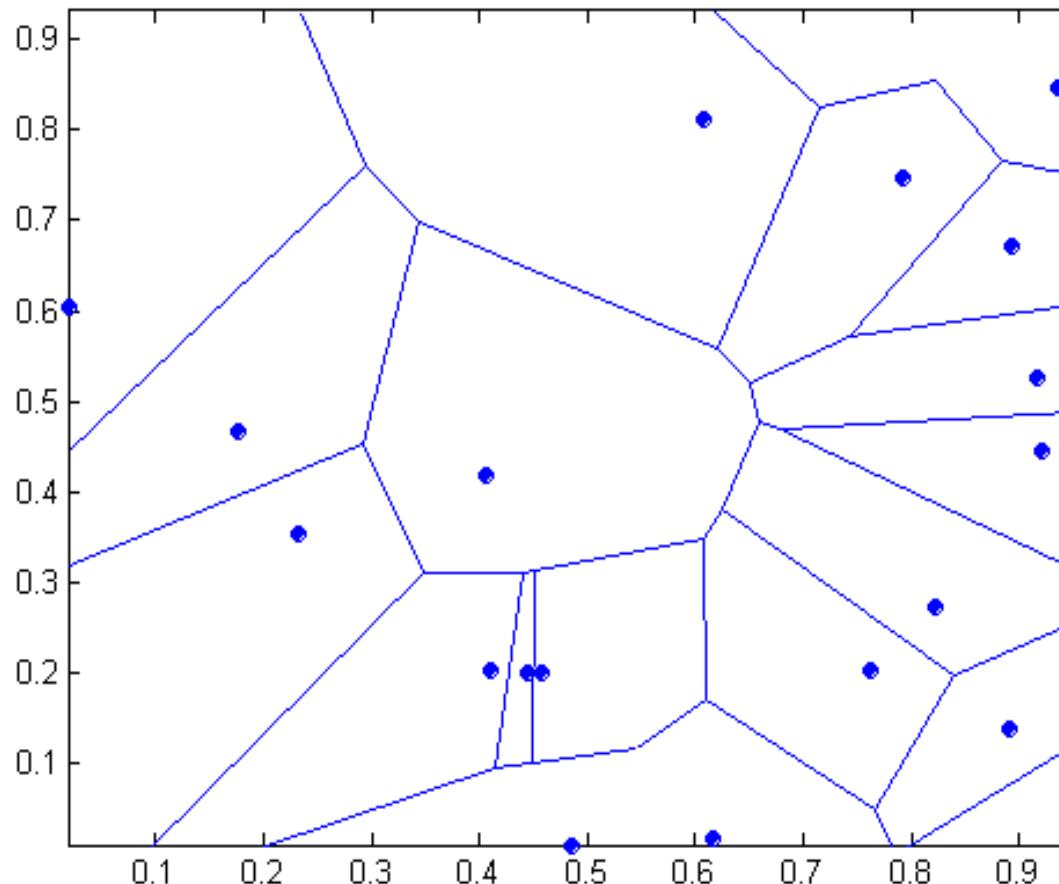


(c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

1 nearest-neighbor

Voronoi Diagram



Nearest Neighbor Classification

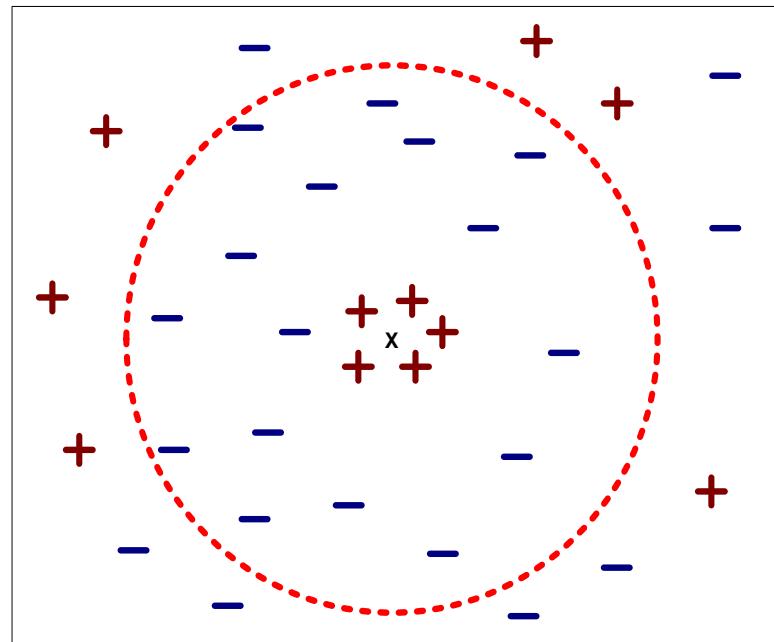
- Compute distance between two points:
 - Euclidean distance

$$d(p, q) = \sqrt{\sum_i (p_i - q_i)^2}$$

- Determine the class from nearest neighbor list
 - take the majority vote of class labels among the k-nearest neighbors
 - Weigh the vote according to distance
 - ◆ weight factor, $w = 1/d^2$

Nearest Neighbor Classification...

- Choosing the value of k :
 - If k is too small, sensitive to noise points
 - If k is too large, neighborhood may include points from other classes



Nearest Neighbor Classification...

- Scaling issues
 - Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
 - Example:
 - ◆ height of a person may vary from 1.5m to 1.8m
 - ◆ weight of a person may vary from 90lb to 300lb
 - ◆ income of a person may vary from \$10K to \$1M

Nearest Neighbor Classification...

- Problem with Euclidean measure:
 - High dimensional data
 - ◆ curse of dimensionality
 - Can produce counter-intuitive results

1 1 1 1 1 1 1 1 1 1 0

0 1 1 1 1 1 1 1 1 1 1

vs

1 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 1

$d = 1.4142$

$d = 1.4142$

- ◆ Solution: Normalize the vectors to unit length

Nearest neighbor Classification...

- k-NN classifiers are lazy learners
 - It does not build models explicitly
 - Unlike eager learners such as decision tree induction and rule-based systems
 - Classifying unknown records are relatively expensive

Example: PEBLS

- PEBLS: Parallel Exemplar-Based Learning System (Cost & Salzberg)
 - Works with both continuous and nominal features
 - ◆ For nominal features, distance between two nominal values is computed using modified value difference metric (MVDM)
 - Each record is assigned a weight factor
 - Number of nearest neighbor, $k = 1$

Example: PEBLS

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

Distance between nominal attribute values:

$$d(\text{Single}, \text{Married})$$

$$= | 2/4 - 0/4 | + | 2/4 - 4/4 | = 1$$

$$d(\text{Single}, \text{Divorced})$$

$$= | 2/4 - 1/2 | + | 2/4 - 1/2 | = 0$$

$$d(\text{Married}, \text{Divorced})$$

$$= | 0/4 - 1/2 | + | 4/4 - 1/2 | = 1$$

$$d(\text{Refund}=\text{Yes}, \text{Refund}=\text{No})$$

$$= | 0/3 - 3/7 | + | 3/3 - 4/7 | = 6/7$$

Class	Marital Status		
	Single	Married	Divorced
Yes	2	0	1
No	2	4	1

Class	Refund	
	Yes	No
Yes	0	3
No	3	4

$$d(V_1, V_2) = \sum_i \left| \frac{n_{1i}}{n_1} - \frac{n_{2i}}{n_2} \right|$$

Example: PEBLS

Tid	Refund	Marital Status	Taxable Income	Cheat
X	Yes	Single	125K	No
Y	No	Married	100K	No

Distance between record X and record Y:

$$\Delta(X, Y) = w_X w_Y \sum_{i=1}^d d(X_i, Y_i)^2$$

where:

$$w_X = \frac{\text{Number of times X is used for prediction}}{\text{Number of times X predicts correctly}}$$

$w_X \approx 1$ if X makes accurate prediction most of the time

$w_X > 1$ if X is not reliable for making predictions

Bayes Classifier

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

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

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

- Bayes theorem:

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

Example of Bayes Theorem

- Given:
 - A doctor knows that meningitis causes stiff neck 50% of the time
 - Prior probability of any patient having meningitis is 1/50,000
 - Prior probability of any patient having stiff neck is 1/20
- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M | S) = \frac{P(S | M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Bayesian Classifiers

- Consider each attribute and class label as random variables
- Given a record with attributes (A_1, A_2, \dots, A_n)
 - Goal is to predict class C
 - Specifically, we want to find the value of C that maximizes $P(C | A_1, A_2, \dots, A_n)$
- Can we estimate $P(C | A_1, A_2, \dots, A_n)$ directly from data?

Bayesian Classifiers

- Approach:
 - compute the posterior probability $P(C | A_1, A_2, \dots, A_n)$ for all values of C using the Bayes theorem

$$P(C | A_1 A_2 \dots A_n) = \frac{P(A_1 A_2 \dots A_n | C) P(C)}{P(A_1 A_2 \dots A_n)}$$

 - Choose value of C that maximizes $P(C | A_1, A_2, \dots, A_n)$
 - Equivalent to choosing value of C that maximizes $P(A_1, A_2, \dots, A_n | C) P(C)$
- How to estimate $P(A_1, A_2, \dots, A_n | C)$?

Naïve Bayes Classifier

- Assume independence among attributes A_i when class is given:
 - $P(A_1, A_2, \dots, A_n | C) = P(A_1| C_j) P(A_2| C_j) \dots P(A_n| C_j)$
 - Can estimate $P(A_i| C_j)$ for all A_i and C_j .
 - New point is classified to C_j if $P(C_j) \prod P(A_i| C_j)$ is maximal.

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
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

- Class: $P(C) = N_c/N$

- e.g., $P(\text{No}) = 7/10$,
 $P(\text{Yes}) = 3/10$

- For discrete attributes:

$$P(A_i | C_k) = |A_{ik}| / N_{C_k}$$

- where $|A_{ik}|$ is number of instances having attribute A_i and belongs to class C_k

- Examples:

$$\begin{aligned}P(\text{Status=Married|No}) &= 4/7 \\ P(\text{Refund=Yes|Yes}) &= 0\end{aligned}$$

How to Estimate Probabilities from Data?

- For continuous attributes:
 - Discretize the range into bins
 - ◆ one ordinal attribute per bin
 - ◆ violates independence assumption $\leftarrow k$
 - Two-way split: $(A < v)$ or $(A > v)$
 - ◆ choose only one of the two splits as new attribute
 - Probability density estimation:
 - ◆ Assume attribute follows a normal distribution
 - ◆ Use data to estimate parameters of distribution (e.g., mean and standard deviation)
 - ◆ Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$

How to Estimate Probabilities from Data?

Tid	Refund	Marital Status	Taxable Income	Evade
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

- Normal distribution:

$$P(A_i | c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i - \mu_{ij})^2}{2\sigma_{ij}^2}}$$

- One for each (A_i, c_i) pair

- For (Income, Class=No):

- If Class=No

- sample mean = 110

- sample variance = 2975

$$P(\text{Income} = 120 | \text{No}) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Example of Naïve Bayes Classifier

Given a Test Record:

$$X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120\text{K})$$

naive Bayes Classifier:

$$P(\text{Refund}=\text{Yes}|\text{No}) = 3/7$$

$$P(\text{Refund}=\text{No}|\text{No}) = 4/7$$

$$P(\text{Refund}=\text{Yes}|\text{Yes}) = 0$$

$$P(\text{Refund}=\text{No}|\text{Yes}) = 1$$

$$P(\text{Marital Status}=\text{Single}|\text{No}) = 2/7$$

$$P(\text{Marital Status}=\text{Divorced}|\text{No}) = 1/7$$

$$P(\text{Marital Status}=\text{Married}|\text{No}) = 4/7$$

$$P(\text{Marital Status}=\text{Single}|\text{Yes}) = 2/7$$

$$P(\text{Marital Status}=\text{Divorced}|\text{Yes}) = 1/7$$

$$P(\text{Marital Status}=\text{Married}|\text{Yes}) = 0$$

For taxable income:

If class=No: sample mean=110
sample variance=2975

If class=Yes: sample mean=90
sample variance=25

- $P(X|\text{Class}=\text{No}) = P(\text{Refund}=\text{No}|\text{Class}=\text{No}) \times P(\text{Married}|\text{ Class}=\text{No}) \times P(\text{Income}=120\text{K}|\text{ Class}=\text{No}) = 4/7 \times 4/7 \times 0.0072 = 0.0024$
- $P(X|\text{Class}=\text{Yes}) = P(\text{Refund}=\text{No}|\text{ Class}=\text{Yes}) \times P(\text{Married}|\text{ Class}=\text{Yes}) \times P(\text{Income}=120\text{K}|\text{ Class}=\text{Yes}) = 1 \times 0 \times 1.2 \times 10^{-9} = 0$

Since $P(X|\text{No})P(\text{No}) > P(X|\text{Yes})P(\text{Yes})$

Therefore $P(\text{No}|X) > P(\text{Yes}|X)$
 $\Rightarrow \text{Class} = \text{No}$

Naïve Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

$$\text{Original : } P(A_i | C) = \frac{N_{ic}}{N_c}$$

c: number of classes

$$\text{Laplace : } P(A_i | C) = \frac{N_{ic} + 1}{N_c + c}$$

p: prior probability

$$\text{m - estimate : } P(A_i | C) = \frac{N_{ic} + mp}{N_c + m}$$

m: parameter

Example of Naïve Bayes Classifier

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

A: attributes

M: mammals

N: non-mammals

$$P(A | M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A | N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A | M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A | N)P(N) = 0.004 \times \frac{13}{20} = 0.0027$$

$P(A|M)P(M) > P(A|N)P(N)$
=> Mammals

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

Naïve Bayes (Summary)

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
 - Use other techniques such as Bayesian Belief Networks (BBN)

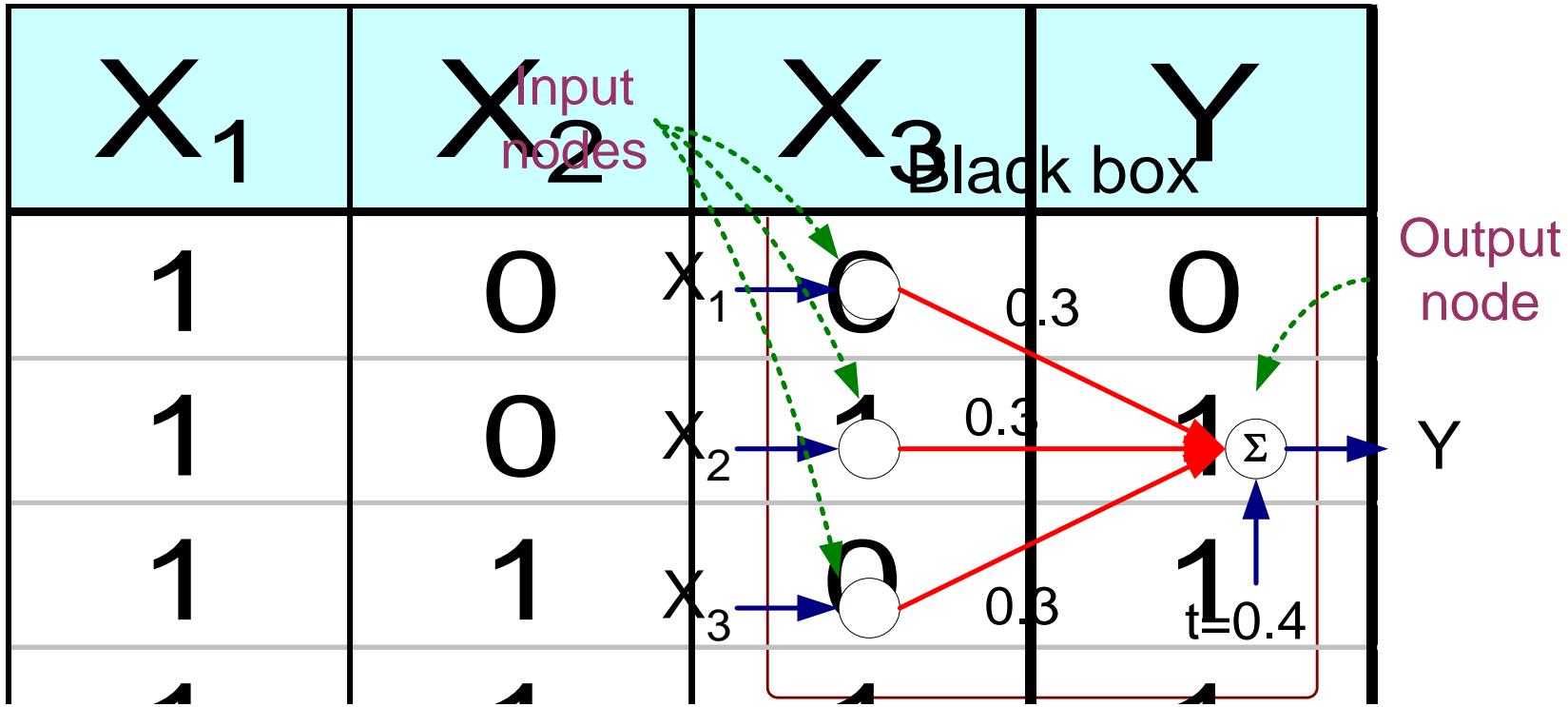
Artificial Neural Networks (ANN)

X_1	X_2	X_3	Y
Black box			
1	0	$x_1 \rightarrow 0$	0
1	0	$x_2 \rightarrow 1$	1
1	1	$x_3 \rightarrow 0$	1

Diagram illustrating the operation of a black box function. The inputs x_1, x_2, x_3 are processed by the black box to produce the output Y. The output Y is labeled as 'Output'.

Output Y is 1 if at least two of the three inputs are equal to 1.

Artificial Neural Networks (ANN)

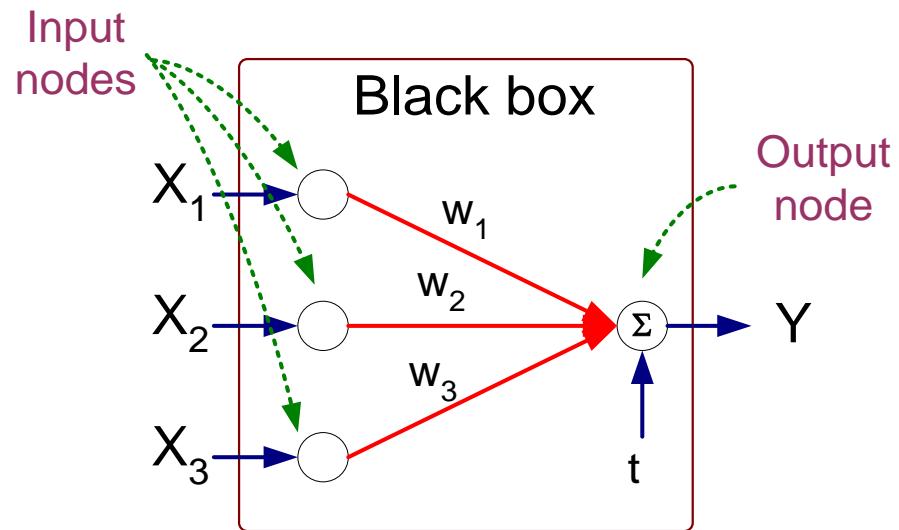


$$Y = I(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0)$$

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

Artificial Neural Networks (ANN)

- Model is an assembly of inter-connected nodes and weighted links
- Output node sums up each of its input value according to the weights of its links
- Compare output node against some threshold t

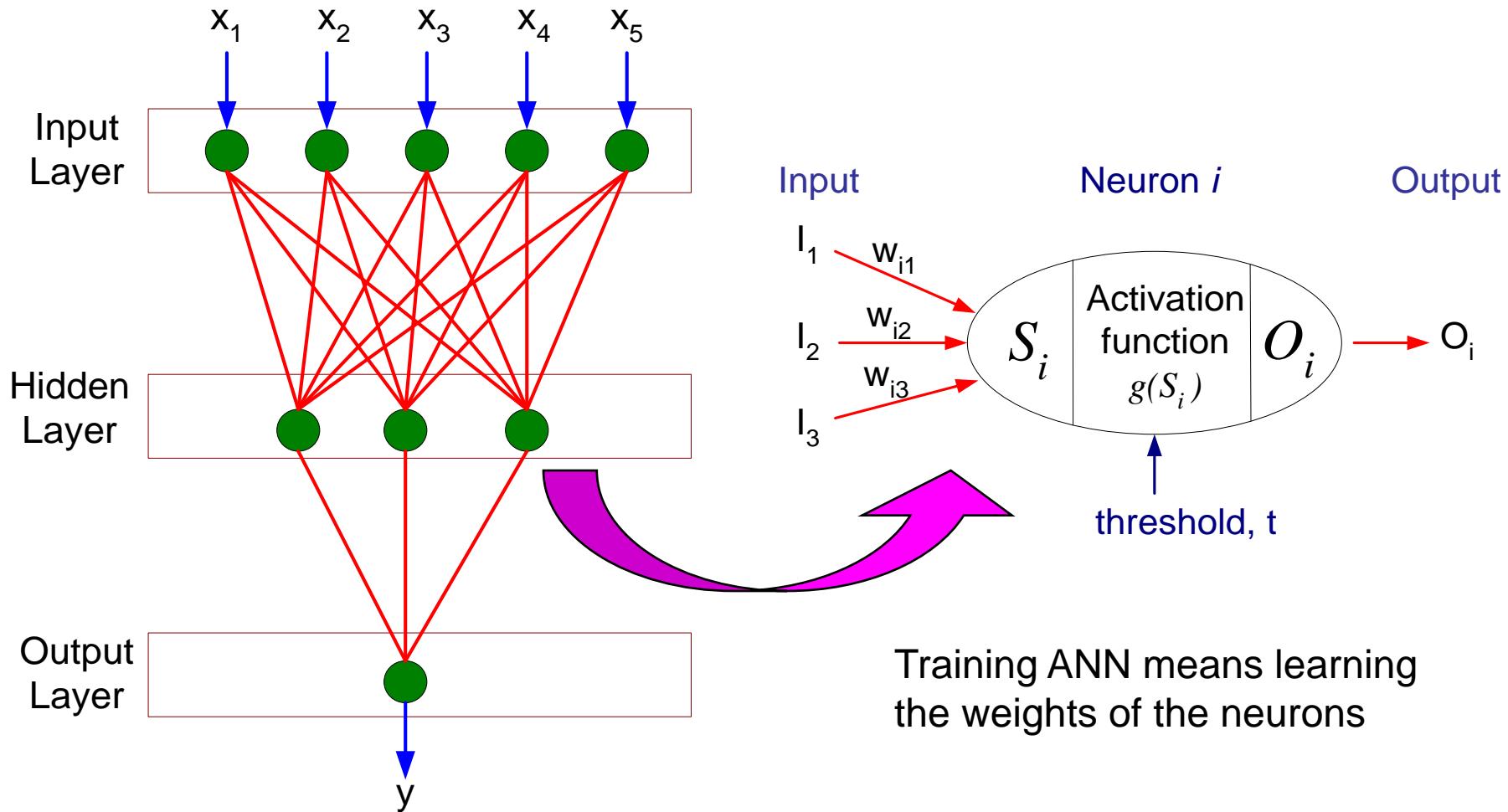


Perceptron Model

$$Y = I(\sum_i w_i X_i - t) \quad \text{or}$$

$$Y = sign(\sum_i w_i X_i - t)$$

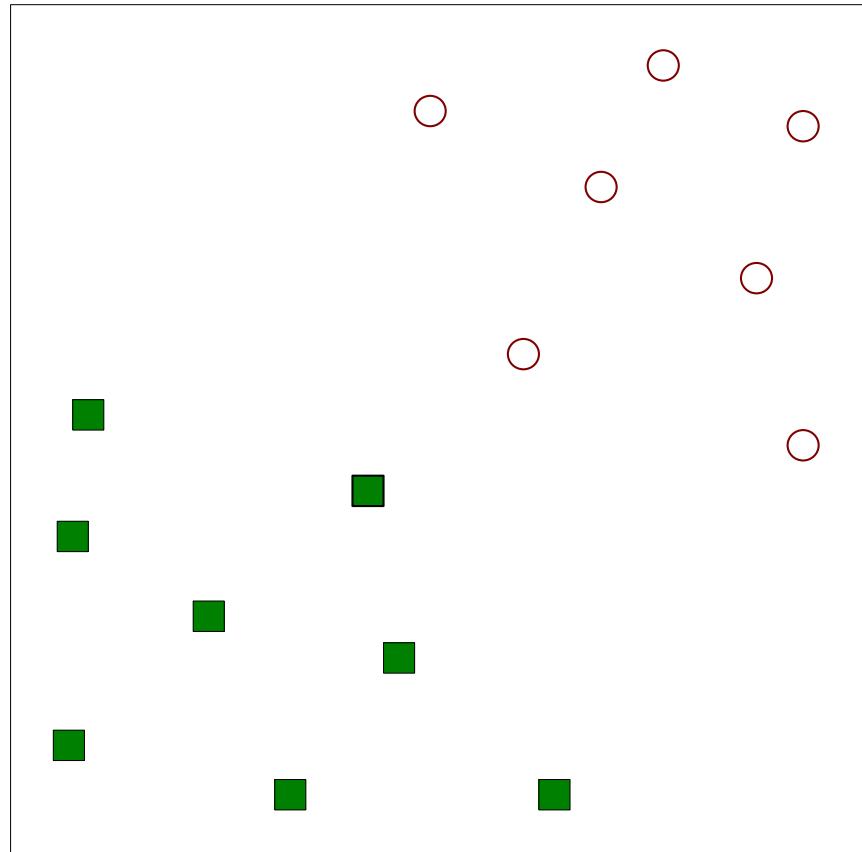
General Structure of ANN



Algorithm for learning ANN

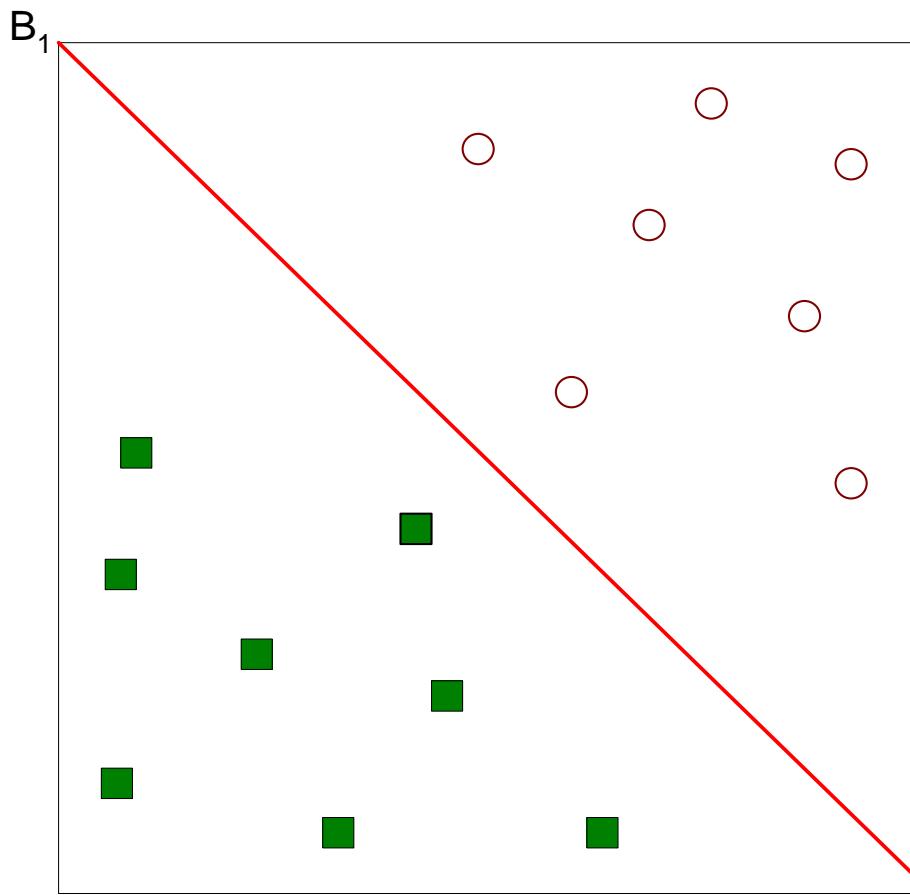
- Initialize the weights (w_0, w_1, \dots, w_k)
- Adjust the weights in such a way that the output of ANN is consistent with class labels of training examples
 - Objective function: $E = \sum_i [Y_i - f(w_i, X_i)]^2$
 - Find the weights w_i 's that minimize the above objective function
 - ◆ e.g., backpropagation algorithm (see lecture notes)

Support Vector Machines



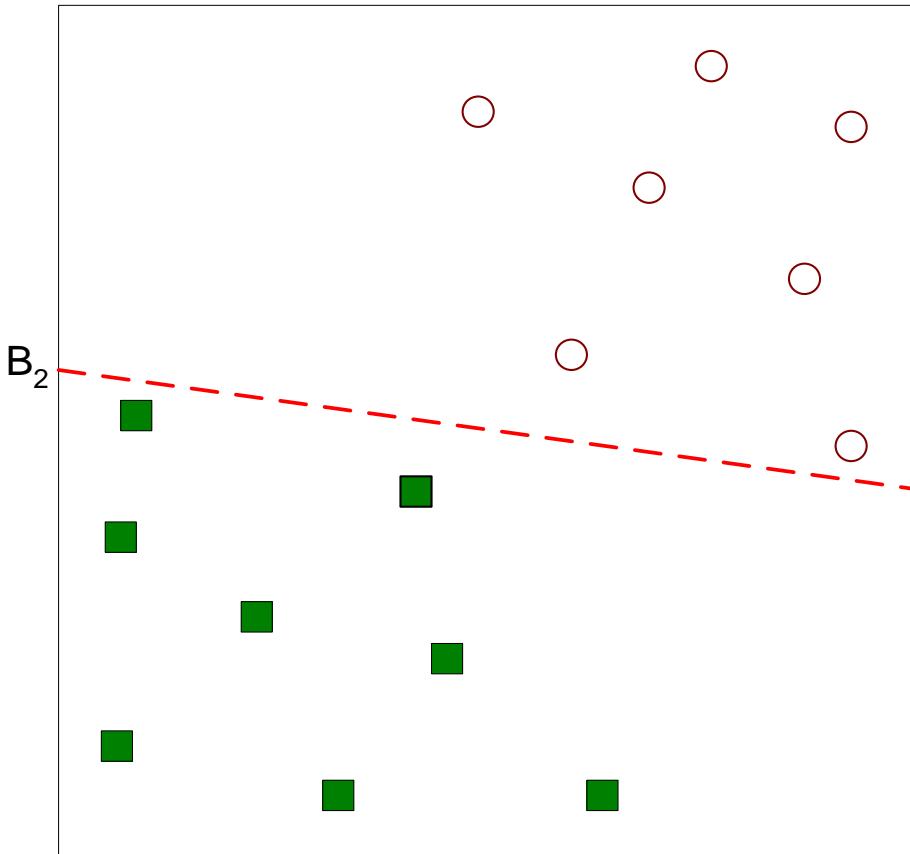
- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



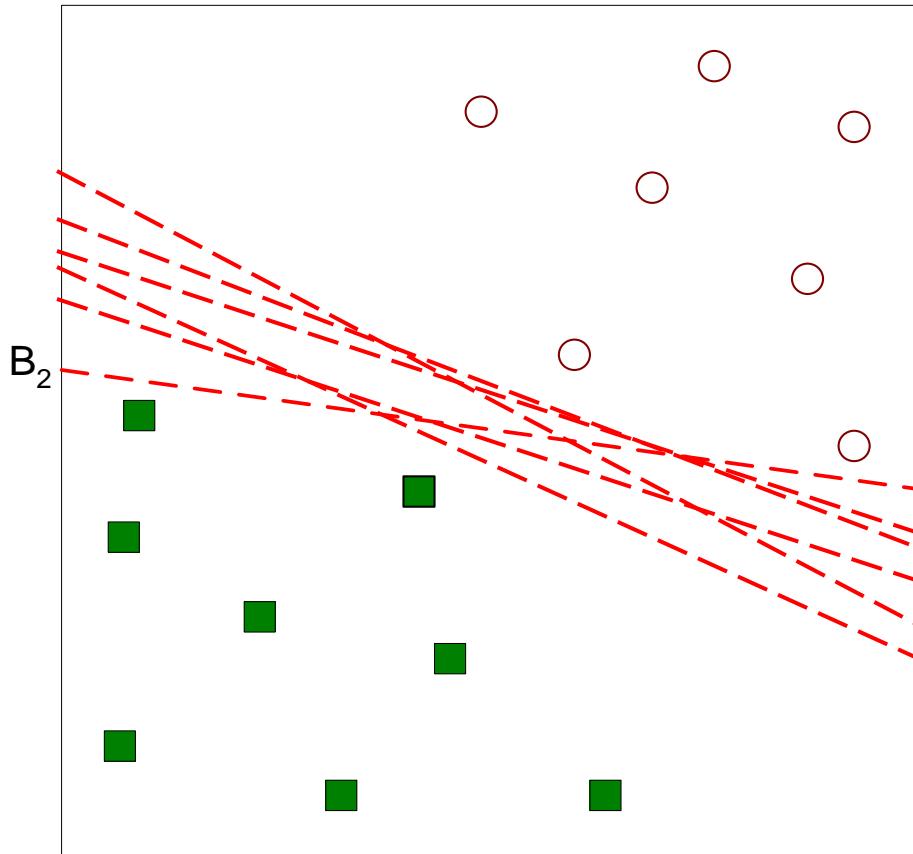
- One Possible Solution

Support Vector Machines



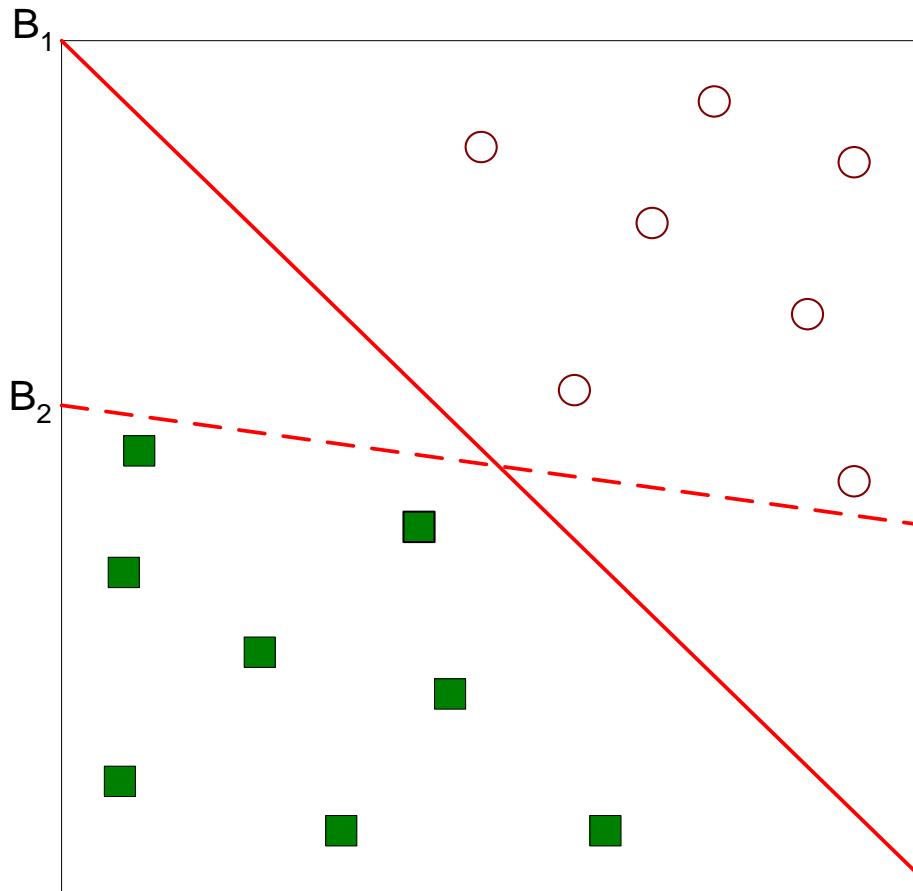
- Another possible solution

Support Vector Machines



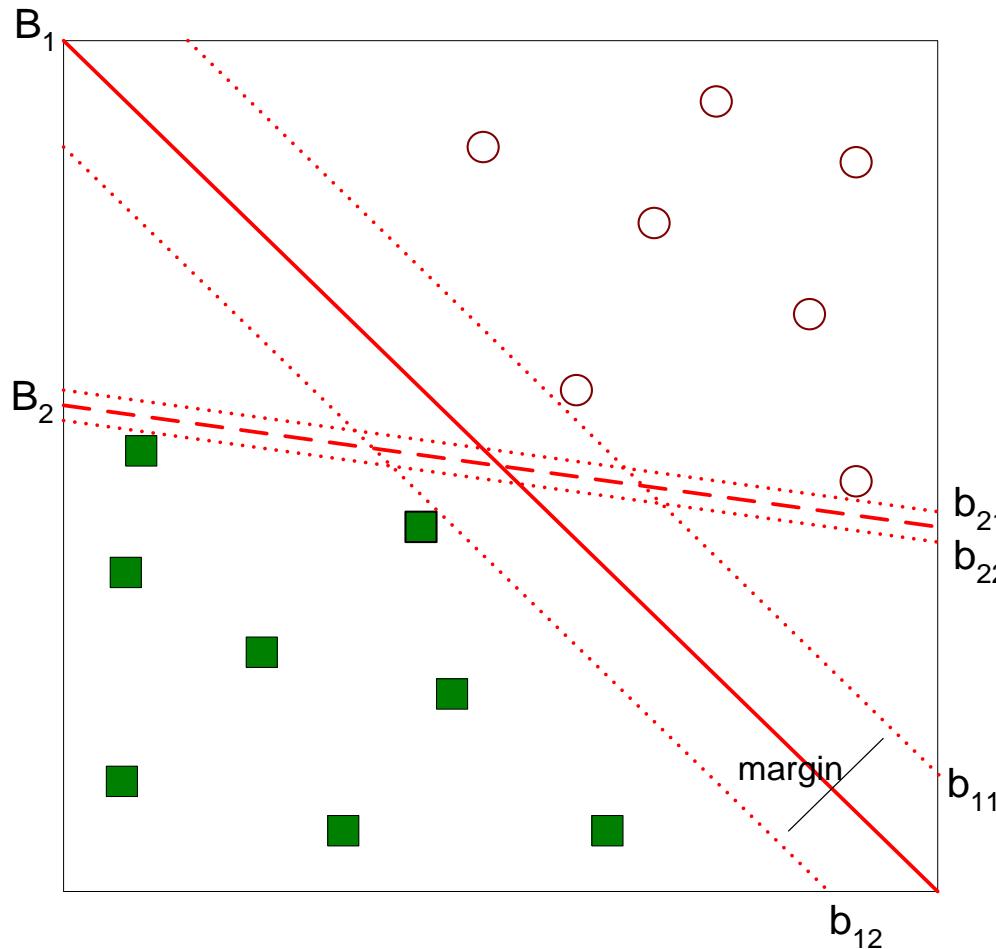
- Other possible solutions

Support Vector Machines



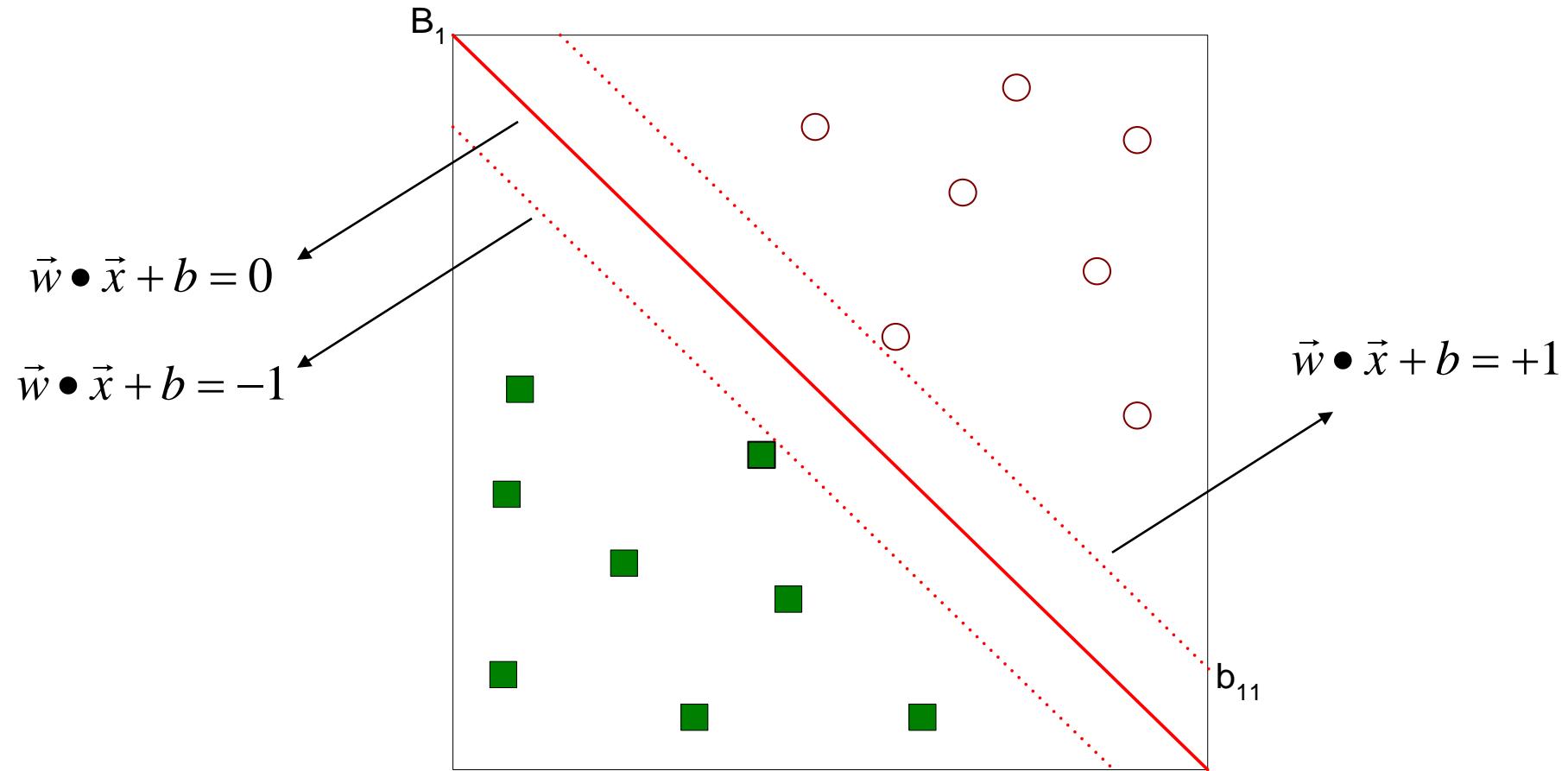
- Which one is better? B_1 or B_2 ?
- How do you define better?

Support Vector Machines



- Find hyperplane **maximizes** the margin => B1 is better than B2

Support Vector Machines



$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

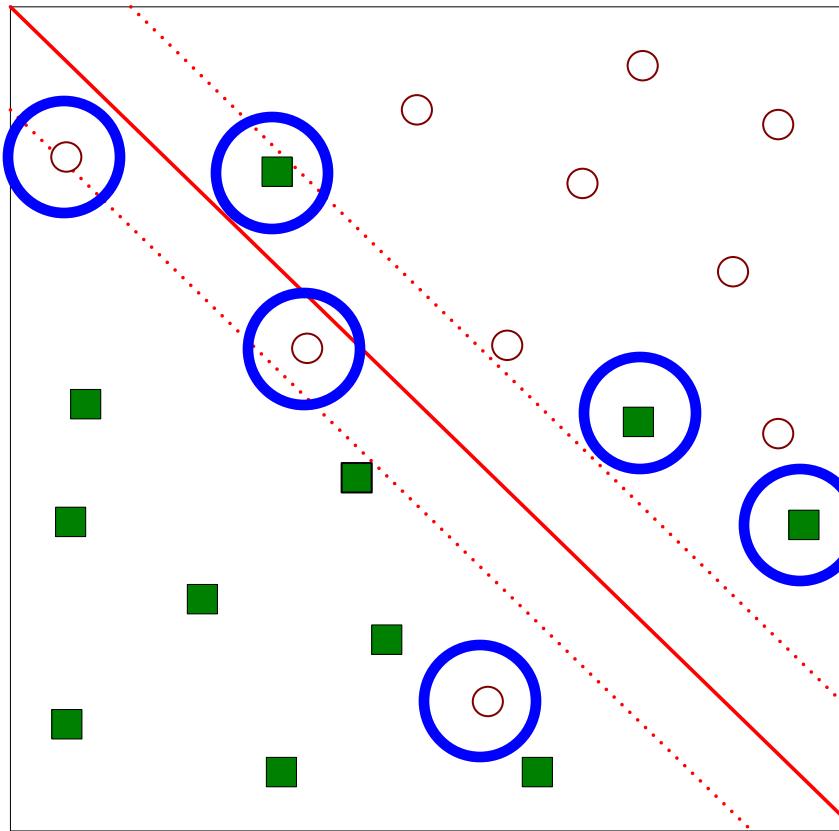
$$\text{Margin} = \frac{2}{\|\vec{w}\|^2}$$

Support Vector Machines

- We want to maximize: Margin = $\frac{2}{\|\vec{w}\|^2}$
 - Which is equivalent to minimizing: $L(w) = \frac{\|\vec{w}\|^2}{2}$
 - But subjected to the following constraints:
$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 \end{cases}$$
- ◆ This is a constrained optimization problem
 - Numerical approaches to solve it (e.g., quadratic programming)

Support Vector Machines

- What if the problem is not linearly separable?



Support Vector Machines

- What if the problem is not linearly separable?

- Introduce slack variables

- ◆ Need to minimize:

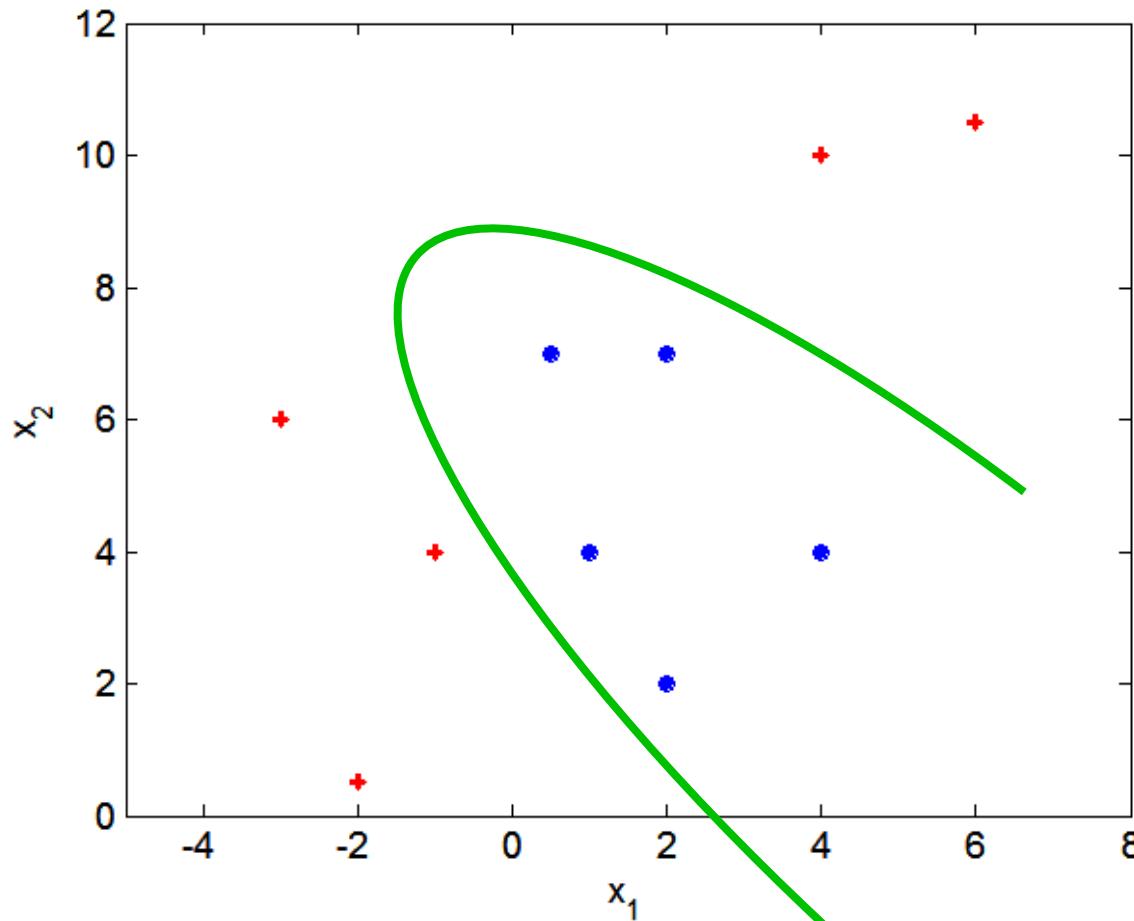
$$L(w) = \frac{\|\vec{w}\|^2}{2} + C \left(\sum_{i=1}^N \xi_i^k \right)$$

- ◆ Subject to:

$$f(\vec{x}_i) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x}_i + b \geq 1 - \xi_i \\ -1 & \text{if } \vec{w} \bullet \vec{x}_i + b \leq -1 + \xi_i \end{cases}$$

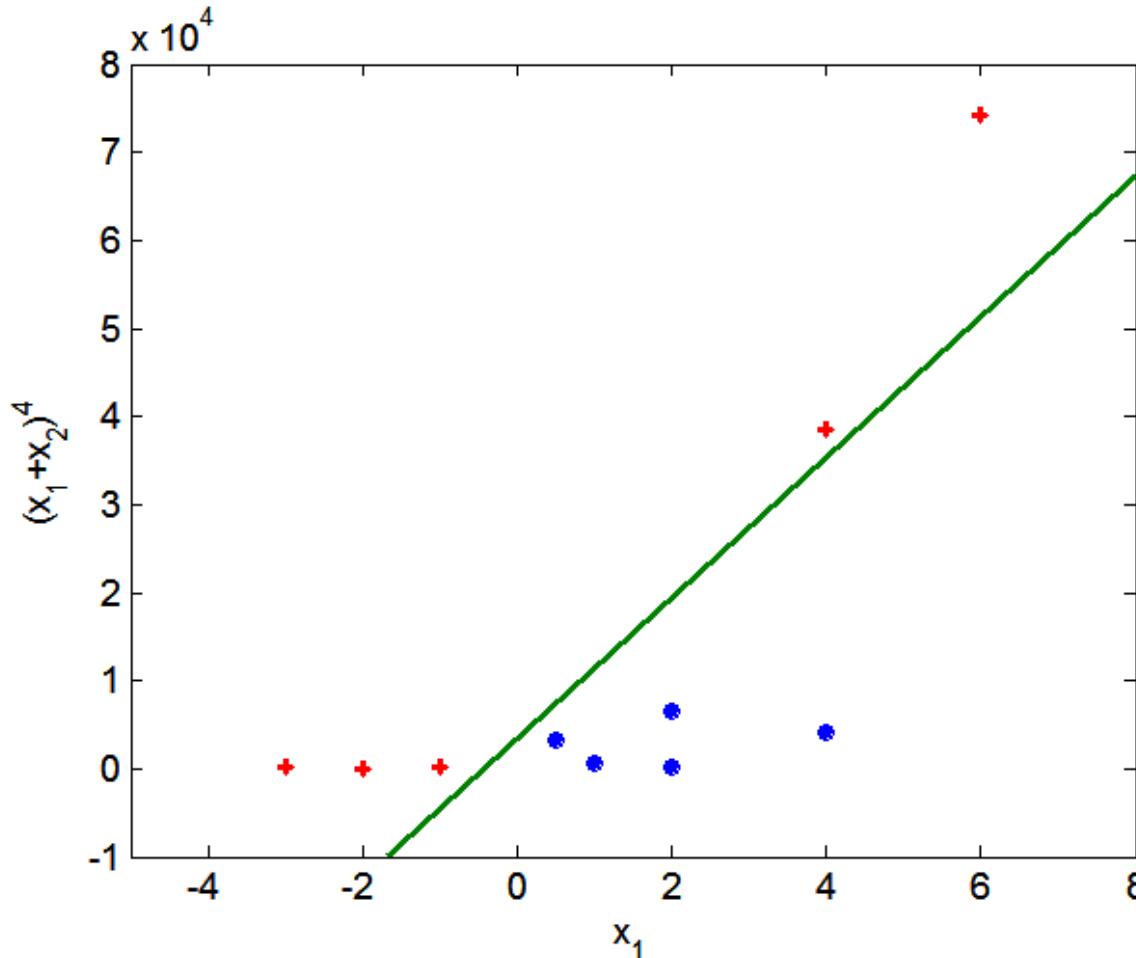
Nonlinear Support Vector Machines

- What if decision boundary is not linear?



Nonlinear Support Vector Machines

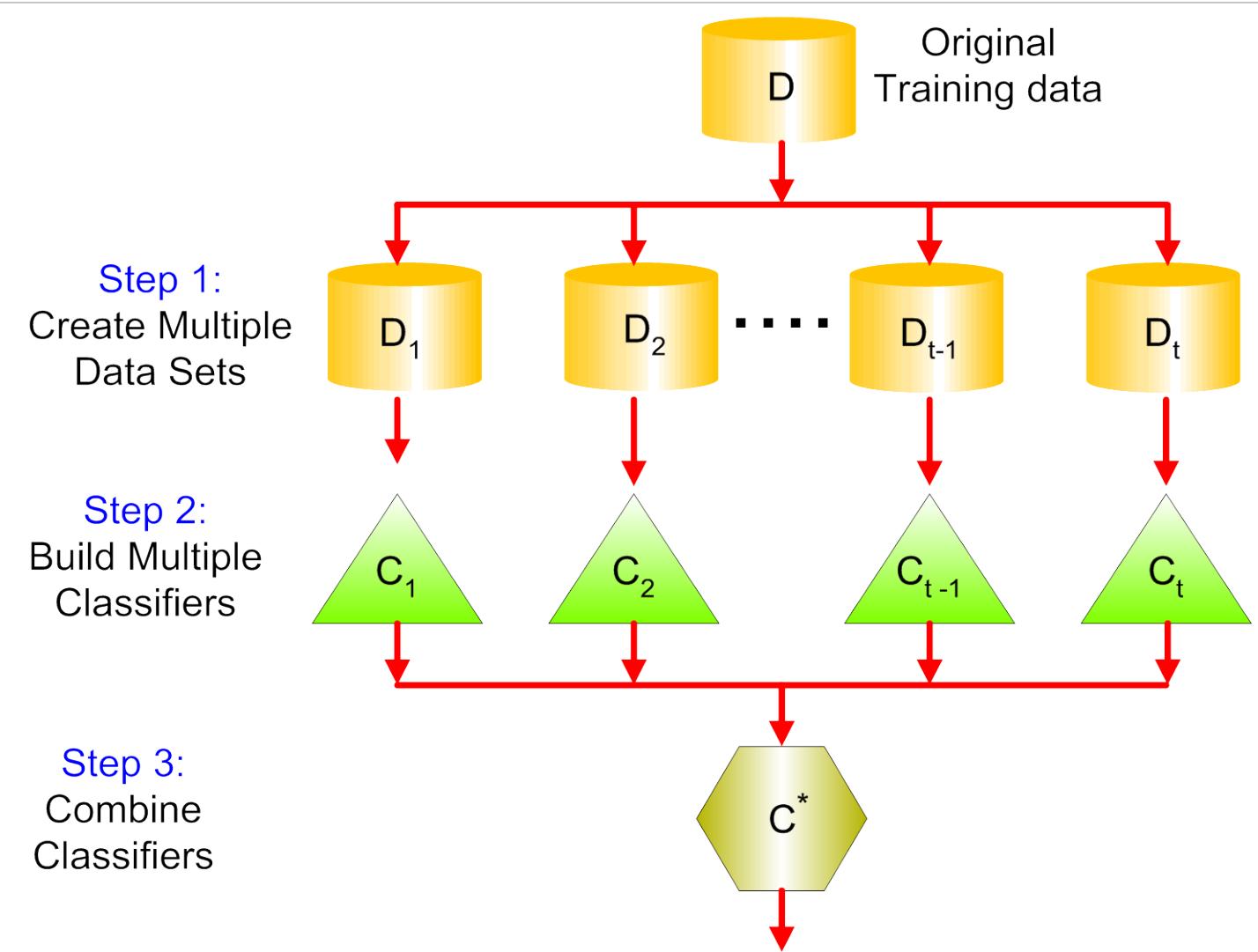
- Transform data into higher dimensional space



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:

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

Examples of Ensemble Methods

- How to generate an ensemble of classifiers?
 - Bagging
 - Boosting

Bagging

- Sampling with replacement

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build classifier on each bootstrap sample
- Each sample has probability $(1 - 1/n)^n$ of being selected

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
 - Initially, all N records are assigned equal weights
 - Unlike bagging, weights may change at the end of boosting round

Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

- Example 4 is hard to classify
- Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

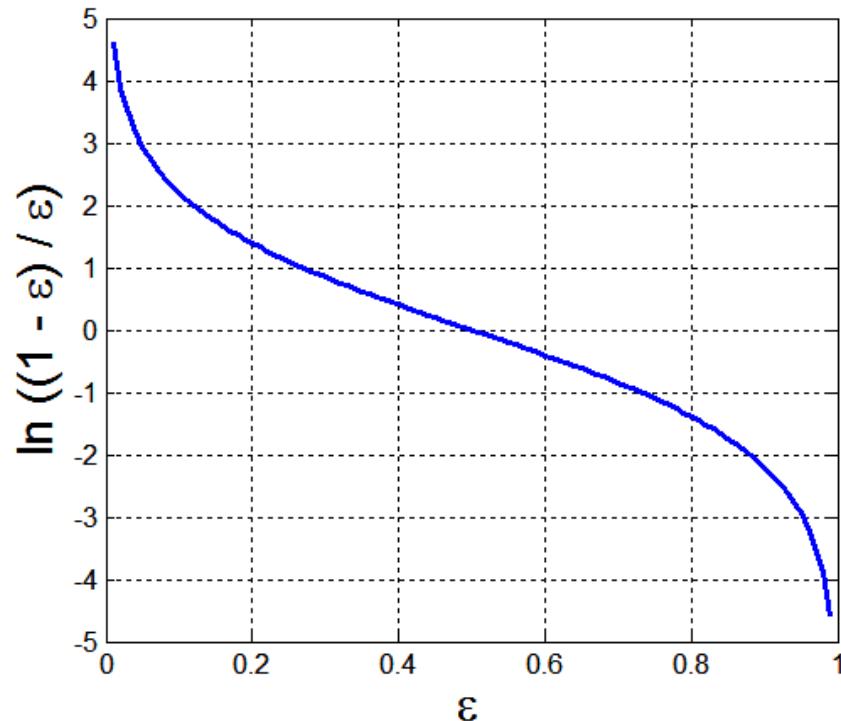
Example: AdaBoost

- Base classifiers: C_1, C_2, \dots, C_T
- Error rate:

$$\varepsilon_i = \frac{1}{N} \sum_{j=1}^N w_j \delta(C_i(x_j) \neq y_j)$$

- Importance of a classifier:

$$\alpha_i = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_i}{\varepsilon_i} \right)$$



Example: AdaBoost

- Weight update:

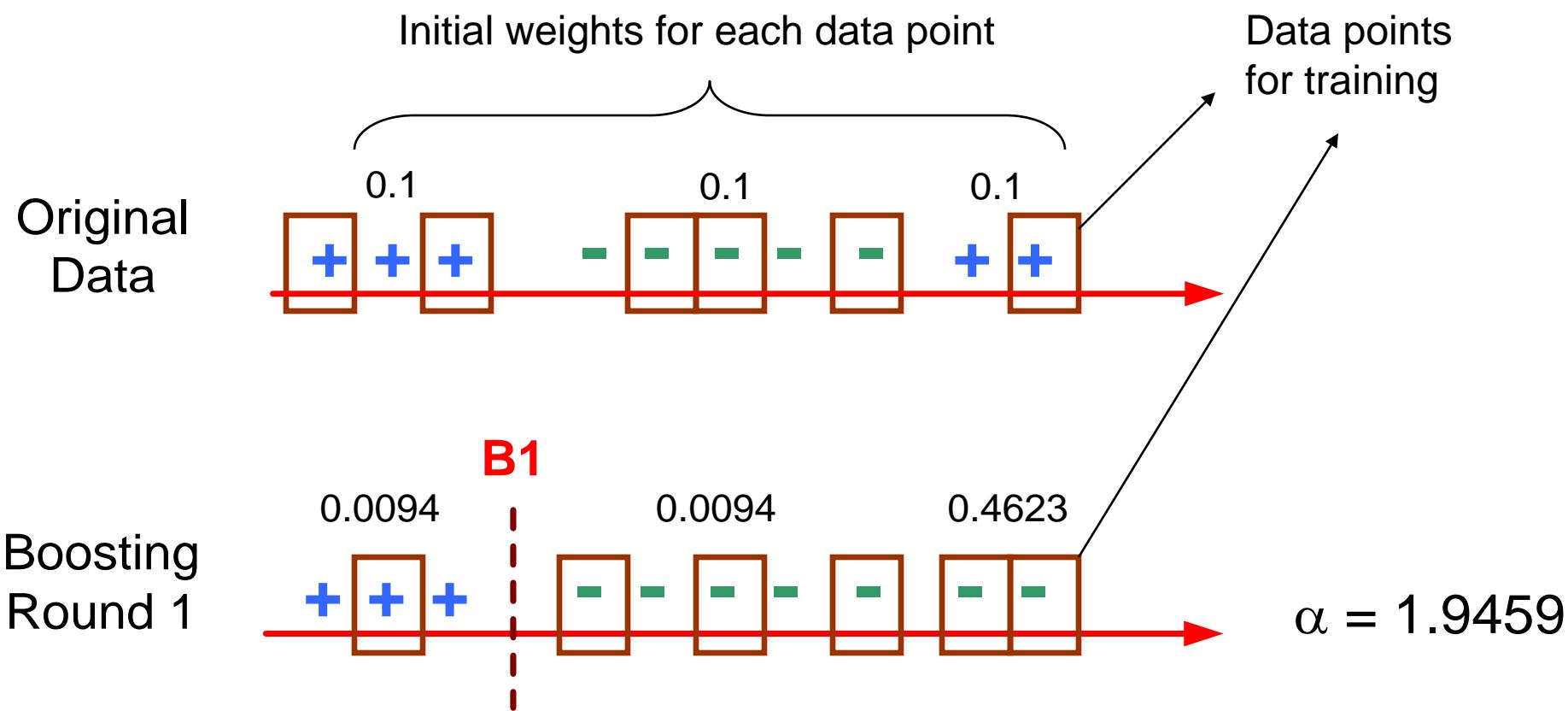
$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \begin{cases} \exp^{-\alpha_j} & \text{if } C_j(x_i) = y_i \\ \exp^{\alpha_j} & \text{if } C_j(x_i) \neq y_i \end{cases}$$

where Z_j is the normalization factor

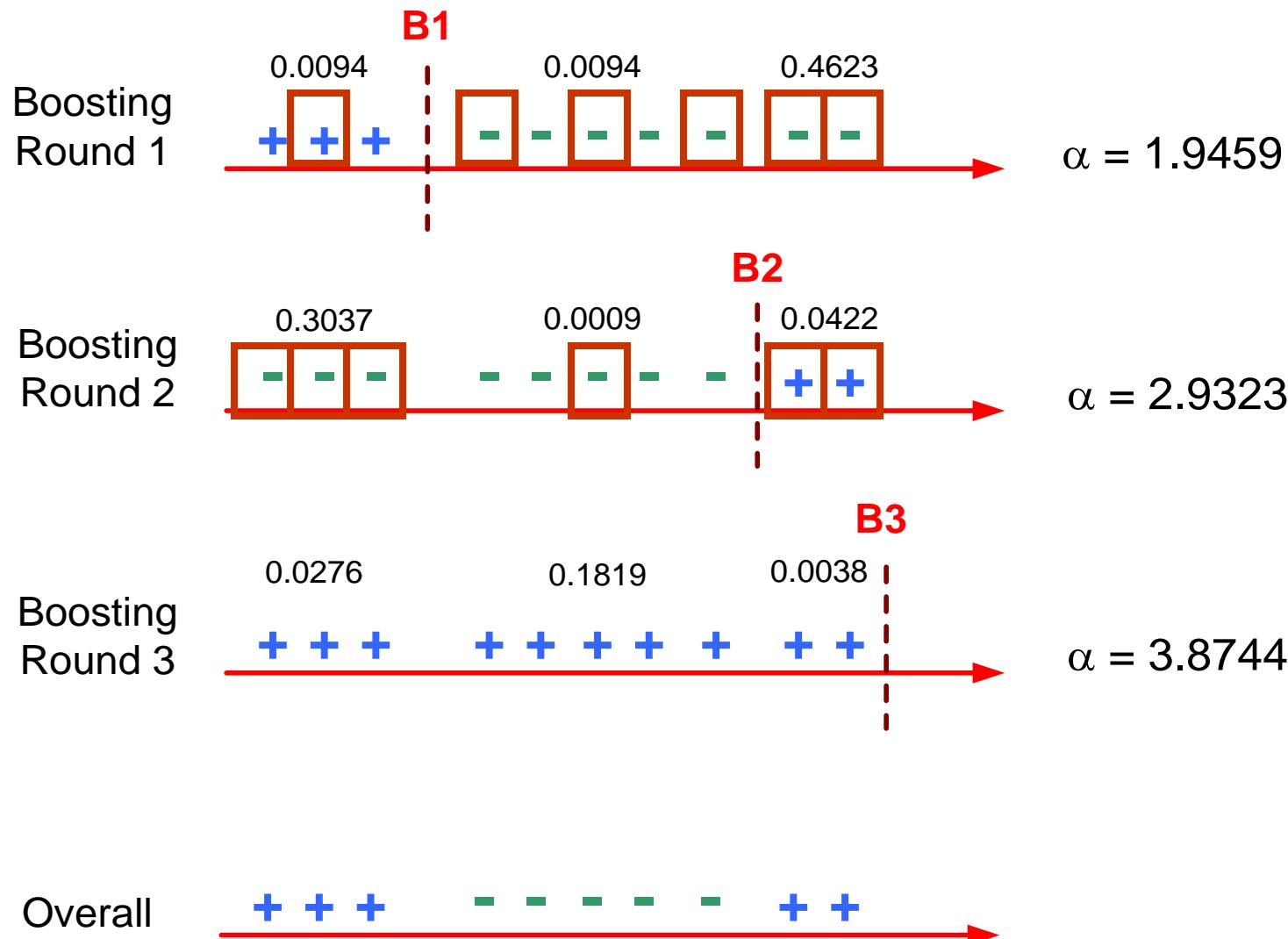
- If any intermediate rounds produce error rate higher than 50%, the weights are reverted back to $1/n$ and the resampling procedure is repeated
- Classification:

$$C^*(x) = \arg \max_y \sum_{j=1}^T \alpha_j \delta(C_j(x) = y)$$

Illustrating AdaBoost



Illustrating AdaBoost



Data Mining Association Analysis: Basic Concepts and Algorithms

Lecture Notes for Chapter 6

Introduction to Data Mining

by

Tan, Steinbach, Kumar

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

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Association Rules

$\{\text{Diaper}\} \rightarrow \{\text{Beer}\}$,
 $\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\}$,
 $\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\}$,

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

<i>TID</i>	<i>Items</i>
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
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:
 $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

- **Rule Evaluation Metrics**

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

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

Example:

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

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

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

Association Rule Mining Task

- Given a set of transactions T , the goal of association rule mining is to find all rules having
 - $\text{support} \geq \text{minsup}$ threshold
 - $\text{confidence} \geq \text{minconf}$ 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, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

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

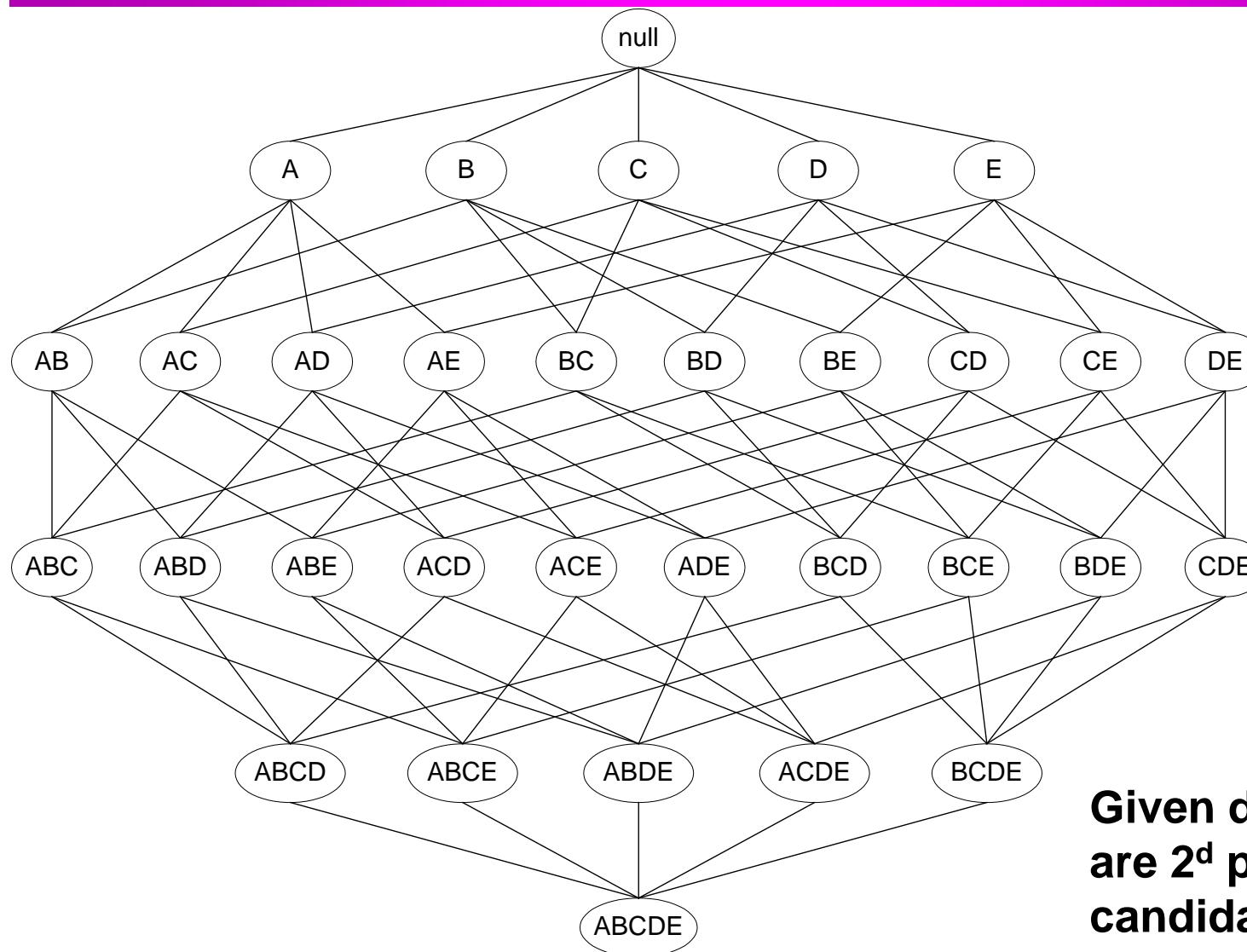
Observations:

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

Mining Association Rules

- 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

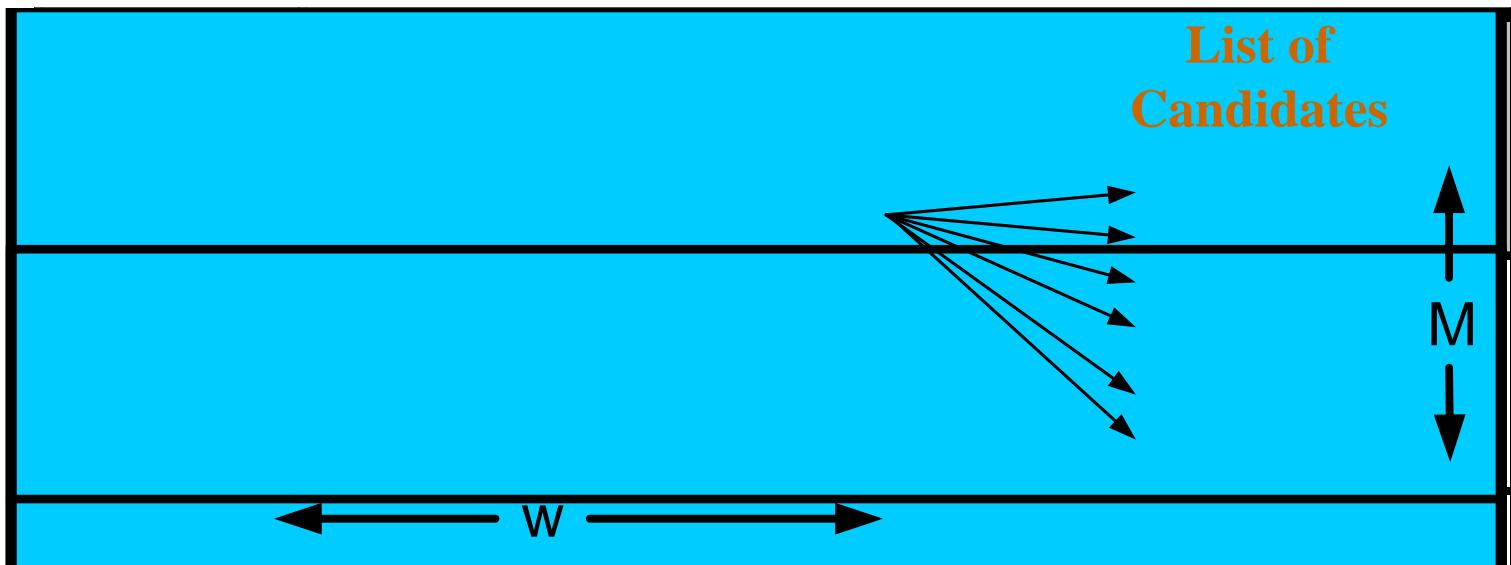


**Given d items, there
are 2^d possible
candidate itemsets**

Frequent Itemset Generation

- Brute-force approach:

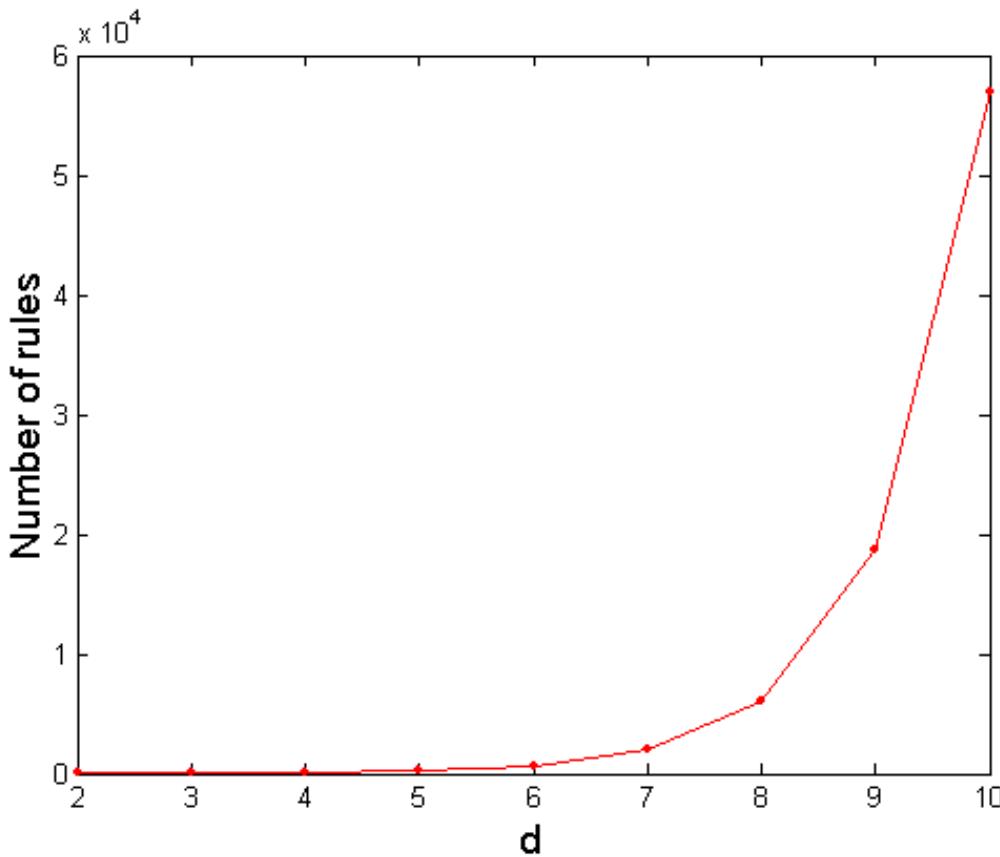
- Each itemset in the lattice is a **candidate** frequent itemset
- Count the support of each candidate by scanning the database



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

Computational Complexity

- Given d unique items:
 - Total number of itemsets = 2^d
 - Total number of possible association rules:



$$R = \sum_{k=1}^{d-1} \binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j}$$
$$= 3^d - 2^{d+1} + 1$$

If $d=6$, $R = 602$ rules

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 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

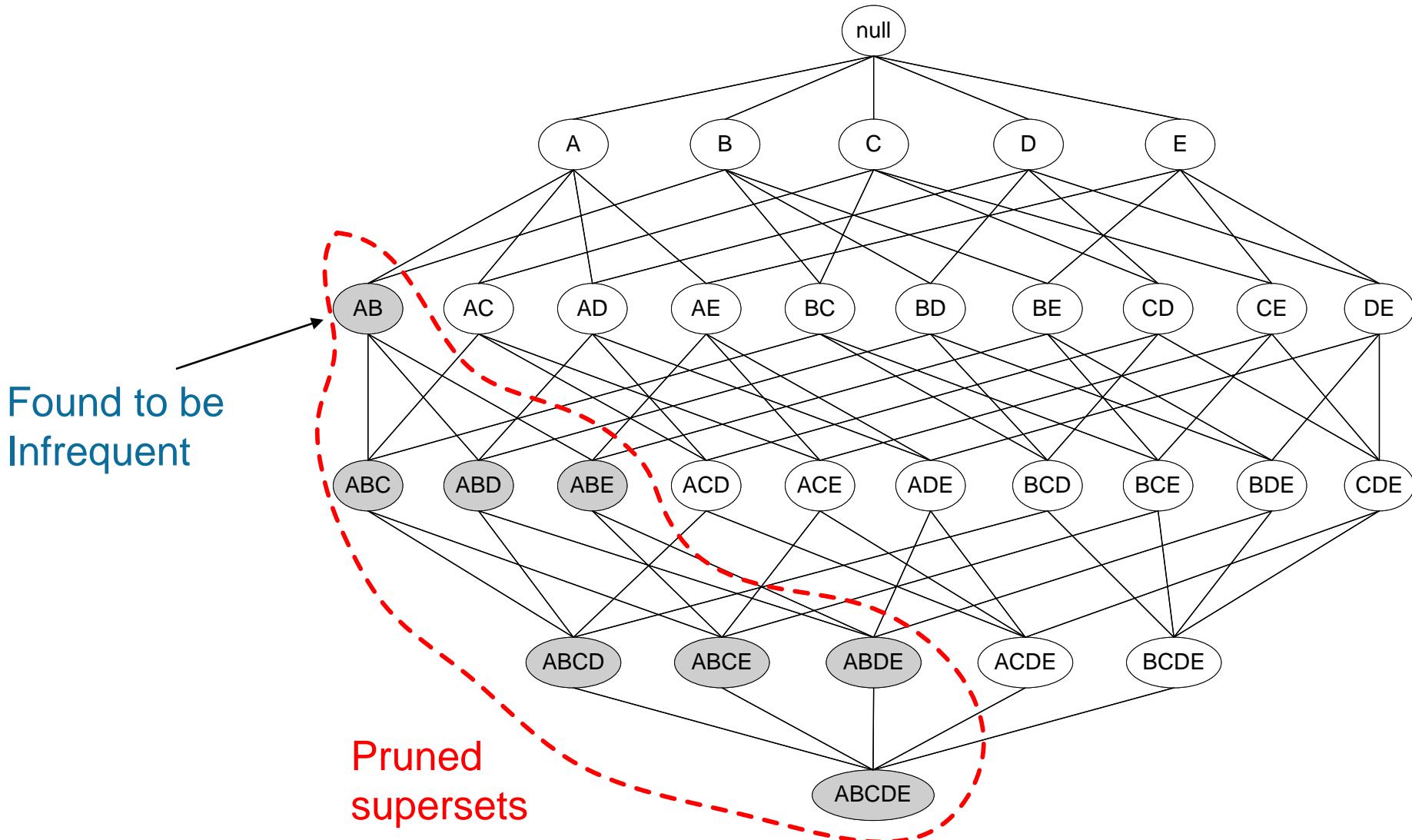
Reducing Number of Candidates

- **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

Illustrating Apriori Principle



Illustrating Apriori Principle

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

Items (1-itemsets)



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

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,

$${}^6C_1 + {}^6C_2 + {}^6C_3 = 41$$

With support-based pruning,

$$6 + 6 + 1 = 13$$



Triplets (3-itemsets)

Itemset	Count
{Bread,Milk,Diaper}	3



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

Reducing Number of Comparisons

- Candidate counting:

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



Generate Hash Tree

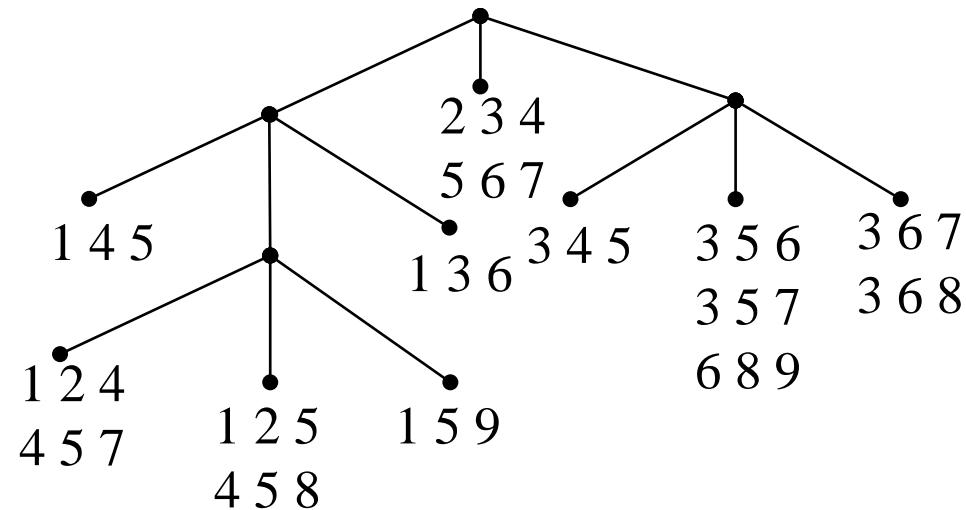
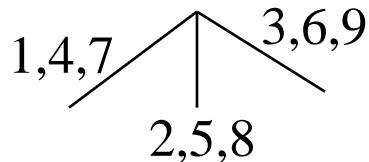
Suppose you have 15 candidate itemsets of length 3:

$\{1\ 4\ 5\}$, $\{1\ 2\ 4\}$, $\{4\ 5\ 7\}$, $\{1\ 2\ 5\}$, $\{4\ 5\ 8\}$, $\{1\ 5\ 9\}$, $\{1\ 3\ 6\}$, $\{2\ 3\ 4\}$, $\{5\ 6\ 7\}$, $\{3\ 4\ 5\}$,
 $\{3\ 5\ 6\}$, $\{3\ 5\ 7\}$, $\{6\ 8\ 9\}$, $\{3\ 6\ 7\}$, $\{3\ 6\ 8\}$

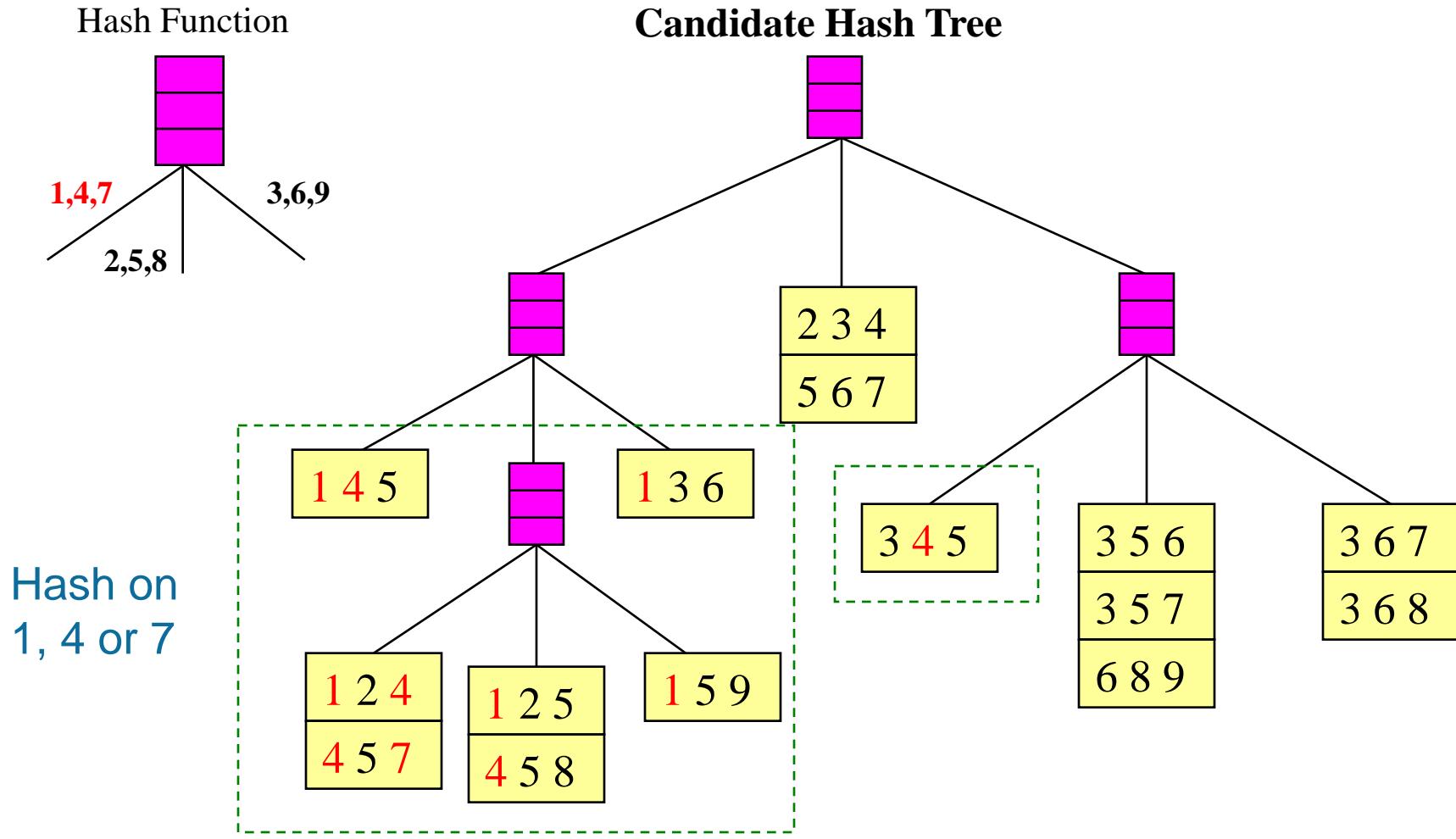
You need:

- Hash function
- Max leaf size: max number of itemsets stored in a leaf node (if number of candidate itemsets exceeds max leaf size, split the node)

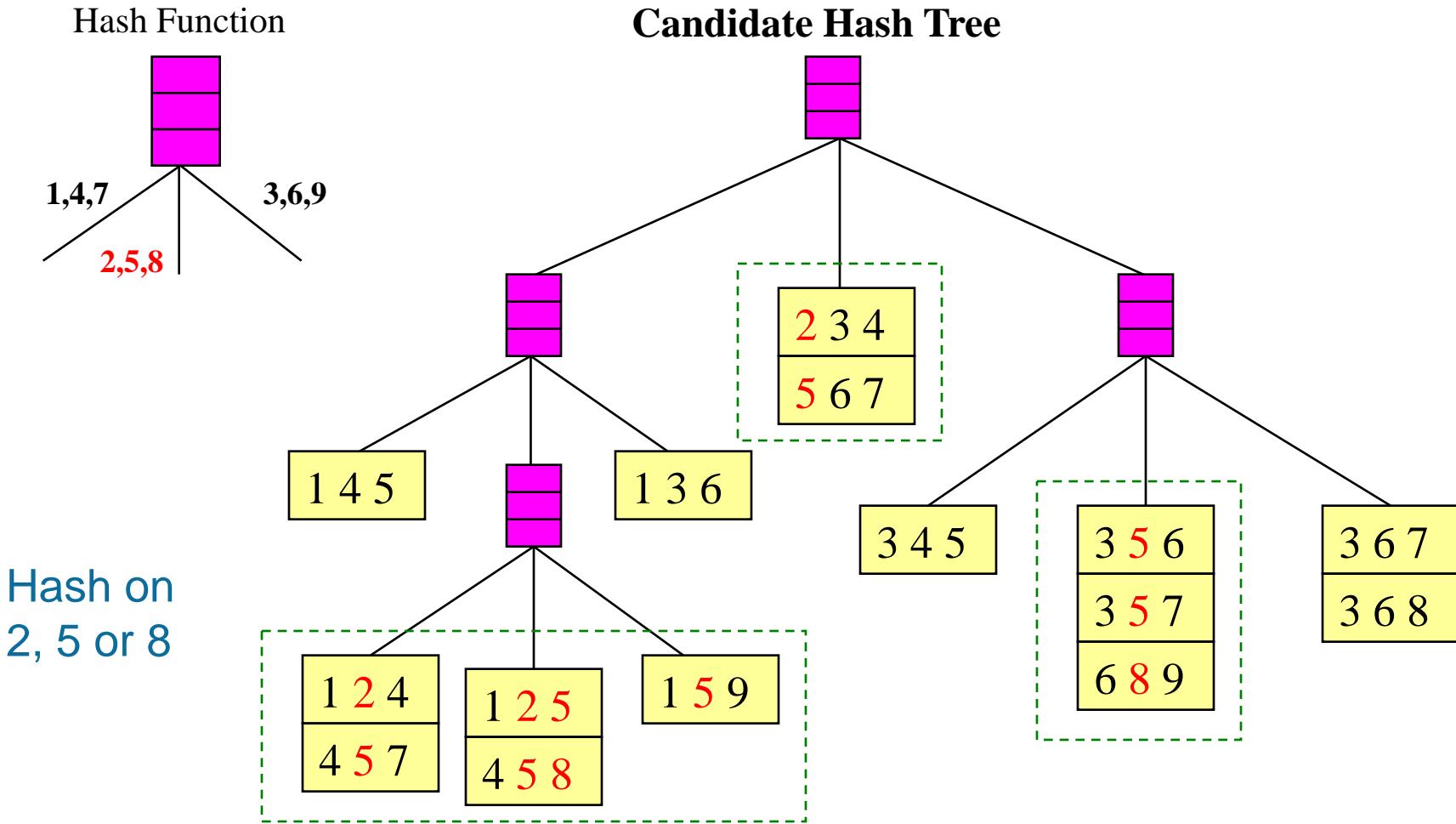
Hash function



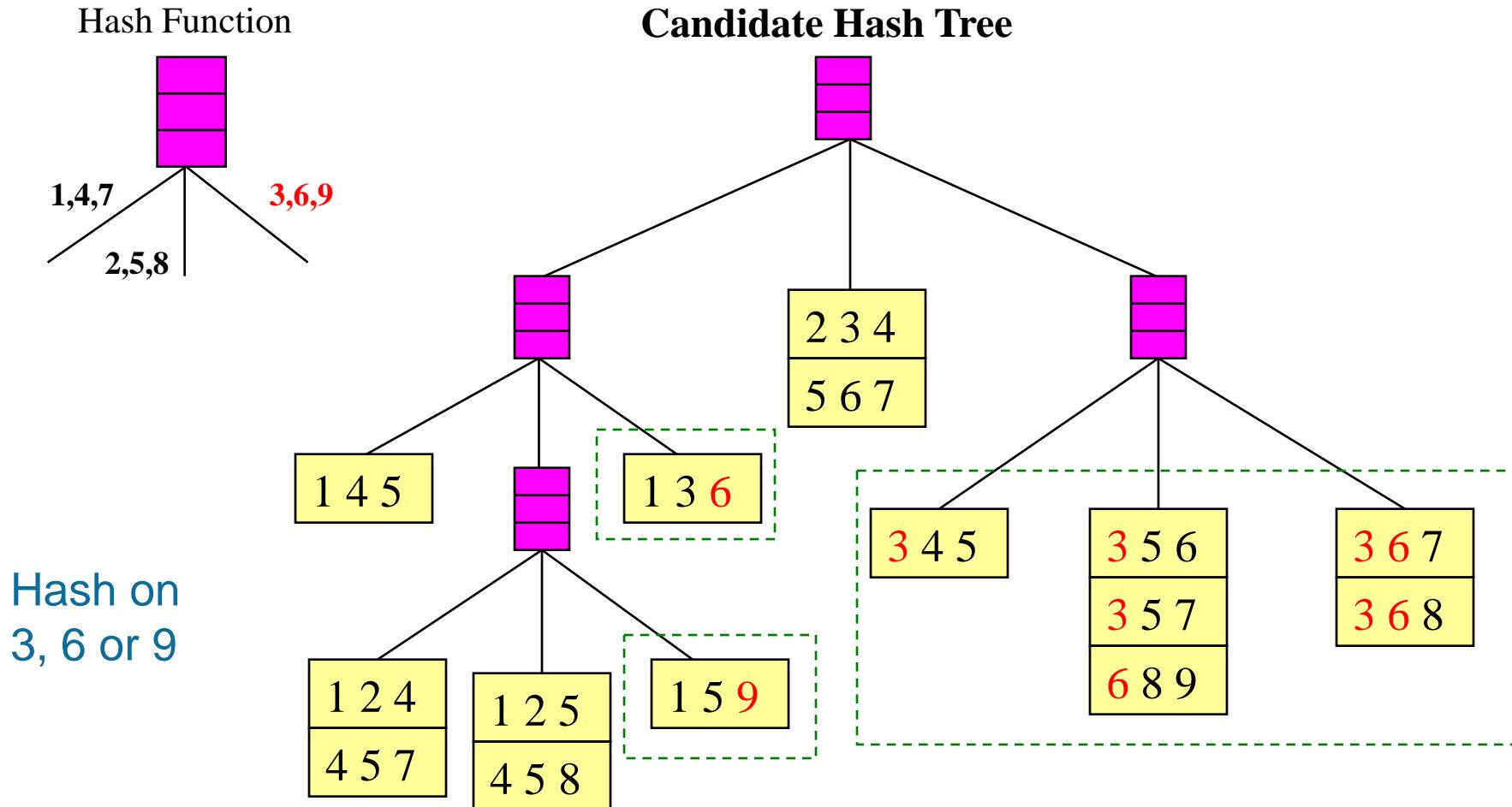
Association Rule Discovery: Hash tree



Association Rule Discovery: Hash tree

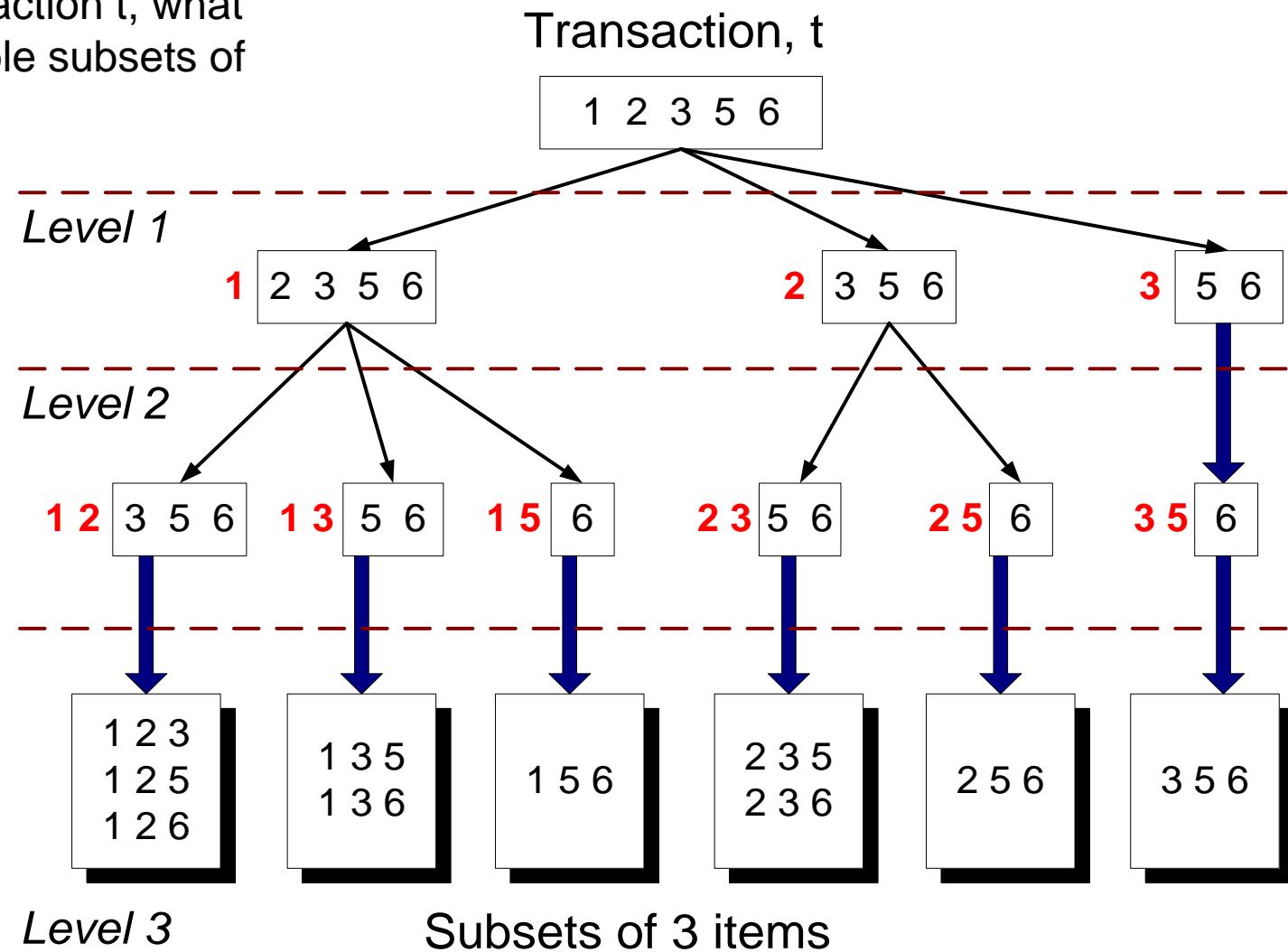


Association Rule Discovery: Hash tree

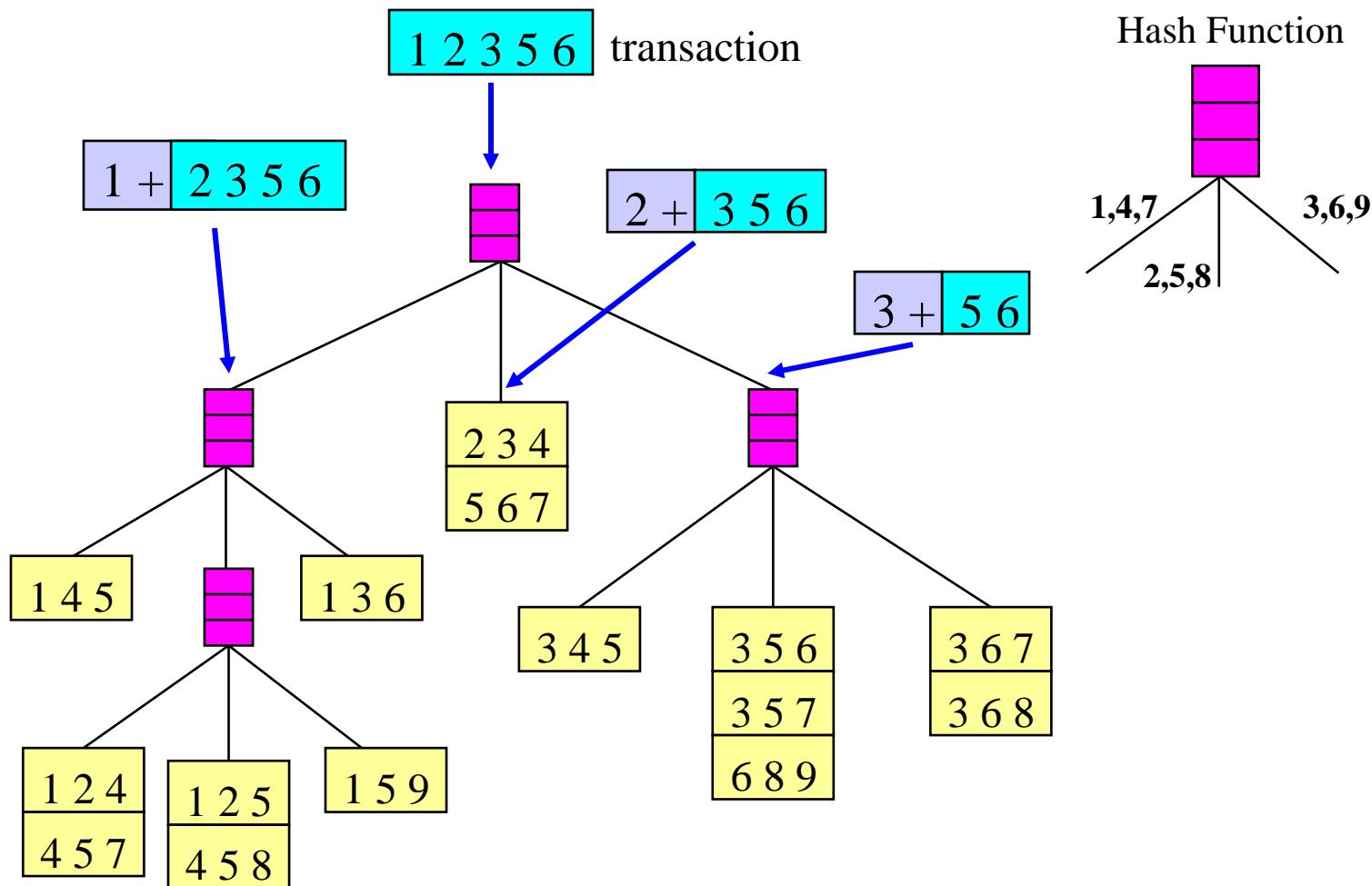


Subset Operation

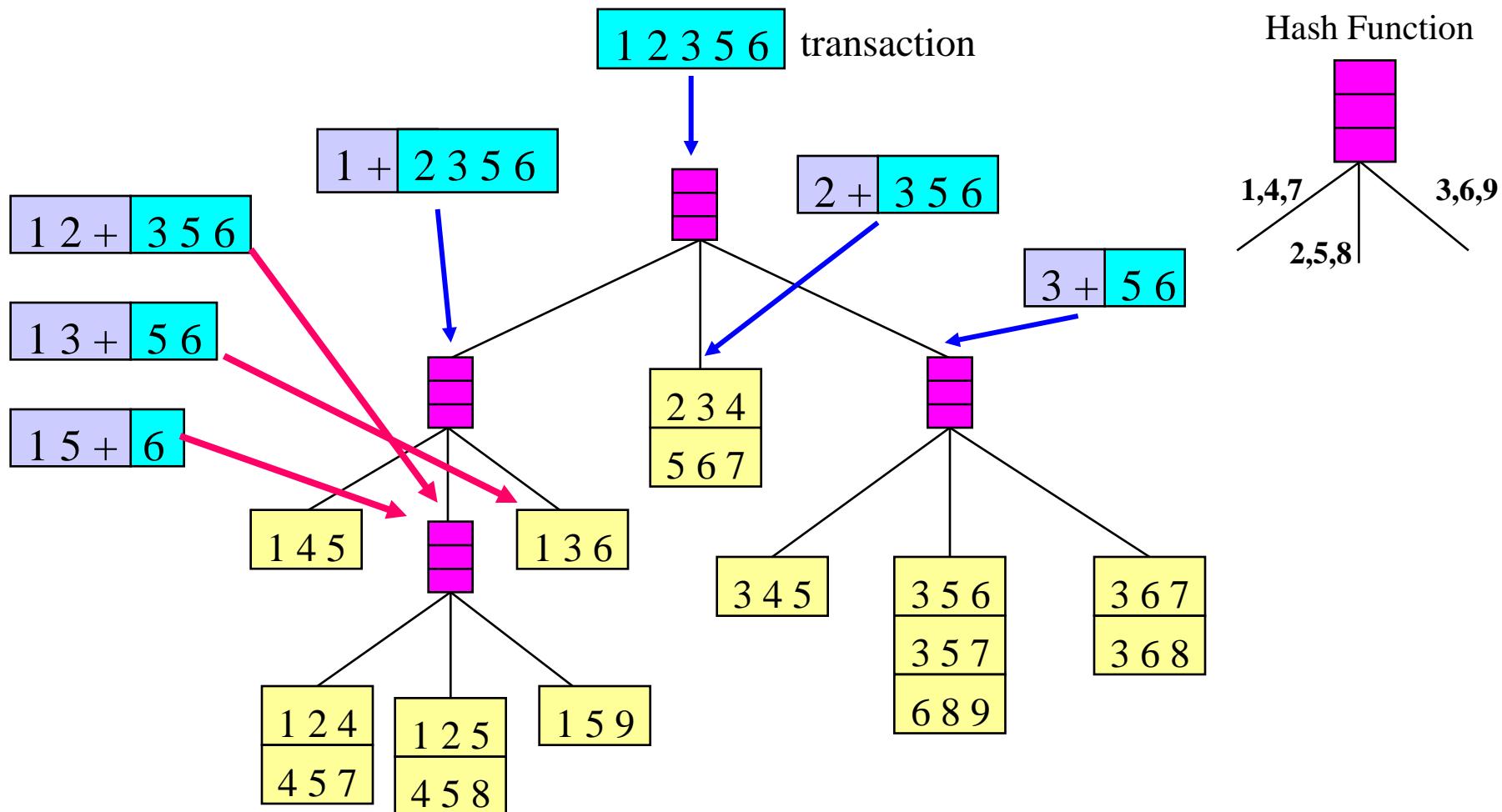
Given a transaction t , what are the possible subsets of size 3?



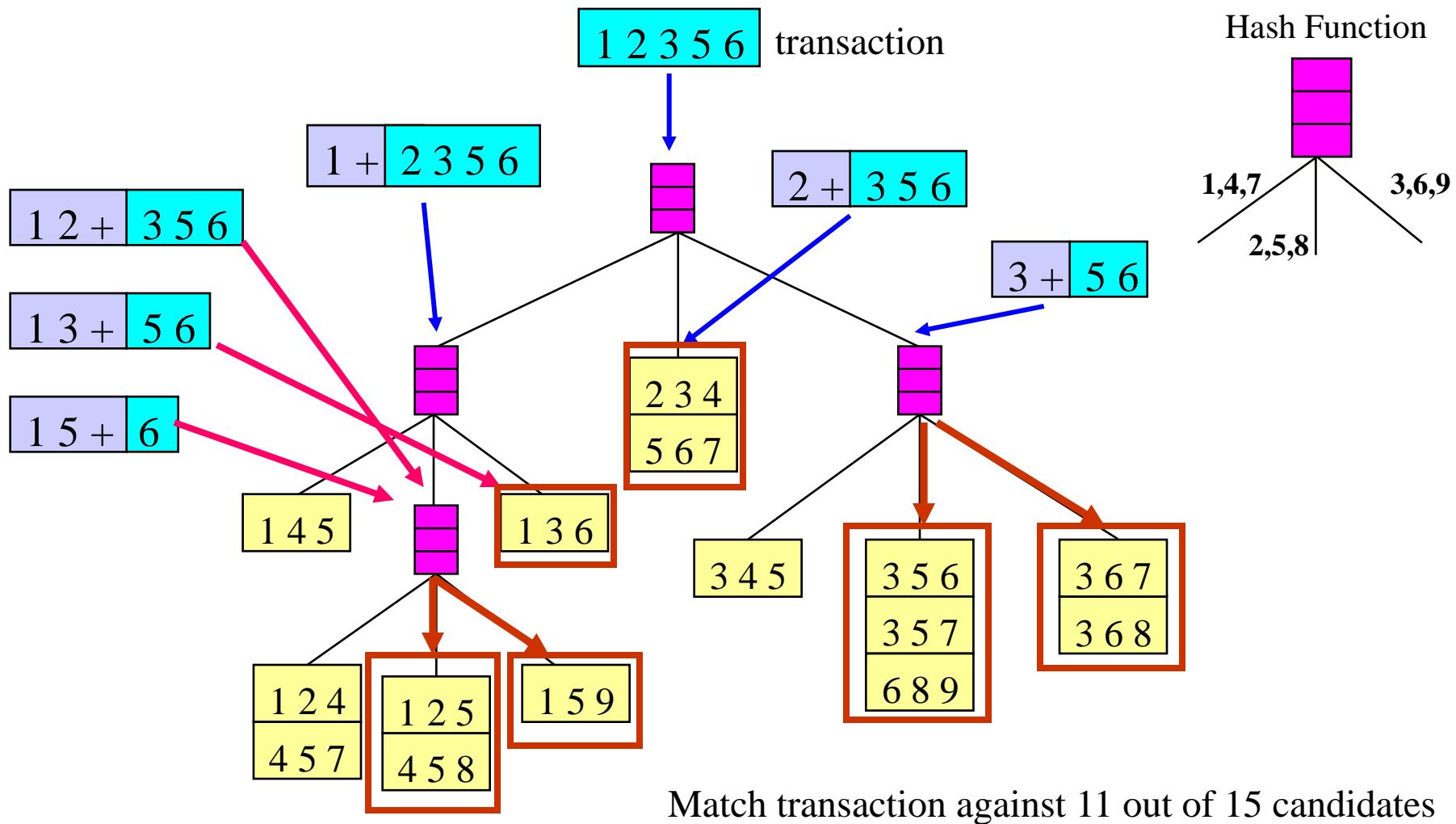
Subset Operation Using Hash Tree



Subset Operation Using Hash Tree



Subset Operation Using Hash Tree



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)

Compact Representation of Frequent Itemsets

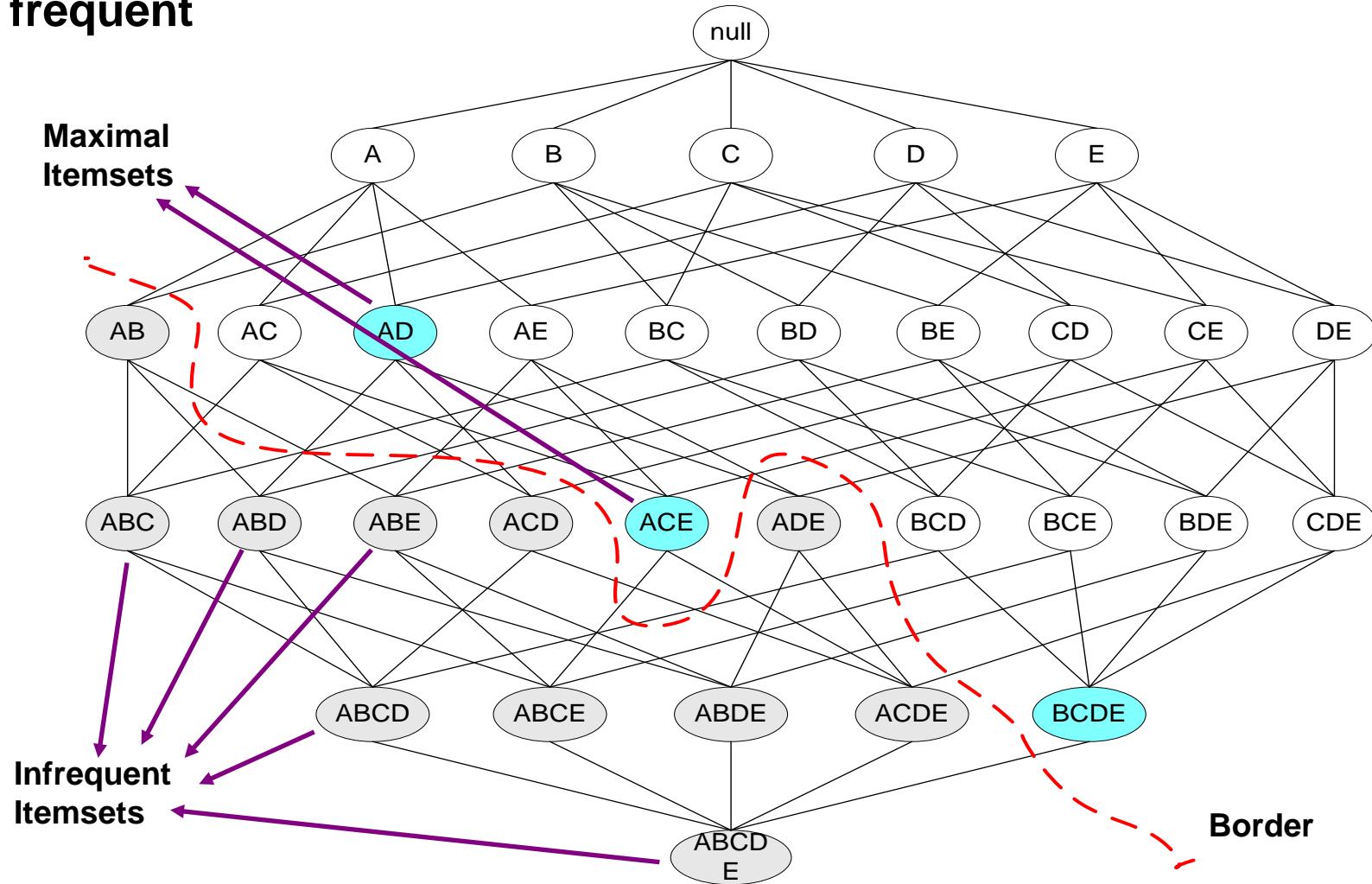
- Some itemsets are redundant because they have identical support as their supersets

TID	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	

- Number of frequent itemsets = $3 \times \sum_{k=1}^{10} \binom{10}{k}$
- Need a compact representation

Maximal Frequent Itemset

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



Closed Itemset

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

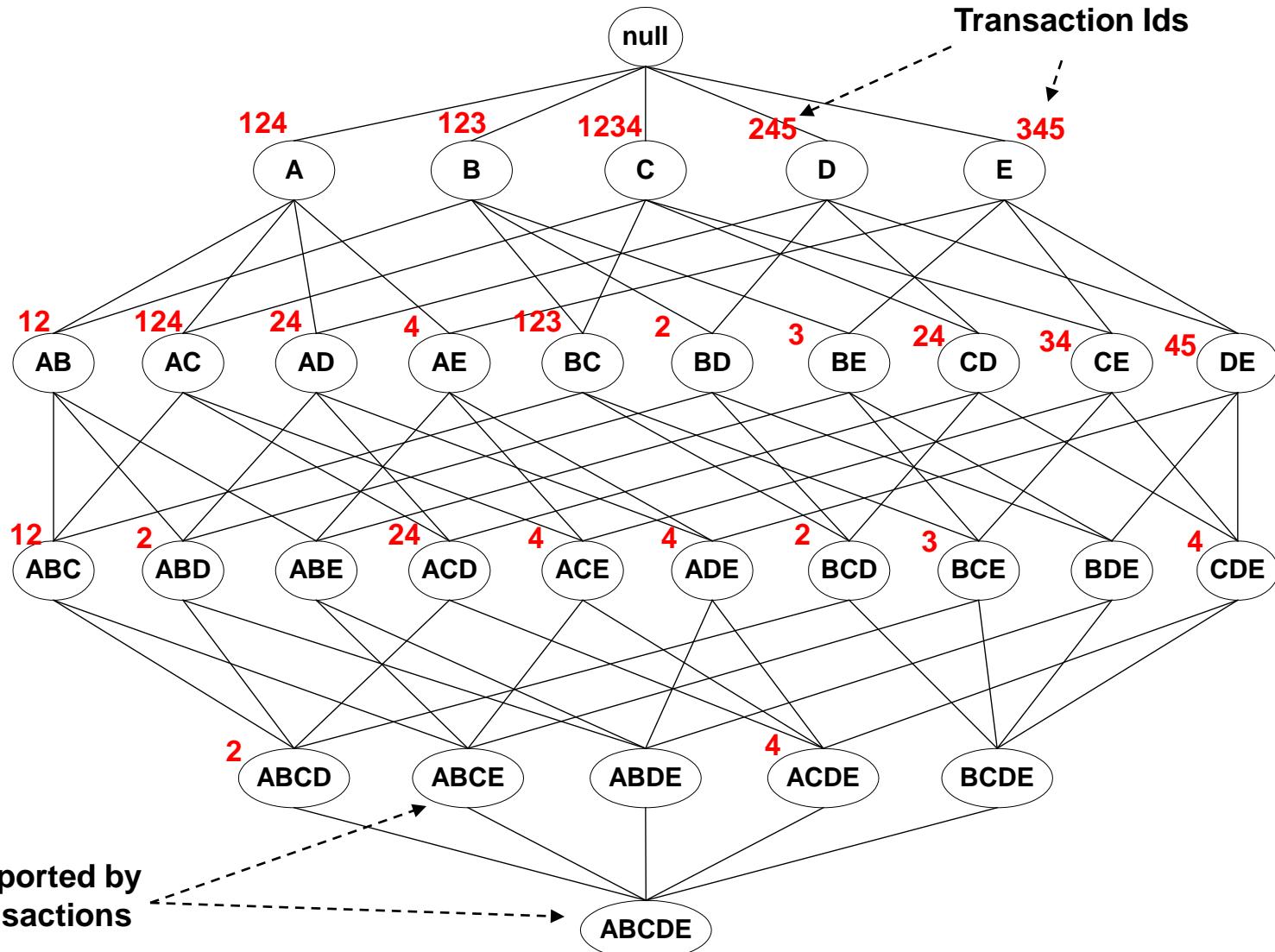
TID	Items
1	{A,B}
2	{B,C,D}
3	{A,B,C,D}
4	{A,B,D}
5	{A,B,C,D}

Itemset	Support
{A}	4
{B}	5
{C}	3
{D}	4
{A,B}	4
{A,C}	2
{A,D}	3
{B,C}	3
{B,D}	4
{C,D}	3

Itemset	Support
{A,B,C}	2
{A,B,D}	3
{A,C,D}	2
{B,C,D}	3
{A,B,C,D}	2

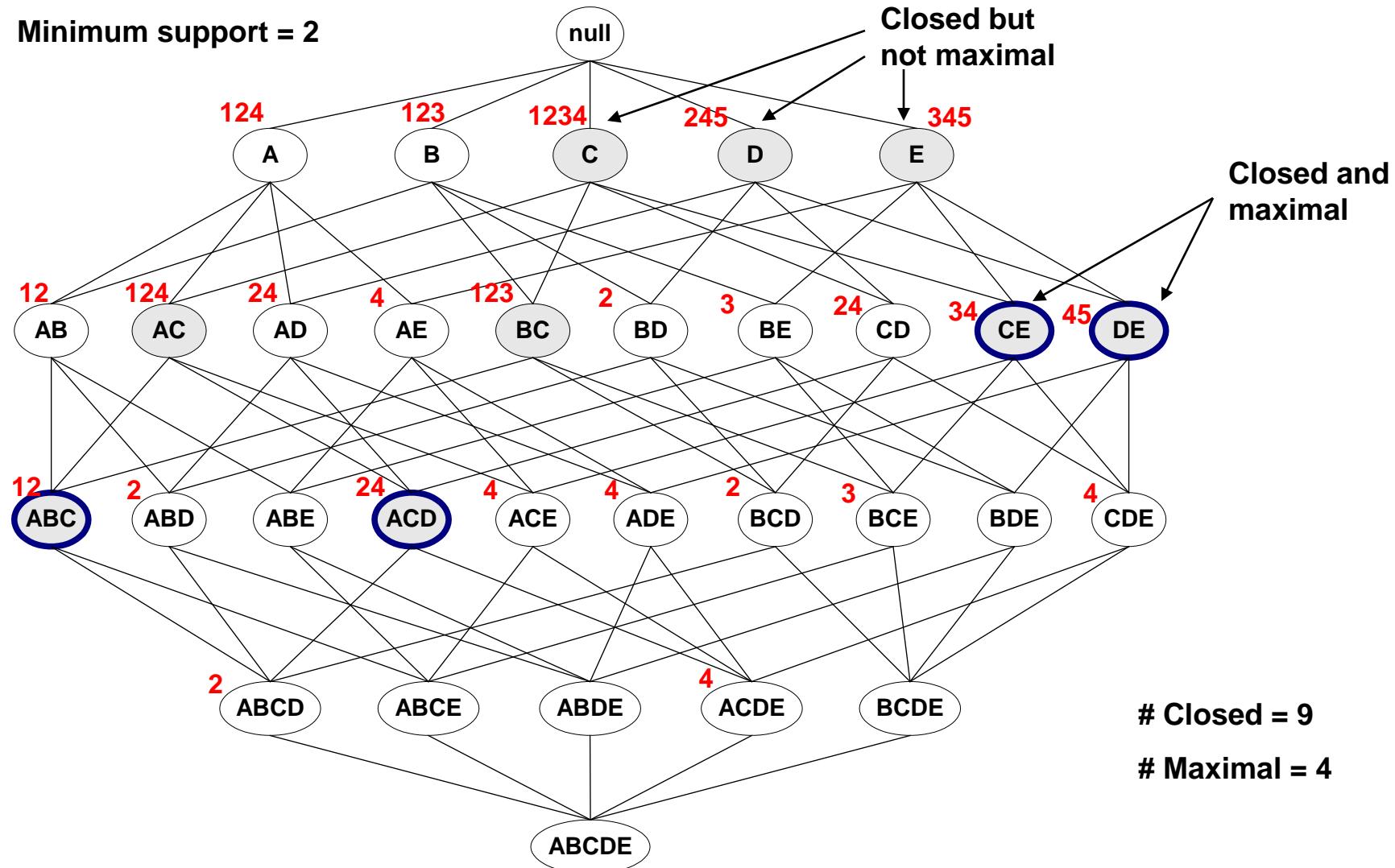
Maximal vs Closed Itemsets

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

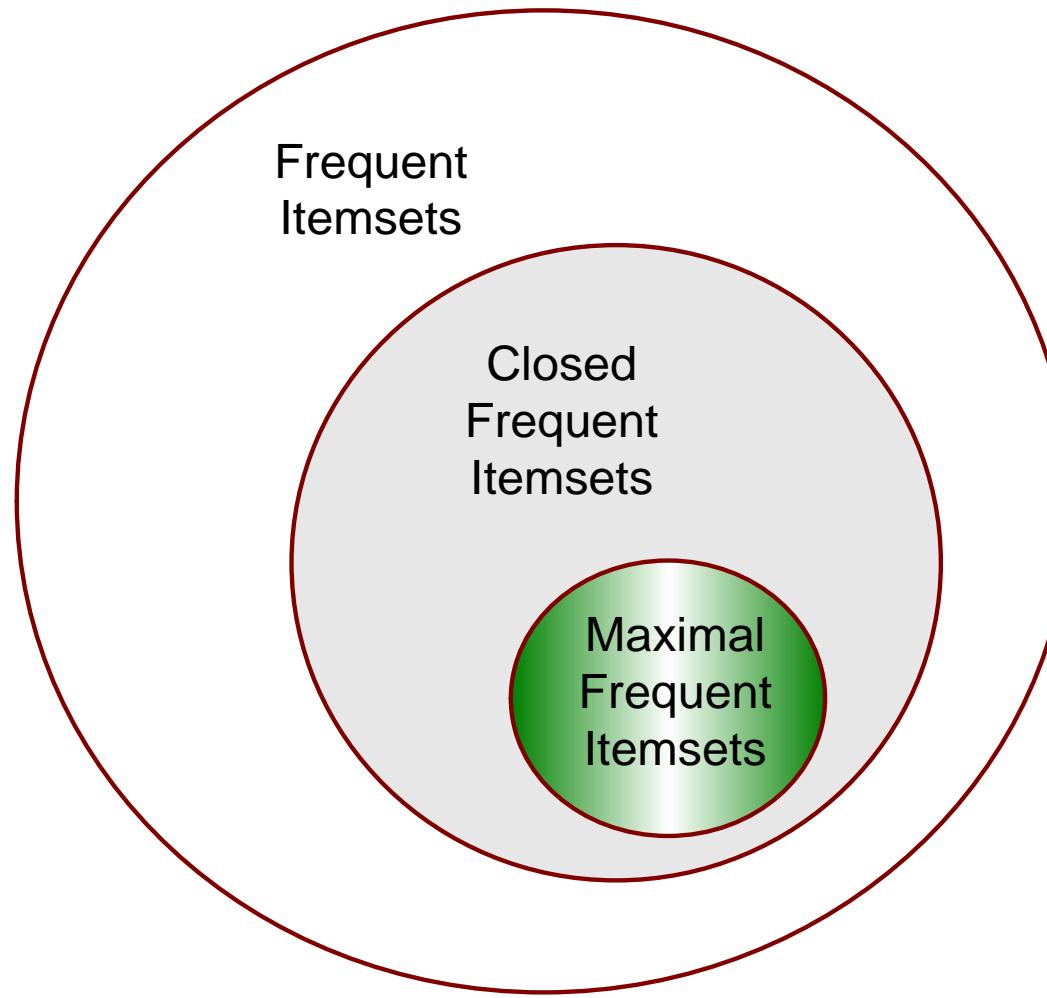


Maximal vs Closed Frequent Itemsets

Minimum support = 2

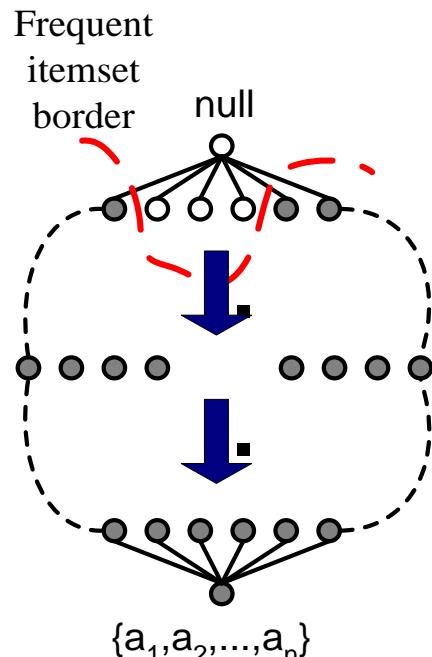


Maximal vs Closed Itemsets

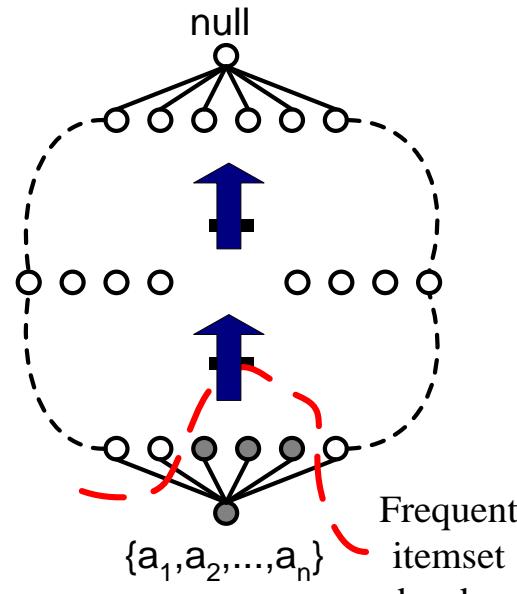


Alternative Methods for Frequent Itemset Generation

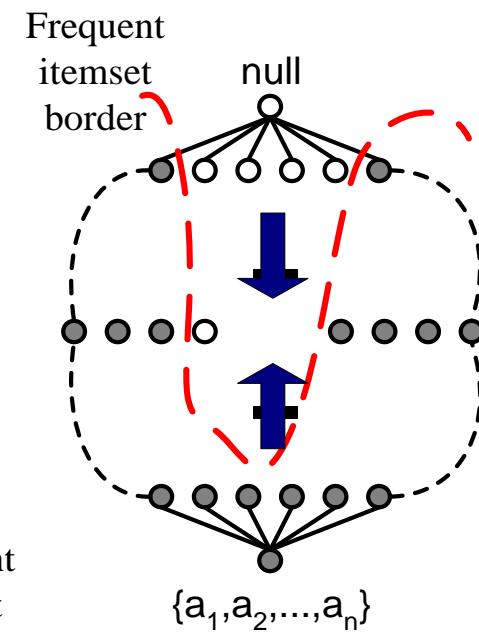
- Traversal of Itemset Lattice
 - General-to-specific vs Specific-to-general



(a) General-to-specific



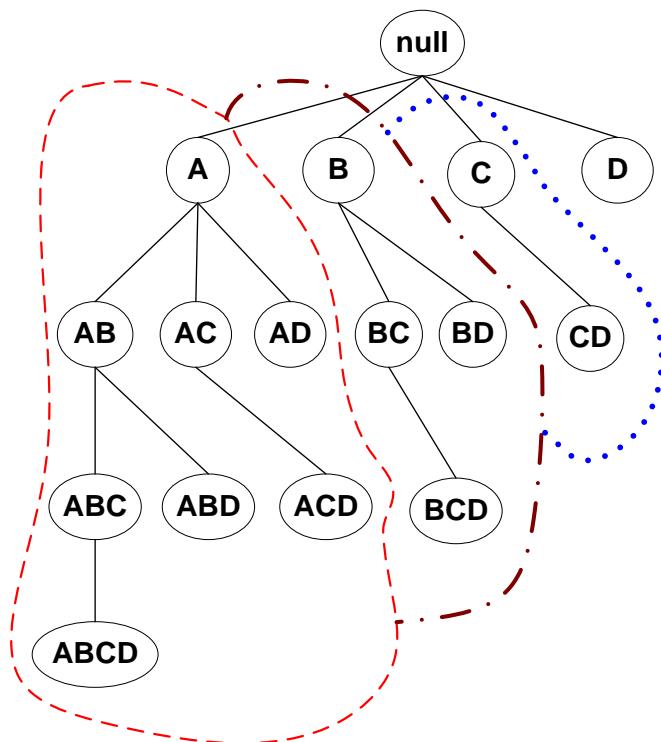
(b) Specific-to-general



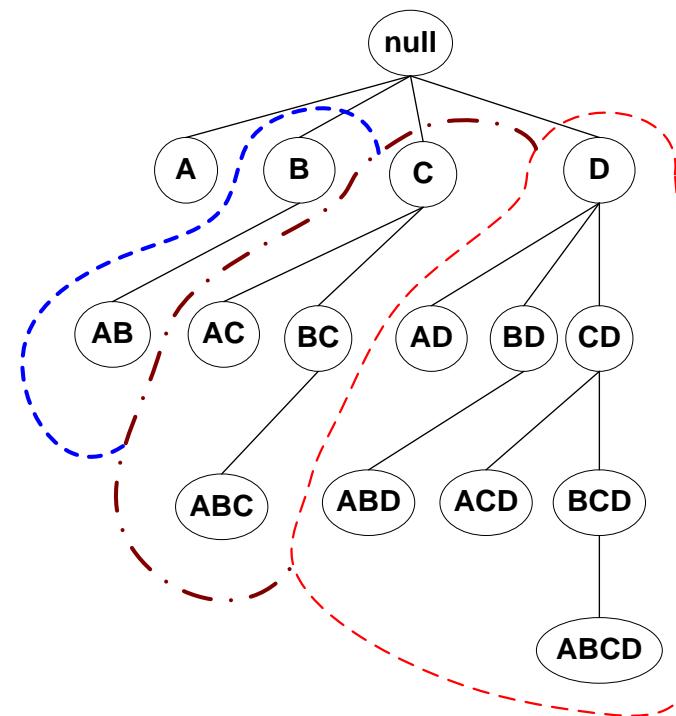
(c) Bidirectional

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice
 - Equivalent Classes



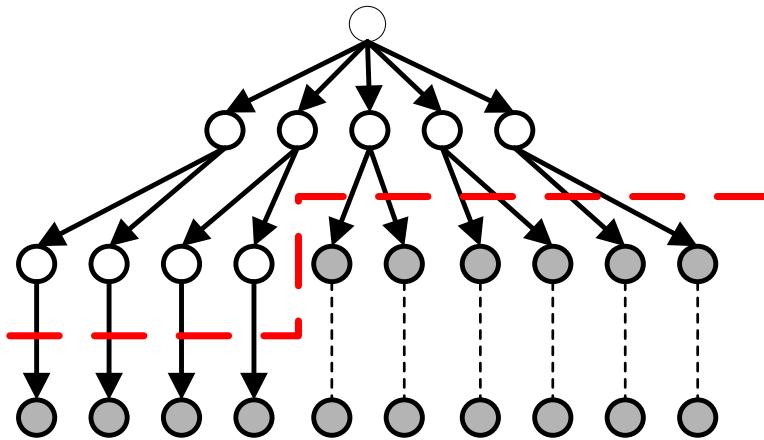
(a) Prefix tree



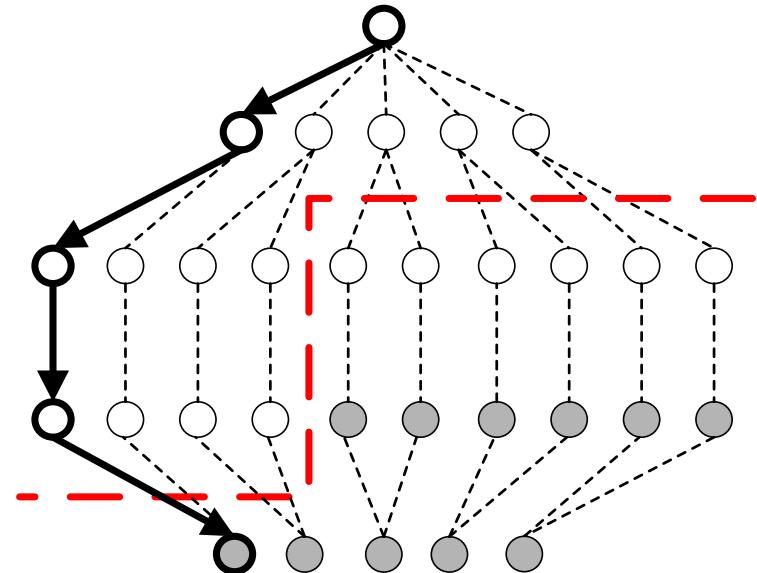
(b) Suffix tree

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice
 - Breadth-first vs Depth-first



(a) Breadth first



(b) Depth first

Alternative Methods for Frequent Itemset Generation

- Representation of Database
 - horizontal vs vertical data layout

A	B	C	D
1	1	A ₁ , B ₁ , C ₁ , D ₁	5
2	2	B ₂ , C ₂ , D ₂	2
3	3	C ₃ , E ₃	4
4	4	E ₄	4
5	5	A ₅ , C ₅ , D ₅	5
6	7	B ₆ , C ₆ , D ₆	1

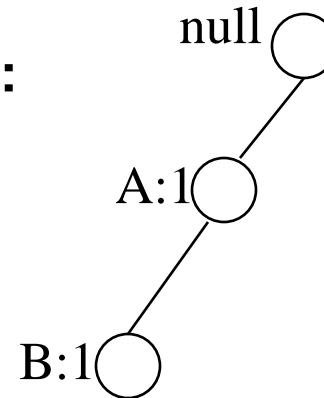
FP-growth Algorithm

- Use a compressed representation of the database using an **FP-tree**
- Once an FP-tree has been constructed, it uses a recursive divide-and-conquer approach to mine the frequent itemsets

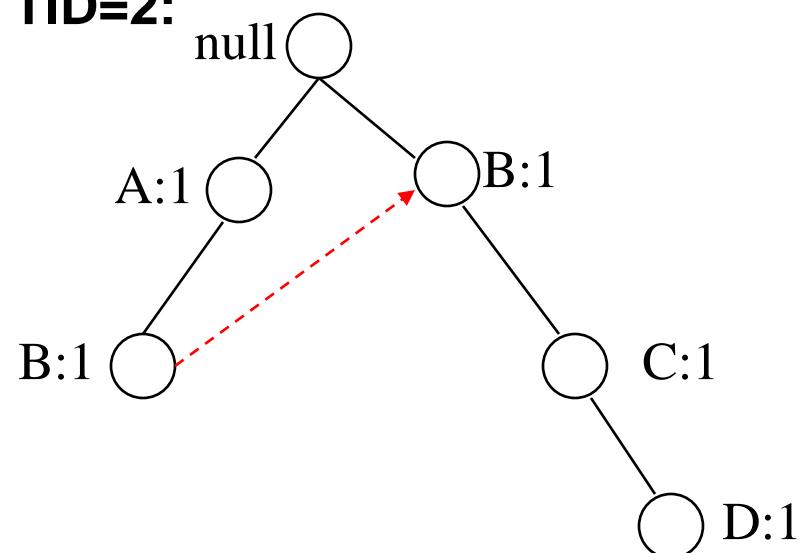
FP-tree construction

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=1:



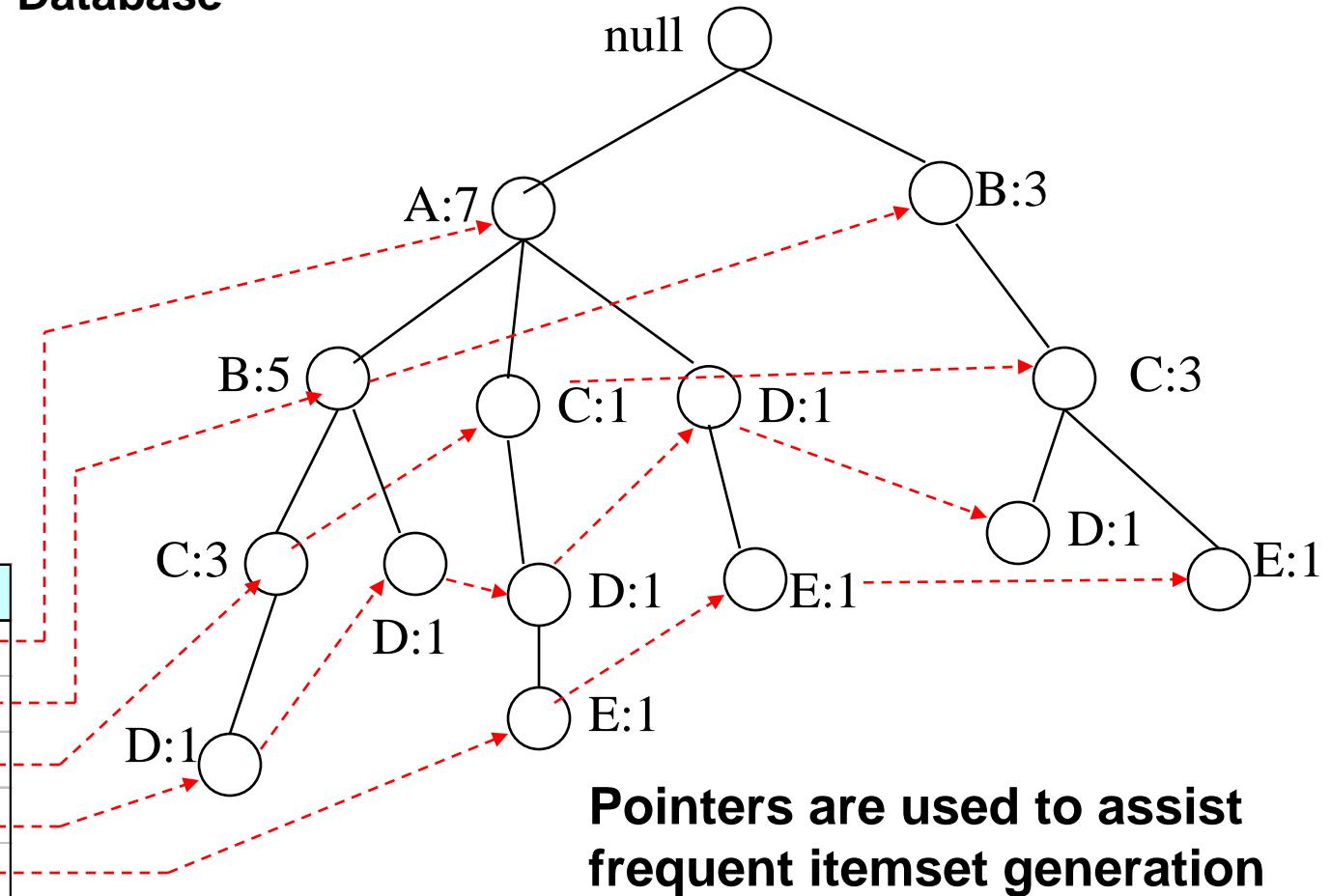
After reading TID=2:



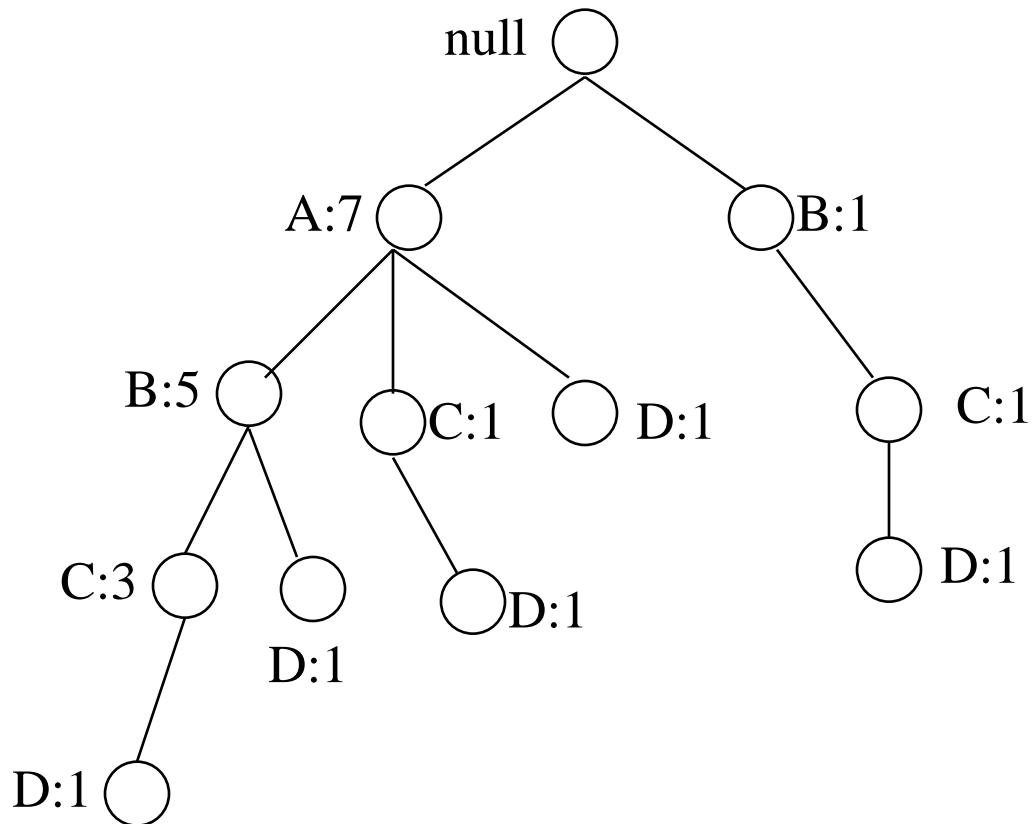
FP-Tree Construction

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Transaction Database



FP-growth



**Conditional Pattern base
for D:**

$$P = \{(A:1, B:1, C:1), (A:1, B:1), (A:1, C:1), (A:1), (B:1, C:1)\}$$

**Recursively apply FP-
growth on P**

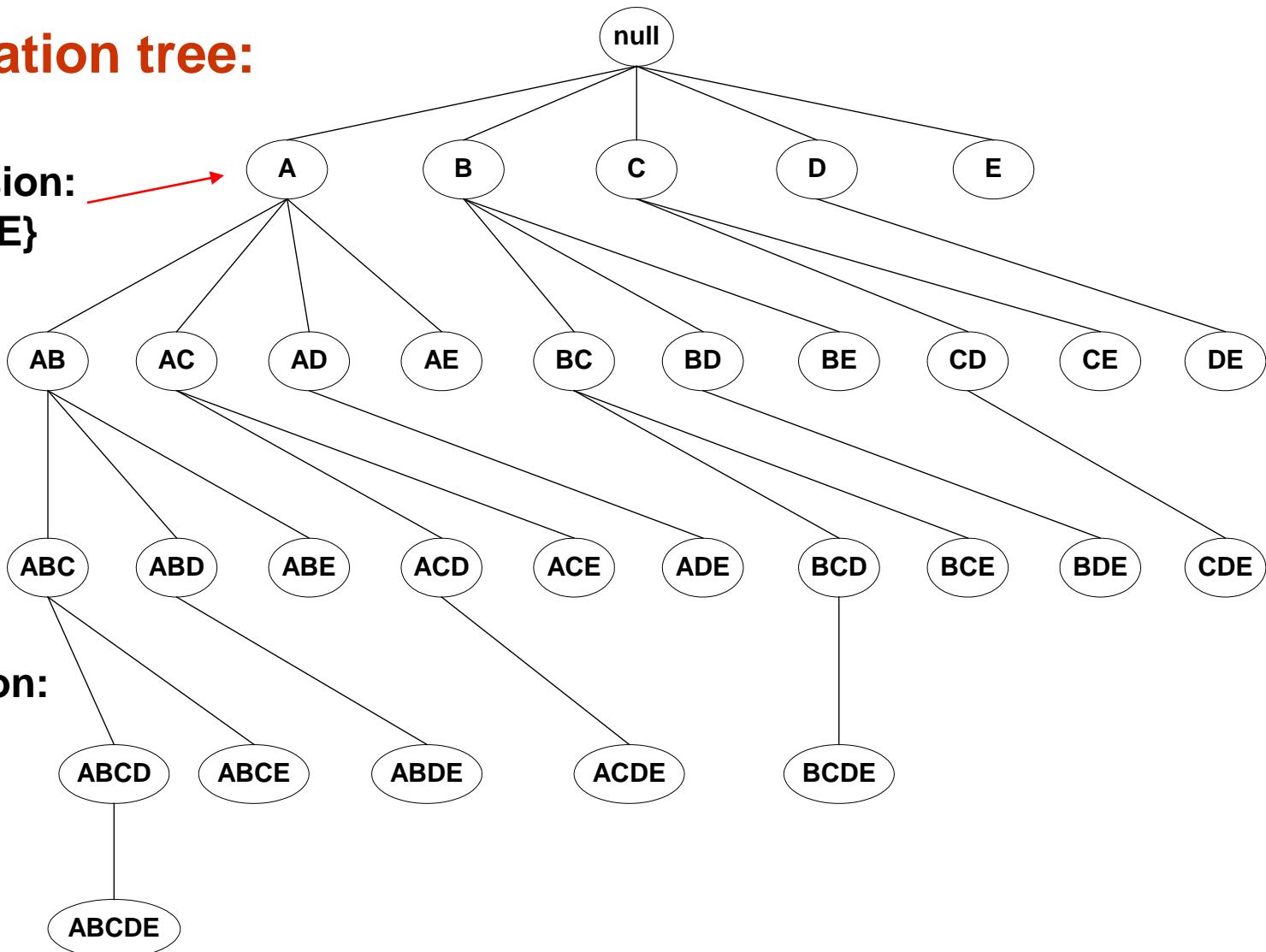
**Frequent Itemsets found
(with sup > 1):**

AD, BD, CD, ACD, BCD

Tree Projection

Set enumeration tree:

Possible Extension:
 $E(A) = \{B, C, D, E\}$



Possible Extension:
 $E(ABC) = \{D, E\}$

Tree Projection

- Items are listed in lexicographic order
- Each node P stores the following information:
 - Itemset for node P
 - List of possible lexicographic extensions of P: $E(P)$
 - Pointer to projected database of its ancestor node
 - Bitvector containing information about which transactions in the projected database contain the itemset

Projected Database

Original Database:

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Projected Database
for node A:

TID	Items
1	{B}
2	{}
3	{C,D,E}
4	{D,E}
5	{B,C}
6	{B,C,D}
7	{}
8	{B,C}
9	{B,D}
10	{}

For each transaction T, projected transaction at node A is $T \cap E(A)$

ECLAT

- For each item, store a list of transaction ids (tids)

Horizontal
Data Layout

TID	Items
1	A,B,E
2	B,C,D
3	C,E
4	A,C,D
5	A,B,C,D
6	A,E
7	A,B
8	A,B,C
9	A,C,D
10	B

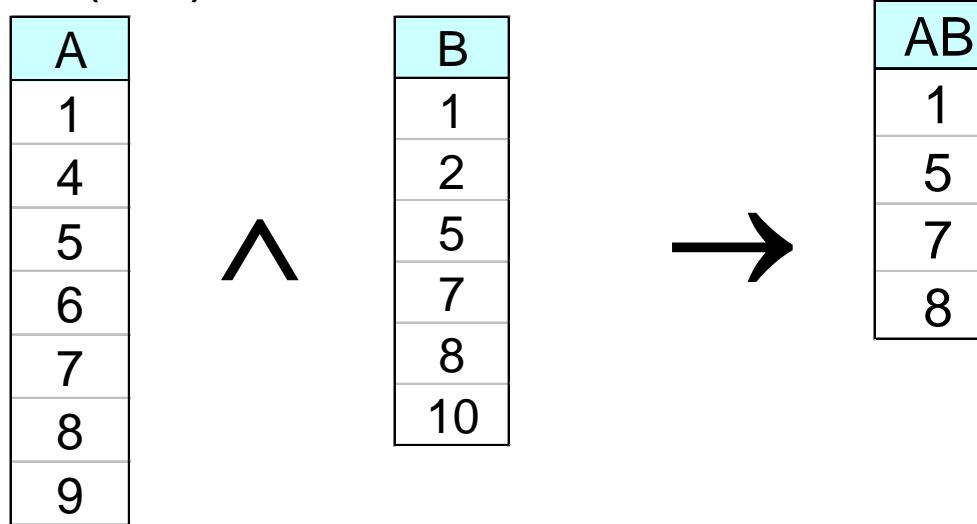
Vertical Data Layout

A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

TID-list

ECLAT

- Determine support of any k-itemset by intersecting tid-lists of two of its (k-1) subsets.



- 3 traversal approaches:
 - top-down, bottom-up and hybrid
- Advantage: very fast support counting
- Disadvantage: intermediate tid-lists may become too large for memory

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:

$ABC \rightarrow D,$	$ABD \rightarrow C,$	$ACD \rightarrow B,$	$BCD \rightarrow A,$
$A \rightarrow BCD,$	$B \rightarrow ACD,$	$C \rightarrow ABD,$	$D \rightarrow ABC$
$AB \rightarrow CD,$	$AC \rightarrow BD,$	$AD \rightarrow BC,$	$BC \rightarrow AD,$
$BD \rightarrow AC,$	$CD \rightarrow AB,$		
- If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$)

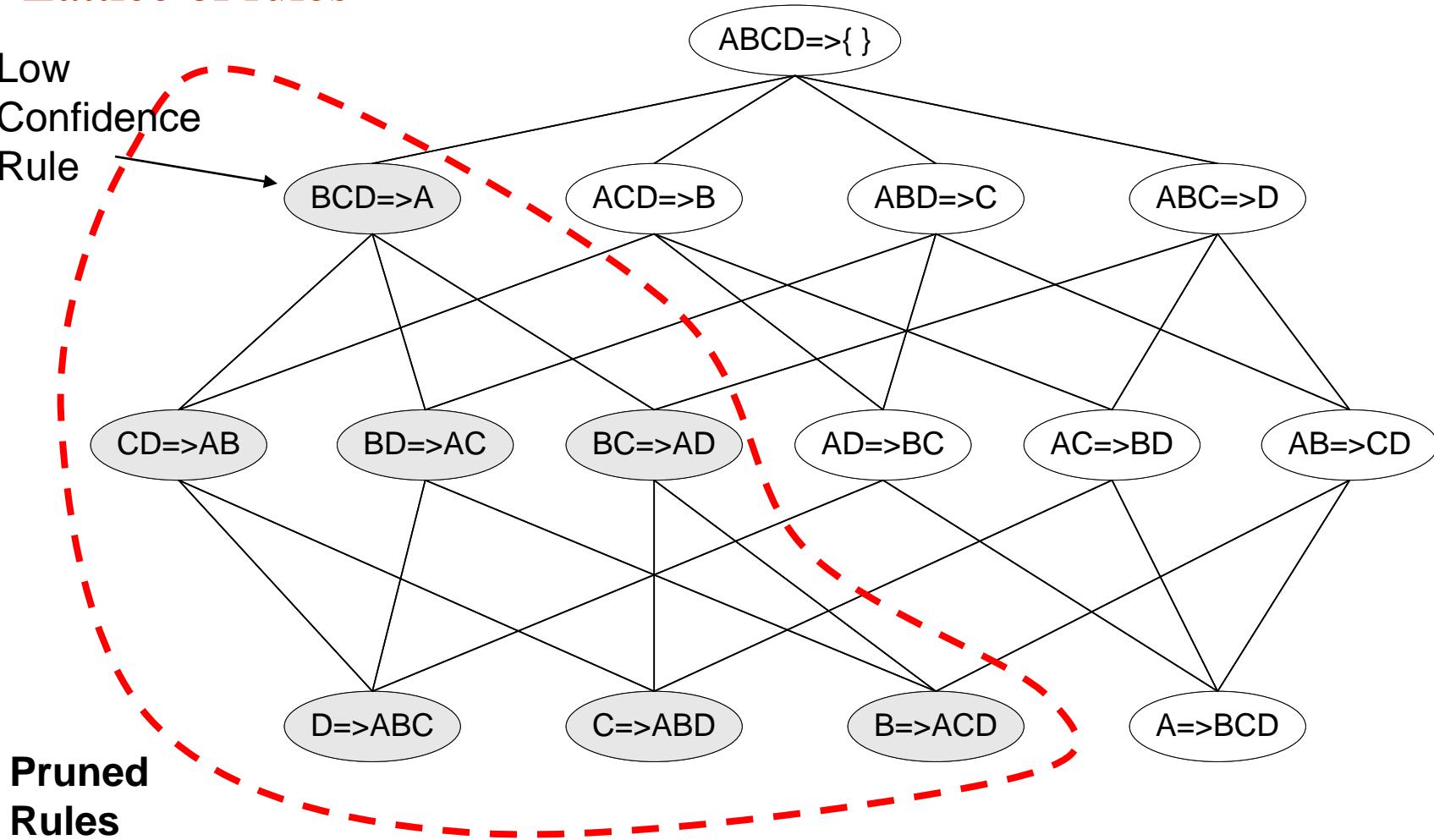
Rule Generation

- 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 w.r.t. number of items on the RHS of the rule

Rule Generation for Apriori Algorithm

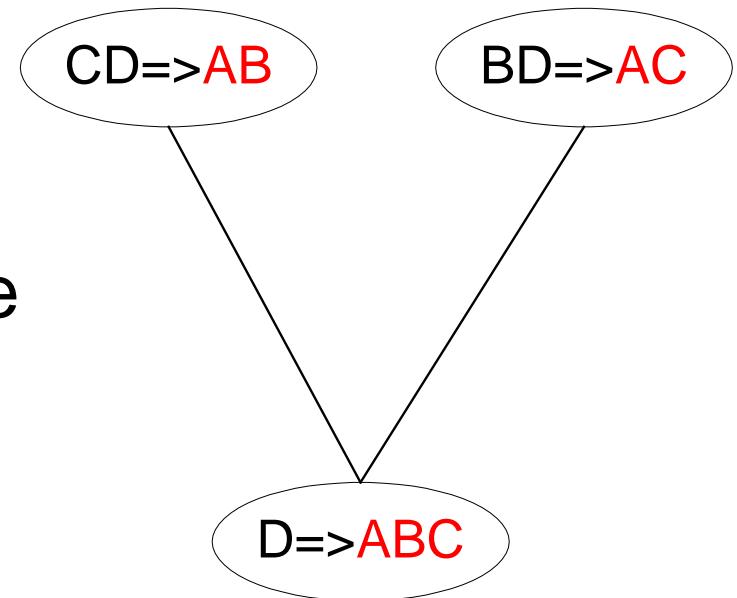
Lattice of rules

Low
Confidence
Rule



Rule Generation for Apriori Algorithm

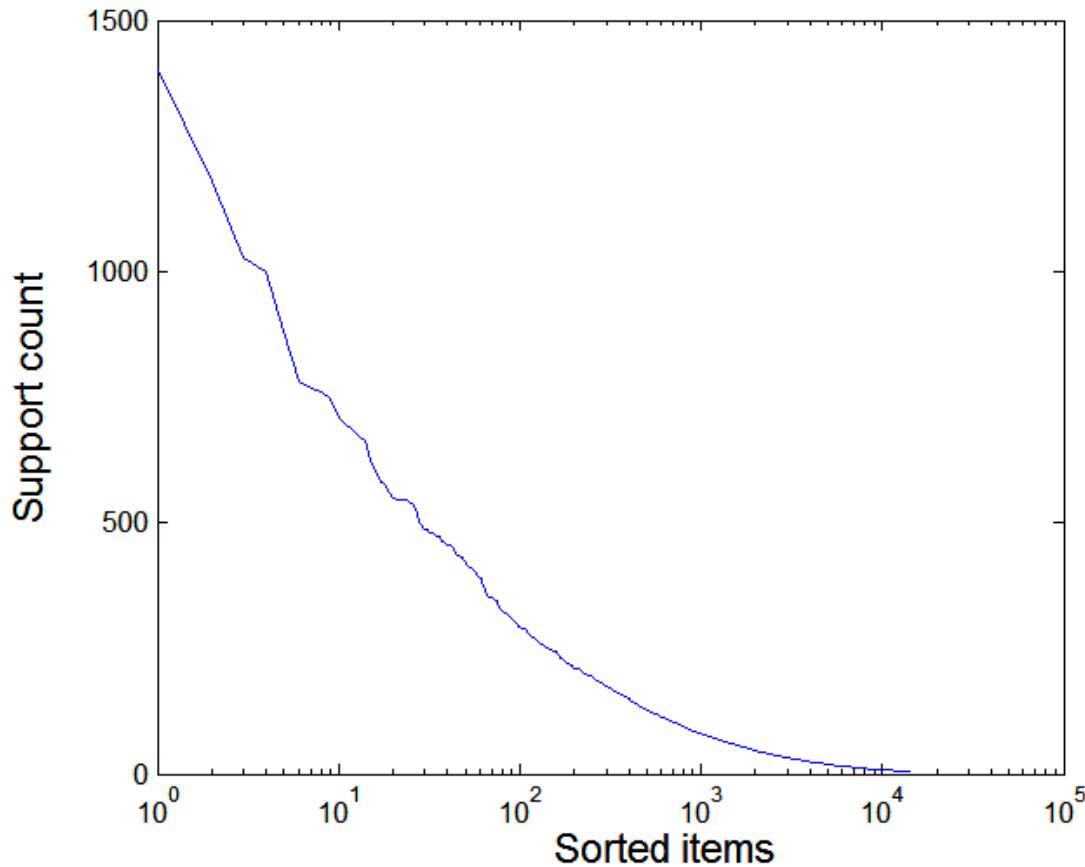
- Candidate rule is generated by merging two rules that share the same prefix in the rule consequent
- $\text{join}(\text{CD} \Rightarrow \text{AB}, \text{BD} \Rightarrow \text{AC})$ would produce the candidate rule $\text{D} \Rightarrow \text{ABC}$
- Prune rule $\text{D} \Rightarrow \text{ABC}$ if its 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

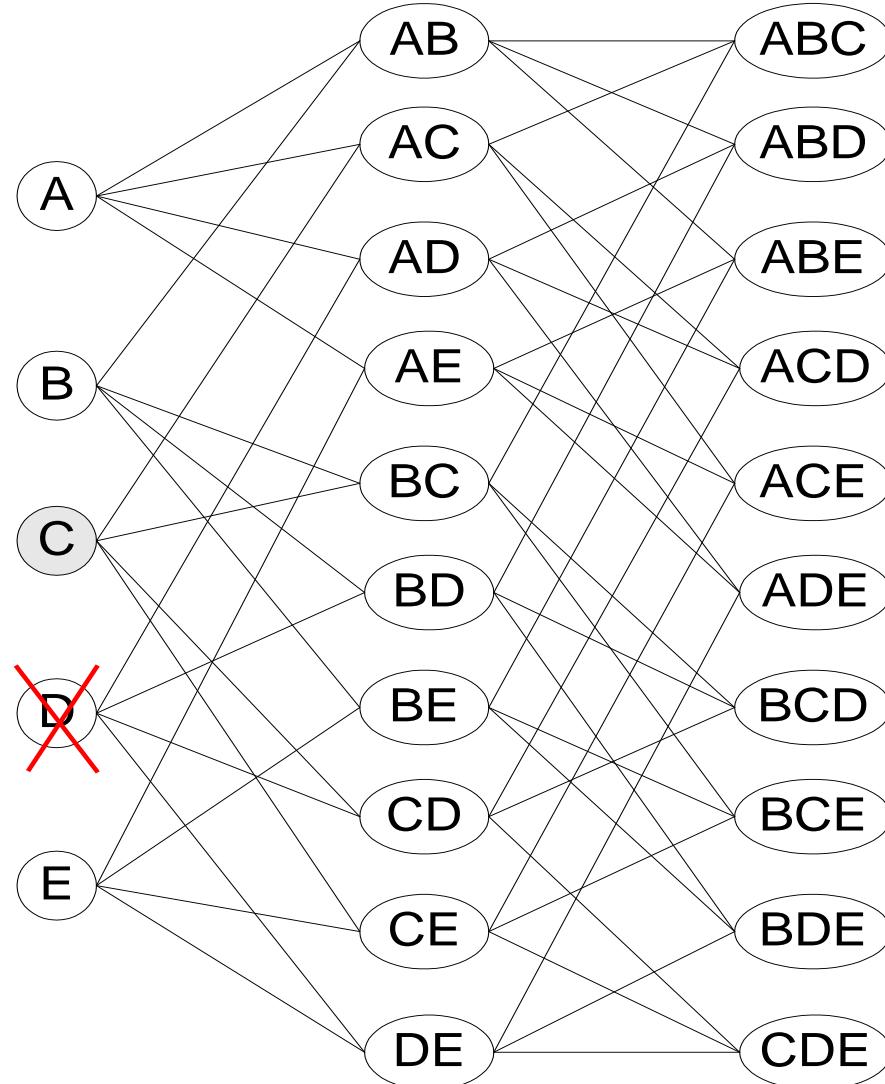
- How to set the appropriate *minsup* threshold?
 - If *minsup* is set too high, we could miss itemsets involving interesting rare items (e.g., expensive products)
 - If *minsup* is set too low, it is computationally expensive and the number of itemsets is very large
- Using a single minimum support 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

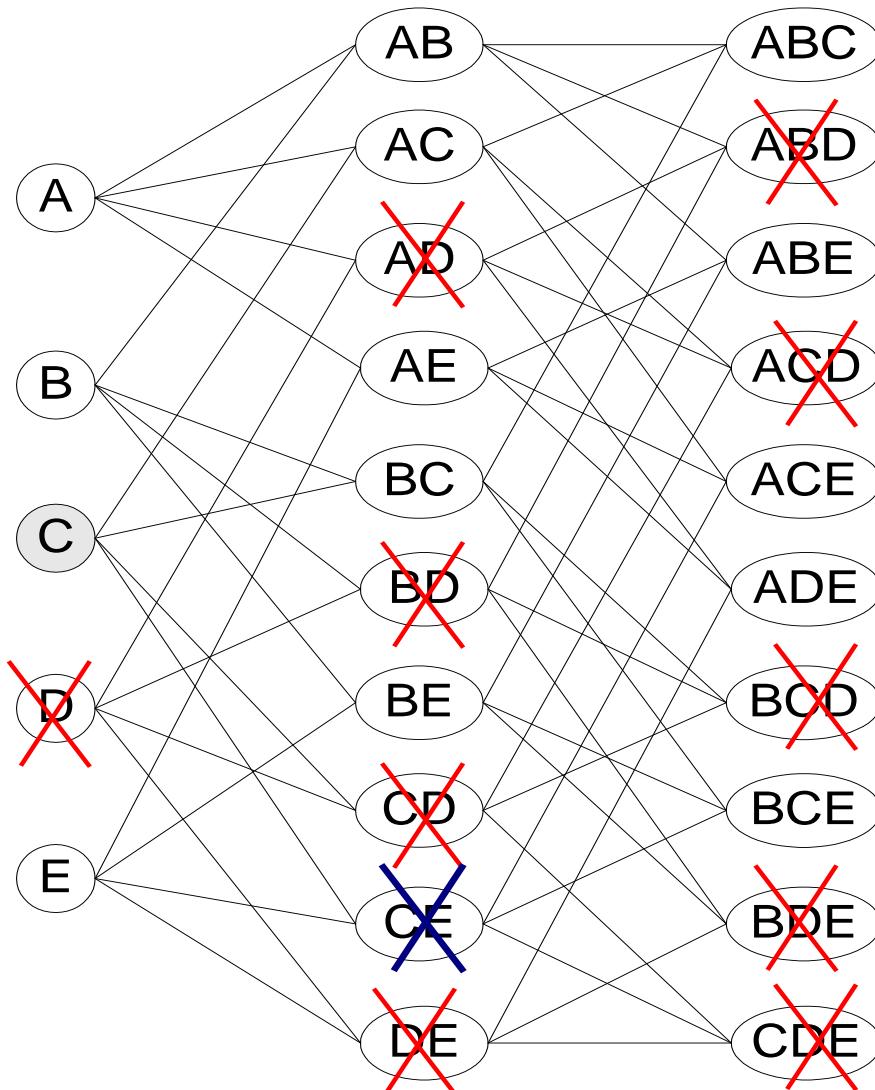
Multiple Minimum Support

Item	MS(I)	Sup(I)
A	0.10%	0.25%
B	0.20%	0.26%
C	0.30%	0.29%
D	0.50%	0.05%
E	3%	4.20%



Multiple Minimum Support

Item	MS(I)	Sup(I)
A	0.10%	0.25%
B	0.20%	0.26%
C	0.30%	0.29%
D	0.50%	0.05%
E	3%	4.20%



Multiple Minimum Support (Liu 1999)

- Order the items according to their minimum support (in ascending order)
 - e.g.: $MS(\text{Milk})=5\%$, $MS(\text{Coke}) = 3\%$,
 $MS(\text{Broccoli})=0.1\%$, $MS(\text{Salmon})=0.5\%$
 - Ordering: Broccoli, Salmon, Coke, Milk
- Need to modify Apriori such that:
 - L_1 : set of frequent items
 - F_1 : set of items whose support is $\geq MS(1)$
where $MS(1)$ is $\min_i(MS(i))$
 - C_2 : candidate itemsets of size 2 is generated from F_1 instead of L_1

Multiple Minimum Support (Liu 1999)

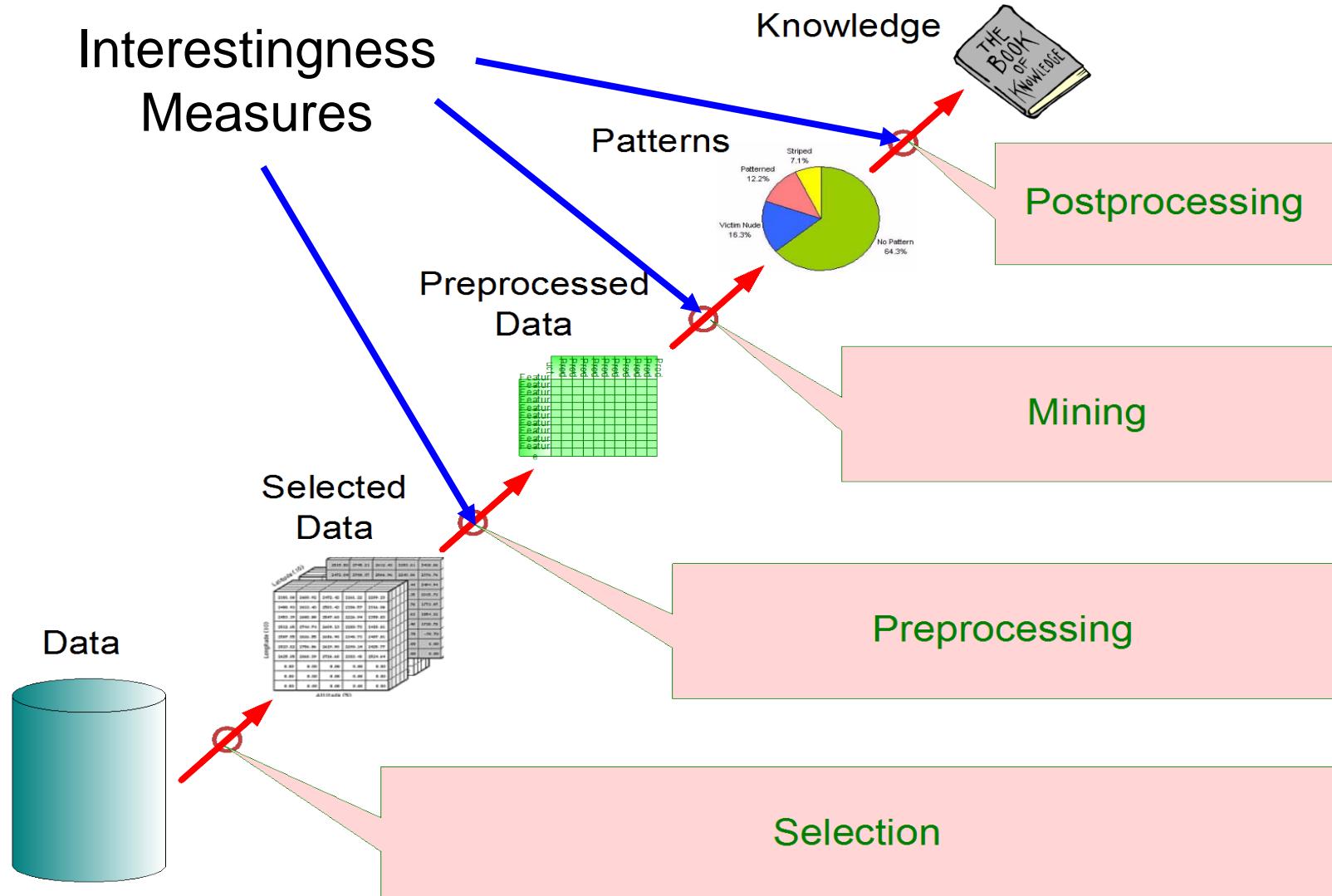
- Modifications to Apriori:

- In traditional Apriori,
 - ◆ A candidate $(k+1)$ -itemset is generated by merging two frequent itemsets of size k
 - ◆ The candidate is pruned if it contains any infrequent subsets of size k
- Pruning step has to be modified:
 - ◆ Prune only if subset contains the first item
 - ◆ e.g.: Candidate={Broccoli, Coke, Milk} (ordered according to minimum support)
 - ◆ {Broccoli, Coke} and {Broccoli, Milk} are frequent but {Coke, Milk} is infrequent
 - Candidate is not pruned because {Coke,Milk} does not contain the first item, i.e., Broccoli.

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

Application of Interestingness Measure



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

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

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

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

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

$$PS = P(X, Y) - P(X)P(Y)$$

$$\phi\text{-coefficient} = \frac{P(X, Y) - P(X)P(Y)}{\sqrt{P(X)[1 - P(X)]P(Y)[1 - P(Y)]}}$$

Example: Lift/Interest

	Coffee	—	Coffee
Tea	15	5	20
—	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 $P(X,Y)=P(X)P(Y)$ => Lift = 1

There are lots of measures proposed in the literature

Some measures are good for certain applications, but not for others

What criteria should we use to determine whether a measure is good or bad?

What about Apriori-style support based pruning? How does it affect these measures?

#	Measure	Formula
1	ϕ -coefficient	$\frac{P(A,B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j, B_k) + \sum_k \max_j P(A_j, B_k) - \max_j P(A_j) - \max_k P(B_k)}{2 - \max_j P(A_j) - \max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\bar{A},\bar{B})}{P(\bar{A},B)P(\bar{A},\bar{B})}$
4	Yule's Q	$\frac{P(A,B)P(\bar{A}\bar{B}) - P(A,\bar{B})P(\bar{A},B)}{P(A,B)P(\bar{A}\bar{B}) + P(A,\bar{B})P(\bar{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\bar{A}\bar{B})} - \sqrt{P(A,\bar{B})P(\bar{A},B)}}{\sqrt{P(A,B)P(\bar{A}\bar{B})} + \sqrt{P(A,\bar{B})P(\bar{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)P(\bar{A},\bar{B}) - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}$
7	Mutual Information (M)	$\sum_i \sum_j P(A_i, B_j) \log \frac{P(A_i, B_j)}{P(A_i)P(B_j)}$
8	J-Measure (J)	$\max \left(P(A,B) \log \left(\frac{P(B A)}{P(B)} \right) + P(\bar{A}\bar{B}) \log \left(\frac{P(\bar{B} \bar{A})}{P(\bar{B})} \right), P(A,B) \log \left(\frac{P(A B)}{P(A)} \right) + P(\bar{A}\bar{B}) \log \left(\frac{P(\bar{A} B)}{P(\bar{A})} \right) \right)$
9	Gini index (G)	$\max \left(P(A)[P(B A)^2 + P(\bar{B} A)^2] + P(\bar{A})[P(B \bar{A})^2 + P(\bar{B} \bar{A})^2] - P(B)^2 - P(\bar{B})^2, P(B)[P(A B)^2 + P(\bar{A} B)^2] + P(\bar{B})[P(A \bar{B})^2 + P(\bar{A} \bar{B})^2] - P(A)^2 - P(\bar{A})^2 \right)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B A), P(A B))$
12	Laplace (L)	$\max \left(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2} \right)$
13	Conviction (V)	$\max \left(\frac{P(A)P(\bar{B})}{P(A\bar{B})}, \frac{P(B)P(\bar{A})}{P(B\bar{A})} \right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B) - P(A)P(B)$
17	Certainty factor (F)	$\max \left(\frac{P(B A) - P(B)}{1 - P(B)}, \frac{P(A B) - P(A)}{1 - P(A)} \right)$
18	Added Value (AV)	$\max(P(B A) - P(B), P(A B) - P(A))$
19	Collective strength (S)	$\frac{P(A,B) + P(\bar{A}\bar{B})}{P(A)P(B) + P(\bar{A})P(\bar{B})} \times \frac{1 - P(A)P(B) - P(\bar{A})P(\bar{B})}{1 - P(A,B) - P(\bar{A}\bar{B})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A) + P(B) - P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)} \max(P(B A) - P(B), P(A B) - P(A))$

Properties of A Good Measure

- **Piatetsky-Shapiro:**

3 properties a good measure M must satisfy:

- $M(A,B) = 0$ if A and B are statistically independent
- $M(A,B)$ increase monotonically with $P(A,B)$ when $P(A)$ and $P(B)$ remain unchanged
- $M(A,B)$ decreases monotonically with $P(A)$ [or $P(B)$] when $P(A,B)$ and $P(B)$ [or $P(A)$] remain unchanged

Comparing Different Measures

10 examples of contingency tables:

Rankings of contingency tables using various measures:

Example	f_{11}	f_{10}	f_{01}	f_{00}
E1	8123	83	424	1370
E2	8330	2	622	1046
E3	9481	94	127	298
E4	3954	3080	5	2961
E5	2886	1363	1320	4431
E6	1500	2000	500	6000
E7	4000	2000	1000	3000
E8	4000	2000	2000	2000
E9	1720	7121	5	1154
E10	61	2483	4	7452

#	ϕ	λ	α	Q	Y	κ	M	J	G	s	c	L	V	I	IS	PS	F	AV	S	ζ	K
E1	1	1	3	3	3	1	2	2	1	3	5	5	4	6	2	2	4	6	1	2	5
E2	2	2	1	1	1	2	1	3	2	2	1	1	1	8	3	5	1	8	2	3	6
E3	3	3	4	4	4	3	3	8	7	1	4	4	6	10	1	8	6	10	3	1	10
E4	4	7	2	2	2	5	4	1	3	6	2	2	2	4	4	1	2	3	4	5	1
E5	5	4	8	8	8	4	7	5	4	7	9	9	9	3	6	3	9	4	5	6	3
E6	6	6	7	7	7	7	6	4	6	9	8	8	7	2	8	6	7	2	7	8	2
E7	7	5	9	9	9	6	8	6	5	4	7	7	8	5	5	4	8	5	6	4	4
E8	8	9	10	10	10	8	10	10	8	4	10	10	10	9	7	7	10	9	8	7	9
E9	9	9	5	5	5	9	9	7	9	8	3	3	3	7	9	9	3	7	9	9	8
E10	10	8	6	6	6	10	5	9	10	10	6	6	5	1	10	10	5	1	10	10	7

Property under Variable Permutation

	B	\bar{B}
A	p	q
\bar{A}	r	s

	A	\bar{A}
B	p	r
\bar{B}	q	s

Does $M(A,B) = M(B,A)$?

Symmetric measures:

- ◆ support, lift, collective strength, cosine, Jaccard, etc

Asymmetric measures:

- ◆ confidence, conviction, Laplace, J-measure, etc

Property under Row/Column Scaling

Grade-Gender Example (Mosteller, 1968):

	Male	Female	
High	2	3	5
Low	1	4	5
	3	7	10

	Male	Female	
High	4	30	34
Low	2	40	42
	6	70	76

$$\begin{array}{cc} \downarrow & \downarrow \\ 2x & 10x \end{array}$$

Mosteller:

Underlying association should be independent of the relative number of male and female students in the samples

Property under Inversion Operation

Example: ϕ -Coefficient

- ϕ -coefficient is analogous to correlation coefficient for continuous variables

	Y	\bar{Y}	
X	60	10	70
\bar{X}	10	20	30
	70	30	100

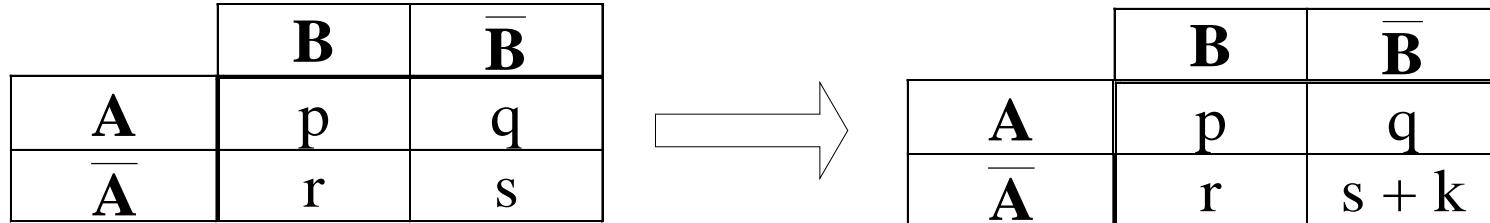
	Y	\bar{Y}	
X	20	10	30
\bar{X}	10	60	70
	30	70	100

$$\begin{aligned}\phi &= \frac{0.6 - 0.7 \times 0.7}{\sqrt{0.7 \times 0.3 \times 0.7 \times 0.3}} \\ &= 0.5238\end{aligned}$$

$$\begin{aligned}\phi &= \frac{0.2 - 0.3 \times 0.3}{\sqrt{0.7 \times 0.3 \times 0.7 \times 0.3}} \\ &= 0.5238\end{aligned}$$

ϕ Coefficient is the same for both tables

Property under Null Addition



The diagram illustrates a transformation of a 2x2 contingency table. On the left, a table with columns labeled **B** and \bar{B} , and rows labeled **A** and \bar{A} , contains values p, q, r, and s. An arrow points to the right, where the same table structure is shown but with the value s replaced by s + k.

	B	\bar{B}
A	p	q
\bar{A}	r	s

→

	B	\bar{B}
A	p	q
\bar{A}	r	$s + k$

Invariant measures:

- ◆ support, cosine, Jaccard, etc

Non-invariant measures:

- ◆ correlation, Gini, mutual information, odds ratio, etc

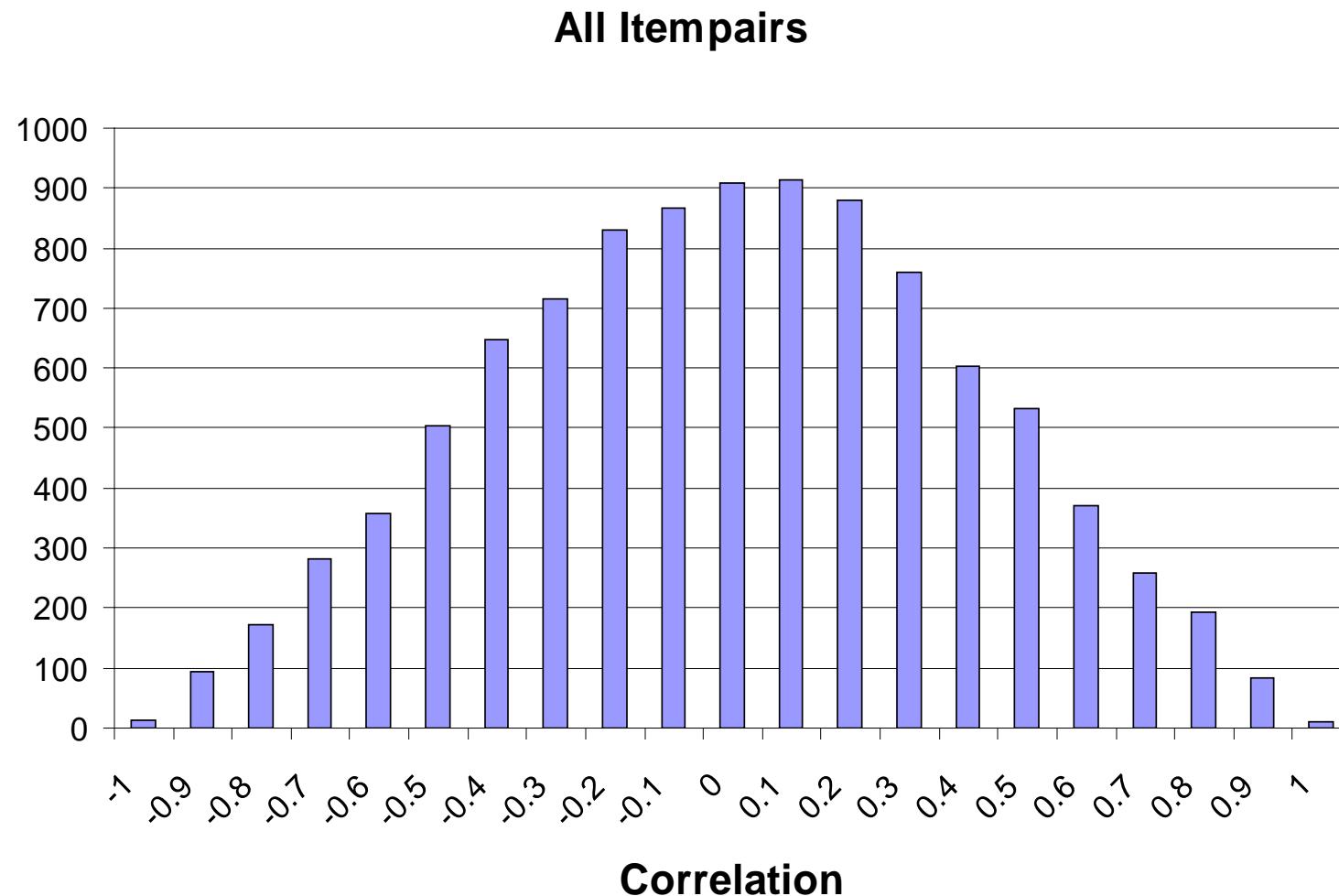
Different Measures have Different Properties

Symbol	Measure	Range	P1	P2	P3	O1	O2	O3	O3'	O4
Φ	Correlation	-1 ... 0 ... 1	Yes	Yes	Yes	Yes	No	Yes	Yes	No
λ	Lambda	0 ... 1	Yes	No	No	Yes	No	No*	Yes	No
α	Odds ratio	0 ... 1 ... ∞	Yes*	Yes	Yes	Yes	Yes	Yes*	Yes	No
Q	Yule's Q	-1 ... 0 ... 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Y	Yule's Y	-1 ... 0 ... 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
κ	Cohen's	-1 ... 0 ... 1	Yes	Yes	Yes	Yes	No	No	Yes	No
M	Mutual Information	0 ... 1	Yes	Yes	Yes	Yes	No	No*	Yes	No
J	J-Measure	0 ... 1	Yes	No	No	No	No	No	No	No
G	Gini Index	0 ... 1	Yes	No	No	No	No	No*	Yes	No
S	Support	0 ... 1	No	Yes	No	Yes	No	No	No	No
c	Confidence	0 ... 1	No	Yes	No	Yes	No	No	No	Yes
L	Laplace	0 ... 1	No	Yes	No	Yes	No	No	No	No
V	Conviction	0.5 ... 1 ... ∞	No	Yes	No	Yes**	No	No	Yes	No
I	Interest	0 ... 1 ... ∞	Yes*	Yes	Yes	Yes	No	No	No	No
IS	IS (cosine)	0 .. 1	No	Yes	Yes	Yes	No	No	No	Yes
PS	Piatetsky-Shapiro's	-0.25 ... 0 ... 0.25	Yes	Yes	Yes	Yes	No	Yes	Yes	No
F	Certainty factor	-1 ... 0 ... 1	Yes	Yes	Yes	No	No	No	Yes	No
AV	Added value	0.5 ... 1 ... 1	Yes	Yes	Yes	No	No	No	No	No
S	Collective strength	0 ... 1 ... ∞	No	Yes	Yes	Yes	No	Yes*	Yes	No
ζ	Jaccard	0 .. 1	No	Yes	Yes	Yes	No	No	No	Yes
K	Klosgen's	$\left(\sqrt{\frac{2}{\sqrt{3}}}-1\right)\left(2-\sqrt{3}-\frac{1}{\sqrt{3}}\right) \dots 0 \dots \frac{2}{3\sqrt{3}}$	Yes	Yes	Yes	No	No	No	No	No

Support-based Pruning

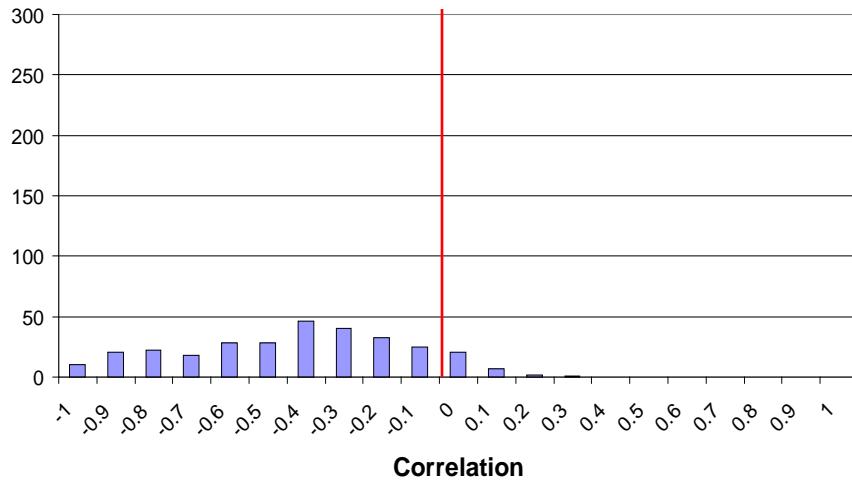
- Most of the association rule mining algorithms use support measure to prune rules and itemsets
- Study effect of support pruning on correlation of itemsets
 - Generate 10000 random contingency tables
 - Compute support and pairwise correlation for each table
 - Apply support-based pruning and examine the tables that are removed

Effect of Support-based Pruning

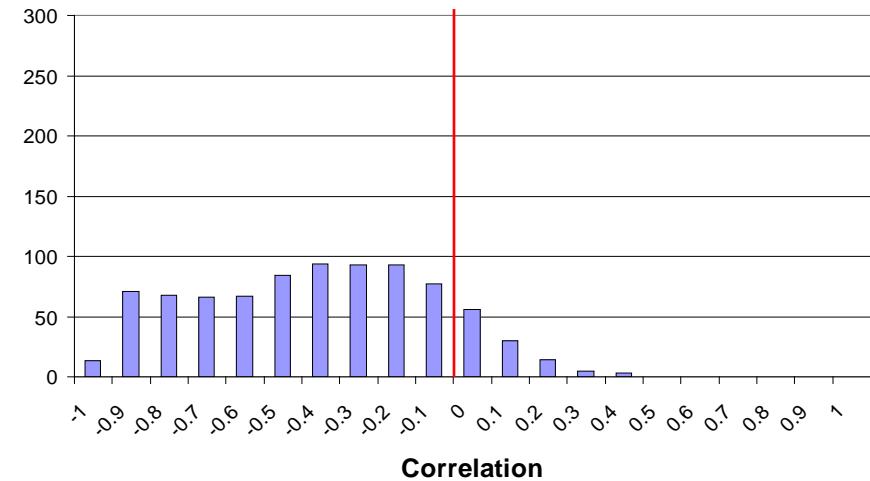


Effect of Support-based Pruning

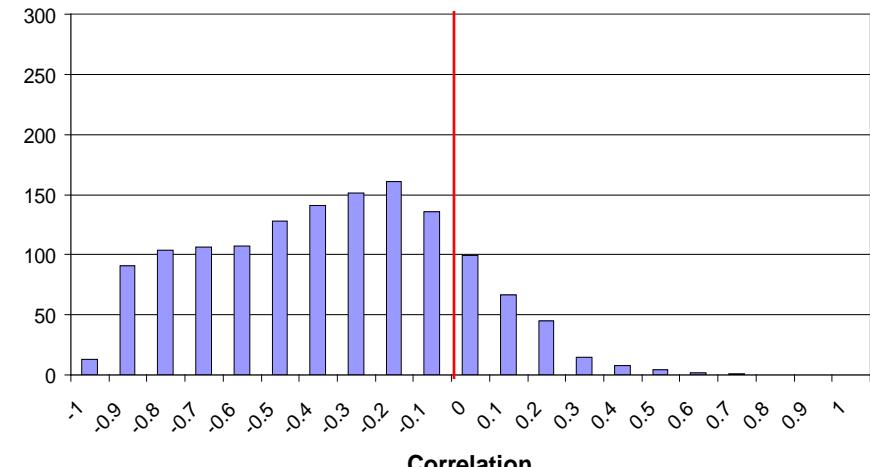
Support < 0.01



Support < 0.03



Support < 0.05



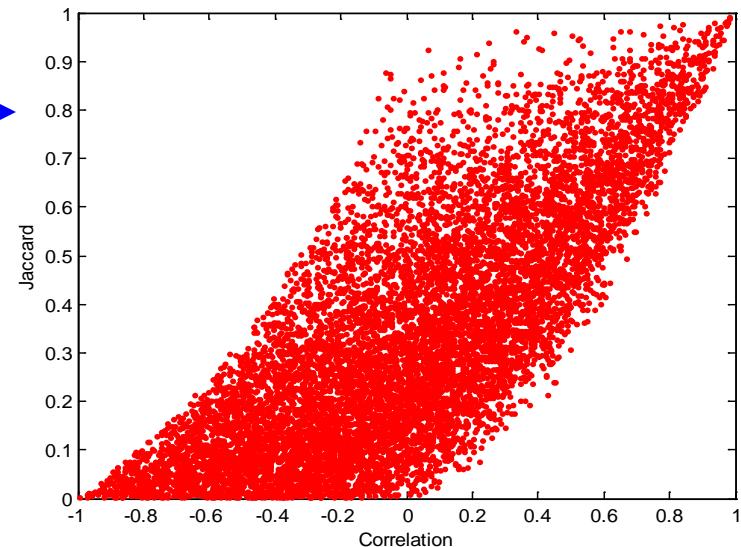
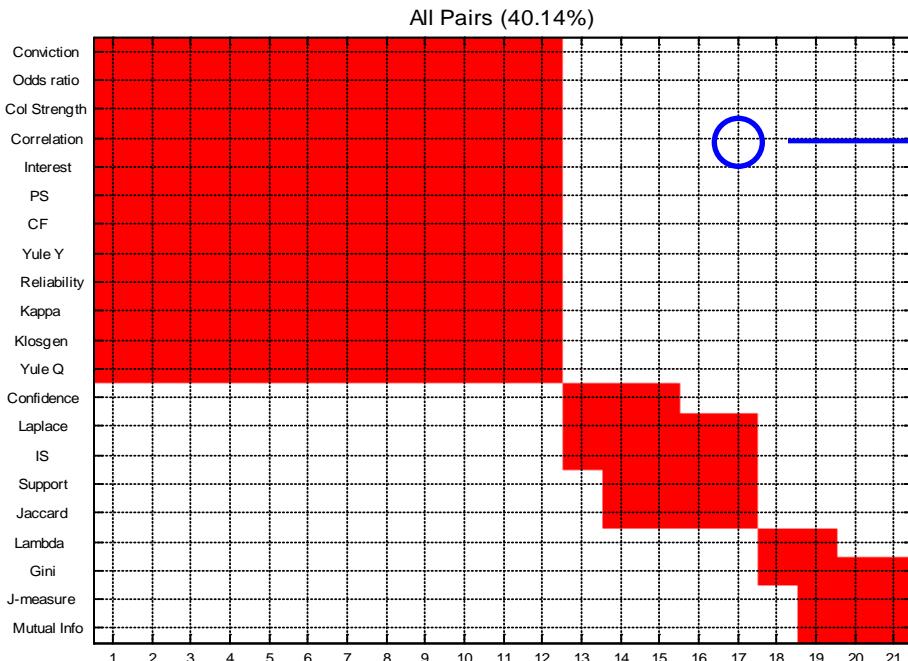
Support-based pruning
eliminates mostly
negatively correlated
itemsets

Effect of Support-based Pruning

- Investigate how support-based pruning affects other measures
- Steps:
 - Generate 10000 contingency tables
 - Rank each table according to the different measures
 - Compute the pair-wise correlation between the measures

Effect of Support-based Pruning

◆ Without Support Pruning (All Pairs)

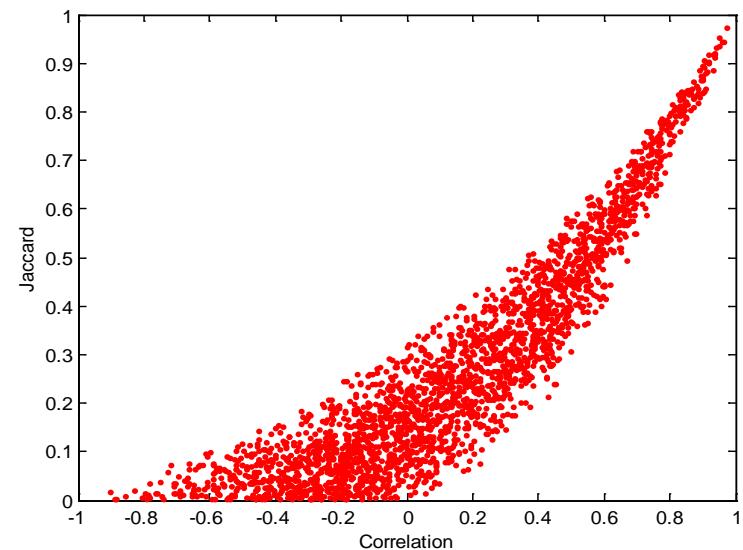
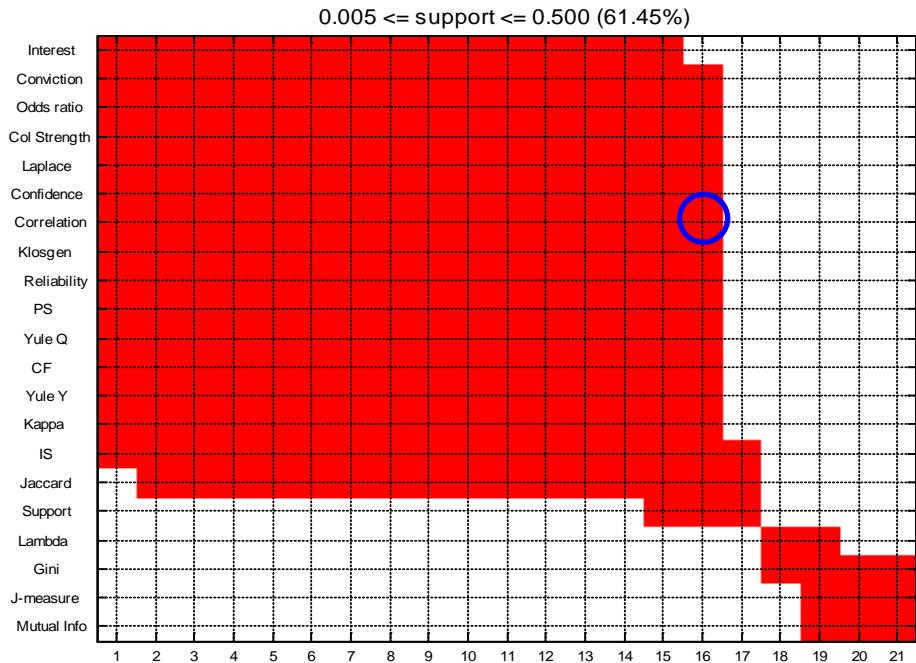


Scatter Plot between Correlation & Jaccard Measure

- ◆ Red cells indicate correlation between the pair of measures > 0.85
- ◆ 40.14% pairs have correlation > 0.85

Effect of Support-based Pruning

- ◆ $0.5\% \leq \text{support} \leq 50\%$

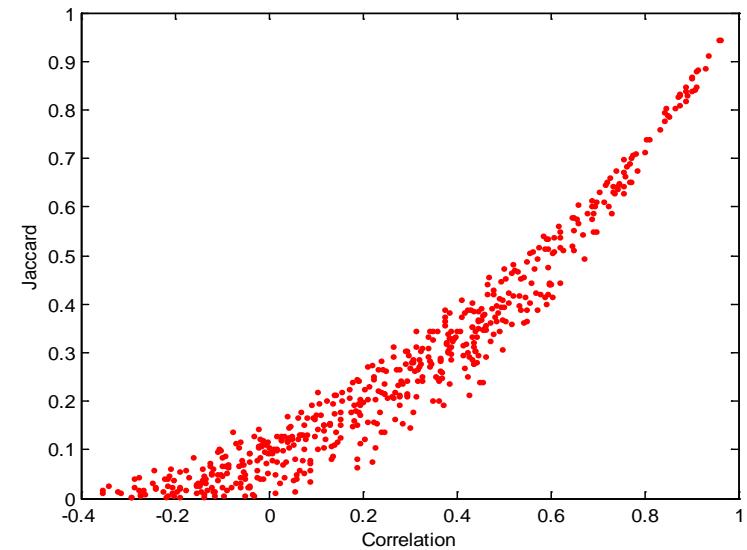
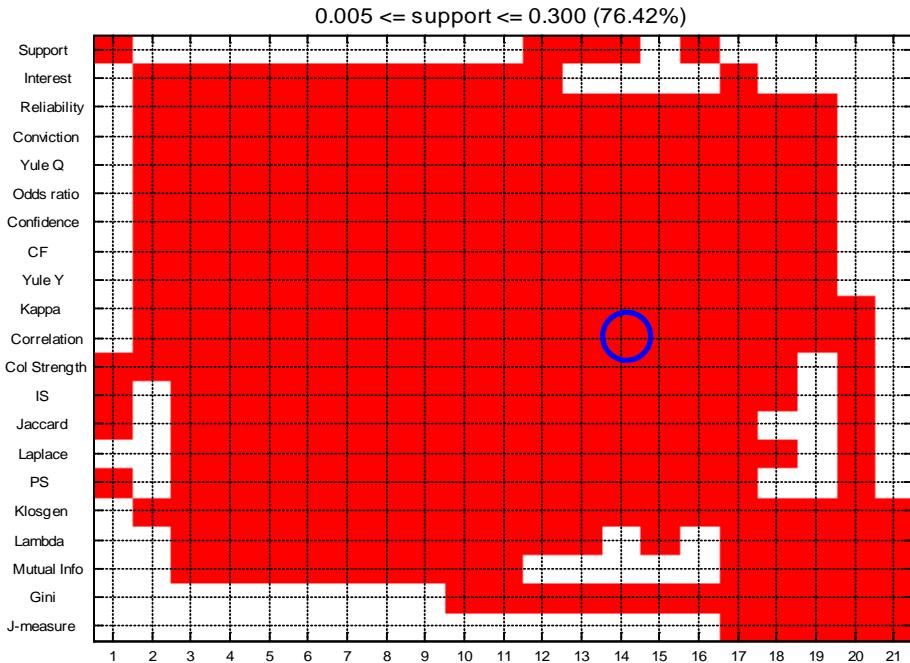


Scatter Plot between Correlation & Jaccard Measure:

- ◆ 61.45% pairs have correlation > 0.85

Effect of Support-based Pruning

- ◆ $0.5\% \leq \text{support} \leq 30\%$



Scatter Plot between Correlation & Jaccard Measure

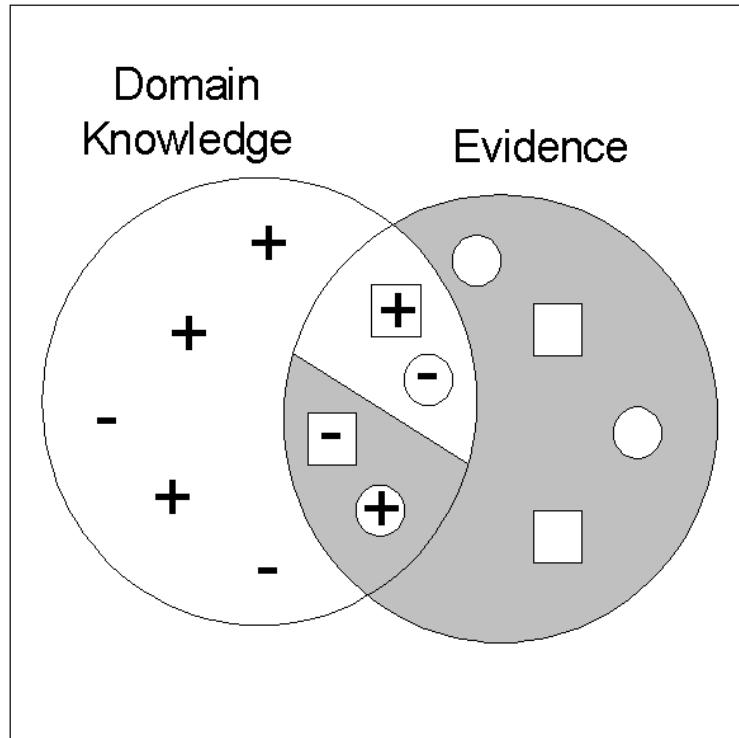
- ◆ 76.42% pairs have correlation > 0.85

Subjective Interestingness Measure

- Objective measure:
 - Rank patterns based on statistics computed from data
 - e.g., 21 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 (Silberschatz & Tuzhilin)
 - ◆ A pattern is subjectively interesting if it is actionable (Silberschatz & Tuzhilin)

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)

Interestingness via Unexpectedness

- Web Data (Cooley et al 2001)
 - Domain knowledge in the form of site structure
 - Given an itemset $F = \{X_1, X_2, \dots, X_k\}$ (X_i : Web pages)
 - ◆ L: number of links connecting the pages
 - ◆ Ifactor = $L / (k \times k-1)$
 - ◆ cfactor = 1 (if graph is connected), 0 (disconnected graph)
 - Structure evidence = cfactor \times Ifactor
 - Usage evidence =
$$\frac{P(X_1 \cap X_2 \cap \dots \cap X_k)}{P(X_1 \cup X_2 \cup \dots \cup X_k)}$$
 - Use Dempster-Shafer theory to combine domain knowledge and evidence from data

Data Mining Association Rules: Advanced Concepts and Algorithms

Lecture Notes for Chapter 7

Introduction to Data Mining

by

Tan, Steinbach, Kumar

Continuous and Categorical Attributes

How to apply association analysis formulation to non-asymmetric binary variables?

Session Id	Country	Session Length (sec)	Number of Web Pages viewed	Gender	Browser Type	Buy
1	USA	982	8	Male	IE	No
2	China	811	10	Female	Netscape	No
3	USA	2125	45	Female	Mozilla	Yes
4	Germany	596	4	Male	IE	Yes
5	Australia	123	9	Male	Mozilla	No
...

Example of Association Rule:

$$\{\text{Number of Pages } \in [5,10] \wedge (\text{Browser}=\text{Mozilla})\} \rightarrow \{\text{Buy} = \text{No}\}$$

Handling Categorical Attributes

- Transform categorical attribute into asymmetric binary variables
- Introduce a new “item” for each distinct attribute-value pair
 - Example: replace Browser Type attribute with
 - ◆ Browser Type = Internet Explorer
 - ◆ Browser Type = Mozilla
 - ◆ Browser Type = Mozilla

Handling Categorical Attributes

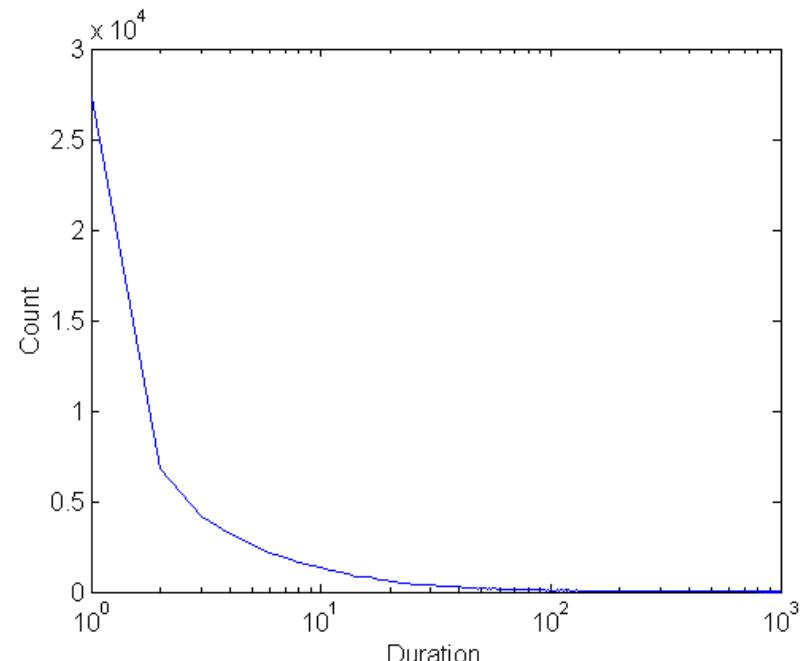
- Potential Issues
 - What if attribute has many possible values
 - ◆ Example: attribute country has more than 200 possible values
 - ◆ Many of the attribute values may have very low support
 - Potential solution: Aggregate the low-support attribute values
 - What if distribution of attribute values is highly skewed
 - ◆ Example: 95% of the visitors have Buy = No
 - ◆ Most of the items will be associated with (Buy=No) item
 - Potential solution: drop the highly frequent items

Handling Continuous Attributes

- Different kinds of rules:
 - $\text{Age} \in [21,35] \wedge \text{Salary} \in [70k,120k] \rightarrow \text{Buy}$
 - $\text{Salary} \in [70k,120k] \wedge \text{Buy} \rightarrow \text{Age: } \mu=28, \sigma=4$
- Different methods:
 - Discretization-based
 - Statistics-based
 - Non-discretization based
 - ◆ minApriori

Handling Continuous Attributes

- Use discretization
- Unsupervised:
 - Equal-width binning
 - Equal-depth binning
 - Clustering
- Supervised:



Attribute values, v

Class	v ₁	v ₂	v ₃	v ₄	v ₅	v ₆	v ₇	v ₈	v ₉
Anomalous	0	0	20	10	20	0	0	0	0
Normal	150	100	0	0	0	100	100	150	100

$\brace{v_1, v_2, v_3}$ $\brace{v_4, v_5}$ $\brace{v_6, v_7, v_8, v_9}$

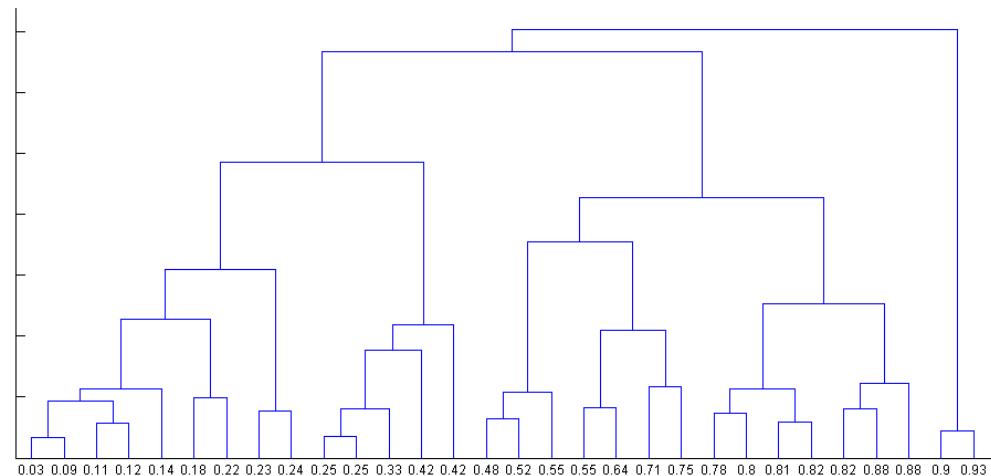
bin₁ bin₂ bin₃

Discretization Issues

- Size of the discretized intervals affect support & confidence
 - {Refund = No, (Income = \$51,250)} → {Cheat = No}
 - {Refund = No, (60K ≤ Income ≤ 80K)} → {Cheat = No}
 - {Refund = No, (0K ≤ Income ≤ 1B)} → {Cheat = No}
- If intervals too small
 - ◆ may not have enough support
- If intervals too large
 - ◆ may not have enough confidence
- Potential solution: use all possible intervals

Discretization Issues

- Execution time
 - If intervals contain n values, there are on average $O(n^2)$ possible ranges



- Too many rules

{Refund = No, (Income = \$51,250)} \rightarrow {Cheat = No}

{Refund = No, (51K \leq Income \leq 52K)} \rightarrow {Cheat = No}

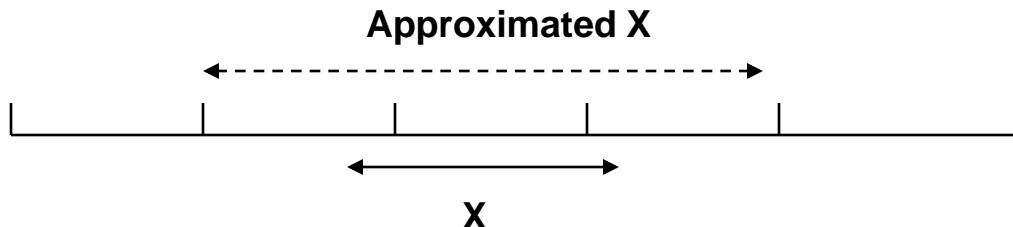
{Refund = No, (50K \leq Income \leq 60K)} \rightarrow {Cheat = No}

Approach by Srikant & Agrawal

- Preprocess the data
 - Discretize attribute using equi-depth partitioning
 - ◆ Use *partial completeness measure* to determine number of partitions
 - ◆ Merge adjacent intervals as long as support is less than max-support
- Apply existing association rule mining algorithms
- Determine interesting rules in the output

Approach by Srikant & Agrawal

- Discretization will lose information



- Use *partial completeness measure* to determine how much information is lost

C: frequent itemsets obtained by considering all ranges of attribute values
P: frequent itemsets obtained by considering all ranges over the partitions

P is *K-complete* w.r.t C if $P \subseteq C$, and $\forall X \in C, \exists X' \in P$ such that:

1. X' is a generalization of X and $\text{support}(X') \leq K \times \text{support}(X)$ ($K \geq 1$)
2. $\forall Y \subseteq X, \exists Y' \subseteq X'$ such that $\text{support}(Y') \leq K \times \text{support}(Y)$

Given *K* (*partial completeness level*), can determine number of intervals (N)

Interestingness Measure

$\{\text{Refund} = \text{No}, (\text{Income} = \$51,250)\} \rightarrow \{\text{Cheat} = \text{No}\}$

$\{\text{Refund} = \text{No}, (51\text{K} \leq \text{Income} \leq 52\text{K})\} \rightarrow \{\text{Cheat} = \text{No}\}$

$\{\text{Refund} = \text{No}, (50\text{K} \leq \text{Income} \leq 60\text{K})\} \rightarrow \{\text{Cheat} = \text{No}\}$

- Given an itemset: $Z = \{z_1, z_2, \dots, z_k\}$ and its generalization $Z' = \{z'_1, z'_2, \dots, z'_k\}$

$P(Z)$: support of Z

$E_{z'}(Z)$: expected support of Z based on Z'

$$E_{z'}(Z) = \frac{P(z_1)}{P(z'_1)} \times \frac{P(z_2)}{P(z'_2)} \times \cdots \times \frac{P(z_k)}{P(z'_k)} \times P(Z')$$

- Z is R -interesting w.r.t. Z' if $P(Z) \geq R \times E_{z'}(Z)$

Interestingness Measure

- For $S: X \rightarrow Y$, and its generalization $S': X' \rightarrow Y'$

$P(Y|X)$: confidence of $X \rightarrow Y$

$P(Y'|X')$: confidence of $X' \rightarrow Y'$

$E_{S'}(Y|X)$: expected support of Z based on Z'

$$E(Y | X) = \frac{P(y_1)}{P(y'_1)} \times \frac{P(y_2)}{P(y'_2)} \times \cdots \times \frac{P(y_k)}{P(y'_k)} \times P(Y' | X')$$

- Rule S is R -interesting w.r.t its ancestor rule S' if
 - Support, $P(S) \geq R \times E_{S'}(S)$ or
 - Confidence, $P(Y|X) \geq R \times E_{S'}(Y|X)$

Statistics-based Methods

- Example:
Browser=Mozilla \wedge Buy=Yes \rightarrow Age: $\mu=23$
- Rule consequent consists of a continuous variable, characterized by their statistics
 - mean, median, standard deviation, etc.
- Approach:
 - Withhold the target variable from the rest of the data
 - Apply existing frequent itemset generation on the rest of the data
 - For each frequent itemset, compute the descriptive statistics for the corresponding target variable
 - ◆ Frequent itemset becomes a rule by introducing the target variable as rule consequent
 - Apply statistical test to determine interestingness of the rule

Statistics-based Methods

- How to determine whether an association rule interesting?
 - Compare the statistics for segment of population covered by the rule vs segment of population not covered by the rule:
$$A \Rightarrow B: \mu \quad \text{versus} \quad \bar{A} \Rightarrow B: \mu'$$
 - Statistical hypothesis testing:
 - ◆ Null hypothesis: $H_0: \mu' = \mu + \Delta$
 - ◆ Alternative hypothesis: $H_1: \mu' > \mu + \Delta$
 - ◆ Z has zero mean and variance 1 under null hypothesis

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Statistics-based Methods

- Example:

$r: \text{Browser}=\text{Mozilla} \wedge \text{Buy}=\text{Yes} \rightarrow \text{Age}: \mu=23$

- Rule is interesting if difference between μ and μ' is greater than 5 years (i.e., $\Delta = 5$)
- For r , suppose $n_1 = 50$, $s_1 = 3.5$
- For r' (complement): $n_2 = 250$, $s_2 = 6.5$

$$Z = \frac{\mu' - \mu - \Delta}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = \frac{30 - 23 - 5}{\sqrt{\frac{3.5^2}{50} + \frac{6.5^2}{250}}} = 3.11$$

- For 1-sided test at 95% confidence level, critical Z-value for rejecting null hypothesis is 1.64.
- Since Z is greater than 1.64, r is an interesting rule

Min-Apriori (Han et al)

Document-term matrix:

TID	W1	W2	W3	W4	W5
D1	2	2	0	0	1
D2	0	0	1	2	2
D3	2	3	0	0	0
D4	0	0	1	0	1
D5	1	1	1	0	2

Example:

W1 and W2 tends to appear together in the same document

Min-Apriori

- Data contains only continuous attributes of the same “type”
 - e.g., frequency of words in a document

TID	W1	W2	W3	W4	W5
D1	2	2	0	0	1
D2	0	0	1	2	2
D3	2	3	0	0	0
D4	0	0	1	0	1
D5	1	1	1	0	2

- Potential solution:
 - Convert into 0/1 matrix and then apply existing algorithms
 - ◆ lose word frequency information
 - Discretization does not apply as users want association among words not ranges of words

Min-Apriori

- How to determine the support of a word?
 - If we simply sum up its frequency, support count will be greater than total number of documents!
 - ◆ Normalize the word vectors – e.g., using L_1 norm
 - ◆ Each word has a support equals to 1.0

TID	W1	W2	W3	W4	W5
D1	2	2	0	0	1
D2	0	0	1	2	2
D3	2	3	0	0	0
D4	0	0	1	0	1
D5	1	1	1	0	2

Normalize
→

TID	W1	W2	W3	W4	W5
D1	0.40	0.33	0.00	0.00	0.17
D2	0.00	0.00	0.33	1.00	0.33
D3	0.40	0.50	0.00	0.00	0.00
D4	0.00	0.00	0.33	0.00	0.17
D5	0.20	0.17	0.33	0.00	0.33

Min-Apriori

- New definition of support:

$$\text{sup}(C) = \sum_{i \in T} \min_{j \in C} D(i, j)$$

TID	W1	W2	W3	W4	W5
D1	0.40	0.33	0.00	0.00	0.17
D2	0.00	0.00	0.33	1.00	0.33
D3	0.40	0.50	0.00	0.00	0.00
D4	0.00	0.00	0.33	0.00	0.17
D5	0.20	0.17	0.33	0.00	0.33

Example:

Sup(W1,W2,W3)

$$= 0 + 0 + 0 + 0 + 0.17$$

$$= 0.17$$

Anti-monotone property of Support

TID	W1	W2	W3	W4	W5
D1	0.40	0.33	0.00	0.00	0.17
D2	0.00	0.00	0.33	1.00	0.33
D3	0.40	0.50	0.00	0.00	0.00
D4	0.00	0.00	0.33	0.00	0.17
D5	0.20	0.17	0.33	0.00	0.33

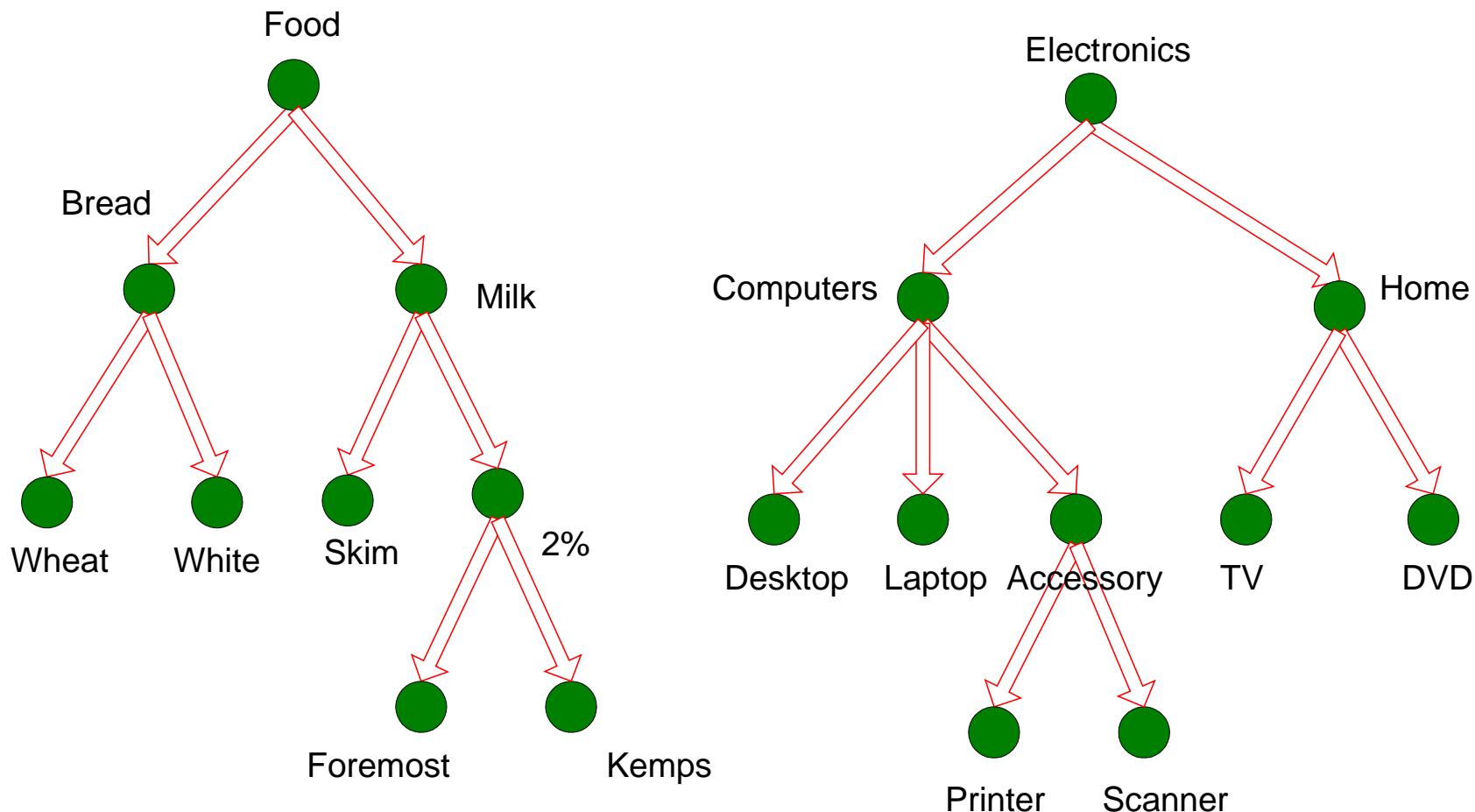
Example:

$$\text{Sup}(W1) = 0.4 + 0 + 0.4 + 0 + 0.2 = 1$$

$$\text{Sup}(W1, W2) = 0.33 + 0 + 0.4 + 0 + 0.17 = 0.9$$

$$\text{Sup}(W1, W2, W3) = 0 + 0 + 0 + 0 + 0.17 = 0.17$$

Multi-level Association Rules



Multi-level Association Rules

- Why should we incorporate concept hierarchy?
 - Rules at lower levels may not have enough support to appear in any frequent itemsets
 - Rules at lower levels of the hierarchy are overly specific
 - ◆ e.g., skim milk → white bread, 2% milk → wheat bread,
skim milk → wheat bread, etc.
are indicative of association between milk and bread

Multi-level Association Rules

- How do support and confidence vary as we traverse the concept hierarchy?
 - If X is the parent item for both X_1 and X_2 , then
$$\sigma(X) \leq \sigma(X_1) + \sigma(X_2)$$
 - If $\sigma(X_1 \cup Y_1) \geq \text{minsup}$,
and X is parent of X_1 , Y is parent of Y_1
then $\sigma(X \cup Y_1) \geq \text{minsup}$, $\sigma(X_1 \cup Y) \geq \text{minsup}$
$$\sigma(X \cup Y) \geq \text{minsup}$$
 - If $\text{conf}(X_1 \Rightarrow Y_1) \geq \text{minconf}$,
then $\text{conf}(X_1 \Rightarrow Y) \geq \text{minconf}$

Multi-level Association Rules

- Approach 1:
 - Extend current association rule formulation by augmenting each transaction with higher level items

Original Transaction: {skim milk, wheat bread}

Augmented Transaction:
{skim milk, wheat bread, milk, bread, food}

- Issues:
 - Items that reside at higher levels have much higher support counts
 - ◆ if support threshold is low, too many frequent patterns involving items from the higher levels
 - Increased dimensionality of the data

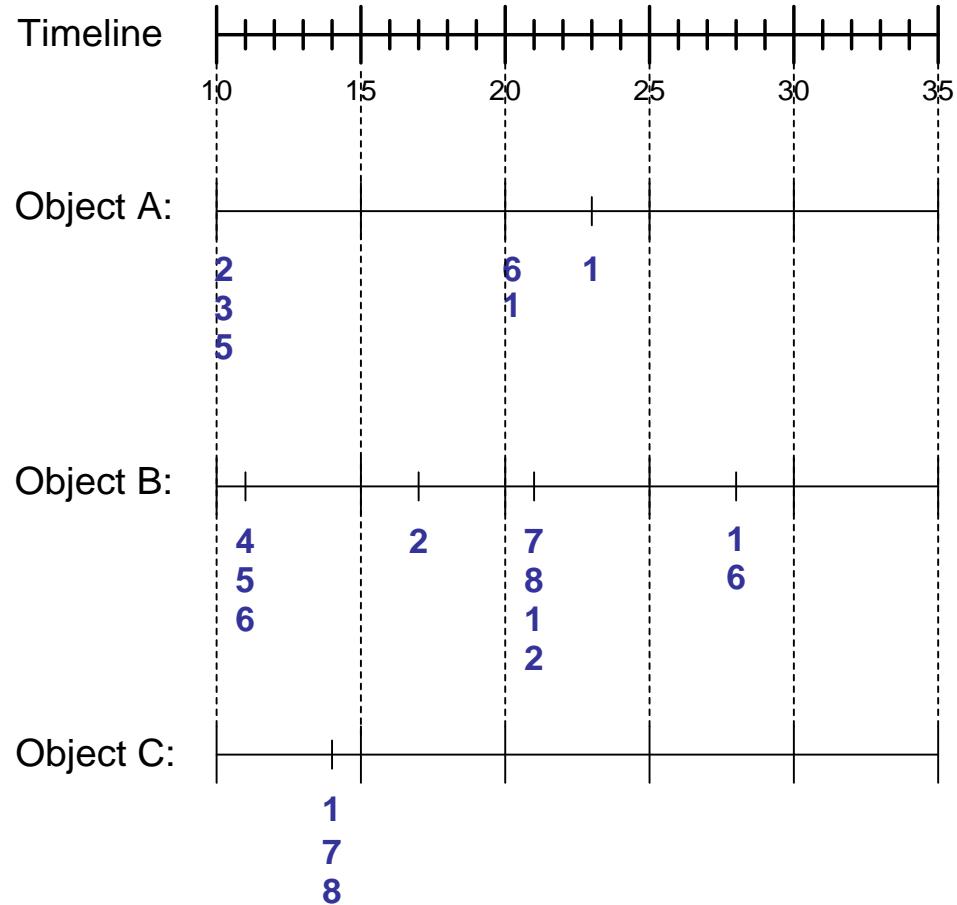
Multi-level Association Rules

- Approach 2:
 - Generate frequent patterns at highest level first
 - Then, generate frequent patterns at the next highest level, and so on
- Issues:
 - I/O requirements will increase dramatically because we need to perform more passes over the data
 - May miss some potentially interesting cross-level association patterns

Sequence Data

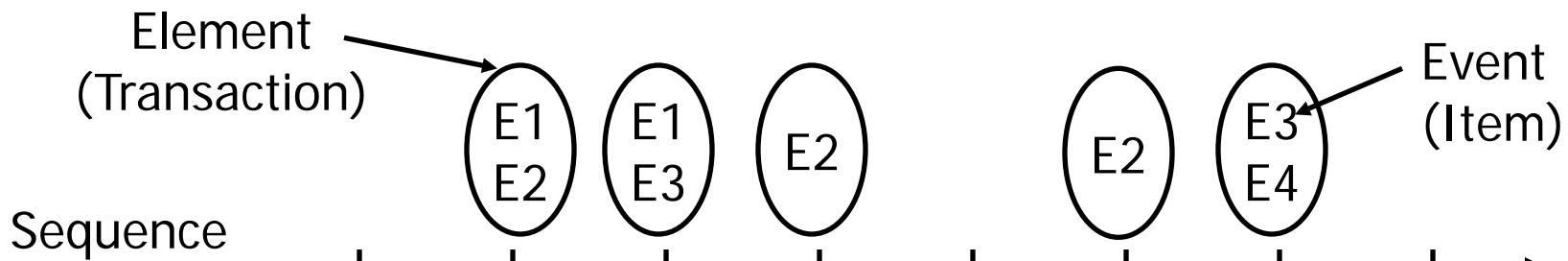
Sequence Database:

Object	Timestamp	Events
A	10	2, 3, 5
A	20	6, 1
A	23	1
B	11	4, 5, 6
B	17	2
B	21	7, 8, 1, 2
B	28	1, 6
C	14	1, 8, 7



Examples of Sequence Data

Sequence Database	Sequence	Element (Transaction)	Event (Item)
Customer	Purchase history of a given customer	A set of items bought by a customer at time t	Books, diary products, CDs, etc
Web Data	Browsing activity of a particular Web visitor	A collection of files viewed by a Web visitor after a single mouse click	Home page, index page, contact info, etc
Event data	History of events generated by a given sensor	Events triggered by a sensor at time t	Types of alarms generated by sensors
Genome sequences	DNA sequence of a particular species	An element of the DNA sequence	Bases A,T,G,C



Formal Definition of a Sequence

- A sequence is an ordered list of elements (transactions)

$$S = < e_1 \ e_2 \ e_3 \ \dots >$$

- Each element contains a collection of events (items)

$$e_i = \{i_1, i_2, \dots, i_k\}$$

- Each element is attributed to a specific time or location

- Length of a sequence, $|S|$, is given by the number of elements of the sequence
- A k-sequence is a sequence that contains k events (items)

Examples of Sequence

- Web sequence:

< {Homepage} {Electronics} {Digital Cameras} {Canon Digital Camera}
 {Shopping Cart} {Order Confirmation} {Return to Shopping} >

- Sequence of initiating events causing the nuclear accident at 3-mile Island:

(http://stellar-one.com/nuclear/staff_reports/summary_SOE_the_initiating_event.htm)

< {clogged resin} {outlet valve closure} {loss of feedwater}
 {condenser polisher outlet valve shut} {booster pumps trip}
 {main waterpump trips} {main turbine trips} {reactor pressure increases}>

- Sequence of books checked out at a library:

<{Fellowship of the Ring} {The Two Towers} {Return of the King}>

Formal Definition of a Subsequence

- A sequence $\langle a_1, a_2, \dots, a_n \rangle$ is contained in another sequence $\langle b_1, b_2, \dots, b_m \rangle$ ($m \geq n$) if there exist integers $i_1 < i_2 < \dots < i_n$ such that $a_1 \subseteq b_{i1}, a_2 \subseteq b_{i2}, \dots, a_n \subseteq b_{in}$

Data sequence	Subsequence	Contain?
$\langle \{2,4\} \{3,5,6\} \{8\} \rangle$	$\langle \{2\} \{3,5\} \rangle$	Yes
$\langle \{1,2\} \{3,4\} \rangle$	$\langle \{1\} \{2\} \rangle$	No
$\langle \{2,4\} \{2,4\} \{2,5\} \rangle$	$\langle \{2\} \{4\} \rangle$	Yes

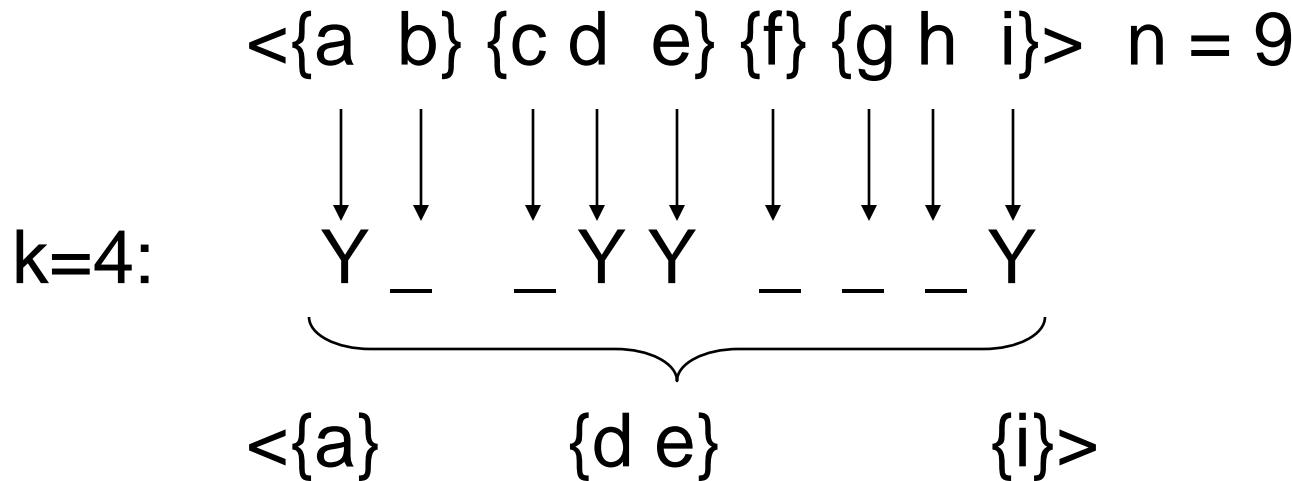
- The support of a subsequence w is defined as the fraction of data sequences that contain w
- A *sequential pattern* is a frequent subsequence (i.e., a subsequence whose support is $\geq \text{minsup}$)

Sequential Pattern Mining: Definition

- Given:
 - a database of sequences
 - a user-specified minimum support threshold, $minsup$
- Task:
 - Find all subsequences with support $\geq minsup$

Sequential Pattern Mining: Challenge

- Given a sequence: $\langle \{a\ b\} \{c\ d\ e\} \{f\} \{g\ h\ i\} \rangle$
 - Examples of subsequences:
 $\langle \{a\} \{c\ d\} \{f\} \{g\} \rangle, \langle \{c\ d\ e\} \rangle, \langle \{b\} \{g\} \rangle$, etc.
- How many k-subsequences can be extracted from a given n-sequence?



Answer :

$$\binom{n}{k} = \binom{9}{4} = 126$$

Sequential Pattern Mining: Example

Object	Timestamp	Events
A	1	1,2,4
A	2	2,3
A	3	5
B	1	1,2
B	2	2,3,4
C	1	1, 2
C	2	2,3,4
C	3	2,4,5
D	1	2
D	2	3, 4
D	3	4, 5
E	1	1, 3
E	2	2, 4, 5

Minsup = 50%

Examples of Frequent Subsequences:

< {1,2} >	s=60%
< {2,3} >	s=60%
< {2,4}>	s=80%
< {3} {5}>	s=80%
< {1} {2} >	s=80%
< {2} {2} >	s=60%
< {1} {2,3} >	s=60%
< {2} {2,3} >	s=60%
< {1,2} {2,3} >	s=60%

Extracting Sequential Patterns

- Given n events: $i_1, i_2, i_3, \dots, i_n$
- Candidate 1-subsequences:
 $\langle\{i_1\}\rangle, \langle\{i_2\}\rangle, \langle\{i_3\}\rangle, \dots, \langle\{i_n\}\rangle$
- Candidate 2-subsequences:
 $\langle\{i_1, i_2\}\rangle, \langle\{i_1, i_3\}\rangle, \dots, \langle\{i_1\} \{i_1\}\rangle, \langle\{i_1\} \{i_2\}\rangle, \dots, \langle\{i_{n-1}\} \{i_n\}\rangle$
- Candidate 3-subsequences:
 $\langle\{i_1, i_2, i_3\}\rangle, \langle\{i_1, i_2, i_4\}\rangle, \dots, \langle\{i_1, i_2\} \{i_1\}\rangle, \langle\{i_1, i_2\} \{i_2\}\rangle, \dots,$
 $\langle\{i_1\} \{i_1, i_2\}\rangle, \langle\{i_1\} \{i_1, i_3\}\rangle, \dots, \langle\{i_1\} \{i_1\} \{i_1\}\rangle, \langle\{i_1\} \{i_1\} \{i_2\}\rangle, \dots$

Generalized Sequential Pattern (GSP)

- **Step 1:**
 - Make the first pass over the sequence database D to yield all the 1-element frequent sequences
- **Step 2:**

Repeat until no new frequent sequences are found

 - **Candidate Generation:**
 - ◆ Merge pairs of frequent subsequences found in the $(k-1)^{th}$ pass to generate candidate sequences that contain k items
 - **Candidate Pruning:**
 - ◆ Prune candidate k -sequences that contain infrequent $(k-1)$ -subsequences
 - **Support Counting:**
 - ◆ Make a new pass over the sequence database D to find the support for these candidate sequences
 - **Candidate Elimination:**
 - ◆ Eliminate candidate k -sequences whose actual support is less than $minsup$

Candidate Generation

- Base case ($k=2$):
 - Merging two frequent 1-sequences $\langle\{i_1\}\rangle$ and $\langle\{i_2\}\rangle$ will produce two candidate 2-sequences: $\langle\{i_1\} \{i_2\}\rangle$ and $\langle\{i_1 i_2\}\rangle$
- General case ($k>2$):
 - A frequent $(k-1)$ -sequence w_1 is merged with another frequent $(k-1)$ -sequence w_2 to produce a candidate k -sequence if the subsequence obtained by removing the first event in w_1 is the same as the subsequence obtained by removing the last event in w_2
 - ◆ The resulting candidate after merging is given by the sequence w_1 extended with the last event of w_2 .
 - If the last two events in w_2 belong to the same element, then the last event in w_2 becomes part of the last element in w_1
 - Otherwise, the last event in w_2 becomes a separate element appended to the end of w_1

Candidate Generation Examples

- Merging the sequences

$w_1 = \langle \{1\} \{2 3\} \{4\} \rangle$ and $w_2 = \langle \{2 3\} \{4 5\} \rangle$

will produce the candidate sequence $\langle \{1\} \{2 3\} \{4 5\} \rangle$ because the last two events in w_2 (4 and 5) belong to the same element

- Merging the sequences

$w_1 = \langle \{1\} \{2 3\} \{4\} \rangle$ and $w_2 = \langle \{2 3\} \{4\} \{5\} \rangle$

will produce the candidate sequence $\langle \{1\} \{2 3\} \{4\} \{5\} \rangle$ because the last two events in w_2 (4 and 5) do not belong to the same element

- We do not have to merge the sequences

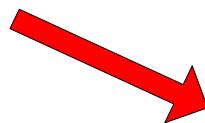
$w_1 = \langle \{1\} \{2 6\} \{4\} \rangle$ and $w_2 = \langle \{1\} \{2\} \{4 5\} \rangle$

to produce the candidate $\langle \{1\} \{2 6\} \{4 5\} \rangle$ because if the latter is a viable candidate, then it can be obtained by merging w_1 with $\langle \{1\} \{2 6\} \{5\} \rangle$

GSP Example

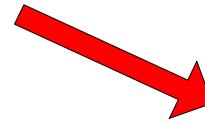
Frequent
3-sequences

```
< {1} {2} {3} >  
< {1} {2 5} >  
< {1} {5} {3} >  
< {2} {3} {4} >  
< {2 5} {3} >  
< {3} {4} {5} >  
< {5} {3 4} >
```



Candidate
Generation

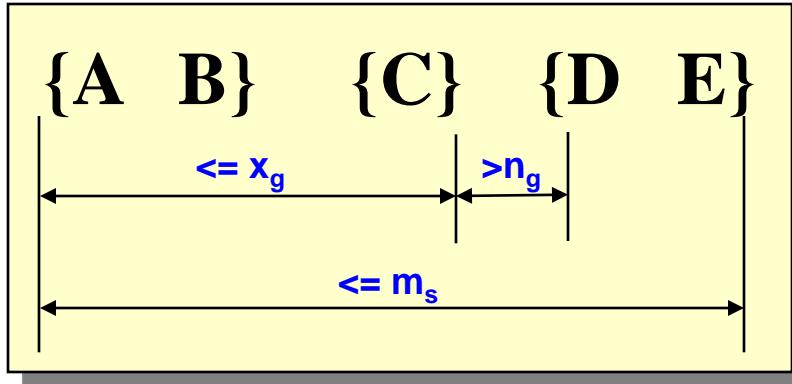
```
< {1} {2} {3} {4} >  
< {1} {2 5} {3} >  
< {1} {5} {3 4} >  
< {2} {3} {4} {5} >  
< {2 5} {3 4} >
```



Candidate
Pruning

```
< {1} {2 5} {3} >
```

Timing Constraints (I)



x_g : max-gap

n_g : min-gap

m_s : maximum span

$$x_g = 2, n_g = 0, m_s = 4$$

Data sequence	Subsequence	Contain?
$< \{2,4\} \{3,5,6\} \{4,7\} \{4,5\} \{8\} >$	$< \{6\} \{5\} >$	Yes
$< \{1\} \{2\} \{3\} \{4\} \{5\} >$	$< \{1\} \{4\} >$	No
$< \{1\} \{2,3\} \{3,4\} \{4,5\} >$	$< \{2\} \{3\} \{5\} >$	Yes
$< \{1,2\} \{3\} \{2,3\} \{3,4\} \{2,4\} \{4,5\} >$	$< \{1,2\} \{5\} >$	No

Mining Sequential Patterns with Timing Constraints

- Approach 1:
 - Mine sequential patterns without timing constraints
 - Postprocess the discovered patterns
- Approach 2:
 - Modify GSP to directly prune candidates that violate timing constraints
 - Question:
 - ◆ Does Apriori principle still hold?

Apriori Principle for Sequence Data

Object	Timestamp	Events
A	1	1,2,4
A	2	2,3
A	3	5
B	1	1,2
B	2	2,3,4
C	1	1, 2
C	2	2,3,4
C	3	2,4,5
D	1	2
D	2	3, 4
D	3	4, 5
E	1	1, 3
E	2	2, 4, 5

Suppose:

$$x_g = 1 \text{ (max-gap)}$$

$$n_g = 0 \text{ (min-gap)}$$

$$m_s = 5 \text{ (maximum span)}$$

$$\text{minsup} = 60\%$$

$$<\{2\} \{5\}> \text{ support} = 40\%$$

but

$$<\{2\} \{3\} \{5\}> \text{ support} = 60\%$$

Problem exists because of max-gap constraint

No such problem if max-gap is infinite

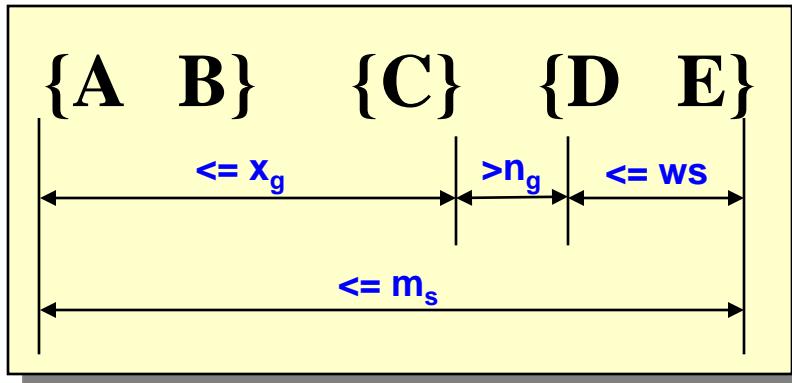
Contiguous Subsequences

- s is a contiguous subsequence of
 $w = \langle e_1 \rangle \langle e_2 \rangle \dots \langle e_k \rangle$
if any of the following conditions hold:
 1. s is obtained from w by deleting an item from either e_1 or e_k
 2. s is obtained from w by deleting an item from any element e_i that contains more than 2 items
 3. s is a contiguous subsequence of s' and s' is a contiguous subsequence of w (recursive definition)
- Examples: $s = \langle \{1\} \{2\} \rangle$
 - is a contiguous subsequence of
 $\langle \{1\} \{2\} \{3\} \rangle$, $\langle \{1\} \{2\} \{3\} \{4\} \rangle$, and $\langle \{1\} \{2\} \{3\} \{4\} \{5\} \rangle$
 - is not a contiguous subsequence of
 $\langle \{1\} \{3\} \{2\} \rangle$ and $\langle \{2\} \{1\} \{3\} \{2\} \rangle$

Modified Candidate Pruning Step

- Without maxgap constraint:
 - A candidate k -sequence is pruned if at least one of its $(k-1)$ -subsequences is infrequent
- With maxgap constraint:
 - A candidate k -sequence is pruned if at least one of its **contiguous** $(k-1)$ -subsequences is infrequent

Timing Constraints (II)



x_g : max-gap

n_g : min-gap

ws : window size

m_s : maximum span

$$x_g = 2, n_g = 0, ws = 1, m_s = 5$$

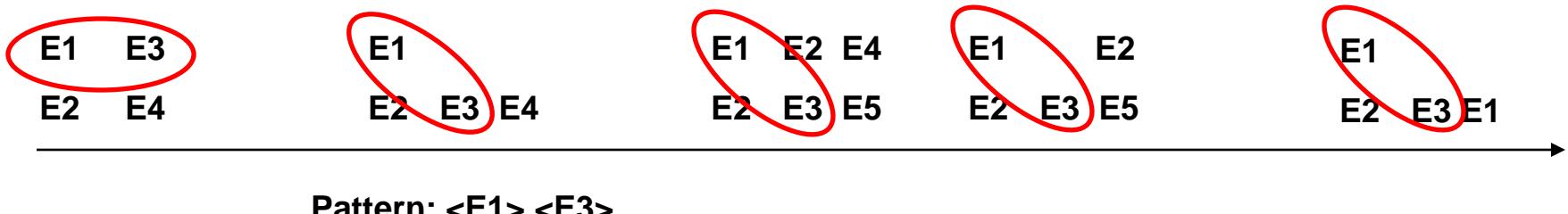
Data sequence	Subsequence	Contain?
$< \{2,4\} \{3,5,6\} \{4,7\} \{4,6\} \{8\} >$	$< \{3\} \{5\} >$	No
$< \{1\} \{2\} \{3\} \{4\} \{5\} >$	$< \{1,2\} \{3\} >$	Yes
$< \{1,2\} \{2,3\} \{3,4\} \{4,5\} >$	$< \{1,2\} \{3,4\} >$	Yes

Modified Support Counting Step

- Given a candidate pattern: $\langle \{a, c\} \rangle$
 - Any data sequences that contain
 - $\langle \dots \{a c\} \dots \rangle,$
 - $\langle \dots \{a\} \dots \{c\} \dots \rangle$ (where $\text{time}(\{c\}) - \text{time}(\{a\}) \leq ws$)
 - $\langle \dots \{c\} \dots \{a\} \dots \rangle$ (where $\text{time}(\{a\}) - \text{time}(\{c\}) \leq ws$)
- will contribute to the support count of candidate pattern

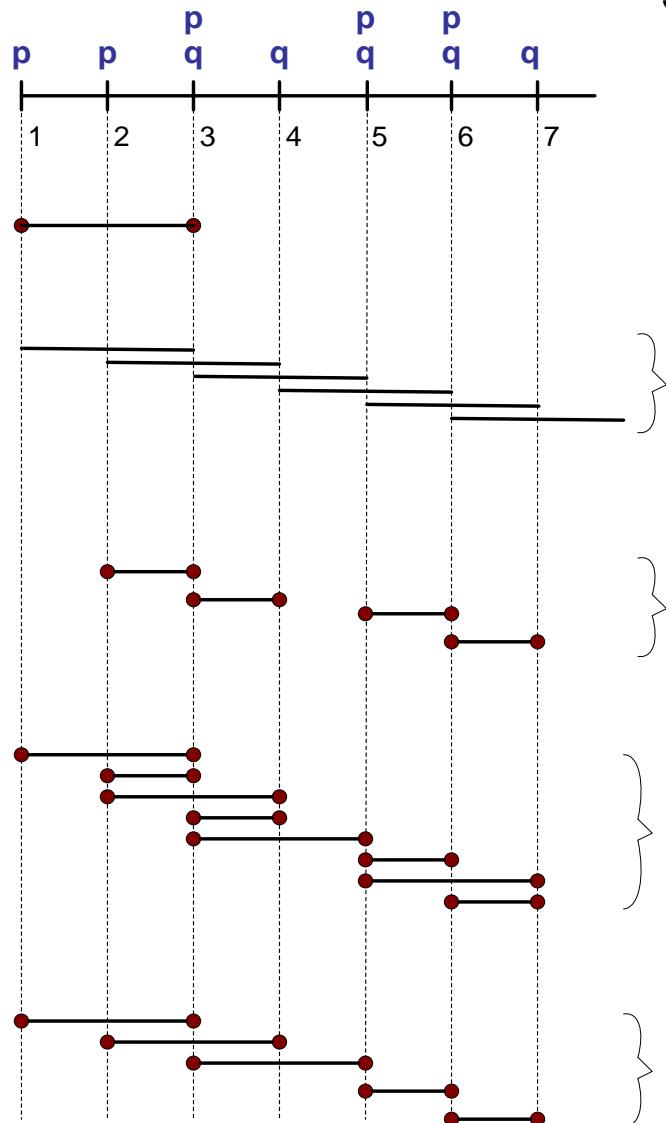
Other Formulation

- In some domains, we may have only one very long time series
 - Example:
 - ◆ monitoring network traffic events for attacks
 - ◆ monitoring telecommunication alarm signals
- Goal is to find frequent sequences of events in the time series
 - This problem is also known as frequent episode mining



General Support Counting Schemes

Object's Timeline



Sequence: (p) (q)

Method Support
Count

COBJ 1

CWIN 6

CMINWIN 4

CDIST_O 8

CDIST 5

Assume:

$x_g = 2$ (max-gap)

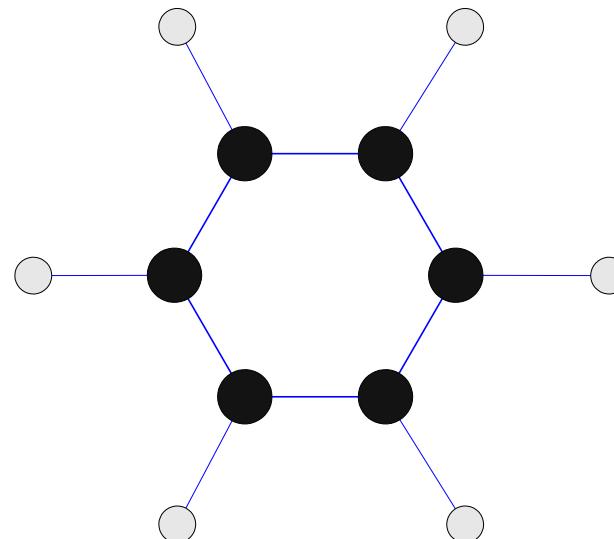
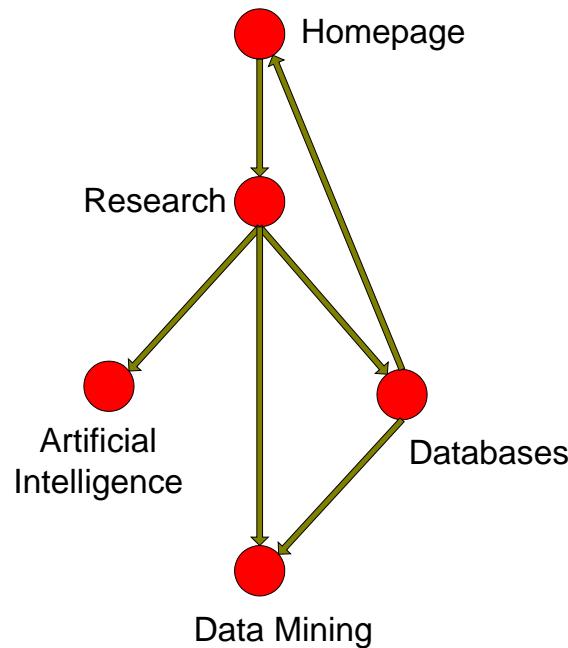
$n_g = 0$ (min-gap)

$ws = 0$ (window size)

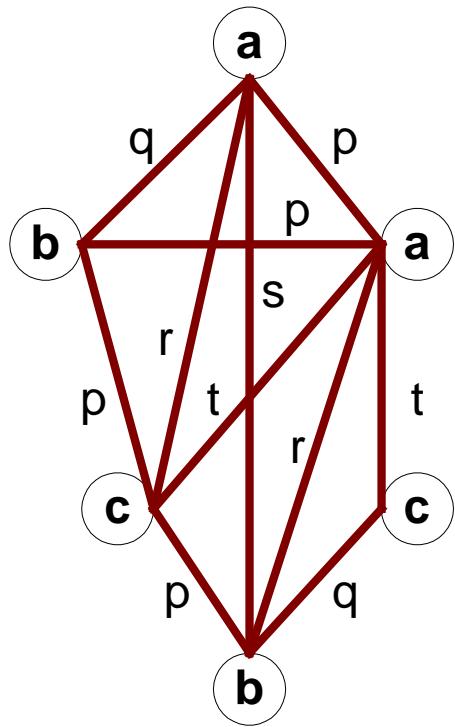
$m_s = 2$ (maximum span)

Frequent Subgraph Mining

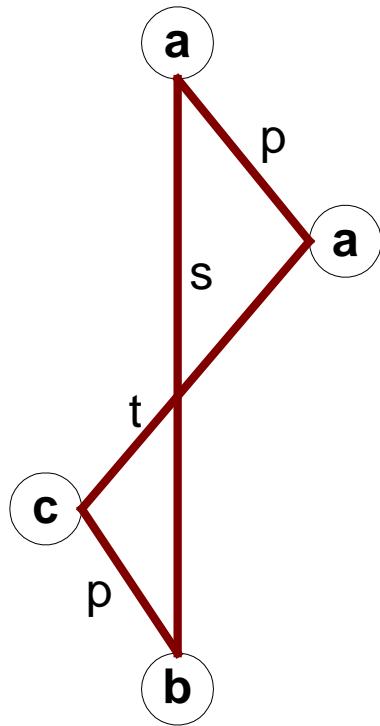
- Extend association rule mining to finding frequent subgraphs
- Useful for Web Mining, computational chemistry, bioinformatics, spatial data sets, etc



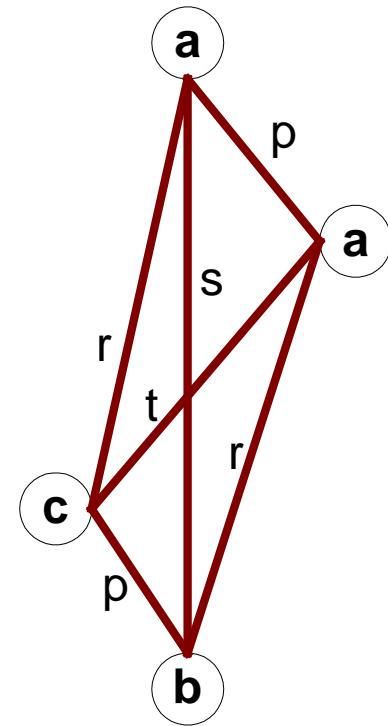
Graph Definitions



(a) Labeled Graph



(b) Subgraph

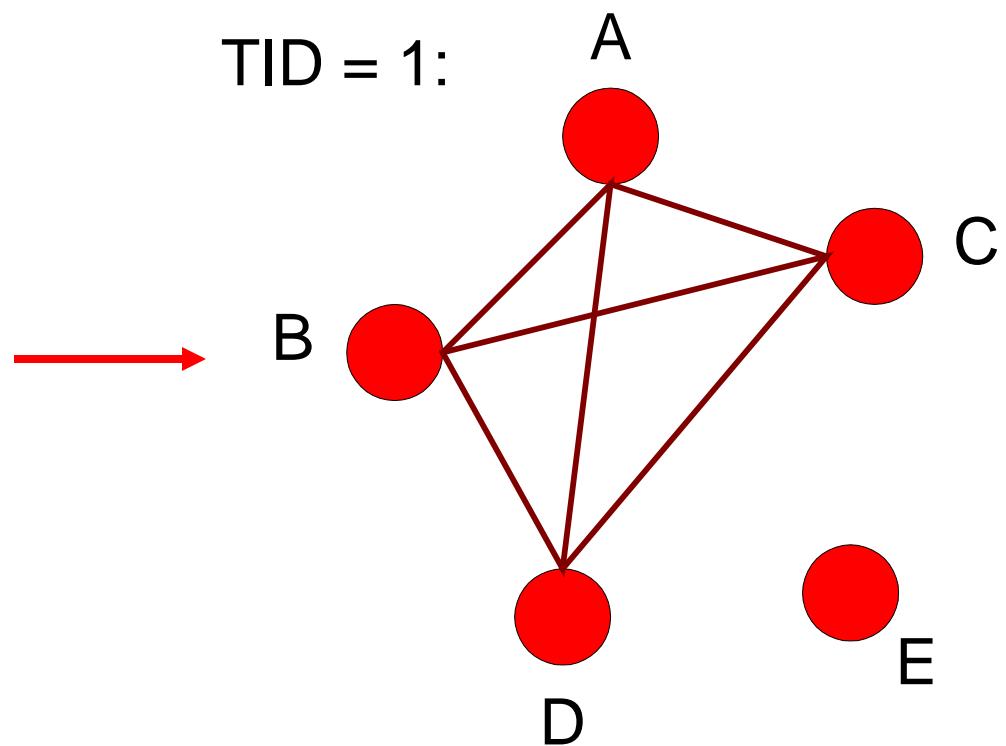


(c) Induced Subgraph

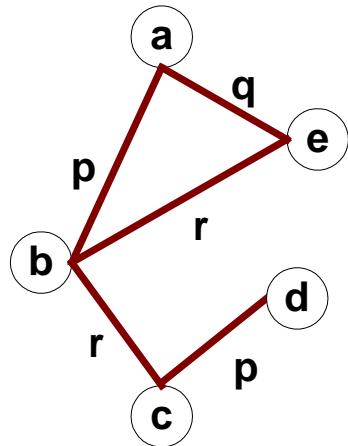
Representing Transactions as Graphs

- Each transaction is a clique of items

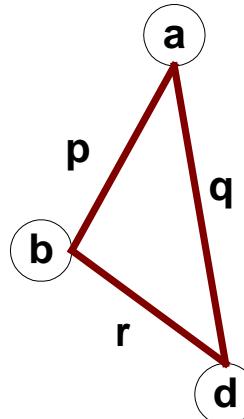
Transaction Id	Items
1	{A,B,C,D}
2	{A,B,E}
3	{B,C}
4	{A,B,D,E}
5	{B,C,D}



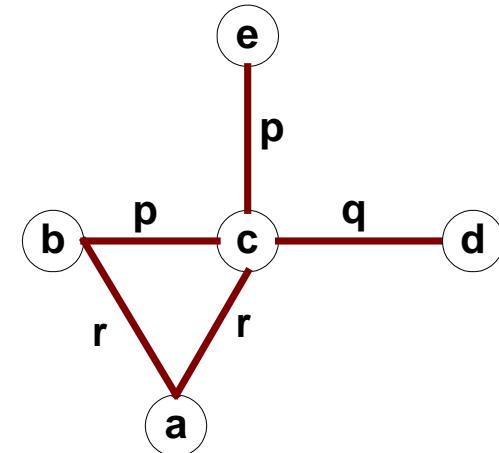
Representing Graphs as Transactions



G1



G2



G3

	(a,b,p)	(a,b,q)	(a,b,r)	(b,c,p)	(b,c,q)	(b,c,r)	...	(d,e,r)
G1	1	0	0	0	0	1	...	0
G2	1	0	0	0	0	0	...	0
G3	0	0	1	1	0	0	...	0
G3

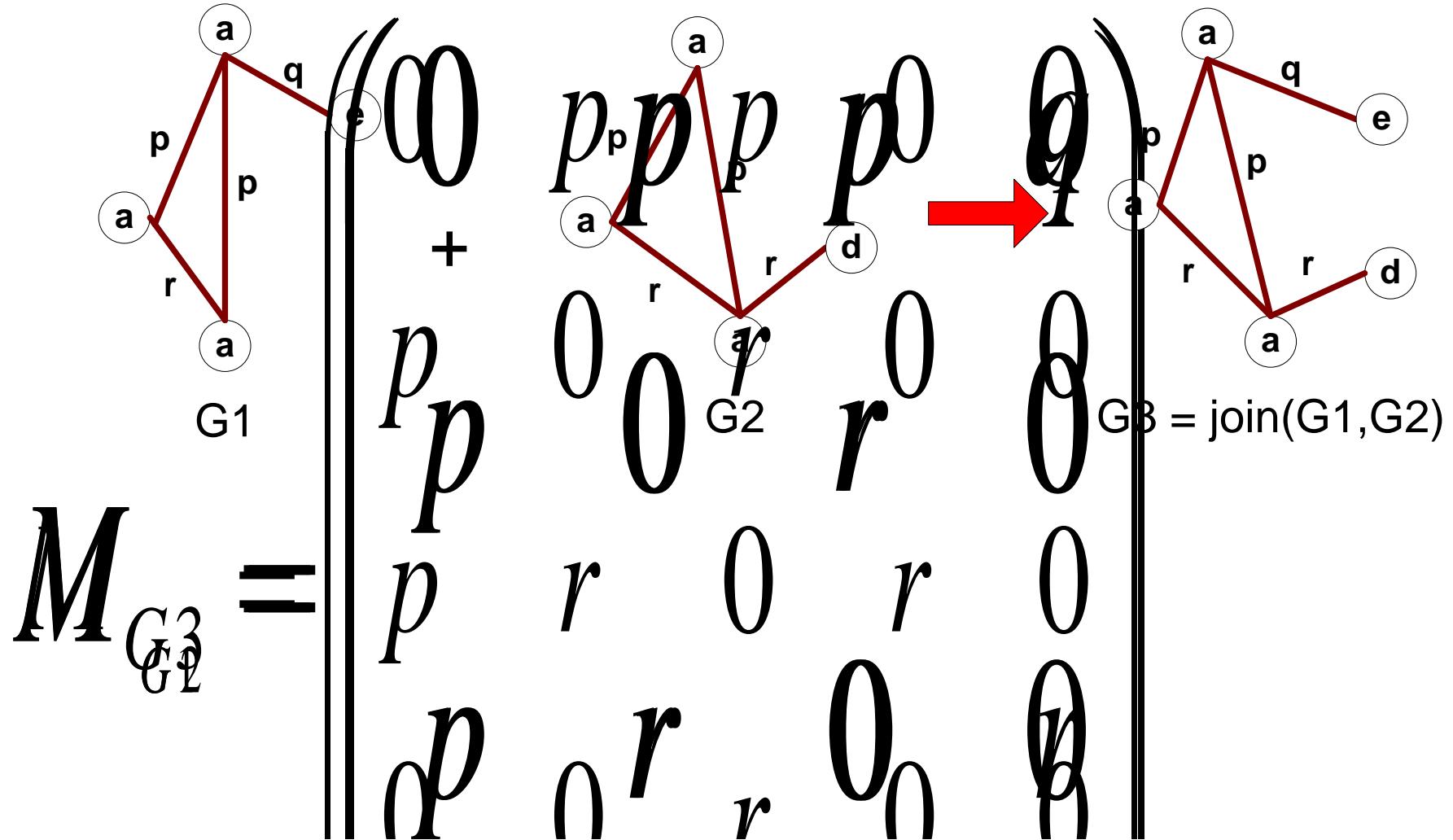
Challenges

- Node may contain duplicate labels
- Support and confidence
 - How to define them?
- Additional constraints imposed by pattern structure
 - Support and confidence are not the only constraints
 - Assumption: frequent subgraphs must be connected
- Apriori-like approach:
 - Use frequent k-subgraphs to generate frequent (k+1) subgraphs
 - ◆ What is k?

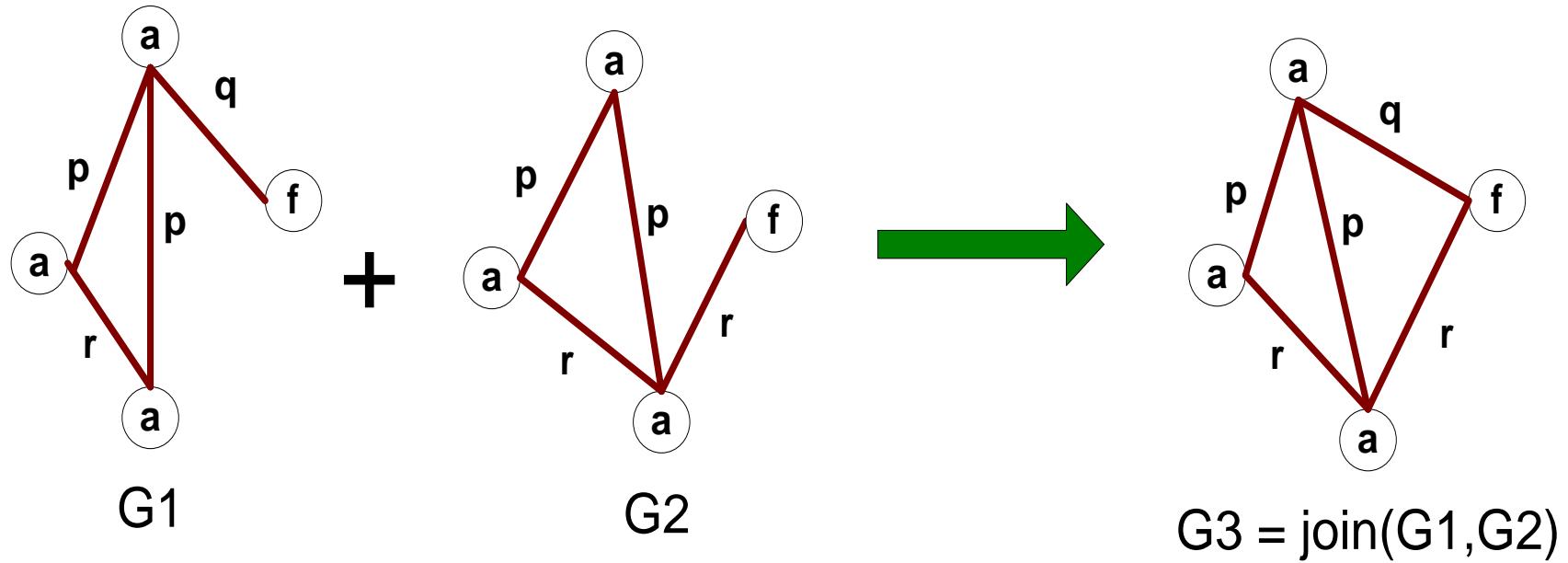
Challenges...

- Support:
 - number of graphs that contain a particular subgraph
- Apriori principle still holds
- Level-wise (Apriori-like) approach:
 - Vertex growing:
 - ◆ k is the number of vertices
 - Edge growing:
 - ◆ k is the number of edges

Vertex Growing



Edge Growing



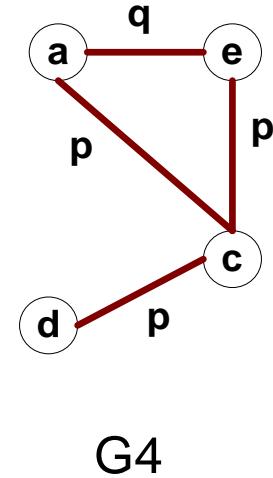
Apriori-like Algorithm

- Find frequent 1-subgraphs
- Repeat
 - Candidate generation
 - ◆ Use frequent $(k-1)$ -subgraphs to generate candidate k -subgraph
 - Candidate pruning
 - ◆ Prune candidate subgraphs that contain infrequent $(k-1)$ -subgraphs
 - Support counting
 - ◆ Count the support of each remaining candidate
 - Eliminate candidate k -subgraphs that are infrequent

In practice, it is not as easy. There are many other issues

Example: Dataset

	(a,b,p)	(a,b,q)	(a,b,r)	(b,c,p)	(b,c,q)	(b,c,r)	...	(d,e,r)
G1	a c d e	b d	b d	c d	c d	a c d	...	c d
G2	1	1	1	1	1	1	...	1
G3	1	1	1	1	1	1	...	1



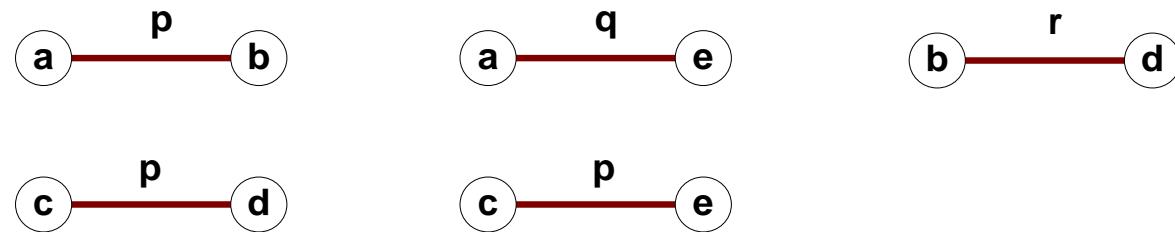
Example

Minimum support count = 2

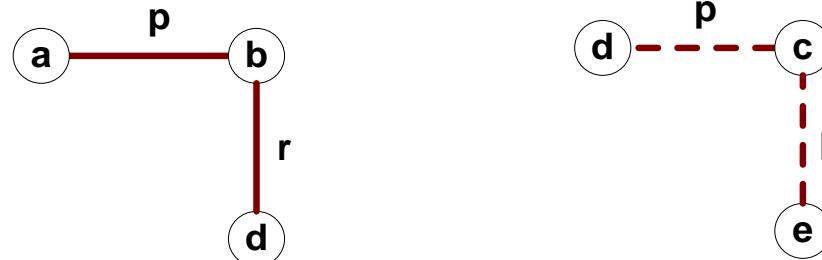
k=1
Frequent
Subgraphs



k=2
Frequent
Subgraphs



k=3
Candidate
Subgraphs

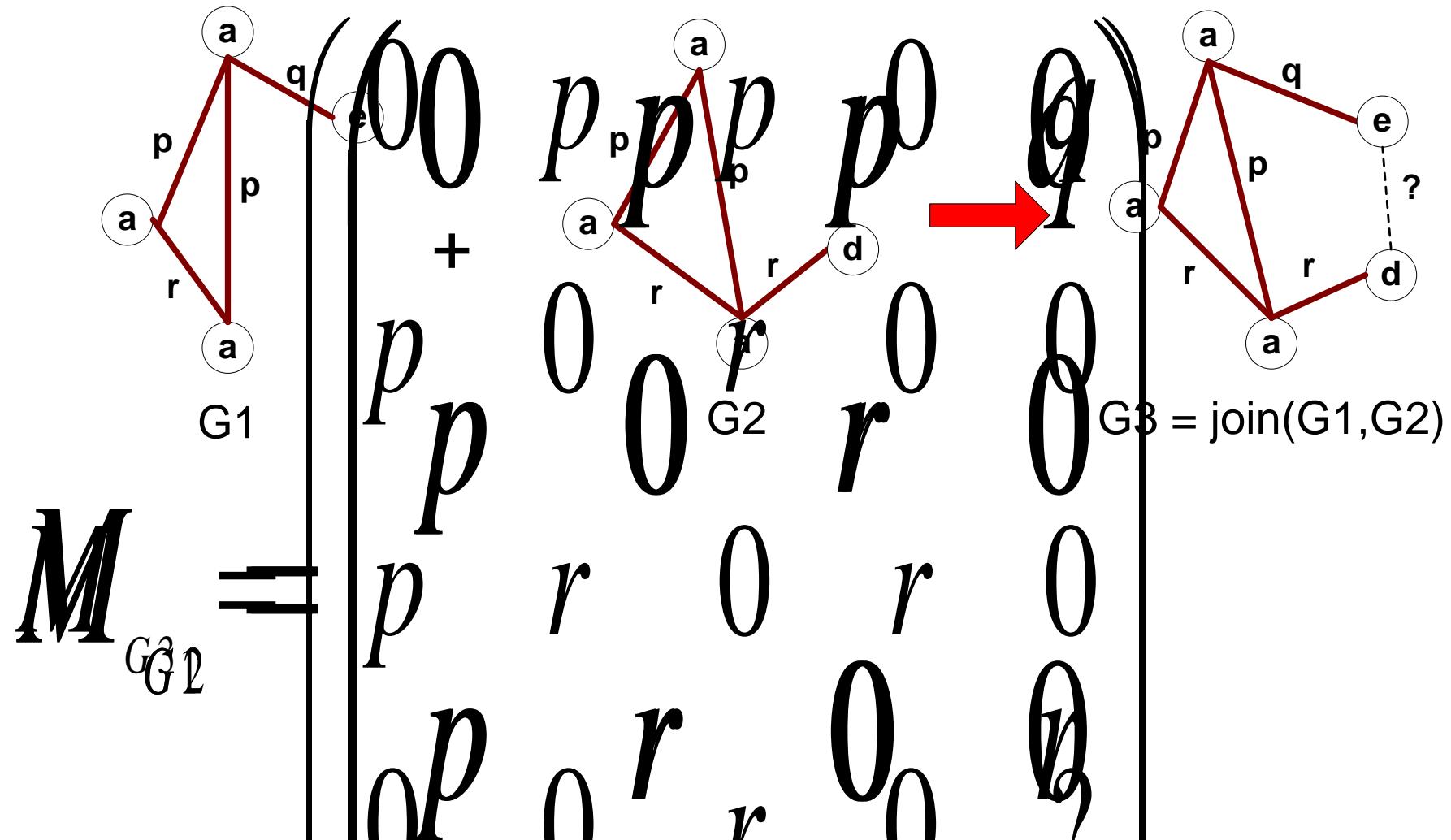


(Pruned candidate)

Candidate Generation

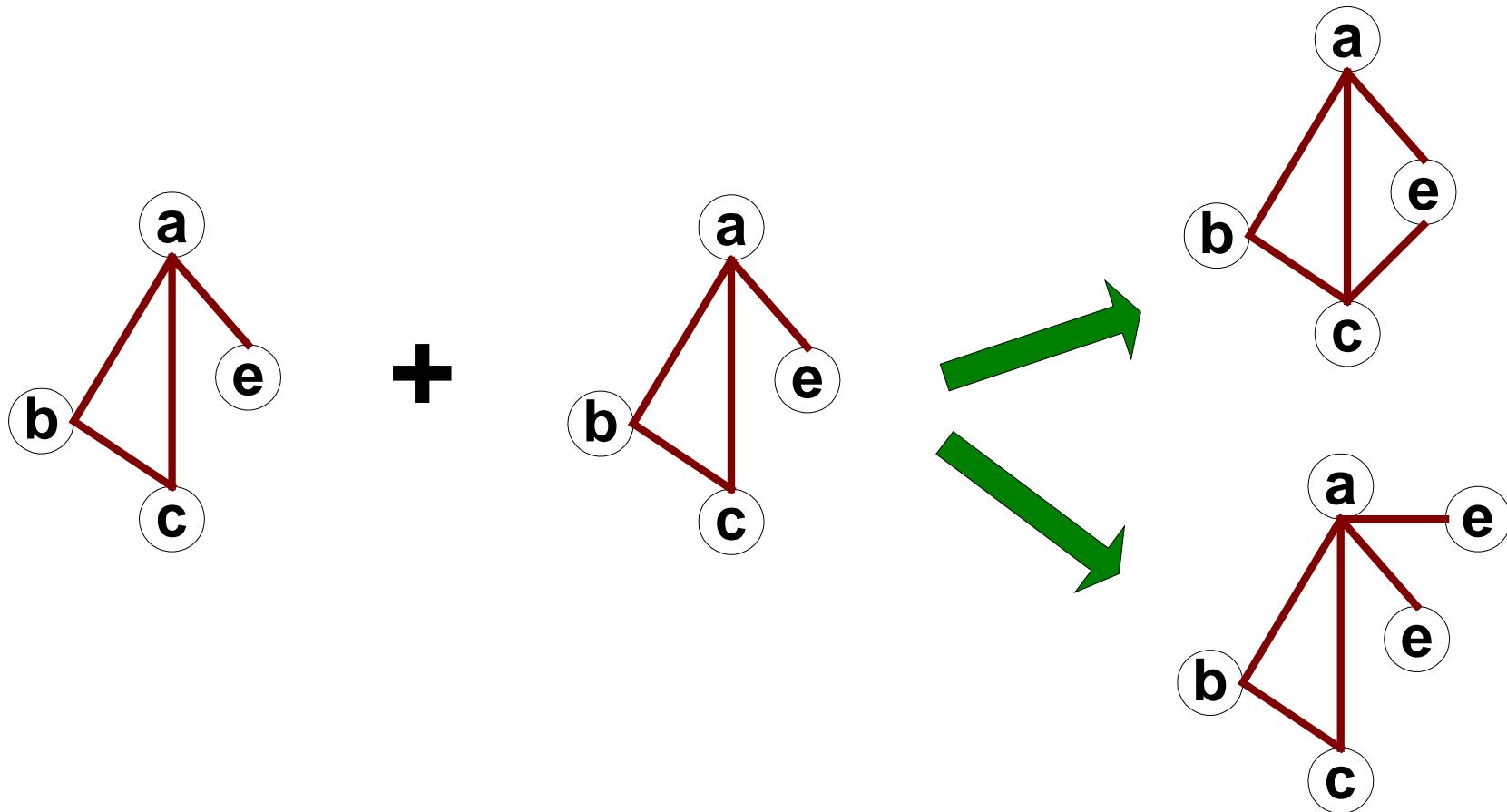
- In Apriori:
 - Merging two frequent k -itemsets will produce a candidate $(k+1)$ -itemset
- In frequent subgraph mining (vertex/edge growing)
 - Merging two frequent k -subgraphs may produce more than one candidate $(k+1)$ -subgraph

Multiplicity of Candidates (Vertex Growing)



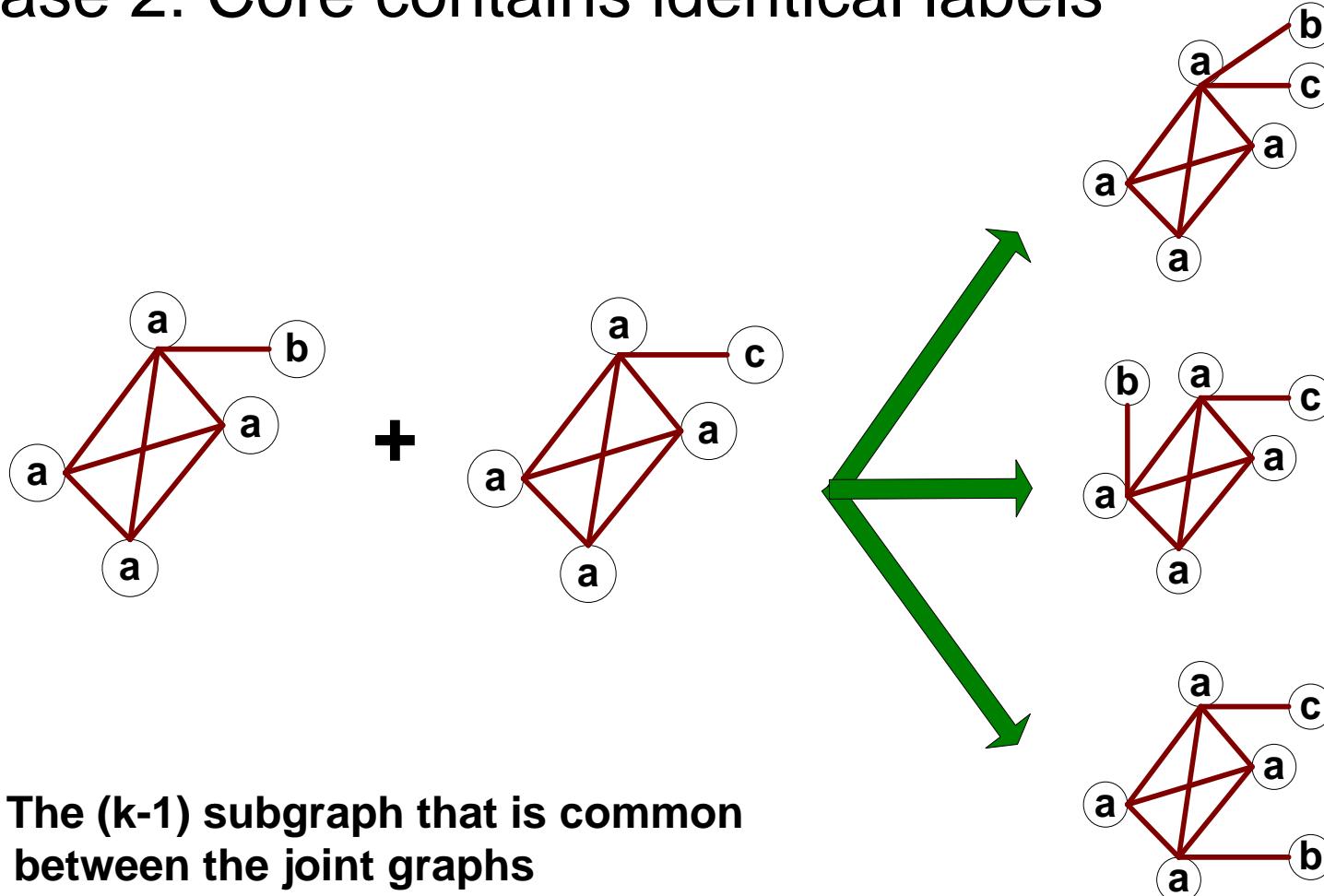
Multiplicity of Candidates (Edge growing)

- Case 1: identical vertex labels



Multiplicity of Candidates (Edge growing)

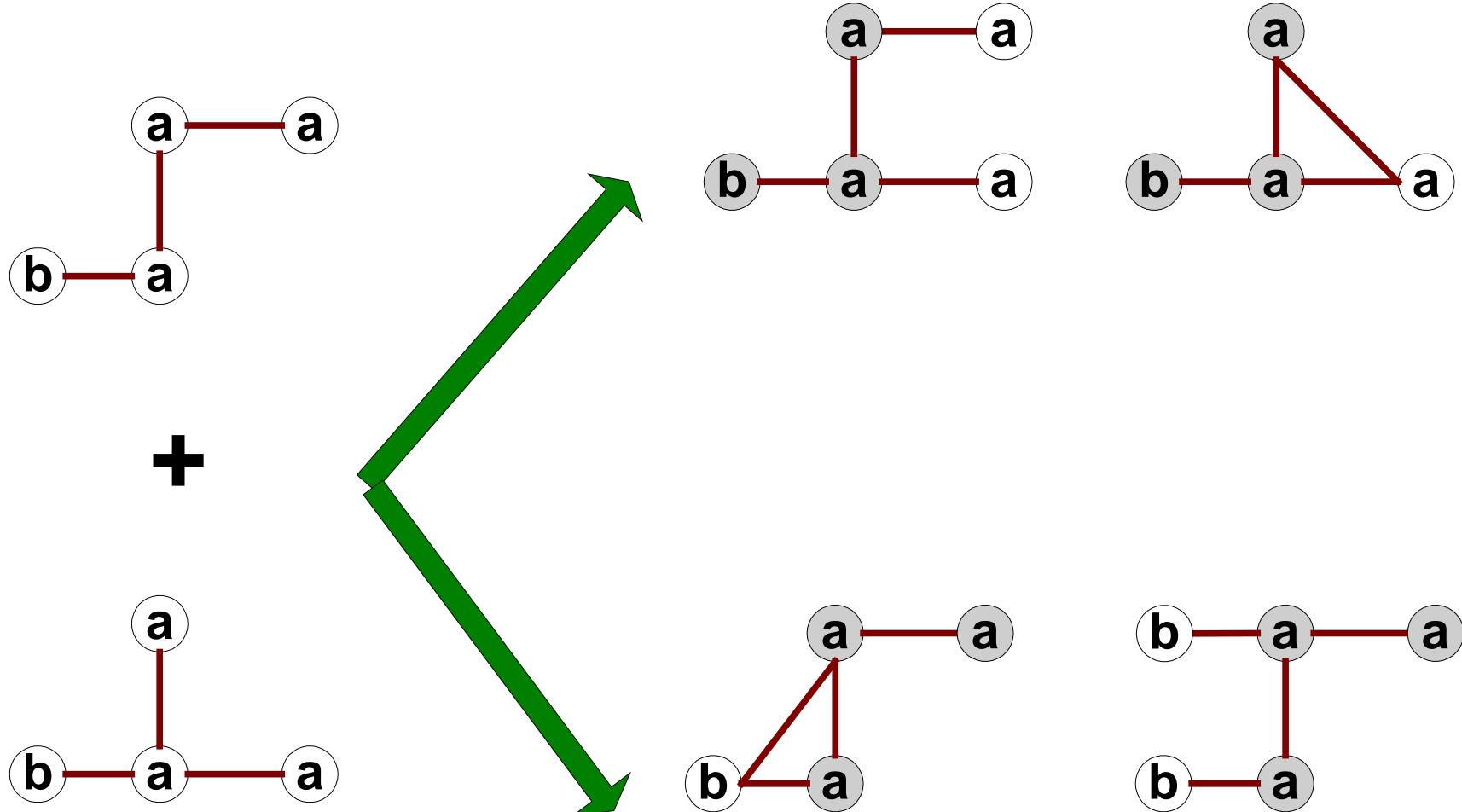
- Case 2: Core contains identical labels



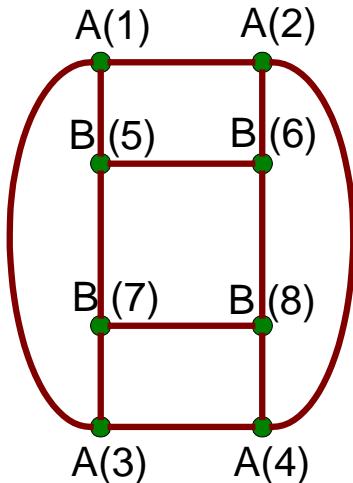
Core: The $(k-1)$ subgraph that is common between the joint graphs

Multiplicity of Candidates (Edge growing)

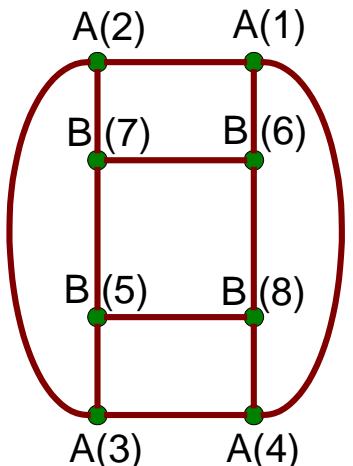
- Case 3: Core multiplicity



Adjacency Matrix Representation



	A(1)	A(2)	A(3)	A(4)	B(5)	B(6)	B(7)	B(8)
A(1)	1	1	1	0	1	0	0	0
A(2)	1	1	0	1	0	1	0	0
A(3)	1	0	1	1	0	0	1	0
A(4)	0	1	1	1	0	0	0	1
B(5)	1	0	0	0	1	1	1	0
B(6)	0	1	0	0	1	1	0	1
B(7)	0	0	1	0	1	0	1	1
B(8)	0	0	0	1	0	1	1	1

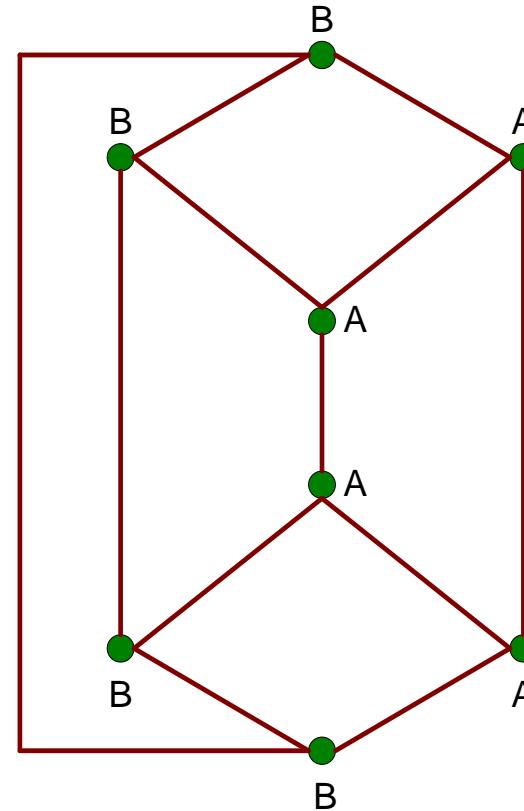
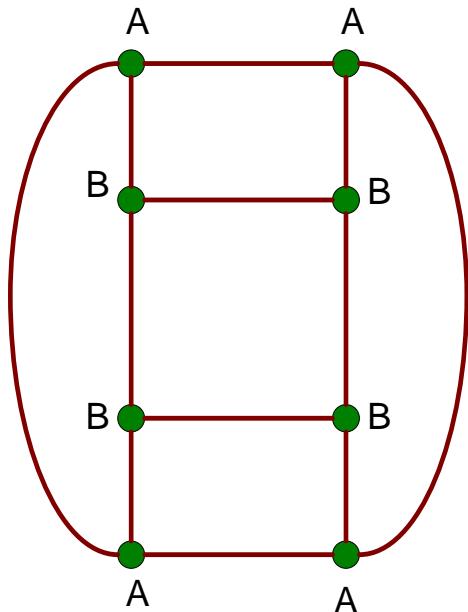


	A(1)	A(2)	A(3)	A(4)	B(5)	B(6)	B(7)	B(8)
A(1)	1	1	0	1	0	1	0	0
A(2)	1	1	1	0	0	0	1	0
A(3)	0	1	1	1	1	0	0	0
A(4)	1	0	1	1	0	0	0	1
B(5)	0	0	1	0	1	0	1	1
B(6)	1	0	0	0	0	1	1	1
B(7)	0	1	0	0	1	1	1	0
B(8)	0	0	0	1	1	1	0	1

- The same graph can be represented in many ways

Graph Isomorphism

- A graph is isomorphic if it is topologically equivalent to another graph

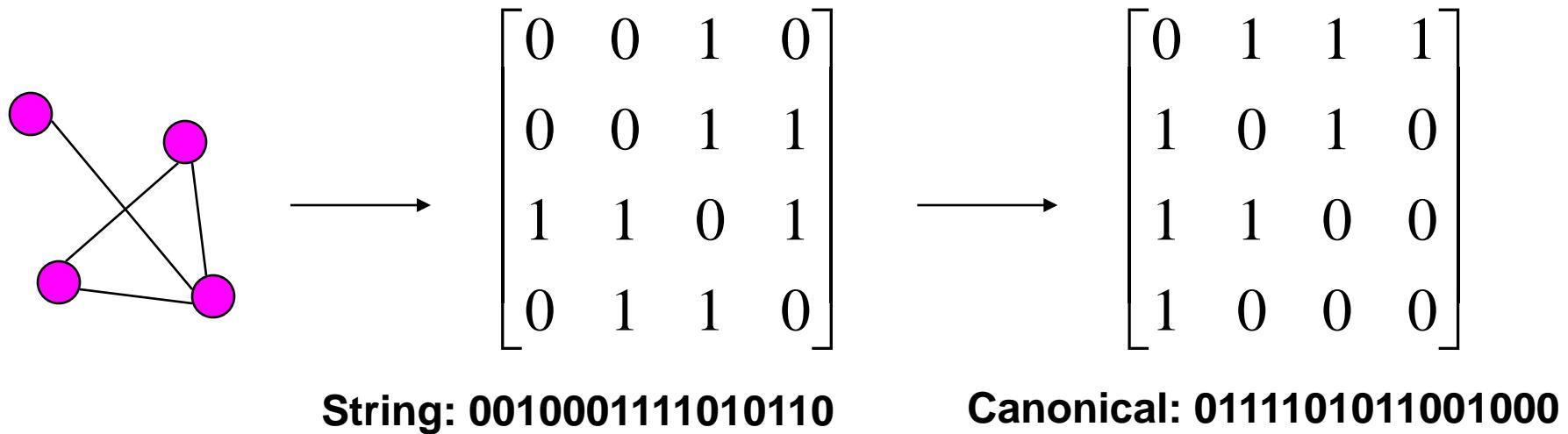


Graph Isomorphism

- Test for graph isomorphism is needed:
 - During candidate generation step, to determine whether a candidate has been generated
 - During candidate pruning step, to check whether its $(k-1)$ -subgraphs are frequent
 - During candidate counting, to check whether a candidate is contained within another graph

Graph Isomorphism

- Use canonical labeling to handle isomorphism
 - Map each graph into an ordered string representation (known as its code) such that two isomorphic graphs will be mapped to the same canonical encoding
 - Example:
 - ◆ Lexicographically largest adjacency matrix



Data Mining Cluster Analysis: Basic Concepts and Algorithms

Lecture Notes for Chapter 8

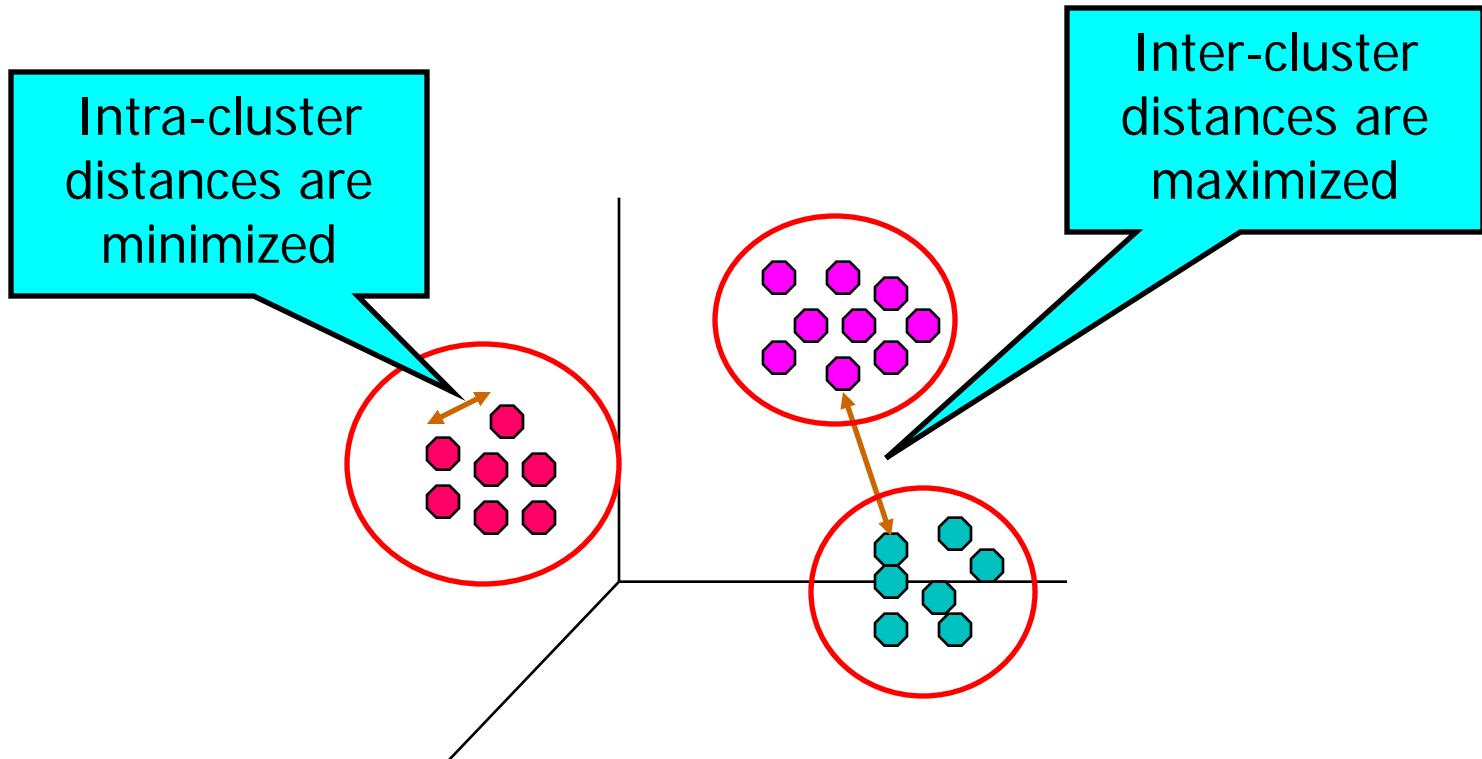
Introduction to Data Mining

by

Tan, Steinbach, Kumar

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



Applications of Cluster Analysis

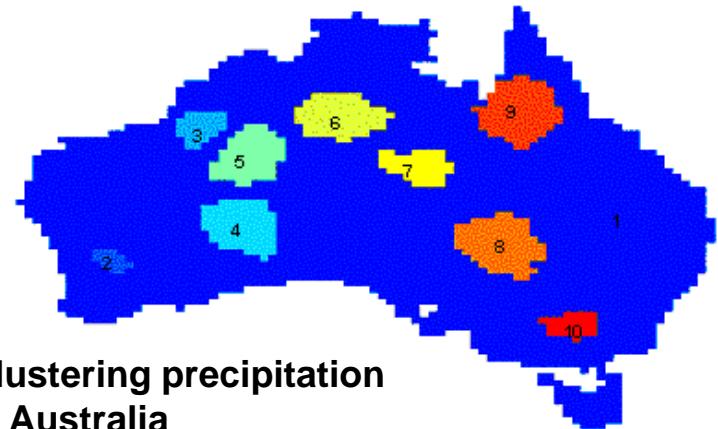
● Understanding

- 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

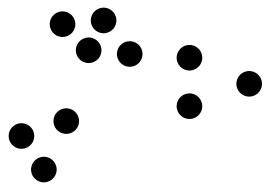
- Reduce the size of large data sets



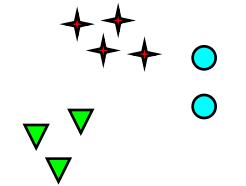
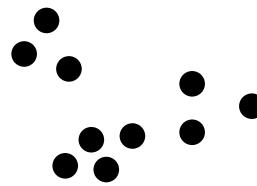
What is not Cluster Analysis?

- 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
- Graph partitioning
 - Some mutual relevance and synergy, but areas are not identical

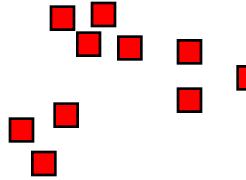
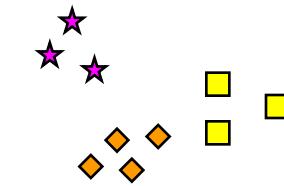
Notion of a Cluster can be Ambiguous



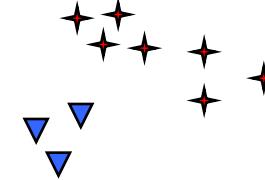
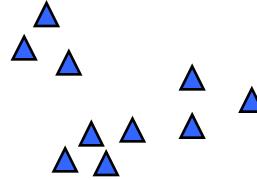
How many clusters?



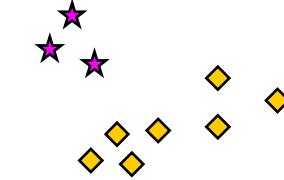
Six Clusters



Two Clusters



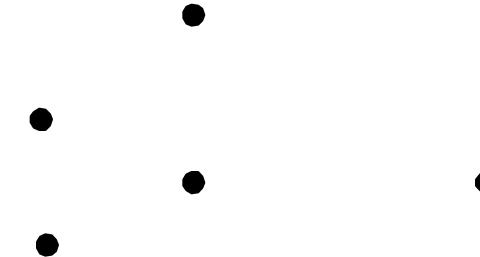
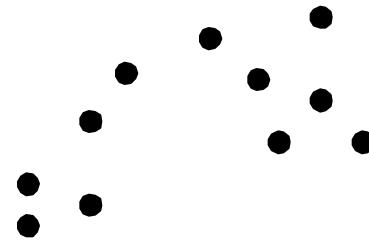
Four Clusters



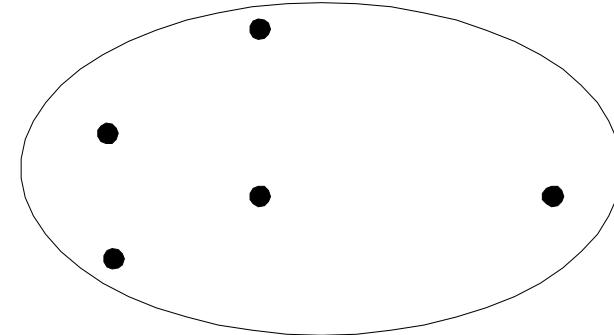
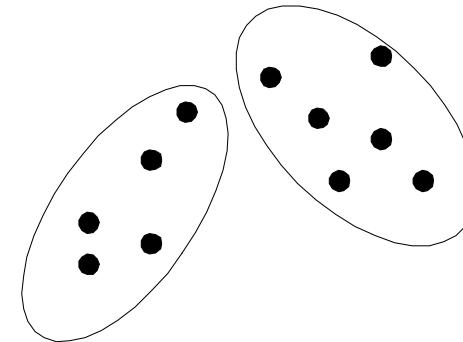
Types of Clusterings

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

Partitional Clustering

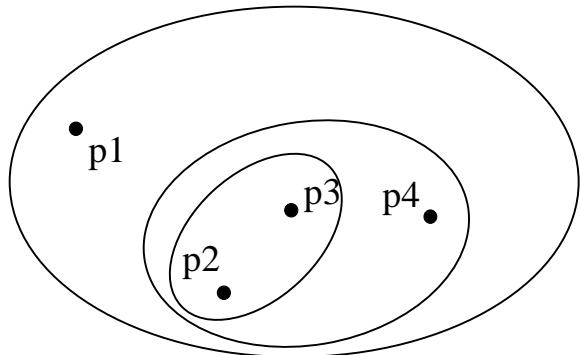


Original Points

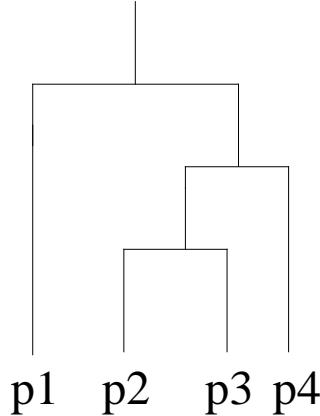


A Partitional Clustering

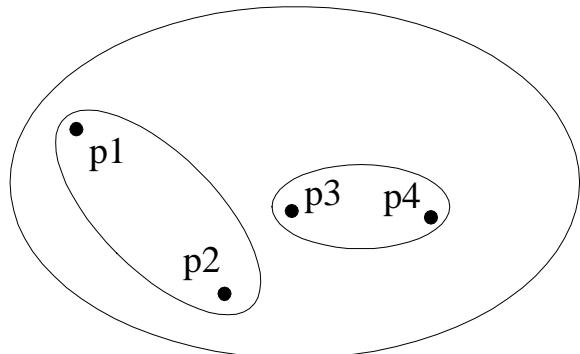
Hierarchical Clustering



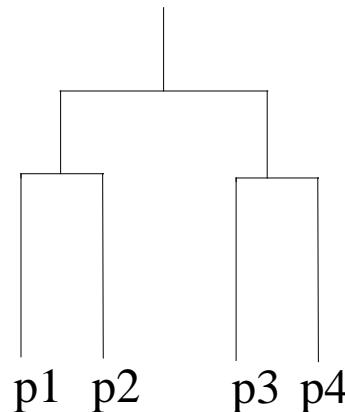
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Sets of Clusters

- Exclusive versus non-exclusive
 - In non-exclusive clusterings, points may belong to multiple clusters.
 - Can represent multiple classes or ‘border’ points
- Fuzzy versus non-fuzzy
 - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
 - Weights must sum to 1
 - Probabilistic clustering has similar characteristics
- 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

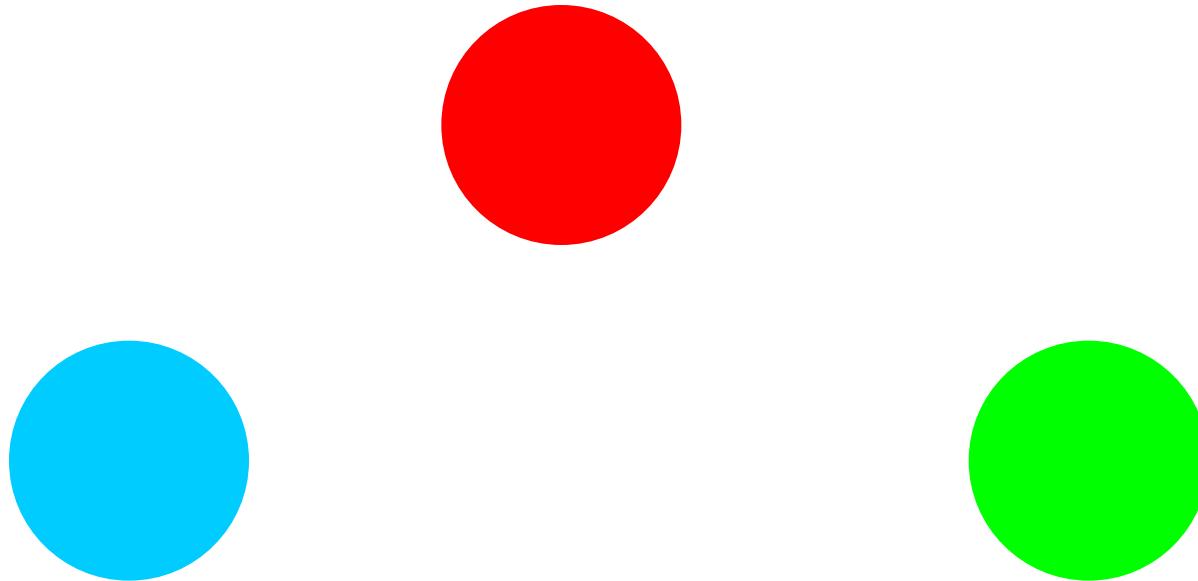
Types of Clusters

- 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

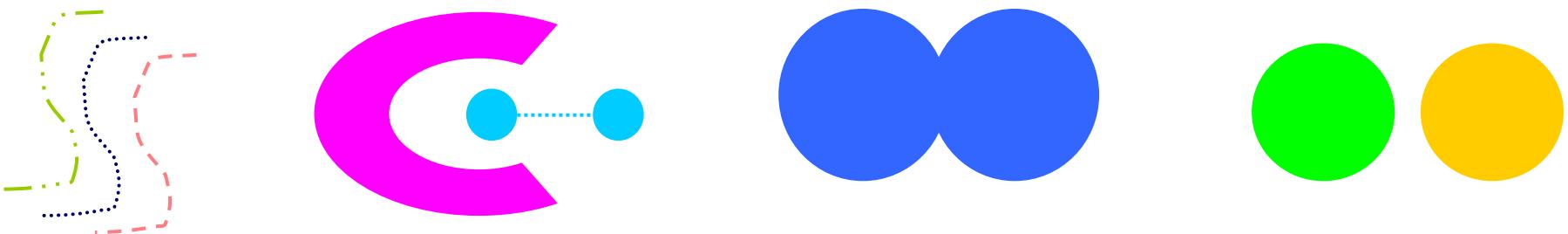
- A cluster is a set of objects such that an object in a cluster is closer (more similar) to the “center” of a cluster, than to the center of any other cluster
- The center of a cluster is often a **centroid**, the average of all the points in the cluster, or a **medoid**, the most “representative” point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

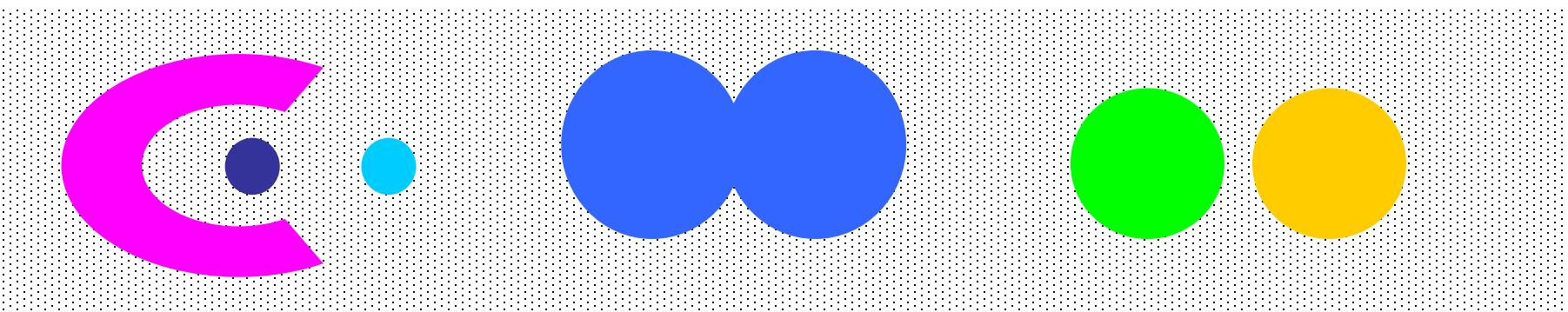


8 contiguous clusters

Types of Clusters: Density-Based

- Density-based

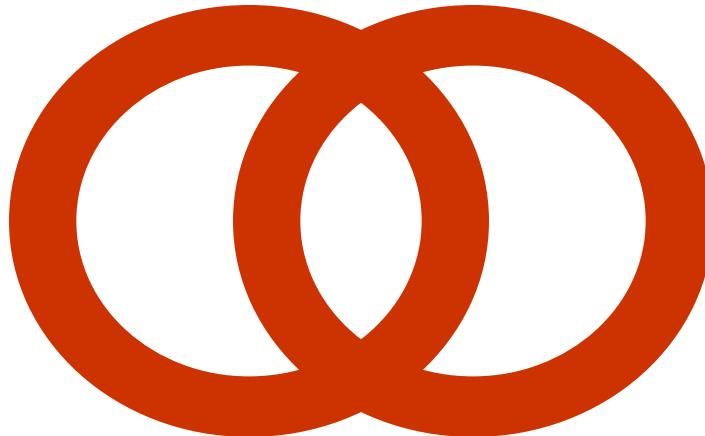
- A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
- Used when the clusters are irregular or intertwined, and when noise and outliers are present.



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

Characteristics of the Input Data Are Important

- 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

Clustering Algorithms

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

K-means Clustering

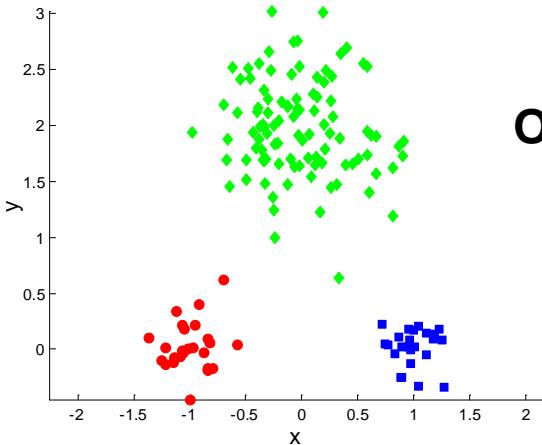
- Partitional clustering approach
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K , must be specified
- The basic algorithm is very simple

-
- 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
-

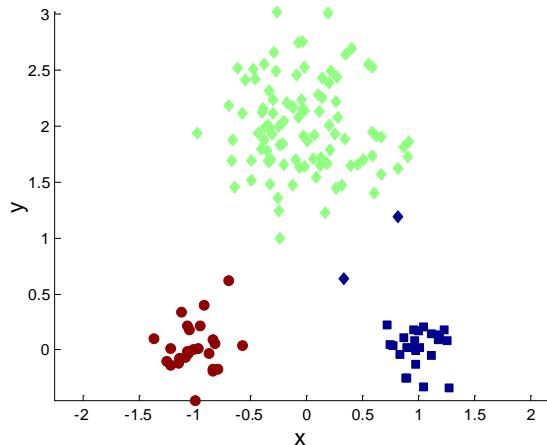
K-means Clustering – Details

- 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, correlation, 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

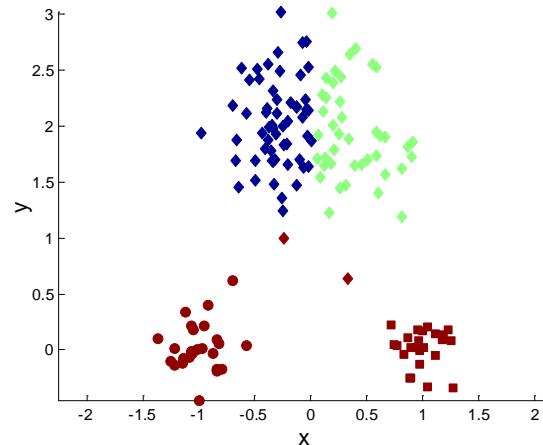
Two different K-means Clusterings



Original Points

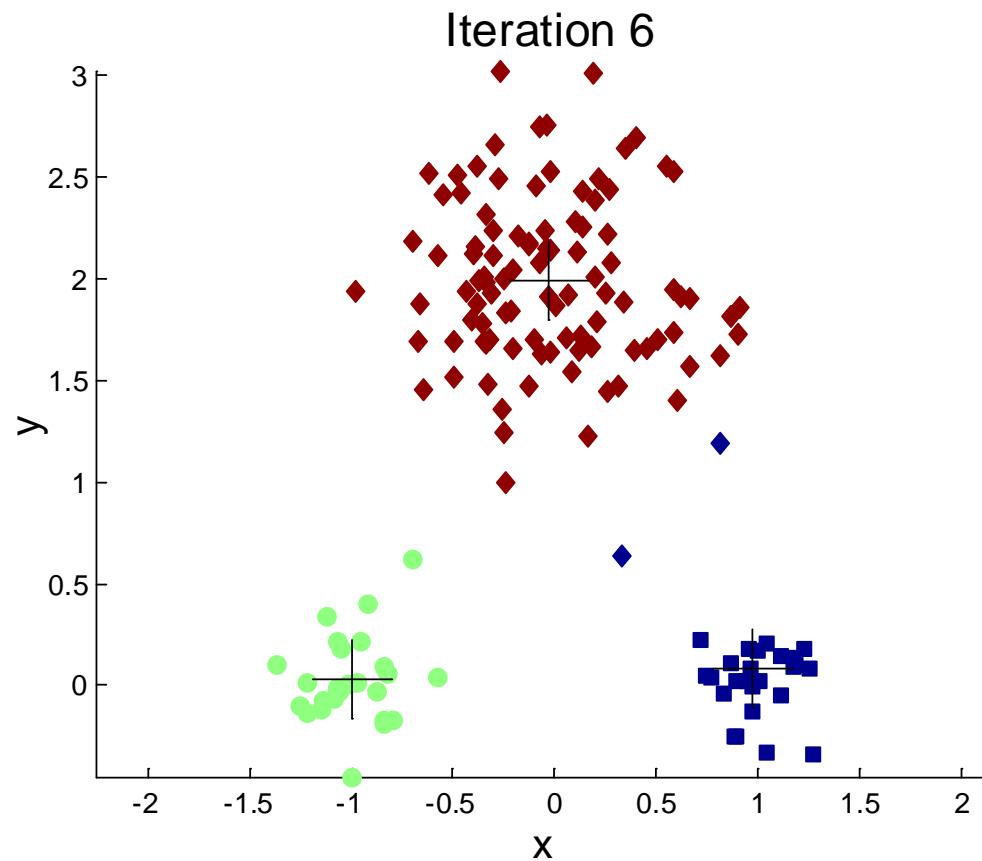


Optimal Clustering

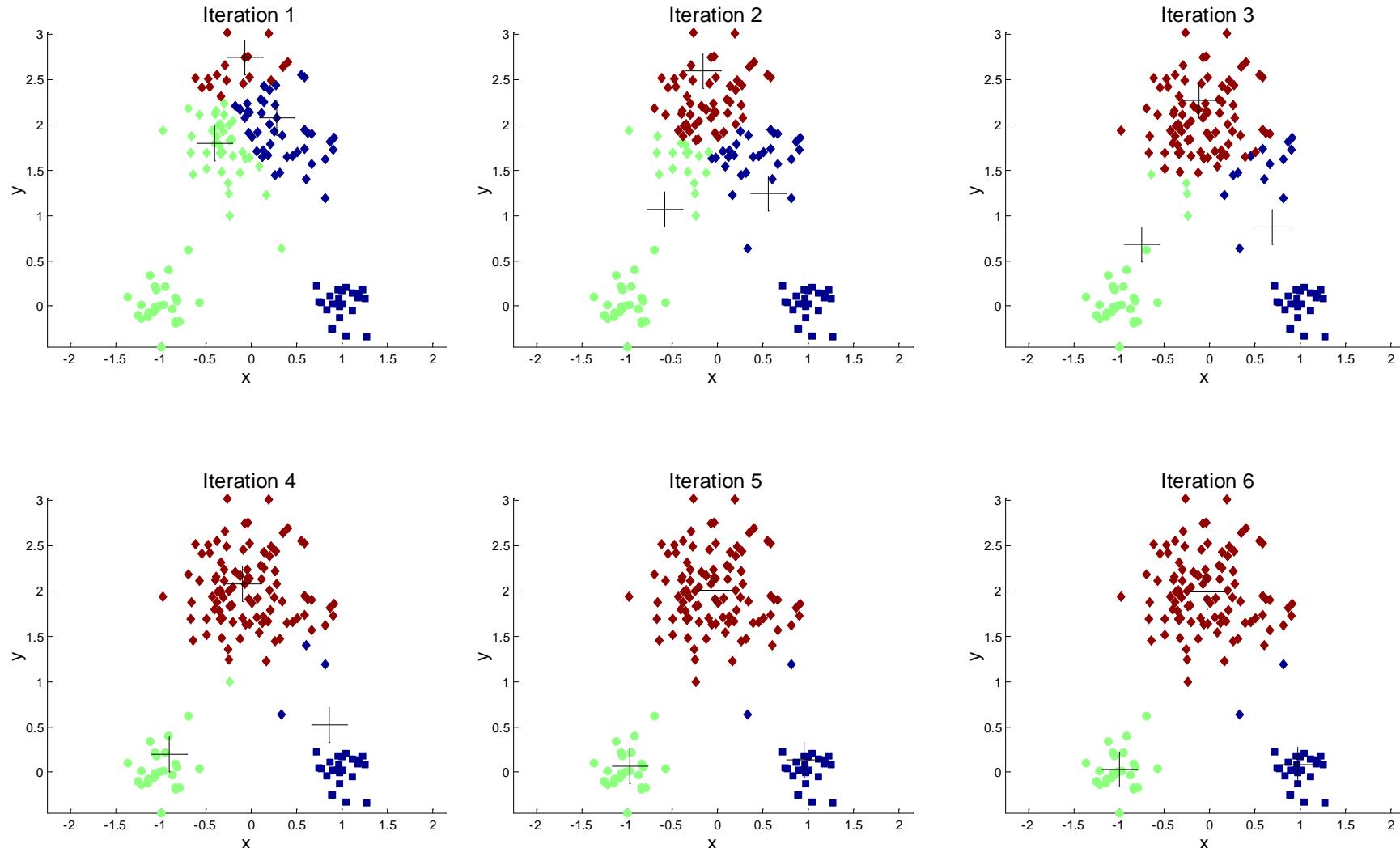


Sub-optimal Clustering

Importance of Choosing Initial Centroids



Importance of Choosing Initial Centroids



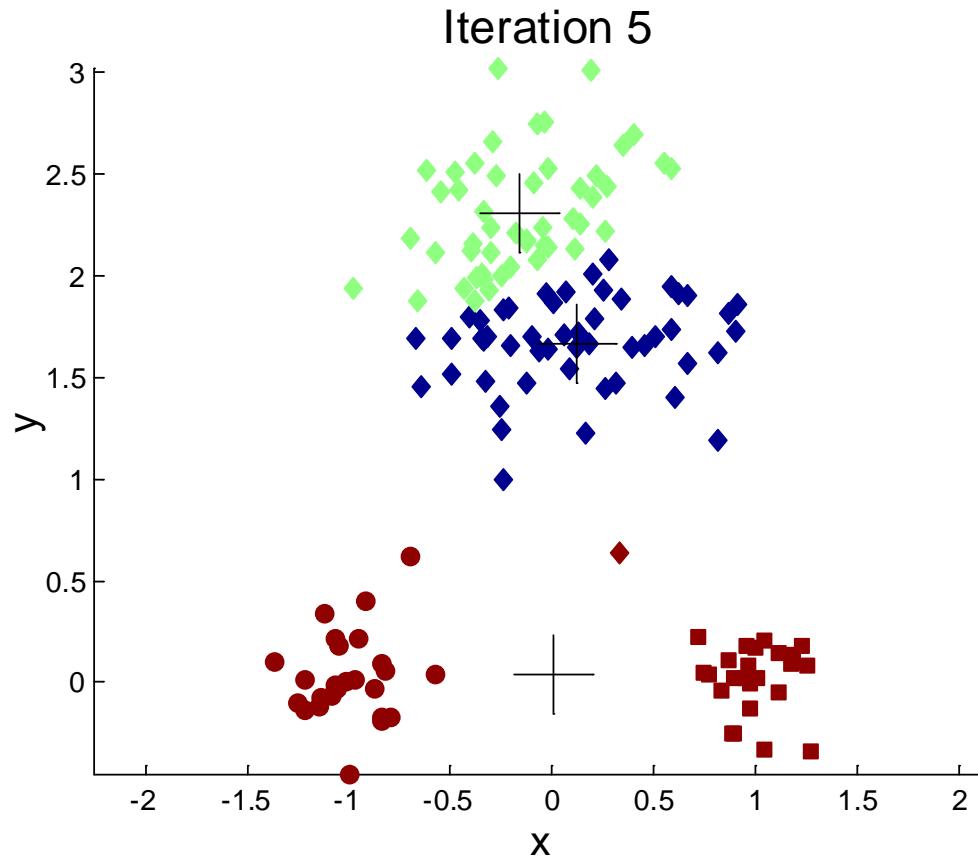
Evaluating K-means Clusters

- 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.

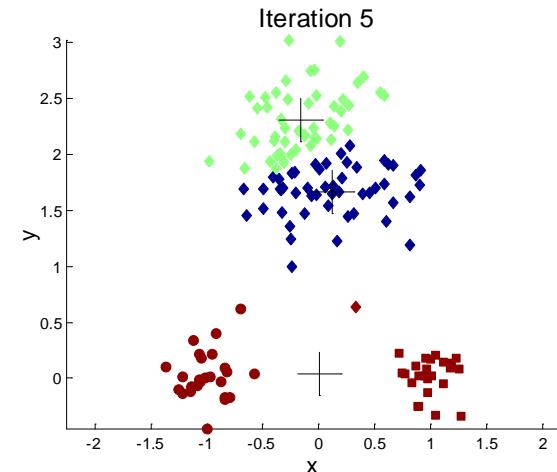
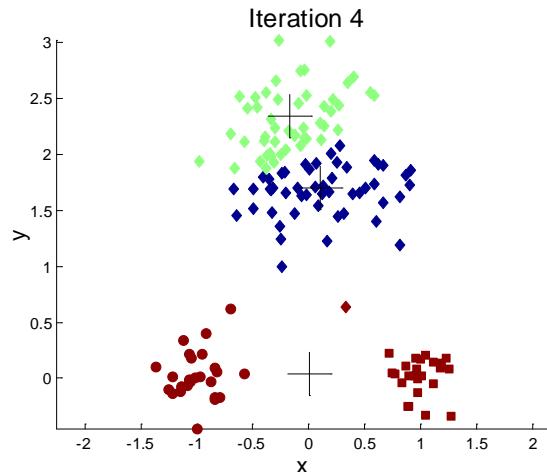
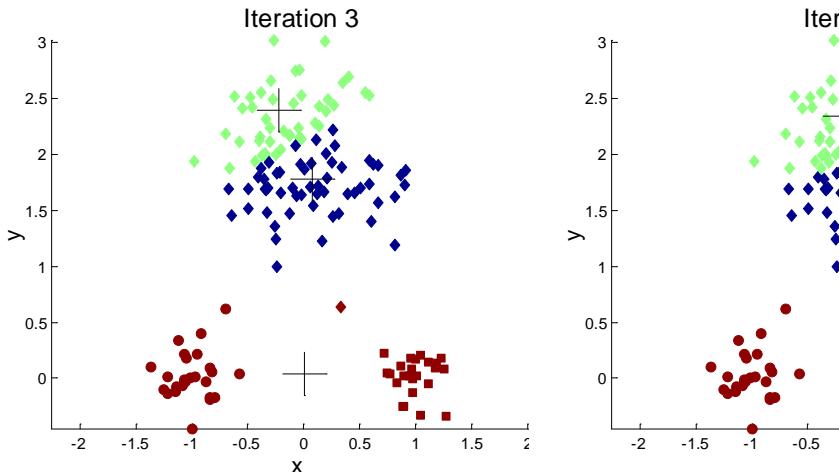
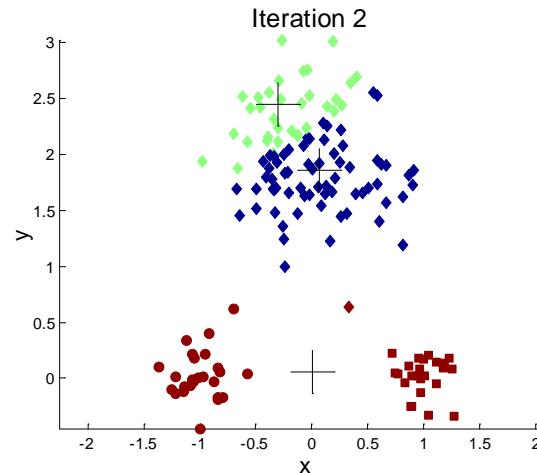
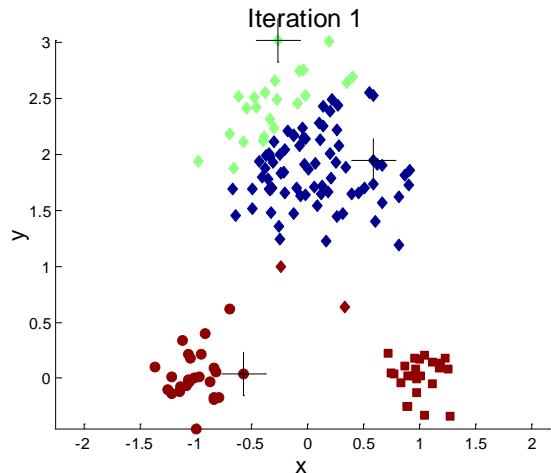
$$SSE = \sum_{i=1}^K \sum_{x \in C_i} dist^2(m_i, x)$$

- 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 ...



Problems with Selecting Initial Points

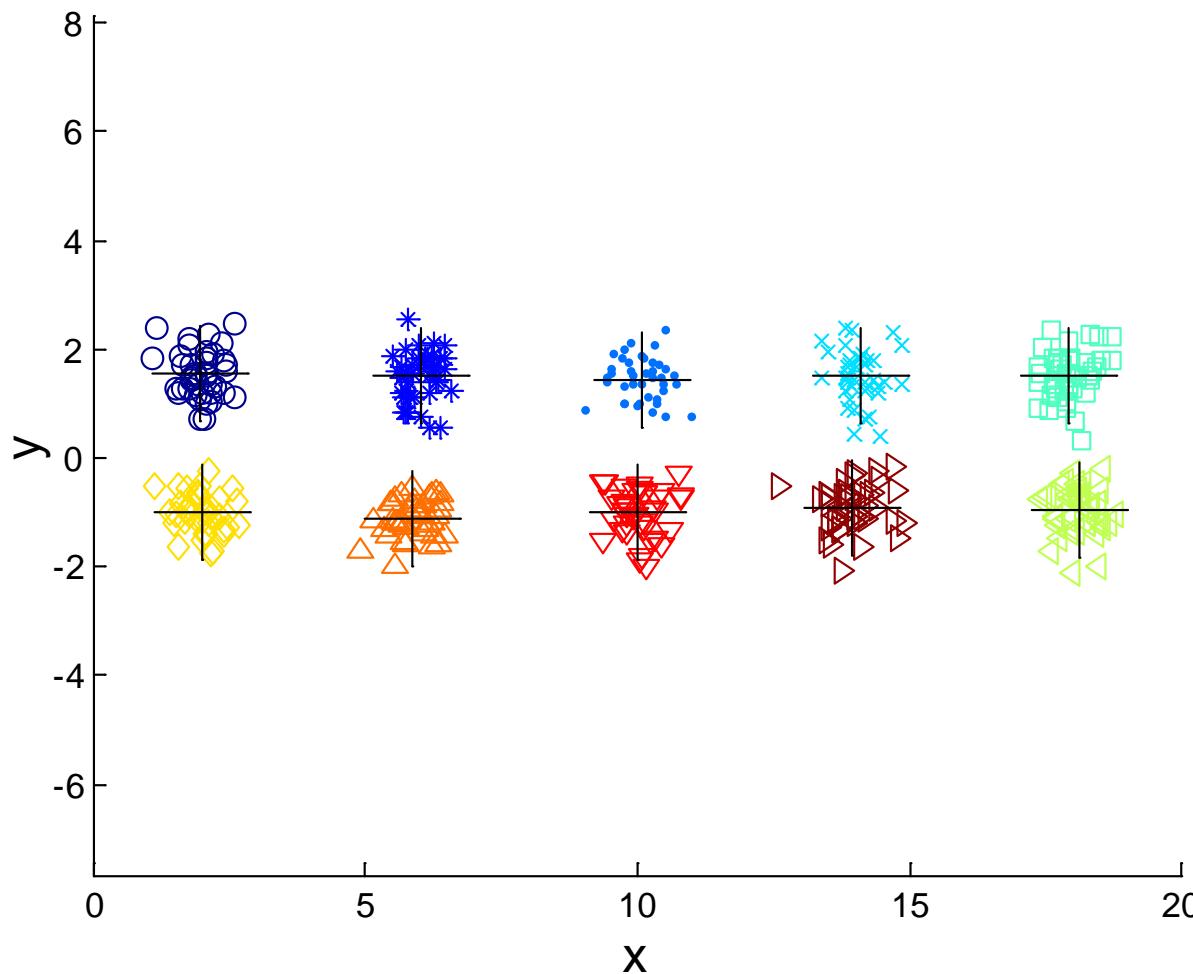
- If there are K ‘real’ clusters then the chance of selecting one centroid from each cluster is small.
 - Chance is relatively small when K is large
 - If clusters are the same size, n , then

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

- 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

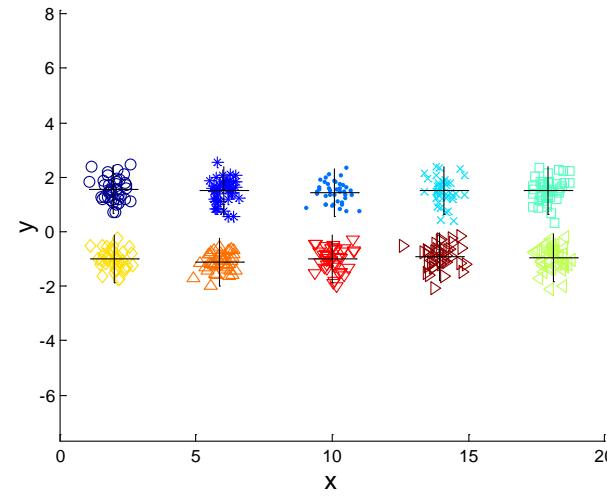
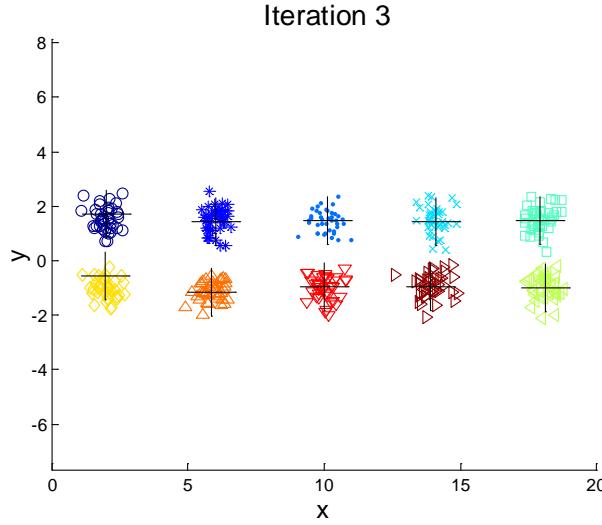
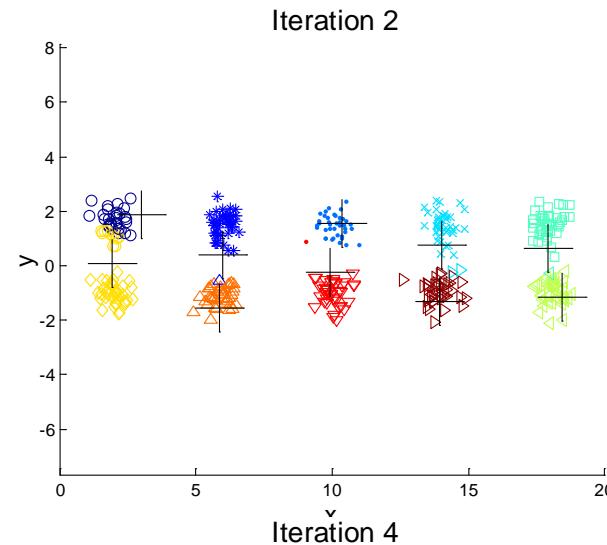
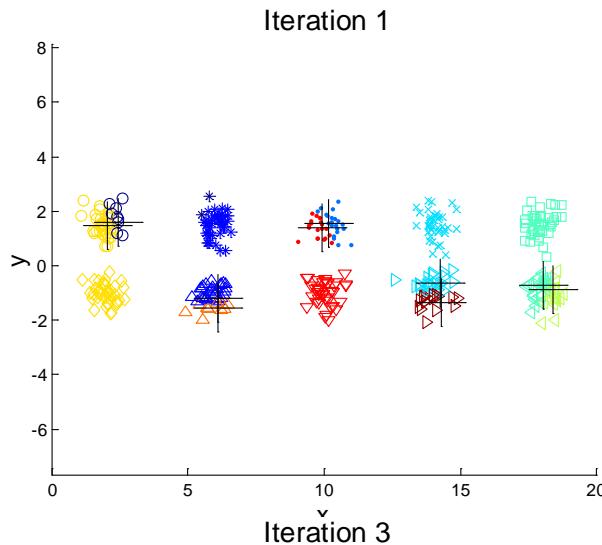
10 Clusters Example

Iteration 4



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

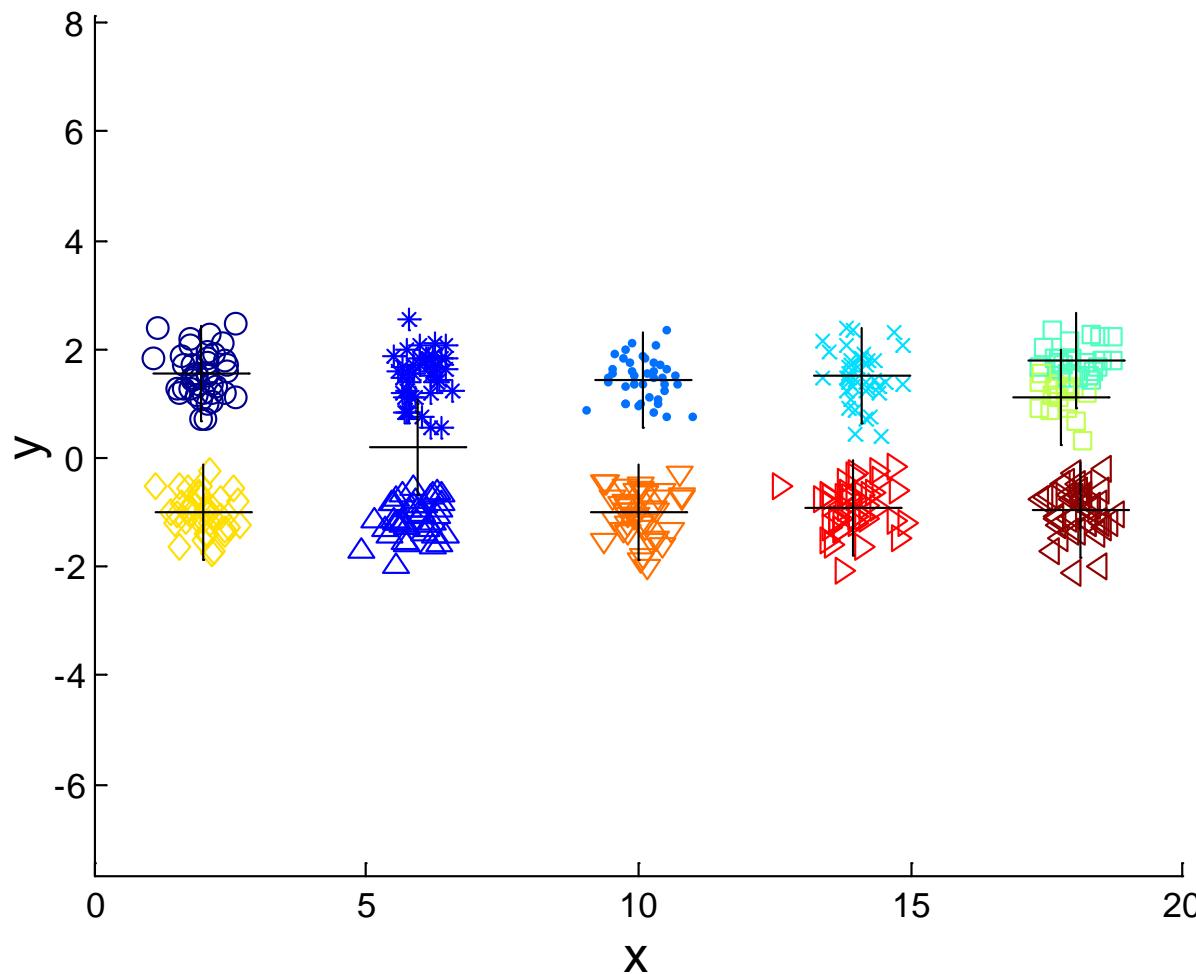
10 Clusters Example



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

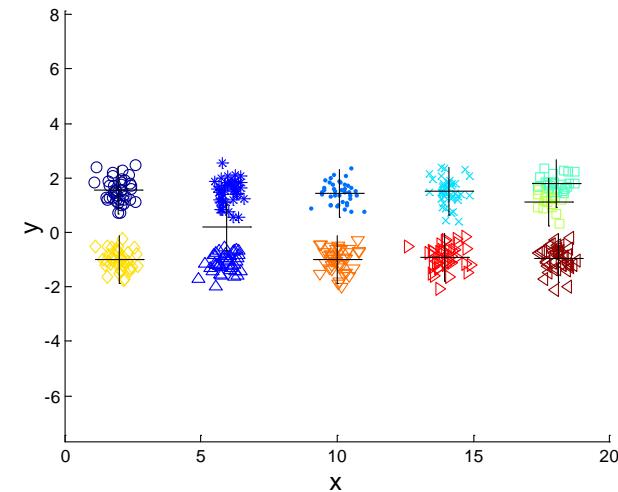
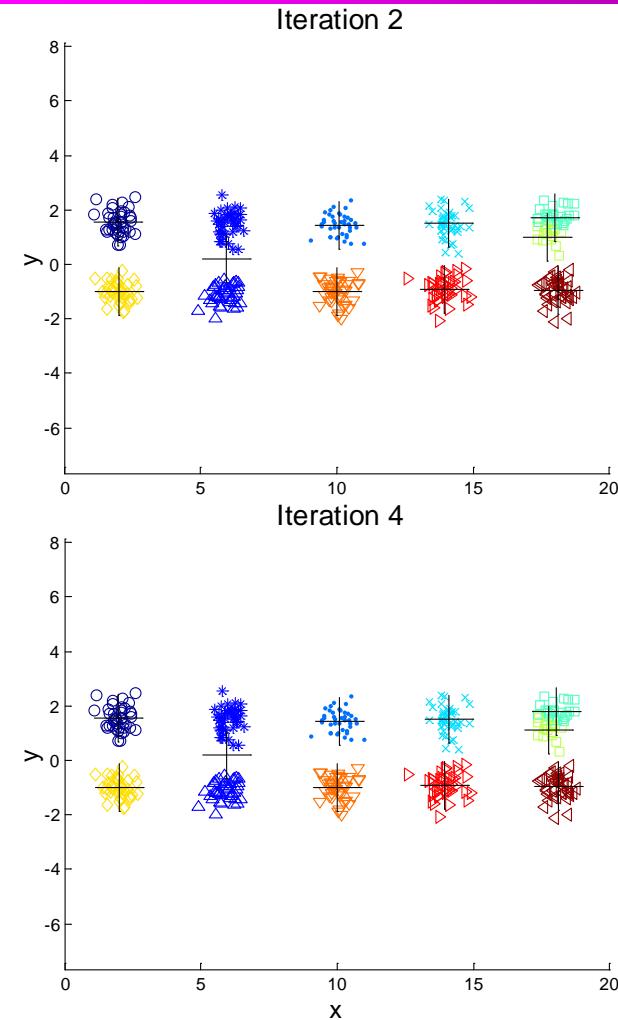
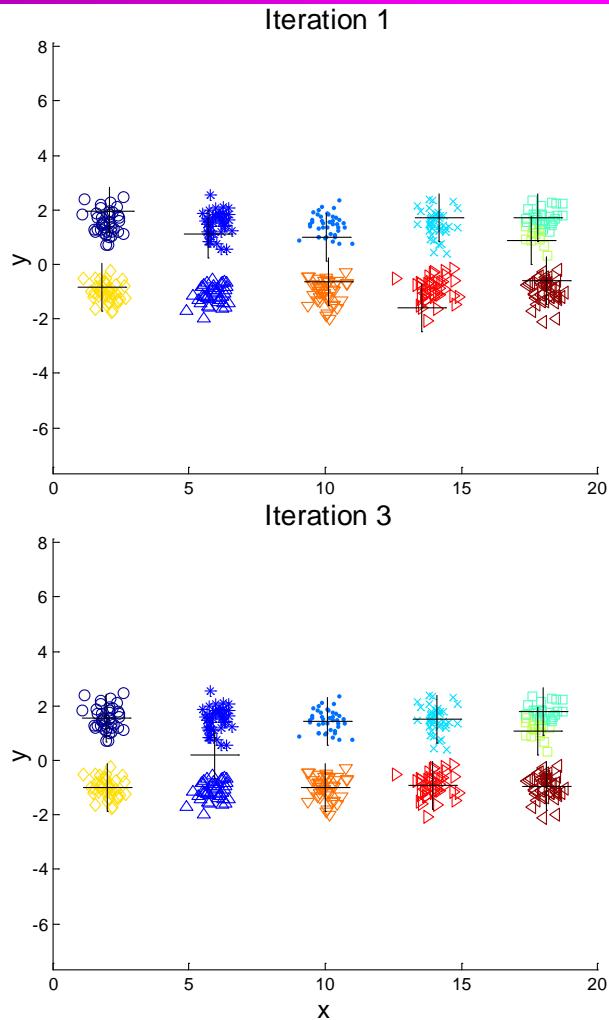
10 Clusters Example

Iteration 4



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

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 on your side
- 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
- Bisecting K-means
 - Not as susceptible to initialization issues

Handling Empty Clusters

- Basic K-means algorithm can yield empty clusters
- Several strategies
 - Choose the point that contributes most to SSE
 - Choose a point from the cluster with the highest SSE
 - If there are several empty clusters, the above can be repeated several times.

Updating Centers Incrementally

- In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid
- An alternative is to update the centroids after each assignment (incremental approach)
 - Each assignment updates zero or two centroids
 - More expensive
 - Introduces an order dependency
 - Never get an empty cluster
 - Can use “weights” to change the impact

Pre-processing and Post-processing

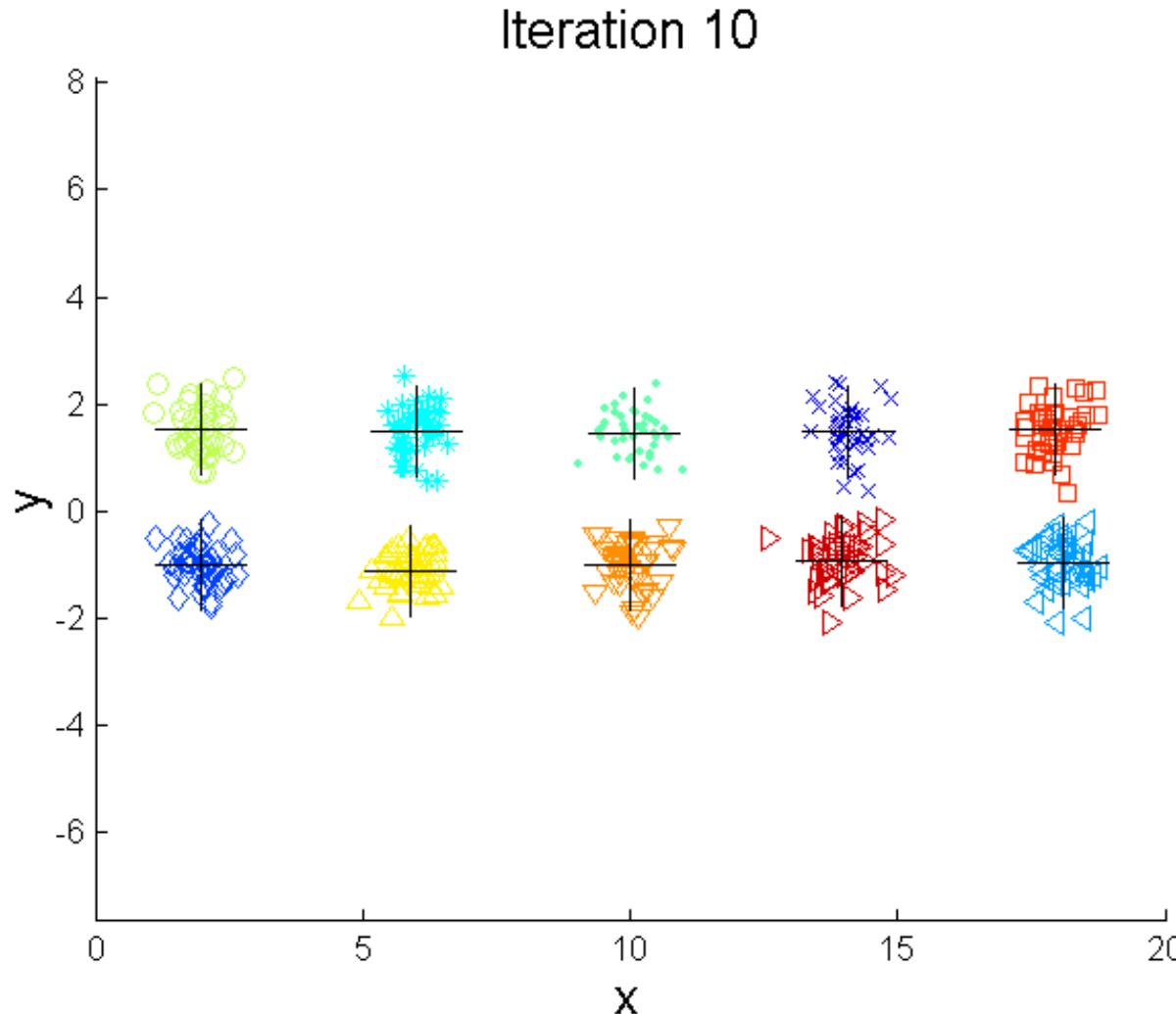
- Pre-processing
 - Normalize the data
 - Eliminate outliers
- Post-processing
 - Eliminate small clusters that may represent outliers
 - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
 - Merge clusters that are ‘close’ and that have relatively low SSE
 - Can use these steps during the clustering process
 - ◆ ISODATA

Bisecting K-means

- Bisecting K-means algorithm
 - Variant of K-means that can produce a partitional or a hierarchical clustering

```
1: Initialize the list of clusters to contain the cluster containing all points.  
2: repeat  
3:   Select a cluster from the list of clusters  
4:   for  $i = 1$  to number_of_iterations do  
5:     Bisect the selected cluster using basic K-means  
6:   end for  
7:   Add the two clusters from the bisection with the lowest SSE to the list of clusters.  
8: until Until the list of clusters contains  $K$  clusters
```

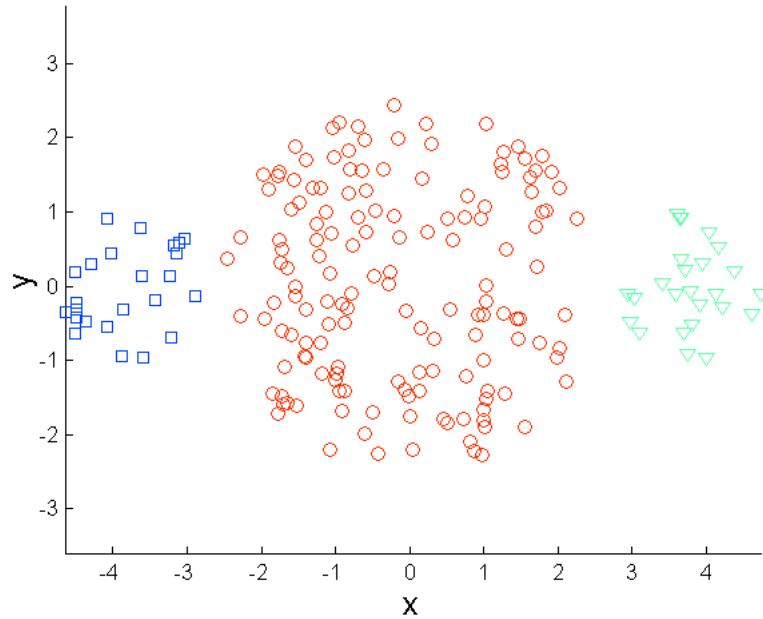
Bisecting K-means Example



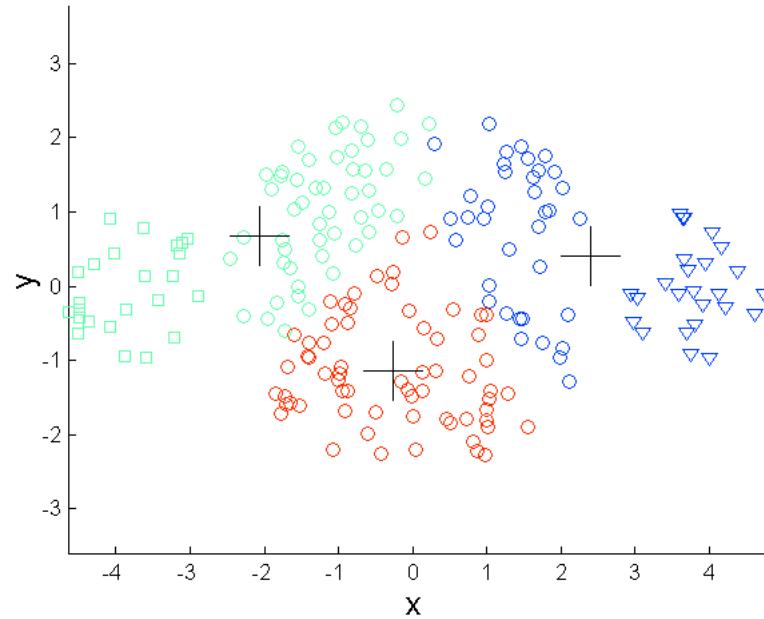
Limitations of K-means

- 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

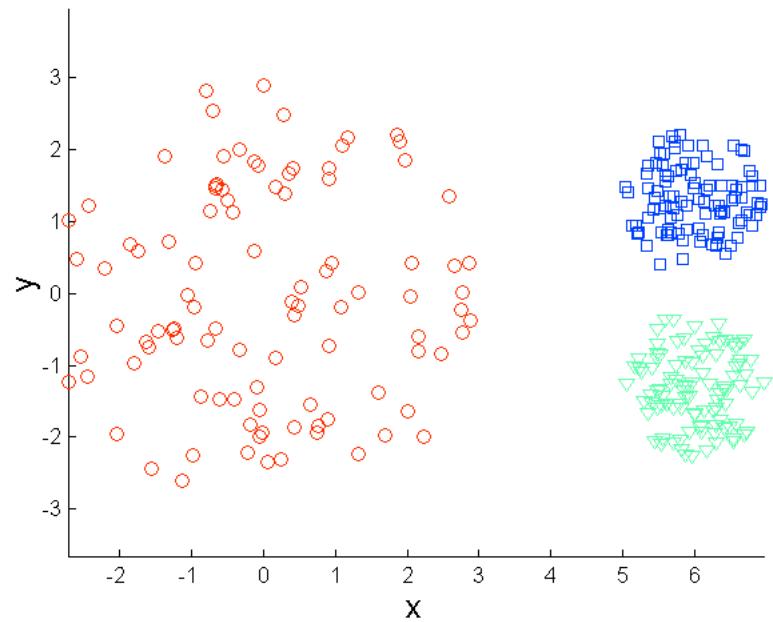


Original Points

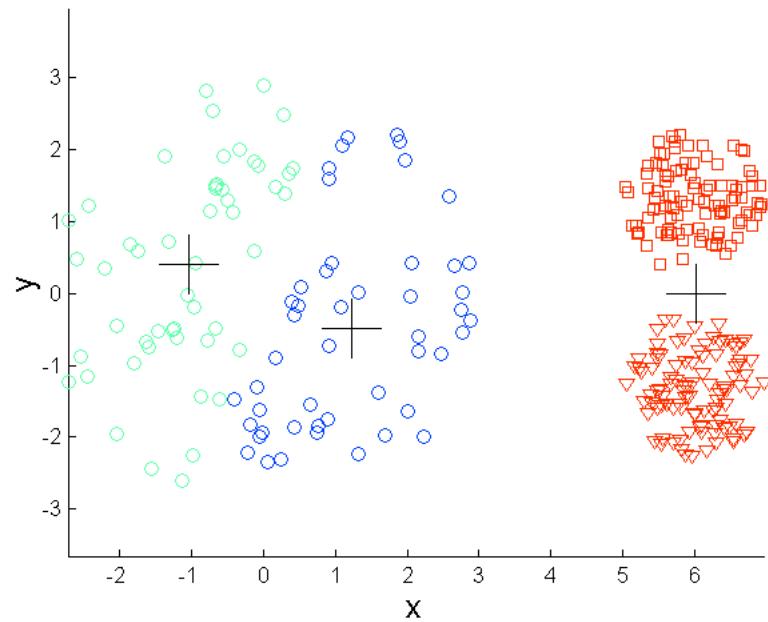


K-means (3 Clusters)

Limitations of K-means: Differing Density

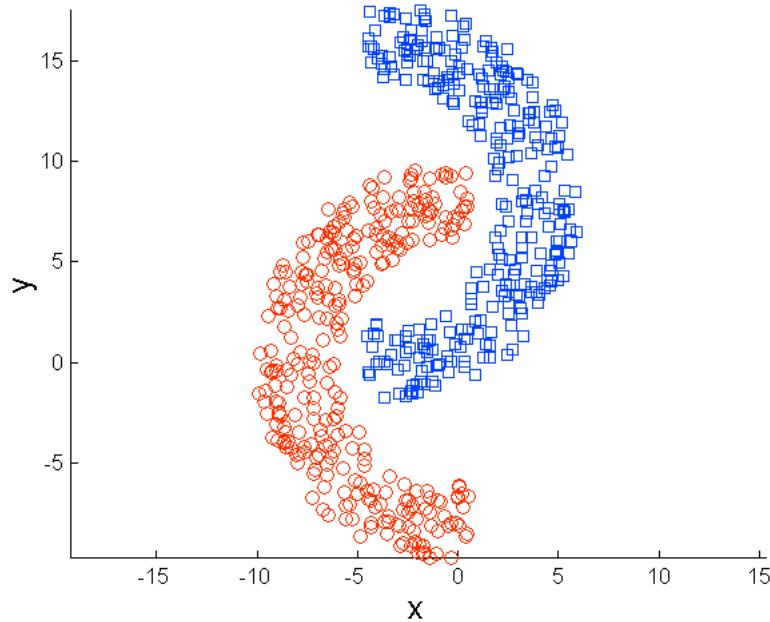


Original Points

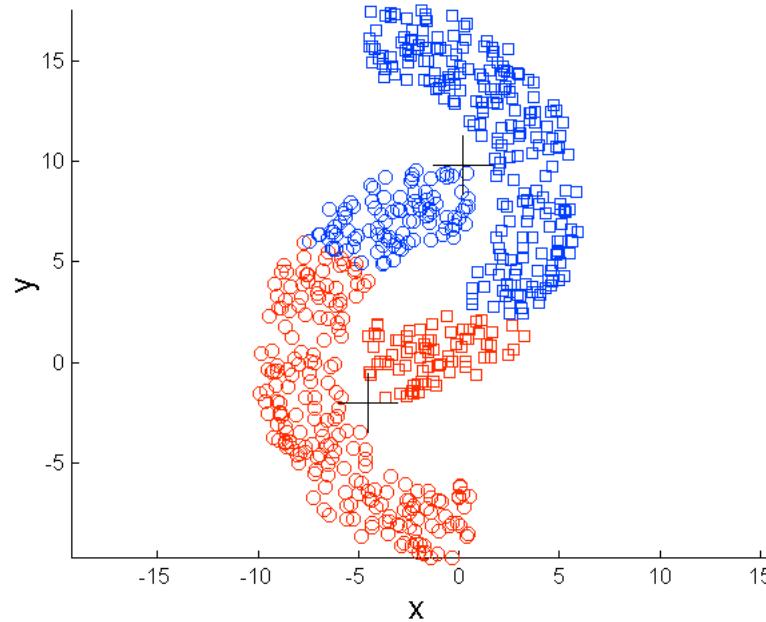


K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

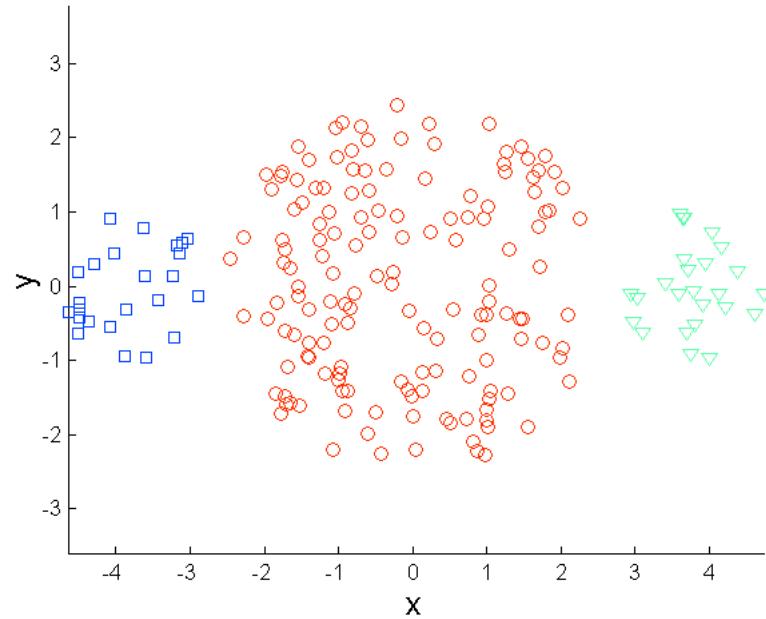


Original Points

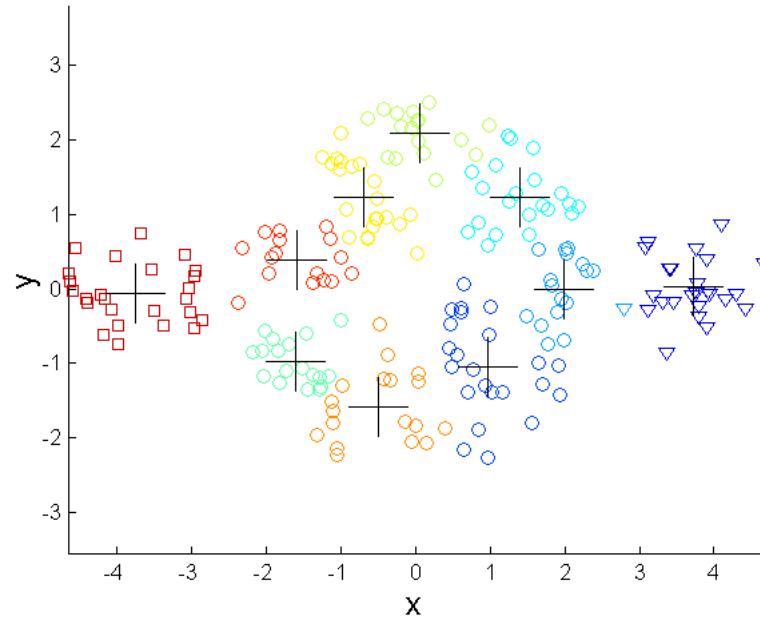


K-means (2 Clusters)

Overcoming K-means Limitations



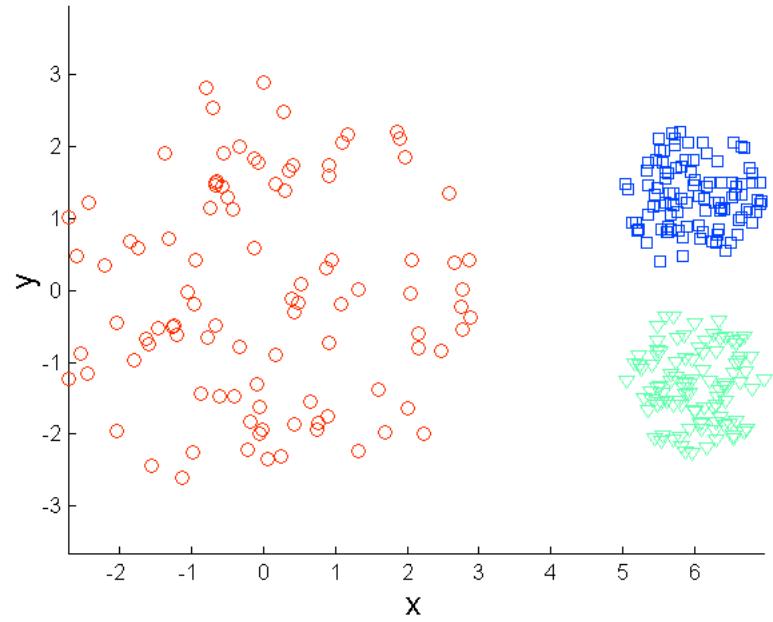
Original Points



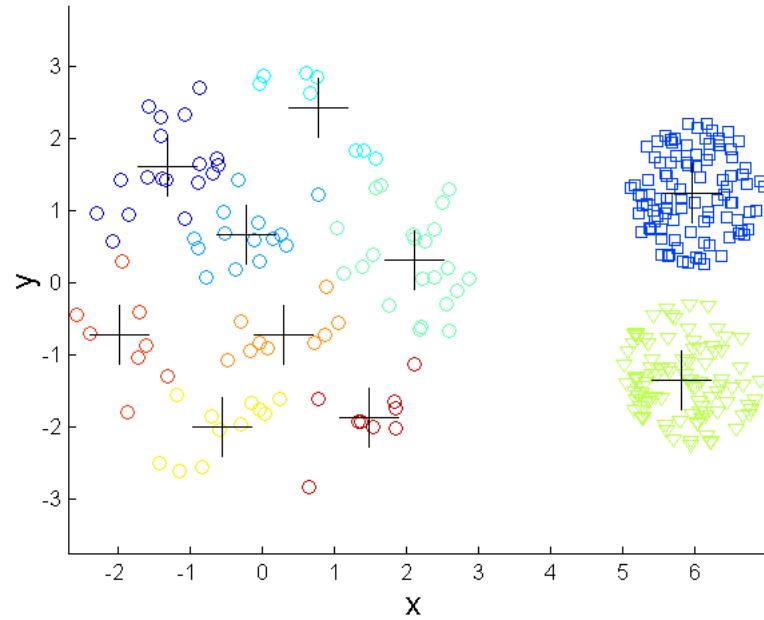
K-means Clusters

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

Overcoming K-means Limitations

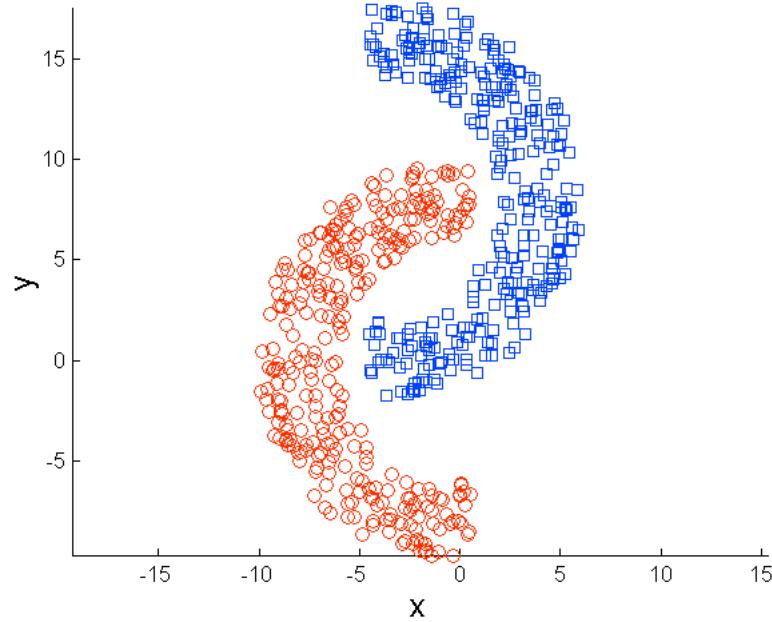


Original Points

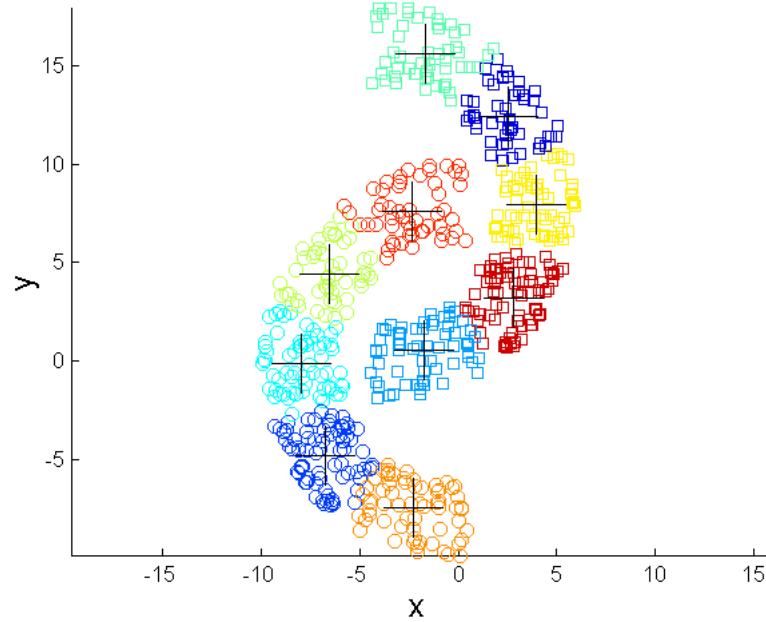


K-means Clusters

Overcoming K-means Limitations



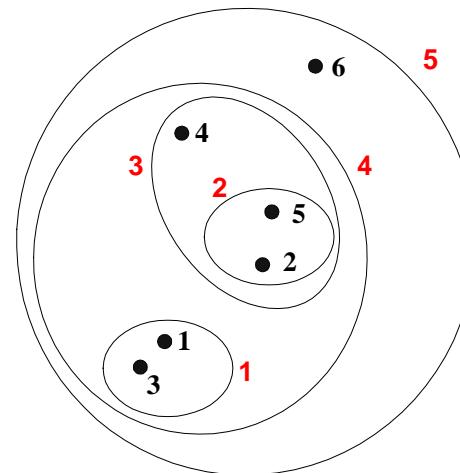
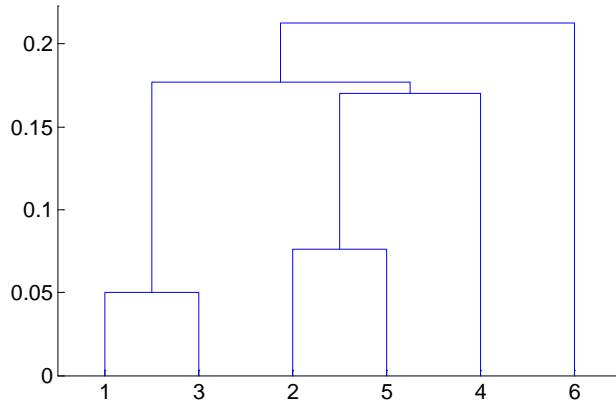
Original Points



K-means Clusters

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

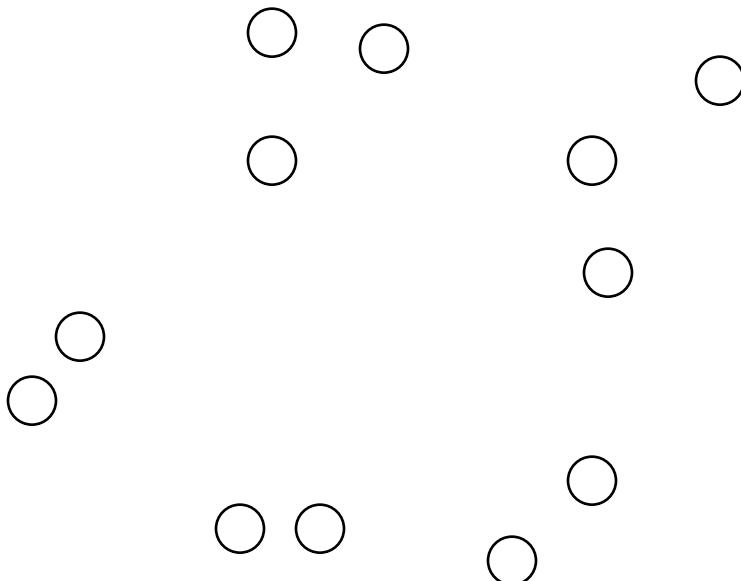
- Two main types of hierarchical clustering
 - Agglomerative:
 - ◆ Start with the points as individual clusters
 - ◆ At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
 - Divisive:
 - ◆ Start with one, all-inclusive cluster
 - ◆ At each step, split a cluster 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



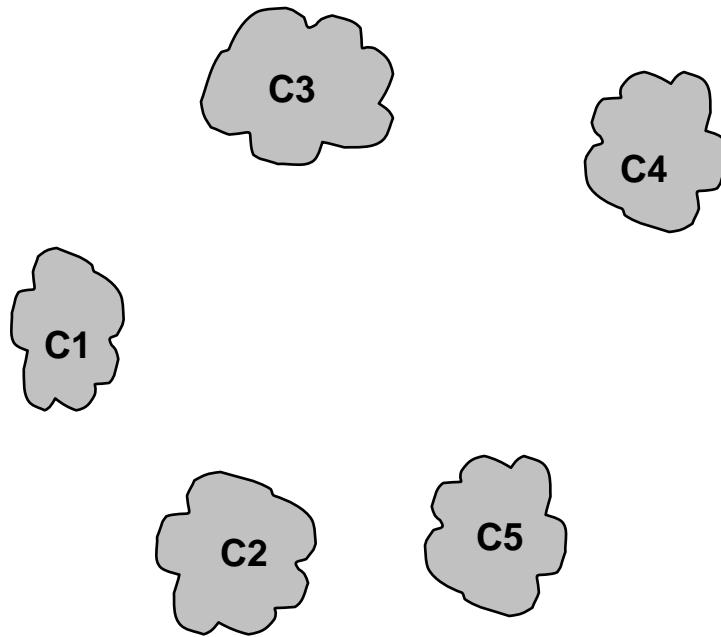
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						

Proximity Matrix



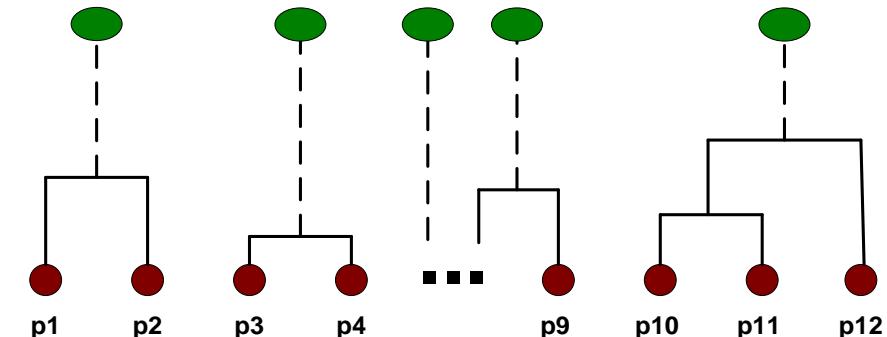
Intermediate Situation

- After some merging steps, we have some clusters



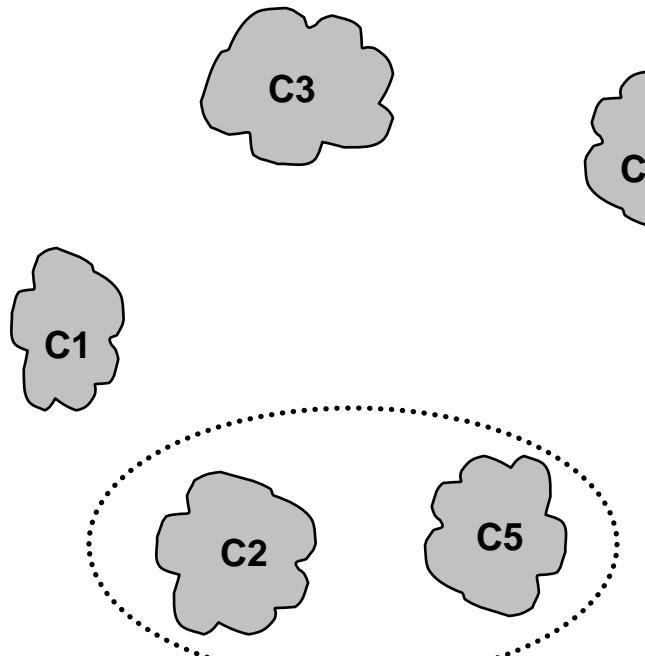
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



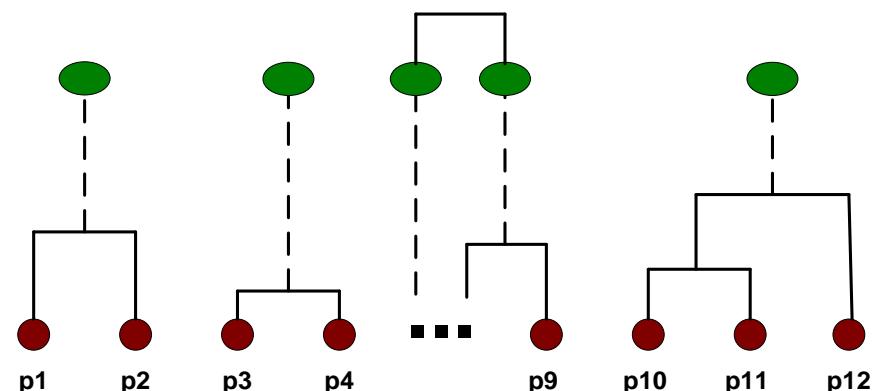
Intermediate Situation

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



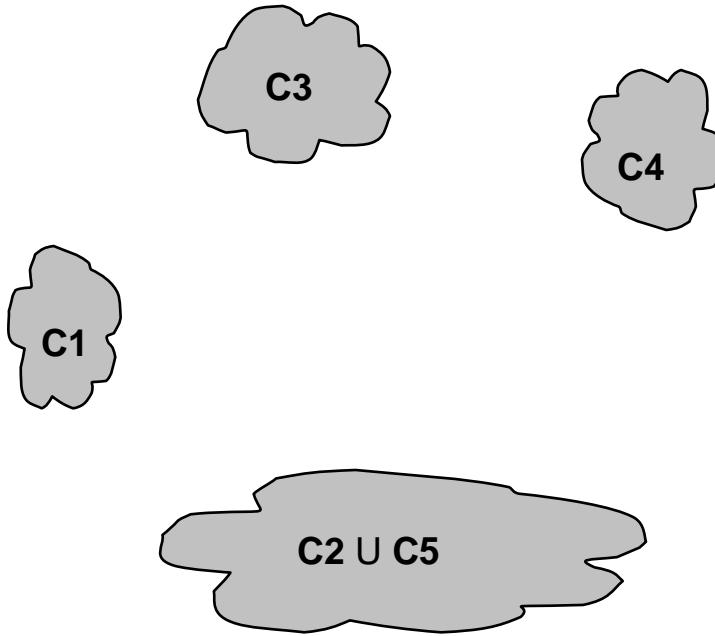
	C_1	C_2	C_3	C_4	C_5
C_1					
C_2					
C_3					
C_4					
C_5					

Proximity Matrix



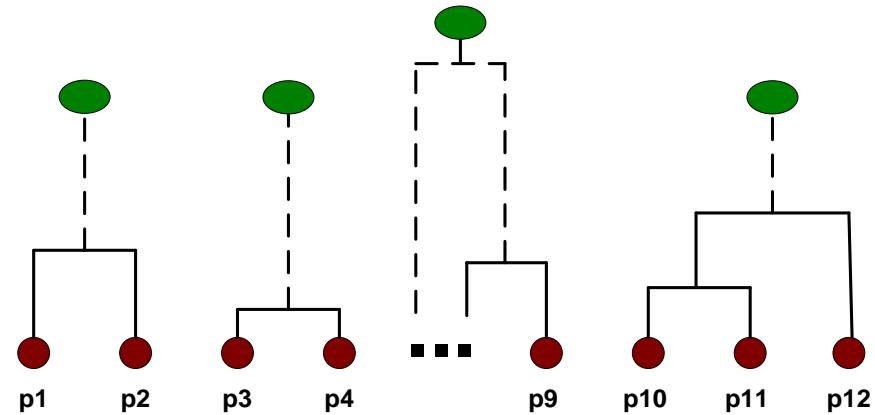
After Merging

- The question is “How do we update the proximity matrix?”

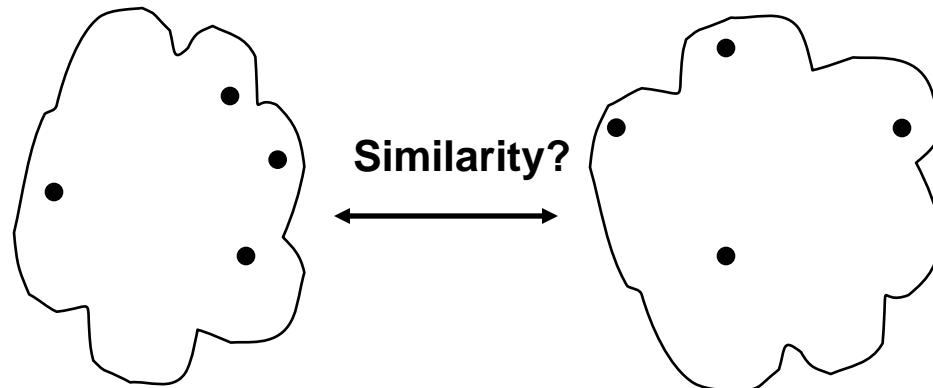


		C_2	U	C_1	C_5	C_3	C_4
		C_1	?				
$C_2 \cup C_5$?	?	?	?	?	?
C_3			?				
C_4		?					

Proximity Matrix



How to Define Inter-Cluster Similarity

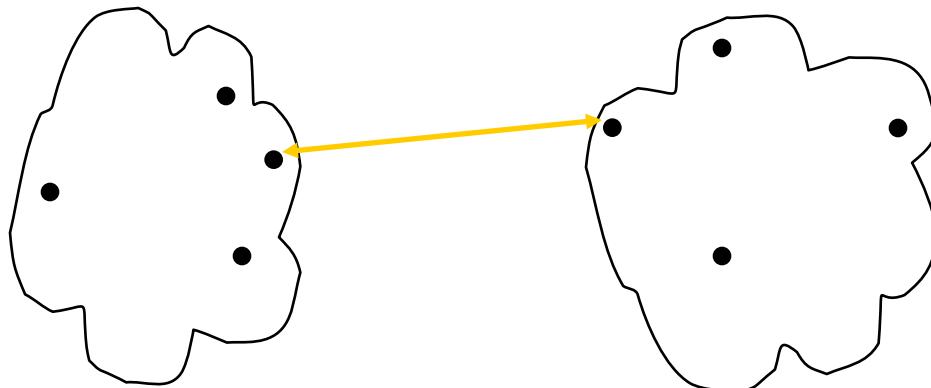


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

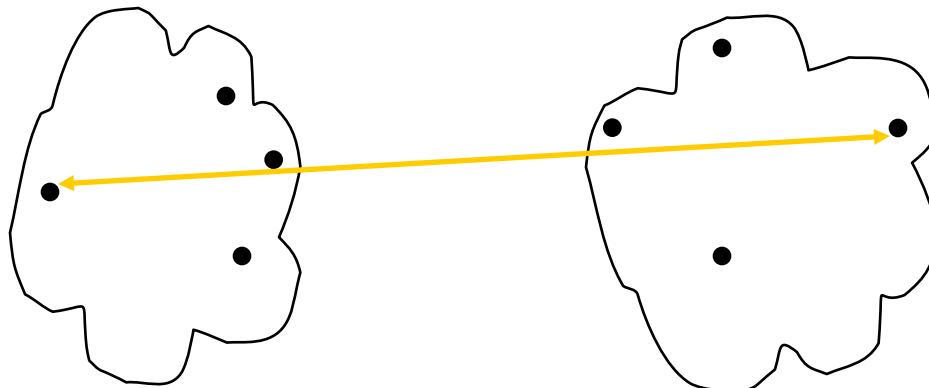


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

• **Proximity Matrix**

How to Define Inter-Cluster Similarity

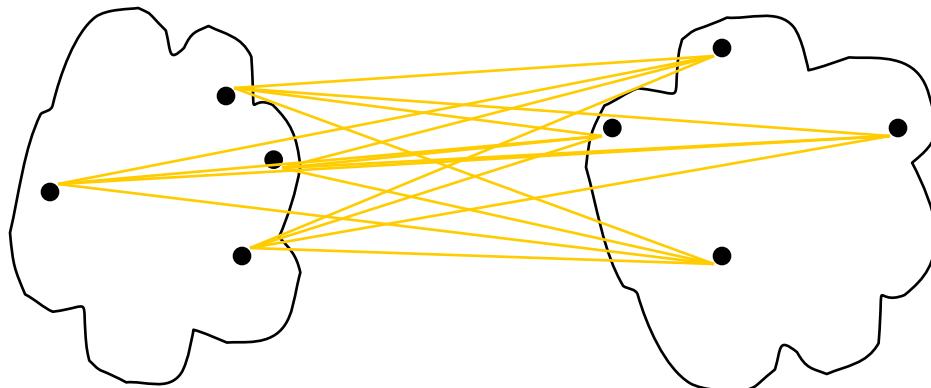


- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

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

Proximity Matrix

How to Define Inter-Cluster Similarity

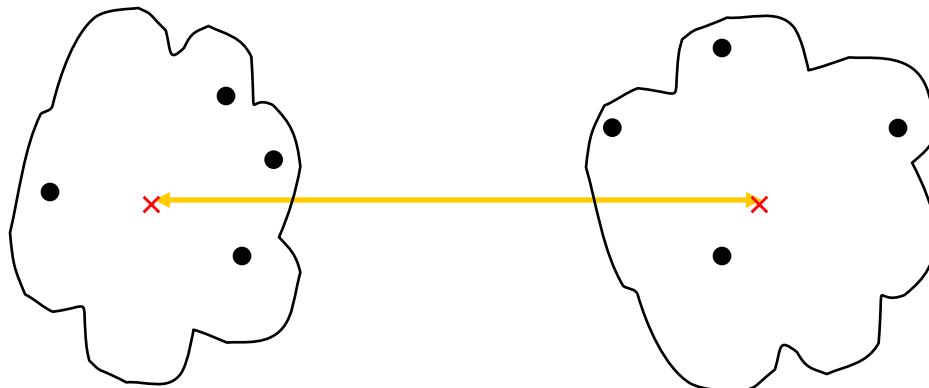


- MIN
- MAX
- **Group Average**
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

	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
 - Ward's Method uses squared error

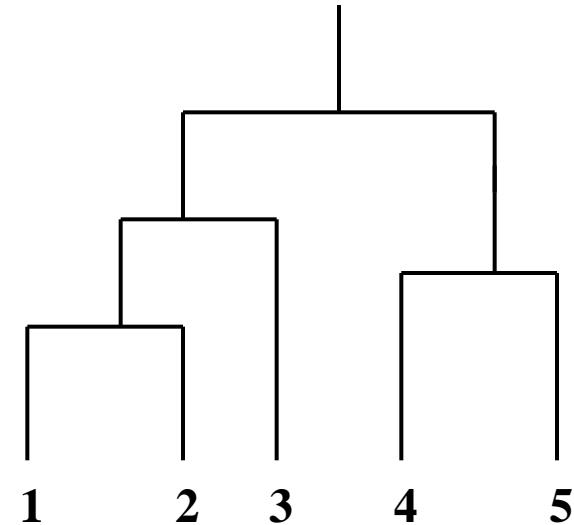
	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

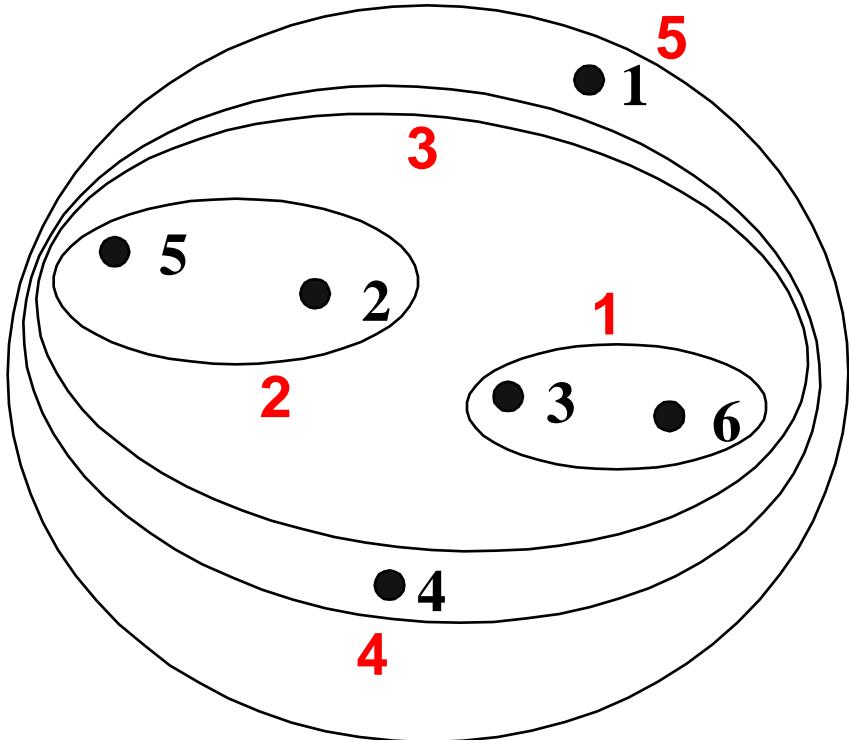
Cluster Similarity: MIN or Single Link

- 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.

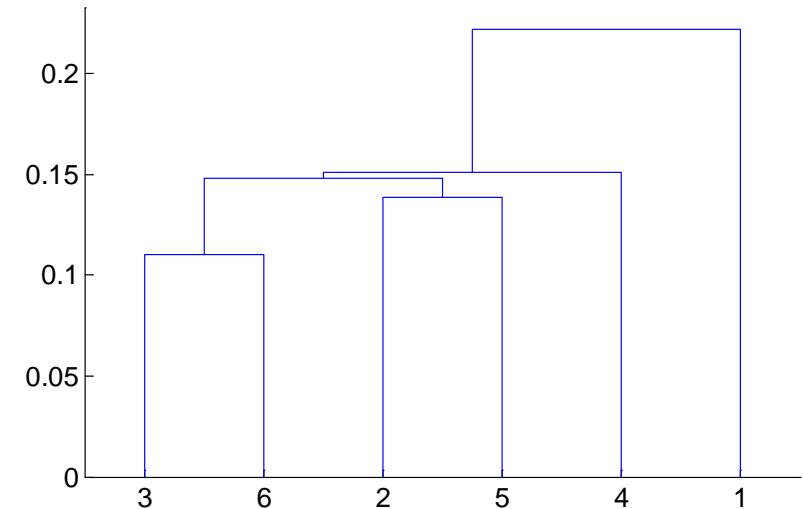
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MIN

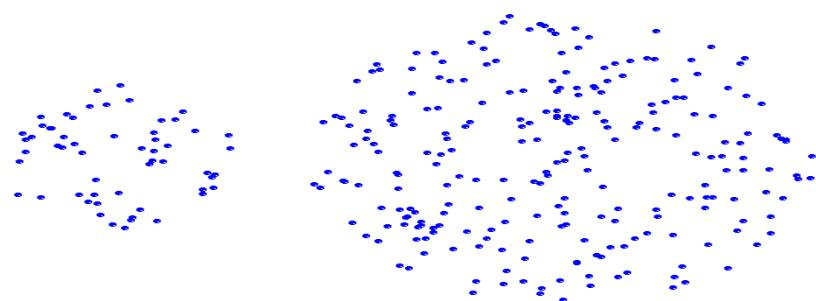


Nested Clusters

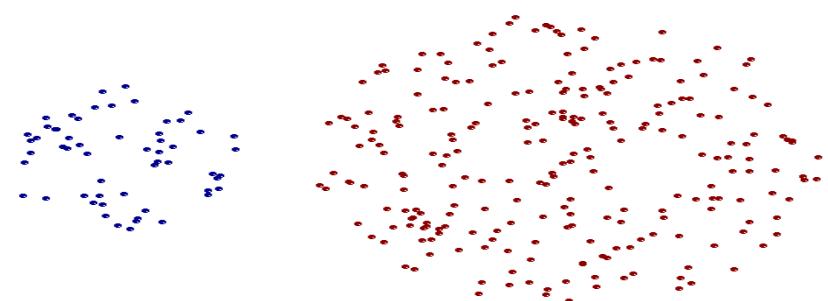


Dendrogram

Strength of MIN



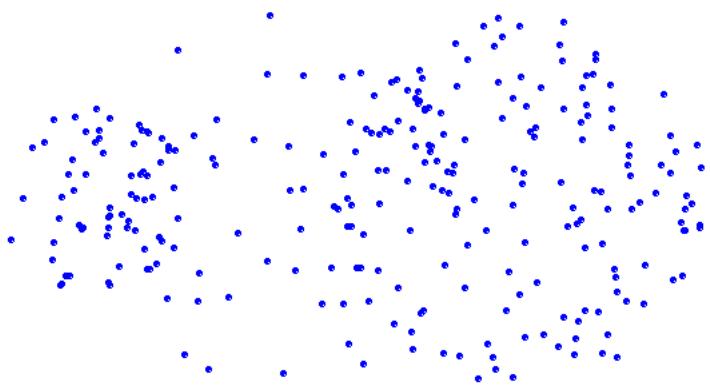
Original Points



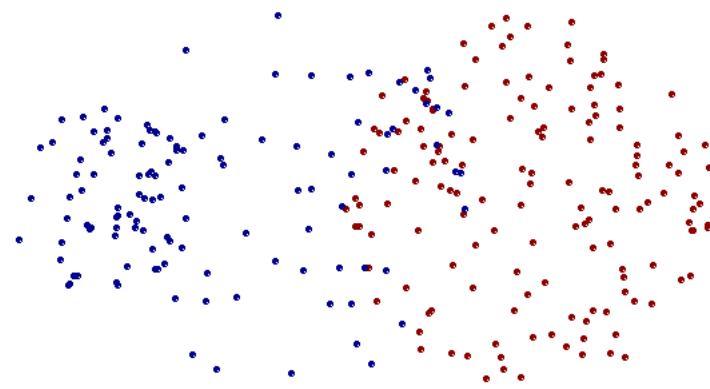
Two Clusters

- Can handle non-elliptical shapes

Limitations of MIN



Original Points



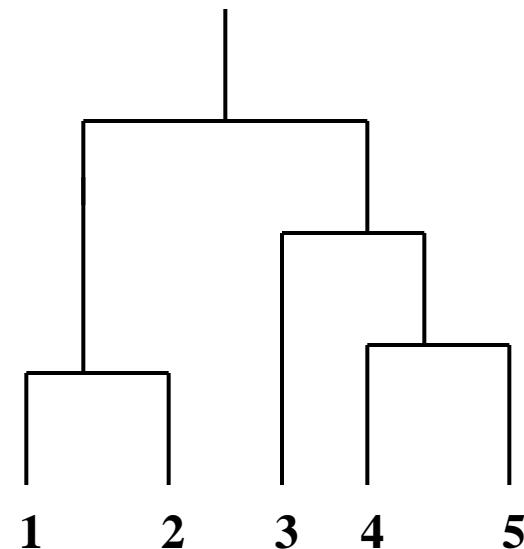
Two Clusters

- Sensitive to noise and outliers

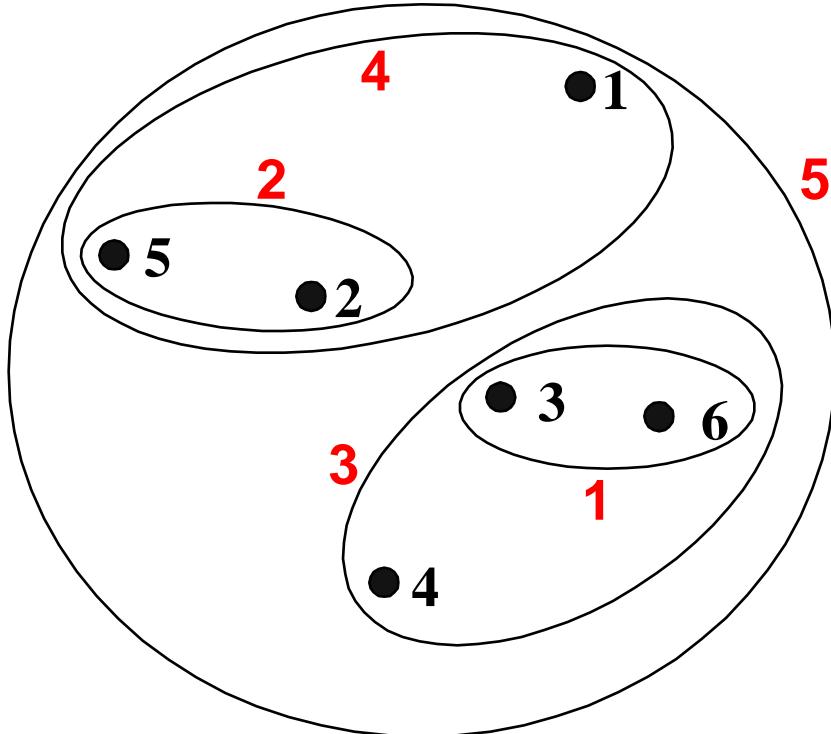
Cluster Similarity: MAX or Complete Linkage

- 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

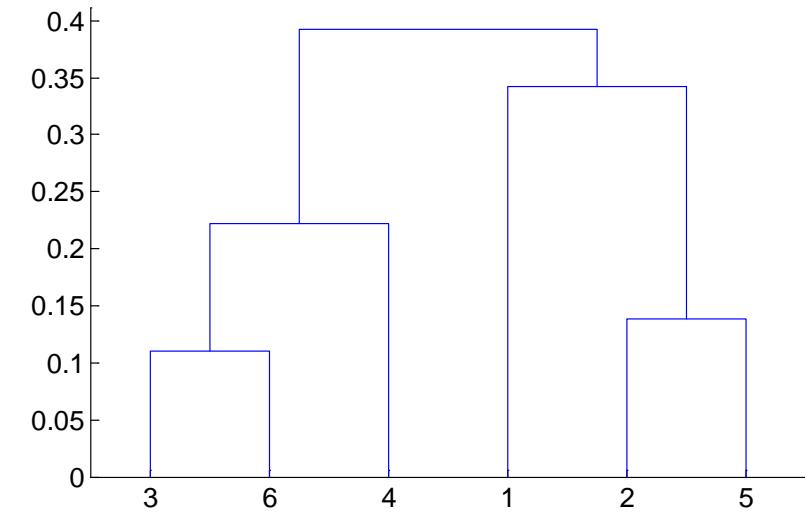
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: MAX

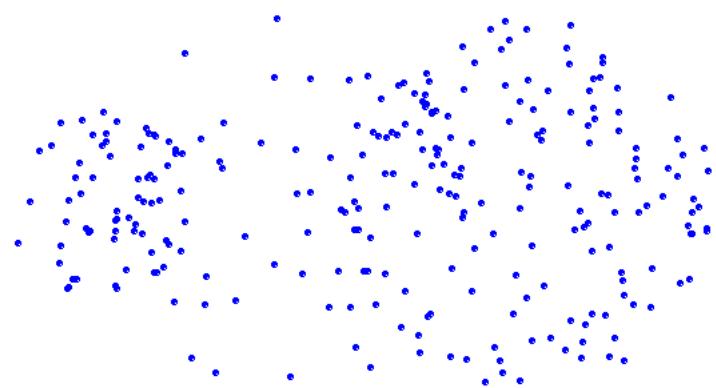


Nested Clusters

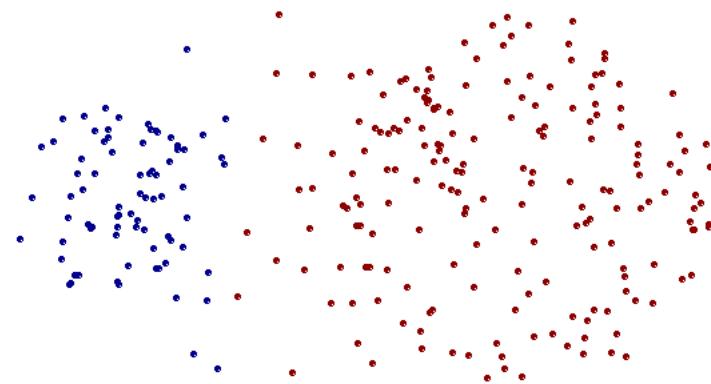


Dendrogram

Strength of MAX



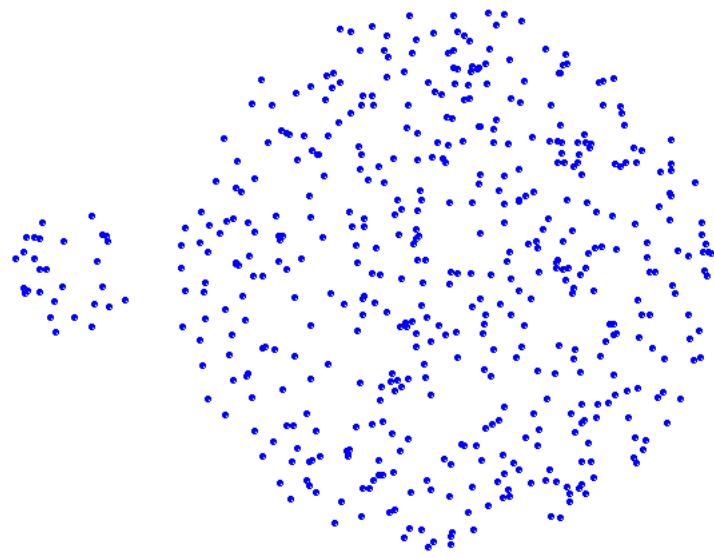
Original Points



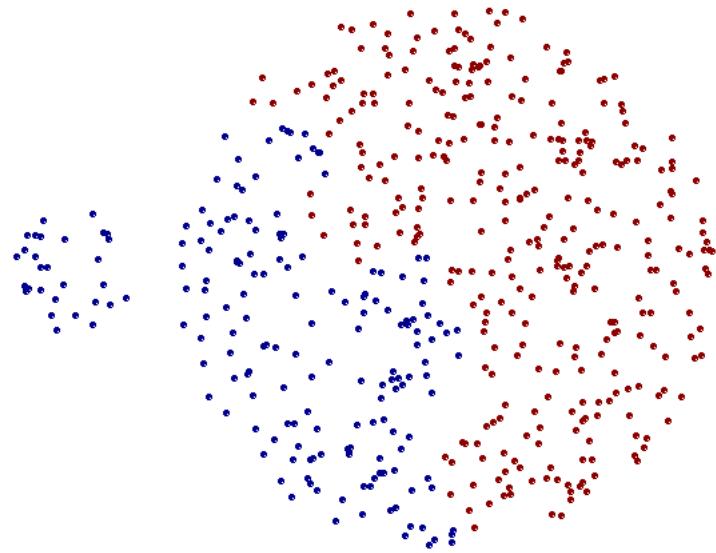
Two Clusters

- Less susceptible to noise and outliers

Limitations of MAX



Original Points



Two Clusters

- Tends to break large clusters
- Biased towards globular clusters

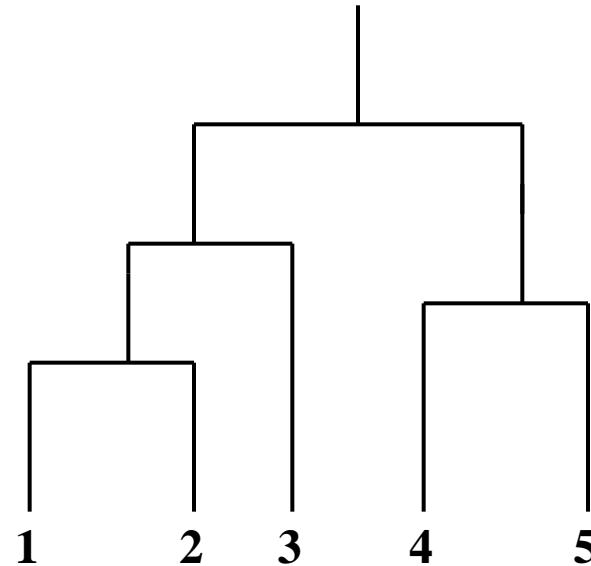
Cluster Similarity: Group Average

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

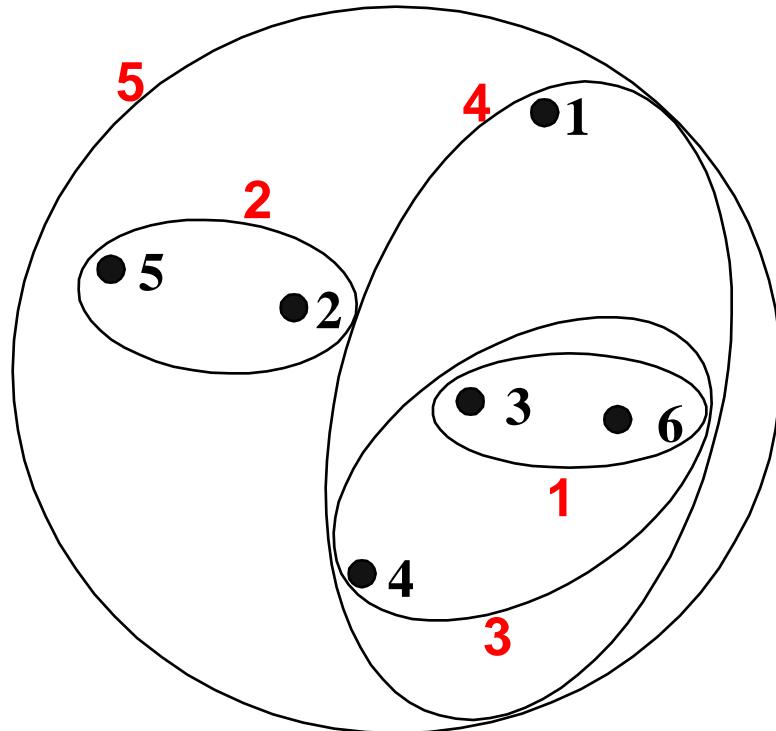
$$\text{proximity}(\text{Cluster}_i, \text{Cluster}_j) = \frac{\sum_{\substack{p_i \in \text{Cluster}_i \\ p_j \in \text{Cluster}_j}} \text{proximity}(p_i, p_j)}{|\text{Cluster}_i| * |\text{Cluster}_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

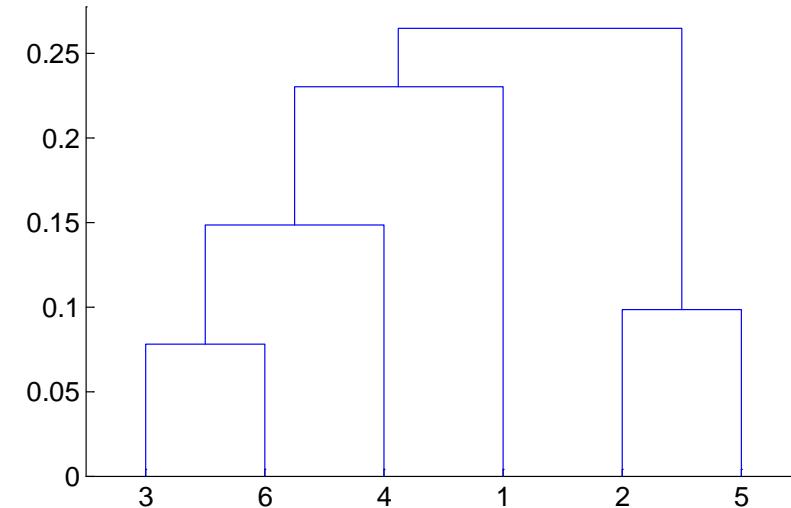
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



Hierarchical Clustering: Group Average



Nested Clusters



Dendrogram

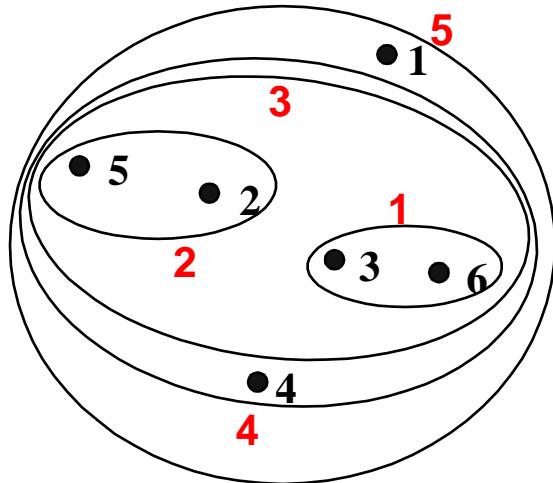
Hierarchical Clustering: Group Average

- Compromise between Single and Complete Link
- Strengths
 - Less susceptible to noise and outliers
- Limitations
 - Biased towards globular clusters

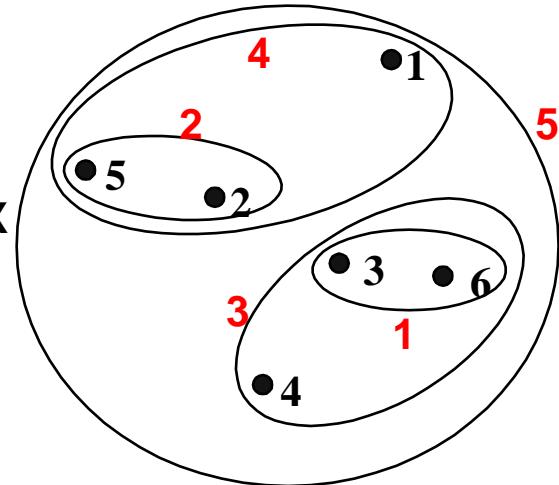
Cluster Similarity: Ward's Method

- Similarity of two clusters is based on the increase in squared error when two clusters are merged
 - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
 - Can be used to initialize K-means

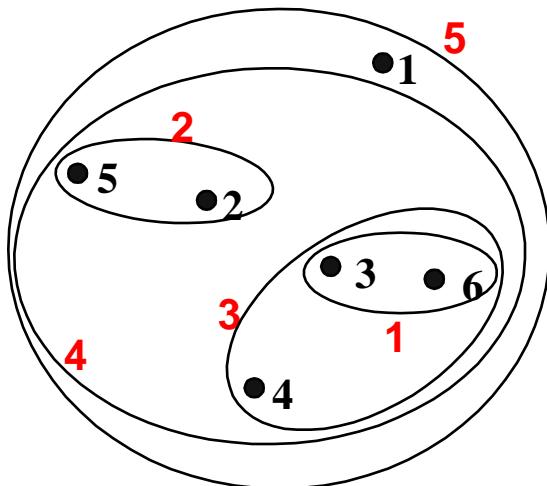
Hierarchical Clustering: Comparison



MIN

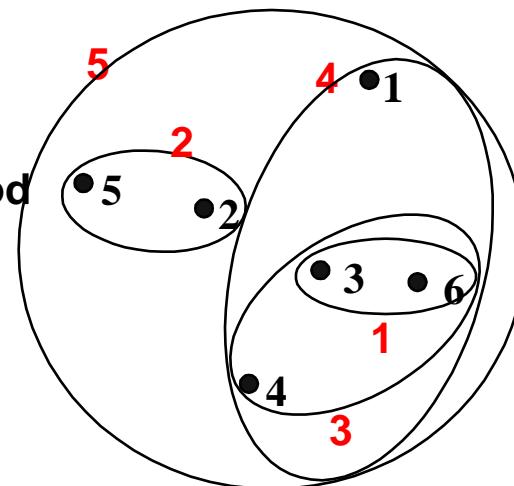


MAX



Group Average

Ward's Method



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

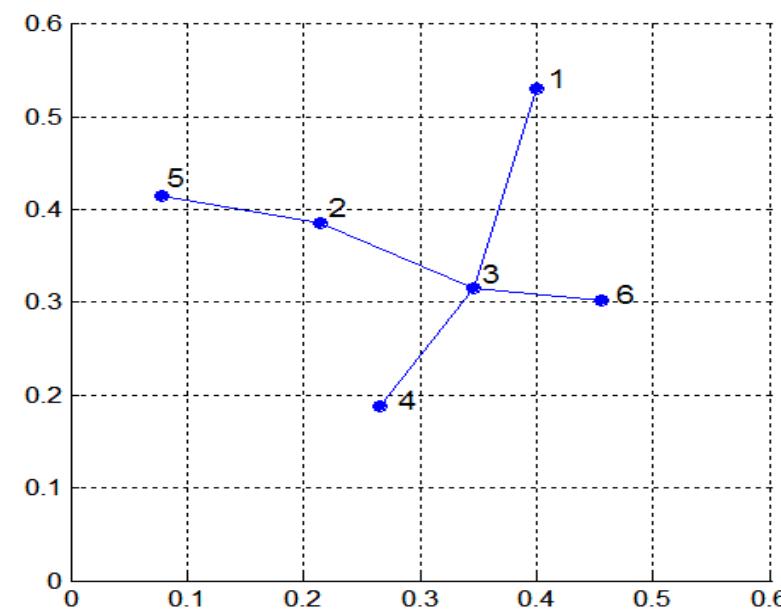
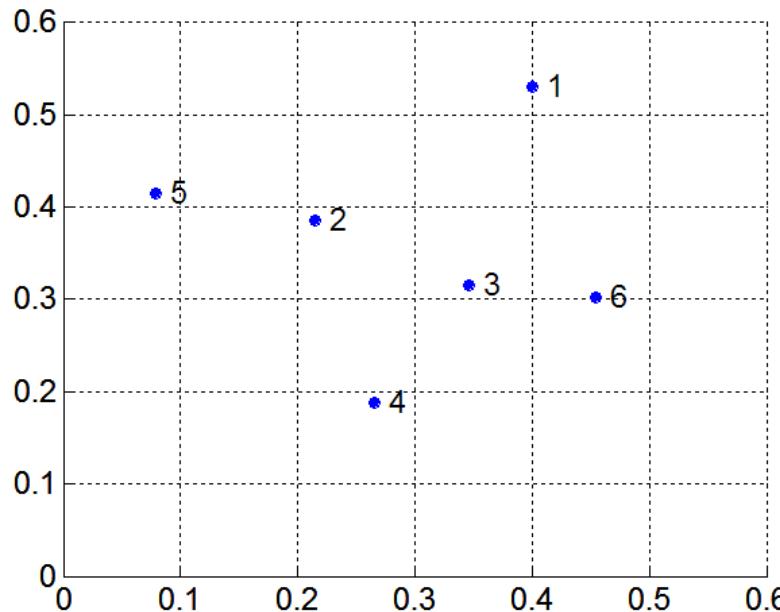
Hierarchical Clustering: Problems and Limitations

- 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

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



MST: Divisive Hierarchical Clustering

- Use MST for constructing hierarchy of clusters

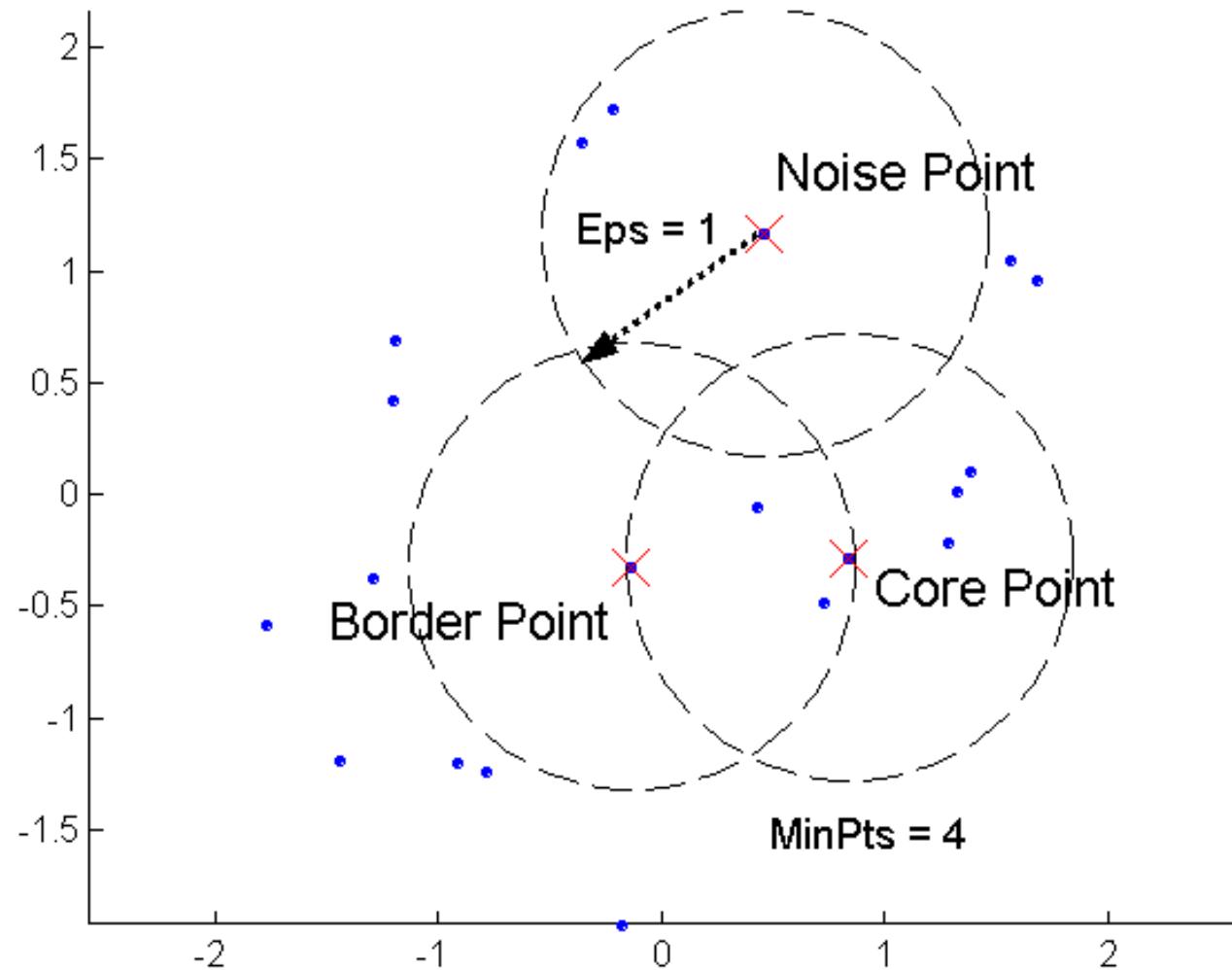
Algorithm 7.5 MST Divisive Hierarchical Clustering Algorithm

- 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
-

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



DBSCAN Algorithm

- Eliminate noise points
- Perform clustering on the remaining points

current-cluster-label $\leftarrow 1$

for all core points **do**

if the core point has no cluster label **then**

current-cluster-label $\leftarrow \text{current-cluster-label} + 1$

 Label the current core point with cluster label *current-cluster-label*

end if

for all points in the Eps -neighborhood, except i^{th} the point itself **do**

if the point does not have a cluster label **then**

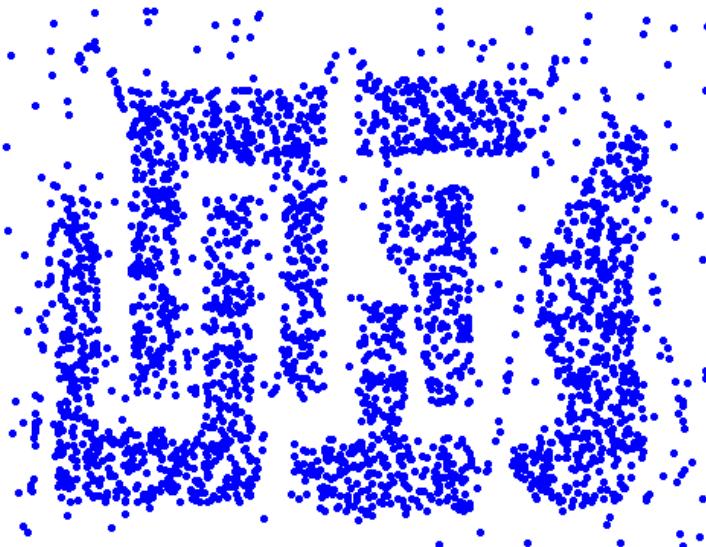
 Label the point with cluster label *current-cluster-label*

end if

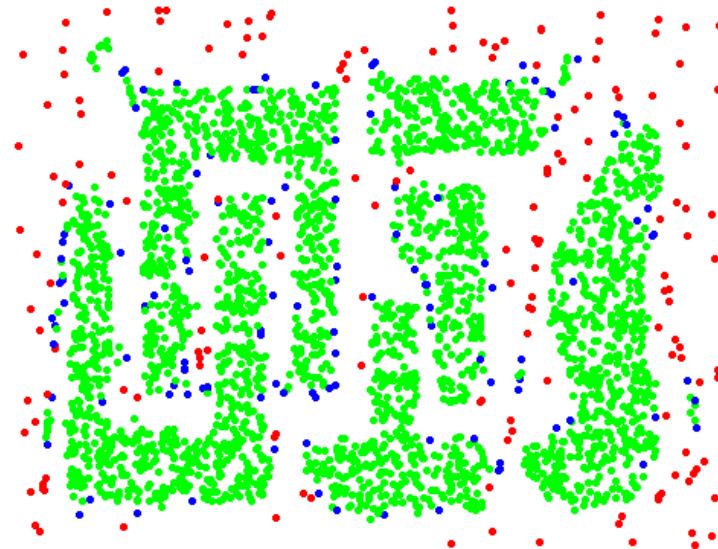
end for

end for

DBSCAN: Core, Border and Noise Points



Original Points



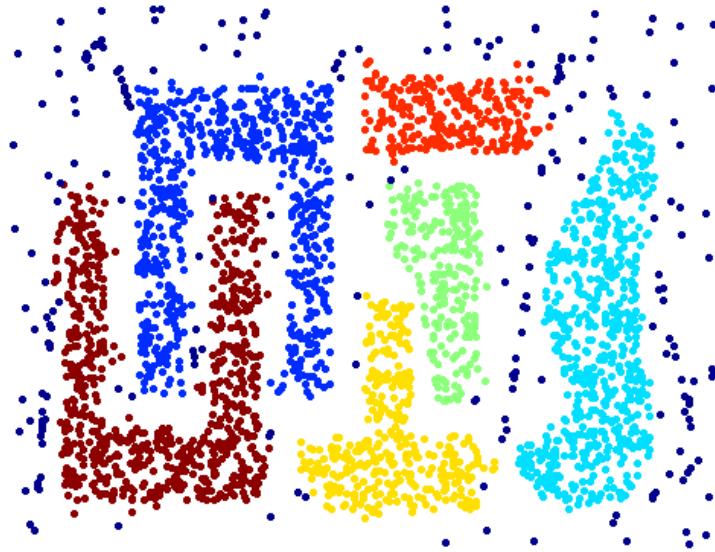
Point types: **core**,
border and **noise**

Eps = 10, MinPts = 4

When DBSCAN Works Well



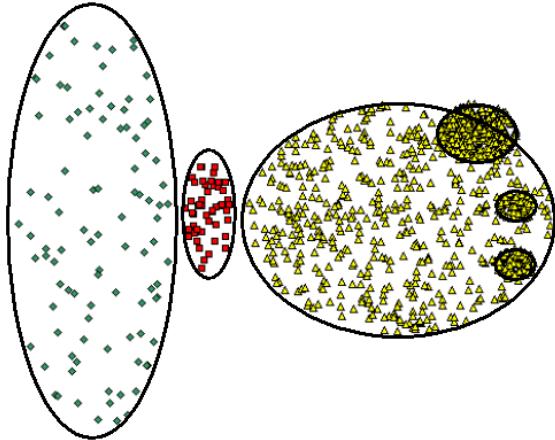
Original Points



Clusters

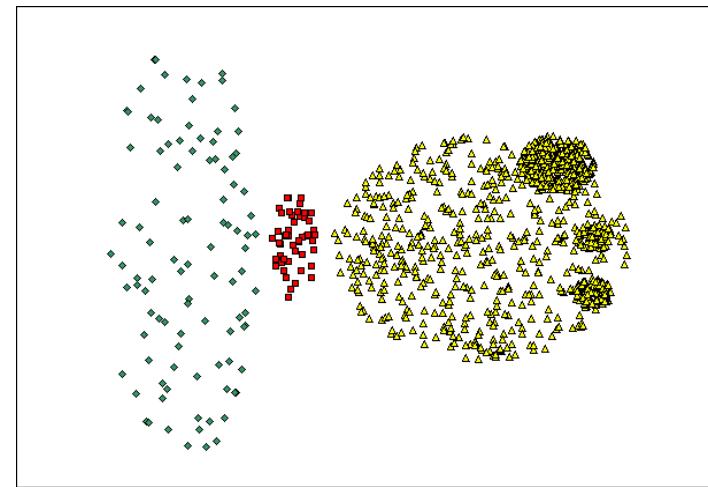
- Resistant to Noise
- Can handle clusters of different shapes and sizes

When DBSCAN Does NOT Work Well

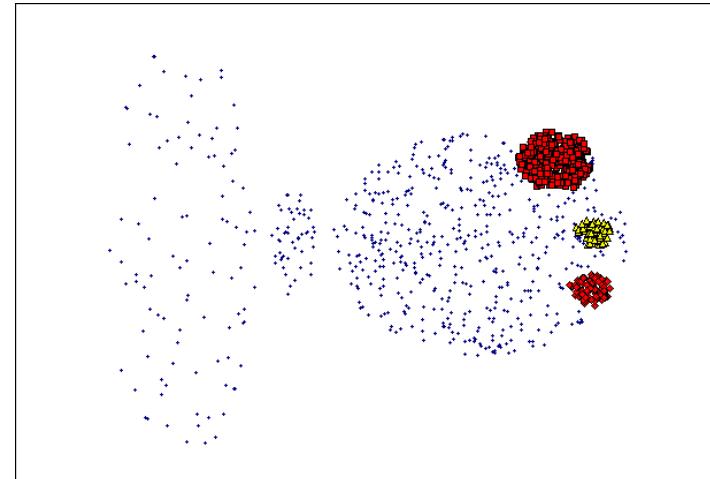


Original Points

- Varying densities
- High-dimensional data



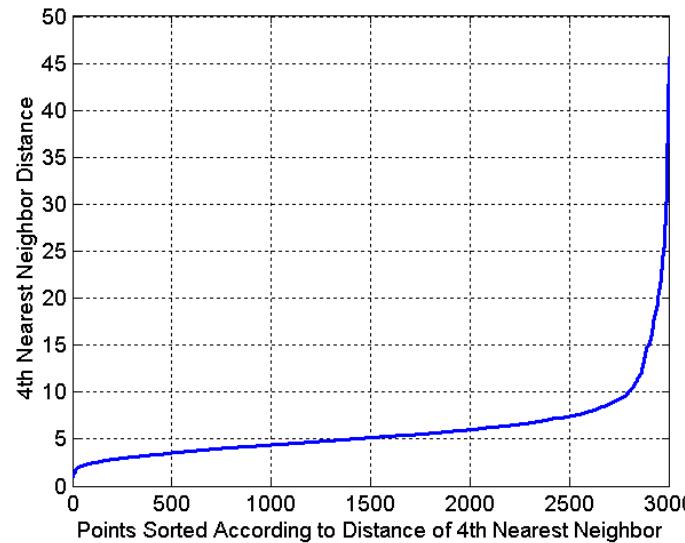
($\text{MinPts}=4$, $\text{Eps}=9.75$).



($\text{MinPts}=4$, $\text{Eps}=9.92$)

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

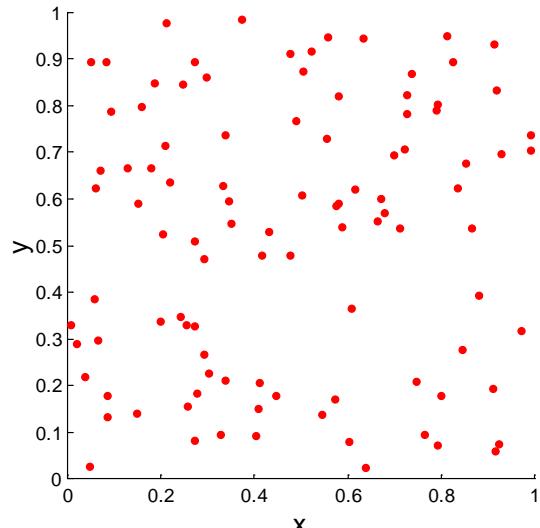


Cluster Validity

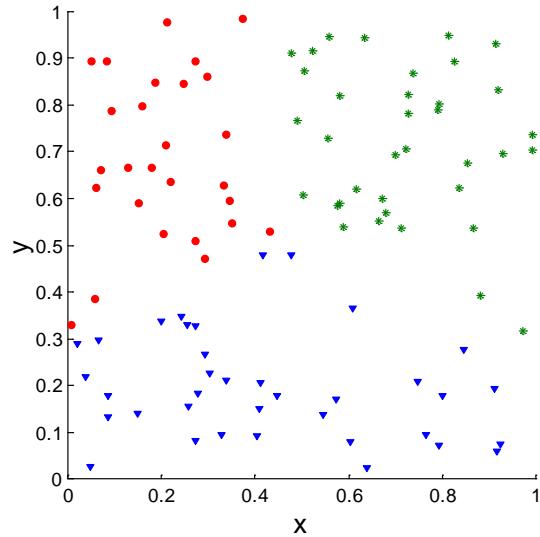
- For supervised classification we have a variety of measures to evaluate how good our model is
 - Accuracy, precision, recall
- For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?
- But “clusters are in the eye of the beholder”!
- Then why do we 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

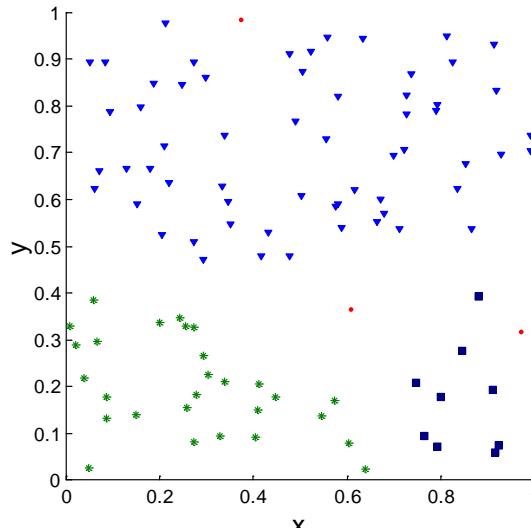
Random Points



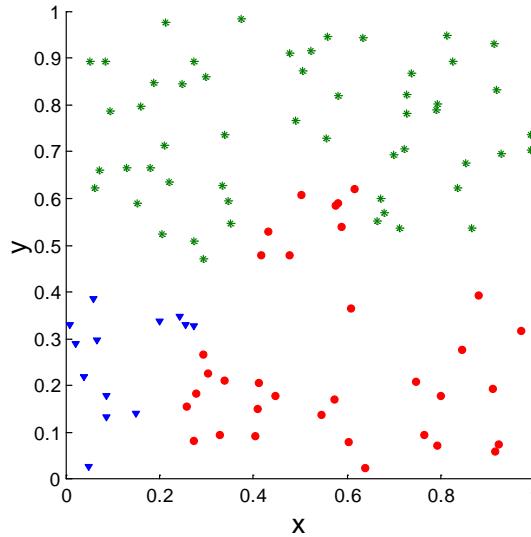
K-means



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.

Measures of Cluster Validity

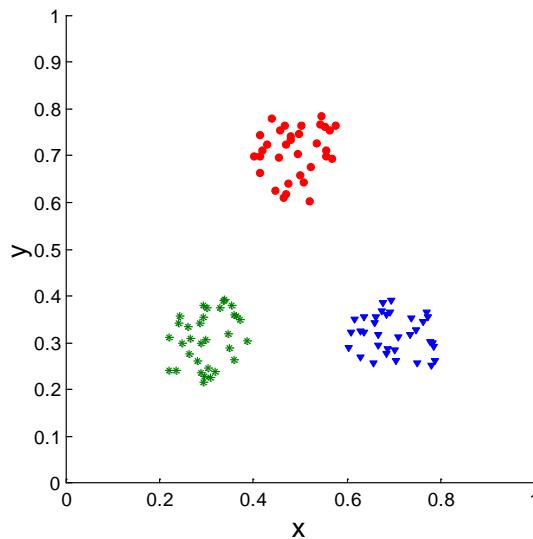
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
 - **External Index:** Used to measure the extent to which cluster labels match externally supplied class labels.
 - ◆ Entropy
 - **Internal Index:** Used to measure the goodness of a clustering structure *without* respect to external information.
 - ◆ Sum of Squared Error (SSE)
 - **Relative Index:** Used to compare two different clusterings or clusters.
 - ◆ Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as **criteria** instead of **indices**
 - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Measuring Cluster Validity Via Correlation

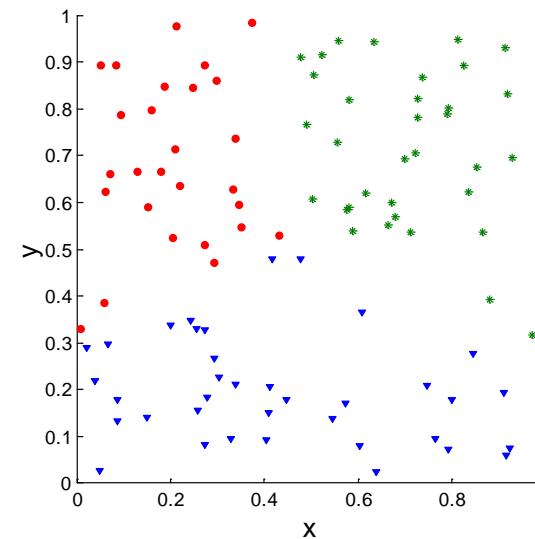
- Two matrices
 - Proximity Matrix
 - “Incidence” Matrix
 - ◆ One row and one column for each data point
 - ◆ An entry is 1 if the associated pair of points belong to the same cluster
 - ◆ An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the 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.

Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



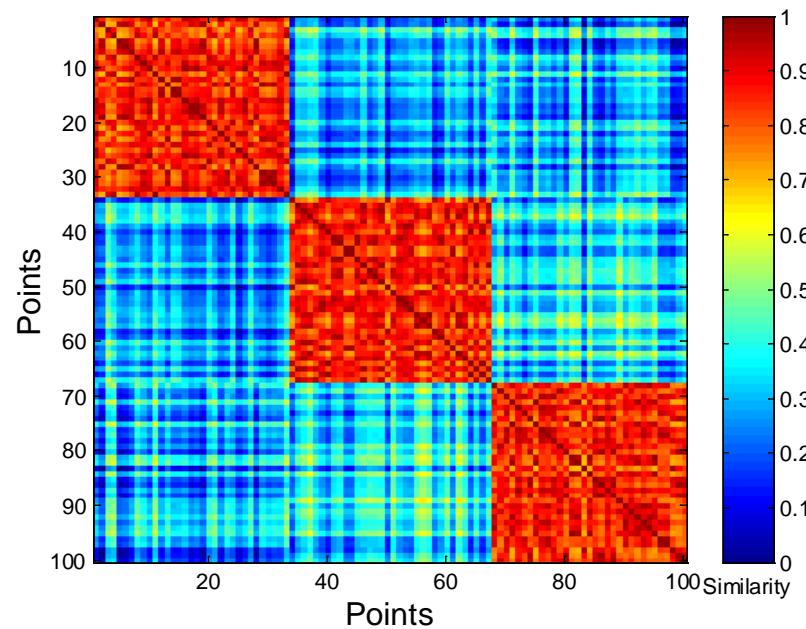
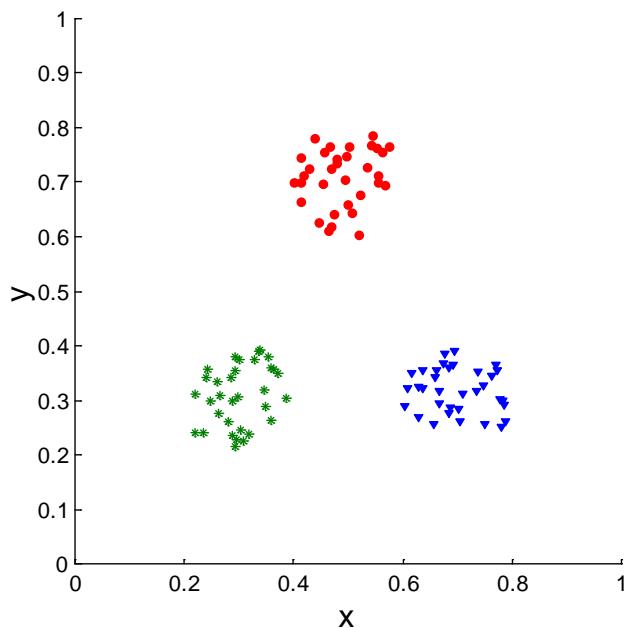
Corr = -0.9235



Corr = -0.5810

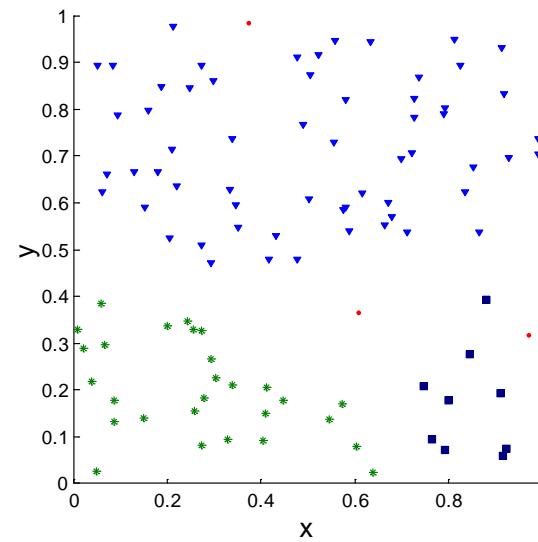
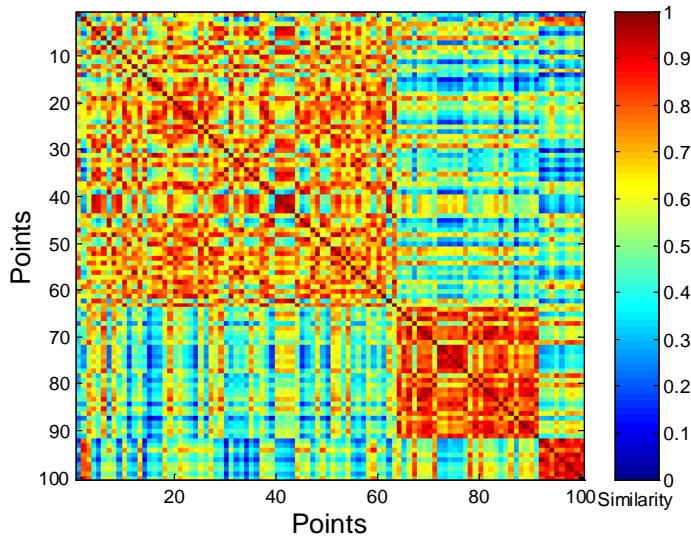
Using Similarity Matrix for Cluster Validation

- Order the similarity matrix with respect to cluster labels and inspect visually.



Using Similarity Matrix for Cluster Validation

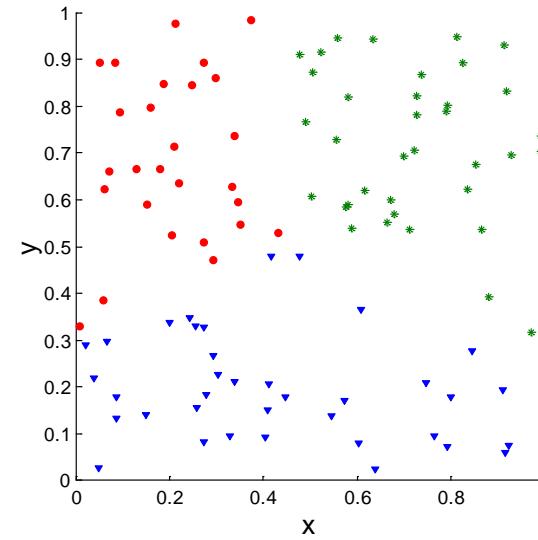
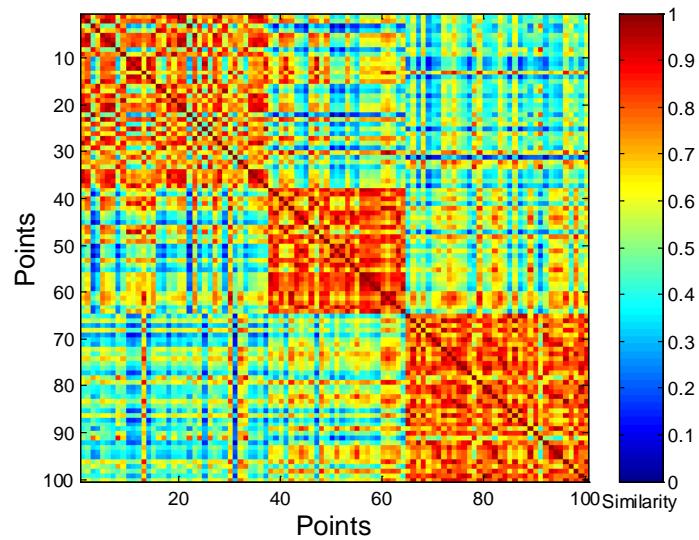
- Clusters in random data are not so crisp



DBSCAN

Using Similarity Matrix for Cluster Validation

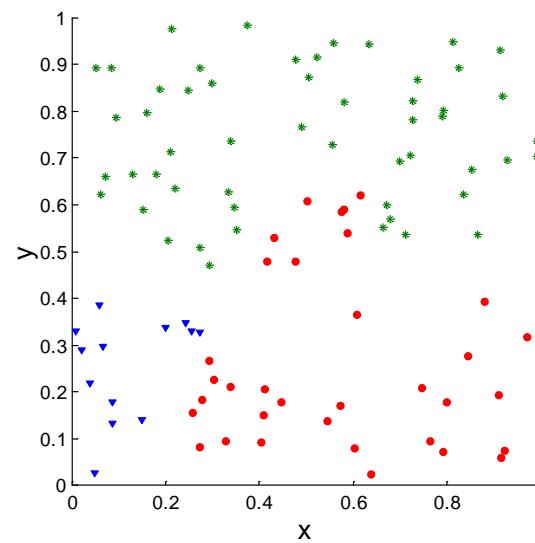
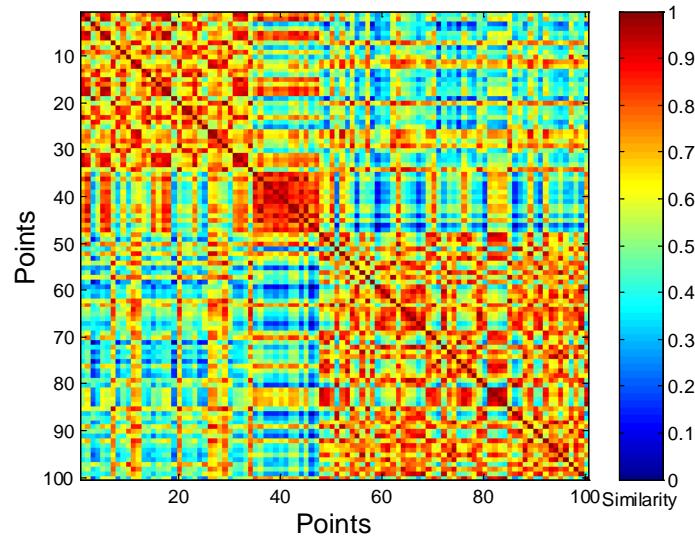
- Clusters in random data are not so crisp



K-means

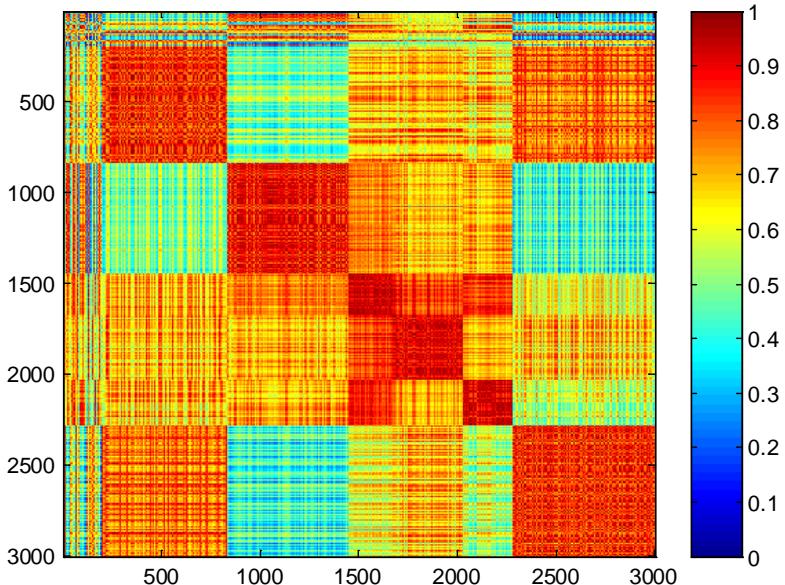
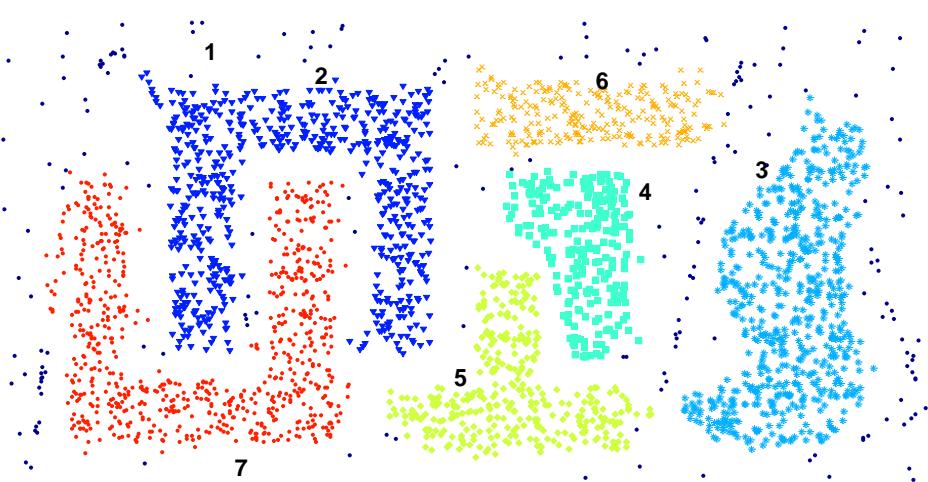
Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp



Complete Link

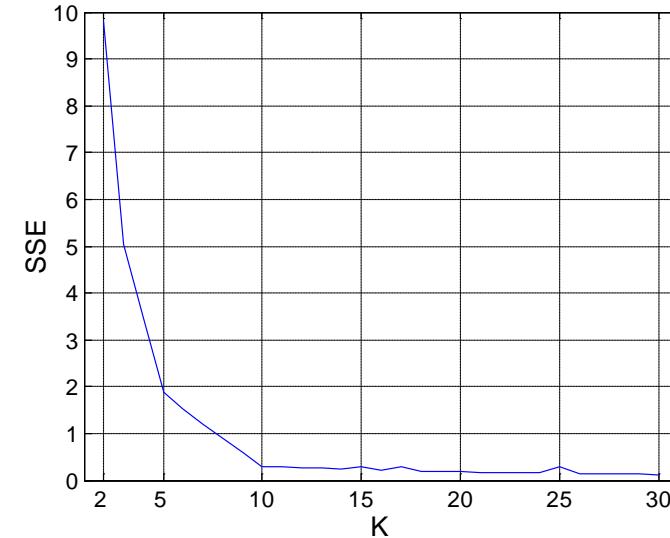
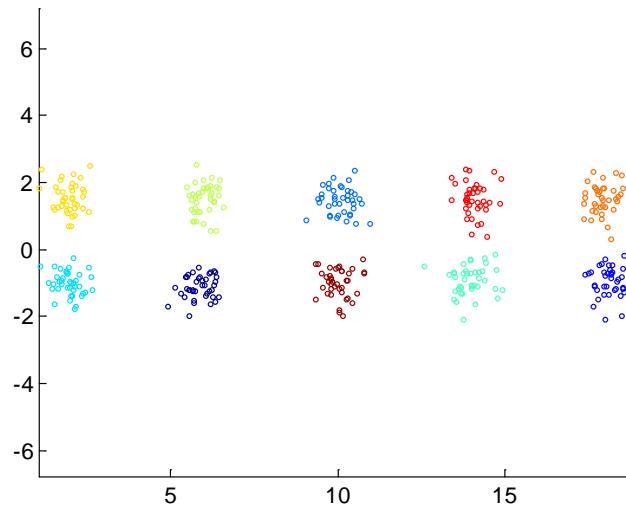
Using Similarity Matrix for Cluster Validation



DBSCAN

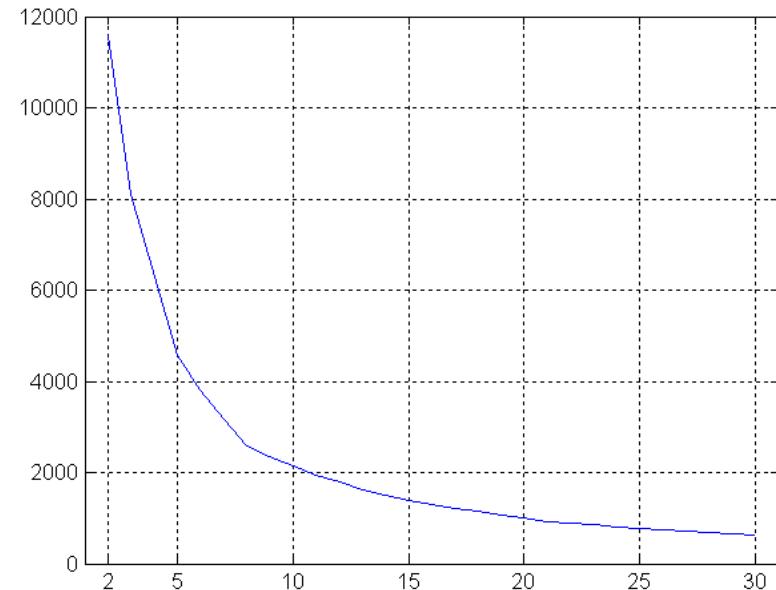
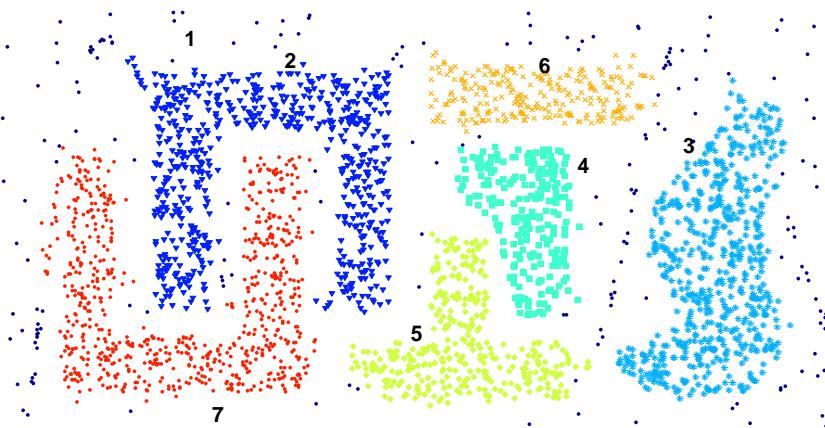
Internal Measures: SSE

- 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: SSE

- SSE curve for a more complicated data set



SSE of clusters found using K-means

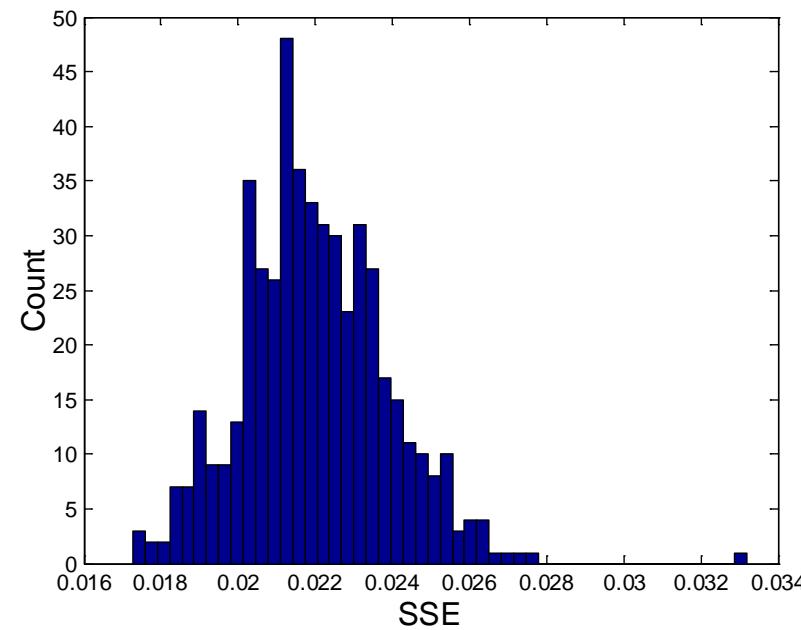
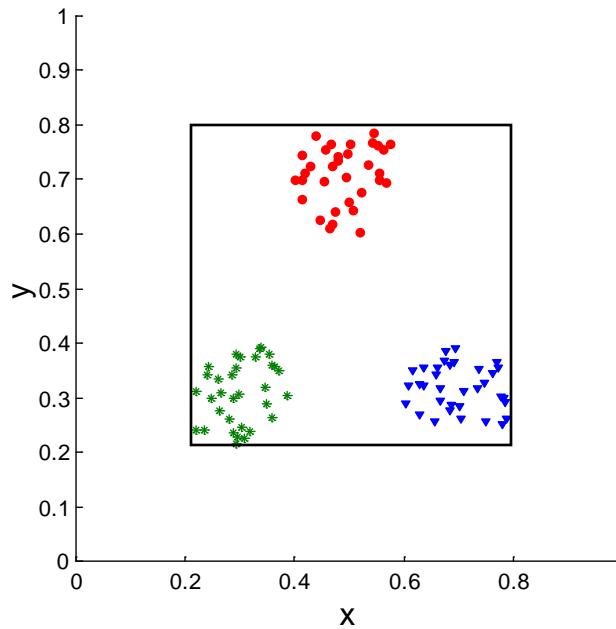
Framework for Cluster Validity

- Need a framework to interpret any measure.
 - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- Statistics provide a framework for cluster validity
 - The more “atypical” a clustering result is, the more likely it represents valid structure in the data
 - Can compare the values of an index that result from random data or clusterings to those of a clustering result.
 - ◆ If the value of the index is unlikely, then the cluster results are valid
 - These approaches are more complicated and harder to understand.
- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
 - However, there is the question of whether the difference between two index values is significant

Statistical Framework for SSE

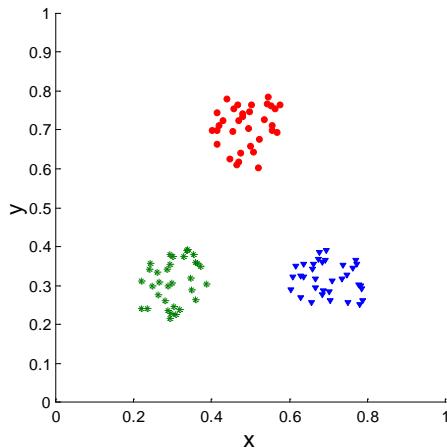
● Example

- Compare SSE of 0.005 against three clusters in random data
- Histogram shows SSE of three clusters in 500 sets of random data points of size 100 distributed over the range 0.2 – 0.8 for x and y values

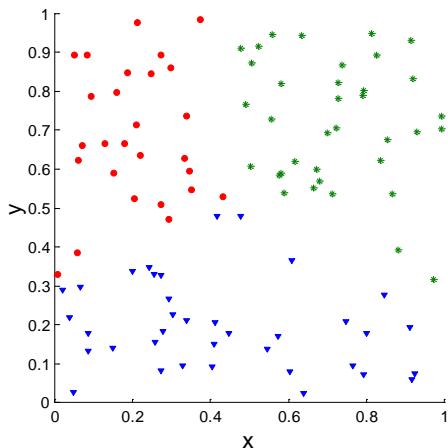


Statistical Framework for Correlation

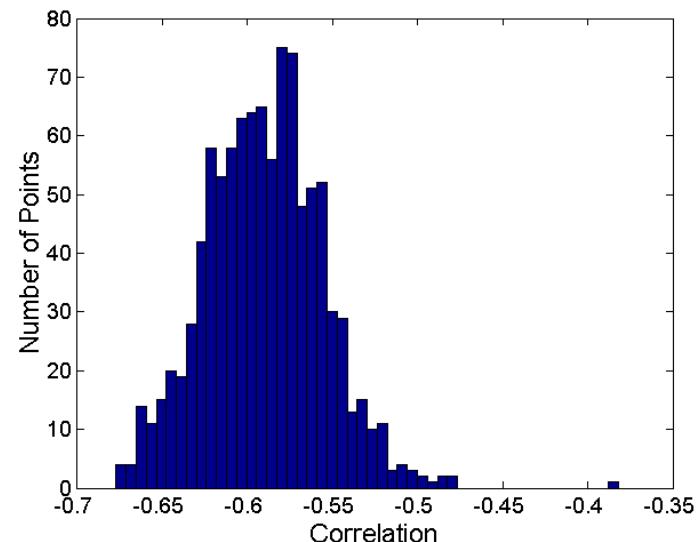
- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



Corr = -0.9235



Corr = -0.5810



Internal Measures: Cohesion and Separation

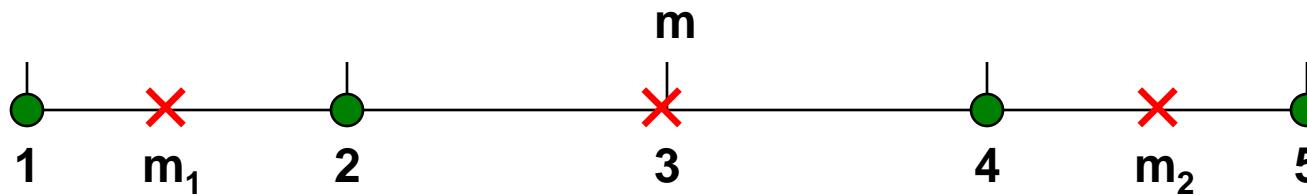
- **Cluster Cohesion:** Measures how closely related are objects in a cluster
 - Example: SSE
- **Cluster Separation:** Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
 - Cohesion is measured by the within cluster sum of squares (SSE)
$$WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$
 - Separation is measured by the between cluster sum of squares

$$BSS = \sum |C_i| (m - m_i)^2$$

– Where $|C_i|$ is the size of cluster i

Internal Measures: Cohesion and Separation

- Example: SSE
 - $BSS + WSS = \text{constant}$



K=1 cluster: $WSS = (1 - 3)^2 + (2 - 3)^2 + (4 - 3)^2 + (5 - 3)^2 = 10$

$$BSS = 4 \times (3 - 3)^2 = 0$$

$$Total = 10 + 0 = 10$$

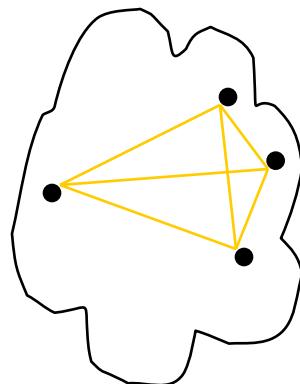
K=2 clusters: $WSS = (1 - 1.5)^2 + (2 - 1.5)^2 + (4 - 4.5)^2 + (5 - 4.5)^2 = 1$

$$BSS = 2 \times (3 - 1.5)^2 + 2 \times (4.5 - 3)^2 = 9$$

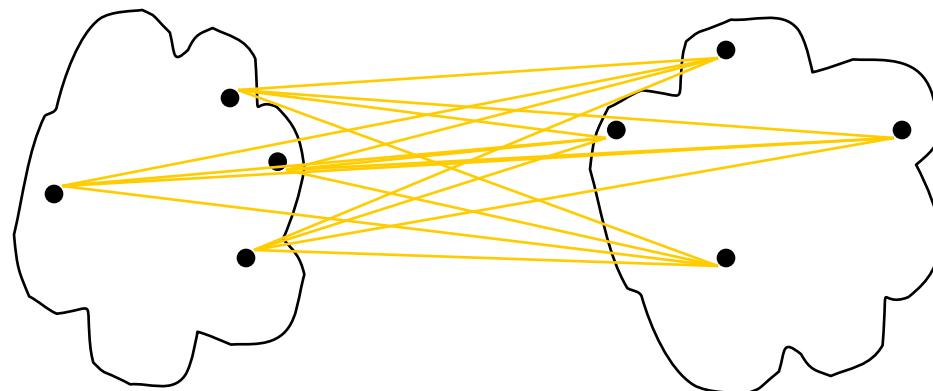
$$Total = 1 + 9 = 10$$

Internal Measures: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation.
 - Cluster cohesion is the sum of the weight of all links within a cluster.
 - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion



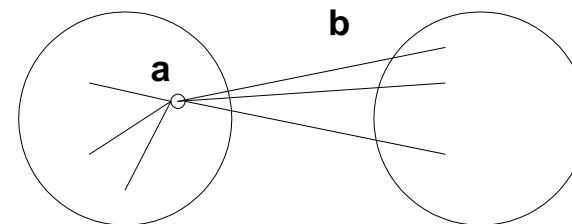
separation

Internal Measures: Silhouette Coefficient

- Silhouette Coefficient combine ideas of both cohesion and separation, but for individual points, as well as clusters and clusterings
- For an individual point, i
 - Calculate $a = \text{average distance of } i \text{ to the points in its cluster}$
 - Calculate $b = \min(\text{average distance of } i \text{ to points in another cluster})$
 - The silhouette coefficient for a point is then given by

$$s = 1 - a/b \quad \text{if } a < b, \quad (\text{or } s = b/a - 1 \quad \text{if } a \geq b, \text{ not the usual case})$$

- Typically between 0 and 1.
- The closer to 1 the better.



- Can calculate the Average Silhouette width for a cluster or a clustering

External Measures of Cluster Validity: Entropy and Purity

Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203

entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij} , the ‘probability’ that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j . Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^L p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^K \frac{m_i}{m} e_j$, where m_j is the size of cluster j , K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j , is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^K \frac{m_i}{m} purity_j$.

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

Data Mining

Cluster Analysis: Advanced Concepts and Algorithms

Lecture Notes for Chapter 9

Introduction to Data Mining

by

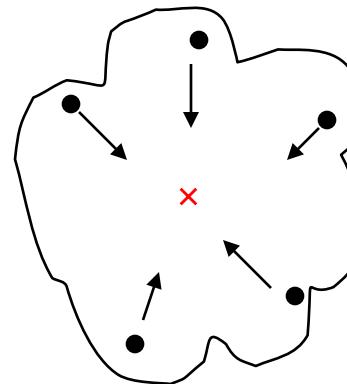
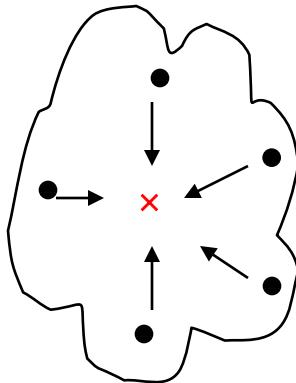
Tan, Steinbach, Kumar

Hierarchical Clustering: Revisited

- Creates nested clusters
- Agglomerative clustering algorithms vary in terms of how the proximity of two clusters are computed
 - ◆ MIN (single link): susceptible to noise/outliers
 - ◆ MAX/GROUP AVERAGE:
may not work well with non-globular clusters
- CURE algorithm tries to handle both problems
- Often starts with a proximity matrix
 - A type of graph-based algorithm

CURE: Another Hierarchical Approach

- Uses a number of points to represent a cluster

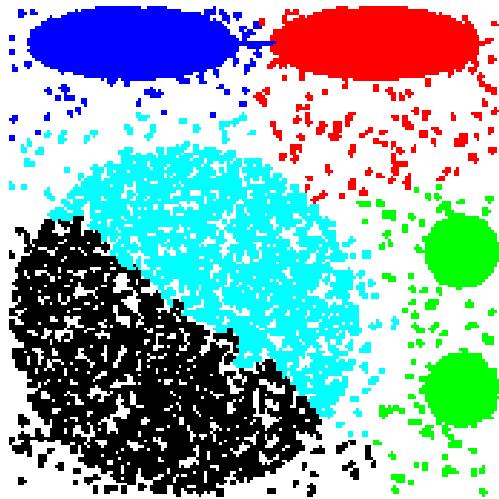


- Representative points are found by selecting a constant number of points from a cluster and then “shrinking” them toward the center of the cluster
- Cluster similarity is the similarity of the closest pair of representative points from different clusters

CURE

- Shrinking representative points toward the center helps avoid problems with noise and outliers
- CURE is better able to handle clusters of arbitrary shapes and sizes

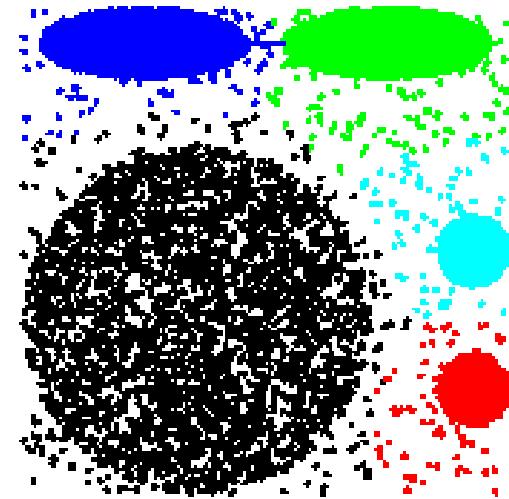
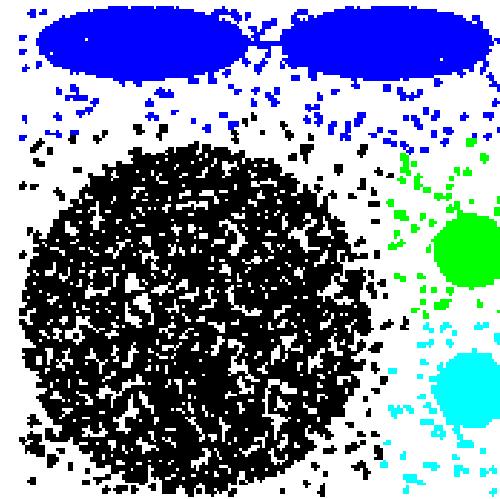
Experimental Results: CURE



a) BIRCH

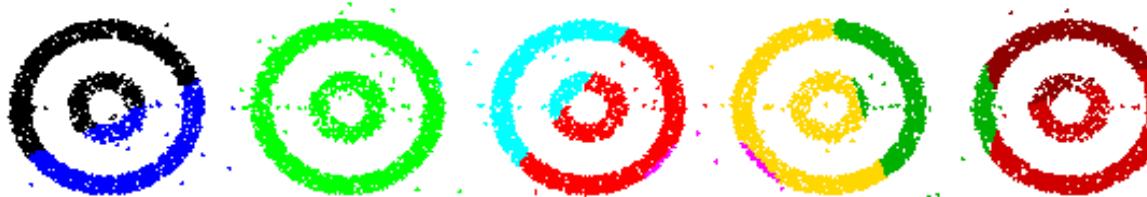
b) MST METHOD

c) CURE



Picture from *CURE*, Guha, Rastogi, Shim.

Experimental Results: CURE



a) BIRCH

(centroid)



b) MST METHOD

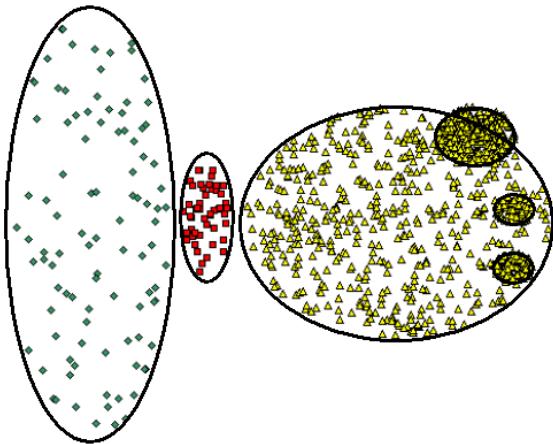
(single link)



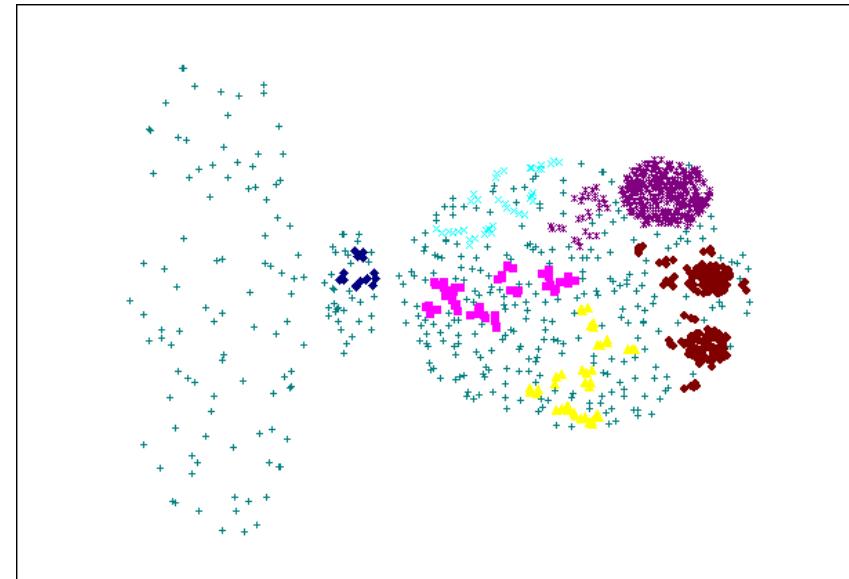
c) CURE

Picture from *CURE*, Guha, Rastogi, Shim.

CURE Cannot Handle Differing Densities



Original Points



CURE

Graph-Based Clustering

- Graph-Based clustering uses the proximity graph
 - Start with the proximity matrix
 - Consider each point as a node in a graph
 - Each edge between two nodes has a weight which is the proximity between the two points
 - Initially the proximity graph is fully connected
 - MIN (single-link) and MAX (complete-link) can be viewed as starting with this graph
- In the simplest case, clusters are connected components in the graph.

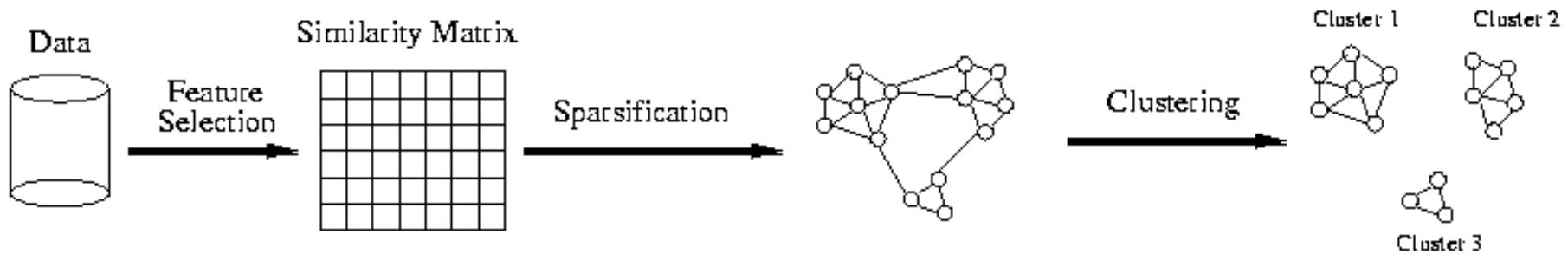
Graph-Based Clustering: Sparsification

- The amount of data that needs to be processed is drastically reduced
 - Sparsification can eliminate more than 99% of the entries in a proximity matrix
 - The amount of time required to cluster the data is drastically reduced
 - The size of the problems that can be handled is increased

Graph-Based Clustering: Sparsification ...

- Clustering may work better
 - Sparsification techniques keep the connections to the most similar (nearest) neighbors of a point while breaking the connections to less similar points.
 - The nearest neighbors of a point tend to belong to the same class as the point itself.
 - This reduces the impact of noise and outliers and sharpens the distinction between clusters.
- Sparsification facilitates the use of graph partitioning algorithms (or algorithms based on graph partitioning algorithms).
 - Chameleon and Hypergraph-based Clustering

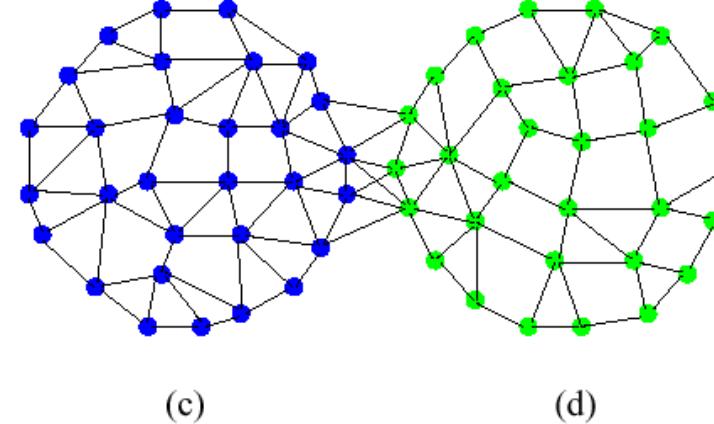
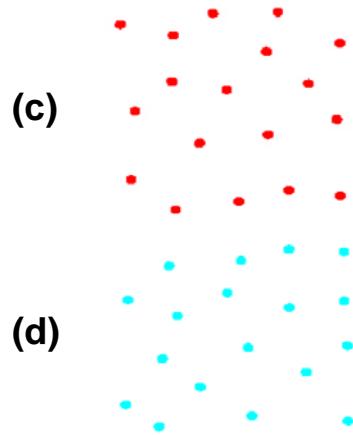
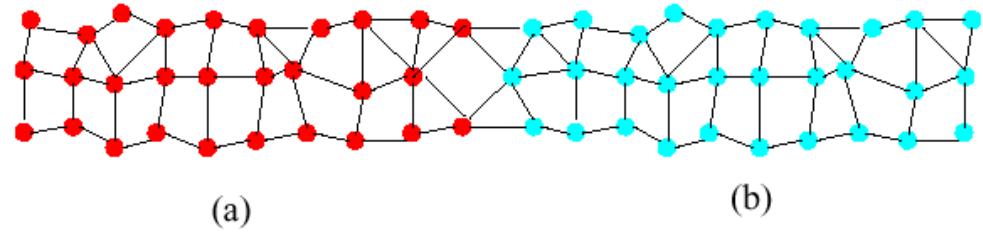
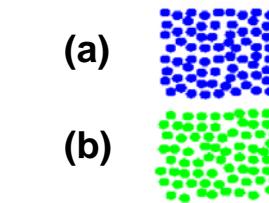
Sparsification in the Clustering Process



Limitations of Current Merging Schemes

- Existing merging schemes in hierarchical clustering algorithms are static in nature
 - MIN or CURE:
 - ◆ merge two clusters based on their *closeness* (or minimum distance)
 - GROUP-AVERAGE:
 - ◆ merge two clusters based on their average *connectivity*

Limitations of Current Merging Schemes

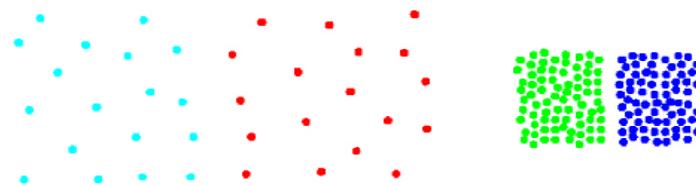


**Closeness schemes
will merge (a) and (b)**

**Average connectivity schemes
will merge (c) and (d)**

Chameleon: Clustering Using Dynamic Modeling

- Adapt to the characteristics of the data set to find the natural clusters
- Use a dynamic model to measure the similarity between clusters
 - Main property is the relative closeness and relative inter-connectivity of the cluster
 - Two clusters are combined if the resulting cluster shares certain *properties* with the constituent clusters
 - The merging scheme preserves *self-similarity*

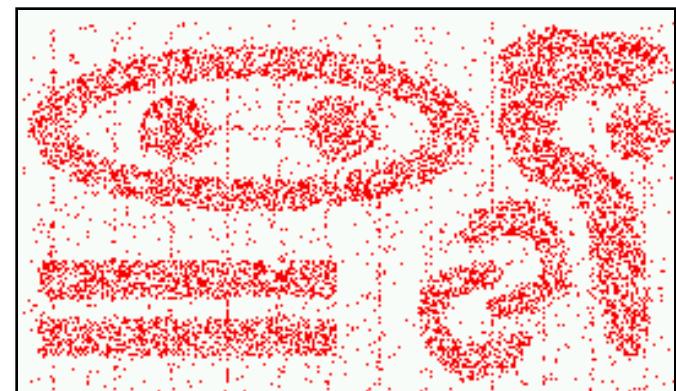
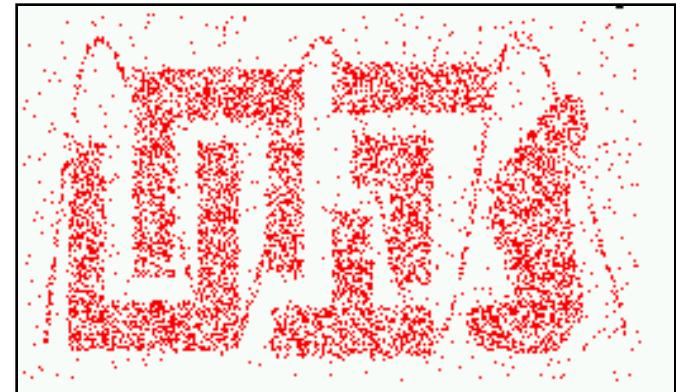


- One of the areas of application is *spatial data*

Characteristics of Spatial Data Sets

- Clusters are defined as densely populated regions of the space
- Clusters have arbitrary shapes, orientation, and non-uniform sizes
- Difference in densities across clusters and variation in density within clusters
- Existence of special artifacts (*streaks*) and noise

The clustering algorithm must address the above characteristics and also require minimal supervision.



Chameleon: Steps

- **Preprocessing Step:**

Represent the Data by a Graph

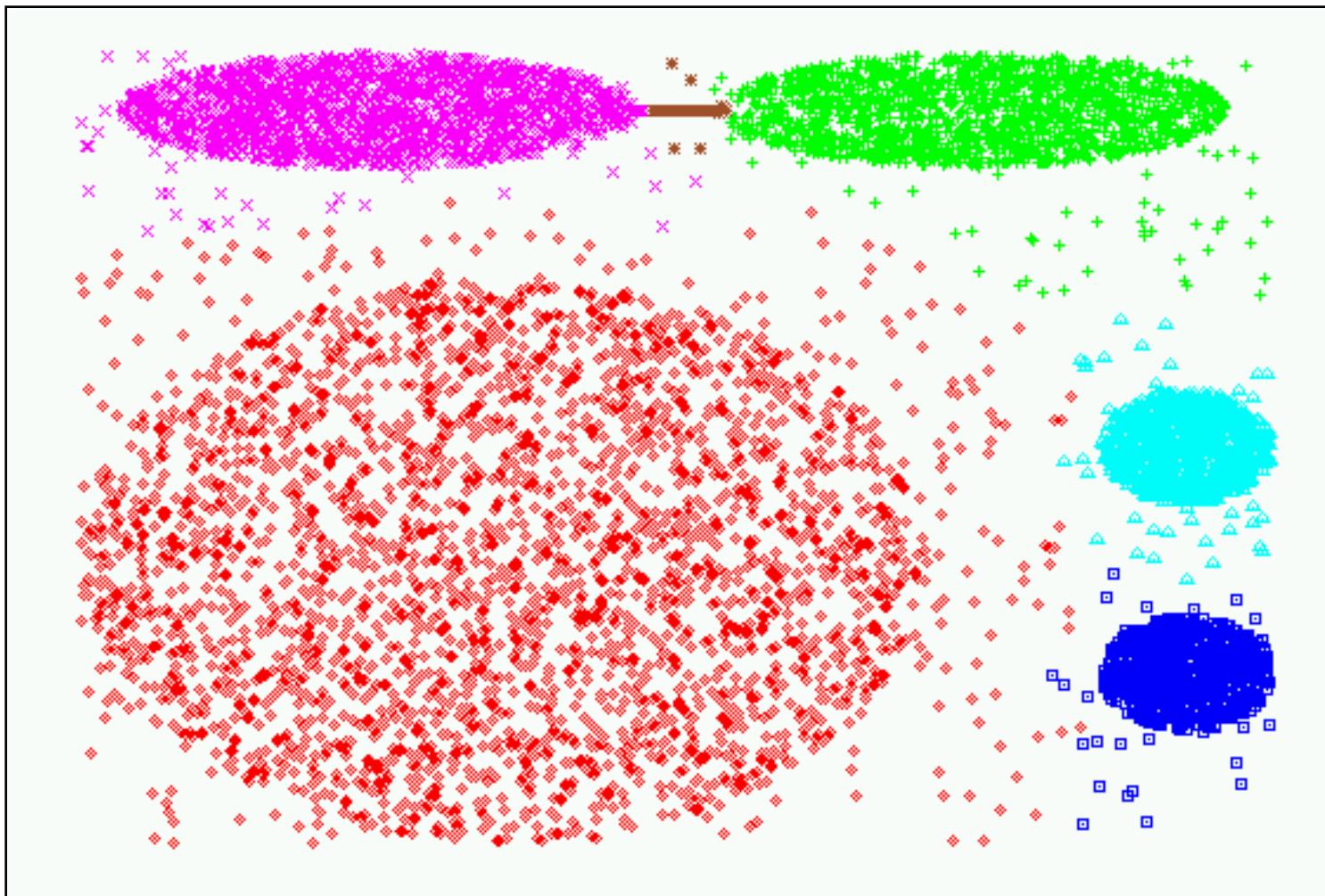
- Given a set of points, construct the k-nearest-neighbor (k-NN) graph to capture the relationship between a point and its k nearest neighbors
- Concept of neighborhood is captured dynamically (even if region is sparse)

- **Phase 1:** Use a multilevel graph partitioning algorithm on the graph to find a large number of clusters of well-connected vertices
 - Each cluster should contain mostly points from one “true” cluster, i.e., is a sub-cluster of a “real” cluster

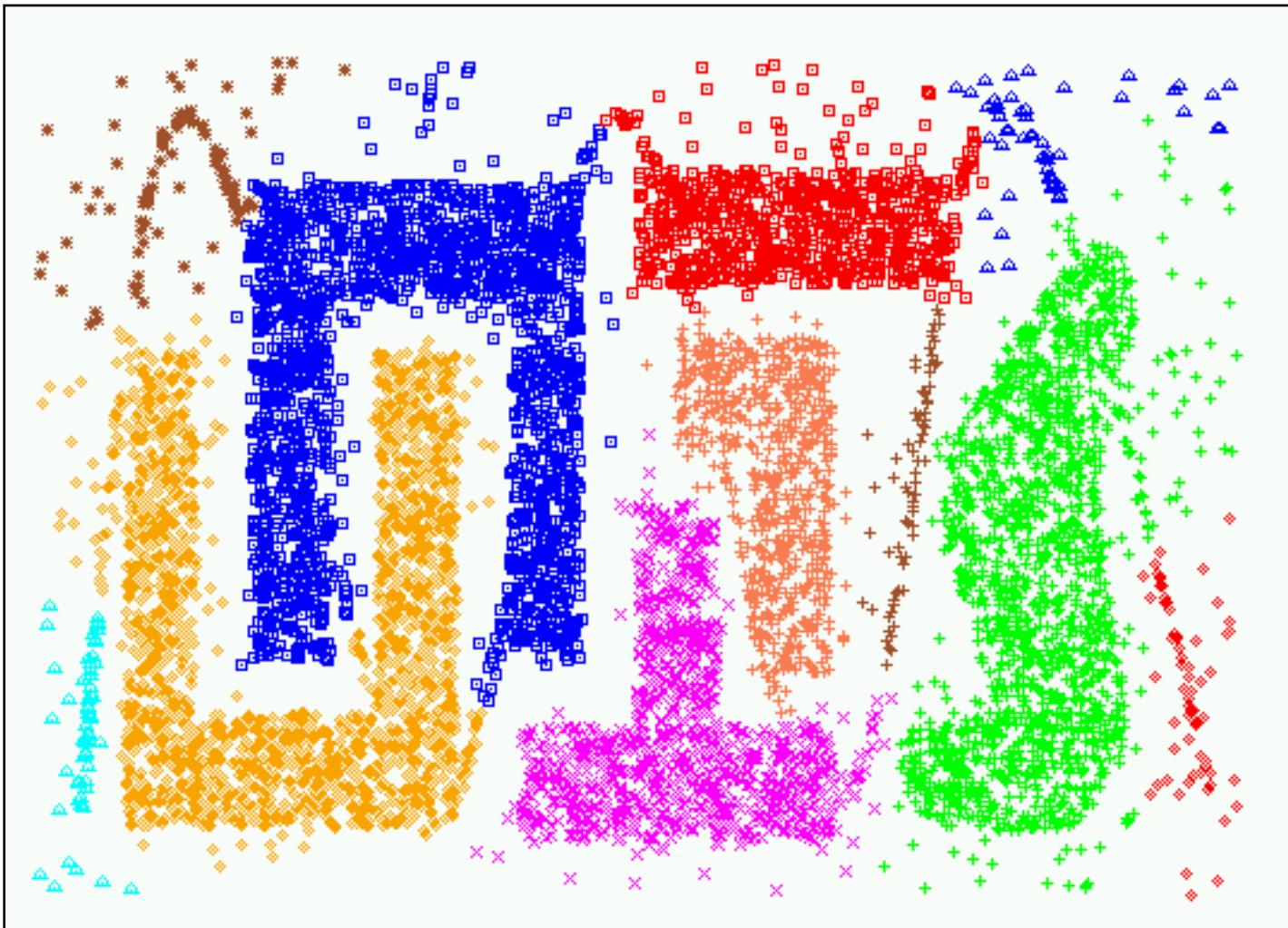
Chameleon: Steps ...

- Phase 2: Use Hierarchical Agglomerative Clustering to merge sub-clusters
 - Two clusters are combined if the *resulting cluster shares certain properties with the constituent clusters*
 - Two key properties used to model cluster similarity:
 - ◆ **Relative Interconnectivity**: Absolute interconnectivity of two clusters normalized by the internal connectivity of the clusters
 - ◆ **Relative Closeness**: Absolute closeness of two clusters normalized by the internal closeness of the clusters

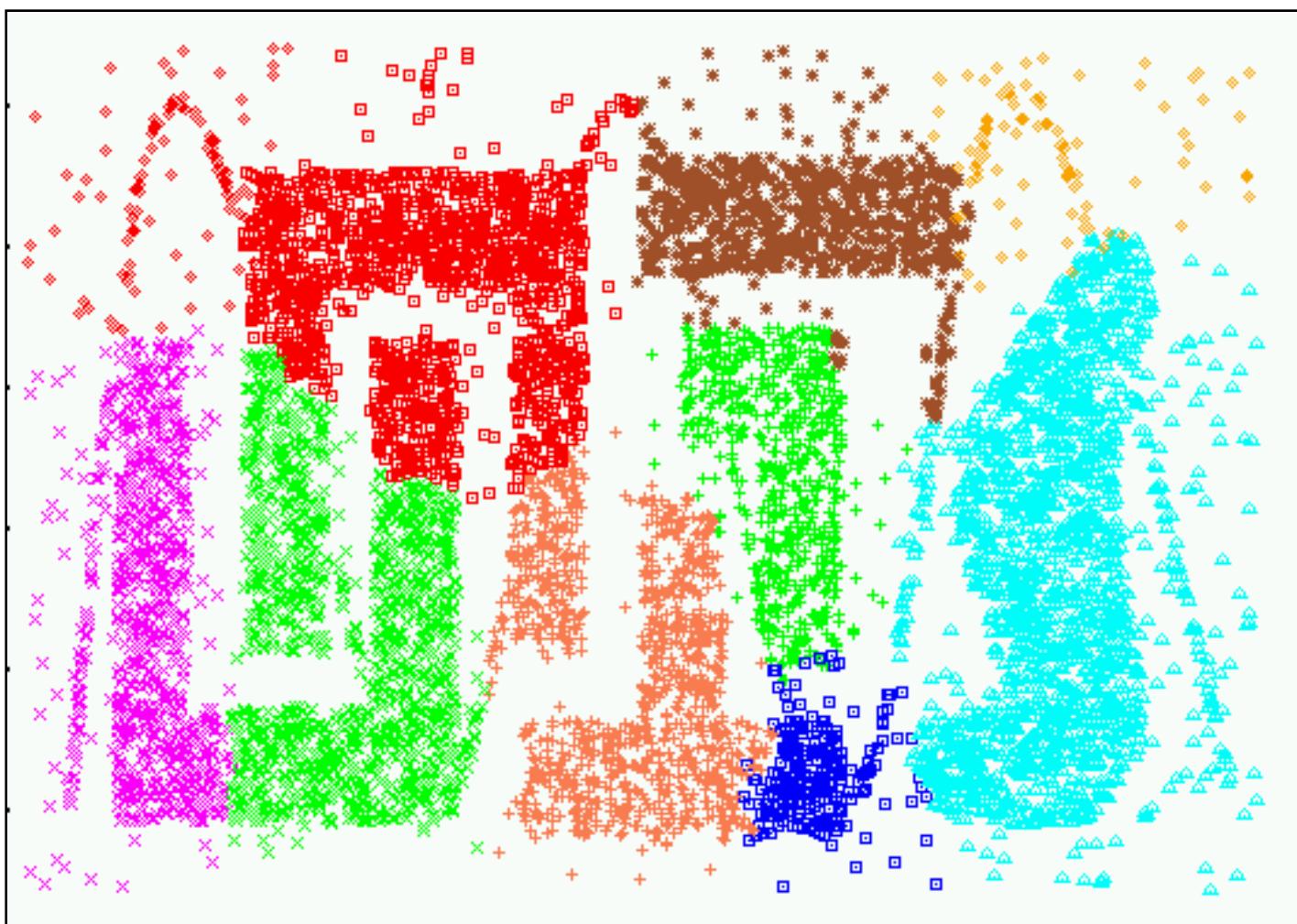
Experimental Results: CHAMELEON



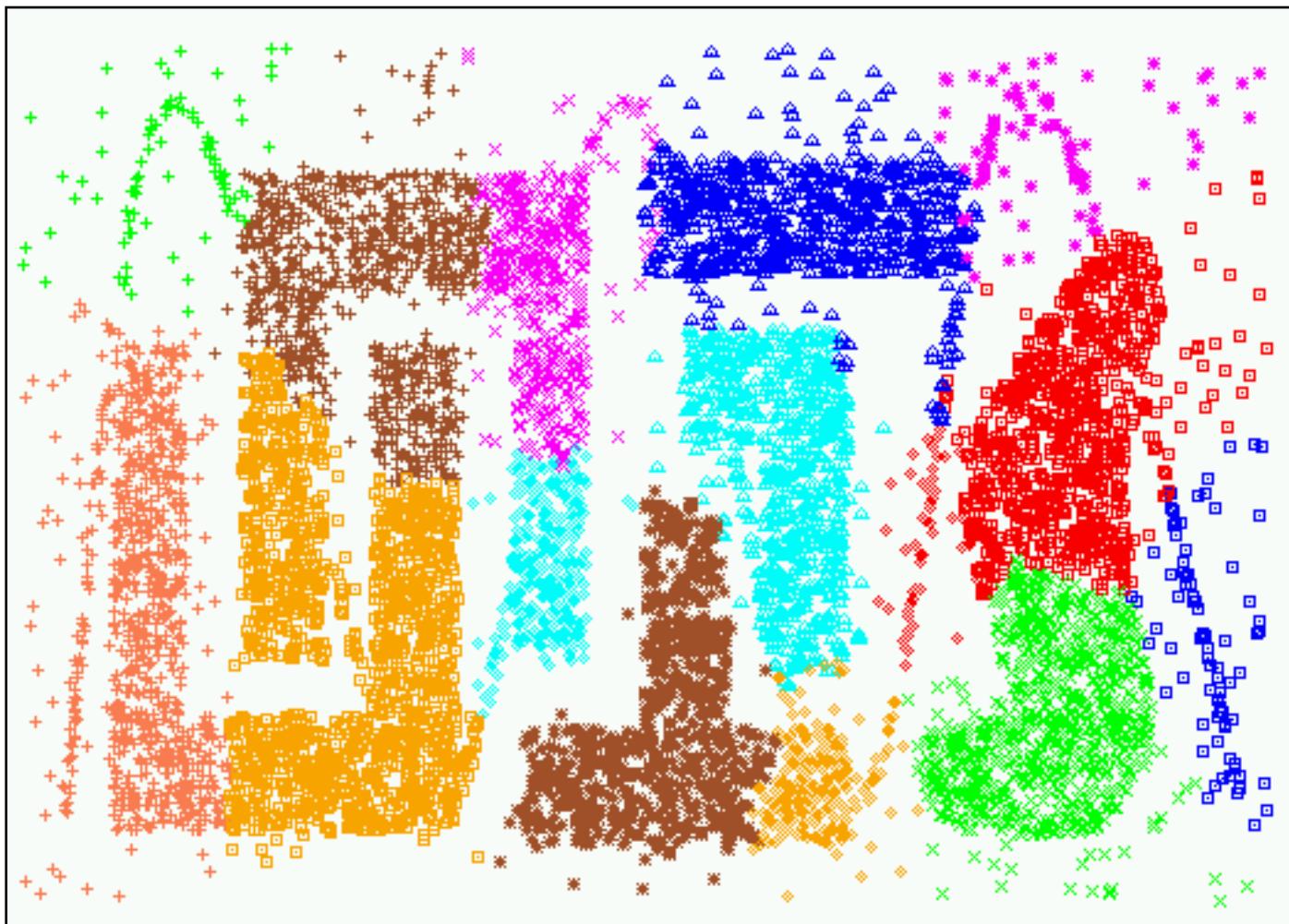
Experimental Results: CHAMELEON



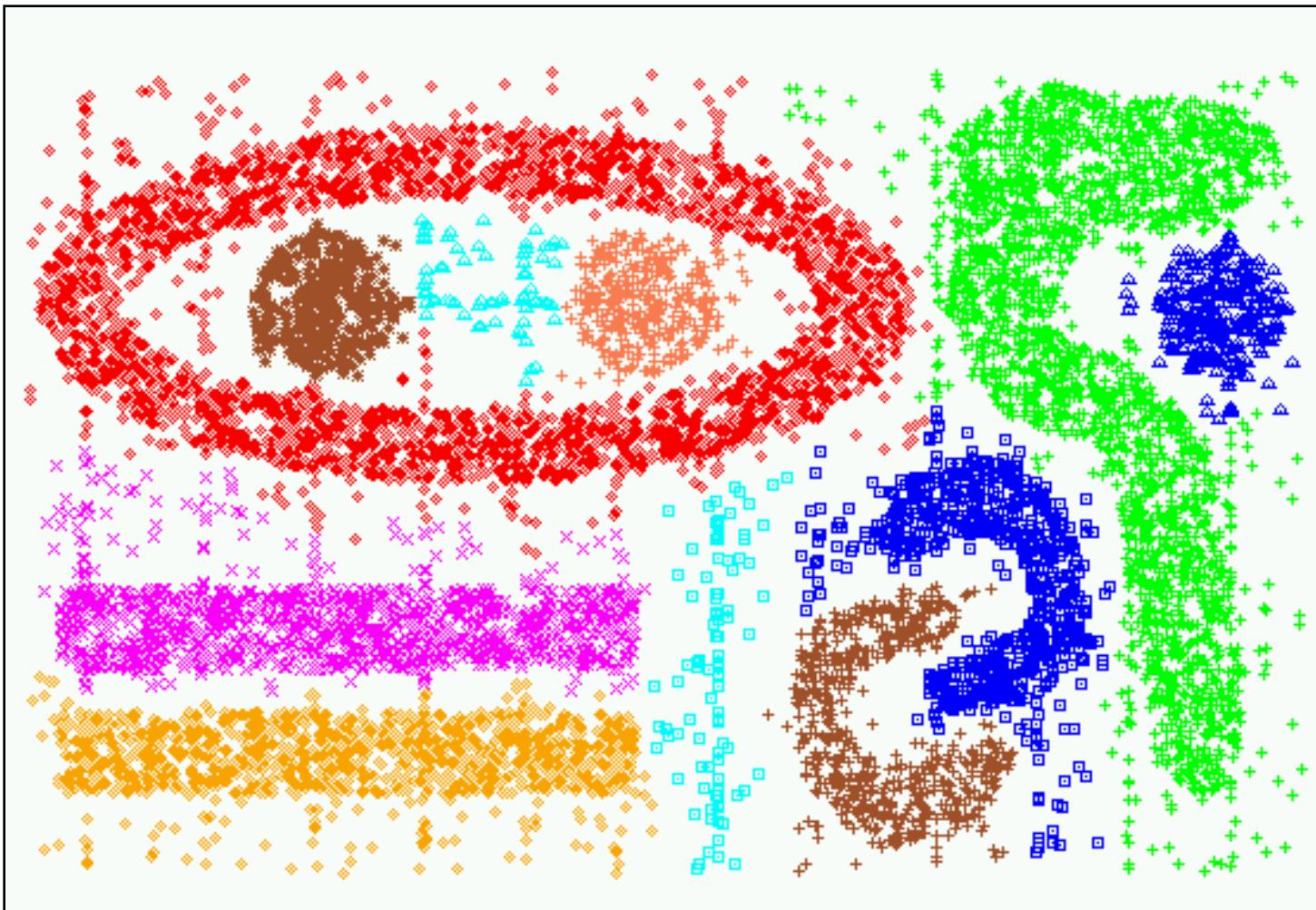
Experimental Results: SURE (# clusters)



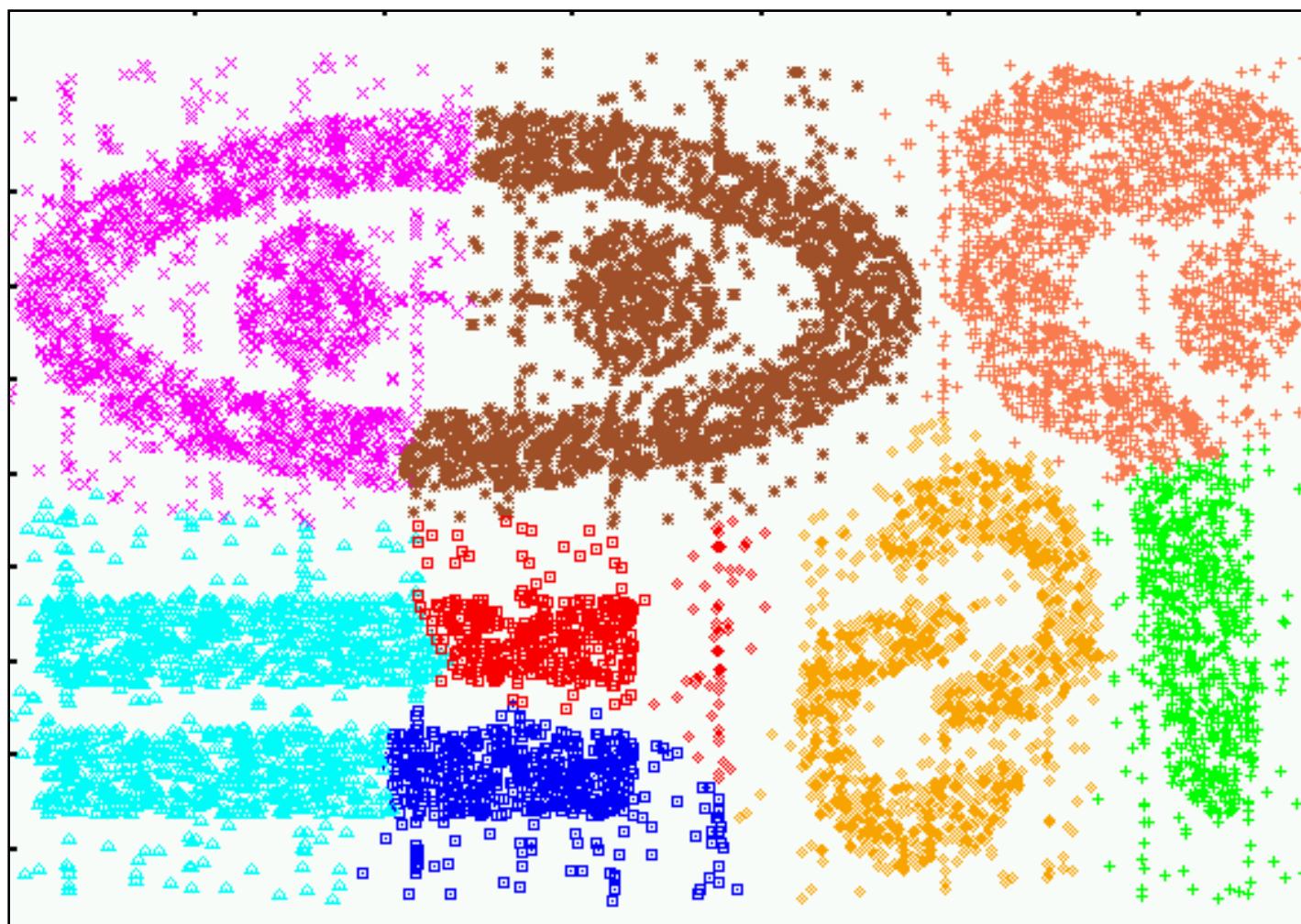
Experimental Results: CURE (15 clusters)



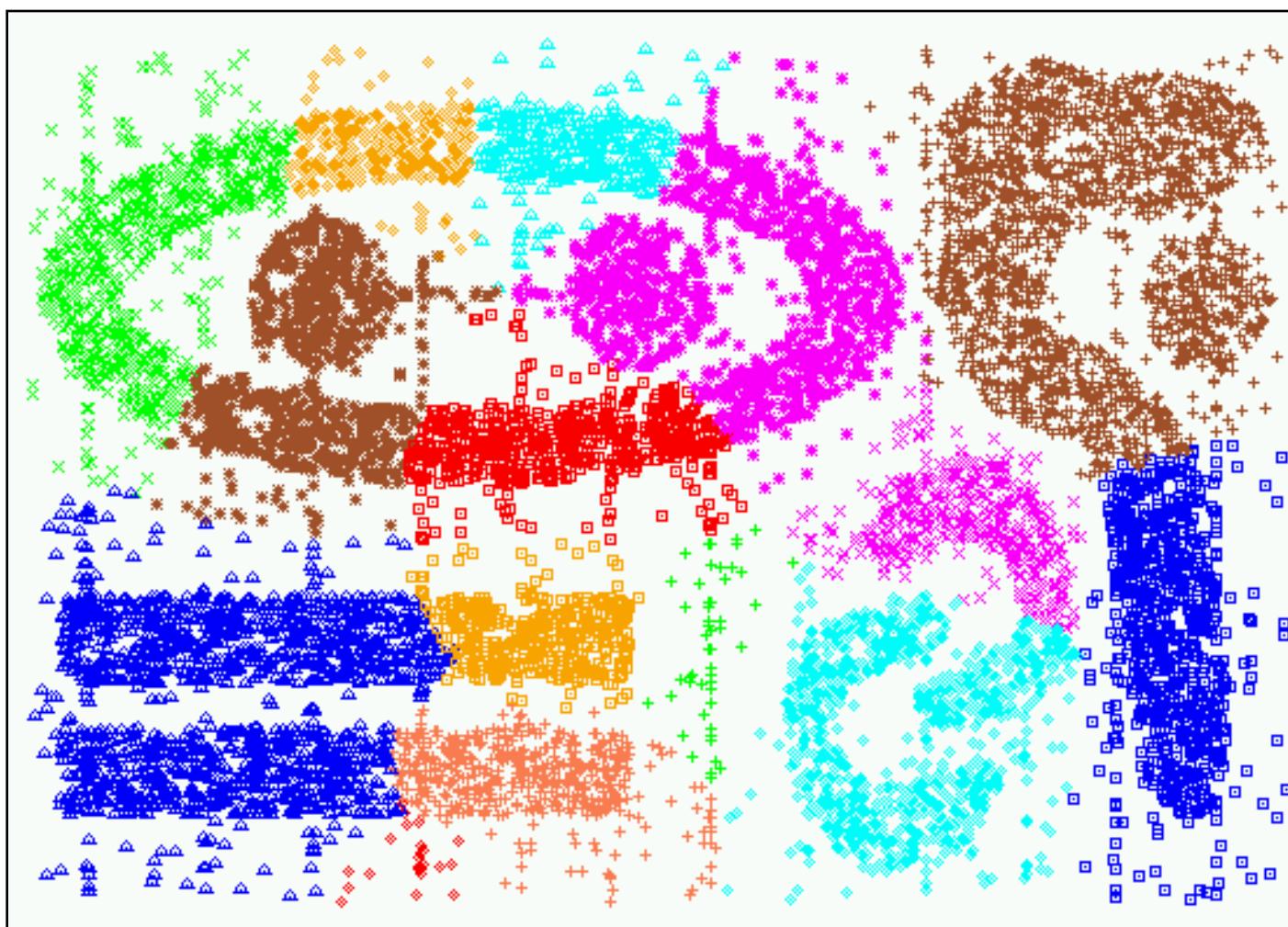
Experimental Results: CHAMELEON



Experimental Results: SURE (5 clusters)

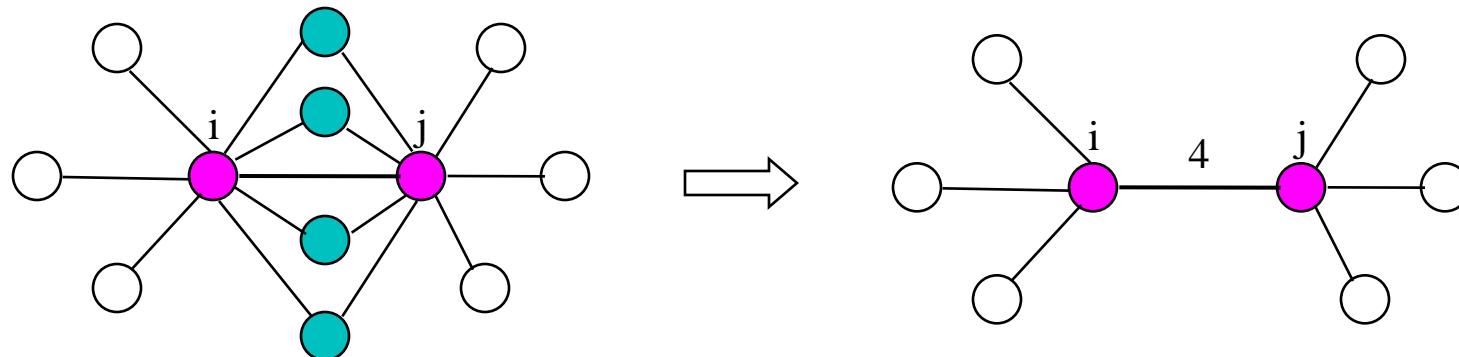


Experimental Results: SURE (75 clusters)

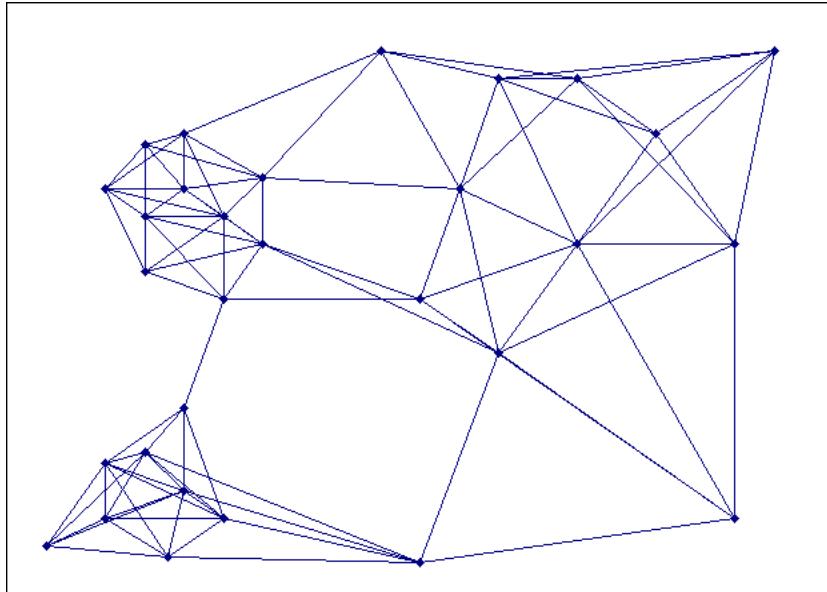


Shared Near Neighbor Approach

SNN graph: the weight of an edge is the number of shared neighbors between vertices given that the vertices are connected

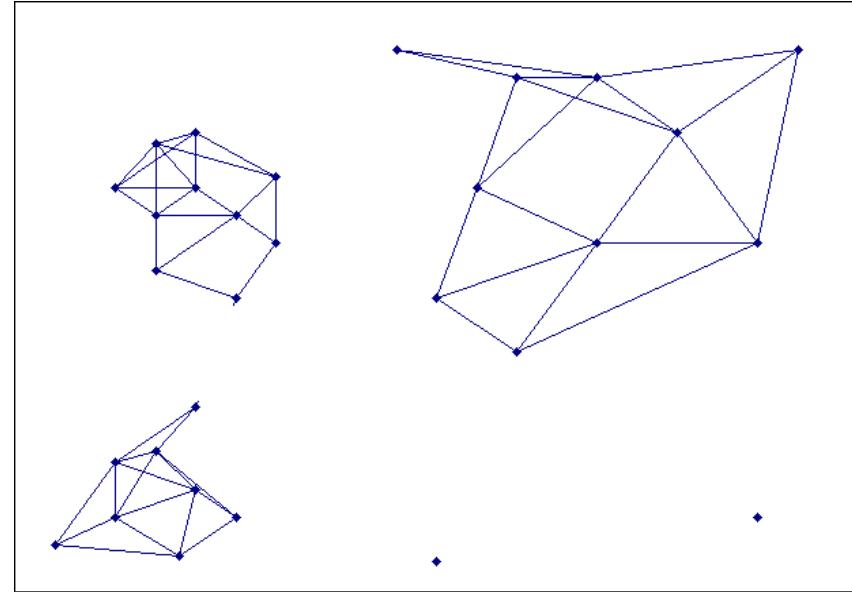


Creating the SNN Graph



Sparse Graph

**Link weights are similarities
between neighboring points**



Shared Near Neighbor Graph

**Link weights are number of
Shared Nearest Neighbors**

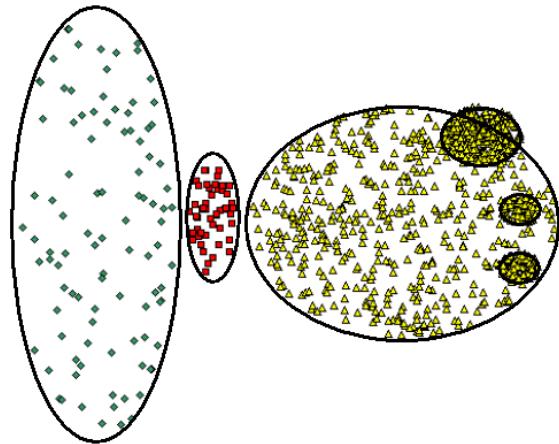
ROCK (RObust Clustering using linKs)

- Clustering algorithm for data with categorical and Boolean attributes
 - A pair of points is defined to be neighbors if their similarity is greater than some threshold
 - Use a hierarchical clustering scheme to cluster the data.
1. Obtain a sample of points from the data set
 2. Compute the link value for each set of points, i.e., transform the original similarities (computed by Jaccard coefficient) into similarities that reflect the number of shared neighbors between points
 3. Perform an agglomerative hierarchical clustering on the data using the “number of shared neighbors” as similarity measure and maximizing “the shared neighbors” objective function
 4. Assign the remaining points to the clusters that have been found

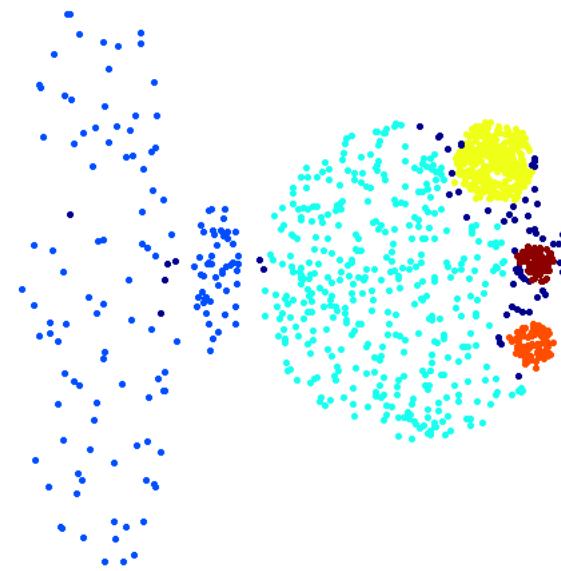
Jarvis-Patrick Clustering

- First, the k-nearest neighbors of all points are found
 - In graph terms this can be regarded as breaking all but the k strongest links from a point to other points in the proximity graph
- A pair of points is put in the same cluster if
 - any two points share more than T neighbors and
 - the two points are in each others k nearest neighbor list
- For instance, we might choose a nearest neighbor list of size 20 and put points in the same cluster if they share more than 10 near neighbors
- Jarvis-Patrick clustering is too brittle

When Jarvis-Patrick Works Reasonably Well

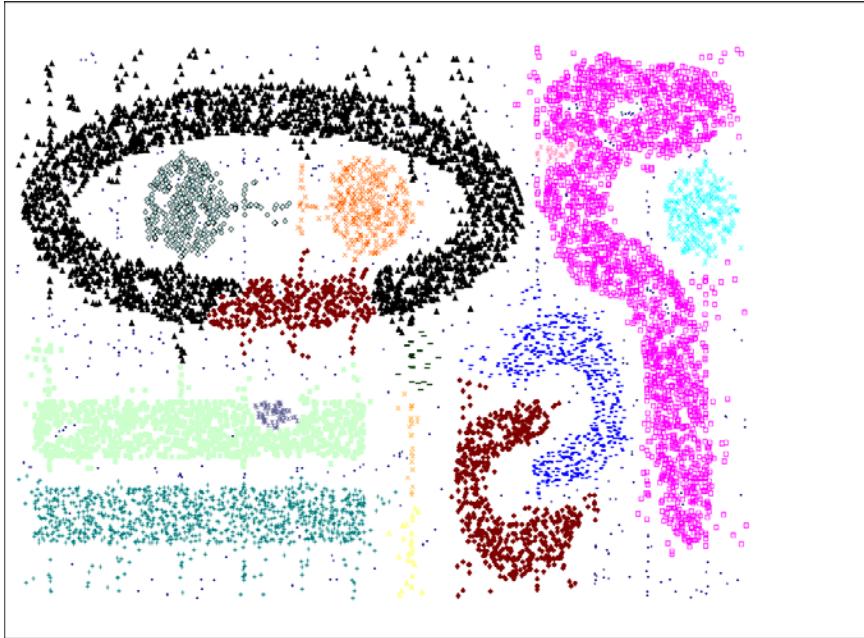


Original Points

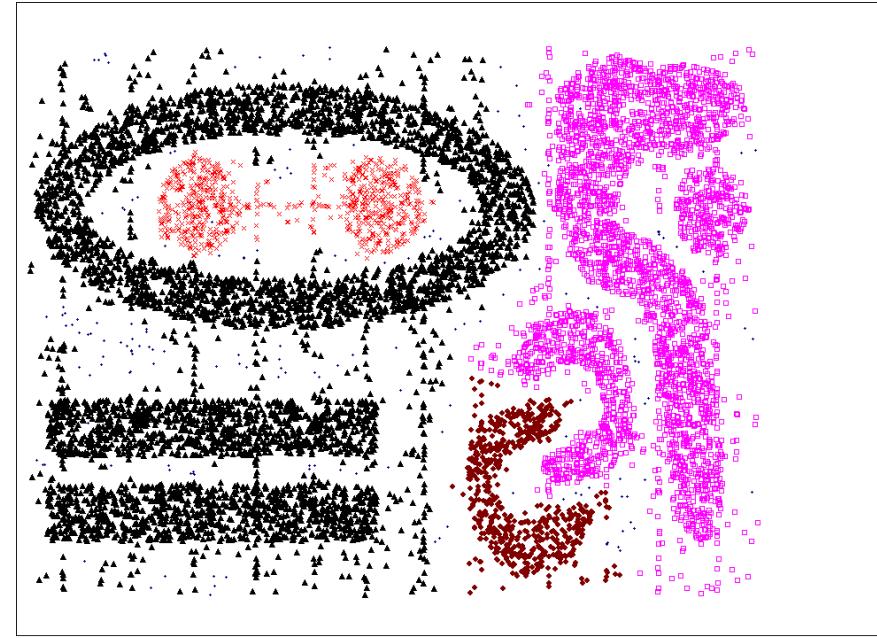


Jarvis Patrick Clustering
6 shared neighbors out of 20

When Jarvis-Patrick Does NOT Work Well



**Smallest threshold, T ,
that does not merge
clusters.**



Threshold of $T - 1$

SNN Clustering Algorithm

1. Compute the similarity matrix

This corresponds to a similarity graph with data points for nodes and edges whose weights are the similarities between data points

2. Sparsify the similarity matrix by keeping only the k most similar neighbors

This corresponds to only keeping the k strongest links of the similarity graph

3. Construct the shared nearest neighbor graph from the sparsified similarity matrix.

At this point, we could apply a similarity threshold and find the connected components to obtain the clusters (Jarvis-Patrick algorithm)

4. Find the SNN density of each Point.

Using a user specified parameters, Eps , find the number points that have an SNN similarity of Eps or greater to each point. This is the SNN density of the point

SNN Clustering Algorithm ...

5. Find the core points

Using a user specified parameter, $MinPts$, find the core points, i.e., all points that have an SNN density greater than $MinPts$

6. Form clusters from the core points

If two core points are within a radius, Eps , of each other they are placed in the same cluster

7. Discard all noise points

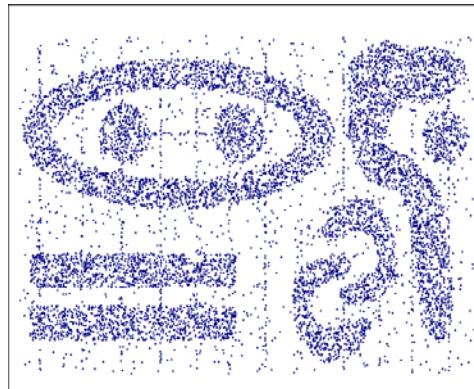
All non-core points that are not within a radius of Eps of a core point are discarded

8. Assign all non-noise, non-core points to clusters

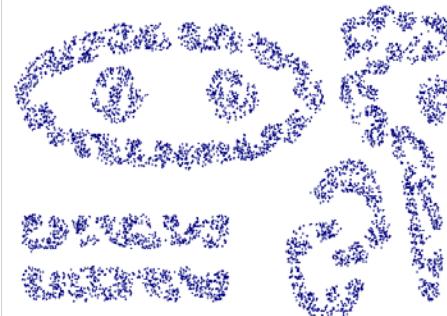
This can be done by assigning such points to the nearest core point

(Note that steps 4-8 are DBSCAN)

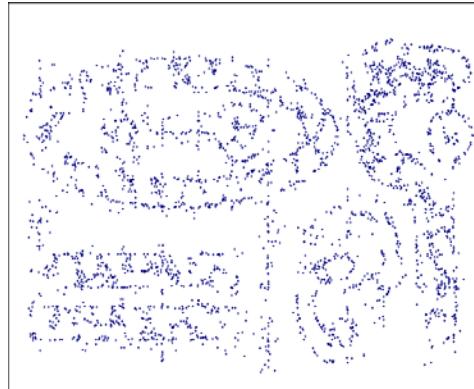
SNN Density



a) All Points



b) High SNN Density

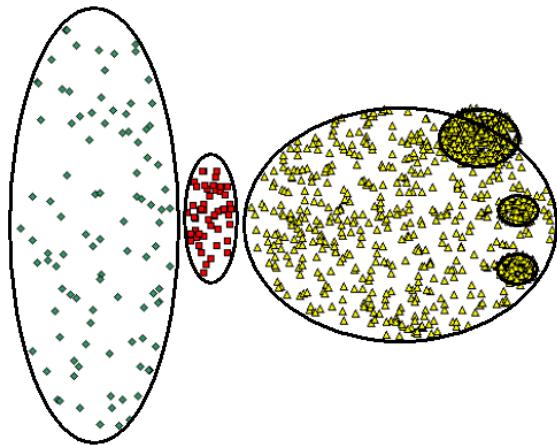


c) Medium SNN Density

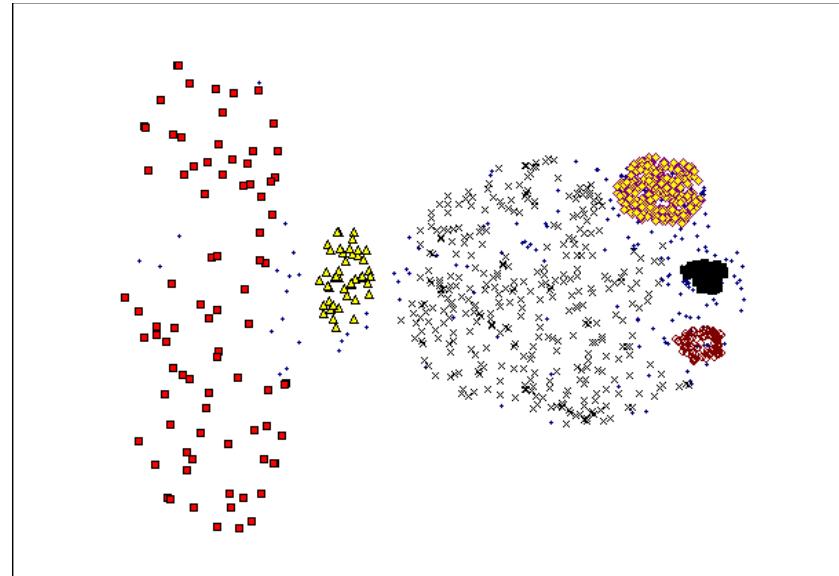


d) Low SNN Density

SNN Clustering Can Handle Differing Densities

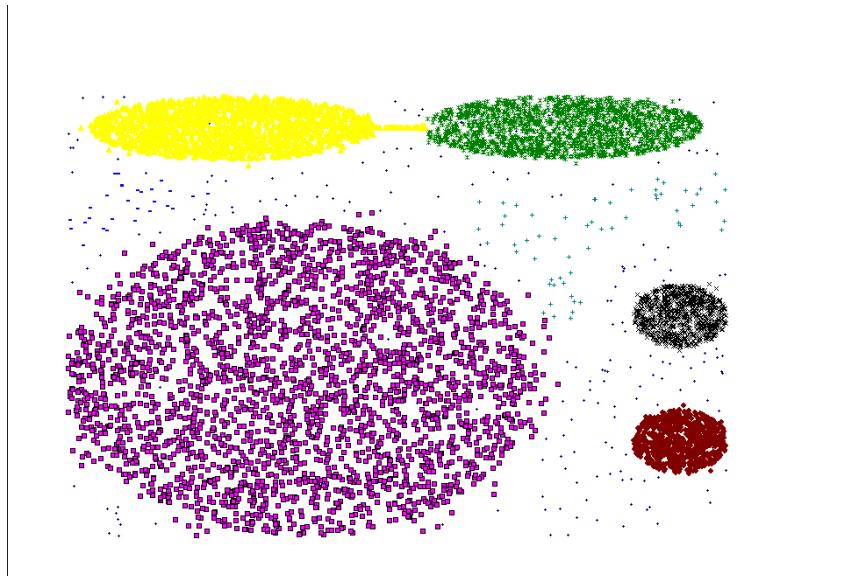
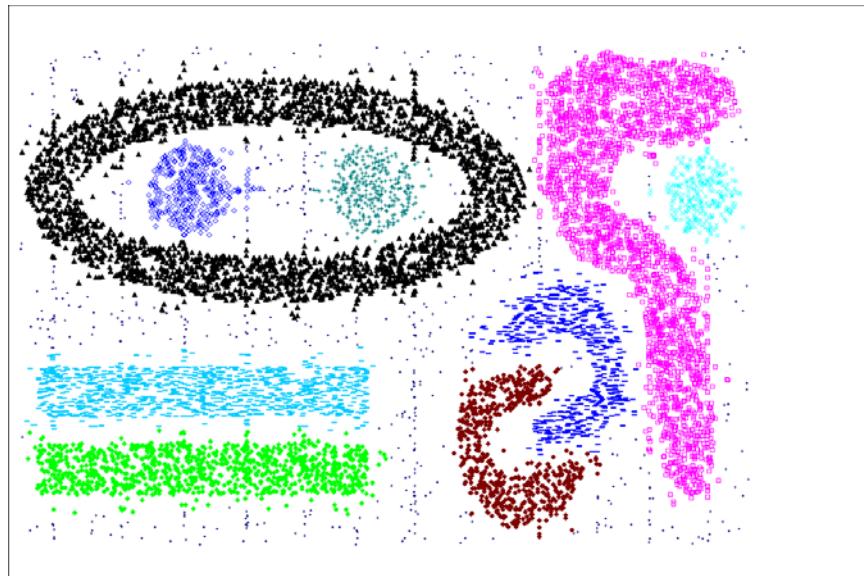


Original Points

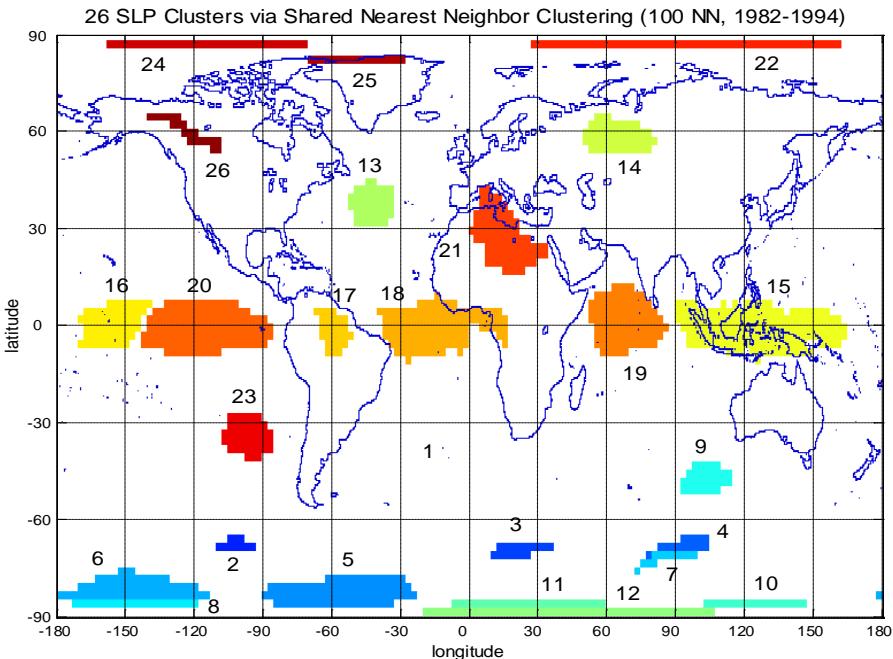


SNN Clustering

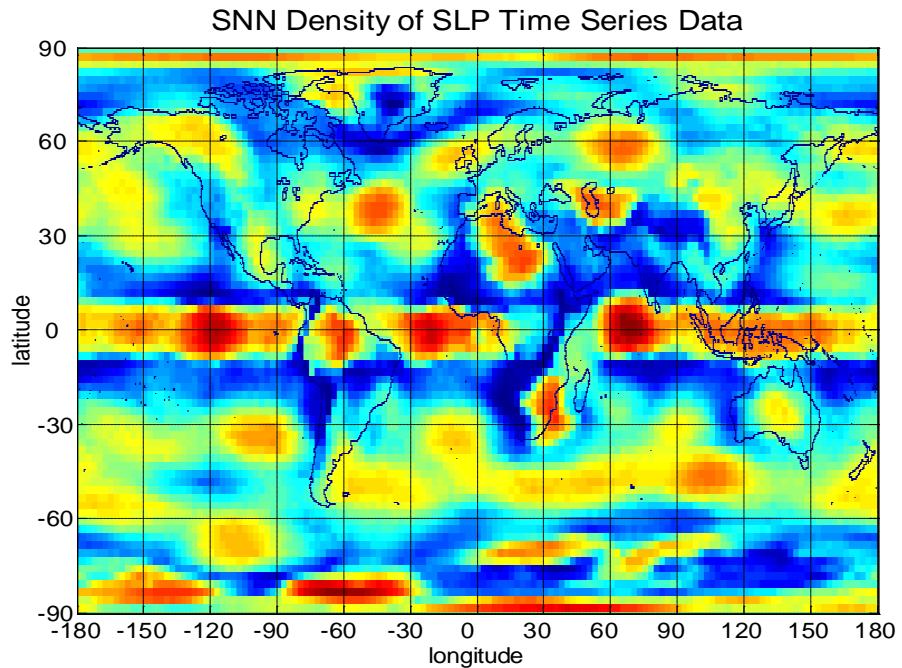
SNN Clustering Can Handle Other Difficult Situations



Finding Clusters of Time Series In Spatio-Temporal Data



SNN Clusters of SLP.



SNN Density of Points on the Globe.

Features and Limitations of SNN Clustering

- Does not cluster all the points
- Complexity of SNN Clustering is high
 - $O(n * \text{time to find numbers of neighbor within } Eps)$
 - In worst case, this is $O(n^2)$
 - For lower dimensions, there are more efficient ways to find the nearest neighbors
 - ◆ R* Tree
 - ◆ k-d Trees

Data Mining Anomaly Detection

Lecture Notes for Chapter 10

Introduction to Data Mining

by

Tan, Steinbach, Kumar

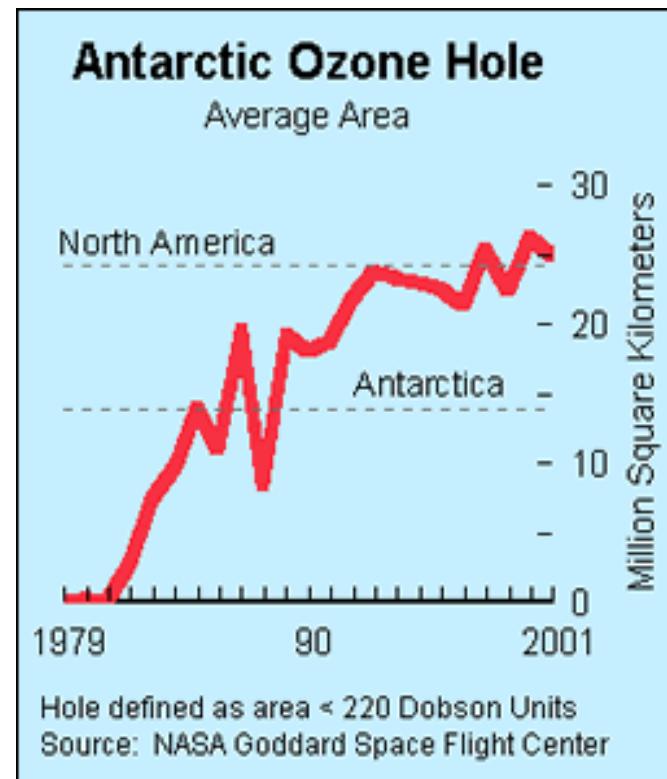
Anomaly/Outlier Detection

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

Importance of Anomaly Detection

Ozone Depletion History

- In 1985 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 ozone concentrations recorded by the satellite were so low they were being treated as outliers by a computer program and discarded!



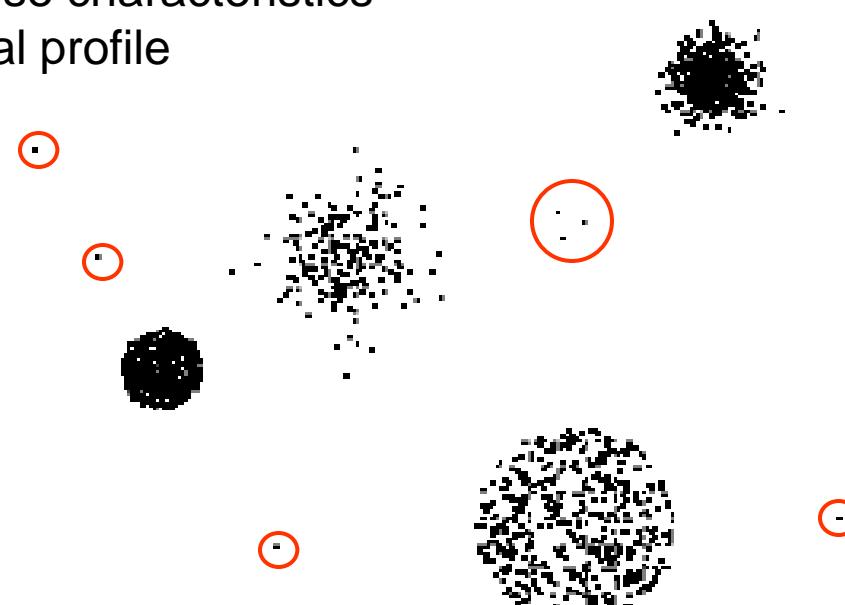
Sources:

<http://exploringdata.cqu.edu.au/ozone.html>
<http://www.epa.gov/ozone/science/hole/size.html>

Anomaly Detection

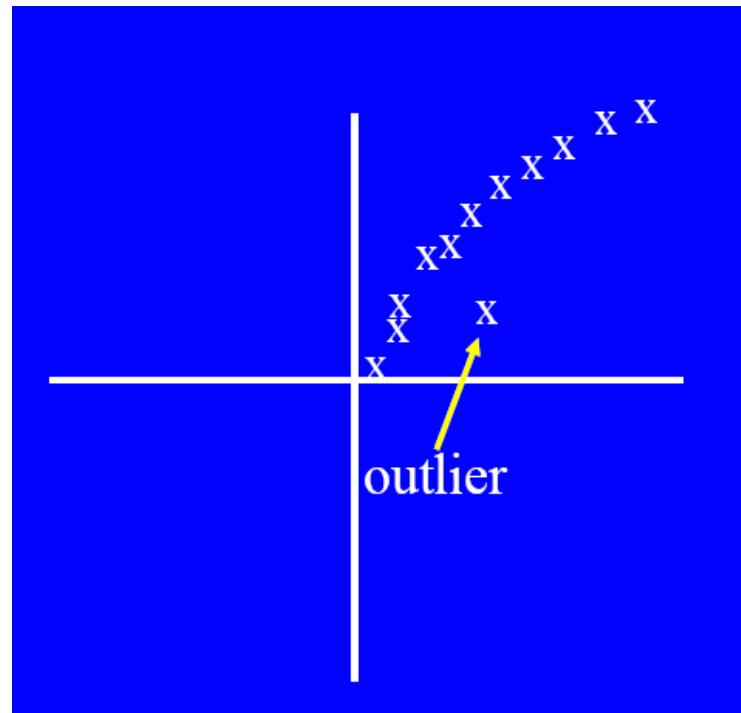
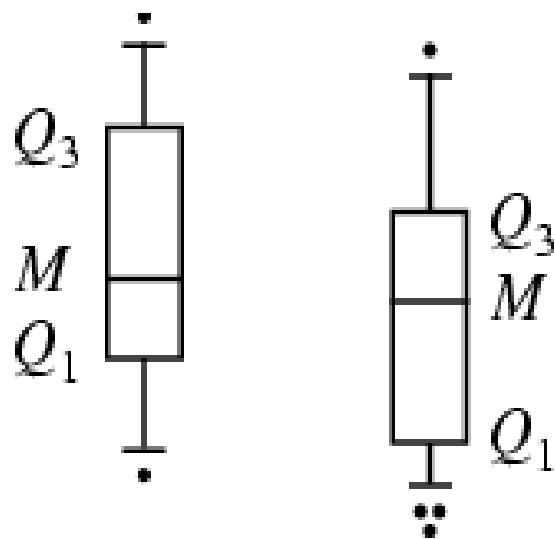
- Challenges
 - How many outliers are there in the data?
 - Method is unsupervised
 - ◆ Validation can be quite challenging (just like for clustering)
 - Finding needle in a haystack
- Working assumption:
 - There are considerably more “normal” observations than “abnormal” observations (outliers/anomalies) in the data

Anomaly Detection Schemes

- General Steps
 - Build a profile of the “normal” behavior
 - ◆ Profile can be patterns or summary statistics for the overall population
 - Use the “normal” profile to detect anomalies
 - ◆ Anomalies are observations whose characteristics differ significantly from the normal profile
 - Types of anomaly detection schemes
 - Graphical & Statistical-based
 - Distance-based
 - Model-based
- 

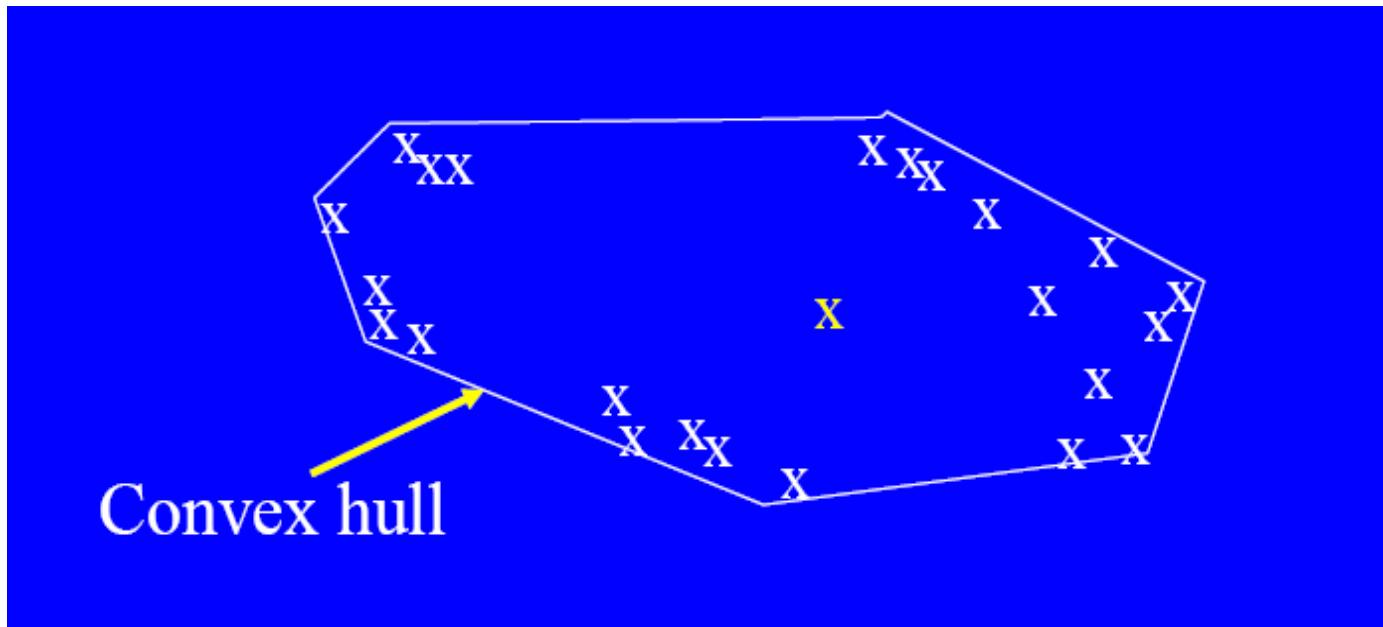
Graphical Approaches

- Boxplot (1-D), Scatter plot (2-D), Spin plot (3-D)
- Limitations
 - Time consuming
 - Subjective



Convex Hull Method

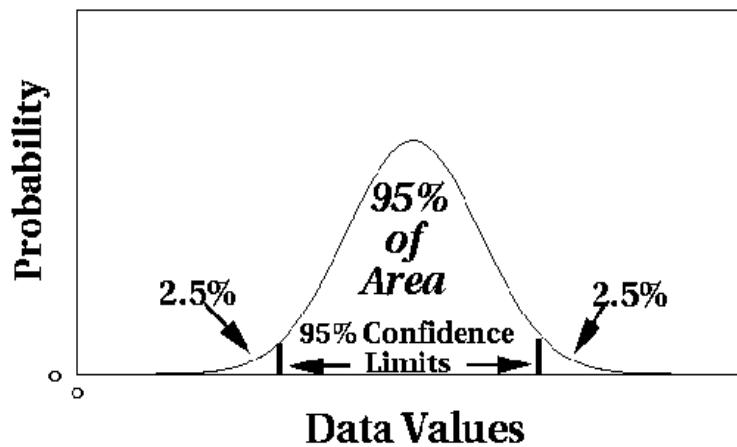
- Extreme points are assumed to be outliers
- Use convex hull method to detect extreme values



- What if the outlier occurs in the middle of the data?

Statistical Approaches

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



Grubbs' Test

- Detect outliers in univariate data
- Assume data comes from normal distribution
- Detects one outlier at a time, remove the outlier, and repeat
 - H_0 : There is no outlier in data
 - H_A : There is at least one outlier
- Grubbs' test statistic:
- Reject H_0 if:

$$G = \frac{\max|X - \bar{X}|}{S}$$
$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t^2_{(\alpha/2, N-2)}}{N-2 + t^2_{(\alpha/2, N-2)}}}$$

Statistical-based – Likelihood Approach

- Assume the data set D contains samples from a mixture of two probability distributions:
 - M (majority distribution)
 - A (anomalous distribution)
- General Approach:
 - Initially, assume all the data points belong to M
 - Let $L_t(D)$ be the log likelihood of D at time t
 - For each point x_t that belongs to M, move it to A
 - ◆ Let $L_{t+1}(D)$ be the new log likelihood.
 - ◆ Compute the difference, $\Delta = L_t(D) - L_{t+1}(D)$
 - ◆ If $\Delta > c$ (some threshold), then x_t is declared as an anomaly and moved permanently from M to A

Statistical-based – Likelihood Approach

- Data distribution, $D = (1 - \lambda) M + \lambda A$
- M is a probability distribution estimated from data
 - Can be based on any modeling method (naïve Bayes, maximum entropy, etc)
- A is initially assumed to be uniform distribution
- Likelihood at time t:

$$L_t(D) = \prod_{i=1}^N P_D(x_i) = \left((1 - \lambda)^{|M_t|} \prod_{x_i \in M_t} P_{M_t}(x_i) \right) \left(\lambda^{|A_t|} \prod_{x_i \in A_t} P_{A_t}(x_i) \right)$$
$$LL_t(D) = |M_t| \log(1 - \lambda) + \sum_{x_i \in M_t} \log P_{M_t}(x_i) + |A_t| \log \lambda + \sum_{x_i \in A_t} \log P_{A_t}(x_i)$$

Limitations of Statistical Approaches

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

Distance-based Approaches

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

Nearest-Neighbor Based Approach

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

Outliers in Lower Dimensional Projection

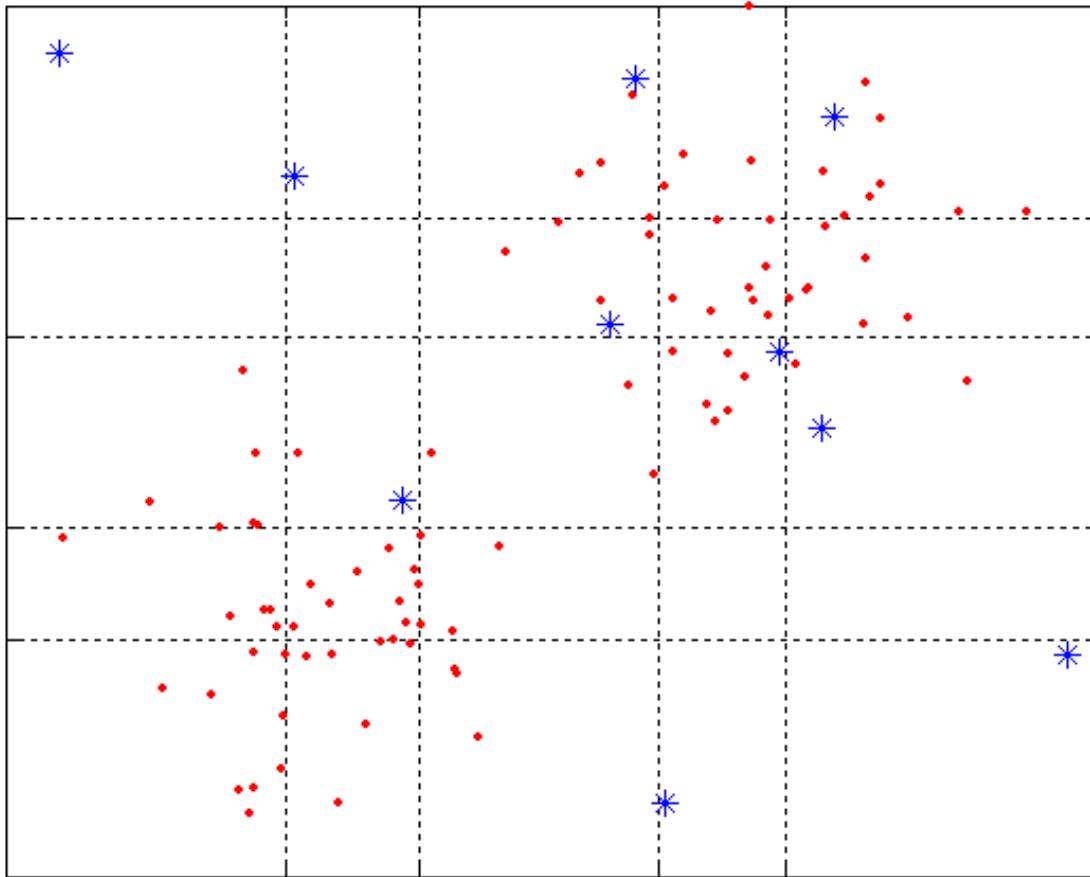
- Divide each attribute into ϕ equal-depth intervals
 - Each interval contains a fraction $f = 1/\phi$ of the records
- Consider a k -dimensional cube created by picking grid ranges from k different dimensions
 - If attributes are independent, we expect region to contain a fraction f^k of the records
 - If there are N points, we can measure sparsity of a cube D as:

$$S(D) = \frac{n(D) - N \cdot f^k}{\sqrt{N \cdot f^k \cdot (1 - f^k)}}$$

- Negative sparsity indicates cube contains smaller number of points than expected

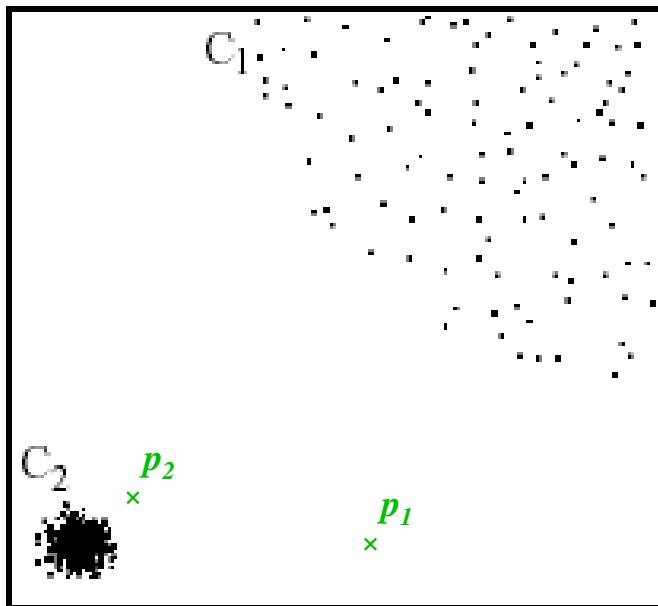
Example

- $N=100, \phi = 5, f = 1/5 = 0.2, N \times f^2 = 4$



Density-based: LOF approach

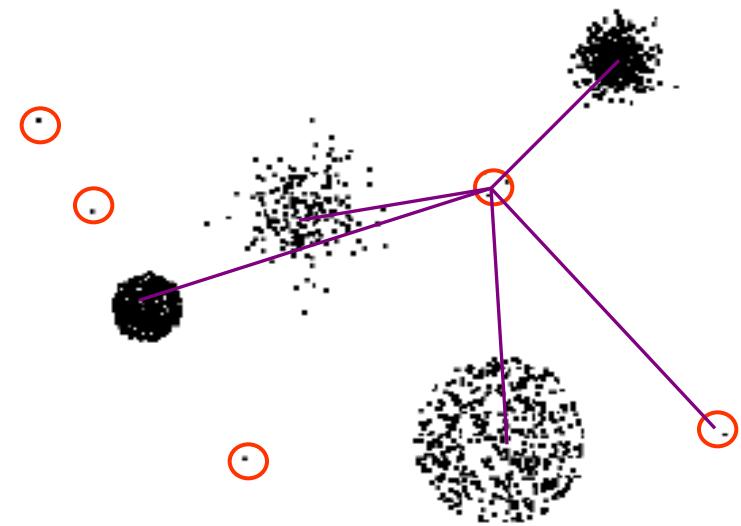
- For each point, compute the density of its local neighborhood
- Compute local outlier factor (LOF) of a sample p as the average of the ratios of the density of sample p and the density of its nearest neighbors
- \hat{C}_1 contains p_1 and p_2 with largest LOF value



In the NN approach, p_2 is not considered as outlier, while LOF approach find both p_1 and p_2 as outliers

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

- 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.² Suppose that your doctor performs a test that is 99% accurate, i.e. when the test was administered to a test population all of whom had the disease, 99% of the tests indicated disease, and likewise, when the test population was known to be 100% free of the disease, 99% of the test results were negative. Upon visiting your doctor to learn the results he tells you he has good news and bad news. The bad news is that indeed you tested positive for the disease. The good news however, is that out of the entire population the rate of incidence is only 1/10000, i.e. only 1 in 10000 people have this ailment. What, given this information, is the probability of you having the disease? The reader is encouraged to make a quick “guesstimate” of the answer at this point.

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

$$P(I|A) = \frac{P(I) \cdot P(A|I)}{P(I) \cdot P(A|I) + P(\neg I) \cdot P(A|\neg I)}$$

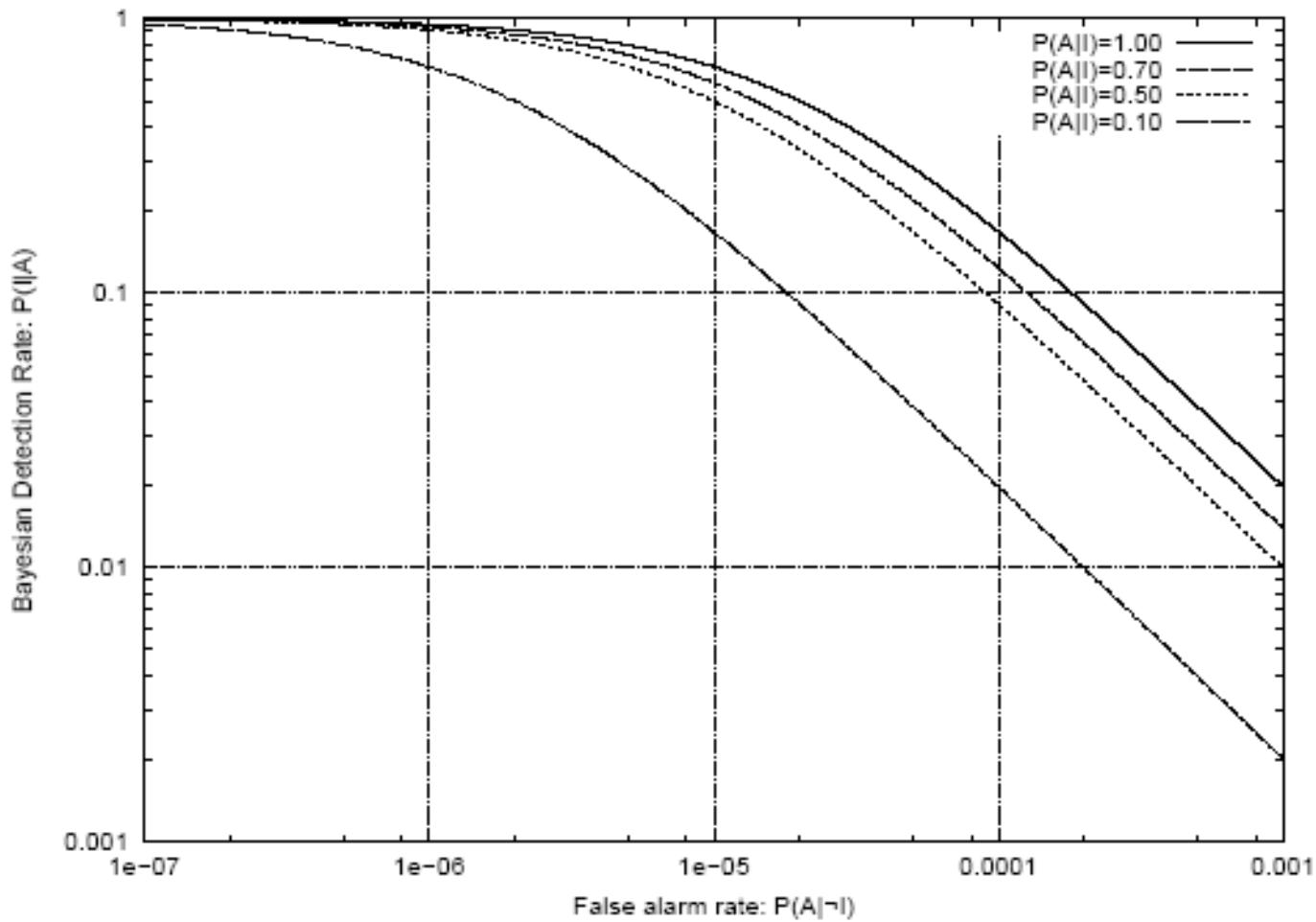
- Suppose: $P(I) = 1 / \frac{1 \cdot 10^6}{2 \cdot 10} = 2 \cdot 10^{-5}$;
 $P(\neg I) = 1 - P(I) = 0.99998$

- Then:

$$P(I|A) = \frac{2 \cdot 10^{-5} \cdot P(A|I)}{2 \cdot 10^{-5} \cdot P(A|I) + 0.99998 \cdot P(A|\neg I)}$$

- False alarm rate becomes more dominant if $P(I)$ is very low

Detection Rate vs False Alarm Rate



- Axelsson: We need a very low false alarm rate to achieve a reasonable Bayesian detection rate