

Data Mining: Introduction

Lecture Notes for Chapter 1

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

1

Large-scale Data is Everywhere!

- There has been enormous data growth in both commercial and scientific databases due to advances in data generation and collection technologies
- New mantra
 - Gather whatever data you can whenever and wherever possible.
- Expectations
 - Gathered data will have value either for the purpose collected or for a purpose not envisioned.



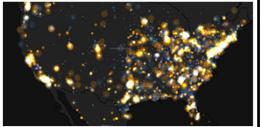
Cyber Security



E-Commerce



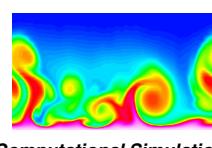
Traffic Patterns



Social Networking: Twitter



Sensor Networks



Computational Simulations

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

2

Why Data Mining? Commercial Viewpoint

- Lots of data is being collected and warehoused
 - Web data
 - ◆ Google has Peta Bytes of web data
 - ◆ Facebook has billions of active users
 - purchases at department/grocery stores, e-commerce
 - ◆ Amazon handles millions of visits/day
 - 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)



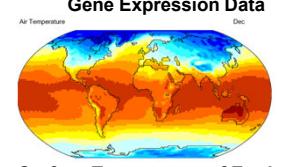
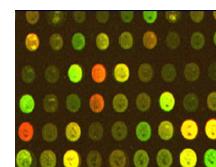
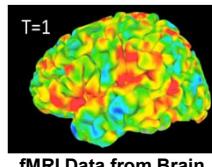
09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

3

Why Data Mining? Scientific Viewpoint

- Data collected and stored at enormous speeds
 - remote sensors on a satellite
 - ◆ NASA EOSDIS archives over petabytes of earth science data / year
 - telescopes scanning the skies
 - ◆ Sky survey data
 - High-throughput biological data
 - scientific simulations
 - ◆ terabytes of data generated in a few hours
- Data mining helps scientists
 - in automated analysis of massive datasets
 - In hypothesis formation



09/09/2020

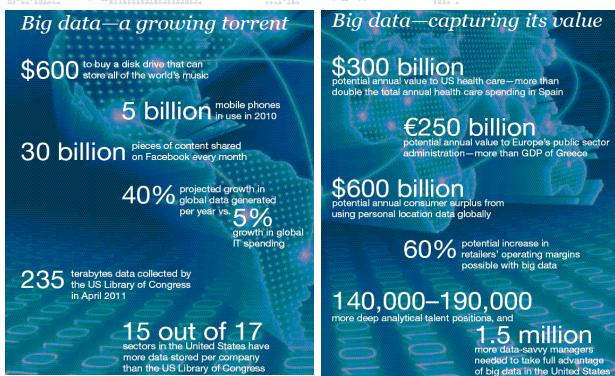
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

4

Great opportunities to improve productivity in all walks of life

McKinsey Global Institute

Big data: The next frontier for innovation, competition, and productivity



09/09/2020

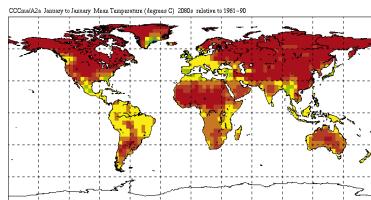
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

5

Great Opportunities to Solve Society's Major Problems



Improving health care and reducing costs



Predicting the impact of climate change



Finding alternative/ green energy sources



Reducing hunger and poverty by increasing agriculture production

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

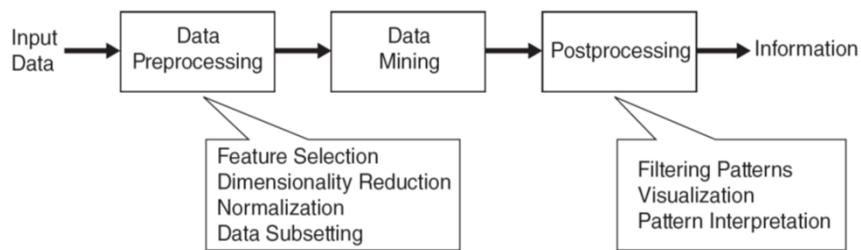
6

6

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



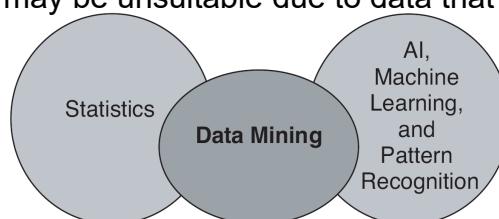
09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

7

Origins of Data Mining

- Draws ideas from machine learning/AI, pattern recognition, statistics, and database systems
- Traditional techniques may be unsuitable due to data that is
 - Large-scale
 - High dimensional
 - Heterogeneous
 - Complex
 - Distributed
- A key component of the emerging field of data science and data-driven discovery



Database Technology, Parallel Computing, Distributed Computing

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

8

8

Data Mining Tasks

● Prediction Methods

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

● Description Methods

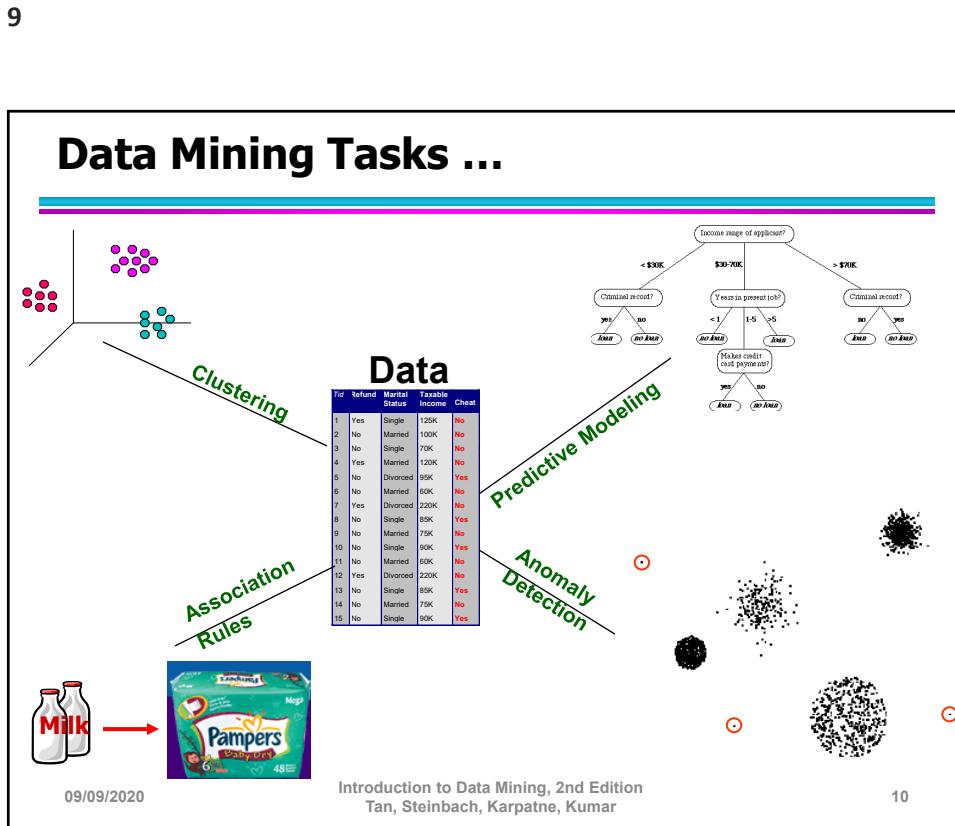
- Find human-interpretable patterns that describe the data.

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

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

9



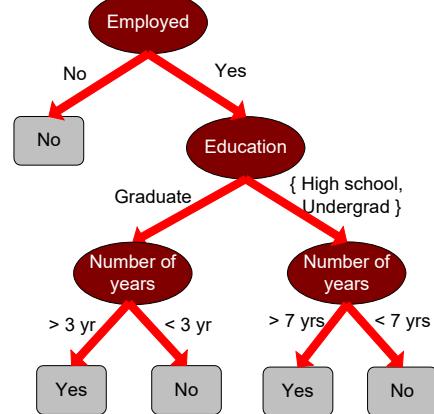
10

Predictive Modeling: Classification

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

Model for predicting credit worthiness

Class				
Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Graduate	5	Yes
2	Yes	High School	2	No
3	No	Undergrad	1	No
4	Yes	High School	10	Yes
...



09/09/2020

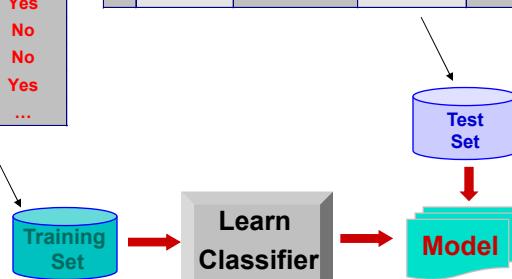
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

11

Classification Example

					categorical	categorical	quantitative	class
Tid	Employed	Level of Education	# years at present address	Credit Worthy				
1	Yes	Graduate	5	Yes				
2	Yes	High School	2	No				
3	No	Undergrad	1	No				
4	Yes	High School	10	Yes				
...				

Tid	Employed	Level of Education	# years at present address	Credit Worthy
1	Yes	Undergrad	7	?
2	No	Graduate	3	?
3	Yes	High School	2	?
...



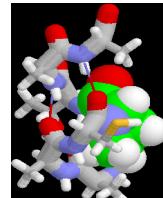
09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

12

Examples of Classification Task

- Classifying credit card transactions as legitimate or fraudulent
- Classifying land covers (water bodies, urban areas, forests, etc.) using satellite data
- Categorizing news stories as finance, weather, entertainment, sports, etc
- Identifying intruders in the cyberspace
- Predicting tumor cells as benign or malignant
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil



09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

13

Classification: Application 1

● 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.

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

14

14

Classification: Application 2

- Churn prediction for telephone customers
 - **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

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

15

Classification: Application 3

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

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

09/09/2020

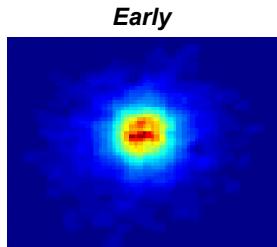
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

16

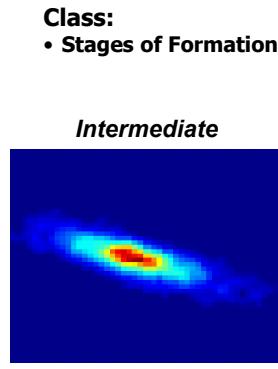
16

Classifying Galaxies

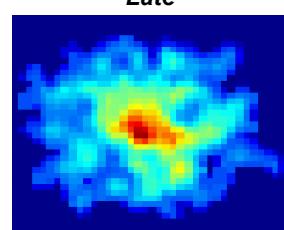
Courtesy: <http://aps.umn.edu>



Early



Intermediate



Late

Data Size:

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

09/09/2020 Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

17

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.
- Extensively 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.

09/09/2020

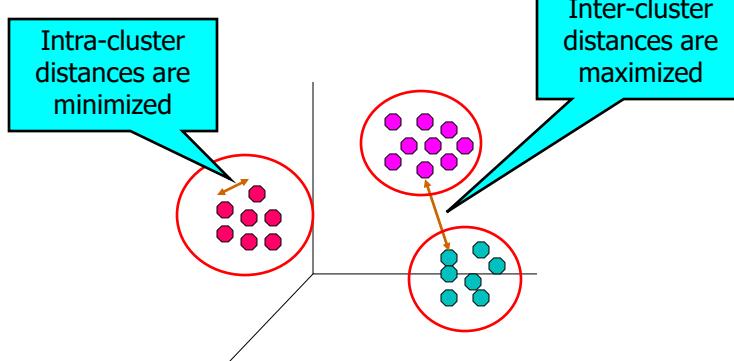
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

18

18

Clustering

- 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



09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

19

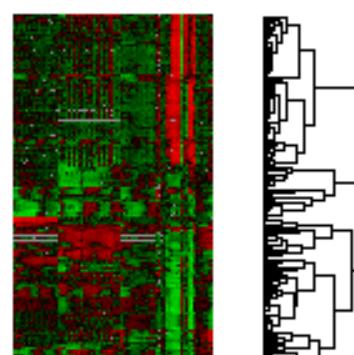
19 Applications of Cluster Analysis

Understanding

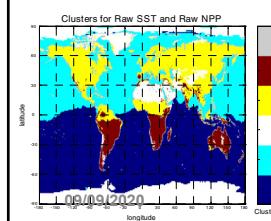
- Custom profiling for targeted marketing
- Group related documents for browsing
- Group genes and proteins that have similar functionality
- Group stocks with similar price fluctuations

Summarization

- Reduce the size of large data sets



Courtesy: Michael Eisen



Use of K-means to partition Sea Surface Temperature (SST) and Net Primary Production (NPP) into clusters that reflect the Northern and Southern Hemispheres.

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar



20

20

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.

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

21

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.

Enron email dataset



09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

22

22

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:
 $\{Milk\} \rightarrow \{Coke\}$
 $\{Diaper, Milk\} \rightarrow \{Beer\}$

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

23

Association Analysis: Applications

- Market-basket analysis
 - Rules are used for sales promotion, shelf management, and inventory management
- Telecommunication alarm diagnosis
 - Rules are used to find combination of alarms that occur together frequently in the same time period
- Medical Informatics
 - Rules are used to find combination of patient symptoms and test results associated with certain diseases

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

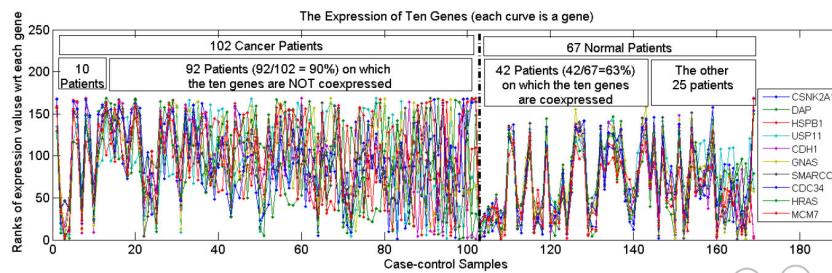
24

24

Association Analysis: Applications

- An Example Subspace Differential Coexpression Pattern from lung cancer dataset

Three lung cancer datasets [Bhattacharjee et al. 2001], [Stearman et al. 2005], [Su et al. 2007]

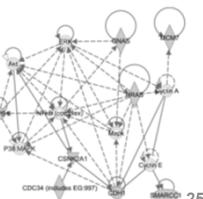


Enriched with the TNF/NFB signaling pathway
which is well-known to be related to lung cancer
P-value: 1.4×10^{-5} (6/10 overlap with the pathway)

[Fang et al PSB 2010]

09/09/2020

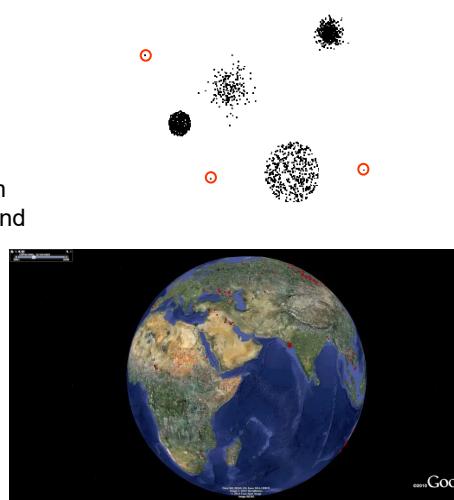
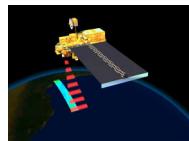
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar



25

Deviation/Anomaly/Change Detection

- Detect significant deviations from normal behavior
- Applications:
 - Credit Card Fraud Detection
 - Network Intrusion Detection
 - Identify anomalous behavior from sensor networks for monitoring and surveillance.
 - Detecting changes in the global forest cover.



09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

26

26

Motivating Challenges

- Scalability
- High Dimensionality
- Heterogeneous and Complex Data
- Data Ownership and Distribution
- Non-traditional Analysis

09/09/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Kumar

27

Data Mining: Data

Lecture Notes for Chapter 2

Introduction to Data Mining , 2nd Edition
by
Tan, Steinbach, Kumar

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

1

Outline

- Attributes and Objects
- Types of Data
- Data Quality
- Similarity and Distance
- Data Preprocessing

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

2

2

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, dimension, or feature
- A collection of attributes describe an **object**
 - Object is also known as record, point, case, sample, entity, or instance

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

3

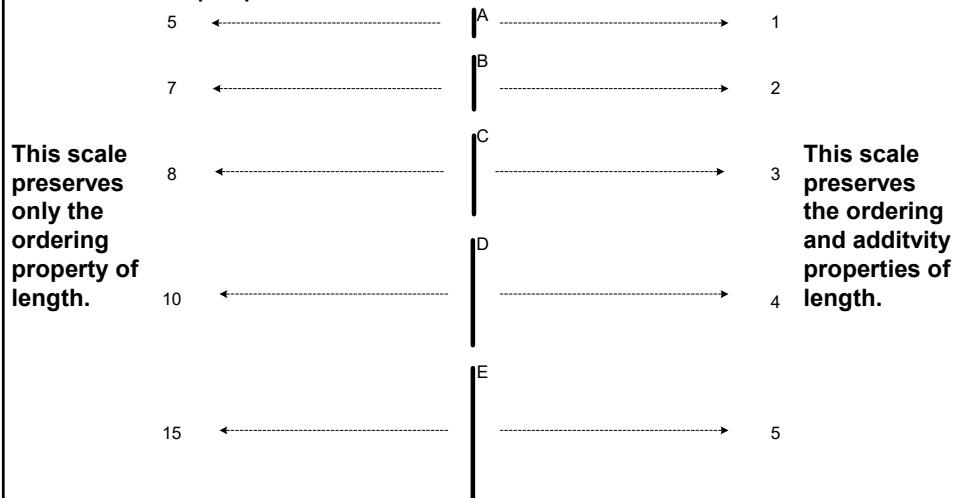
Attribute Values

- **Attribute values** are numbers or symbols assigned to an attribute for a particular object
- 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 can be different than the properties of the values used to represent the attribute

4

Measurement of Length

- The way you measure an attribute may not match the attribute's properties.



5

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 {tall, medium, short}
 - Interval**
 - Examples: calendar dates, temperatures in Celsius or Fahrenheit.
 - Ratio**
 - Examples: temperature in Kelvin, length, counts, elapsed time (e.g., time to run a race)

6

Properties of Attribute Values

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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

7

Difference Between Ratio and Interval

- Is it physically meaningful to say that a temperature of 10° is twice that of 5° on
 - the Celsius scale?
 - the Fahrenheit scale?
 - the Kelvin scale?
- Consider measuring the height above average
 - If Bill's height is three inches above average and Bob's height is six inches above average, then would we say that Bob is twice as tall as Bill?
 - Is this situation analogous to that of temperature?

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

8

8

	Attribute Type	Description	Examples	Operations
Categorical Qualitative Numeric Quantitative	Nominal	Nominal attribute values only distinguish. ($=$, \neq)	zip codes, employee ID numbers, eye color, sex: {male, female}	mode, entropy, contingency correlation, χ^2 test
	Ordinal	Ordinal attribute values also order objects. ($<$, $>$)	hardness of minerals, {good, better, best}, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
	Interval	For interval attributes, differences between values are meaningful. (+, -)	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, current	geometric mean, harmonic mean, percent variation

This categorization of attributes is due to S. S. Stevens

9

	Attribute Type	Transformation	Comments
Categorical Qualitative Numeric Quantitative	Nominal	Any permutation of values	If all employee ID numbers were reassigned, would it make any difference?
	Ordinal	An order preserving change of values, i.e., $new_value = f(old_value)$ where f is a monotonic function	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	$new_value = a * old_value + b$ 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	$new_value = a * old_value$	Length can be measured in meters or feet.

This categorization of attributes is due to S. S. Stevens

10

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.

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

11

11

Asymmetric Attributes

● Only presence (a non-zero attribute value) is regarded as important

- ◆ Words present in documents
- ◆ Items present in customer transactions

● If we met a friend in the grocery store would we ever say the following?

"I see our purchases are very similar since we didn't buy most of the same things."

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

12

12

Critiques of the attribute categorization

- Incomplete
 - Asymmetric binary
 - Cyclical
 - Multivariate
 - Partially ordered
 - Partial membership
 - Relationships between the data
- Real data is approximate and noisy
 - This can complicate recognition of the proper attribute type
 - Treating one attribute type as another may be approximately correct

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

13

Key Messages for Attribute Types

- The types of operations you choose should be “meaningful” for the type of data you have
 - Distinctness, order, meaningful intervals, and meaningful ratios are only four (among many possible) properties of data
 - The data type you see – often numbers or strings – may not capture all the properties or may suggest properties that are not present
 - Analysis may depend on these other properties of the data
 - ◆ Many statistical analyses depend only on the distribution
 - In the end, what is meaningful can be specific to domain

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

14

14

Important Characteristics of Data

- Dimensionality (number of attributes)
 - ◆ High dimensional data brings a number of challenges
- Sparsity
 - ◆ Only presence counts
- Resolution
 - ◆ Patterns depend on the scale
- Size
 - ◆ Type of analysis may depend on size of data

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

15

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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

16

16

Record Data

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

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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

17

17

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 a 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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

18

18

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	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

19

19

Transaction Data

- A special type of data, where
 - Each 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.
 - Can represent transaction data as record data

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

09/14/2020

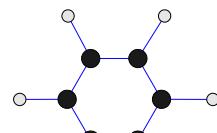
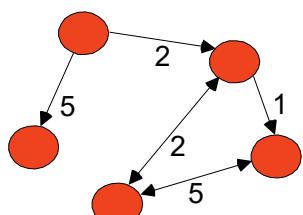
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

20

20

Graph Data

- Examples: Generic graph, a molecule, and webpages



Benzene Molecule: C₆H₆

Useful Links:

- [Bibliography](#)
- Other Useful Web sites
 - [ACM SIGKDD](#)
 - [KDnuggets](#)
 - [The Data Mine](#)

Knowledge Discovery and Data Mining Bibliography

(Gets updated frequently, so visit often!)

- [Books](#)
- [General Data Mining](#)

Book References in Data Mining and Knowledge Discovery

- Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Iyer, "Advances in Knowledge Discovery and Data Mining", AAAI Press/The MIT Press, 1996.
J. Ross Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, 1993.
Michael Berry and Gordon Linoff, "Data Mining Techniques (For Marketing, Sales, and Customer Support)", John Wiley & Sons, 1997.

General Data Mining

- Usama Fayyad, "Mining Databases: Towards Algorithms for Knowledge Discovery", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, vol. 21, no. 1, March 1998.
Christopher Matheus, Philip Chan, and Gregory Piatetsky-Shapiro, "Systems for knowledge discovery in databases", IEEE Transactions on Knowledge and Data Engineering, 5(6):903-913, December 1993.

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

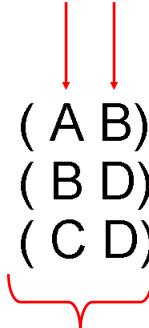
21

21

Ordered Data

- Sequences of transactions

Items/Events



An element of
the sequence

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

22

22

Ordered Data

- Genomic sequence data

```
GGTTCCGCCTTCAGCCCCGCGCC  
CGCAGGGCCCAGCCCCGCGCCGTC  
GAGAAGGGCCCAGCCCCGCGCCG  
GGGGGAGGCAGGGCCGCCCCGAGC  
CCAACCGAGTCCGACCAGGTGCC  
CCCTCTGCTCGGCCTAGACCTGA  
GCTCATTAGGCAGCAGCGAACAG  
GCCAAGTAGAACACCGCGAACAGC  
TGGGCTGCCTGCTGCGACCAGGG
```

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

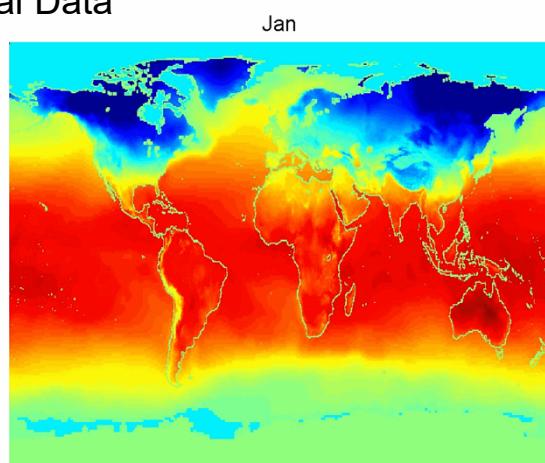
23

23

Ordered Data

- Spatio-Temporal Data

Average Monthly
Temperature of
land and ocean



09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

24

24

Data Quality

- Poor data quality negatively affects many data processing efforts
- Data mining example: a classification model for detecting people who are loan risks is built using poor data
 - Some credit-worthy candidates are denied loans
 - More loans are given to individuals that default

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

25

25

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
 - Wrong data
 - Fake data
 - Missing values
 - Duplicate data

09/14/2020

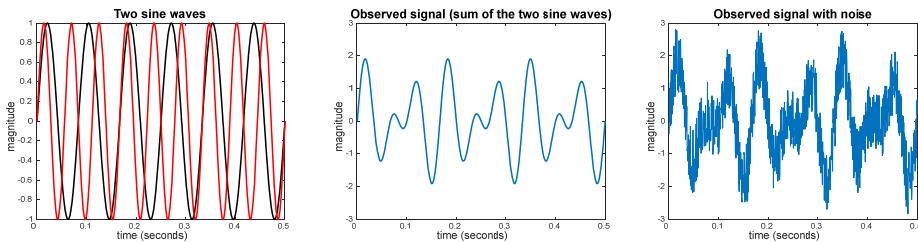
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

26

26

Noise

- For objects, noise is an extraneous object
- For attributes, 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
 - The figures below show two sine waves of the same magnitude and different frequencies, the waves combined, and the two sine waves with random noise
 - ◆ The magnitude and shape of the original signal is distorted



09/14/2020

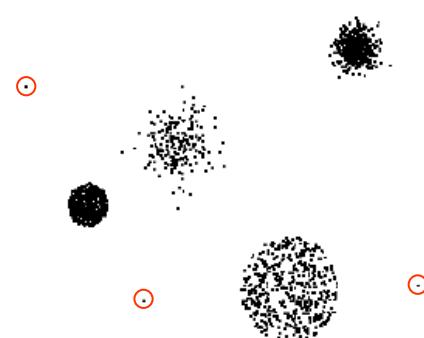
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

27

27

Outliers

- **Outliers** are data objects with characteristics that are considerably different than most of the other data objects in the data set
 - **Case 1:** Outliers are noise that interferes with data analysis
 - **Case 2:** Outliers are the goal of our analysis
 - ◆ Credit card fraud
 - ◆ Intrusion detection
- Causes?



09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

28

28

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 or variables
 - Estimate missing values
 - ◆ Example: time series of temperature
 - ◆ Example: census results
 - Ignore the missing value during analysis

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

29

29

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
- When should duplicate data not be removed?

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

30

30

Similarity and Dissimilarity Measures

- Similarity measure

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

- Dissimilarity measure

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

- Proximity refers to a similarity or dissimilarity

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

31

31

Similarity/Dissimilarity for Simple Attributes

The following table shows the similarity and dissimilarity between two objects, x and y , with respect to a single, simple attribute.

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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

32

32

Euclidean Distance

- Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$

where n is the number of dimensions (attributes) and x_k and y_k are, respectively, the k^{th} attributes (components) or data objects \mathbf{x} and \mathbf{y} .

- Standardization is necessary, if scales differ.

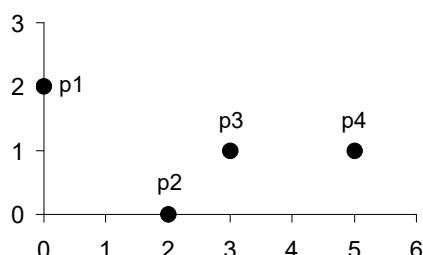
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

33

33

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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

34

34

Minkowski Distance

- Minkowski Distance is a generalization of Euclidean Distance

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^n |x_k - y_k|^r \right)^{1/r}$$

Where r is a parameter, n is the number of dimensions (attributes) and x_k and y_k are, respectively, the k^{th} attributes (components) or data objects \mathbf{x} and \mathbf{y} .

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

35

35

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
 - A common example of this for binary vectors 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.

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

36

36

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

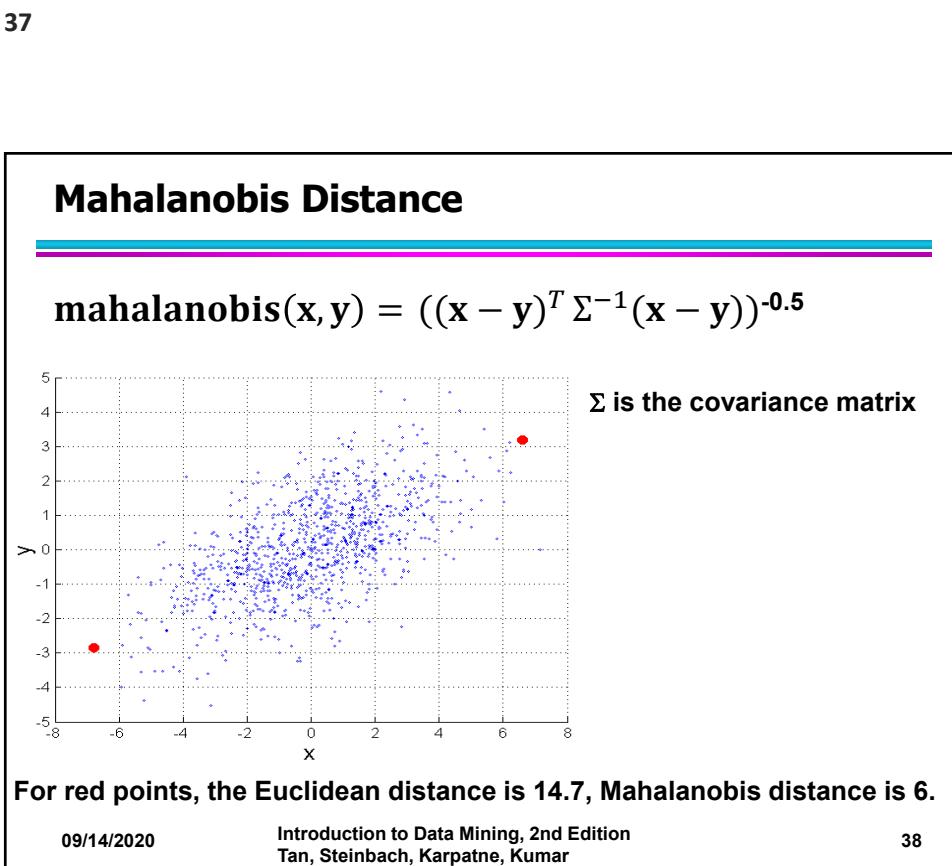
L ∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

37



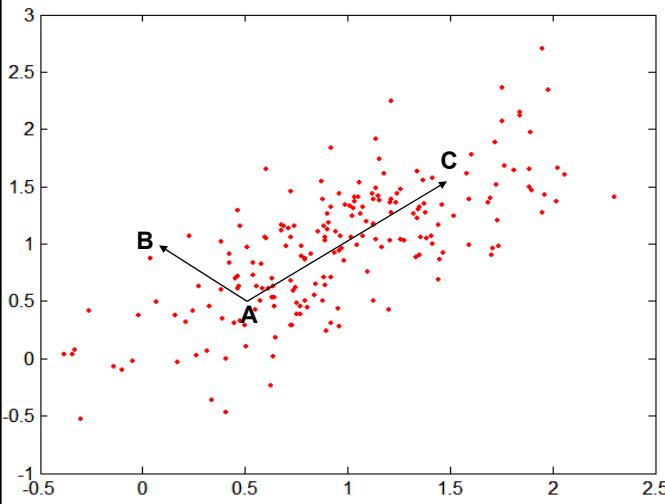
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

38

38

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)

$\text{Mahal}(A,B) = 5$
 $\text{Mahal}(A,C) = 4$

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

39

39

Common Properties of a Distance

- Distances, such as the Euclidean distance, have some well known properties.
 1. $d(x, y) \geq 0$ for all x and y and $d(x, y) = 0$ if and only if $x = y$.
 2. $d(x, y) = d(y, x)$ for all x and y . (Symmetry)
 3. $d(x, z) \leq d(x, y) + d(y, z)$ for all points x, y , and z . (Triangle Inequality)

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

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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

40

40

Common Properties of a Similarity

- Similarities, also have some well known properties.
 1. $s(x, y) = 1$ (or maximum similarity) only if $x = y$.
(does not always hold, e.g., cosine)
 2. $s(x, y) = s(y, x)$ for all x and y . (Symmetry)

where $s(x, y)$ is the similarity between points (data objects), x and y .

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

41

41

Similarity Between Binary Vectors

- Common situation is that objects, x and y , have only binary attributes
- Compute similarities using the following quantities
 - f_{01} = the number of attributes where x was 0 and y was 1
 - f_{10} = the number of attributes where x was 1 and y was 0
 - f_{00} = the number of attributes where x was 0 and y was 0
 - f_{11} = the number of attributes where x was 1 and y was 1
- Simple Matching and Jaccard Coefficients
 - $SMC = \text{number of matches} / \text{number of attributes}$
 $= (f_{11} + f_{00}) / (f_{01} + f_{10} + f_{11} + f_{00})$
 - $J = \text{number of } 11 \text{ matches} / \text{number of non-zero attributes}$
 $= (f_{11}) / (f_{01} + f_{10} + f_{11})$

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

42

42

SMC versus Jaccard: Example

$\mathbf{x} = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$

$\mathbf{y} = 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1$

$f_{01} = 2$ (the number of attributes where \mathbf{x} was 0 and \mathbf{y} was 1)

$f_{10} = 1$ (the number of attributes where \mathbf{x} was 1 and \mathbf{y} was 0)

$f_{00} = 7$ (the number of attributes where \mathbf{x} was 0 and \mathbf{y} was 0)

$f_{11} = 0$ (the number of attributes where \mathbf{x} was 1 and \mathbf{y} was 1)

$$\begin{aligned}\text{SMC} &= (f_{11} + f_{00}) / (f_{01} + f_{10} + f_{11} + f_{00}) \\ &= (0+7) / (2+1+0+7) = 0.7\end{aligned}$$

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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

43

43

Cosine Similarity

- If \mathbf{d}_1 and \mathbf{d}_2 are two document vectors, then

$$\cos(\mathbf{d}_1, \mathbf{d}_2) = \langle \mathbf{d}_1, \mathbf{d}_2 \rangle / \|\mathbf{d}_1\| \|\mathbf{d}_2\|,$$

where $\langle \mathbf{d}_1, \mathbf{d}_2 \rangle$ indicates inner product or vector dot product of vectors, \mathbf{d}_1 and \mathbf{d}_2 , and $\|\mathbf{d}\|$ is the length of vector \mathbf{d} .

- Example:

$\mathbf{d}_1 = 3 \ 2 \ 0 \ 5 \ 0 \ 0 \ 0 \ 2 \ 0 \ 0$

$\mathbf{d}_2 = 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 2$

$$\langle \mathbf{d}_1, \mathbf{d}_2 \rangle = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$$

$$\|\mathbf{d}_1\| = (3^2 + 2^2 + 0^2 + 5^2 + 0^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2)^{0.5} = (42)^{0.5} = 6.481$$

$$\|\mathbf{d}_2\| = (1^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 1^2 + 0^2 + 2^2)^{0.5} = (6)^{0.5} = 2.449$$

$$\cos(\mathbf{d}_1, \mathbf{d}_2) = 0.3150$$

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

44

44

Correlation measures the linear relationship between objects

$$\text{corr}(\mathbf{x}, \mathbf{y}) = \frac{\text{covariance}(\mathbf{x}, \mathbf{y})}{\text{standard_deviation}(\mathbf{x}) * \text{standard_deviation}(\mathbf{y})} = \frac{s_{xy}}{s_x s_y}, \quad (2.11)$$

where we are using the following standard statistical notation and definitions

$$\text{covariance}(\mathbf{x}, \mathbf{y}) = s_{xy} = \frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})(y_k - \bar{y}) \quad (2.12)$$

$$\begin{aligned} \text{standard_deviation}(\mathbf{x}) &= s_x = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (x_k - \bar{x})^2} \\ \text{standard_deviation}(\mathbf{y}) &= s_y = \sqrt{\frac{1}{n-1} \sum_{k=1}^n (y_k - \bar{y})^2} \end{aligned}$$

$$\begin{aligned} \bar{x} &= \frac{1}{n} \sum_{k=1}^n x_k \text{ is the mean of } \mathbf{x} \\ \bar{y} &= \frac{1}{n} \sum_{k=1}^n y_k \text{ is the mean of } \mathbf{y} \end{aligned}$$

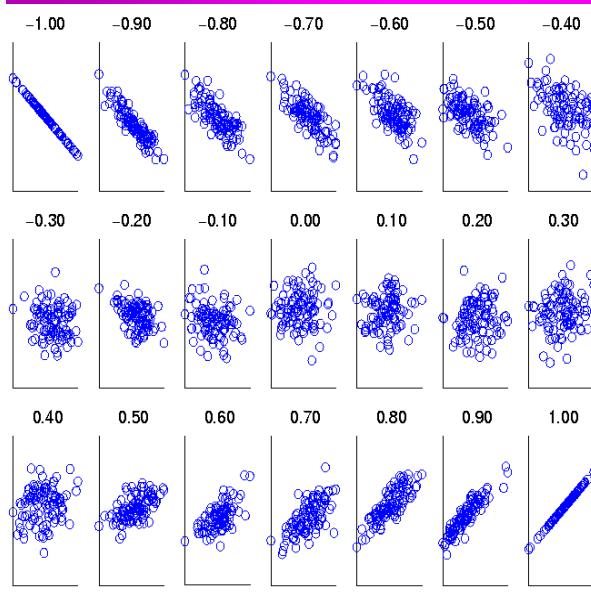
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

45

45

Visually Evaluating Correlation



Scatter plots showing the similarity from -1 to 1.

Tan, Steinbach, Karpatne, Kumar

46

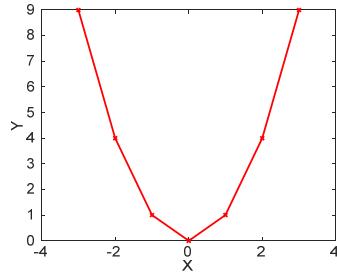
46

Drawback of Correlation

- $\mathbf{x} = (-3, -2, -1, 0, 1, 2, 3)$

- $\mathbf{y} = (9, 4, 1, 0, 1, 4, 9)$

$$y_i = x_i^2$$



- $\text{mean}(\mathbf{x}) = 0, \text{mean}(\mathbf{y}) = 4$

- $\text{std}(\mathbf{x}) = 2.16, \text{std}(\mathbf{y}) = 3.74$

- $\text{corr} = (-3)(5) + (-2)(0) + (-1)(-3) + (0)(-4) + (1)(-3) + (2)(0) + 3(5) / (6 * 2.16 * 3.74)$
= 0

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

47

47

Correlation vs Cosine vs Euclidean Distance

- Compare the three proximity measures according to their behavior under variable transformation
 - scaling: multiplication by a value
 - translation: adding a constant

Property	Cosine	Correlation	Euclidean Distance
Invariant to scaling (multiplication)	Yes	Yes	No
Invariant to translation (addition)	No	Yes	No

- Consider the example
 - $\mathbf{x} = (1, 2, 4, 3, 0, 0, 0)$, $\mathbf{y} = (1, 2, 3, 4, 0, 0, 0)$
 - $\mathbf{y}_s = \mathbf{y} * 2$ (scaled version of \mathbf{y}), $\mathbf{y}_t = \mathbf{y} + 5$ (translated version)

Measure	(\mathbf{x}, \mathbf{y})	$(\mathbf{x}, \mathbf{y}_s)$	$(\mathbf{x}, \mathbf{y}_t)$
Cosine	0.9667	0.9667	0.7940
Correlation	0.9429	0.9429	0.9429
Euclidean Distance	1.4142	5.8310	14.2127

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

48

48

Correlation vs cosine vs Euclidean distance

- Choice of the right proximity measure depends on the domain
- What is the correct choice of proximity measure for the following situations?
 - Comparing documents using the frequencies of words
 - ◆ Documents are considered similar if the word frequencies are similar
 - Comparing the temperature in Celsius of two locations
 - ◆ Two locations are considered similar if the temperatures are similar in magnitude
 - Comparing two time series of temperature measured in Celsius
 - ◆ Two time series are considered similar if their “shape” is similar, i.e., they vary in the same way over time, achieving minimums and maximums at similar times, etc.

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

49

49

Comparison of Proximity Measures

- Domain of application
 - Similarity measures tend to be specific to the type of attribute and data
 - Record data, images, graphs, sequences, 3D-protein structure, etc. tend to have different measures
- However, one can talk about various properties that you would like a proximity measure to have
 - Symmetry is a common one
 - Tolerance to noise and outliers is another
 - Ability to find more types of patterns?
 - Many others possible
- The measure must be applicable to the data and produce results that agree with domain knowledge

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

50

50

Information Based Measures

- Information theory is a well-developed and fundamental discipline with broad applications
- Some similarity measures are based on information theory
 - Mutual information in various versions
 - Maximal Information Coefficient (MIC) and related measures
 - General and can handle non-linear relationships
 - Can be complicated and time intensive to compute

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

51

51

Information and Probability

- Information relates to possible outcomes of an event
 - transmission of a message, flip of a coin, or measurement of a piece of data
- The more certain an outcome, the less information that it contains and vice-versa
 - For example, if a coin has two heads, then an outcome of heads provides no information
 - More quantitatively, the information is related to the probability of an outcome
 - ◆ The smaller the probability of an outcome, the more information it provides and vice-versa
 - Entropy is the commonly used measure



09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

52

52

Entropy

- For
 - a variable (event), X ,
 - with n possible values (outcomes), $x_1, x_2 \dots, x_n$
 - each outcome having probability, $p_1, p_2 \dots, p_n$
 - the entropy of X , $H(X)$, is given by
$$H(X) = - \sum_{i=1}^n p_i \log_2 p_i$$
- Entropy is between 0 and $\log_2 n$ and is measured in bits
 - Thus, entropy is a measure of how many bits it takes to represent an observation of X on average

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

53

53

Entropy Examples

- For a coin with probability p of heads and probability $q = 1 - p$ of tails

$$H = -p \log_2 p - q \log_2 q$$

- For $p = 0.5, q = 0.5$ (fair coin) $H = 1$
- For $p = 1$ or $q = 1, H = 0$

- What is the entropy of a fair four-sided die?

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

54

54

Entropy for Sample Data: Example

Hair Color	Count	p	$-p \log_2 p$
Black	75	0.75	0.3113
Brown	15	0.15	0.4105
Blond	5	0.05	0.2161
Red	0	0.00	0
Other	5	0.05	0.2161
Total	100	1.0	1.1540

Maximum entropy is $\log_2 5 = 2.3219$

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

55

55

Entropy for Sample Data

- Suppose we have

- a number of observations (m) of some attribute, X ,
e.g., the hair color of students in the class,
- where there are n different possible values
- And the number of observation in the i^{th} category is m_i
- Then, for this sample

$$H(X) = - \sum_{i=1}^n \frac{m_i}{m} \log_2 \frac{m_i}{m}$$

- For continuous data, the calculation is harder

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

56

56

Mutual Information

- Information one variable provides about another

Formally, $I(X, Y) = H(X) + H(Y) - H(X, Y)$, where

$H(X, Y)$ is the joint entropy of X and Y ,

$$H(X, Y) = - \sum_i \sum_j p_{ij} \log_2 p_{ij}$$

Where p_{ij} is the probability that the i^{th} value of X and the j^{th} value of Y occur together

- For discrete variables, this is easy to compute
- Maximum mutual information for discrete variables is $\log_2(\min(n_X, n_Y))$, where n_X (n_Y) is the number of values of X (Y)

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

57

57

Mutual Information Example

Student Status	Count	p	$-p \log_2 p$
Undergrad	45	0.45	0.5184
Grad	55	0.55	0.4744
Total	100	1.00	0.9928

Grade	Count	p	$-p \log_2 p$
A	35	0.35	0.5301
B	50	0.50	0.5000
C	15	0.15	0.4105
Total	100	1.00	1.4406

Student Status	Grade	Count	p	$-p \log_2 p$
Undergrad	A	5	0.05	0.2161
Undergrad	B	30	0.30	0.5211
Undergrad	C	10	0.10	0.3322
Grad	A	30	0.30	0.5211
Grad	B	20	0.20	0.4644
Grad	C	5	0.05	0.2161
Total		100	1.00	2.2710

Mutual information of Student Status and Grade = $0.9928 + 1.4406 - 2.2710 = 0.1624$

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

58

58

Maximal Information Coefficient

- Reshef, David N., Yakir A. Reshef, Hilary K. Finucane, Sharon R. Grossman, Gilean McVean, Peter J. Turnbaugh, Eric S. Lander, Michael Mitzenmacher, and Pardis C. Sabeti. "Detecting novel associations in large data sets." *science* 334, no. 6062 (2011): 1518-1524.
- Applies mutual information to two continuous variables
- Consider the possible binnings of the variables into discrete categories
 - $n_X \times n_Y \leq N^{0.6}$ where
 - ◆ n_X is the number of values of X
 - ◆ n_Y is the number of values of Y
 - ◆ N is the number of samples (observations, data objects)
- Compute the mutual information
 - Normalized by $\log_2(\min(n_X, n_Y))$
- Take the highest value

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

59

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, $s_k(\mathbf{x}, \mathbf{y})$, in the range $[0, 1]$.
 - 2: Define an indicator variable, δ_k , for the k^{th} attribute as follows:
$$\delta_k = 0 \text{ if the } k^{\text{th}} \text{ attribute is an asymmetric attribute and both objects have a value of 0, or if one of the objects has a missing value for the } k^{\text{th}} \text{ attribute}$$

$$\delta_k = 1 \text{ otherwise}$$
 3. Compute $\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^n \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^n \delta_k}$

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

60

60

Using Weights to Combine Similarities

- May not want to treat all attributes the same.
 - Use non-negative weights ω_k

$$\text{similarity}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{k=1}^n \omega_k \delta_k s_k(\mathbf{x}, \mathbf{y})}{\sum_{k=1}^n \omega_k \delta_k}$$

- Can also define a weighted form of distance

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^n w_k |x_k - y_k|^r \right)^{1/r}$$

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

61

61

Data Preprocessing

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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

62

62

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.
 - ◆ Days aggregated into weeks, months, or years
 - More “stable” data
 - ◆ Aggregated data tends to have less variability

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

63

63

Example: Precipitation in Australia

- This example is based on precipitation in Australia from the period 1982 to 1993.

The next slide shows

 - A histogram for the standard deviation of average monthly precipitation for 3,030 0.5° by 0.5° grid cells in Australia, and
 - A histogram for the standard deviation of the average yearly precipitation for the same locations.
- The average yearly precipitation has less variability than the average monthly precipitation.
- All precipitation measurements (and their standard deviations) are in centimeters.

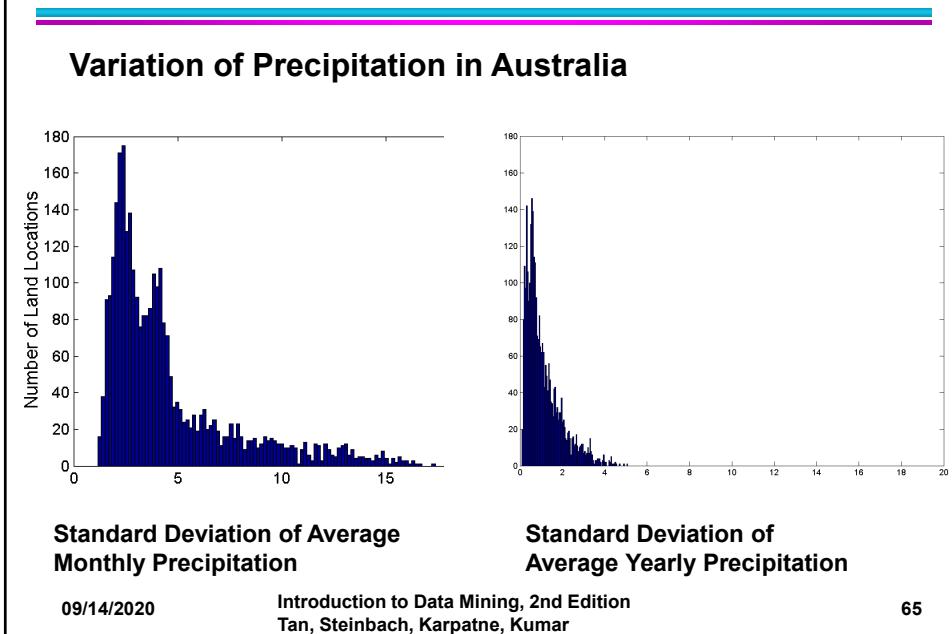
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

64

64

Example: Precipitation in Australia ...



65

Sampling

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

66

Sampling ...

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

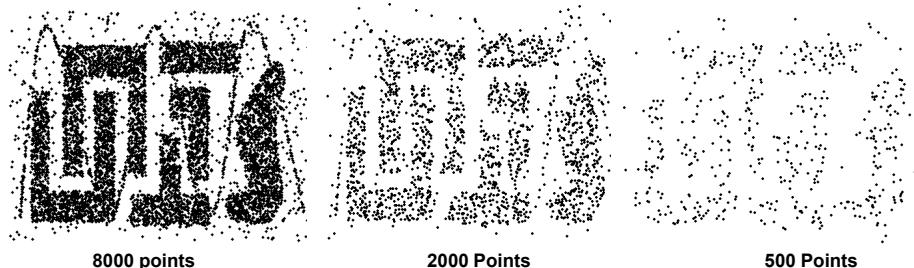
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

67

67

Sample Size



09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

68

68

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

09/14/2020

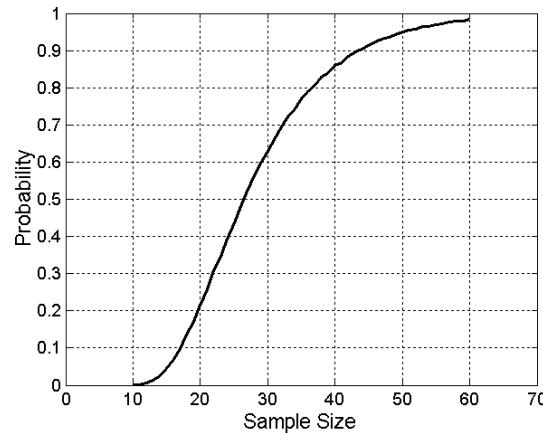
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

69

69

Sample Size

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



09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

70

70

Discretization

- **Discretization** is the process of converting a continuous attribute into an ordinal attribute
 - A potentially infinite number of values are mapped into a small number of categories
 - Discretization is used in both unsupervised and supervised settings

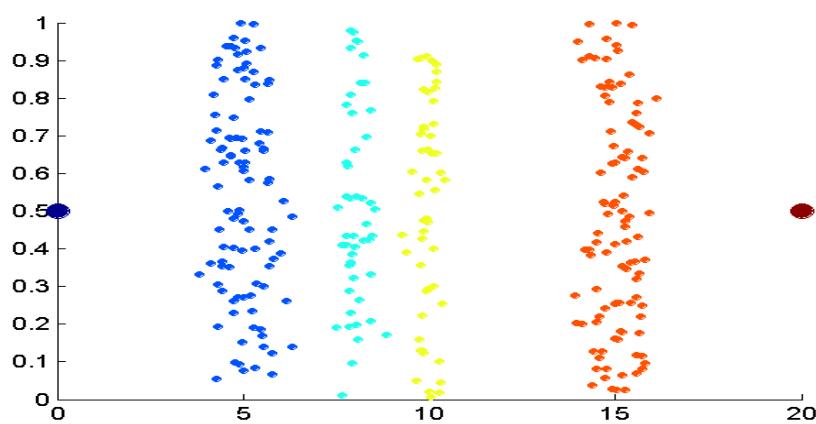
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

71

71

Unsupervised Discretization



Data consists of four groups of points and two outliers. Data is one-dimensional, but a random y component is added to reduce overlap.

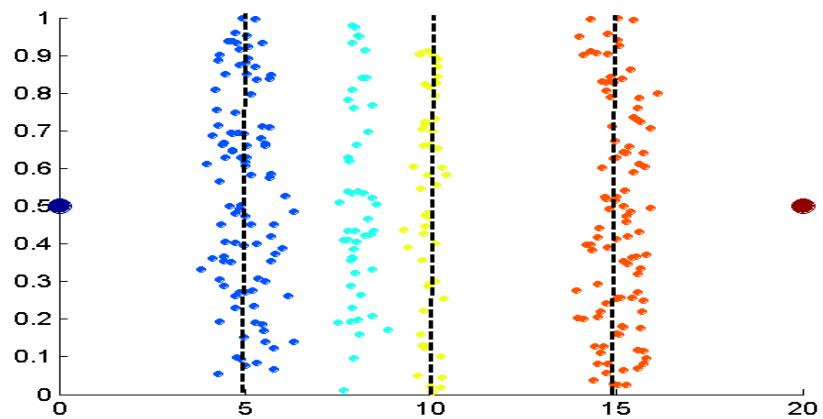
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

72

72

Unsupervised Discretization



Equal interval width approach used to obtain 4 values.

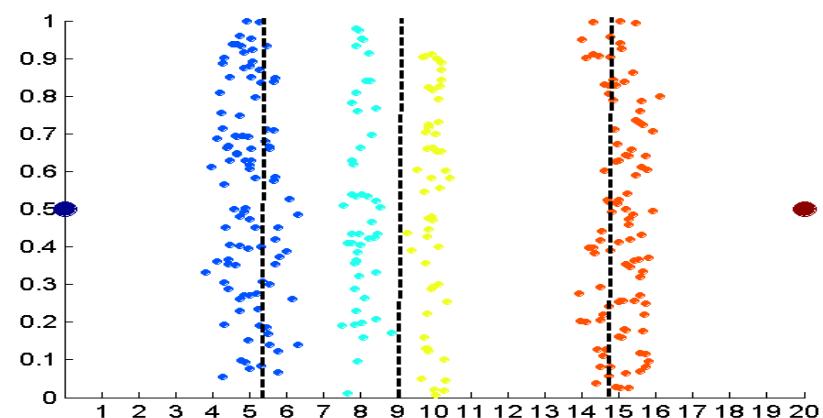
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

73

73

Unsupervised Discretization



Equal frequency approach used to obtain 4 values.

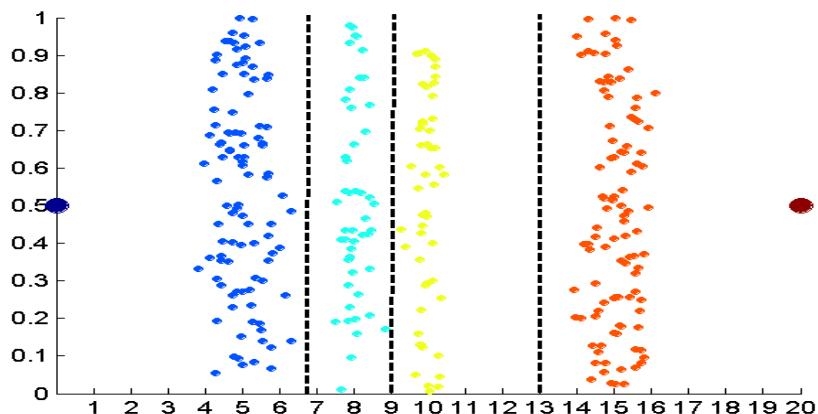
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

74

74

Unsupervised Discretization



K-means approach to obtain 4 values.

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

75

Discretization in Supervised Settings

- Many classification algorithms work best if both the independent and dependent variables have only a few values
- We give an illustration of the usefulness of discretization using the Iris data set

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

76

76

Iris Sample Data Set

- 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
 - ◆ Versicolour
 - ◆ Virginica
 - 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.

09/14/2020

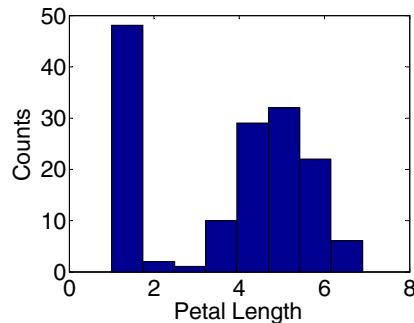
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

77

77

Discretization: Iris Example ...

- How can we tell what the best discretization is?
 - **Unsupervised discretization:** find breaks in the data values
 - ◆ Example:
Petal Length



- **Supervised discretization:** Use class labels to find breaks

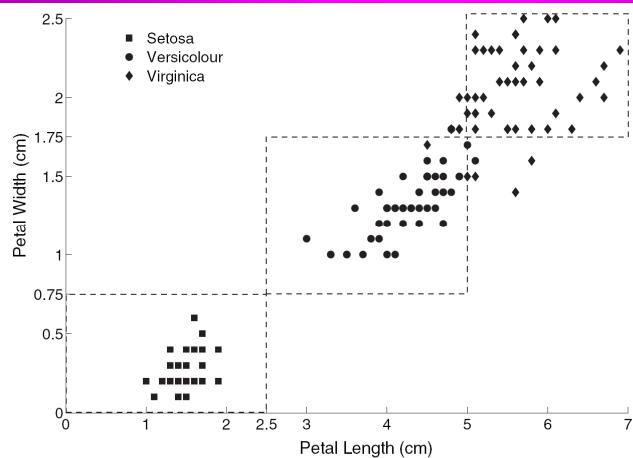
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

78

78

Discretization: Iris Example



Petal width low or petal length low implies Setosa.

Petal width medium or petal length medium implies Versicolour.

Petal width high or petal length high implies Virginica.

09/14/2020

Introduction to Data Mining, 2nd Edition

Tan, Steinbach, Karpatne, Kumar

79

79

Binarization

- Binarization maps a continuous or categorical attribute into one or more binary variables
- Typically used for association analysis
- Often convert a continuous attribute to a categorical attribute and then convert a categorical attribute to a set of binary attributes
 - Association analysis needs asymmetric binary attributes
 - Examples: eye color and height measured as {low, medium, high}

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

80

80

Attribute Transformation

- An **attribute transform** is 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 , $|x|$
 - **Normalization**
 - ◆ Refers to various techniques to adjust to differences among attributes in terms of frequency of occurrence, mean, variance, range
 - ◆ Take out unwanted, common signal, e.g., seasonality
 - In statistics, **standardization** refers to subtracting off the means and dividing by the standard deviation

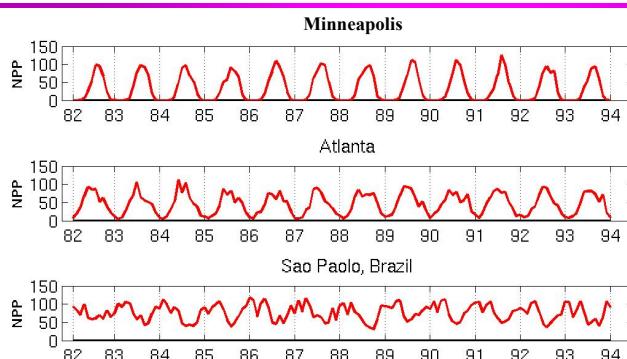
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

81

81

Example: Sample Time Series of Plant Growth



Net Primary Production (NPP) is a measure of plant growth used by ecosystem scientists.

Correlations between time series

	Minneapolis	Atlanta	Sao Paolo
Minneapolis	1.0000	0.7591	-0.7581
Atlanta	0.7591	1.0000	-0.5739
Sao Paolo	-0.7581	-0.5739	1.0000

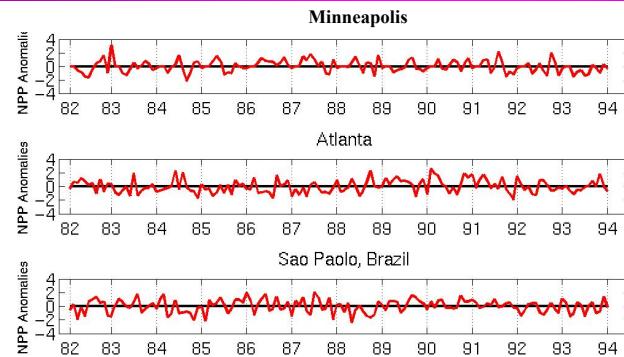
09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

82

82

Seasonality Accounts for Much Correlation



Normalized using monthly Z Score:
Subtract off monthly mean and divide by monthly standard deviation

Correlations between time series

	Minneapolis	Atlanta	Sao Paolo
Minneapolis	1.0000	0.0492	0.0906
Atlanta	0.0492	1.0000	-0.0154
Sao Paolo	0.0906	-0.0154	1.0000

09/14/2020

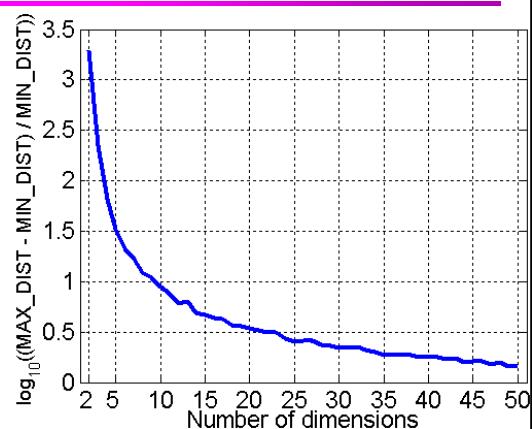
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

83

83

Curse of Dimensionality

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



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

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

84

84

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

- Principal Components Analysis (PCA)
- Singular Value Decomposition
- Others: supervised and non-linear techniques

09/14/2020

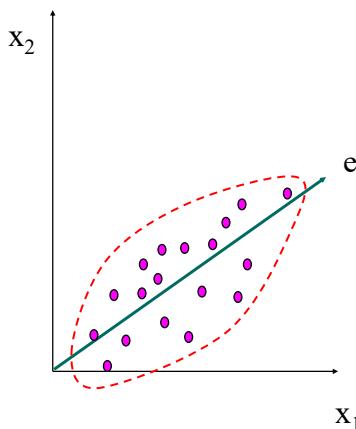
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

85

85

Dimensionality Reduction: PCA

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



09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

86

86

Dimensionality Reduction: PCA

256



09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

87

87

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
- Many techniques developed, especially for classification

09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

88

88

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
 - ◆ Example: extracting edges from images
 - Feature construction
 - ◆ Example: dividing mass by volume to get density
 - Mapping data to new space
 - ◆ Example: Fourier and wavelet analysis

09/14/2020

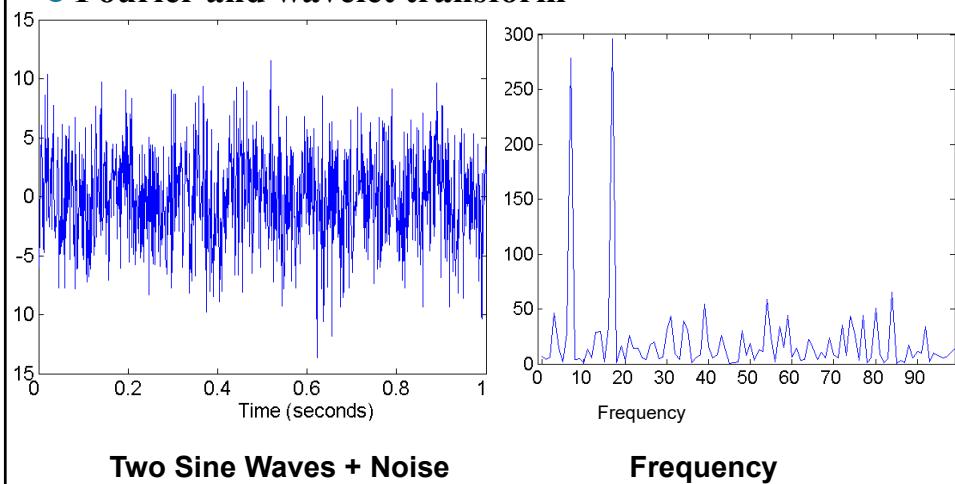
Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

89

89

Mapping Data to a New Space

- Fourier and wavelet transform



09/14/2020

Introduction to Data Mining, 2nd Edition
Tan, Steinbach, Karpatne, Kumar

90

90

Data Mining Classification: Basic Concepts and Techniques

Lecture Notes for Chapter 3

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

09/21/2020

Introduction to Data Mining, 2nd Edition

1

Classification: Definition

- Given a collection of records (training set)
 - Each record is characterized by a tuple (x,y) , where x is the attribute set and y is the class label
 - ◆ x : attribute, predictor, independent variable, input
 - ◆ y : class, response, dependent variable, output
- Task:
 - Learn a model that maps each attribute set x into one of the predefined class labels y

09/21/2020

Introduction to Data Mining, 2nd Edition

2

2

Examples of Classification Task

Task	Attribute set, x	Class label, y
Categorizing email messages	Features extracted from email message header and content	spam or non-spam
Identifying tumor cells	Features extracted from x-rays or MRI scans	malignant or benign cells
Cataloging galaxies	Features extracted from telescope images	Elliptical, spiral, or irregular-shaped galaxies

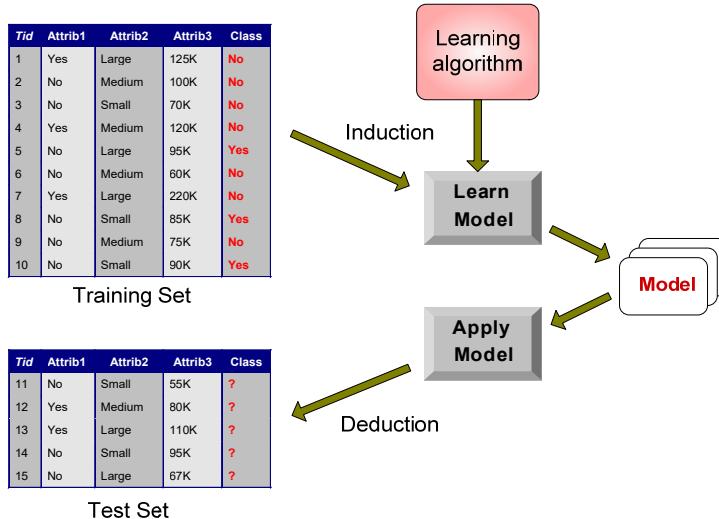
09/21/2020

Introduction to Data Mining, 2nd Edition

3

3

General Approach for Building Classification Model



09/21/2020

Introduction to Data Mining, 2nd Edition

4

4

Classification Techniques

● Base Classifiers

- Decision Tree based Methods
- Rule-based Methods
- Nearest-neighbor
- Neural Networks, Deep Neural Nets
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

● Ensemble Classifiers

- Boosting, Bagging, Random Forests

09/21/2020

Introduction to Data Mining, 2nd Edition

5

Example of a Decision Tree

Training Data

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

Model: Decision Tree

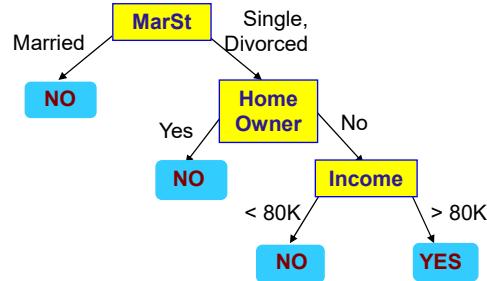
```
graph TD; Root[Home Owner] -- Yes --> Node1[NO]; Root -- No --> MarSt[MarSt]; MarSt -- Single, Divorced --> Node2[Income]; MarSt -- Married --> Node3[NO]; Node2 -- < 80K --> Node4[NO]; Node2 -- > 80K --> Node5[YES];
```

The diagram illustrates a decision tree model. The root node is "Home Owner". A red arrow points from the "Training Data" table to this node. The "Home Owner" node splits into "Yes" and "No". The "Yes" branch leads to a blue box labeled "NO". The "No" branch leads to a yellow box labeled "MarSt". From "MarSt", two paths emerge: "Single, Divorced" leading to a yellow box labeled "Income", and "Married" leading to a blue box labeled "NO". The "Income" node further splits into " $< 80K$ " leading to a blue box labeled "NO", and " $> 80K$ " leading to a blue box labeled "YES".

6

Another Example of Decision Tree

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



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

09/21/2020

Introduction to Data Mining, 2nd Edition

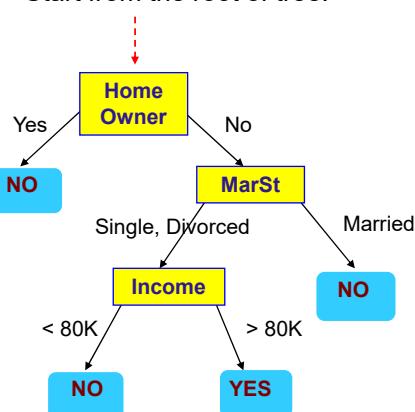
7

Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Start from the root of tree.



09/21/2020

Introduction to Data Mining, 2nd Edition

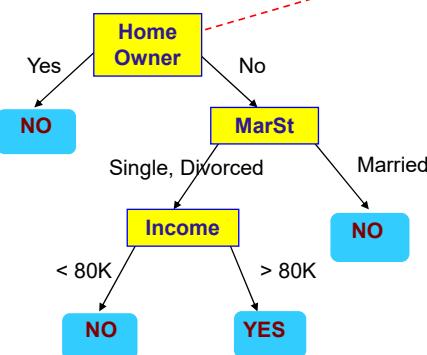
8

8

Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



09/21/2020

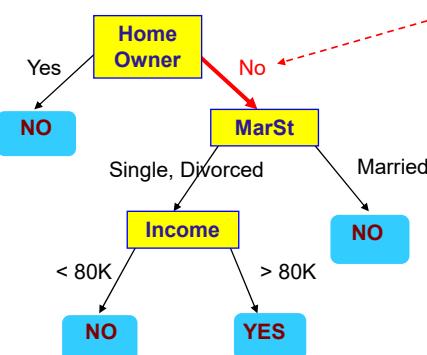
Introduction to Data Mining, 2nd Edition

9

Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



09/21/2020

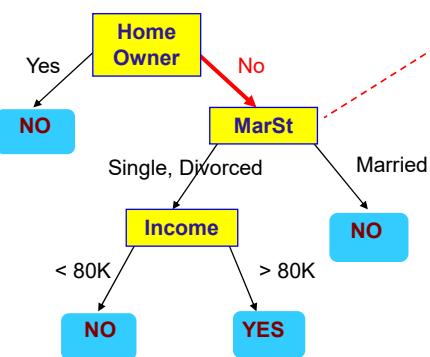
Introduction to Data Mining, 2nd Edition

10

Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



09/21/2020

Introduction to Data Mining, 2nd Edition

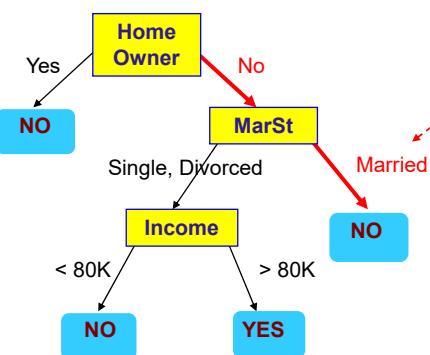
11

11

Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



09/21/2020

Introduction to Data Mining, 2nd Edition

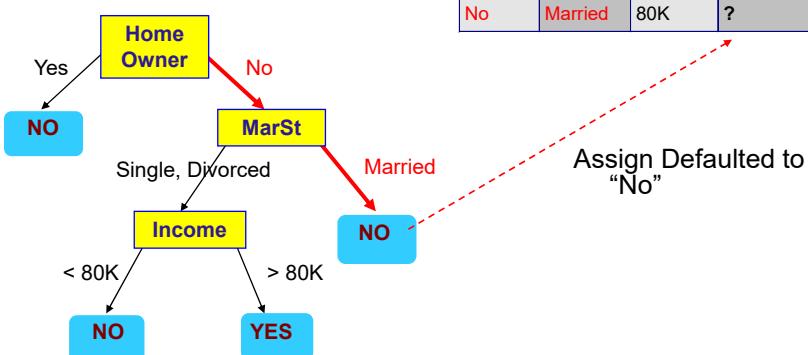
12

12

Apply Model to Test Data

Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



09/21/2020

Introduction to Data Mining, 2nd Edition

13

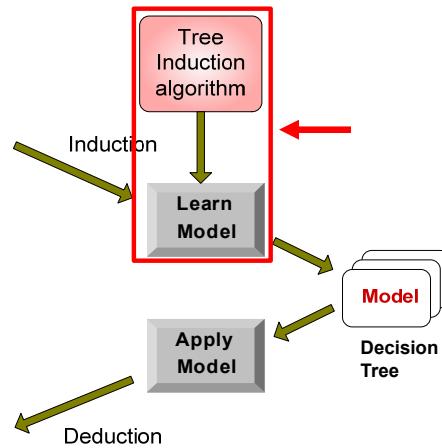
Decision Tree Classification Task

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

Training Set

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

Test Set



09/21/2020

Introduction to Data Mining, 2nd Edition

14

14

Decision Tree Induction

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

09/21/2020

Introduction to Data Mining, 2nd Edition

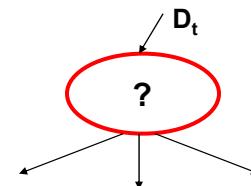
15

15

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 to the same class y_t , then t is a leaf node labeled as y_t
 - If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets. Recursively apply the procedure to each subset.

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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



09/21/2020

Introduction to Data Mining, 2nd Edition

16

16

Hunt's Algorithm

Defaulted = No

(7,3)

(a)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

09/21/2020

Introduction to Data Mining, 2nd Edition

17

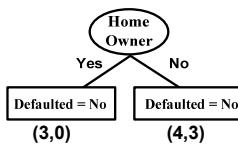
17

Hunt's Algorithm

Defaulted = No

(7,3)

(a)



(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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

09/21/2020

Introduction to Data Mining, 2nd Edition

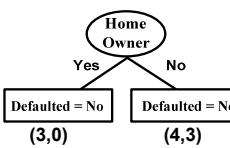
18

18

Hunt's Algorithm

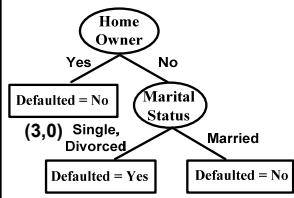
Defaulted = No
(7,3)

(a)



(b)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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



(c)

09/21/2020

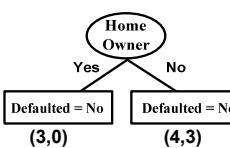
Introduction to Data Mining, 2nd Edition

19

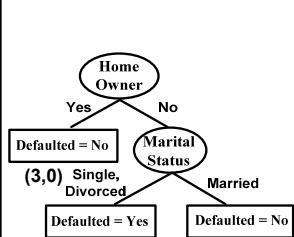
Hunt's Algorithm

Defaulted = No
(7,3)

(a)

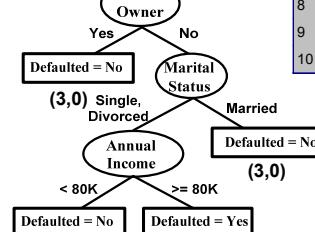


(b)



(c)

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
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



(d)

09/21/2020

Introduction to Data Mining, 2nd Edition

20

20

Design Issues of Decision Tree Induction

- How should training records be split?
 - Method for specifying test condition
 - ◆ depending on attribute types
 - Measure for evaluating the goodness of a test condition
- How should the splitting procedure stop?
 - Stop splitting if all the records belong to the same class or have identical attribute values
 - Early termination

09/21/2020

Introduction to Data Mining, 2nd Edition

21

21

Methods for Expressing Test Conditions

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

09/21/2020

Introduction to Data Mining, 2nd Edition

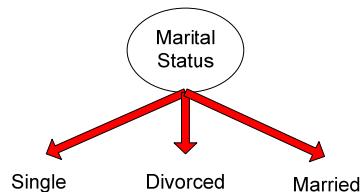
22

22

Test Condition for Nominal Attributes

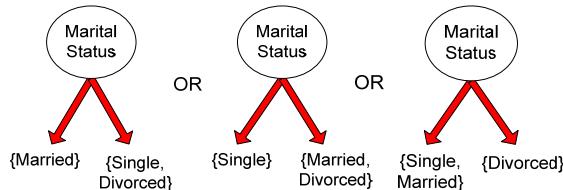
- Multi-way split:

- Use as many partitions as distinct values.



- Binary split:

- Divides values into two subsets



09/21/2020

Introduction to Data Mining, 2nd Edition

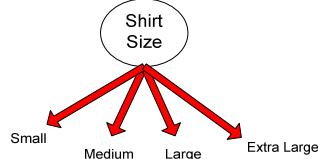
23

23

Test Condition for Ordinal Attributes

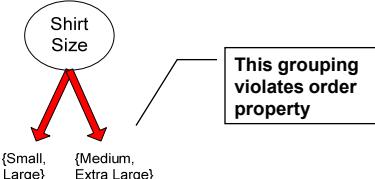
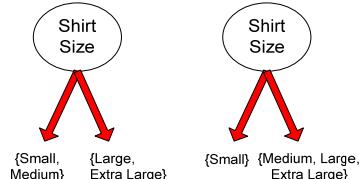
- Multi-way split:

- Use as many partitions as distinct values



- Binary split:

- Divides values into two subsets
 - Preserve order property among attribute values



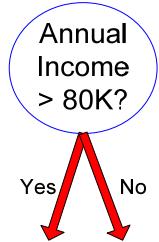
09/21/2020

Introduction to Data Mining, 2nd Edition

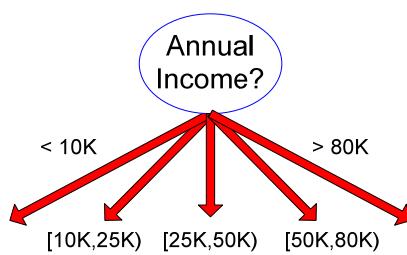
24

24

Test Condition for Continuous Attributes



(i) Binary split



(ii) Multi-way split

09/21/2020

Introduction to Data Mining, 2nd Edition

25

25

Splitting Based on Continuous Attributes

- Different ways of handling

- **Discretization** to form an ordinal categorical attribute

Ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.

- ◆ Static – discretize once at the beginning
 - ◆ Dynamic – repeat at each node

- **Binary Decision:** $(A < v)$ or $(A \geq v)$

- ◆ consider all possible splits and finds the best cut
 - ◆ can be more compute intensive

09/21/2020

Introduction to Data Mining, 2nd Edition

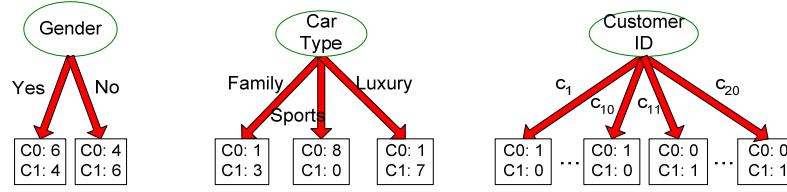
26

26

How to determine the Best Split

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

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



Which test condition is the best?

09/21/2020

Introduction to Data Mining, 2nd Edition

27

27

How to determine the Best Split

- Greedy approach:
 - Nodes with **purer** class distribution are preferred
- Need a measure of node impurity:

C0: 5
C1: 5

High degree of impurity

C0: 9
C1: 1

Low degree of impurity

09/21/2020

Introduction to Data Mining, 2nd Edition

28

28

Measures of Node Impurity

- Gini Index

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Where $p_i(t)$ is the frequency of class i at node t , and c is the total number of classes

- Entropy

$$Entropy = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

- Misclassification error

$$Classification\ error = 1 - \max[p_i(t)]$$

09/21/2020

Introduction to Data Mining, 2nd Edition

29

29

Finding the Best Split

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

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

or equivalently, lowest impurity measure after splitting (M)

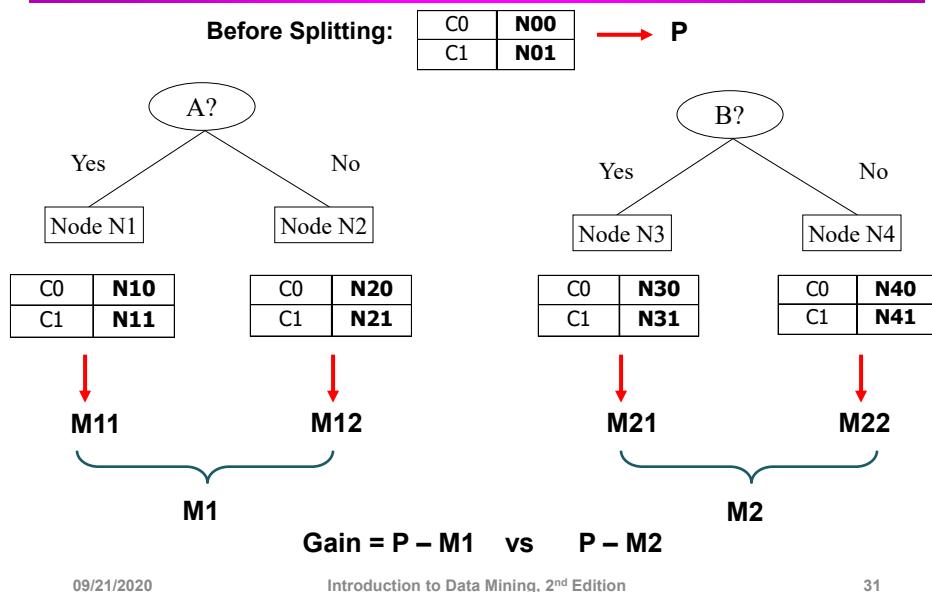
09/21/2020

Introduction to Data Mining, 2nd Edition

30

30

Finding the Best Split



31

Measure of Impurity: GINI

- Gini Index for a given node t

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

Where $p_i(t)$ is the frequency of class i at node t , and c is the total number of classes

- Maximum of $1 - 1/c$ when records are equally distributed among all classes, implying the least beneficial situation for classification
- Minimum of 0 when all records belong to one class, implying the most beneficial situation for classification
- Gini index is used in decision tree algorithms such as CART, SLIQ, SPRINT

32

Measure of Impurity: GINI

- Gini Index for a given node t :

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

- For 2-class problem (p, 1 – p):

- ◆ $GINI = 1 - p^2 - (1 - p)^2 = 2p(1-p)$

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	

09/21/2020

Introduction to Data Mining, 2nd Edition

33

33

Computing Gini Index of a Single Node

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

C1	0
C2	6

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

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

C1	1
C2	5

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

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

C1	2
C2	4

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

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

09/21/2020

Introduction to Data Mining, 2nd Edition

34

34

Computing Gini Index for a Collection of Nodes

- When a node p is split into k partitions (children)

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

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

09/21/2020

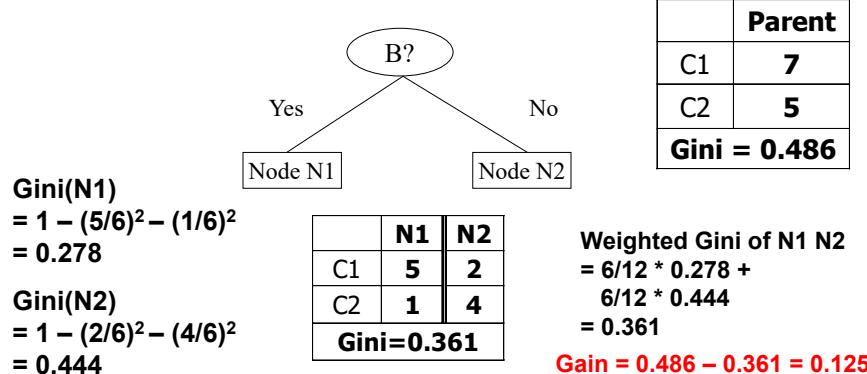
Introduction to Data Mining, 2nd Edition

35

35

Binary Attributes: Computing GINI Index

- Splits into two partitions (child nodes)
- Effect of Weighing partitions:
 - Larger and purer partitions are sought



09/21/2020

Introduction to Data Mining, 2nd Edition

36

36

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	8	1
C2	3	0	7
Gini	0.163		

Two-way split
(find best partition of values)

CarType			
	{Sports, Luxury}	{Family}	
C1	9	1	
C2	7	3	
Gini	0.468		

CarType			
	{Sports}	{Family, Luxury}	
C1	8	2	
C2	0	10	
Gini	0.167		

Which of these is the best?

09/21/2020

Introduction to Data Mining, 2nd Edition

37

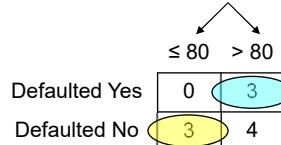
37

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, $A \leq v$ and $A > v$
- 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.

ID	Home Owner	Marital Status	Annual Income	Defaulted
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

Annual Income ?



09/21/2020

Introduction to Data Mining, 2nd Edition

38

38

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
Annual Income										
Sorted Values	60	70	75	85	90	95	100	120	125	220

09/21/2020

Introduction to Data Mining, 2nd Edition

39

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	
Annual Income											
Sorted Values	60	70	75	85	90	95	100	120	125	220	
Split Positions	55	65	72	80	87	92	97	110	122	172	230

09/21/2020

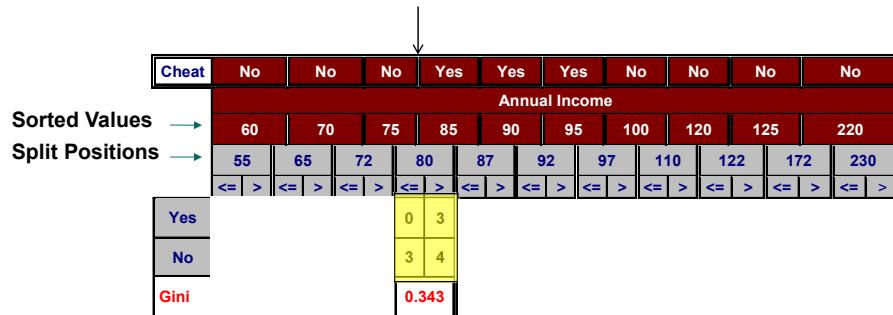
Introduction to Data Mining, 2nd Edition

40

40

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



09/21/2020

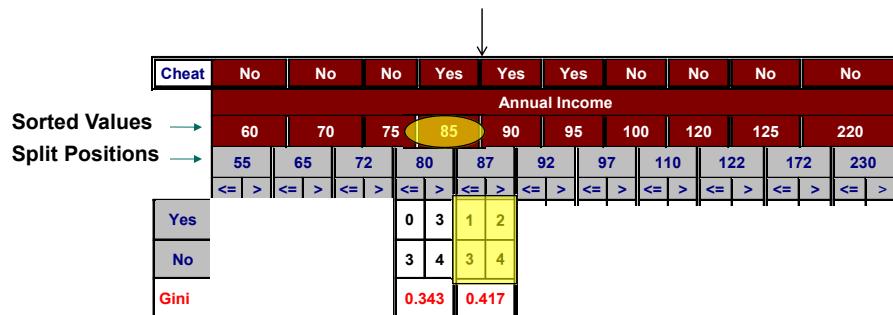
Introduction to Data Mining, 2nd Edition

41

41

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



09/21/2020

Introduction to Data Mining, 2nd Edition

42

42

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

09/21/2020

Introduction to Data Mining, 2nd Edition

43

43

Measure of Impurity: Entropy

- Entropy at a given node t

$$\text{Entropy} = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

Where $p_i(t)$ is the frequency of class i at node t , and c is the total number of classes

- Maximum of $\log_2 c$ when records are equally distributed among all classes, implying the least beneficial situation for classification
- Minimum of 0 when all records belong to one class, implying most beneficial situation for classification

- Entropy based computations are quite similar to the GINI index computations

09/21/2020

Introduction to Data Mining, 2nd Edition

44

44

Computing Entropy of a Single Node

$$Entropy = - \sum_{i=0}^{c-1} p_i(t) \log_2 p_i(t)$$

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

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

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

09/21/2020

Introduction to Data Mining, 2nd Edition

45

45

Computing Information Gain After Splitting

- Information Gain:

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

Parent Node, p is split into k partitions (children)

n_i is number of records in child node i

- Choose the split that achieves most reduction (maximizes GAIN)
- Used in the ID3 and C4.5 decision tree algorithms
- Information gain is the mutual information between the class variable and the splitting variable

09/21/2020

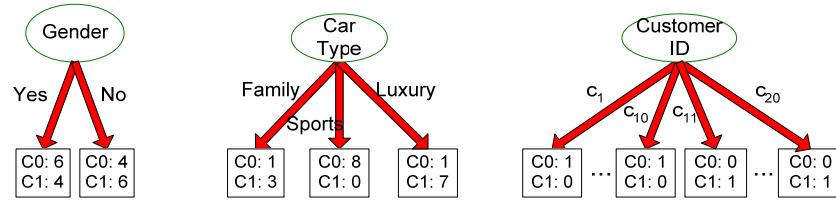
Introduction to Data Mining, 2nd Edition

46

46

Problem with large number of partitions

- Node impurity measures tend to prefer splits that result in large number of partitions, each being small but pure



- Customer ID has highest information gain because entropy for all the children is zero

09/21/2020

Introduction to Data Mining, 2nd Edition

47

Gain Ratio

- Gain Ratio:

$$\text{Gain Ratio} = \frac{\text{Gain}_{\text{split}}}{\text{Split Info}} \quad \text{Split Info} = - \sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

Parent Node, p is split into k partitions (children)
 n_i is number of records in child node i

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

09/21/2020

Introduction to Data Mining, 2nd Edition

48

48

Gain Ratio

- Gain Ratio:

$$Gain\ Ratio = \frac{Gain_{split}}{Split\ Info} \quad Split\ Info = \sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

Parent Node, p is split into k partitions (children)

n_i is number of records in child node i

CarType			
	Family	Sports	Luxury
C1	1	8	1
C2	3	0	7
Gini	0.163		

SplitINFO = 1.52

CarType			
	{Sports, Luxury}	{Family}	
C1	9	1	
C2	7	3	
Gini	0.468		

SplitINFO = 0.72

CarType			
	{Sports}	{Family, Luxury}	
C1	8	2	
C2	0	10	
Gini	0.167		

SplitINFO = 0.97

09/21/2020

Introduction to Data Mining, 2nd Edition

49

Measure of Impurity: Classification Error

- Classification error at a node t

$$Error(t) = 1 - \max_i [p_i(t)]$$

- Maximum of $1 - 1/c$ when records are equally distributed among all classes, implying the least interesting situation
- Minimum of 0 when all records belong to one class, implying the most interesting situation

09/21/2020

Introduction to Data Mining, 2nd Edition

50

50

Computing Error of a Single Node

$$Error(t) = 1 - \max_i[p_i(t)]$$

C1	0
C2	6

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

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

C1	1
C2	5

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

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

C1	2
C2	4

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

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

09/21/2020

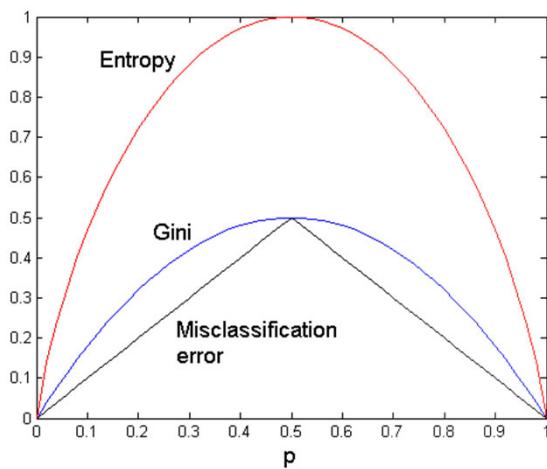
Introduction to Data Mining, 2nd Edition

51

51

Comparison among Impurity Measures

For a 2-class problem:



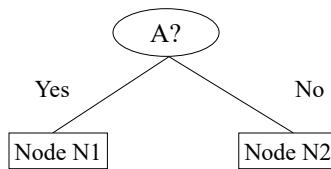
09/21/2020

Introduction to Data Mining, 2nd Edition

52

52

Misclassification Error vs Gini Index



	Parent
C1	7
C2	3
Gini = 0.42	

$$\begin{aligned} \text{Gini}(N1) &= 1 - (3/3)^2 - (0/3)^2 \\ &= 0 \end{aligned}$$

$$\begin{aligned} \text{Gini}(N2) &= 1 - (4/7)^2 - (3/7)^2 \\ &= 0.489 \end{aligned}$$

	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

$$\begin{aligned} \text{Gini(Children)} &= 3/10 * 0 \\ &+ 7/10 * 0.489 \\ &= 0.342 \end{aligned}$$

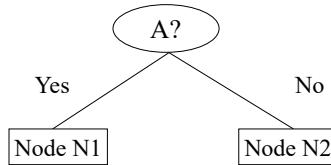
Gini improves but error remains the same!!

09/21/2020

Introduction to Data Mining, 2nd Edition

53

Misclassification Error vs Gini Index



	Parent
C1	7
C2	3
Gini = 0.42	

	N1	N2
C1	3	4
C2	0	3
Gini=0.342		

	N1	N2
C1	3	4
C2	1	2
Gini=0.416		

Misclassification error for all three cases = 0.3 !

09/21/2020

Introduction to Data Mining, 2nd Edition

54

54

Decision Tree Based Classification

● Advantages:

- Relatively inexpensive to construct
- Extremely fast at classifying unknown records
- Easy to interpret for small-sized trees
- Robust to noise (especially when methods to avoid overfitting are employed)
- Can easily handle redundant or irrelevant attributes (unless the attributes are interacting)

● Disadvantages:

- Due to the greedy nature of splitting criterion, interacting attributes (that can distinguish between classes together but not individually) may be passed over in favor of other attributes that are less discriminating.
- Each decision boundary involves only a single attribute

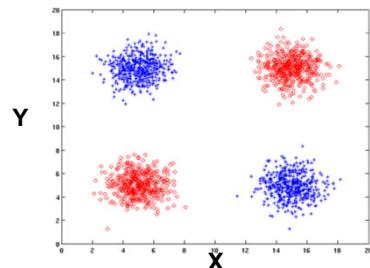
09/21/2020

Introduction to Data Mining, 2nd Edition

55

55

Handling interactions



+ : 1000 instances

o : 1000 instances

Entropy (X) : 0.99

Entropy (Y) : 0.99

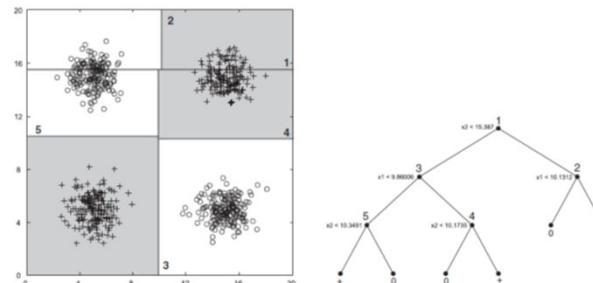
09/21/2020

Introduction to Data Mining, 2nd Edition

56

56

Handling interactions



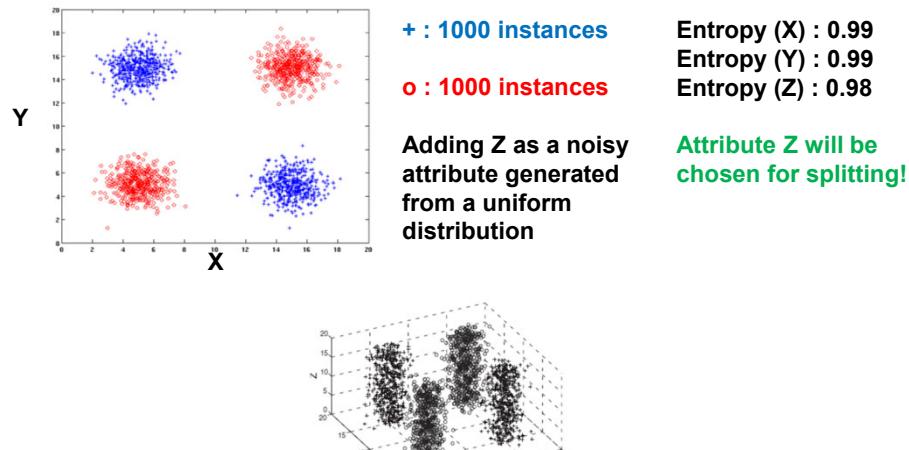
(a) Decision boundary for tree with 6 leaf nodes.

(b) Decision tree with 6 leaf nodes.

Figure 3.28. Decision tree with 6 leaf nodes using X and Y as attributes. Splits have been numbered from 1 to 5 in order of other occurrence in the tree.

57

Handling interactions given irrelevant attributes



(a) Three-dimensional data with attributes X, Y, and Z.

09/21/2020

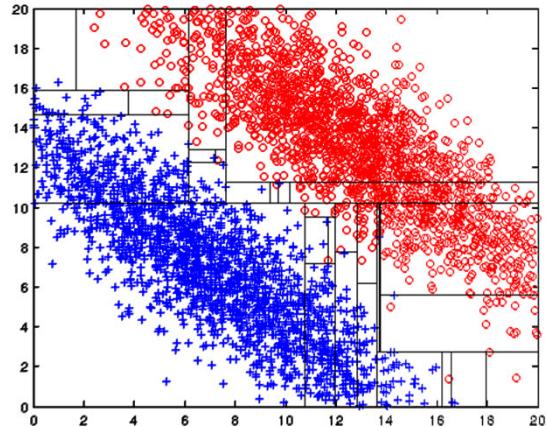
In

1

58

58

Limitations of single attribute-based decision boundaries



Both **positive (+)** and **negative (o)** classes generated from skewed Gaussians with centers at (8,8) and (12,12) respectively.

Data Mining

Model Overfitting

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

09/23/2020

Introduction to Data Mining, 2nd Edition

1

Classification Errors

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

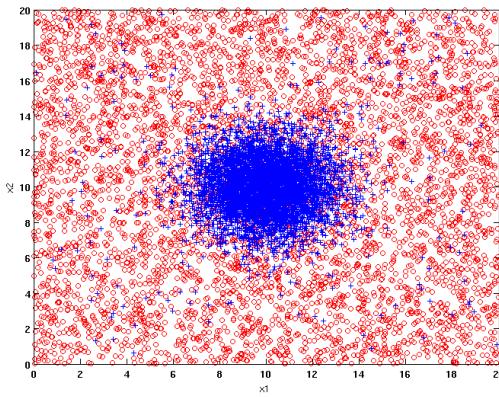
09/23/2020

Introduction to Data Mining, 2nd Edition

2

2

Example Data Set



Two class problem:

+ : 5400 instances

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

o : 5400 instances

- Generated from a uniform distribution

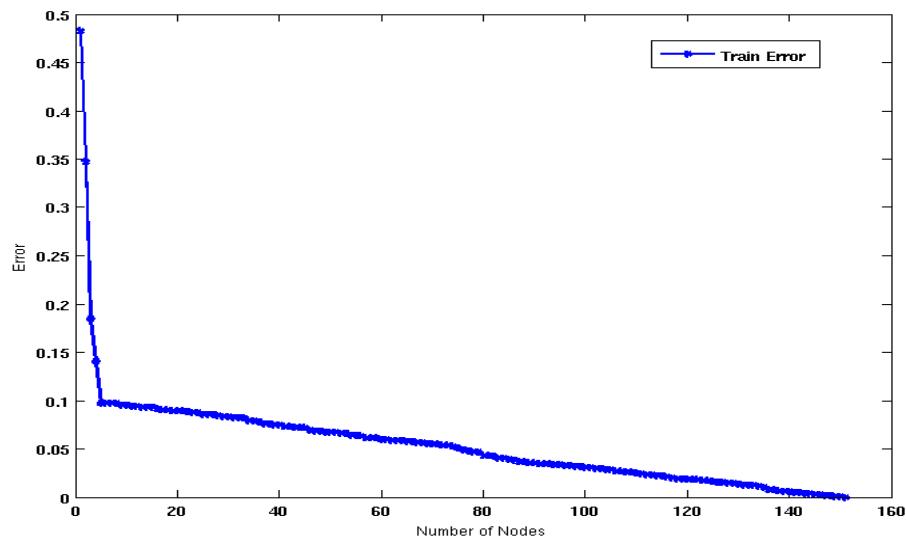
10 % of the data used for training and 90% of the data used for testing

09/23/2020

Introduction to Data Mining, 2nd Edition

3

Increasing number of nodes in Decision Trees

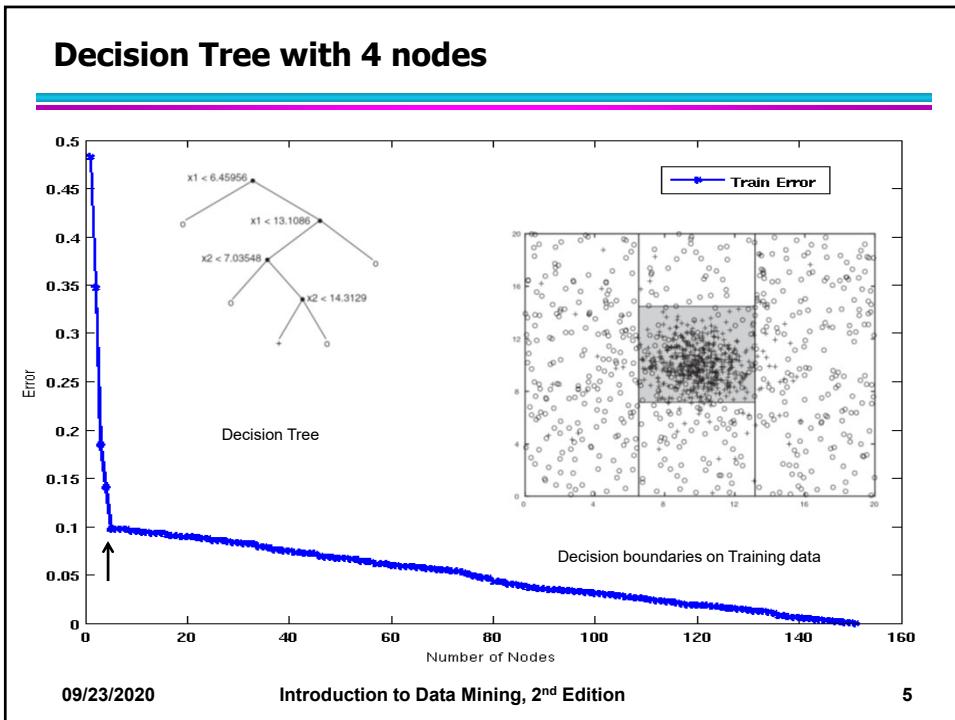


09/23/2020

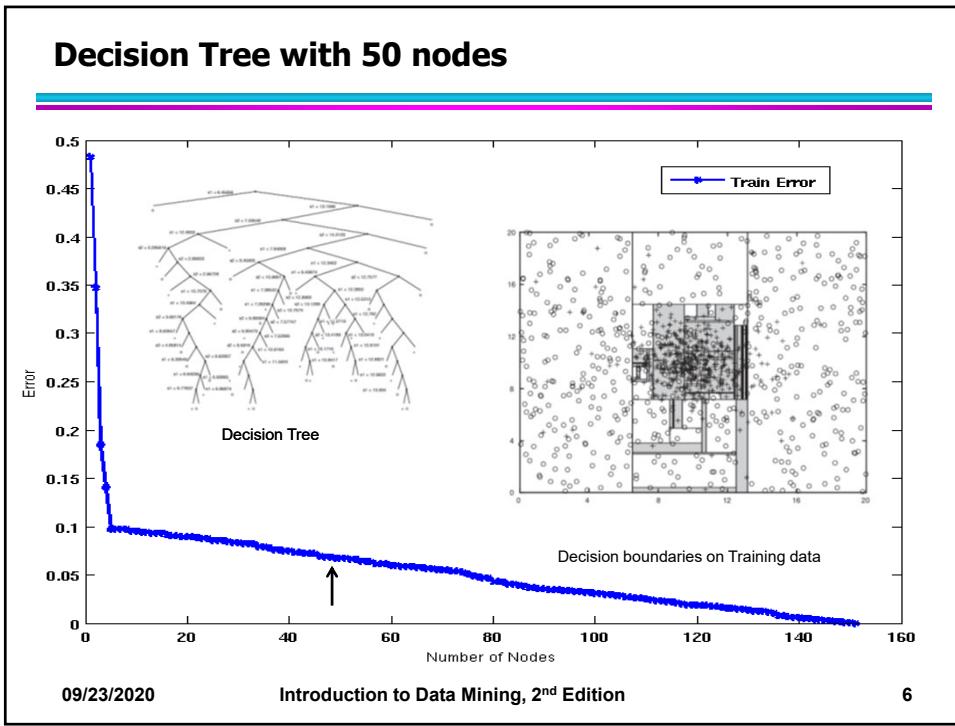
Introduction to Data Mining, 2nd Edition

4

4

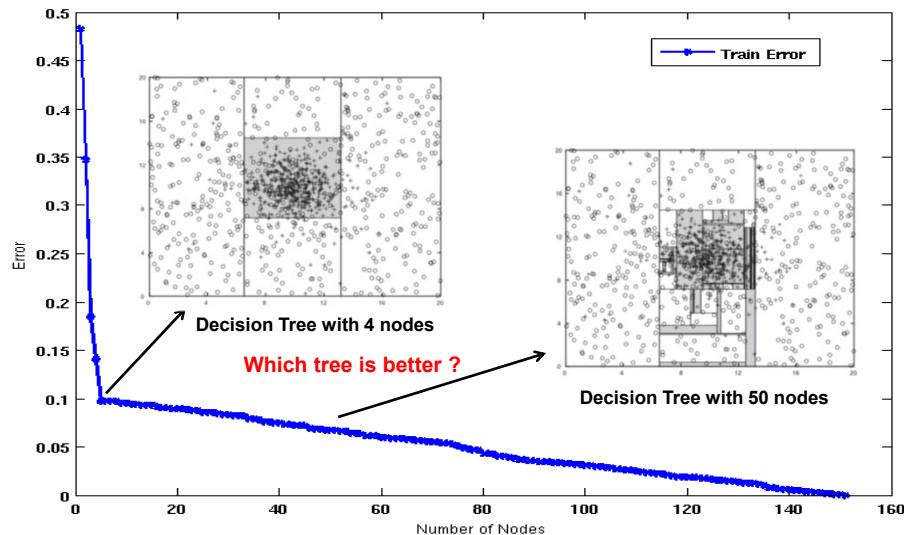


5



6

Which tree is better?

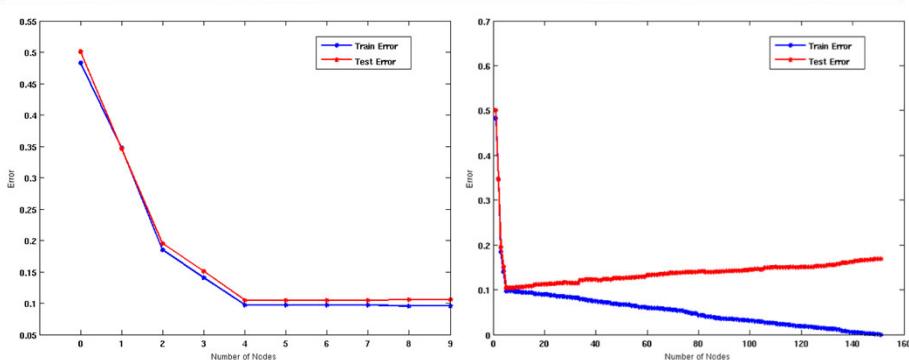


09/23/2020

Introduction to Data Mining, 2nd Edition

7

Model Overfitting



- As the model becomes more and more complex, test errors can start increasing even though training error may be decreasing

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

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

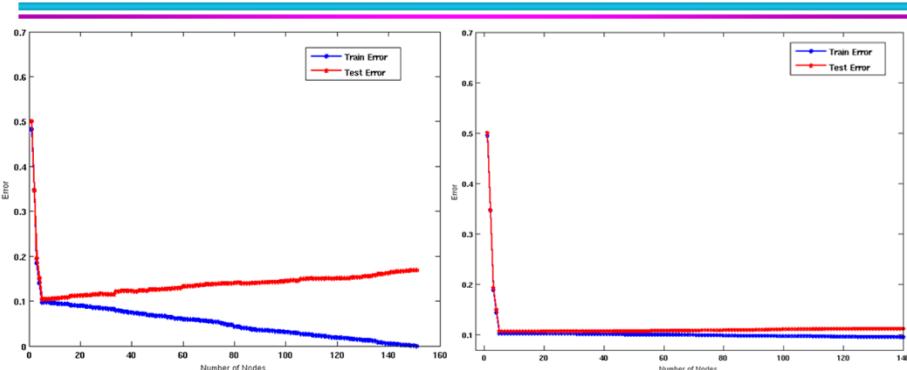
09/23/2020

Introduction to Data Mining, 2nd Edition

8

8

Model Overfitting



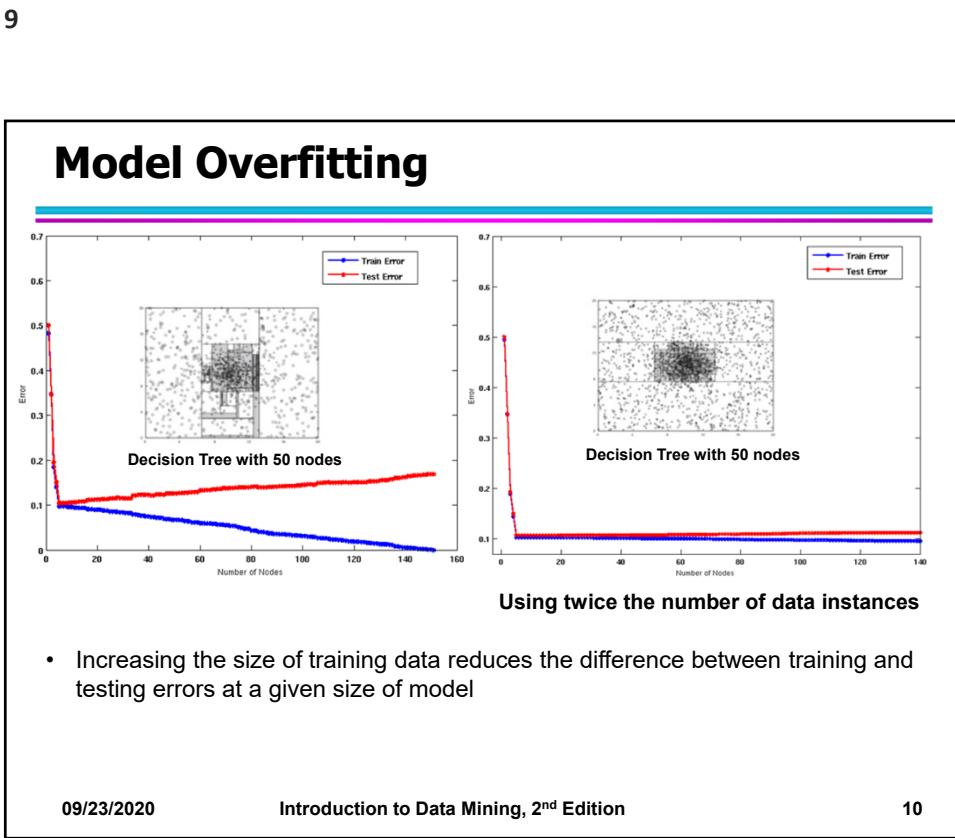
Using twice the number of data instances

- Increasing the size of training data reduces the difference between training and testing errors at a given size of model

09/23/2020

Introduction to Data Mining, 2nd Edition

9



Using twice the number of data instances

- Increasing the size of training data reduces the difference between training and testing errors at a given size of model

09/23/2020

Introduction to Data Mining, 2nd Edition

10

10

Reasons for Model Overfitting

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

09/23/2020

Introduction to Data Mining, 2nd Edition

11

11

Effect of Multiple Comparison Procedure

- Consider the task of predicting whether stock market will rise/fall in the next 10 trading days
- Random guessing:
 $P(\text{correct}) = 0.5$
- Make 10 random guesses in a row:

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

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

09/23/2020

Introduction to Data Mining, 2nd Edition

12

12

Effect of Multiple Comparison Procedure

- Approach:
 - Get 50 analysts
 - Each analyst makes 10 random guesses
 - Choose the analyst that makes the most number of correct predictions
- Probability that at least one analyst makes at least 8 correct predictions

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

09/23/2020

Introduction to Data Mining, 2nd Edition

13

Effect of Multiple Comparison Procedure

- Many algorithms employ the following greedy strategy:
 - Initial model: M
 - Alternative model: $M' = M \cup \gamma$, where γ is a component to be added to the model (e.g., a test condition of a decision tree)
 - Keep M' if improvement, $\Delta(M, M') > \alpha$
- Often times, γ is chosen from a set of alternative components, $\Gamma = \{\gamma_1, \gamma_2, \dots, \gamma_k\}$
- If many alternatives are available, one may inadvertently add irrelevant components to the model, resulting in model overfitting

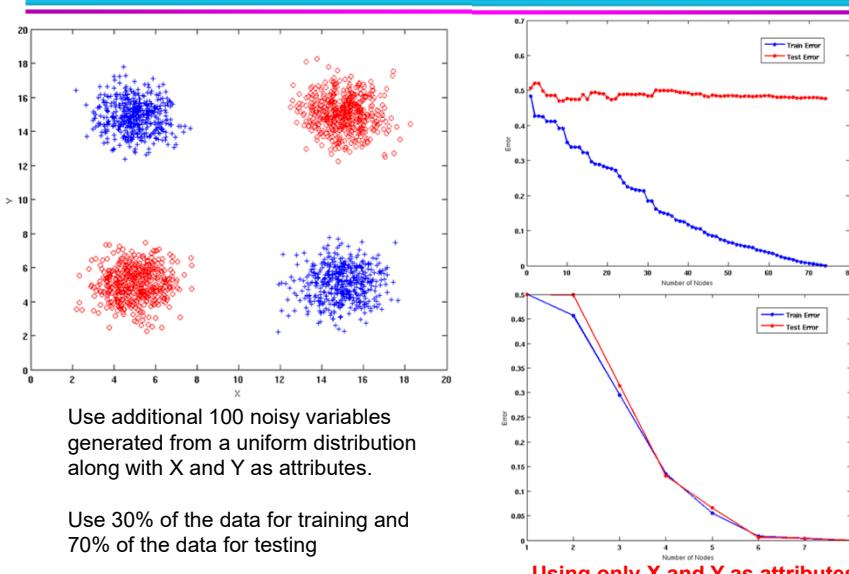
09/23/2020

Introduction to Data Mining, 2nd Edition

14

14

Effect of Multiple Comparison - Example



09/23/2020

Introduction to Data Mining, 2nd Edition

15

Notes on Overfitting

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

09/23/2020

Introduction to Data Mining, 2nd Edition

16

16

Model Selection

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

09/23/2020

Introduction to Data Mining, 2nd Edition

17

17

Model Selection:

Using Validation Set

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

09/23/2020

Introduction to Data Mining, 2nd Edition

18

18

Model Selection:

Incorporating Model Complexity

- Rationale: Occam's Razor
 - Given two models of similar generalization errors, one should prefer the simpler model over the more complex model
 - A complex model has a greater chance of being fitted accidentally
 - Therefore, one should include model complexity when evaluating a model

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

09/23/2020

Introduction to Data Mining, 2nd Edition

19

19

Estimating the Complexity of Decision Trees

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

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

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

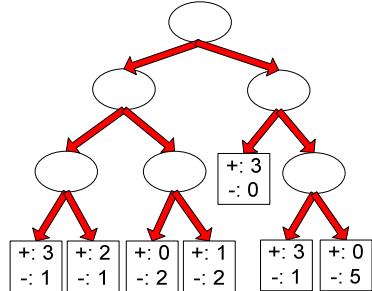
09/23/2020

Introduction to Data Mining, 2nd Edition

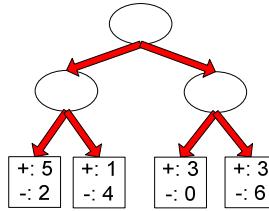
20

20

Estimating the Complexity of Decision Trees: Example



Decision Tree, T_L



Decision Tree, T_R

$$e(T_L) = 4/24$$

$$e(T_R) = 6/24$$

$$\Omega = 1$$

$$e_{gen}(T_L) = 4/24 + 1 \cdot 7/24 = 11/24 = 0.458$$

$$e_{gen}(T_R) = 6/24 + 1 \cdot 4/24 = 10/24 = 0.417$$

09/23/2020

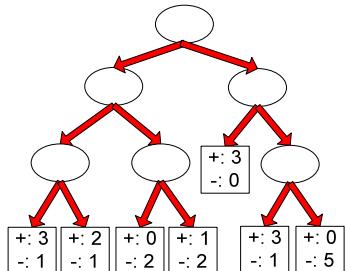
Introduction to Data Mining, 2nd Edition

21

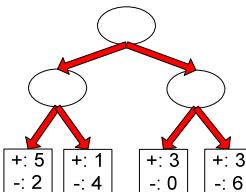
Estimating the Complexity of Decision Trees

● Resubstitution Estimate:

- Using training error as an optimistic estimate of generalization error
- Referred to as optimistic error estimate



Decision Tree, T_L



Decision Tree, T_R

$$e(T_L) = 4/24$$

$$e(T_R) = 6/24$$

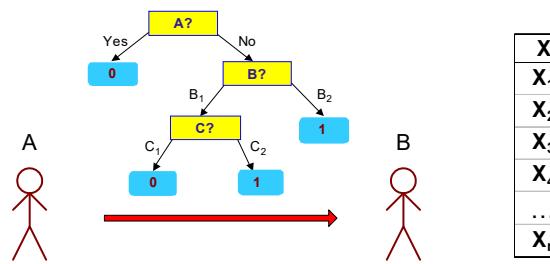
09/23/2020

Introduction to Data Mining, 2nd Edition

22

Minimum Description Length (MDL)

X	y
X ₁	1
X ₂	0
X ₃	0
X ₄	1
...	...
X _n	1



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

09/23/2020

Introduction to Data Mining, 2nd Edition

23

23

Model Selection for Decision Trees

- 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).
 - ◆ Stop if estimated generalization error falls below certain threshold

09/23/2020

Introduction to Data Mining, 2nd Edition

24

24

Model Selection for Decision Trees

● Post-pruning

- Grow decision tree to its entirety
- Subtree replacement
 - ◆ 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

09/23/2020

Introduction to Data Mining, 2nd Edition

25

25

Example of Post-Pruning

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

Training Error (Before splitting) = 10/30

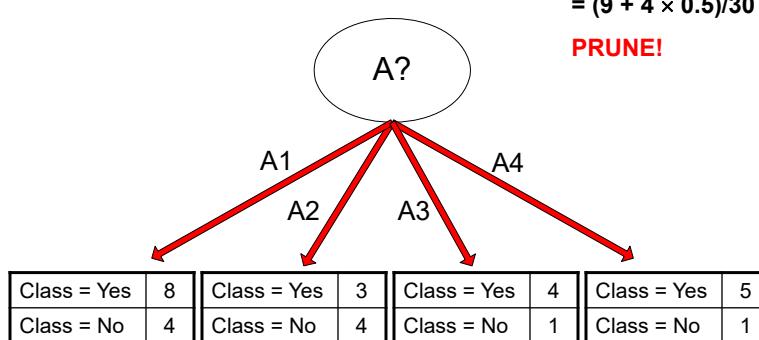
Pessimistic error = $(10 + 0.5)/30 = 10.5/30$

Training Error (After splitting) = 9/30

Pessimistic error (After splitting)

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

PRUNE!



09/23/2020

Introduction to Data Mining, 2nd Edition

26

26

Examples of Post-pruning

Decision Tree:

```
depth = 1 :  
| breadth > 7 : class 1  
| breadth <= 7 :  
| | breadth <= 3 :  
| | | ImagePages <= 0.375 : class 0  
| | | ImagePages <= 0.375 :  
| | | | totalPages <= 6 : class 1  
| | | | totalPages > 6 :  
| | | | | breadth <= 1 : class 1  
| | | | | breadth > 1 : class 0  
| | width > 3 :  
| | | MultiP = 0:  
| | | | ImagePages <= 0.1333 : class 1  
| | | | ImagePages > 0.1333 :  
| | | | | breadth <= 6 : class 0  
| | | | | breadth > 6 : class 1  
| | | | MultiP = 1:  
| | | | TotalTime <= 361 : class 0  
| | | | TotalTime > 361 : class 1  
depth > 1 :  
| MultiAgent = 0:  
| | depth > 2 : class 0  
| | depth <= 2 :  
| | | MultiP = 1: class 0  
| | | MultiP = 0:  
| | | | breadth <= 6 : class 0  
| | | | breadth > 6 :  
| | | | | RepeatedAccess <= 0.0322 : class 0  
| | | | | RepeatedAccess > 0.0322 : class 1  
| | | MultiAgent = 1:  
| | | | totalPages <= 81 : class 0  
| | | | totalPages > 81 : class 1
```

Simplified Decision Tree:

```
depth = 1 :  
| | ImagePages <= 0.1333 : class 1  
| | ImagePages > 0.1333 :  
| | | breadth <= 6 : class 0  
| | | breadth > 6 : class 1  
depth > 1 :  
| | MultiAgent = 0: class 0  
| | MultiAgent = 1:  
| | | totalPages <= 81 : class 0  
| | | totalPages > 81 : class 1
```

Subtree Raising

Subtree Replacement

09/23/2020

Introduction to Data Mining, 2nd Edition

27

27

Model Evaluation

- Purpose:
 - To estimate performance of classifier on previously unseen data (test set)
- Holdout
 - Reserve k% for training and (100-k)% for testing
 - Random subsampling: repeated holdout
- Cross validation
 - Partition data into k disjoint subsets
 - k-fold: train on k-1 partitions, test on the remaining one
 - Leave-one-out: k=n

09/23/2020

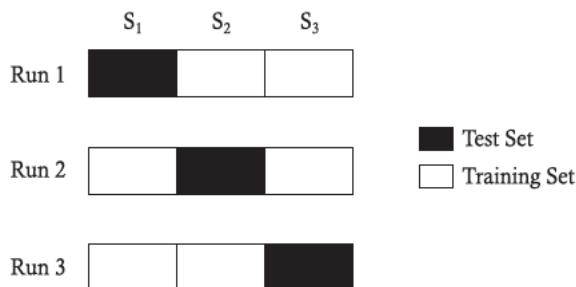
Introduction to Data Mining, 2nd Edition

28

28

Cross-validation Example

- 3-fold cross-validation



09/23/2020

Introduction to Data Mining, 2nd Edition

29

Variations on Cross-validation

- Repeated cross-validation
 - Perform cross-validation a number of times
 - Gives an estimate of the variance of the generalization error
- Stratified cross-validation
 - Guarantee the same percentage of class labels in training and test
 - Important when classes are imbalanced and the sample is small
- Use nested cross-validation approach for model selection and evaluation

09/23/2020

Introduction to Data Mining, 2nd Edition

30

30

Data Mining

Lecture Notes for Chapter 4

Artificial Neural Networks

Introduction to Data Mining , 2nd Edition

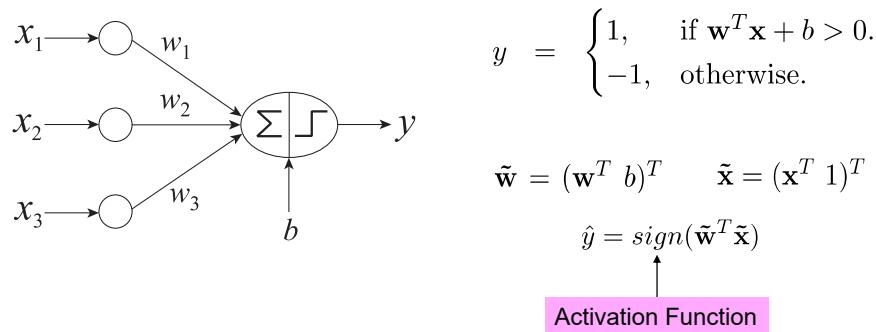
by

Tan, Steinbach, Karpatne, Kumar

Artificial Neural Networks (ANN)

- **Basic Idea:** A complex non-linear function can be learned as a composition of simple processing units
- ANN is a collection of simple processing units (nodes) that are connected by directed links (edges)
 - Every node receives signals from incoming edges, performs computations, and transmits signals to outgoing edges
 - Analogous to *human brain* where nodes are neurons and signals are electrical impulses
 - Weight of an edge determines the strength of connection between the nodes
 - Simplest ANN: **Perceptron** (single neuron)

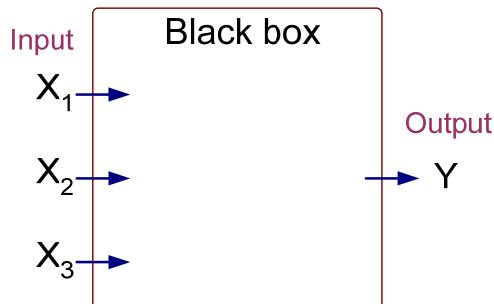
Basic Architecture of Perceptron



- Learns linear decision boundaries
- Similar to logistic regression (activation function is sign instead of sigmoid)

Perceptron Example

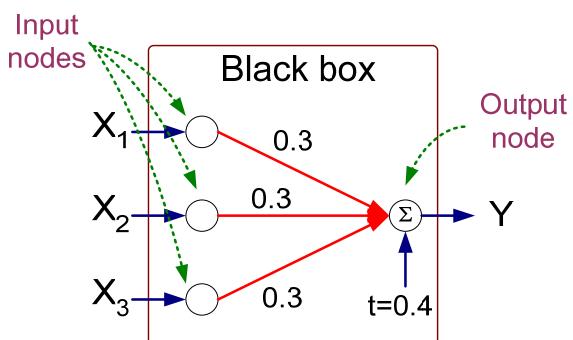
X ₁	X ₂	X ₃	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1



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

Perceptron Example

X ₁	X ₂	X ₃	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1



$$Y = \text{sign}(0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4)$$

$$\text{where } \text{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases}$$

Perceptron Learning Rule

- Initialize the weights (w_0, w_1, \dots, w_d)

- Repeat

- For each training example (x_i, y_i)

- ◆ Compute \hat{y}_i

- ◆ Update the weights:

$$w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}$$

- Until stopping condition is met

- k: iteration number; λ : learning rate

Perceptron Learning Rule

- Weight update formula:

$$w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}$$

- Intuition:

- Update weight based on error: $e = (y_i - \hat{y}_i)$
- If $y = \hat{y}$, $e=0$: no update needed
- If $y > \hat{y}$, $e=2$: weight must be increased so that \hat{y} will increase
- If $y < \hat{y}$, $e=-2$: weight must be decreased so that \hat{y} will decrease

Example of Perceptron Learning

$$\lambda = 0.1$$

X ₁	X ₂	X ₃	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1

	w ₀	w ₁	w ₂	w ₃
0	0	0	0	0
1	-0.2	-0.2	0	0
2	0	0	0	0.2
3	0	0	0	0.2
4	0	0	0	0.2
5	-0.2	0	0	0
6	-0.2	0	0	0
7	0	0	0.2	0.2
8	-0.2	0	0.2	0.2

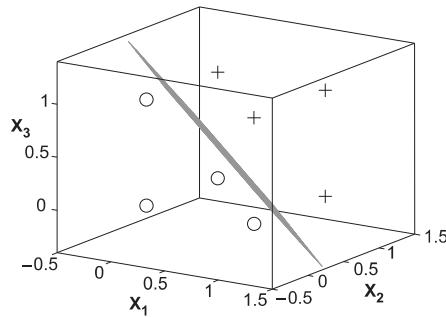
Weight updates over first epoch

Epoch	w ₀	w ₁	w ₂	w ₃
0	0	0	0	0
1	-0.2	0	0.2	0.2
2	-0.2	0	0.4	0.2
3	-0.4	0	0.4	0.2
4	-0.4	0.2	0.4	0.4
5	-0.6	0.2	0.4	0.2
6	-0.6	0.4	0.4	0.2

Weight updates over all epochs

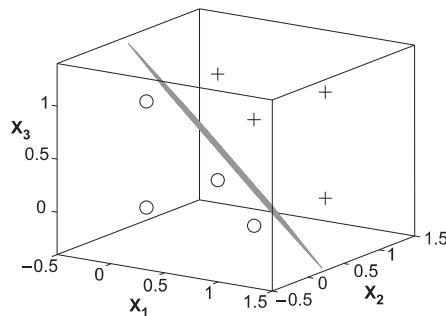
Perceptron Learning

- Since y is a linear combination of input variables, decision boundary is linear



Perceptron Learning

- Since y is a linear combination of input variables, decision boundary is linear

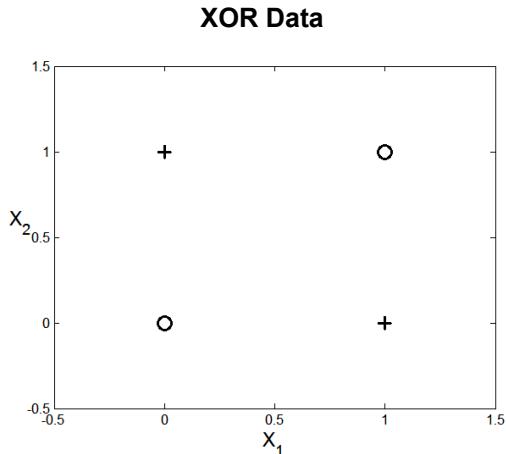


- For nonlinearly separable problems, perceptron learning algorithm will fail because no linear hyperplane can separate the data perfectly

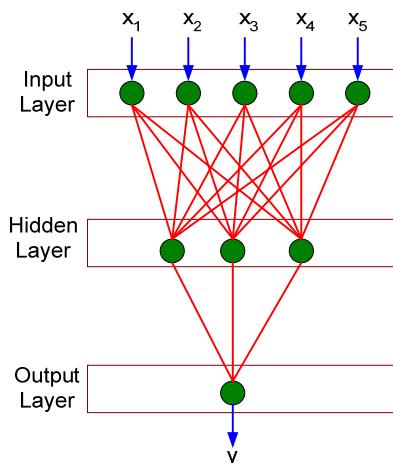
Nonlinearly Separable Data

$$y = x_1 \oplus x_2$$

x_1	x_2	y
0	0	-1
1	0	1
0	1	1
1	1	-1



Multi-layer Neural Network

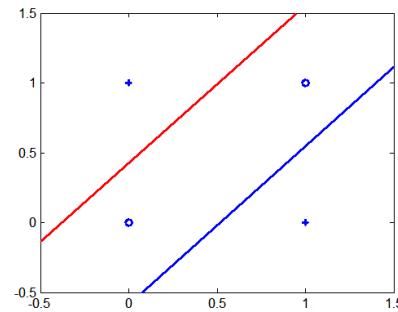
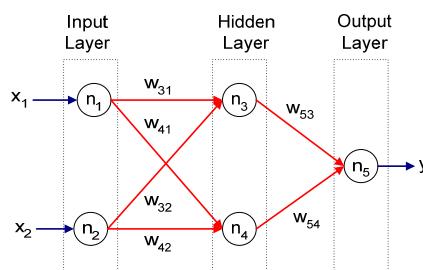


- More than one *hidden layer* of computing nodes
- Every node in a hidden layer operates on activations from preceding layer and transmits activations forward to nodes of next layer
- Also referred to as “feedforward neural networks”

Multi-layer Neural Network

- Multi-layer neural networks with at least one hidden layer can solve any type of classification task involving nonlinear decision surfaces

XOR Data



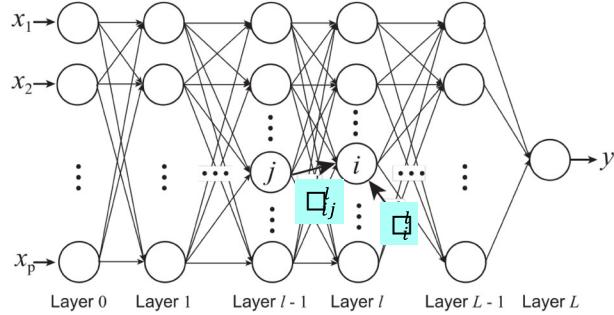
Why Multiple Hidden Layers?

- Activations at hidden layers can be viewed as features extracted as functions of inputs
- Every hidden layer represents a level of abstraction
 - Complex features are compositions of simpler features*



- Number of layers is known as **depth** of ANN
 - Deeper networks express complex hierarchy of features*

Multi-Layer Network Architecture

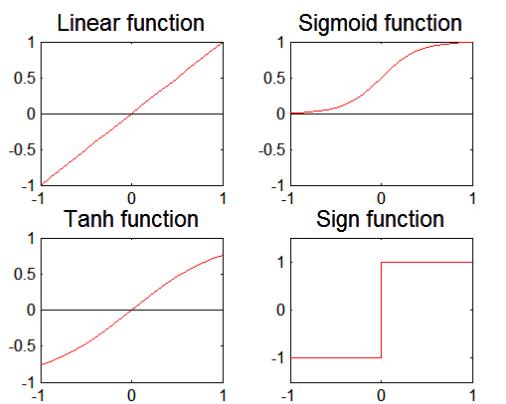


$$a_i^l = f(z_i^l) = f\left(\sum_j w_{ij}^l a_j^{l-1} + b_i^l\right)$$

Activation value
 at node i at layer l
Activation Function
Linear Predictor

Activation Functions

$$a_i^l = f(z_i^l) = f\left(\sum_j w_{ij}^l a_j^{l-1} + b_i^l\right)$$



$$a_i^l = \sigma(z_i^l) = \frac{1}{1 + e^{-z_i^l}}$$

$$\frac{\partial a_i^l}{\partial z_i^l} = \frac{\partial \sigma(z_i^l)}{\partial z_i^l} = a_i^l(1 - a_i^l)$$

Learning Multi-layer Neural Network

- Can we apply perceptron learning rule to each node, including hidden nodes?
 - Perceptron learning rule computes error term $e = y - \hat{y}$ and updates weights accordingly
 - ◆ Problem: how to determine the true value of y for hidden nodes?
 - Approximate error in hidden nodes by error in the output nodes
 - ◆ Problem:
 - Not clear how adjustment in the hidden nodes affect overall error
 - No guarantee of convergence to optimal solution

Gradient Descent

- Loss Function to measure errors across all training points

$$E(\mathbf{w}, \mathbf{b}) = \sum_{k=1}^n \text{Loss}(y_k, \hat{y}_k) \quad \begin{aligned} &\text{Squared Loss:} \\ &\text{Loss}(y_k, \hat{y}_k) = (y_k - \hat{y}_k)^2. \end{aligned}$$

- Gradient descent: Update parameters in the direction of “maximum descent” in the loss function across all points

$$\begin{aligned} w_{ij}^l &\leftarrow w_{ij}^l - \lambda \frac{\partial E}{\partial w_{ij}^l}, & \lambda: \text{learning rate} \\ b_i^l &\leftarrow b_i^l - \lambda \frac{\partial E}{\partial b_i^l}, \end{aligned}$$

- Stochastic gradient descent (SGD): update the weight for every instance (minibatch SGD: update over min-batches of instances)

Computing Gradients

$$\frac{\partial E}{\partial w_{ij}^l} = \sum_{k=1}^n \frac{\partial \text{Loss}(y_k, \hat{y}_k)}{\partial w_{ij}^l}. \quad \hat{y} = a^L$$
$$a_i^l = f(z_i^l) = f\left(\sum_j w_{ij}^l a_j^{l-1} + b_i^l\right)$$

- Using chain rule of differentiation (on a single instance):

$$\frac{\partial \text{Loss}}{\partial w_{ij}^l} = \frac{\partial \text{Loss}}{\partial a_i^l} \times \frac{\partial a_i^l}{\partial z_i^l} \times \frac{\partial z_i^l}{\partial w_{ij}^l}.$$

- For sigmoid activation function:

$$\frac{\partial \text{Loss}}{\partial w_{ij}^l} = \delta_i^l \times a_i^l (1 - a_i^l) \times a_j^{l-1},$$

$$\text{where } \delta_i^l = \frac{\partial \text{Loss}}{\partial a_i^l}.$$

- How can we compute δ_i^l for every layer?

Backpropagation Algorithm

- At output layer L:

$$\delta^L = \frac{\partial \text{Loss}}{\partial a^L} = \frac{\partial (y - a^L)^2}{\partial a^L} = 2(a^L - y).$$

- At a hidden layer l (using chain rule):

$$\delta_j^l = \sum_i (\delta_i^{l+1} \times a_i^{l+1} (1 - a_i^{l+1}) \times w_{ij}^{l+1}).$$

- Gradients at layer l can be computed using gradients at layer l + 1
- Start from layer L and “backpropagate” gradients to all previous layers

- Use gradient descent to update weights at every epoch
- For next epoch, use updated weights to compute loss fn. and its gradient
- Iterate until convergence (loss does not change)

Design Issues in ANN

- Number of nodes in input layer
 - One input node per binary/continuous attribute
 - k or $\log_2 k$ nodes for each categorical attribute with k values
- Number of nodes in output layer
 - One output for binary class problem
 - k or $\log_2 k$ nodes for k -class problem
- Number of hidden layers and nodes per layer
- Initial weights and biases
- Learning rate, max. number of epochs, mini-batch size for mini-batch SGD, ...

Characteristics of ANN

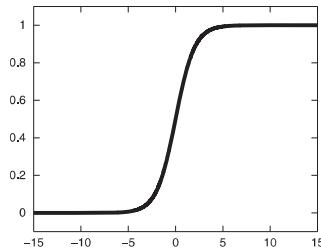
- Multilayer ANN are universal approximators but could suffer from overfitting if the network is too large
- Gradient descent may converge to local minimum
- Model building can be very time consuming, but testing can be very fast
- Can handle redundant and irrelevant attributes because weights are automatically learnt for all attributes
- Sensitive to noise in training data
- Difficult to handle missing attributes

Deep Learning Trends

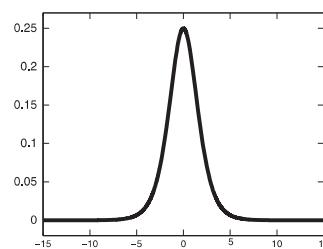
- Training **deep** neural networks (more than 5-10 layers) could only be possible in recent times with:
 - Faster computing resources (GPU)
 - Larger labeled training sets
 - **Algorithmic Improvements in Deep Learning**
- Recent Trends:
 - Specialized ANN Architectures:
 - ◆ Convolutional Neural Networks (for image data)
 - ◆ Recurrent Neural Networks (for sequence data)
 - ◆ Residual Networks (with skip connections)
 - Unsupervised Models: Autoencoders
 - Generative Models: Generative Adversarial Networks

23

Vanishing Gradient Problem



(a) $\sigma(z)$.



(b) $\partial\sigma(z)/\partial z$.

- Sigmoid activation function easily saturates (show zero gradient with z) when z is too large or too small
- Lead to small (or zero) gradients of squared loss with weights, especially at hidden layers, leading to slow (or no) learning

$$\frac{\partial \text{Loss}}{\partial w_j^L} = 2(a^L - y) \times \sigma(z^L)(1 - \sigma(z^L)) \times a_j^{L-1}.$$

24

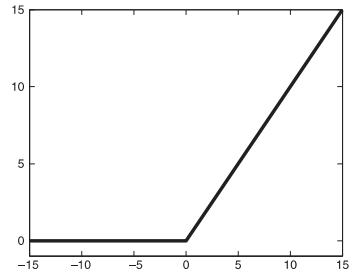
Handling Vanishing Gradient Problem

- Use of Cross-entropy loss function

$$\text{Loss } (y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

$$\frac{\partial \text{Loss}}{\partial w_j^L} = (a^L - y) \times a_j^{L-1}$$

- Use of Rectified Linear Unit (ReLU) Activations:



$$a = f(z) = \begin{cases} z, & \text{if } z > 0. \\ 0, & \text{otherwise.} \end{cases}$$

$$\frac{\partial a}{\partial z} = \begin{cases} 1, & \text{if } z > 0. \\ 0, & \text{if } z < 0. \end{cases}$$

Data Mining

Ensemble Techniques

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

10/7/2020 Intro to Data Mining, 2nd Edition

1

Ensemble Methods

- Construct a set of base classifiers learned from the training data
- Predict class label of test records by combining the predictions made by multiple classifiers (e.g., by taking majority vote)

10/7/2020 Intro to Data Mining, 2nd Edition

2

2

Example: Why Do Ensemble Methods Work?

- Suppose there are 25 base classifiers
 - Each classifier has error rate, $\epsilon = 0.35$
 - Majority vote of classifiers used for classification
 - If all classifiers are identical:
 - ◆ Error rate of ensemble = $\epsilon (0.35)$
 - If all classifiers are independent (errors are uncorrelated):
 - ◆ Error rate of ensemble = probability of having more than half of base classifiers being wrong

$$e_{\text{ensemble}} = \sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1 - \epsilon)^{25-i} = 0.06$$

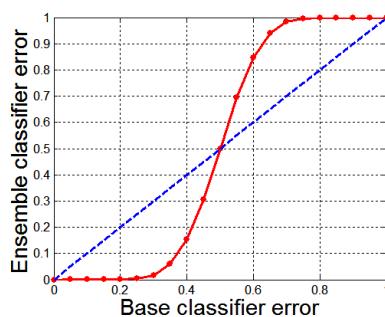
10/7/2020 Intro to Data Mining, 2nd Edition

3

3

Necessary Conditions for Ensemble Methods

- Ensemble Methods work better than a single base classifier if:
 1. All base classifiers are independent of each other
 2. All base classifiers perform better than random guessing (error rate < 0.5 for binary classification)



Classification error for an ensemble of 25 base classifiers, assuming their errors are uncorrelated.

10/7/2020 Intro to Data Mining, 2nd Edition

4

4

Rationale for Ensemble Learning

- Ensemble Methods work best with **unstable base classifiers**
 - Classifiers that are sensitive to minor perturbations in training set, due to *high model complexity*
 - Examples: Unpruned decision trees, ANNs, ...
 - **Low Bias** in finding optimal decision boundary
 - **High Variance** for minor changes in training set or model selection procedure

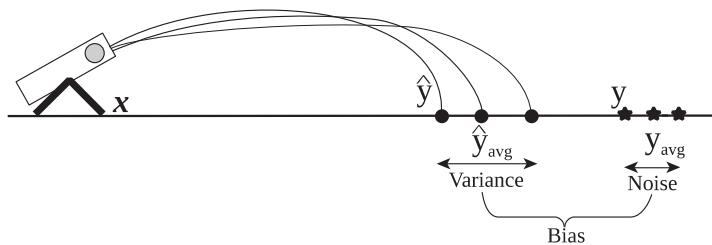
10/7/2020 Intro to Data Mining, 2nd Edition

5

5

Bias-Variance Decomposition

- Analogous problem of reaching a target y by firing projectiles from x (regression problem)



- For classification, gen. error or model m can be given by:

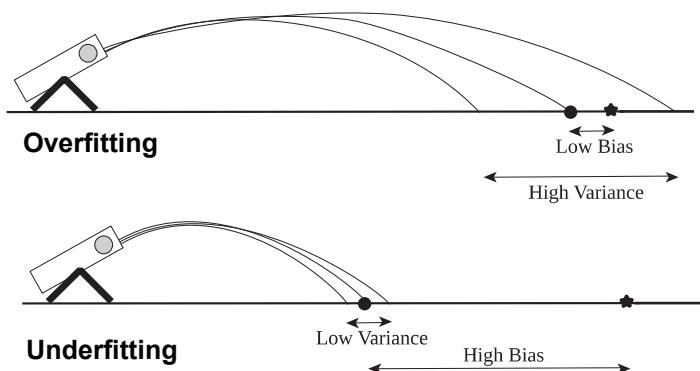
$$gen.error(m) = c_1 \times noise + bias(m) + c_2 \times variance(m)$$

10/7/2020 Intro to Data Mining, 2nd Edition

6

6

Bias-Variance Trade-off and Overfitting



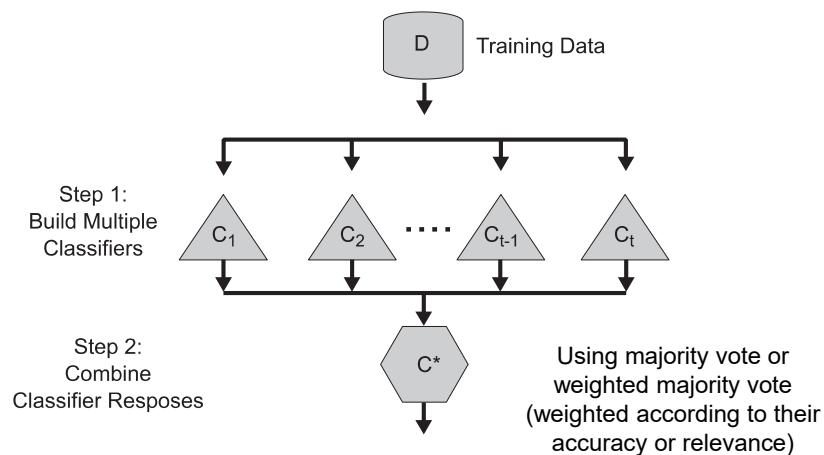
- Ensemble methods try to reduce the variance of complex models (with low bias) by *aggregating* responses of multiple base classifiers

10/7/2020 Intro to Data Mining, 2nd Edition

7

7

General Approach of Ensemble Learning



10/7/2020 Intro to Data Mining, 2nd Edition

8

8

Constructing Ensemble Classifiers

- By manipulating training set
 - Example: bagging, boosting
- By manipulating input features
 - Example: random forests
- By manipulating class labels
 - Example: error-correcting output coding
- By manipulating learning algorithm
 - Example: injecting randomness in ANN or decision tree

10/7/2020 Intro to Data Mining, 2nd Edition

9

9

Bagging (Bootstrap AGGRegatING)

- Bootstrap sampling: 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
- Probability of a training instance being selected in a bootstrap sample is:
 - $1 - (1 - 1/n)^n$ (n : number of training instances)
 - ~ 0.632 when n is large

10/7/2020 Intro to Data Mining, 2nd Edition

10

10

Bagging Algorithm

Algorithm 4.5 Bagging algorithm.

- 1: Let k be the number of bootstrap samples.
- 2: **for** $i = 1$ to k **do**
- 3: Create a bootstrap sample of size N , D_i .
- 4: Train a base classifier C_i on the bootstrap sample D_i .
- 5: **end for**
- 6: $C^*(x) = \operatorname{argmax}_y \sum_i \delta(C_i(x) = y).$
 $\{\delta(\cdot) = 1 \text{ if its argument is true and 0 otherwise.}\}$

10/7/2020 Intro to Data Mining, 2nd Edition

11

11

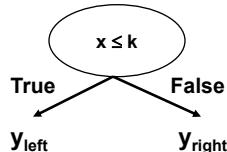
Bagging Example

- Consider 1-dimensional data set:

Original Data:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

- Classifier is a decision stump (decision tree of size 1)
 - Decision rule: $x \leq k$ versus $x > k$
 - Split point k is chosen based on entropy



10/7/2020 Intro to Data Mining, 2nd Edition

12

12

Bagging Example

Bagging Round 1:

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

$$x \leq 0.35 \rightarrow y = 1$$
$$x > 0.35 \rightarrow y = -1$$

10/7/2020 Intro to Data Mining, 2nd Edition

13

13

Bagging Example

Bagging Round 1:

x	0.1	0.2	0.2	0.3	0.4	0.4	0.5	0.6	0.9	0.9
y	1	1	1	1	-1	-1	-1	-1	1	1

$$x \leq 0.35 \rightarrow y = 1$$
$$x > 0.35 \rightarrow y = -1$$

Bagging Round 2:

x	0.1	0.2	0.3	0.4	0.5	0.5	0.9	1	1	1
y	1	1	1	-1	-1	-1	1	1	1	1

$$x \leq 0.7 \rightarrow y = 1$$
$$x > 0.7 \rightarrow y = 1$$

Bagging Round 3:

x	0.1	0.2	0.3	0.4	0.4	0.5	0.7	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

$$x \leq 0.35 \rightarrow y = 1$$
$$x > 0.35 \rightarrow y = -1$$

Bagging Round 4:

x	0.1	0.1	0.2	0.4	0.4	0.5	0.5	0.7	0.8	0.9
y	1	1	1	-1	-1	-1	-1	-1	1	1

$$x \leq 0.3 \rightarrow y = 1$$
$$x > 0.3 \rightarrow y = -1$$

Bagging Round 5:

x	0.1	0.1	0.2	0.5	0.6	0.6	0.6	1	1	1
y	1	1	1	-1	-1	-1	-1	1	1	1

$$x \leq 0.35 \rightarrow y = 1$$
$$x > 0.35 \rightarrow y = -1$$

10/7/2020 Intro to Data Mining, 2nd Edition

14

14

Bagging Example

Bagging Round 6:

x	0.2	0.4	0.5	0.6	0.7	0.7	0.7	0.8	0.9	1
y	1	-1	-1	-1	-1	-1	-1	1	1	1

$$\begin{aligned}x \leq 0.75 &\rightarrow y = -1 \\x > 0.75 &\rightarrow y = 1\end{aligned}$$

Bagging Round 7:

x	0.1	0.4	0.4	0.6	0.7	0.8	0.9	0.9	0.9	1
y	1	-1	-1	-1	-1	1	1	1	1	1

$$\begin{aligned}x \leq 0.75 &\rightarrow y = -1 \\x > 0.75 &\rightarrow y = 1\end{aligned}$$

Bagging Round 8:

x	0.1	0.2	0.5	0.5	0.5	0.7	0.7	0.8	0.9	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

$$\begin{aligned}x \leq 0.75 &\rightarrow y = -1 \\x > 0.75 &\rightarrow y = 1\end{aligned}$$

Bagging Round 9:

x	0.1	0.3	0.4	0.4	0.6	0.7	0.7	0.8	1	1
y	1	1	-1	-1	-1	-1	-1	1	1	1

$$\begin{aligned}x \leq 0.75 &\rightarrow y = -1 \\x > 0.75 &\rightarrow y = 1\end{aligned}$$

Bagging Round 10:

x	0.1	0.1	0.1	0.1	0.3	0.3	0.8	0.8	0.9	0.9
y	1	1	1	1	1	1	1	1	1	1

$$\begin{aligned}x \leq 0.05 &\rightarrow y = 1 \\x > 0.05 &\rightarrow y = 1\end{aligned}$$

10/7/2020 Intro to Data Mining, 2nd Edition

15

15

Bagging Example

- Summary of Trained Decision Stumps:

Round	Split Point	Left Class	Right Class
1	0.35	1	-1
2	0.7	1	1
3	0.35	1	-1
4	0.3	1	-1
5	0.35	1	-1
6	0.75	-1	1
7	0.75	-1	1
8	0.75	-1	1
9	0.75	-1	1
10	0.05	1	1

10/7/2020 Intro to Data Mining, 2nd Edition

16

16

Bagging Example

- Use majority vote (sign of sum of predictions) to determine class of ensemble classifier

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	1	1	1	-1	-1	-1	-1	-1	-1	-1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
4	1	1	1	-1	-1	-1	-1	-1	-1	-1
5	1	1	1	-1	-1	-1	-1	-1	-1	-1
6	-1	-1	-1	-1	-1	-1	-1	1	1	1
7	-1	-1	-1	-1	-1	-1	-1	1	1	1
8	-1	-1	-1	-1	-1	-1	-1	1	1	1
9	-1	-1	-1	-1	-1	-1	-1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
Sum	2	2	2	-6	-6	-6	-6	2	2	2
Sign	1	1	1	-1	-1	-1	-1	1	1	1

- Bagging can also increase the complexity (representation capacity) of simple classifiers such as decision stumps

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 (for being selected for training)
 - Unlike bagging, weights may change at the end of each boosting round

Boosting

- Records that are wrongly classified will have their weights increased in the next round
- Records that are classified correctly will have their weights decreased in the next round

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

10/7/2020 Intro to Data Mining, 2nd Edition

19

19

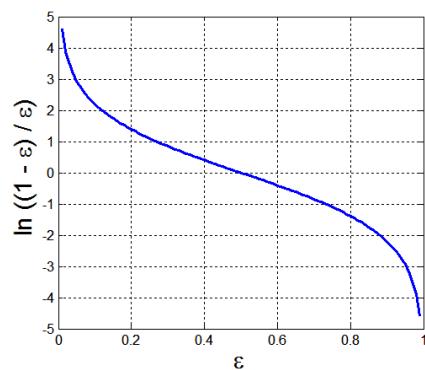
AdaBoost

- Base classifiers: C_1, C_2, \dots, C_T
- Error rate of a base classifier:

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

- Importance of a classifier:

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



10/7/2020 Intro to Data Mining, 2nd Edition

20

20

AdaBoost Algorithm

- Weight update:

$$w_j^{(i+1)} = \frac{w_j^{(i)}}{Z_i} \times \begin{cases} e^{-\alpha_i} & \text{if } C_i(x_j) = y_j \\ e^{\alpha_i} & \text{if } C_i(x_j) \neq y_j \end{cases}$$

Where Z_i 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:

$$C^*(x) = \arg \max_y \sum_{i=1}^T \alpha_i \delta(C_i(x) = y)$$

10/7/2020 Intro to Data Mining, 2nd Edition

21

21

AdaBoost Algorithm

Algorithm 4.6 AdaBoost algorithm.

```
1:  $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \dots, N\}$ . {Initialize the weights for all  $N$  examples.}
2: Let  $k$  be the number of boosting rounds.
3: for  $i = 1$  to  $k$  do
4:   Create training set  $D_i$  by sampling (with replacement) from  $D$  according to  $\mathbf{w}$ .
5:   Train a base classifier  $C_i$  on  $D_i$ .
6:   Apply  $C_i$  to all examples in the original training set,  $D$ .
7:    $\epsilon_i = \frac{1}{N} [\sum_j w_j \delta(C_i(x_j) \neq y_j)]$  {Calculate the weighted error.}
8:   if  $\epsilon_i > 0.5$  then
9:      $\mathbf{w} = \{w_j = 1/N \mid j = 1, 2, \dots, N\}$ . {Reset the weights for all  $N$  examples.}
10:    Go back to Step 4.
11:   end if
12:    $\alpha_i = \frac{1}{2} \ln \frac{1-\epsilon_i}{\epsilon_i}$ .
13:   Update the weight of each example according to Equation 4.103.
14: end for
15:  $C^*(\mathbf{x}) = \operatorname{argmax}_y \sum_{j=1}^T \alpha_j \delta(C_j(\mathbf{x}) = y)$ .
```

10/7/2020 Intro to Data Mining, 2nd Edition

22

22

AdaBoost Example

- Consider 1-dimensional data set:

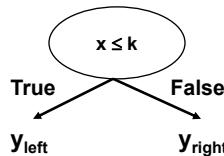
Original Data:

x	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
y	1	1	1	-1	-1	-1	-1	1	1	1

- Classifier is a decision stump

Decision rule: $x \leq k$ versus $x > k$

Split point k is chosen based on entropy



10/7/2020 Intro to Data Mining, 2nd Edition

23

23

AdaBoost Example

- Training sets for the first 3 boosting rounds:

Boosting Round 1:

x	0.1	0.4	0.5	0.6	0.6	0.7	0.7	0.7	0.8	1
y	1	-1	-1	-1	-1	-1	-1	-1	1	1

Boosting Round 2:

x	0.1	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3
y	1	1	1	1	1	1	1	1	1	1

Boosting Round 3:

x	0.2	0.2	0.4	0.4	0.4	0.4	0.5	0.6	0.6	0.7
y	1	1	-1	-1	-1	-1	-1	-1	-1	-1

- Summary:

Round	Split Point	Left Class	Right Class	alpha
1	0.75	-1	1	1.738
2	0.05	1	1	2.7784
3	0.3	1	-1	4.1195

10/7/2020 Intro to Data Mining, 2nd Edition

24

24

AdaBoost Example

- Weights

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
2	0.311	0.311	0.311	0.01	0.01	0.01	0.01	0.01	0.01	0.01
3	0.029	0.029	0.029	0.228	0.228	0.228	0.228	0.009	0.009	0.009

- Classification

Round	x=0.1	x=0.2	x=0.3	x=0.4	x=0.5	x=0.6	x=0.7	x=0.8	x=0.9	x=1.0
1	-1	-1	-1	-1	-1	-1	-1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	-1	-1	-1	-1	-1	-1	-1
Sum	5.16	5.16	5.16	-3.08	-3.08	-3.08	-3.08	0.397	0.397	0.397
Sign	1	1	1	-1	-1	-1	-1	1	1	1

25

10/7/2020 Intro to Data Mining, 2nd Edition

25

Random Forest Algorithm

- Construct an ensemble of decision trees by manipulating training set as well as features
 - Use bootstrap sample to train every decision tree (similar to Bagging)
 - Use the following tree induction algorithm:
 - At every internal node of decision tree, randomly sample p attributes for selecting split criterion
 - Repeat this procedure until all leaves are pure (unpruned tree)

10/7/2020 Intro to Data Mining, 2nd Edition

26

26

Characteristics of Random Forest

- Base classifiers are unpruned trees and hence are *unstable classifiers*
- Base classifiers are *decorrelated* (due to randomization in training set as well as features)
- Random forests reduce variance of unstable classifiers without negatively impacting the bias
- Selection of hyper-parameter p
 - Small value ensures lack of correlation
 - High value promotes strong base classifiers
 - Common default choices: \sqrt{d} , $\log_2(d + 1)$

Data Mining Classification: Alternative Techniques

Imbalanced Class Problem

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

1

Class Imbalance Problem

- Lots of classification problems where the classes are skewed (more records from one class than another)
 - Credit card fraud
 - Intrusion detection
 - Defective products in manufacturing assembly line
 - COVID-19 test results on a random sample

2

Challenges

- Evaluation measures such as accuracy are not well-suited for imbalanced class
- Detecting the rare class is like finding a needle in a haystack

Confusion Matrix

- 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)

Accuracy

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

- Most widely-used metric:

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

Problem with Accuracy

- Consider a 2-class problem
 - Number of Class NO examples = 990
 - Number of Class YES examples = 10

Problem with Accuracy

- Consider a 2-class problem
 - Number of Class NO examples = 990
 - Number of Class YES examples = 10

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	0	10
	Class>No	0	990

Problem with Accuracy

- Consider a 2-class problem
 - Number of Class NO examples = 990
 - Number of Class YES examples = 10
- If a model predicts everything to be class NO, accuracy is $990/1000 = 99\%$
 - This is misleading because the model does not detect any class YES example
 - Detecting the rare class is usually more interesting (e.g., frauds, intrusions, defects, etc)

Which model is better?

A

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes	0	10
	Class>No	0	990

B

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes	10	0
	Class>No	90	900

Which model is better?

A

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes	5	5
	Class>No	0	990

B

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes	10	0
	Class>No	90	900

Alternative Measures

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

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

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

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

Alternative Measures

		PREDICTED CLASS	
		Class=Yes	Class>No
ACTUAL CLASS	Class=Yes	10	0
	Class>No	10	980

$$\text{Precision (p)} = \frac{10}{10+10} = 0.5$$

$$\text{Recall (r)} = \frac{10}{10+0} = 1$$

$$\text{F - measure (F)} = \frac{2 * 1 * 0.5}{1 + 0.5} = 0.62$$

$$\text{Accuracy} = \frac{990}{1000} = 0.99$$

Alternative Measures

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	10	0
	Class>No	10	980

$$\text{Precision (p)} = \frac{10}{10+10} = 0.5$$

$$\text{Recall (r)} = \frac{10}{10+0} = 1$$

$$\text{F - measure (F)} = \frac{2*1*0.5}{1+0.5} = 0.62$$

$$\text{Accuracy} = \frac{990}{1000} = 0.99$$

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	1	9
	Class>No	0	990

$$\text{Precision (p)} = \frac{1}{1+0} = 1$$

$$\text{Recall (r)} = \frac{1}{1+9} = 0.1$$

$$\text{F - measure (F)} = \frac{2*0.1*1}{1+0.1} = 0.18$$

$$\text{Accuracy} = \frac{991}{1000} = 0.991$$

Alternative Measures

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	40	10
	Class>No	10	40

$$\text{Precision (p)} = 0.8$$

$$\text{Recall (r)} = 0.8$$

$$\text{F - measure (F)} = 0.8$$

$$\text{Accuracy} = 0.8$$

Alternative Measures

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
		Class=Yes	40
		Class>No	10

Precision (p) = 0.8
 Recall (r) = 0.8
 F - measure (F) = 0.8
 Accuracy = 0.8

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
		Class=Yes	40
		Class>No	1000

Precision (p) = ~ 0.04
 Recall (r) = 0.8
 F - measure (F) = ~ 0.08
 Accuracy = ~ 0.8

Measures of Classification Performance

		PREDICTED CLASS	
ACTUAL CLASS		Yes	No
		Yes	TP
		No	FP

α is the probability that we reject the null hypothesis when it is true. This is a Type I error or a false positive (FP).

β is the probability that we accept the null hypothesis when it is false. This is a Type II error or a false negative (FN).

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{ErrorRate} = 1 - \text{accuracy}$$

$$\text{Precision} = \text{Positive Predictive Value} = \frac{TP}{TP + FP}$$

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

$$\text{Specificity} = \text{TN Rate} = \frac{TN}{TN + FP}$$

$$\text{FP Rate} = \alpha = \frac{FP}{TN + FP} = 1 - \text{specificity}$$

$$\text{FN Rate} = \beta = \frac{FN}{FN + TP} = 1 - \text{sensitivity}$$

$$\text{Power} = \text{sensitivity} = 1 - \beta$$

Alternative Measures

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	40	10
	Class>No	10	40

Precision (p) = 0.8
 TPR = Recall (r) = 0.8
 FPR = 0.2
 F-measure (F) = 0.8
 Accuracy = 0.8

$$\frac{\text{TPR}}{\text{FPR}} = 4$$

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	40	10
	Class>No	1000	4000

Precision (p) = 0.038
 TPR = Recall (r) = 0.8
 FPR = 0.2
 F-measure (F) = 0.07
 Accuracy = 0.8

$$\frac{\text{TPR}}{\text{FPR}} = 4$$

Alternative Measures

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	10	40
	Class>No	10	40

Precision (p) = 0.5
 TPR = Recall (r) = 0.2
 FPR = 0.2
 F-measure = 0.28

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	25	25
	Class>No	25	25

Precision (p) = 0.5
 TPR = Recall (r) = 0.5
 FPR = 0.5
 F-measure = 0.5

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes	40	10
	Class>No	40	10

Precision (p) = 0.5
 TPR = Recall (r) = 0.8
 FPR = 0.8
 F-measure = 0.61

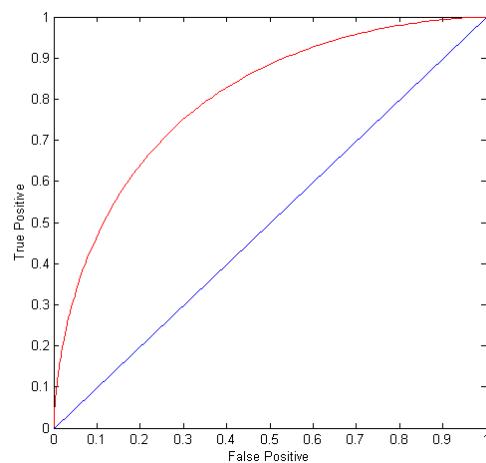
ROC (Receiver Operating Characteristic)

- A graphical approach for displaying trade-off between detection rate and false alarm rate
- Developed in 1950s for signal detection theory to analyze noisy signals
- ROC curve plots TPR against FPR
 - Performance of a model represented as a point in an ROC curve
 - Changing the threshold parameter of classifier changes the location of the point

ROC Curve

(TPR,FPR):

- (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

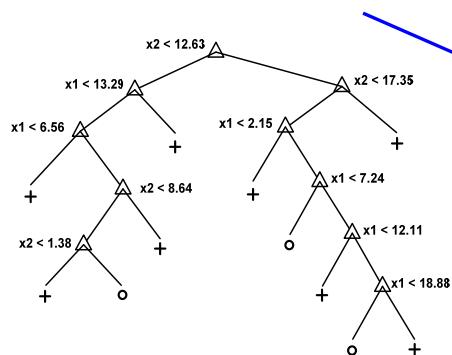


ROC (Receiver Operating Characteristic)

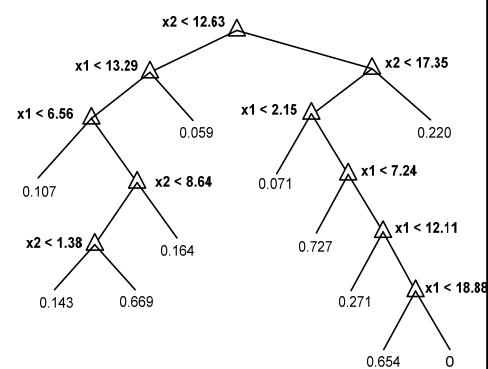
- To draw ROC curve, classifier must produce continuous-valued output
 - Outputs are used to rank test records, from the most likely positive class record to the least likely positive class record
- Many classifiers produce only discrete outputs (i.e., predicted class)
 - How to get continuous-valued outputs?
 - ◆ Decision trees, rule-based classifiers, neural networks, Bayesian classifiers, k-nearest neighbors, SVM

Example: Decision Trees

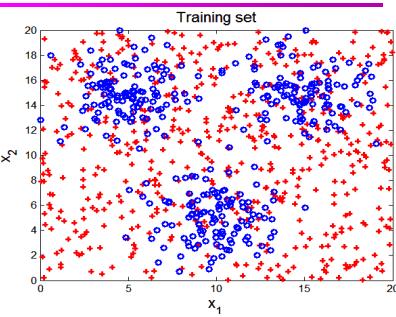
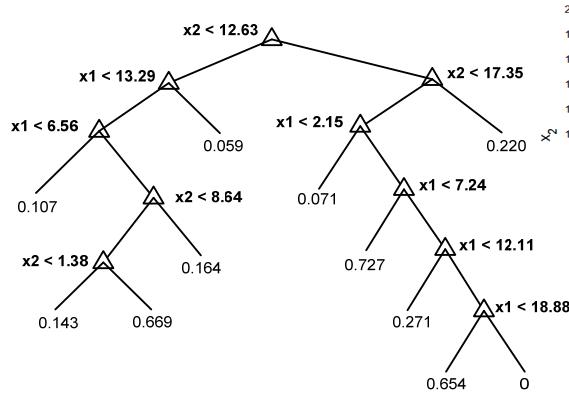
Decision Tree



Continuous-valued outputs



ROC Curve Example

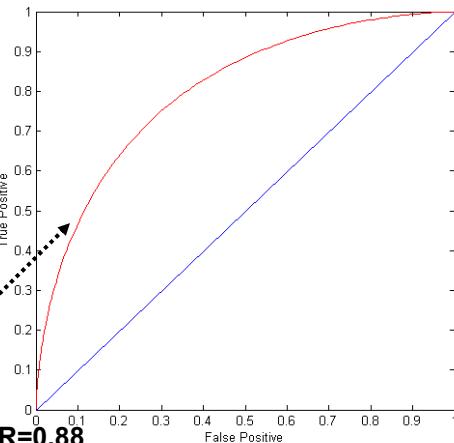
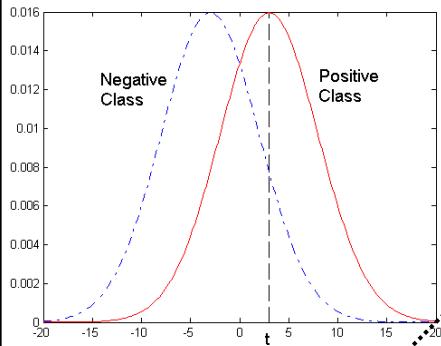


$\alpha = 0.3$		Predicted Class	
		Class o	Class +
Actual	Class o	645	209
Class	Class +	298	948

$\alpha = 0.7$		Predicted Class	
		Class o	Class +
Actual	Class o	181	673
Class	Class +	78	1168

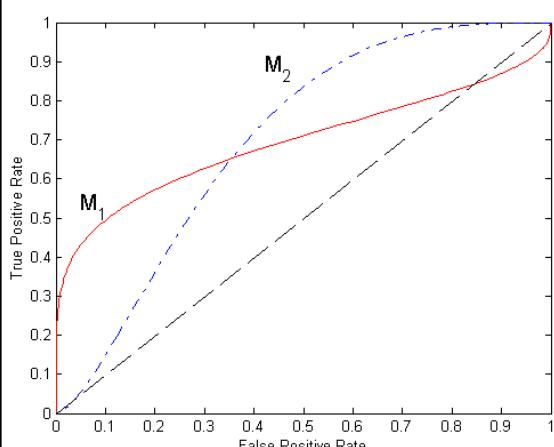
ROC Curve Example

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



TPR=0.5, FNR=0.5, FPR=0.12, TNR=0.88

Using ROC for Model Comparison



- No model consistently outperforms 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

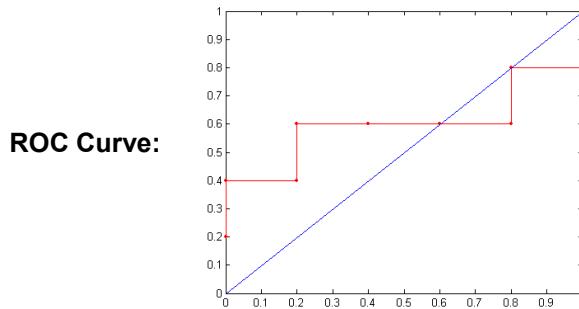
How to Construct an ROC curve

Instance	Score	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 a classifier that produces a continuous-valued score for each instance
 - The more likely it is for the instance to be in the + class, the higher the score
- Sort the instances in decreasing order according to the score
- Apply a threshold at each unique value of the score
- Count the number of TP, FP, TN, FN at each threshold
 - $TPR = TP/(TP+FN)$
 - $FPR = FP/(FP + TN)$

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	



Building Classifiers with Imbalanced Training Set

- Modify the distribution of training data so that rare class is well-represented in training set
 - Undersample the majority class
 - Oversample the rare class

Which model is better?

A

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes		
	Class>No		

B

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes		
	Class>No		

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes		
	Class>No		

		PREDICTED	
ACTUAL		Class=Yes	Class>No
	Class=Yes		
	Class>No		

	PREDICTED		
ACTUAL		Class=Yes	Class>No
	Class=Yes		
	Class>No		

	PREDICTED		
ACTUAL		Class=Yes	Class>No
	Class=Yes		
	Class>No		

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class>No
	Class=Yes		
	Class>No		

Data Mining Classification: Alternative Techniques

Lecture Notes for Chapter 4

Instance-Based Learning

Introduction to Data Mining , 2nd Edition

by

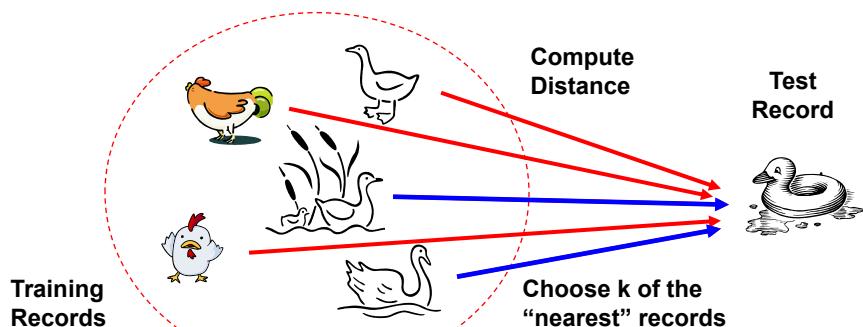
Tan, Steinbach, Karpatne, Kumar

1

Nearest Neighbor Classifiers

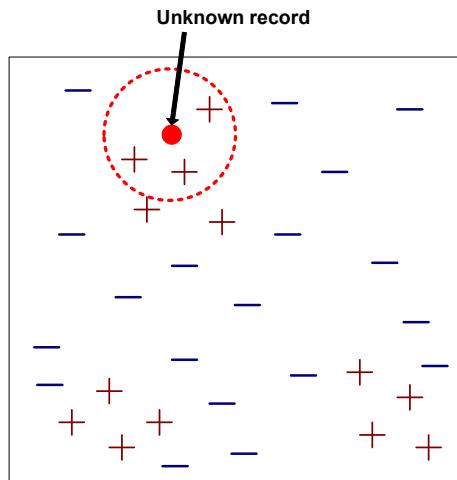
- Basic idea:

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



2

Nearest-Neighbor Classifiers



- Requires the following:
 - A set of labeled records
 - Proximity metric to compute distance/similarity between a pair of records (e.g., Euclidean distance)
 - The value of k , the number of nearest neighbors to retrieve
 - A method for using class labels of K nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

9/30/2020

Introduction to Data Mining, 2nd Edition

3

How to Determine the class label of a Test Sample?

- Take the majority vote of class labels among the k -nearest neighbors
- Weight the vote according to distance
 - weight factor, $w = 1/d^2$

9/30/2020

Introduction to Data Mining, 2nd Edition

4

4

Choice of proximity measure matters

- For documents, cosine is better than correlation or Euclidean

1 1 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1 1

vs

0 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 0

Euclidean distance = 1.4142 for both pairs, but the cosine similarity measure has different values for these pairs.

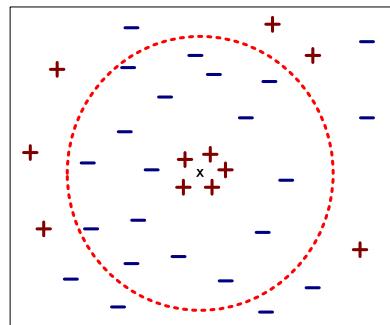
9/30/2020

Introduction to Data Mining, 2nd Edition

5

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



9/30/2020

Introduction to Data Mining, 2nd Edition

6

6

Nearest Neighbor Classification...

- Data preprocessing is often required

- 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

- Time series are often standardized to have 0 means a standard deviation of 1

9/30/2020

Introduction to Data Mining, 2nd Edition

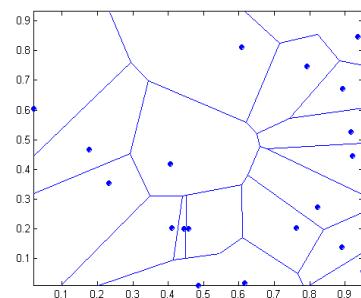
7

7

Nearest-neighbor classifiers

- Nearest neighbor classifiers are local classifiers
- They can produce decision boundaries of arbitrary shapes.

1-nn decision boundary is a Voronoi Diagram



9/30/2020

Introduction to Data Mining, 2nd Edition

8

8

Nearest Neighbor Classification...

- How to handle missing values in training and test sets?
 - Proximity computations normally require the presence of all attributes
 - Some approaches use the subset of attributes present in two instances
 - ◆ This may not produce good results since it effectively uses different proximity measures for each pair of instances
 - ◆ Thus, proximities are not comparable

9/30/2020

Introduction to Data Mining, 2nd Edition

9

Nearest Neighbor Classification...

- Handling irrelevant and redundant attributes
 - Irrelevant attributes add noise to the proximity measure
 - Redundant attributes bias the proximity measure towards certain attributes
 - Can use variable selection or dimensionality reduction to address irrelevant and redundant attributes

9/30/2020

Introduction to Data Mining, 2nd Edition

10

10

Improving KNN Efficiency

- Avoid having to compute distance to all objects in the training set
 - Multi-dimensional access methods (k-d trees)
 - Fast approximate similarity search
 - Locality Sensitive Hashing (LSH)
- Condensing
 - Determine a smaller set of objects that give the same performance
- Editing
 - Remove objects to improve efficiency

Data Mining Classification: Alternative Techniques

Bayesian Classifiers

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

1

Bayes Classifier

- A probabilistic framework for solving classification problems
- Conditional Probability:
$$P(Y | X) = \frac{P(X, Y)}{P(X)}$$

$$P(X | Y) = \frac{P(X, Y)}{P(Y)}$$
- Bayes theorem:

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

2

Using Bayes Theorem for Classification

- Consider each attribute and class label as random variables
- Given a record with attributes (X_1, X_2, \dots, X_d)
 - Goal is to predict class Y
 - Specifically, we want to find the value of Y that maximizes $P(Y|X_1, X_2, \dots, X_d)$
- Can we estimate $P(Y|X_1, X_2, \dots, X_d)$ directly 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

09/28/2020

Introduction to Data Mining, 2nd Edition

3

3

Using Bayes Theorem for Classification

- Approach:
 - compute posterior probability $P(Y | X_1, X_2, \dots, X_d)$ using the Bayes theorem
 - *Maximum a-posteriori*: Choose Y that maximizes $P(Y | X_1, X_2, \dots, X_d)$
 - Equivalent to choosing value of Y that maximizes $P(X_1, X_2, \dots, X_d | Y) P(Y)$
- How to estimate $P(X_1, X_2, \dots, X_d | Y)$?

$$P(Y | X_1 X_2 \dots X_n) = \frac{P(X_1 X_2 \dots X_d | Y) P(Y)}{P(X_1 X_2 \dots X_d)}$$

09/28/2020

Introduction to Data Mining, 2nd Edition

4

4

Example Data

Given a Test Record:

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

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

- Can we estimate

$$P(\text{Evade} = \text{Yes} | X) \text{ and } P(\text{Evade} = \text{No} | X)?$$

In the following we will replace

Evade = Yes by Yes, and

Evade = No by No

09/28/2020

Introduction to Data Mining, 2nd Edition

5

Example Data

Given a Test Record:

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

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

Using Bayes Theorem:

$$\square P(\text{Yes} | X) = \frac{P(X | \text{Yes})P(\text{Yes})}{P(X)}$$

$$\square P(\text{No} | X) = \frac{P(X | \text{No})P(\text{No})}{P(X)}$$

- How to estimate $P(X | \text{Yes})$ and $P(X | \text{No})$?

09/28/2020

Introduction to Data Mining, 2nd Edition

6

6

Conditional Independence

- X and Y are conditionally independent given Z if $P(X|YZ) = P(X|Z)$
- Example: Arm length and reading skills
 - Young child has shorter arm length and limited reading skills, compared to adults
 - If age is fixed, no apparent relationship between arm length and reading skills
 - Arm length and reading skills are conditionally independent given age

09/28/2020

Introduction to Data Mining, 2nd Edition

7

Naïve Bayes Classifier

- Assume independence among attributes X_i when class is given:
 - $P(X_1, X_2, \dots, X_d | Y_j) = P(X_1 | Y_j) P(X_2 | Y_j) \dots P(X_d | Y_j)$
 - Now we can estimate $P(X_i | Y_j)$ for all X_i and Y_j combinations from the training data
 - New point is classified to Y_j if $P(Y_j) \prod P(X_i | Y_j)$ is maximal.

09/28/2020

Introduction to Data Mining, 2nd Edition

8

8

Naïve Bayes on Example Data

Given a Test Record:

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

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

$$P(X | \text{Yes}) =$$

$$\begin{aligned} P(\text{Refund} = \text{No} | \text{Yes}) \times \\ P(\text{Divorced} | \text{Yes}) \times \\ P(\text{Income} = 120K | \text{Yes}) \end{aligned}$$

$$P(X | \text{No}) =$$

$$\begin{aligned} P(\text{Refund} = \text{No} | \text{No}) \times \\ P(\text{Divorced} | \text{No}) \times \\ P(\text{Income} = 120K | \text{No}) \end{aligned}$$

09/28/2020

Introduction to Data Mining, 2nd Edition

9

Estimate Probabilities from Data

- $P(y)$ = fraction of instances of class y

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

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

- For categorical attributes:

$$P(X_i = c | y) = n_c / n$$

— where $|X_i = c|$ is number of instances having attribute value $X_i = c$ and belonging to class y

— Examples:

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

09/28/2020

Introduction to Data Mining, 2nd Edition

10

10

Estimate Probabilities from Data

- For continuous attributes:
 - **Discretization:** Partition the range into bins:
 - ◆ Replace continuous value with bin value
 - Attribute changed from continuous to ordinal
 - **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, use it to estimate the conditional probability $P(X_i|Y)$

09/28/2020

Introduction to Data Mining, 2nd Edition

11

11

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:

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

- One for each (X_i, Y_j) pair

- For (Income, Class=No):

- If Class=No
 - ◆ sample mean = 110
 - ◆ sample variance = 2975

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

09/28/2020

Introduction to Data Mining, 2nd Edition

12

12

Example of Naïve Bayes Classifier

Given a Test Record:

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

Naïve Bayes Classifier:

$P(\text{Refund} = \text{Yes} | \text{No}) = 3/7$
 $P(\text{Refund} = \text{No} | \text{No}) = 4/7$
 $P(\text{Refund} = \text{Yes} | \text{Yes}) = 0$
 $P(\text{Refund} = \text{No} | \text{Yes}) = 1$
 $P(\text{Marital Status} = \text{Single} | \text{No}) = 2/7$
 $P(\text{Marital Status} = \text{Divorced} | \text{No}) = 1/7$
 $P(\text{Marital Status} = \text{Married} | \text{No}) = 4/7$
 $P(\text{Marital Status} = \text{Single} | \text{Yes}) = 2/3$
 $P(\text{Marital Status} = \text{Divorced} | \text{Yes}) = 1/3$
 $P(\text{Marital Status} = \text{Married} | \text{Yes}) = 0$

For Taxable Income:

If class = No: sample mean = 110
sample variance = 2975
If class = Yes: sample mean = 90
sample variance = 25

- $P(X | \text{No}) = P(\text{Refund} = \text{No} | \text{No}) \times P(\text{Divorced} | \text{No}) \times P(\text{Income} = 120\text{K} | \text{No}) = 4/7 \times 1/7 \times 0.0072 = 0.0006$
- $P(X | \text{Yes}) = P(\text{Refund} = \text{No} | \text{Yes}) \times P(\text{Divorced} | \text{Yes}) \times P(\text{Income} = 120\text{K} | \text{Yes}) = 1 \times 1/3 \times 1.2 \times 10^{-9} = 4 \times 10^{-10}$

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

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

09/28/2020

Introduction to Data Mining, 2nd Edition

13

13

Naïve Bayes Classifier can make decisions with partial information about attributes in the test record

Even in absence of information about any attributes, we can use Apriori Probabilities of Class Variable:

Naïve Bayes Classifier:

$P(\text{Refund} = \text{Yes} | \text{No}) = 3/7$
 $P(\text{Refund} = \text{No} | \text{No}) = 4/7$
 $P(\text{Refund} = \text{Yes} | \text{Yes}) = 0$
 $P(\text{Refund} = \text{No} | \text{Yes}) = 1$
 $P(\text{Marital Status} = \text{Single} | \text{No}) = 2/7$
 $P(\text{Marital Status} = \text{Divorced} | \text{No}) = 1/7$
 $P(\text{Marital Status} = \text{Married} | \text{No}) = 4/7$
 $P(\text{Marital Status} = \text{Single} | \text{Yes}) = 2/3$
 $P(\text{Marital Status} = \text{Divorced} | \text{Yes}) = 1/3$
 $P(\text{Marital Status} = \text{Married} | \text{Yes}) = 0$

For Taxable Income:

If class = No: sample mean = 110
sample variance = 2975
If class = Yes: sample mean = 90
sample variance = 25

$$P(\text{Yes}) = 3/10$$

$$P(\text{No}) = 7/10$$

If we only know that marital status is Divorced, then:

$$P(\text{Yes} | \text{Divorced}) = 1/3 \times 3/10 / P(\text{Divorced})$$
$$P(\text{No} | \text{Divorced}) = 1/7 \times 7/10 / P(\text{Divorced})$$

If we also know that Refund = No, then

$$P(\text{Yes} | \text{Refund} = \text{No}, \text{Divorced}) = 1 \times 1/3 \times 3/10 / P(\text{Divorced, Refund} = \text{No})$$
$$P(\text{No} | \text{Refund} = \text{No}, \text{Divorced}) = 4/7 \times 1/7 \times 7/10 / P(\text{Divorced, Refund} = \text{No})$$

If we also know that Taxable Income = 120, then

$$P(\text{Yes} | \text{Refund} = \text{No}, \text{Divorced, Income} = 120) = 1.2 \times 10^{-9} \times 1 \times 1/3 \times 3/10 / P(\text{Divorced, Refund} = \text{No, Income} = 120)$$
$$P(\text{No} | \text{Refund} = \text{No}, \text{Divorced, Income} = 120) = 0.0072 \times 4/7 \times 1/7 \times 7/10 / P(\text{Divorced, Refund} = \text{No, Income} = 120)$$

09/28/2020

Introduction to Data Mining, 2nd Edition

14

14

Issues with Naïve Bayes Classifier

Given a Test Record:

$X = (\text{Married})$

Naïve Bayes Classifier:

$P(\text{Refund} = \text{Yes} | \text{No}) = 3/7$
 $P(\text{Refund} = \text{No} | \text{No}) = 4/7$
 $P(\text{Refund} = \text{Yes} | \text{Yes}) = 0$
 $P(\text{Refund} = \text{No} | \text{Yes}) = 1$
 $P(\text{Marital Status} = \text{Single} | \text{No}) = 2/7$
 $P(\text{Marital Status} = \text{Divorced} | \text{No}) = 1/7$
 $P(\text{Marital Status} = \text{Married} | \text{No}) = 4/7$
 $P(\text{Marital Status} = \text{Single} | \text{Yes}) = 2/3$
 $P(\text{Marital Status} = \text{Divorced} | \text{Yes}) = 1/3$
 $P(\text{Marital Status} = \text{Married} | \text{Yes}) = 0$

$$P(\text{Yes}) = 3/10$$

$$P(\text{No}) = 7/10$$

$$P(\text{Yes} | \text{Married}) = 0 \times 3/10 / P(\text{Married})$$

$$P(\text{No} | \text{Married}) = 4/7 \times 7/10 / P(\text{Married})$$

For Taxable Income:

If class = No: sample mean = 110
sample variance = 2975
If class = Yes: sample mean = 90
sample variance = 25

09/28/2020

Introduction to Data Mining, 2nd Edition

15

15

Issues with Naïve Bayes Classifier

Consider the table with $Tid = 7$ deleted

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				
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Naïve Bayes Classifier:

$P(\text{Refund} = \text{Yes} | \text{No}) = 2/6$
 $P(\text{Refund} = \text{No} | \text{No}) = 4/6$
 $\rightarrow P(\text{Refund} = \text{Yes} | \text{Yes}) = 0$
 $P(\text{Refund} = \text{No} | \text{Yes}) = 1$
 $P(\text{Marital Status} = \text{Single} | \text{No}) = 2/6$
 $P(\text{Marital Status} = \text{Divorced} | \text{No}) = 0$
 $P(\text{Marital Status} = \text{Married} | \text{No}) = 4/6$
 $P(\text{Marital Status} = \text{Single} | \text{Yes}) = 2/3$
 $P(\text{Marital Status} = \text{Divorced} | \text{Yes}) = 1/3$
 $P(\text{Marital Status} = \text{Married} | \text{Yes}) = 0/3$
For Taxable Income:
If class = No: sample mean = 91
sample variance = 685
If class = Yes: sample mean = 90
sample variance = 25

Given $X = (\text{Refund} = \text{Yes}, \text{Divorced}, 120\text{K})$

$$P(X | \text{No}) = 2/6 \times 0 \times 0.0083 = 0$$

$$P(X | \text{Yes}) = 0 \times 1/3 \times 1.2 \times 10^{-9} = 0$$

Naïve Bayes will not be able to
classify X as Yes or No!

09/28/2020

Introduction to Data Mining, 2nd Edition

16

16

Issues with Naïve Bayes Classifier

- If one of the conditional probabilities is zero, then the entire expression becomes zero
- Need to use other estimates of conditional probabilities than simple fractions
- Probability estimation:

$$\text{original: } P(X_i = c|y) = \frac{n_c}{n}$$

$$\text{Laplace Estimate: } P(X_i = c|y) = \frac{n_c + 1}{n + v}$$

$$m - \text{estimate: } P(X_i = c|y) = \frac{n_c + mp}{n + m}$$

n : number of training instances belonging to class y

n_c : number of instances with $X_i = c$ and $Y = y$

v : total number of attribute values that X_i can take

p : initial estimate of $(P(X_i = c|y))$ known apriori

m : hyper-parameter for our confidence in p

09/28/2020

Introduction to Data Mining, 2nd Edition

17

17

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

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

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

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

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

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

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

09/28/2020

Introduction to Data Mining, 2nd Edition

18

18

Naïve Bayes (Summary)

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Redundant and correlated attributes will violate class conditional assumption
 - Use other techniques such as Bayesian Belief Networks (BBN)

09/28/2020

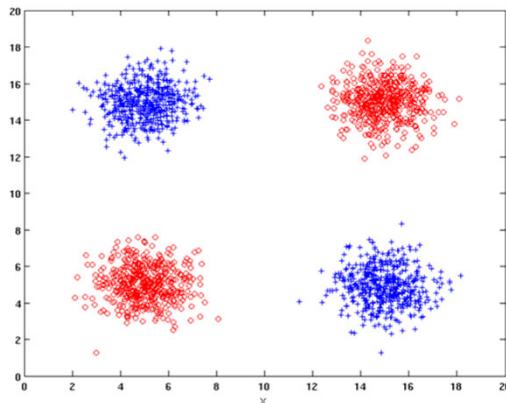
Introduction to Data Mining, 2nd Edition

19

19

Naïve Bayes

- How does Naïve Bayes perform on the following dataset?



Conditional independence of attributes is violated

09/28/2020

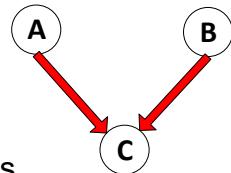
Introduction to Data Mining, 2nd Edition

20

20

Bayesian Belief Networks

- Provides graphical representation of probabilistic relationships among a set of random variables
- Consists of:
 - A directed acyclic graph (dag)
 - ◆ Node corresponds to a variable
 - ◆ Arc corresponds to dependence relationship between a pair of variables
 - A probability table associating each node to its immediate parent

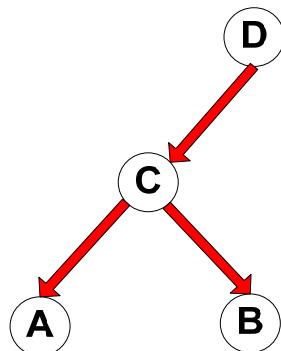


09/28/2020

Introduction to Data Mining, 2nd Edition

21

Conditional Independence



D is parent of C

A is child of C

B is descendant of D

D is ancestor of A

- A node in a Bayesian network is conditionally independent of all of its nondescendants, if its parents are known

09/28/2020

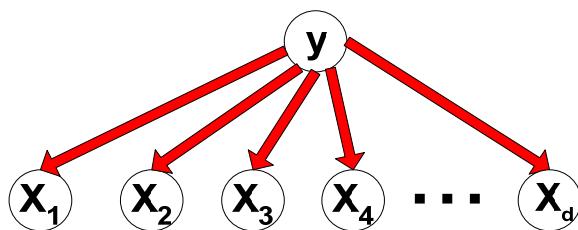
Introduction to Data Mining, 2nd Edition

22

22

Conditional Independence

- Naïve Bayes assumption:



09/28/2020

Introduction to Data Mining, 2nd Edition

23

23

Probability Tables

- If X does not have any parents, table contains prior probability $P(X)$
- If X has only one parent (Y), table contains conditional probability $P(X|Y)$
- If X has multiple parents (Y_1, Y_2, \dots, Y_k), table contains conditional probability $P(X|Y_1, Y_2, \dots, Y_k)$



09/28/2020

Introduction to Data Mining, 2nd Edition

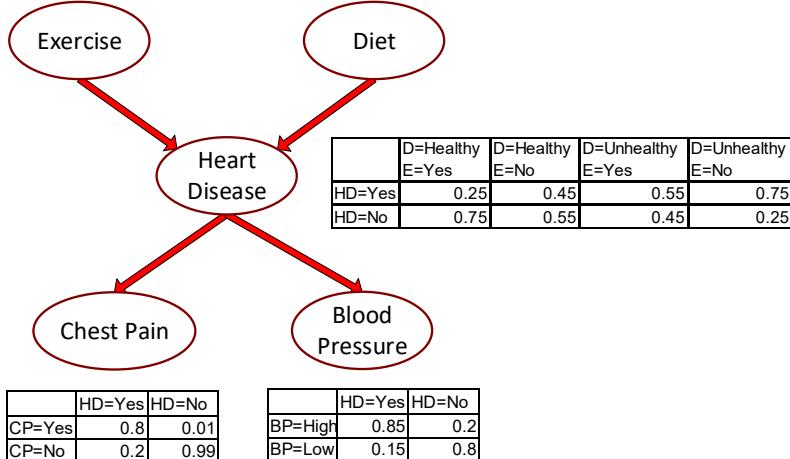
24

24

Example of Bayesian Belief Network

Exercise=Yes	0.7
Exercise>No	0.3

Diet=Healthy	0.25
Diet=Unhealthy	0.75



09/28/2020

Introduction to Data Mining, 2nd Edition

25

25

Example of Inferencing using BBN

- Given: $X = (E=No, D=Yes, CP=Yes, BP=High)$
 - Compute $P(HD|E,D,CP,BP)$?
- $P(HD=Yes| E=No,D=Yes) = 0.55$
 $P(CP=Yes| HD=Yes) = 0.8$
 $P(BP=High| HD=Yes) = 0.85$
 - $P(HD=Yes|E=No,D=Yes,CP=Yes,BP=High) \propto 0.55 \times 0.8 \times 0.85 = 0.374$
- $P(HD=No| E=No,D=Yes) = 0.45$
 $P(CP=Yes| HD=No) = 0.01$
 $P(BP=High| HD=No) = 0.2$
 - $P(HD=No|E=No,D=Yes,CP=Yes,BP=High) \propto 0.45 \times 0.01 \times 0.2 = 0.0009$

{

Classify X as Yes

09/28/2020

Introduction to Data Mining, 2nd Edition

26

26

Data Mining Classification: Alternative Techniques

Lecture Notes for Chapter 4

Rule-Based

Introduction to Data Mining , 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar

1

Rule-Based Classifier

- Classify records by using a collection of “if...then...” rules
- Rule: $(Condition) \rightarrow y$
 - where
 - ◆ *Condition* is a conjunction of tests on attributes
 - ◆ *y* is the class label
 - Examples of classification rules:
 - ◆ (Blood Type=Warm) \wedge (Lay Eggs=Yes) \rightarrow Birds
 - ◆ (Taxable Income < 50K) \wedge (Refund=Yes) \rightarrow Evade=No

2

Rule-based Classifier (Example)

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

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

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

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

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

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

Application of Rule-Based Classifier

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

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

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

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

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

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

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

The rule R1 covers a hawk => Bird

The rule R3 covers the grizzly bear => Mammal

Rule Coverage and Accuracy

- Coverage of a rule:
 - Fraction of records that satisfy the antecedent of a rule
- Accuracy of a rule:
 - Fraction of records that satisfy the antecedent that also satisfy the 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 Sets: Strategy 1

- 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

Characteristics of Rule Sets: Strategy 2

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

Ordered Rule Set

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

R1: (Give Birth = no) \wedge (Can Fly = yes) \rightarrow Birds
R2: (Give Birth = no) \wedge (Live in Water = yes) \rightarrow Fishes
R3: (Give Birth = yes) \wedge (Blood Type = warm) \rightarrow Mammals
R4: (Give Birth = no) \wedge (Can Fly = no) \rightarrow Reptiles
R5: (Live in Water = sometimes) \rightarrow Amphibians

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

Rule Ordering Schemes

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

Rule-based Ordering

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

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

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

Class-based Ordering

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

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

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

Building Classification Rules

- Direct Method:

- ◆ Extract rules directly from data
- ◆ Examples: RIPPER, CN2, Holte's 1R

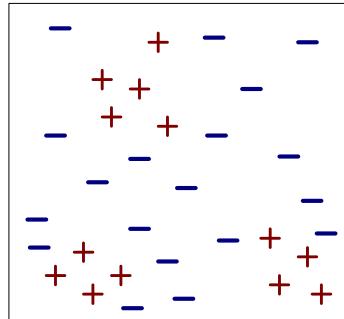
- Indirect Method:

- ◆ Extract rules from other classification models (e.g. decision trees, neural networks, etc).
- ◆ Examples: C4.5rules

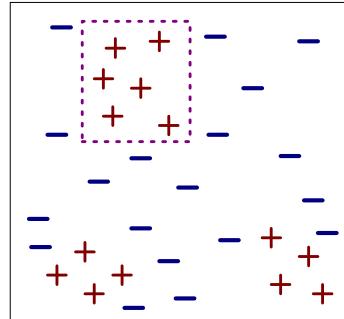
Direct Method: Sequential Covering

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

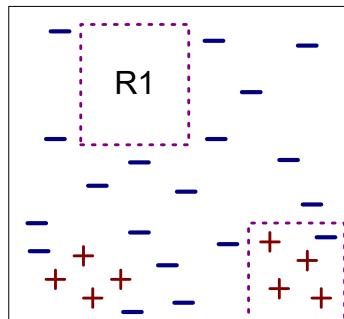


(i) Original Data

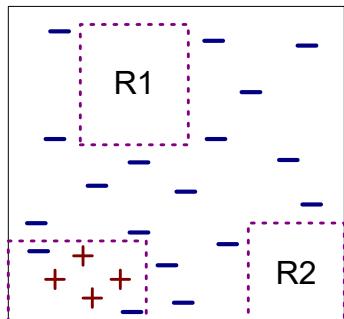


(ii) Step 1

Example of Sequential Covering...



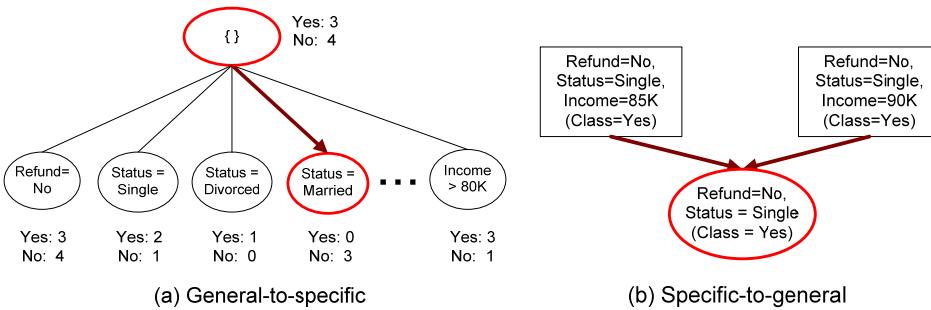
(iii) Step 2



(iv) Step 3

Rule Growing

- Two common strategies



Rule Evaluation

- Foil's Information Gain

FOIL: First Order Inductive Learner – an early rule-based learning algorithm

- R0: {} => class (initial rule)
- R1: {A} => class (rule after adding conjunct)
- $$Gain(R_0, R_1) = p_1 \times [\log_2 \left(\frac{p_1}{p_1 + n_1} \right) - \log_2 \left(\frac{p_0}{p_0 + n_0} \right)]$$
- p_0 : number of positive instances covered by R0
- n_0 : number of negative instances covered by R0
- p_1 : number of positive instances covered by R1
- n_1 : number of negative instances covered by R1

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

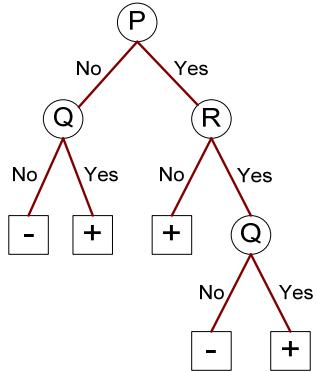
Direct Method: RIPPER

- 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 alternative rules has lower pessimistic error rate
 - Repeat until we can no longer improve generalization error

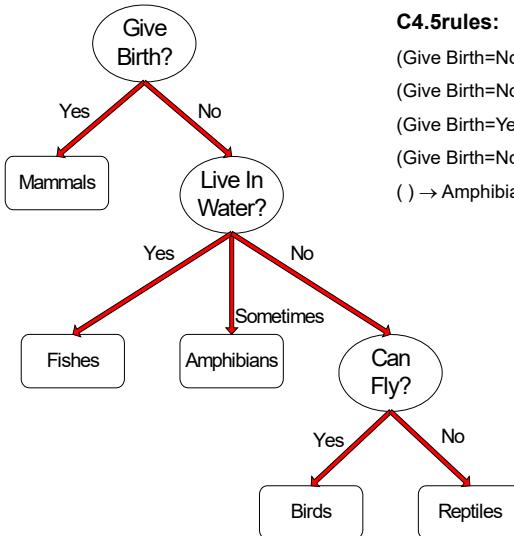
Indirect Method: C4.5rules

- 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:

$(\text{Give Birth}=\text{No}, \text{Can Fly}=\text{Yes}) \rightarrow \text{Birds}$
 $(\text{Give Birth}=\text{No}, \text{Live in Water}=\text{Yes}) \rightarrow \text{Fishes}$
 $(\text{Give Birth}=\text{Yes}) \rightarrow \text{Mammals}$
 $(\text{Give Birth}=\text{No}, \text{Can Fly}=\text{No}, \text{Live in Water}=\text{No}) \rightarrow \text{Reptiles}$
 $() \rightarrow \text{Amphibians}$

RIPPER:

$(\text{Live in Water}=\text{Yes}) \rightarrow \text{Fishes}$
 $(\text{Have Legs}=\text{No}) \rightarrow \text{Reptiles}$
 $(\text{Give Birth}=\text{No}, \text{Can Fly}=\text{No}, \text{Live In Water}=\text{No}) \rightarrow \text{Reptiles}$
 $(\text{Can Fly}=\text{Yes}, \text{Give Birth}=\text{No}) \rightarrow \text{Birds}$
 $() \rightarrow \text{Mammals}$

C4.5 versus C4.5rules versus RIPPER

C4.5 and C4.5rules:

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

RIPPER:

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

Advantages of Rule-Based Classifiers

- Has characteristics quite similar to decision trees
 - As highly expressive as decision trees
 - Easy to interpret (if rules are ordered by class)
 - Performance comparable to decision trees
 - ◆ Can handle redundant and irrelevant attributes
 - ◆ Variable interaction can cause issues (e.g., X-OR problem)
- Better suited for handling imbalanced classes
- Harder to handle missing values in the test set

Data Mining

Support Vector Machines

Introduction to Data Mining, 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar

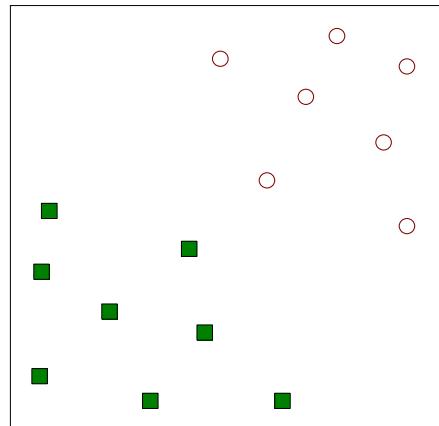
02/17/2020

Introduction to Data Mining, 2nd Edition

1

1

Support Vector Machines



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

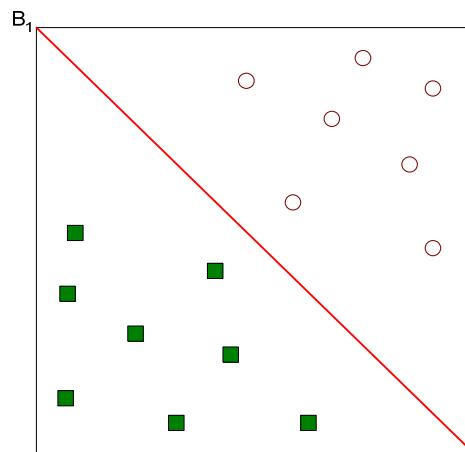
02/17/2020

Introduction to Data Mining, 2nd Edition

2

2

Support Vector Machines



- One Possible Solution

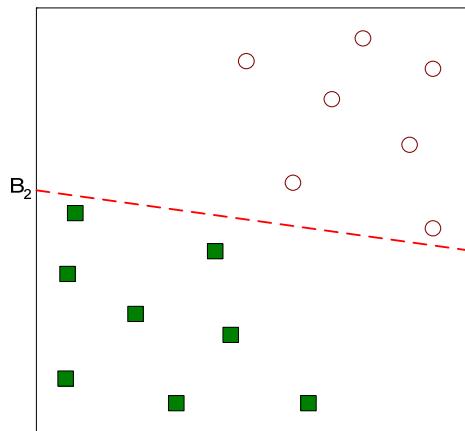
02/17/2020

Introduction to Data Mining, 2nd Edition

3

3

Support Vector Machines



- Another possible solution

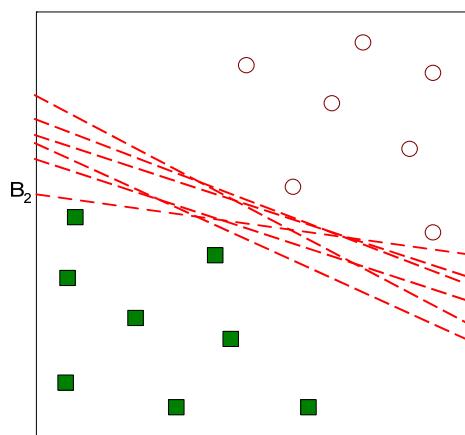
02/17/2020

Introduction to Data Mining, 2nd Edition

4

4

Support Vector Machines



- Other possible solutions

02/17/2020

Introduction to Data Mining, 2nd Edition

5

5

Support Vector Machines

- Which one is better? B1 or B2?
- How do you define better?

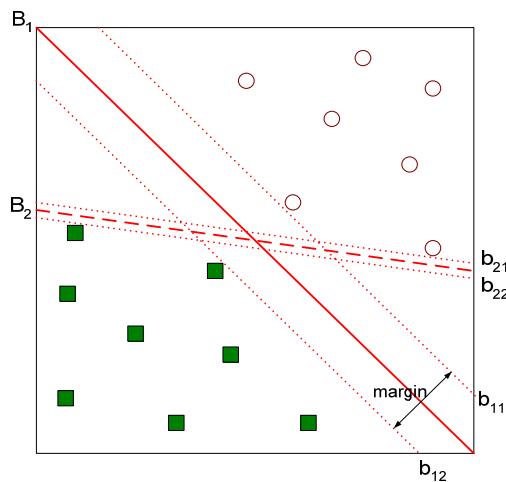
02/17/2020

Introduction to Data Mining, 2nd Edition

6

6

Support Vector Machines



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

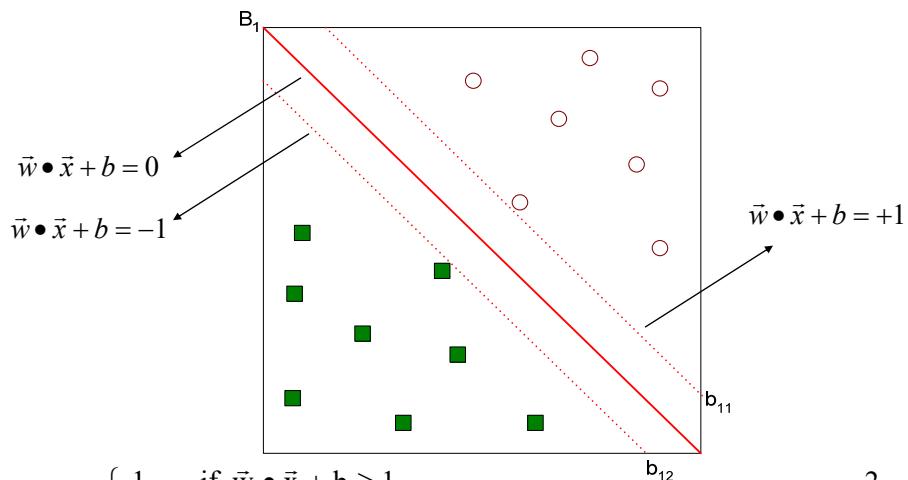
02/17/2020

Introduction to Data Mining, 2nd Edition

7

7

Support Vector Machines



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

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

02/17/2020

Introduction to Data Mining, 2nd Edition

8

8

Linear SVM

- Linear model:

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

- Learning the model is equivalent to determining the values of \vec{w} and b
 - How to find \vec{w} and b from training data?

02/17/2020

Introduction to Data Mining, 2nd Edition

9

9

Learning Linear SVM

- Objective is to maximize: Margin = $\frac{2}{\|\vec{w}\|}$
 - Which is equivalent to minimizing: $L(\vec{w}) = \frac{\|\vec{w}\|^2}{2}$
 - Subject to the following constraints:

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

or $y_i(\vec{w} \bullet \vec{x}_i + b) \geq 1, \quad i = 1, 2, \dots, N$

- ◆ This is a constrained optimization problem
 - Solve it using Lagrange multiplier method

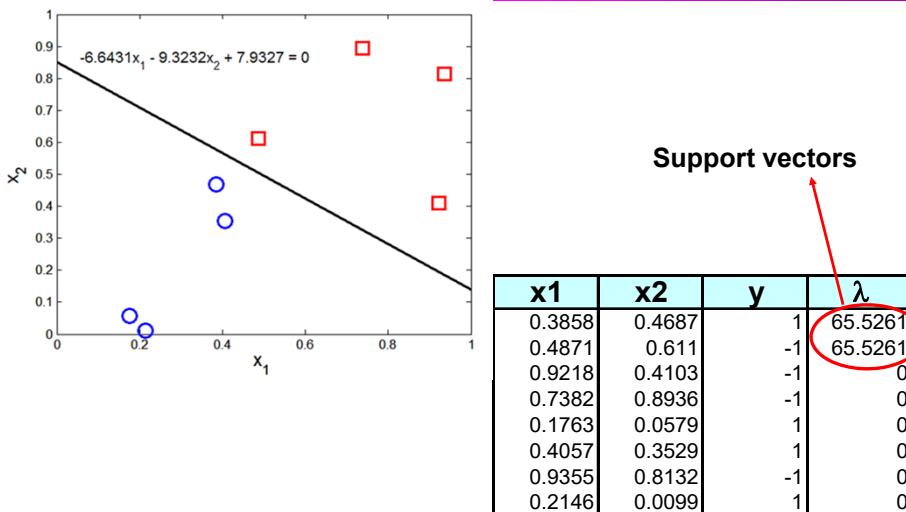
02/17/2020

Introduction to Data Mining, 2nd Edition

10

10

Example of Linear SVM



02/17/2020

Introduction to Data Mining, 2nd Edition

11

11

Learning Linear SVM

- Decision boundary depends only on support vectors
 - If you have data set with same support vectors, decision boundary will not change
 - How to classify using SVM once w and b are found? Given a test record, x_i

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

02/17/2020

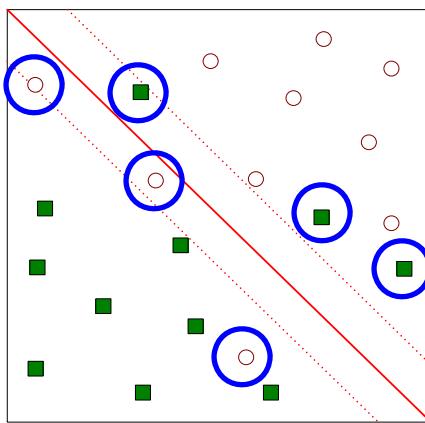
Introduction to Data Mining, 2nd Edition

12

12

Support Vector Machines

- What if the problem is not linearly separable?



02/17/2020

Introduction to Data Mining, 2nd Edition

13

13

Support Vector Machines

- What if the problem is not linearly separable?

– Introduce slack variables

◆ Need to minimize:

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

◆ Subject to:

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

◆ If k is 1 or 2, this leads to similar objective function as linear SVM but with different constraints (see textbook)

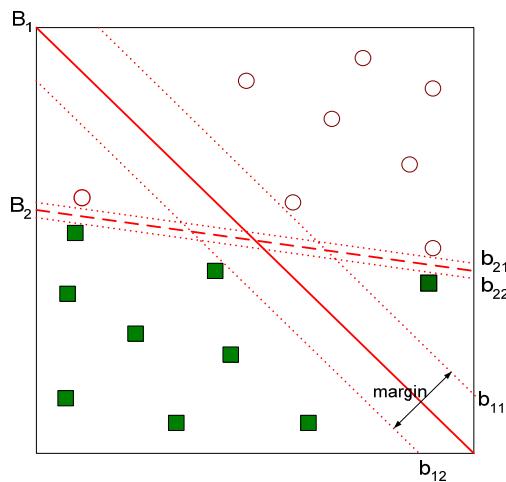
02/17/2020

Introduction to Data Mining, 2nd Edition

14

14

Support Vector Machines



- Find the hyperplane that optimizes both factors

02/17/2020

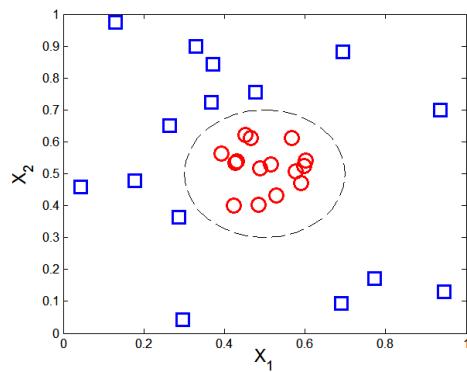
Introduction to Data Mining, 2nd Edition

15

15

Nonlinear Support Vector Machines

- What if decision boundary is not linear?



$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2 \\ -1 & \text{otherwise} \end{cases}$$

02/17/2020

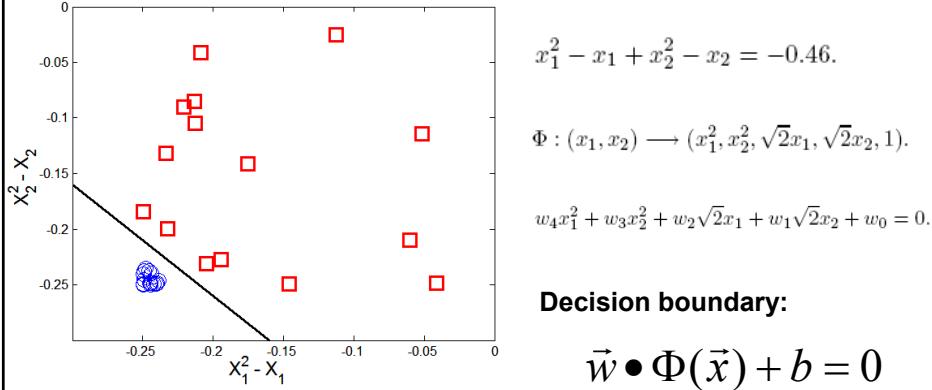
Introduction to Data Mining, 2nd Edition

16

16

Nonlinear Support Vector Machines

- Transform data into higher dimensional space



02/17/2020

Introduction to Data Mining, 2nd Edition

17

17

Learning Nonlinear SVM

- Optimization problem:

$$\min_w \frac{\|w\|^2}{2}$$

subject to $y_i(w \cdot \Phi(x_i) + b) \geq 1, \forall \{(x_i, y_i)\}$

- Which leads to the same set of equations (but involve $\Phi(x)$ instead of x)

$$L_D = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \quad w = \sum_i \lambda_i y_i \Phi(x_i)$$

$$\lambda_i \{y_i (\sum_j \lambda_j y_j \Phi(x_j) \cdot \Phi(x_i) + b) - 1\} = 0,$$

$$f(z) = sign(w \cdot \Phi(z) + b) = sign(\sum_{i=1}^n \lambda_i y_i \Phi(x_i) \cdot \Phi(z) + b).$$

02/17/2020 Introduction to Data Mining, 2nd Edition 18

18

Learning NonLinear SVM

- Issues:
 - What type of mapping function Φ should be used?
 - How to do the computation in high dimensional space?
 - ◆ Most computations involve dot product $\Phi(x_i) \bullet \Phi(x_j)$
 - ◆ Curse of dimensionality?

02/17/2020

Introduction to Data Mining, 2nd Edition

19

19

Learning Nonlinear SVM

- Kernel Trick:
 - $\Phi(x_i) \bullet \Phi(x_j) = K(x_i, x_j)$
 - $K(x_i, x_j)$ is a kernel function (expressed in terms of the coordinates in the original space)
 - ◆ Examples:

$$K(x, y) = (x \cdot y + 1)^p$$

$$K(x, y) = e^{-\|x-y\|^2/(2\sigma^2)}$$

$$K(x, y) = \tanh(kx \cdot y - \delta)$$

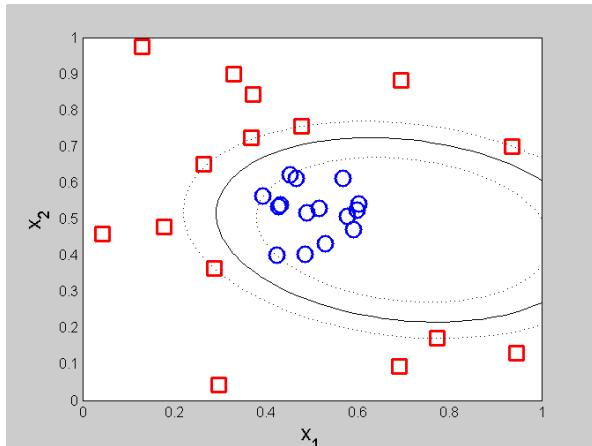
02/17/2020

Introduction to Data Mining, 2nd Edition

20

20

Example of Nonlinear SVM



SVM with polynomial degree 2 kernel

02/17/2020

Introduction to Data Mining, 2nd Edition

21

21

Learning Nonlinear SVM

- Advantages of using kernel:
 - Don't have to know the mapping function Φ
 - Computing dot product $\Phi(x_i) \cdot \Phi(x_j)$ in the original space avoids curse of dimensionality
- Not all functions can be kernels
 - Must make sure there is a corresponding Φ in some high-dimensional space
 - Mercer's theorem (see textbook)

02/17/2020

Introduction to Data Mining, 2nd Edition

22

22

Characteristics of SVM

- The learning problem is formulated as a convex optimization problem
 - Efficient algorithms are available to find the global minima
 - Many of the other methods use greedy approaches and find locally optimal solutions
 - High computational complexity for building the model
- Robust to noise
- Overfitting is handled by maximizing the margin of the decision boundary,
- SVM can handle irrelevant and redundant better than many other techniques
- The user needs to provide the type of kernel function and cost function
- Difficult to handle missing values
- What about categorical variables?

Data Mining

Chapter 5 Association Analysis: Basic Concepts

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

10/26/2020

Introduction to Data Mining, 2nd Edition

1

Association Rule Mining

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

Market-Basket transactions

TID	Items
1	Bread, Milk
2	Bread, Diaper, 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!

10/26/2020

Introduction to Data Mining, 2nd Edition

2

2

Definition: Frequent Itemset

● Itemset

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

● Support count (σ)

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

● Support

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

● Frequent Itemset

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

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

10/26/2020

Introduction to Data Mining, 2nd Edition

3

Definition: Association Rule

● Association Rule

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

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

● Rule Evaluation Metrics

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

Example:

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

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

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

10/26/2020

Introduction to Data Mining, 2nd Edition

4

4

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!

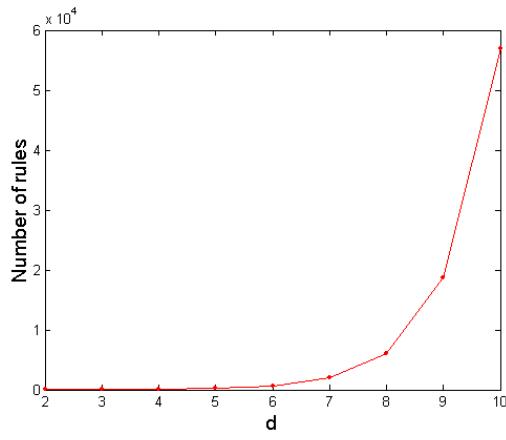
10/26/2020

Introduction to Data Mining, 2nd Edition

5

Computational Complexity

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



$$\begin{aligned} R &= \sum_{k=1}^{d-1} \left[\binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right] \\ &= 3^d - 2^{d+1} + 1 \end{aligned}$$

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

10/26/2020

Introduction to Data Mining, 2nd Edition

6

6

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

10/26/2020

Introduction to Data Mining, 2nd Edition

7

7

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

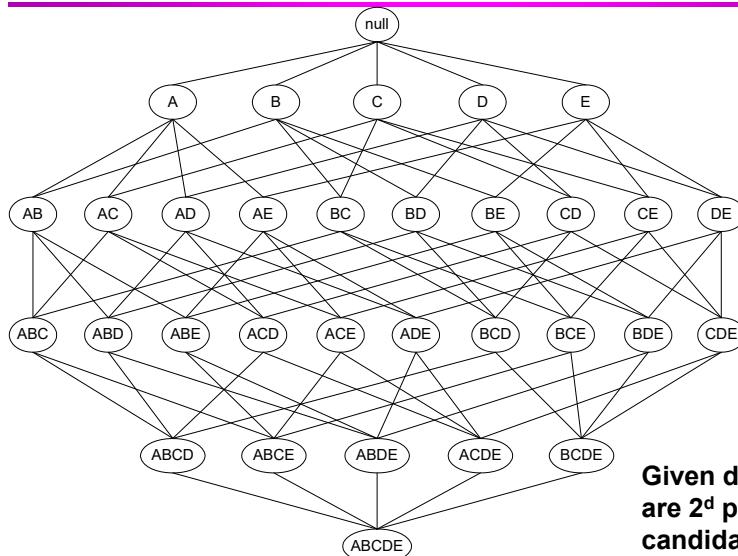
10/26/2020

Introduction to Data Mining, 2nd Edition

8

8

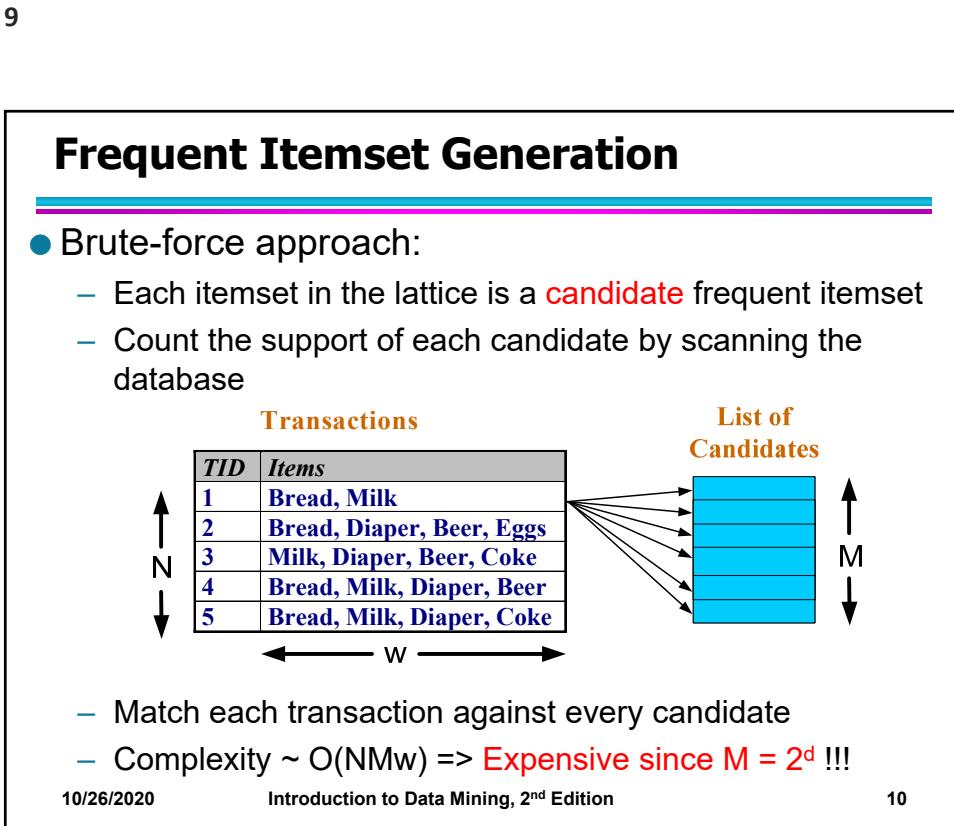
Frequent Itemset Generation



10/26/2020

Introduction to Data Mining, 2nd Edition

9



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

10/26/2020

Introduction to Data Mining, 2nd Edition

11

11

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

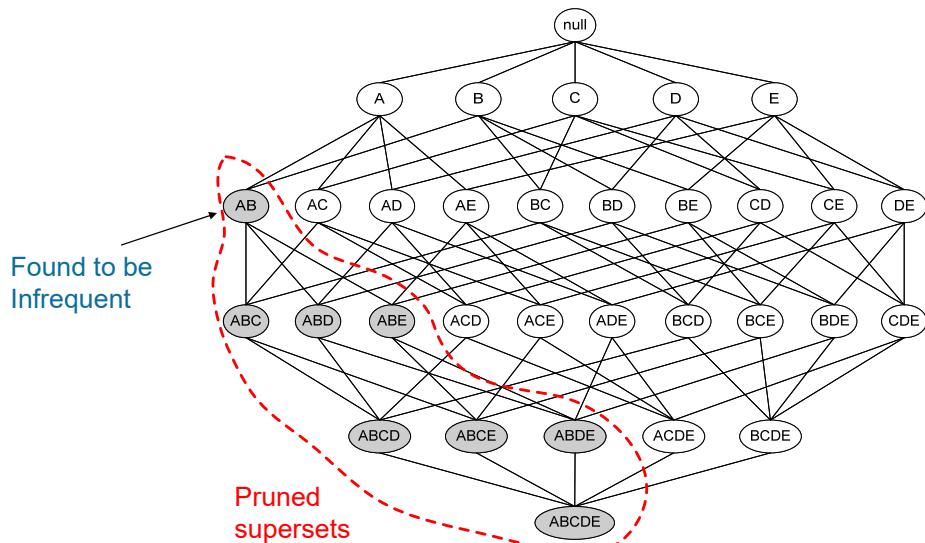
10/26/2020

Introduction to Data Mining, 2nd Edition

12

12

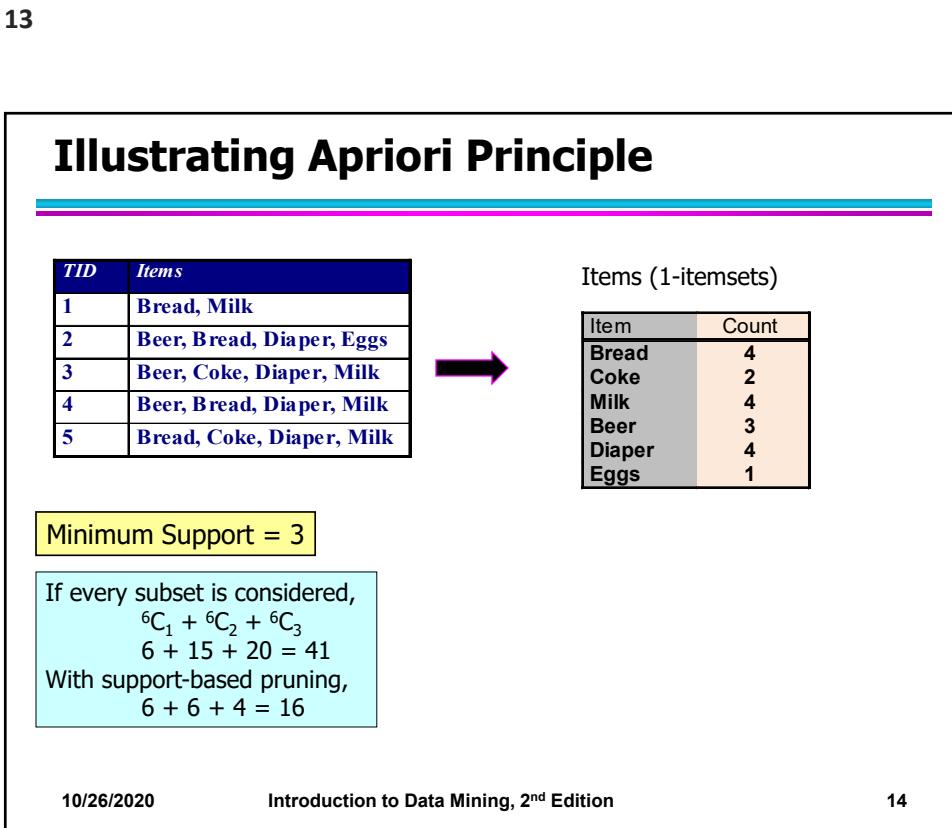
Illustrating Apriori Principle



10/26/2020

Introduction to Data Mining, 2nd Edition

13



Illustrating Apriori Principle

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



Items (1-itemsets)

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

Minimum Support = 3

If every subset is considered,

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

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

10/26/2020

Introduction to Data Mining, 2nd Edition

15

15

Illustrating Apriori Principle

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

Items (1-itemsets)



Itemset
{Bread, Milk}
{Bread, Beer}
{Bread, Diaper}
{Beer, Milk}
{Diaper, Milk}
{Beer, Diaper}

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,

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

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

10/26/2020

Introduction to Data Mining, 2nd Edition

16

16

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

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

Minimum Support = 3

If every subset is considered,

$$^6C_1 + ^6C_2 + ^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

10/26/2020

Introduction to Data Mining, 2nd Edition

17

17

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,

$$^6C_1 + ^6C_2 + ^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

Triplets (3-itemsets)

Itemset
{ Beer, Diaper, Milk }
{ Beer, Bread, Diaper }
{ Bread, Diaper, Milk }
{ Beer, Bread, Milk }

10/26/2020

Introduction to Data Mining, 2nd Edition

18

18

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,

$$^6C_1 + ^6C_2 + ^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

Itemset	Count
{ Beer, Diaper, Milk }	2
{ Beer, Bread, Diaper }	2
{ Bread, Diaper, Milk }	2
{ Beer, Bread, Milk }	1

Triplets (3-itemsets)

10/26/2020

Introduction to Data Mining, 2nd Edition

19

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,

$$^6C_1 + ^6C_2 + ^6C_3$$

$$6 + 15 + 20 = 41$$

With support-based pruning,

$$6 + 6 + 4 = 16$$

$$\textcolor{red}{6 + 6 + 1 = 13}$$

Itemset	Count
{ Beer, Diaper, Milk }	2
{ Beer, Bread, Diaper }	2
{ Bread, Diaper, Milk }	2
{ Beer, Bread, Milk }	1

Triplets (3-itemsets)

10/26/2020

Introduction to Data Mining, 2nd Edition

20

20

Apriori Algorithm

- F_k : frequent k-itemsets
- L_k : candidate k-itemsets

● Algorithm

- Let $k=1$
- Generate $F_1 = \{\text{frequent 1-itemsets}\}$
- Repeat until F_k is empty
 - ◆ **Candidate Generation:** Generate L_{k+1} from F_k
 - ◆ **Candidate Pruning:** Prune candidate itemsets in L_{k+1} containing subsets of length k that are infrequent
 - ◆ **Support Counting:** Count the support of each candidate in L_{k+1} by scanning the DB
 - ◆ **Candidate Elimination:** Eliminate candidates in L_{k+1} that are infrequent, leaving only those that are frequent => F_{k+1}

10/26/2020

Introduction to Data Mining, 2nd Edition

21

21

Candidate Generation: Brute-force method

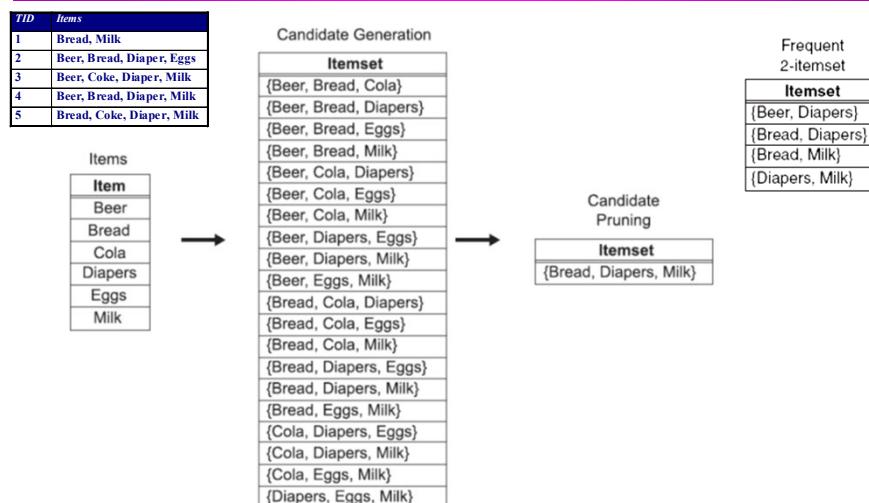


Figure 5.6. A brute-force method for generating candidate 3-itemsets.

10/26/2020

Introduction to Data Mining, 2nd Edition

22

22

Candidate Generation: Merge F_{k-1} and F₁ itemsets

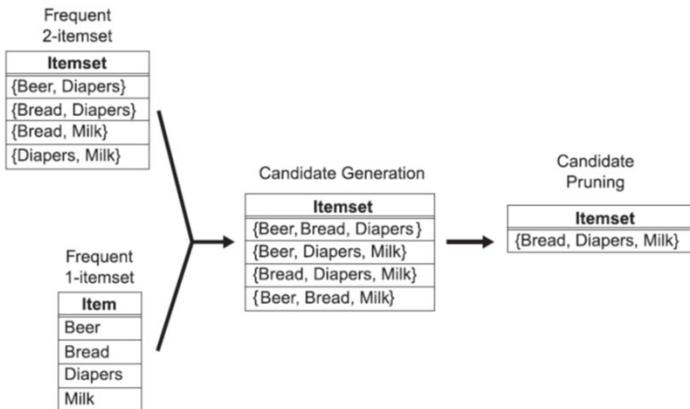


Figure 5.7. Generating and pruning candidate k -itemsets by merging a frequent $(k - 1)$ -itemset with a frequent item. Note that some of the candidates are unnecessary because their subsets are infrequent.

10/26/2020

Introduction to Data Mining, 2nd Edition

23

23

Candidate Generation: $F_{k-1} \times F_{k-1}$ Method

- Merge two frequent $(k-1)$ -itemsets if their first $(k-2)$ items are identical
- $F_3 = \{\text{ABC, ABD, ABE, ACD, BCD, BDE, CDE}\}$
 - Merge(ABC, ABD) = ABCD
 - Merge(ABC, ABE) = ABCE
 - Merge(ABD, ABE) = ABDE
 - Do not merge(ABD, ACD) because they share only prefix of length 1 instead of length 2

10/26/2020

Introduction to Data Mining, 2nd Edition

24

24

Candidate Pruning

- Let $F_3 = \{ABC, ABD, ABE, ACD, BCD, BDE, CDE\}$ be the set of frequent 3-itemsets
- $L_4 = \{ABCD, ABCE, ABDE\}$ is the set of candidate 4-itemsets generated (from previous slide)
- Candidate pruning
 - Prune ABCE because ACE and BCE are infrequent
 - Prune ABDE because ADE is infrequent
- After candidate pruning: $L_4 = \{ABCD\}$

10/26/2020

Introduction to Data Mining, 2nd Edition

25

25

Candidate Generation: $F_{k-1} \times F_{k-1}$ Method

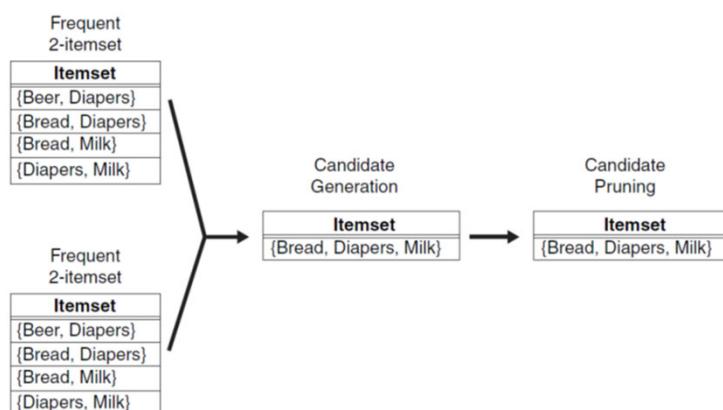


Figure 5.8. Generating and pruning candidate k -itemsets by merging pairs of frequent $(k-1)$ -itemsets.

10/26/2020

Introduction to Data Mining, 2nd Edition

26

26

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,
 ${}^6C_1 + {}^6C_2 + {}^6C_3$
 $6 + 15 + 20 = 41$
 With support-based pruning,
 $6 + 6 + 1 = 13$

Use of $F_{k-1} \times F_{k-1}$ method for candidate generation results in only one 3-itemset. This is eliminated after the support counting step.

Itemset	Count
{Bread, Diaper, Milk}	2

Triplets (3-itemsets)

10/26/2020

Introduction to Data Mining, 2nd Edition

27

27

Alternate $F_{k-1} \times F_{k-1}$ Method

- Merge two frequent (k-1)-itemsets if the last (k-2) items of the first one is identical to the first (k-2) items of the second.
- $F_3 = \{\text{ABC, ABD, ABE, ACD, BCD, BDE, CDE}\}$
 - Merge(ABC, BCD) = ABCD
 - Merge(ABD, BDE) = ABDE
 - Merge(ACD, CDE) = ACDE
 - Merge(BCD, CDE) = BCDE

10/26/2020

Introduction to Data Mining, 2nd Edition

28

28

Candidate Pruning for Alternate $F_{k-1} \times F_{k-1}$ Method

- Let $F_3 = \{ABC, ABD, ABE, ACD, BCD, BDE, CDE\}$ be the set of frequent 3-itemsets
- $L_4 = \{ABCD, ABDE, ACDE, BCDE\}$ is the set of candidate 4-itemsets generated (from previous slide)
- Candidate pruning
 - Prune ABDE because ADE is infrequent
 - Prune ACDE because ACE and ADE are infrequent
 - Prune BCDE because BCE
- After candidate pruning: $L_4 = \{ABCD\}$

10/26/2020

Introduction to Data Mining, 2nd Edition

29

Support Counting of Candidate Itemsets

- Scan the database of transactions to determine the support of each candidate itemset
 - Must match every candidate itemset against every transaction, which is an expensive operation

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

Itemset
{ Beer, Diaper, Milk }
{ Beer, Bread, Diaper }
{ Bread, Diaper, Milk }
{ Beer, Bread, Milk }

10/26/2020

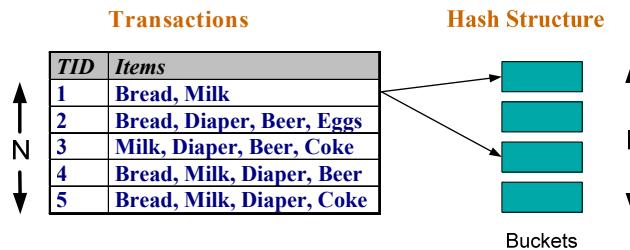
Introduction to Data Mining, 2nd Edition

30

30

Support Counting of Candidate Itemsets

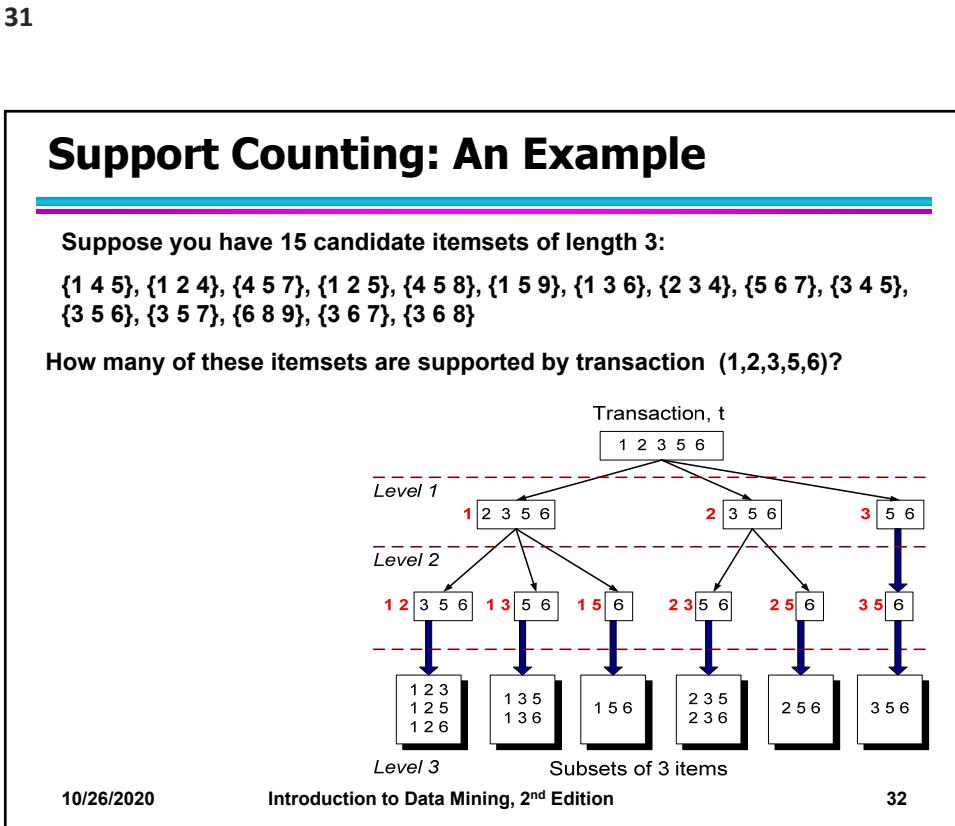
- To reduce number of comparisons, store the candidate itemsets in a hash structure
 - Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets



10/26/2020

Introduction to Data Mining, 2nd Edition

31



10/26/2020

Introduction to Data Mining, 2nd Edition

32

32

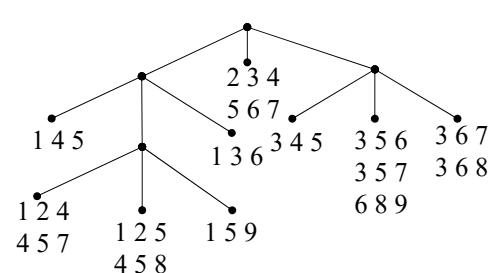
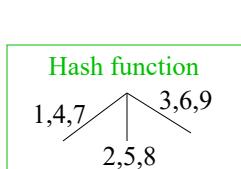
Support Counting Using a Hash Tree

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:

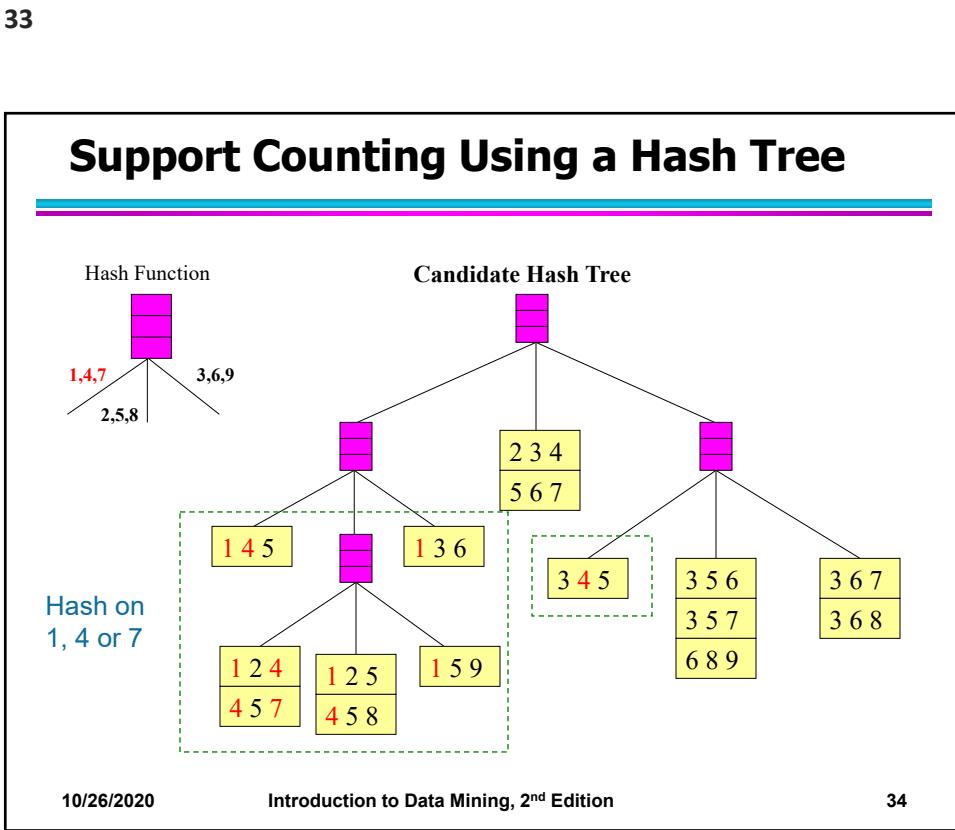
- Hash function
- Max leaf size: max number of itemsets stored in a leaf node (if number of candidate itemsets exceeds max leaf size, split the node)



10/26/2020

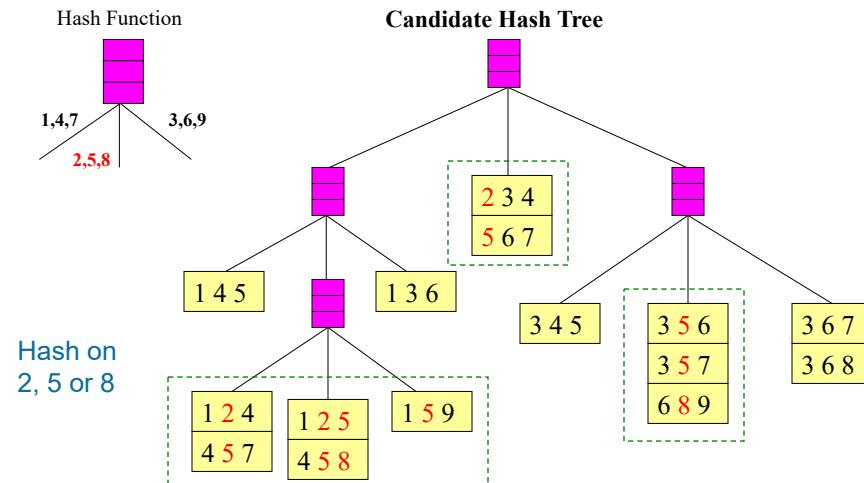
Introduction to Data Mining, 2nd Edition

33



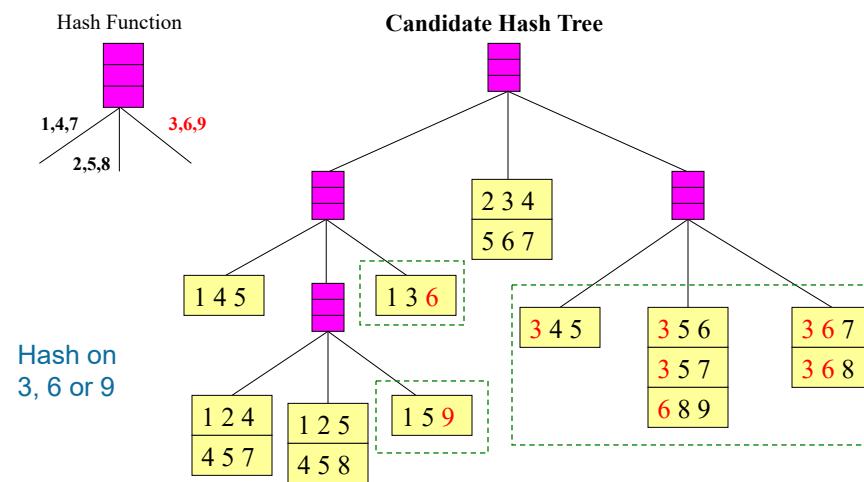
34

Support Counting Using a Hash Tree



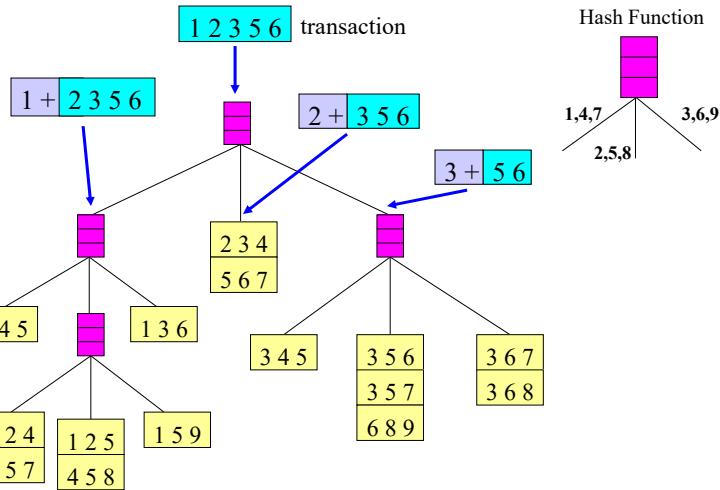
35

Support Counting Using a Hash Tree



36

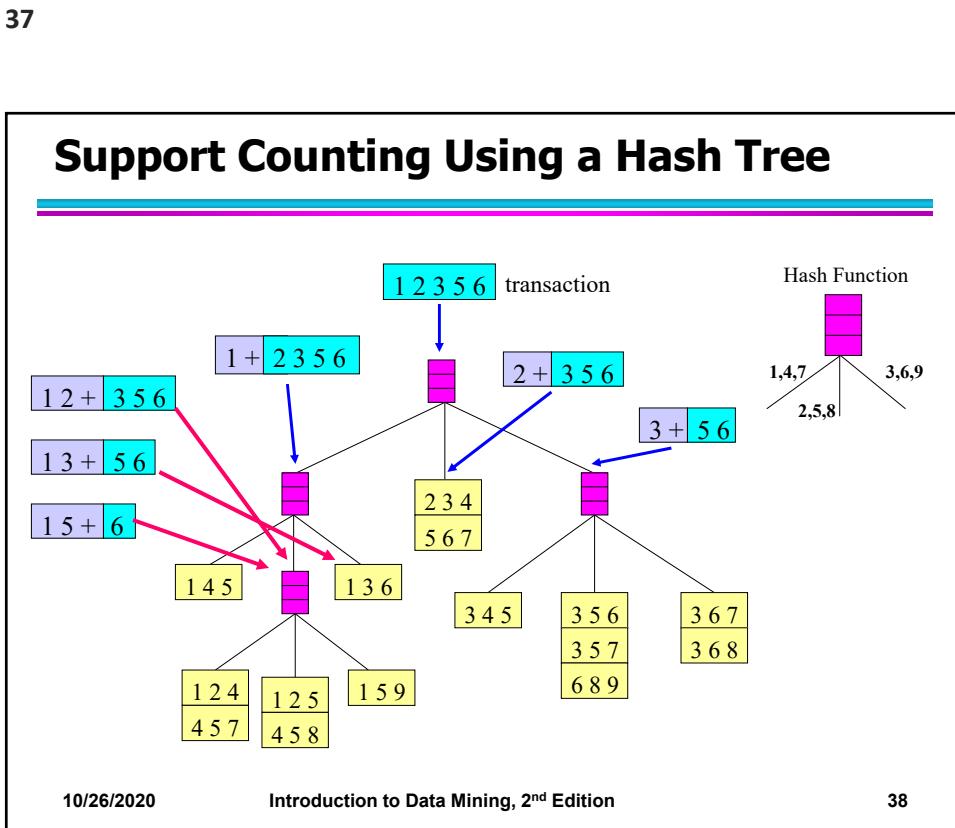
Support Counting Using a Hash Tree



10/26/2020

Introduction to Data Mining, 2nd Edition

37



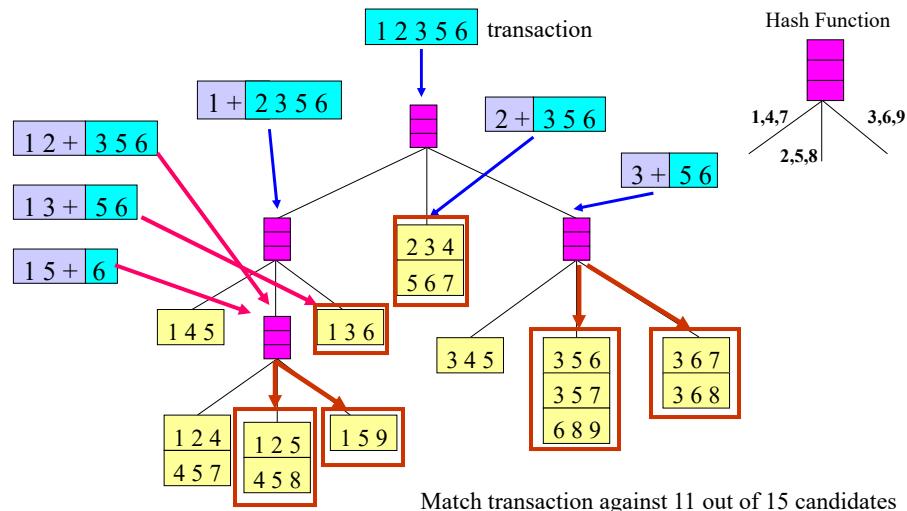
10/26/2020

Introduction to Data Mining, 2nd Edition

38

38

Support Counting Using a Hash Tree



10/26/2020

Introduction to Data Mining, 2nd Edition

39

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 $\emptyset \rightarrow L$)

10/26/2020

Introduction to Data Mining, 2nd Edition

40

40

Rule Generation

- 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., Suppose $\{A,B,C,D\}$ is a frequent 4-itemset:
 $c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)$
 - Confidence is anti-monotone w.r.t. number of items on the RHS of the rule

10/26/2020

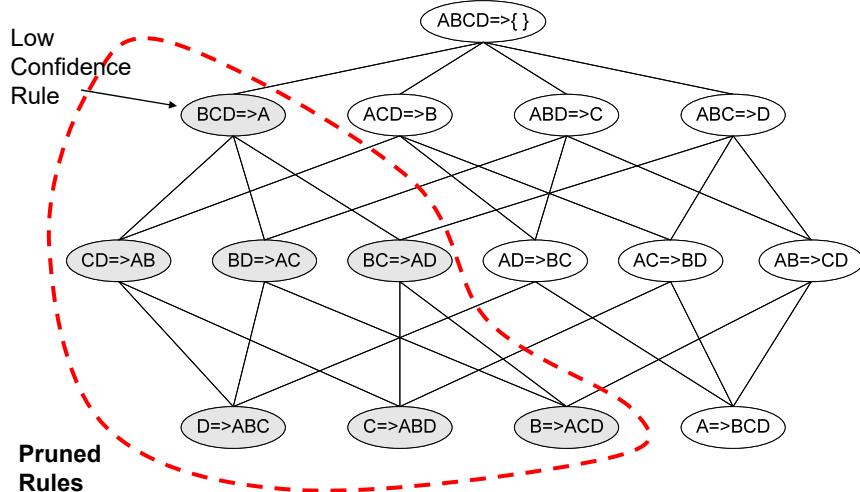
Introduction to Data Mining, 2nd Edition

41

41

Rule Generation for Apriori Algorithm

Lattice of rules



10/26/2020

Introduction to Data Mining, 2nd Edition

42

42

Association Analysis: Basic Concepts and Algorithms

Algorithms and Complexity

10/26/2020

Introduction to Data Mining, 2nd Edition

43

43

Factors Affecting Complexity of Apriori

- Choice of minimum support threshold
- Dimensionality (number of items) of the data set
- Size of database
- Average transaction width

10/26/2020

Introduction to Data Mining, 2nd Edition

44

44

Factors Affecting Complexity of Apriori

- 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
 -

- Size of database
 -
- Average transaction width
 -

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

10/26/2020

Introduction to Data Mining, 2nd Edition

45

45

Impact of Support Based Pruning

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



Items (1-itemsets)

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

Minimum Support = 3

If every subset is considered,
 ${}^6C_1 + {}^6C_2 + {}^6C_3$
 $6 + 15 + 20 = 41$
With support-based pruning,
 $6 + 6 + 4 = 16$

Minimum Support = 2

If every subset is considered,
 ${}^6C_1 + {}^6C_2 + {}^6C_3 + {}^6C_4$
 $6 + 15 + 20 + 15 = 56$

10/26/2020

Introduction to Data Mining, 2nd Edition

46

46

Factors Affecting Complexity of Apriori

- 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 itemsets
 - if number of frequent itemsets also increases, both computation and I/O costs may also increase
- Size of database
- Average transaction width
 -

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

10/26/2020

Introduction to Data Mining, 2nd Edition

47

47

Factors Affecting Complexity of Apriori

- 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 itemsets
 - if number of frequent itemsets also increases, both computation and I/O costs may also increase
- Size of database
 - run time of algorithm increases with number of transactions
- Average transaction width

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

10/26/2020

Introduction to Data Mining, 2nd Edit

48

Factors Affecting Complexity of Apriori

- 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 itemsets
 - if number of frequent itemsets also increases, both computation and I/O costs may also increase
- Size of database
 - run time of algorithm increases with number of transactions
- Average transaction width
 - transaction width increases the max length of frequent itemsets
 - number of subsets in a transaction increases with its width, increasing computation time for support counting

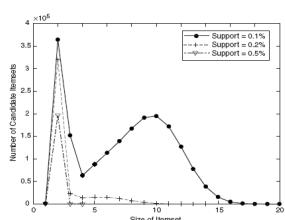
10/26/2020

Introduction to Data Mining, 2nd Edition

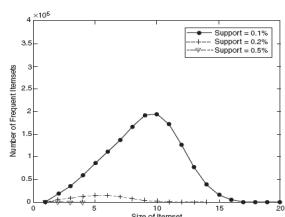
49

49

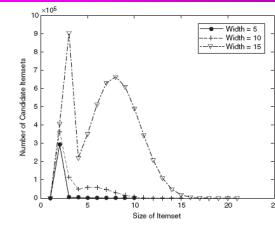
Factors Affecting Complexity of Apriori



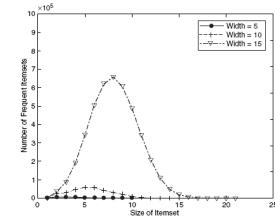
(a) Number of candidate itemsets.



(b) Number of frequent itemsets.



(a) Number of candidate itemsets.



(b) Number of frequent itemsets.

Figure 6.13. Effect of support threshold on the number of candidate and frequent itemsets.

Figure 6.14. Effect of average transaction width on the number of candidate and frequent itemsets.

10/26/2020

Introduction to Data Mining, 2nd Edition

50

50

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	0	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	0	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	0	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	0	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	0	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	1	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	1	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	1	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	1	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	1	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

10/26/2020

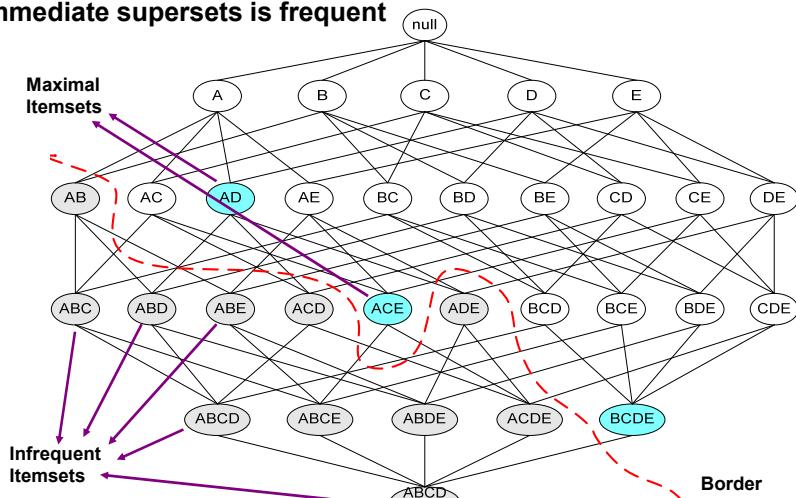
Introduction to Data Mining, 2nd Edition

51

51

Maximal Frequent Itemset

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



10/26/2020

Introduction to Data Mining, 2nd Edition

52

52

What are the Maximal Frequent Itemsets in this Data?

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	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	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	1	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	1	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	1	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	1	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	1	1	1	1	1	1	1	1	1	

Minimum support threshold = 5

(A1-A10)
(B1-B10)
(C1-C10)

10/26/2020

Introduction to Data Mining, 2nd Edition

53

An illustrative example

Transactions	Items										Support threshold (by count) : 5	Frequent itemsets: ?	Maximal itemsets: ?	
	A	B	C	D	E	F	G	H	I	J				
1														
2	1													
3		1												
4			1											
5				1										
6					1									
7						1								
8							1							
9								1						
10									1					

10/26/2020

Introduction to Data Mining, 2nd Edition

54

54

An illustrative example

		Items									
		A	B	C	D	E	F	G	H	I	J
Transactions	1										
	2	■		■	■	■	■			■	
	3		■	■	■	■		■	■		
	4									■	
	5				■						
	6					■					
	7								■		
	8										
	9									■	
	10										

Support threshold (by count) : 5
Frequent itemsets: {F}
Maximal itemsets: {F}

Support threshold (by count): 4
Frequent itemsets: ?
Maximal itemsets: ?

10/26/2020

Introduction to Data Mining, 2nd Edition

55

		Items									
		A	B	C	D	E	F	G	H	I	J
Transactions	1										
	2	■		■	■	■	■			■	
	3		■	■	■	■		■	■		
	4									■	
	5				■						
	6					■					
	7								■		
	8										
	9									■	
	10										

Support threshold (by count) : 5
Frequent itemsets: {F}
Maximal itemsets: {F}

Support threshold (by count): 4
Frequent itemsets: {E}, {F}, {E,F}, {J}
Maximal itemsets: {E,F}, {J}

Support threshold (by count): 3
Frequent itemsets: ?
Maximal itemsets: ?

10/26/2020

Introduction to Data Mining, 2nd Edition

56

56

An illustrative example

		Items									
		A	B	C	D	E	F	G	H	I	J
Transactions	1										
	2	■		■	■	■	■			■	
	3			■	■	■		■	■		
	4				■	■				■	
	5					■	■				
	6						■				
	7								■		
	8										
	9									■	
	10										

Support threshold (by count) : 5
Frequent itemsets: {F}
Maximal itemsets: {F}

Support threshold (by count): 4
Frequent itemsets: {E}, {F}, {E,F}, {J}
Maximal itemsets: {E,F}, {J}

Support threshold (by count): 3
Frequent itemsets:
All subsets of {C,D,E,F} + {J}
Maximal itemsets:
{C,D,E,F}, {J}

10/26/2020

Introduction to Data Mining, 2nd Edition

57

		Items									
		A	B	C	D	E	F	G	H	I	J
Transactions	1										
	2	■	■	■							
	3										
	4	■	■	■							
	5		■	■							
	6	■		■	■						
	7										
	8		■	■							
	9										
	10										

Support threshold (by count) : 5
Maximal itemsets: {A}, {B}, {C}

Support threshold (by count): 4
Maximal itemsets: {A,B}, {A,C},{B,C}

Support threshold (by count): 3
Maximal itemsets: {A,B,C}

10/26/2020

Introduction to Data Mining, 2nd Edition

58

58

Closed Itemset

- An itemset X is closed if none of its immediate supersets has the same support as the itemset X.
- X is not closed if at least one of its immediate supersets has support count as X.**

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}	2
{A,B,C,D}	2

10/26/2020

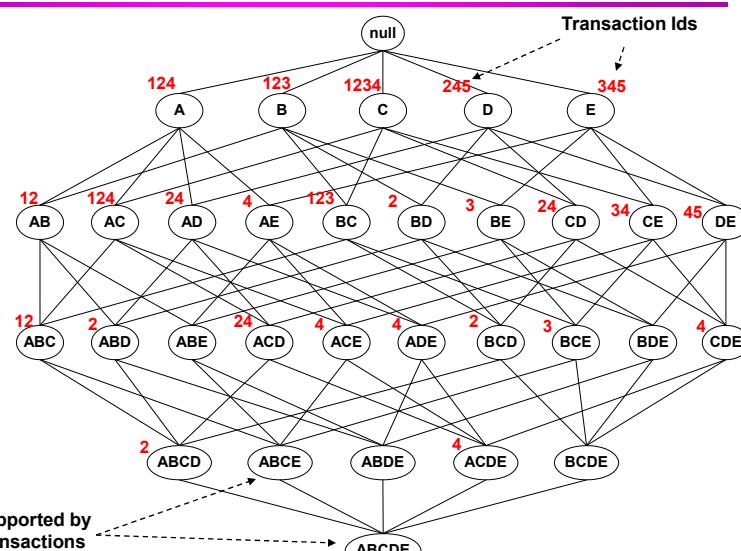
Introduction to Data Mining, 2nd Edition

59

59

Maximal vs Closed Itemsets

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

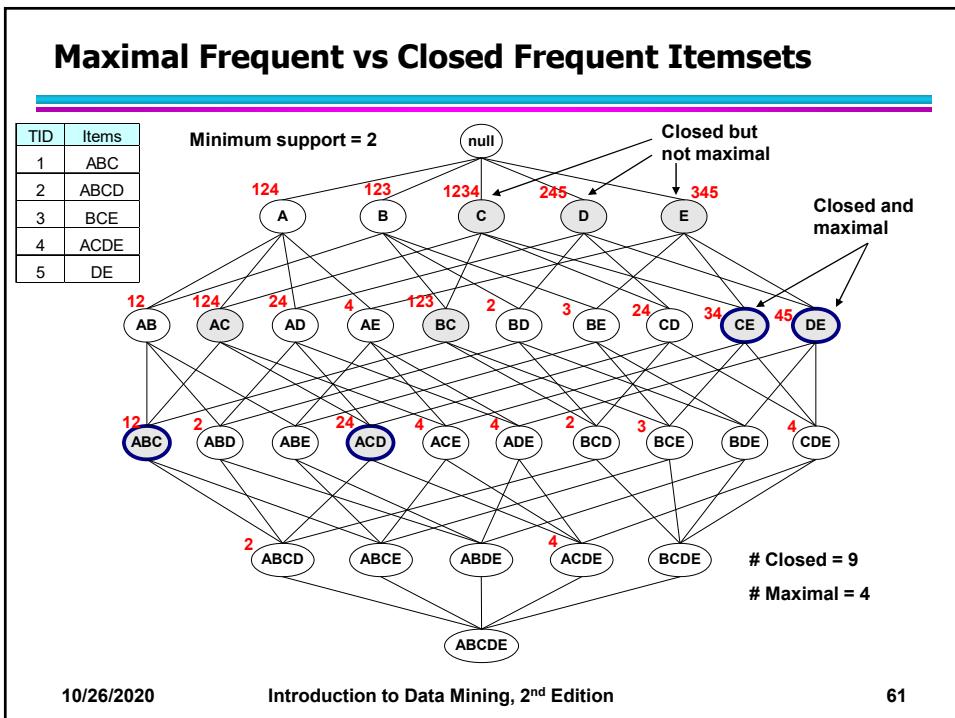


10/26/2020

Introduction to Data Mining, 2nd Edition

60

60



61

What are the Closed Itemsets in this Data?

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	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	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	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	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	0
6	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	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	1	1	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	1	1	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	1	1	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	1	1	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	1	1	1	1	1	1	1	1	1	1

(A1-A10)
(B1-B10)
(C1-C10)

10/26/2020 Introduction to Data Mining, 2nd Edition 62

Example 1

Transactions	Items										Itemsets	Support (counts)	Closed itemsets
	A	B	C	D	E	F	G	H	I	J			
1											{C}	3	
2											{D}	2	
3			■	■							{C,D}	2	
4				■	■								
5				■									
6													
7													
8													
9													
10													

10/26/2020

Introduction to Data Mining, 2nd Edition

63

Transactions	Items										Itemsets	Support (counts)	Closed itemsets
	A	B	C	D	E	F	G	H	I	J			
1											{C}	3	✓
2											{D}	2	
3			■	■							{C,D}	2	✓
4				■	■								
5				■									
6													
7													
8													
9													
10													

10/26/2020

Introduction to Data Mining, 2nd Edition

64

64

Example 2

		Items									
		A	B	C	D	E	F	G	H	I	J
Transactions	1										
	2										
	3			■	■	■					
	4			■	■	■					
	5		■	■							
	6										
	7										
	8										
	9										
	10										

		Items									
		A	B	C	D	E	F	G	H	I	J
Transactions	1										
	2										
	3			■	■	■					
	4			■	■	■					
	5		■	■							
	6										
	7										
	8										
	9										
	10										

		Itemsets									
		Itemsets	Support (counts)	Closed itemsets							
		{C}	3								
		{D}	2								
		{E}	2								
		{C,D}	2								
		{C,E}	2								
		{D,E}	2								
		{C,D,E}	2								

10/26/2020

Introduction to Data Mining, 2nd Edition

65

		Items									
		A	B	C	D	E	F	G	H	I	J
Transactions	1										
	2										
	3			■	■	■					
	4			■	■	■					
	5		■	■							
	6										
	7										
	8										
	9										
	10										

		Itemsets									
		Itemsets	Support (counts)	Closed itemsets							
		{C}	3	✓							
		{D}	2								
		{E}	2								
		{C,D}	2								
		{C,E}	2								
		{D,E}	2								
		{C,D,E}	2	✓							

10/26/2020

Introduction to Data Mining, 2nd Edition

66

66

Example 3

		Items									
		A	B	C	D	E	F	G	H	I	J
Transactions	1										
	2										
	3			■	■	■	■				
	4										
	5			■		■					
	6										
	7										
	8										
	9										
	10										

Closed itemsets: {C,D,E,F}, {C,F}

10/26/2020

Introduction to Data Mining, 2nd Edition

67

		Items									
		A	B	C	D	E	F	G	H	I	J
Transactions	1										
	2										
	3			■	■	■	■				
	4			■	■	■	■				
	5			■		■					
	6						■				
	7										
	8										
	9										
	10										

Closed itemsets: {C,D,E,F}, {C}, {F}

10/26/2020

Introduction to Data Mining, 2nd Edition

68

68

Maximal vs Closed Itemsets

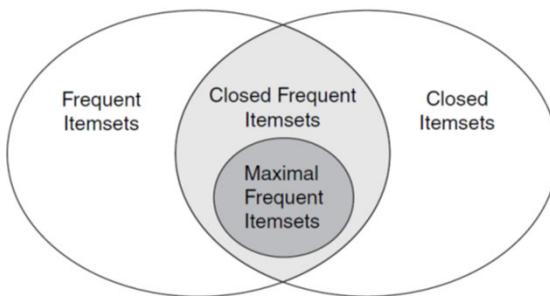


Figure 5.18. Relationships among frequent, closed, closed frequent, and maximal frequent itemsets.

10/26/2020

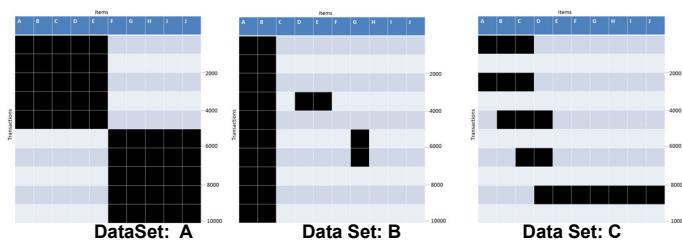
Introduction to Data Mining, 2nd Edition

69

69

Example question

- Given the following transaction data sets (dark cells indicate presence of an item in a transaction) and a support threshold of 20%, answer the following questions



- What is the number of frequent itemsets for each dataset? Which dataset will produce the most number of frequent itemsets?
- Which dataset will produce the longest frequent itemset?
- Which dataset will produce frequent itemsets with highest maximum support?
- Which dataset will produce frequent itemsets containing items with widely varying support levels (i.e., itemsets containing items with mixed support, ranging from 20% to more than 70%)?
- What is the number of maximal frequent itemsets for each dataset? Which dataset will produce the most number of maximal frequent itemsets?
- What is the number of closed frequent itemsets for each dataset? Which dataset will produce the most number of closed frequent itemsets?

10/26/2020

Introduction to Data Mining, 2nd Edition

70

70

Pattern Evaluation

- Association rule algorithms can produce large number of rules
- Interestingness measures can be used to prune/rank the patterns
 - In the original formulation, support & confidence are the only measures used

10/26/2020

Introduction to Data Mining, 2nd Edition

71

71

Computing Interestingness Measure

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

Contingency table

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

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, Gini, entropy, etc.

10/26/2020

Introduction to Data Mining, 2nd Edition

72

72

Drawback of Confidence

Customers	Tea	Coffee	...
C1	0	1	...
C2	1	0	...
C3	1	1	...
C4	1	0	...
...			

	<i>Coffee</i>	<i><u>Coffee</u></i>	
<i>Tea</i>	150	50	200
<i><u>Tea</u></i>	650	150	800
	800	200	1000

Association Rule: Tea → Coffee

$$\text{Confidence} \cong P(\text{Coffee}|\text{Tea}) = 150/200 = 0.75$$

Confidence > 50%, meaning people who drink tea are more likely to drink coffee than not drink coffee

So rule seems reasonable

10/26/2020

Introduction to Data Mining, 2nd Edition

73

73

Drawback of Confidence

	<u>Coffee</u>	<u><u>Coffee</u></u>	
Tea	150	50	200
<u>Tea</u>	650	150	800
	800	200	1000

Association Rule: Tea → Coffee

$$\text{Confidence} = P(\text{Coffee}|\text{Tea}) = 150/200 = 0.75$$

but $P(\text{Coffee}) = 0.8$, which means knowing that a person drinks tea reduces the probability that the person drinks coffee!

⇒ Note that $P(\text{Coffee}|\text{Tea}) = 650/800 = 0.8125$

10/26/2020

Introduction to Data Mining, 2nd Edition

74

74

Drawback of Confidence

Customers	Tea	Honey	...
C1	0	1	...
C2	1	0	...
C3	1	1	...
C4	1	0	...
...			

	Honey	\overline{Honey}	
Tea	100	100	200
\overline{Tea}	20	780	800
	120	880	1000

Association Rule: Tea \rightarrow Honey

$$\text{Confidence} \cong P(\text{Honey}|\text{Tea}) = 100/200 = 0.50$$

Confidence = 50%, which may mean that drinking tea has little influence whether honey is used or not

So rule seems uninteresting

But $P(\text{Honey}) = 120/1000 = .12$ (hence tea drinkers are far more likely to have honey)

10/26/2020 Introduction to Data Mining, 2nd Edition

75

Measure for Association Rules

- So, what kind of rules do we really want?
 - Confidence($X \rightarrow Y$) should be sufficiently high
 - ◆ To ensure that people who buy X will more likely buy Y than not buy Y
 - Confidence($X \rightarrow Y$) $>$ support(Y)
 - ◆ Otherwise, rule will be misleading because having item X actually reduces the chance of having item Y in the same transaction
 - ◆ Is there any measure that capture this constraint?
 - Answer: Yes. There are many of them.

10/26/2020

Introduction to Data Mining, 2nd Edition

76

76

Statistical Relationship between X and Y

- The criterion

$$\text{confidence}(X \rightarrow Y) = \text{support}(Y)$$

is equivalent to:

- $P(Y|X) = P(Y)$
- $P(X,Y) = P(X) \times P(Y)$ (X and Y are independent)

If $P(X,Y) > P(X) \times P(Y)$: X & Y are positively correlated

If $P(X,Y) < P(X) \times P(Y)$: X & Y are negatively correlated

10/26/2020

Introduction to Data Mining, 2nd Edition

77

77

Measures that take into account statistical dependence

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

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

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

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

lift is used for rules while
interest is used for itemsets

10/26/2020

Introduction to Data Mining, 2nd Edition

78

78

Example: Lift/Interest

	Coffee	<u>Coffee</u>	
Tea	150	50	200
<u>Tea</u>	650	150	800
	800	200	1000

Association Rule: Tea → Coffee

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

$$\text{but } P(\text{Coffee}) = 0.8$$

⇒ Interest = $0.15 / (0.2 \times 0.8) = 0.9375 (< 1, \text{ therefore is negatively associated})$

So, is it enough to use confidence/Interest for pruning?

10/26/2020

Introduction to Data Mining, 2nd Edition

79

79

There are lots of measures proposed in the literature

Measure (Symbol)	Definition
Correlation (ϕ)	$\frac{Nf_{11}-f_{1+}f_{+1}}{\sqrt{f_{1+}f_{+1}f_{0+}f_{+0}}}$
Odds ratio (α)	$(f_{11}f_{00}) / (f_{10}f_{01})$
Kappa (κ)	$\frac{Nf_{11}+Nf_{00}-f_{1+}f_{+1}-f_{0+}f_{+0}}{N^2-f_{1+}f_{+1}-f_{0+}f_{+0}}$
Interest (I)	$(Nf_{11}) / (f_{1+}f_{+1})$
Cosine (IS)	$(f_{11}) / (\sqrt{f_{1+}f_{+1}})$
Piatetsky-Shapiro (PS)	$\frac{f_{11}}{N} - \frac{f_{1+}f_{+1}}{N^2}$
Collective strength (S)	$\frac{f_{11}+f_{00}}{f_{1+}f_{+1}+f_{0+}f_{+0}} \times \frac{N-f_{1+}f_{+1}-f_{0+}f_{+0}}{N-f_{11}-f_{00}}$
Jaccard (ζ)	$f_{11} / (f_{1+} + f_{+1} - f_{11})$
All-confidence (h)	$\min \left[\frac{f_{11}}{f_{1+}}, \frac{f_{11}}{f_{+1}} \right]$

10/26/2020

Introduction to Data Mining, 2nd Edition

80

80

Comparing Different Measures

10 examples of contingency tables:

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

Rankings of contingency tables using various measures:

	ϕ	α	κ	I	IS	PS	S	ζ	h
E_1	1	3	1	6	2	2	1	2	2
E_2	2	1	2	7	3	5	2	3	3
E_3	3	2	4	4	5	1	3	6	8
E_4	4	8	3	3	7	3	4	7	5
E_5	5	7	6	2	9	6	6	9	9
E_6	6	9	5	5	6	4	5	5	7
E_7	7	6	7	9	1	8	7	1	1
E_8	8	10	8	8	8	7	8	8	7
E_9	9	4	9	10	4	9	9	4	4
E_{10}	10	5	10	1	10	10	10	10	10

10/26/2020

Introduction to Data Mining, 2nd Edition

81

Property under Inversion Operation					
Transaction 1 →		A	B	\bar{A}	\bar{B}
■		1	0	0	1
■		0	0	1	1
■		0	0	1	1
■		0	1	1	1
■		0	0	1	0
■		0	0	1	1
■		0	0	1	1
■		0	0	1	1
■		0	0	0	1
Transaction N →		1	0		
(a)			(b)		

10/26/2020

Introduction to Data Mining, 2nd Edition

82

82

Property under Inversion Operation

Figure 1 consists of two parts, (a) and (b), showing correlation matrices between two sets of binary vectors, A and B.

Part (a): This section shows the raw binary data. It has two vertical columns labeled "A" and "B". The first column "A" contains 10 rows of binary digits (0 or 1). The second column "B" also contains 10 rows of binary digits. A red arrow points from the label "Transaction 1" to the first row of column A. Another red arrow points from the label "Transaction N" to the last row of column A. The data is as follows:

	A	B
Transaction 1	1	0
■	0	0
■	0	0
■	0	0
■	0	1
■	0	0
■	0	0
■	0	0
■	0	0
Transaction N	1	0

Part (b): This section shows the normalized data. The same two columns "A" and "B" are present, but every entry in both columns is now either 0 or 1. The red arrows from part (a) still point to the first and last rows of column A. The data is as follows:

	A	B
Transaction 1	0	1
■	1	1
■	1	1
■	1	1
■	1	0
■	1	1
■	1	1
■	1	1
■	1	1
Transaction N	0	1

Correlation:

IS/cosine	-0.1667	-0.1667
0.0	0.0	0.825

10/26/2020

Introduction to Data Mining, 2nd Edition

83

83

Property under Null Addition

	B	\bar{B}		
A	700	100	800	
\bar{A}	100	100	200	
	800	200	1000	

→

	B	\bar{B}	
A	700	100	800
\bar{A}	10	1100	1200
	800	1200	2000

Invariant measures:

- ◆ cosine, Jaccard, All-confidence, confidence

Non-invariant measures:

- ◆ correlation, Interest/Lift, odds ratio, etc

10/26/2020

Introduction to Data Mining, 2nd Edition

84

84

Property under Row/Column Scaling

Grade-Gender Example (Mosteller, 1968):

	Female	Male	
High	30	20	50
Low	40	10	50
	70	30	100
	Female	Male	
High	60	60	120
Low	80	30	110
	140	90	230

↓ ↓
2x 3x

Mosteller:

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

Odds-Ratio has this property

10/26/2020

Introduction to Data Mining, 2nd Edition

85

85

Property under Row/Column Scaling

Relationship between Mask use and susceptibility to Covid:

	Covid-Positive	Covid-Free			Covid-Positive	Covid-Free	
Mask	20	30	50	Mask	40	300	340
No-Mask	40	10	50	No-Mask	80	100	180
	60	40	100		120	400	520

↓ ↓
2x 10x

Mosteller:

Underlying association should be independent of the relative number of Covid-positive and Covid-free subjects

Odds-Ratio has this property

10/26/2020

Introduction to Data Mining, 2nd Edition

86

86

Different Measures have Different Properties

Symbol	Measure	Inversion	Null Addition	Scaling
ϕ	ϕ -coefficient	Yes	No	No
α	odds ratio	Yes	No	Yes
κ	Cohen's	Yes	No	No
I	Interest	No	No	No
IS	Cosine	No	Yes	No
PS	Piatetsky-Shapiro's	Yes	No	No
S	Collective strength	Yes	No	No
ζ	Jaccard	No	Yes	No
h	All-confidence	No	Yes	No
s	Support	No	No	No

10/26/2020

Introduction to Data Mining, 2nd Edition

87

87

Simpson's Paradox

- Observed relationship in data may be influenced by the presence of other confounding factors (hidden variables)
 - Hidden variables may cause the observed relationship to disappear or reverse its direction!
- Proper stratification is needed to avoid generating spurious patterns

10/26/2020

Introduction to Data Mining, 2nd Edition

88

88

Simpson's Paradox

- Recovery rate from Covid
 - Hospital A: 80%
 - Hospital B: 90%
- Which hospital is better?

10/26/2020

Introduction to Data Mining, 2nd Edition

89

89

Simpson's Paradox

- Recovery rate from Covid
 - Hospital A: 80%
 - Hospital B: 90%
- Which hospital is better?
- Covid recovery rate on older population
 - Hospital A: 50%
 - Hospital B: 30%
- Covid recovery rate on younger population
 - Hospital A: 99%
 - Hospital B: 98%

10/26/2020

Introduction to Data Mining, 2nd Edition

90

90

Simpson's Paradox

- Covid-19 death: (per 100,000 of population)
 - County A: 15
 - County B: 10
- Which state is managing the pandemic better?

10/26/2020

Introduction to Data Mining, 2nd Edition

91

91

Simpson's Paradox

- Covid-19 death: (per 100,000 of population)
 - County A: 15
 - County B: 10
- Which state is managing the pandemic better?
- Covid death rate on older population
 - County A: 20
 - County B: 40
- Covid death rate on younger population
 - County A: 2
 - County B: 5

10/26/2020

Introduction to Data Mining, 2nd Edition

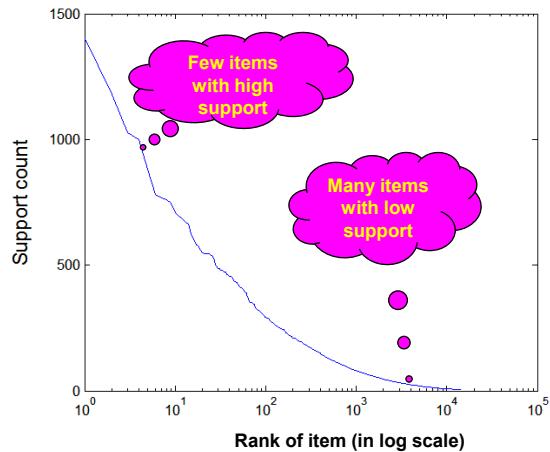
92

92

Effect of Support Distribution on Association Mining

- Many real data sets have skewed support distribution

Support distribution of a retail data set



10/26/2020

Introduction to Data Mining, 2nd Edition

93

Effect of Support Distribution

- Difficult to set the appropriate *minsup* threshold
 - If *minsup* is too high, we could miss itemsets involving interesting rare items (e.g., {caviar, vodka})
 - If *minsup* is too low, it is computationally expensive and the number of itemsets is very large

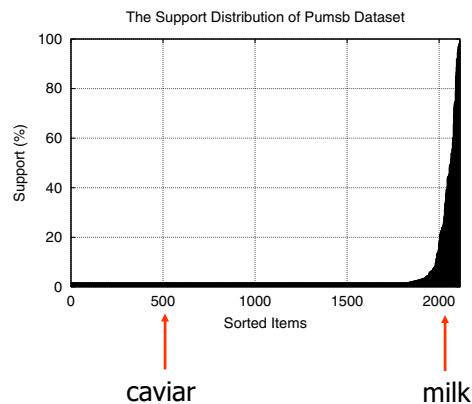
10/26/2020

Introduction to Data Mining, 2nd Edition

94

94

Cross-Support Patterns



A cross-support pattern involves items with varying degree of support

- Example: {caviar,milk}

How to avoid such patterns?

A Measure of Cross Support

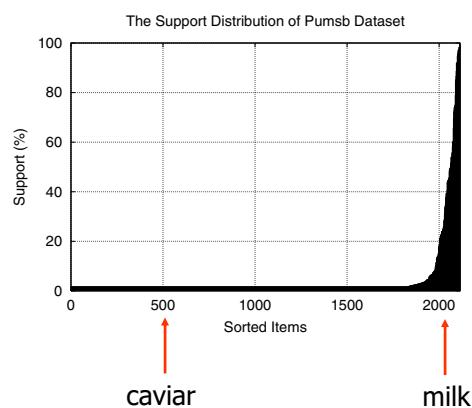
- Given an itemset, $X = \{x_1, x_2, \dots, x_d\}$, with d items, we can define a measure of cross support, r , for the itemset

$$r(X) = \frac{\min\{s(x_1), s(x_2), \dots, s(x_d)\}}{\max\{s(x_1), s(x_2), \dots, s(x_d)\}}$$

where $s(x_i)$ is the support of item x_i

- Can use $r(X)$ to prune cross support patterns

Confidence and Cross-Support Patterns



Observation:

$\text{conf}(\text{caviar} \rightarrow \text{milk})$ is very high

but

$\text{conf}(\text{milk} \rightarrow \text{caviar})$ is very low

Therefore,

$\min(\text{conf}(\text{caviar} \rightarrow \text{milk}), \text{conf}(\text{milk} \rightarrow \text{caviar}))$

is also very low

10/26/2020

Introduction to Data Mining, 2nd Edition

97

97

H-Confidence

- To avoid patterns whose items have very different support, define a new evaluation measure for itemsets
 - Known as h-confidence or all-confidence
- Specifically, given an itemset $X = \{x_1, x_2, \dots, x_d\}$
 - h-confidence is the minimum confidence of any association rule formed from itemset X
 - $\text{hconf}(X) = \min(\text{conf}(X_1 \rightarrow X_2), \dots)$, where $X_1, X_2 \subset X, X_1 \cap X_2 = \emptyset, X_1 \cup X_2 = X$
 - For example: $X_1 = \{x_1, x_2\}, X_2 = \{x_3, \dots, x_d\}$

10/26/2020

Introduction to Data Mining, 2nd Edition

98

98

H-Confidence ...

- But, given an itemset $X = \{x_1, x_2, \dots, x_d\}$
 - What is the lowest confidence rule you can obtain from X ?
 - Recall $\text{conf}(X_1 \rightarrow X_2) = s(X_1 \cup X_2) / \text{support}(X_1)$
 - ◆ The numerator is fixed: $s(X_1 \cup X_2) = s(X)$
 - ◆ Thus, to find the lowest confidence rule, we need to find the X_1 with highest support
 - ◆ Consider only rules where X_1 is a single item, i.e.,
 $\{x_1\} \rightarrow X - \{x_1\}$, $\{x_2\} \rightarrow X - \{x_2\}$, ..., or $\{x_d\} \rightarrow X - \{x_d\}$

$$\begin{aligned}\text{hconf}(X) &= \min \left\{ \frac{s(X)}{s(x_1)}, \frac{s(X)}{s(x_2)}, \dots, \frac{s(X)}{s(x_d)} \right\} \\ &= \frac{s(X)}{\max\{s(x_1), s(x_2), \dots, s(x_d)\}}\end{aligned}$$

10/26/2020

Introduction to Data Mining, 2nd Edition

99

99

Cross Support and H-confidence

- By the anti-montone property of support

$$s(X) \leq \min\{s(x_1), s(x_2), \dots, s(x_d)\}$$

- Therefore, we can derive a relationship between the h-confidence and cross support of an itemset

$$\begin{aligned}\text{hconf}(X) &= \frac{s(X)}{\max\{s(x_1), s(x_2), \dots, s(x_d)\}} \\ &\leq \frac{\min\{s(x_1), s(x_2), \dots, s(x_d)\}}{\max\{s(x_1), s(x_2), \dots, s(x_d)\}} \\ &= r(X)\end{aligned}$$

Thus, $\text{hconf}(X) \leq r(X)$

10/26/2020

Introduction to Data Mining, 2nd Edition

100

100

Cross Support and H-confidence ...

- Since, $\text{hconf}(X) \leq r(X)$, we can eliminate cross support patterns by finding patterns with h-confidence $< h_c$, a user set threshold
- Notice that

$$0 \leq \text{hconf}(X) \leq r(X) \leq 1$$

- Any itemset satisfying a given h-confidence threshold, h_c , is called a **hyperclique**
- H-confidence can be used instead of or in conjunction with support

10/26/2020

Introduction to Data Mining, 2nd Edition

101

101

Properties of Hypercliques

- Hypercliques are itemsets, but not necessarily frequent itemsets
 - Good for finding low support patterns
- H-confidence is anti-monotone
- Can define closed and maximal hypercliques in terms of h-confidence
 - A hyperclique X is closed if none of its immediate supersets has the same h-confidence as X
 - A hyperclique X is maximal if $\text{hconf}(X) \leq h_c$ and none of its immediate supersets, Y , have $\text{hconf}(Y) \leq h_c$

10/26/2020

Introduction to Data Mining, 2nd Edition

102

102

Properties of Hypercliques ...

- Hypercliques have the high-affinity property
 - Think of the individual items as sparse binary vectors
 - h-confidence gives us information about their pairwise Jaccard and cosine similarity
 - ◆ Assume x_1 and x_2 are any two items in an itemset X
 - ◆ $\text{Jaccard}(x_1, x_2) \geq \text{hconf}(X)/2$
 - ◆ $\cos(x_1, x_2) \geq \text{hconf}(X)$
 - Hypercliques that have a high h-confidence consist of very similar items as measured by Jaccard and cosine
- The items in a hyperclique cannot have widely different support
 - Allows for more efficient pruning

10/26/2020

Introduction to Data Mining, 2nd Edition

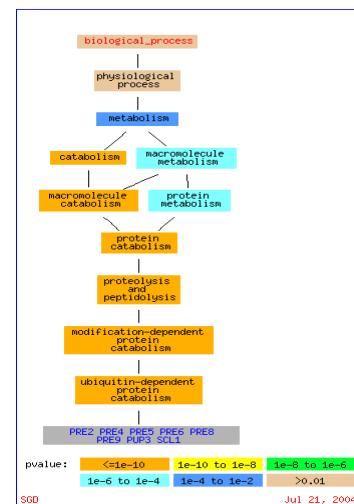
103

103

Example Applications of Hypercliques

- Hypercliques are used to find strongly coherent groups of items
 - Words that occur together in documents
 - Proteins in a protein interaction network

In the figure at the right, a gene ontology hierarchy for biological process shows that the identified proteins in the hyperclique (PRE2, ..., SCL1) perform the same function and are involved in the same biological process



10/26/2020

Introduction to Data Mining, 2nd Edition

104

104

Data Mining

Chapter 6 Association Analysis: Advance Concepts

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

Data Mining **Association Analysis: Advanced Concepts**

Extensions of Association Analysis to
Continuous and Categorical Attributes and
Multi-level Rules

Continuous and Categorical Attributes

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

Gender	...	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern
Female	...	26	90K	20	4	Yes
Male	...	51	135K	10	2	No
Male	...	29	80K	10	3	Yes
Female	...	45	120K	15	3	Yes
Female	...	31	95K	20	5	Yes
Male	...	25	55K	25	5	Yes
Male	...	37	100K	10	1	No
Male	...	41	65K	8	2	No
Female	...	26	85K	12	1	No
...

Example of Association Rule:

{Gender=Male, Age ∈ [21,30)} → {No of hours online ≥ 10}

02/03/2018

Introduction to Data Mining

3

Handling Categorical Attributes

- Example: Internet Usage Data

Gender	Level of Education	State	Computer at Home	Online Auction	Chat Online	Online Banking	Privacy Concerns
Female	Graduate	Illinois	Yes	Yes	Daily	Yes	Yes
Male	College	California	No	No	Never	No	No
Male	Graduate	Michigan	Yes	Yes	Monthly	Yes	Yes
Female	College	Virginia	No	Yes	Never	Yes	Yes
Female	Graduate	California	Yes	No	Never	No	Yes
Male	College	Minnesota	Yes	Yes	Weekly	Yes	Yes
Male	College	Alaska	Yes	Yes	Daily	Yes	No
Male	High School	Oregon	Yes	No	Never	No	No
Female	Graduate	Texas	No	No	Monthly	No	No
...

{Level of Education=Graduate, Online Banking=Yes}
→ {Privacy Concerns = Yes}

02/03/2018

Introduction to Data Mining

4

Handling Categorical Attributes

- Introduce a new “item” for each distinct attribute-value pair

Male	Female	Education = Graduate	Education = College	Education = High School	...	Privacy = Yes	Privacy = No
0	1	1	0	0	...	1	0
1	0	0	1	0	...	0	1
1	0	1	0	0	...	1	0
0	1	0	1	0	...	1	0
0	1	1	0	0	...	1	0
1	0	0	1	0	...	1	0
1	0	0	0	0	...	0	1
1	0	0	0	1	...	0	1
0	1	1	0	0	...	0	1
...

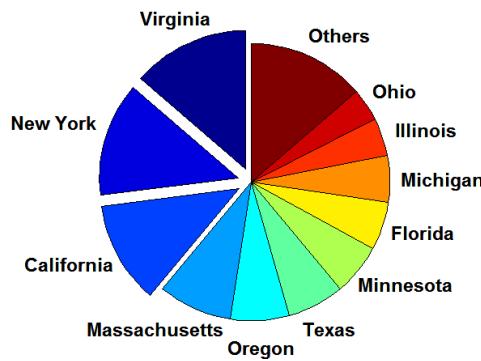
02/03/2018

Introduction to Data Mining

5

Handling Categorical Attributes

- Some attributes can have many possible values
 - Many of their attribute values have very low support
 - Potential solution: Aggregate the low-support attribute values



02/03/2018

Introduction to Data Mining

6

Handling Categorical Attributes

- Distribution of attribute values can be highly skewed
 - Example: 85% of survey participants own a computer at home
 - ◆ Most records have Computer at home = Yes
 - ◆ Computation becomes expensive; many frequent itemsets involving the binary item (Computer at home = Yes)
 - ◆ Potential solution:
 - discard the highly frequent items
 - Use alternative measures such as h-confidence
- Computational Complexity
 - Binarizing the data increases the number of items
 - But the width of the “transactions” remain the same as the number of original (non-binarized) attributes
 - Produce more frequent itemsets but maximum size of frequent itemset is limited to the number of original attributes

Handling Continuous Attributes

- Different methods:
 - Discretization-based
 - Statistics-based
 - Non-discretization based
 - ◆ minApriori
- Different kinds of rules can be produced:
 - $\{\text{Age} \in [21,30], \text{No of hours online} \in [10,20]\}$
→ {Chat Online = Yes}
 - $\{\text{Age} \in [21,30], \text{Chat Online} = \text{Yes}\}$
→ No of hours online: $\mu=14, \sigma=4$

Discretization-based Methods



Gender	...	Age	Annual Income	No of hours spent online per week	No of email accounts	Privacy Concern
Female	...	26	90K	20	4	Yes
Male	...	51	135K	10	2	No
Male	...	29	80K	10	3	Yes
Female	...	45	120K	15	3	Yes
Female	...	31	95K	20	5	Yes
Male	...	25	55K	25	5	Yes
Male	...	37	100K	10	1	No
Male	...	41	65K	8	2	No
Female	...	26	85K	12	1	No
...

Male	Female	...	Age < 13	Age ∈ [13, 21)	Age ∈ [21, 30)	...	Privacy = Yes	Privacy = No
0	1	...	0	0	1	...	1	0
1	0	...	0	0	0	...	0	1
1	0	...	0	0	1	...	1	0
0	1	...	0	0	0	...	1	0
0	1	...	0	0	0	...	1	0
1	0	...	0	0	1	...	1	0
1	0	...	0	0	0	...	0	1
1	0	...	0	0	0	...	0	1
0	1	...	0	0	1	...	0	1
...

02/03/2018

Introduction to Data Mining

9

Discretization-based Methods

Unsupervised:

- Equal-width binning <1 2 3> <4 5 6> <7 8 9>
- Equal-depth binning <1 2> <3 4 5 6 7> <8 9>
- Cluster-based

Supervised discretization

Continuous attribute, v

	1	2	3	4	5	6	7	8	9
Chat Online = Yes	0	0	20	10	20	0	0	0	0
Chat Online = No	150	100	0	0	0	100	100	150	100

bin1 bin2 bin3

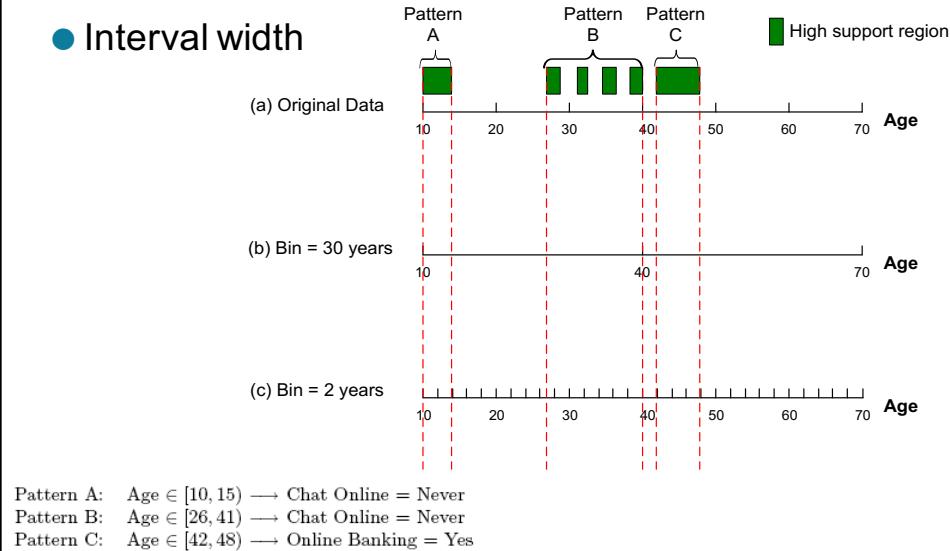
02/03/2018

Introduction to Data Mining

10

Discretization Issues

● Interval width



02/03/2018

Introduction to Data Mining

11

Discretization Issues

● Interval too wide (e.g., Bin size= 30)

- May merge several disparate patterns
 - ◆ Patterns A and B are merged together
- May lose some of the interesting patterns
 - ◆ Pattern C may not have enough confidence

● Interval too narrow (e.g., Bin size = 2)

- Pattern A is broken up into two smaller patterns
 - ◆ Can recover the pattern by merging adjacent subpatterns
- Pattern B is broken up into smaller patterns
 - ◆ Cannot recover the pattern by merging adjacent subpatterns
- Some windows may not meet support threshold

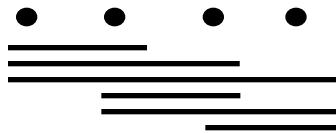
02/03/2018

Introduction to Data Mining

12

Discretization: all possible intervals

Number of intervals = k
Total number of Adjacent intervals = $k(k-1)/2$



● Execution time

- If the range is partitioned into k intervals, there are $O(k^2)$ new items
- If an interval $[a,b]$ is frequent, then all intervals that subsume $[a,b]$ must also be frequent
 - ◆ E.g.: if $\{\text{Age} \in [21,25], \text{Chat Online}=Yes\}$ is frequent, then $\{\text{Age} \in [10,50], \text{Chat Online}=Yes\}$ is also frequent
- Improve efficiency:
 - ◆ Use maximum support to avoid intervals that are too wide

Discretization Issues

● Redundant rules

R1: $\{\text{Age} \in [18,20], \text{Age} \in [10,12]\} \rightarrow \{\text{Chat Online}=Yes\}$

R2: $\{\text{Age} \in [18,23], \text{Age} \in [10,20]\} \rightarrow \{\text{Chat Online}=Yes\}$

- If both rules have the same support and confidence, prune the more specific rule (R1)

Statistics-based Methods

- Example:

$\{\text{Income} > 100K, \text{Online Banking} = \text{Yes}\} \rightarrow \text{Age: } \mu = 34$

- Rule consequent consists of a continuous variable, characterized by their statistics

- mean, median, standard deviation, etc.

- Approach:

- Withhold the target attribute from the rest of the data
 - Extract frequent itemsets from the rest of the attributes
 - ◆ Binarized the continuous attributes (except for the target attribute)
 - For each frequent itemset, compute the corresponding descriptive statistics of the target attribute
 - ◆ 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



- Frequent Itemsets:

- {Male, Income > 100K}
 - {Income < 30K, No hours $\in [10, 15]$ }
 - {Income > 100K, Online Banking = Yes}
 -

- Association Rules:

- {Male, Income > 100K} \rightarrow Age: $\mu = 30$
 - {Income < 40K, No hours $\in [10, 15]$ } \rightarrow Age: $\mu = 24$
 - {Income > 100K, Online Banking = Yes} \rightarrow Age: $\mu = 34$
 -

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

Statistics-based Methods

- Example:

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

- Rule is interesting if difference between μ and μ' is more 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

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
- 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

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

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 L₁ norms
 - 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:

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

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:

$$\text{Sup}(W1, W2, W3)$$

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

$$= 0.17$$

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

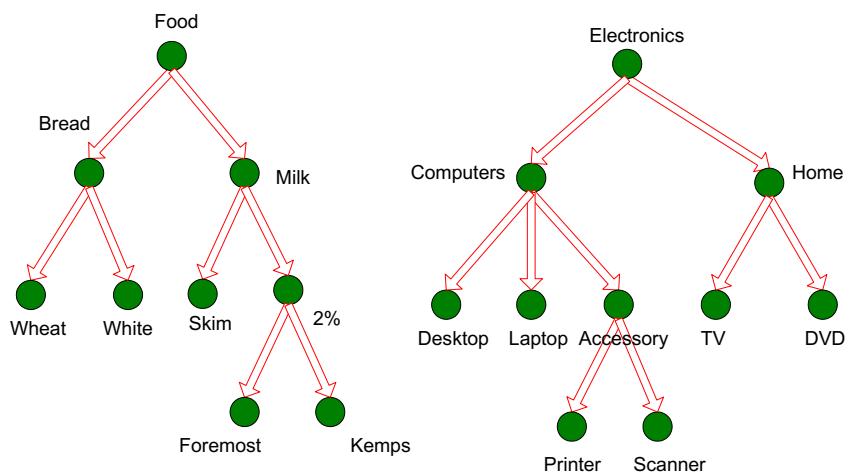
Example:

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

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

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

Concept Hierarchies



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
 - Rules at higher level of hierarchy may be too generic

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 $\text{conf}(X \Rightarrow Y) \geq \text{minconf}$

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

Data Mining Association Analysis: Advanced Concepts

Sequential Patterns

Examples of Sequence

- Sequence of different transactions by a customer at an online store:
< {Digital Camera,iPad} {memory card} {headphone,iPad cover} >
- Sequence of initiating events causing the nuclear accident at 3-mile Island:
(http://stellar-one.com/nuclear/staf_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}>

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)

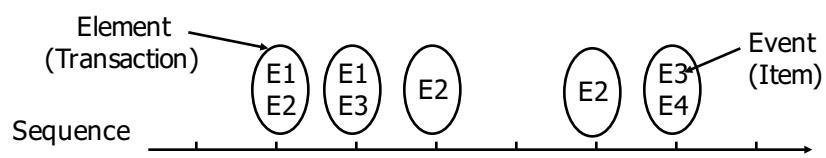
02/03/2018

Introduction to Data Mining

31

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



02/03/2018

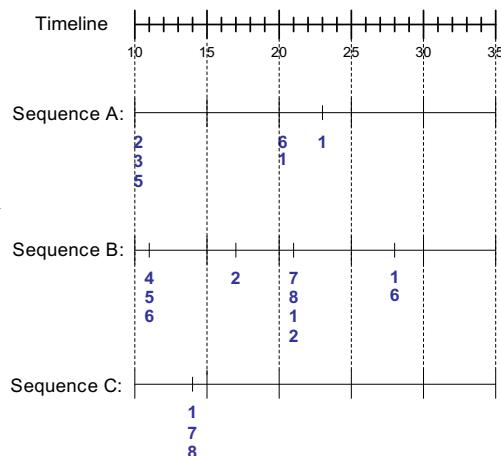
Introduction to Data Mining

32

Sequence Data

Sequence Database:

Sequence ID	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



02/03/2018

Introduction to Data Mining

33

Sequence Data vs. Market-basket Data

Sequence Database:

Customer	Date	Items bought
A	10	2, 3, 5
A	20	1, 6
A	23	1
B	11	4, 5, 6
B	17	2
B	21	1, 2, 7, 8
B	28	1, 6
C	14	1, 7, 8

Market- basket Data

Events
2, 3, 5
1, 6
1
4, 5, 6
2
1, 2, 7, 8
1, 6
1, 7, 8

02/03/2018

Introduction to Data Mining

34

Sequence Data vs. Market-basket Data

Sequence Database:

Customer	Date	Items bought
A	10	2, 3, 5
A	20	1, 6
A	23	1
B	11	4, 5, 6
B	17	2
B	21	1, 2, 7, 8
B	28	1, 6
C	14	1, 7, 8

Market- basket Data

Events
2, 3, 5
1, 6
1
4, 5, 6
2
1, 2, 7, 8
1, 6
1, 7, 8

02/03/2018

Introduction to Data Mining

35

Formal Definition of a Sequence

- A sequence is an ordered list of elements

$$s = \langle e_1 e_2 e_3 \dots \rangle$$

- Each element contains a collection of events (items)

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

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

02/03/2018

Introduction to Data Mining

36

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 $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \dots, a_n \subseteq b_{i_n}$
- Illustrative Example:

s: b_1 b_2 b_3 b_4 b_5
t: a_1 a_2 a_3

t is a subsequence of s if $a_1 \subseteq b_2, a_2 \subseteq b_3, a_3 \subseteq b_5$.

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

Sequential Pattern Mining: Definition

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

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%

02/03/2018

Introduction to Data Mining

39

Sequence Data vs. Market-basket Data

Sequence Database:

Customer	Date	Items bought
A	10	2, 3, 5
A	20	1,6
A	23	1
B	11	4, 5, 6
B	17	2
B	21	1,2,7,8
B	28	1, 6
C	14	1,7,8

{2} \Rightarrow {1}

$$conf(\{2\} \rightarrow \{1\}) = \frac{\sigma(\{2\} \{1\})}{\sigma(\{2\})}$$

Market- basket Data

Events

2, 3, 5

1,6

1

4,5,6

2

1,2,7,8

1,6

1,7,8

(1,8) \Rightarrow (7)

$$conf(1,8) \rightarrow (7) = \frac{\sigma(1,7,8)}{\sigma(\{1,8\})}$$

02/03/2018

Introduction to Data Mining

40

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:
 $\langle\{i_1, i_2\}\rangle, \langle\{i_1, i_3\}\rangle, \dots,$
 $\langle\{i_1\} \{i_1\}\rangle, \langle\{i_1\} \{i_2\}\rangle, \dots, \langle\{i_n\} \{i_n\}\rangle$
- Candidate 3 subsequences:
 $\langle\{i_1, i_2, i_3\}\rangle, \langle\{i_1, i_2, i_4\}\rangle, \dots,$
 $\langle\{i_1, i_2\} \{i_1\}\rangle, \langle\{i_1, i_2\} \{i_2\}\rangle, \dots,$
 $\langle\{i_1\} \{i_1, i_2\}\rangle, \langle\{i_1\} \{i_1, i_3\}\rangle, \dots,$
 $\langle\{i_1\} \{i_1\} \{i_1\}\rangle, \langle\{i_1\} \{i_1\} \{i_2\}\rangle, \dots$

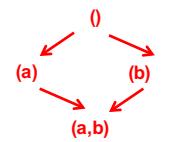
02/03/2018

Introduction to Data Mining

41

Extracting Sequential Patterns: Simple example

- Given 2 events: a, b
- Candidate 1 subsequences:
 $\langle\{a\}\rangle, \langle\{b\}\rangle.$
- Candidate 2 subsequences:
 $\langle\{a\} \{a\}\rangle, \langle\{a\} \{b\}\rangle, \langle\{b\} \{a\}\rangle, \langle\{b\} \{b\}\rangle, \langle\{a, b\}\rangle.$
- Candidate 3 subsequences:
 $\langle\{a\} \{a\} \{a\}\rangle, \langle\{a\} \{a\} \{b\}\rangle, \langle\{a\} \{b\} \{a\}\rangle, \langle\{a\} \{b\} \{b\}\rangle,$
 $\langle\{b\} \{b\} \{b\}\rangle, \langle\{b\} \{b\} \{a\}\rangle, \langle\{b\} \{a\} \{b\}\rangle, \langle\{b\} \{a\} \{a\}\rangle$
 $\langle\{a, b\} \{a\}\rangle, \langle\{a, b\} \{b\}\rangle, \langle\{a\} \{a, b\}\rangle, \langle\{b\} \{a, b\}\rangle$



02/03/2018

Introduction to Data Mining

42

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 the following candidate 2-sequences: $\langle\{i_1\} \{i_1\}\rangle$, $\langle\{i_1\} \{i_2\}\rangle$, $\langle\{i_2\} \{i_2\}\rangle$, $\langle\{i_2\} \{i_1\}\rangle$ 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 an event from the first element in w_1 is the same as the subsequence obtained by removing an event from the last element in w_2
 - ◆ The resulting candidate after merging is given by extending the sequence w_1 as follows
 - If the last element of w_2 has only one event, append it to w_1
 - Otherwise add the event from the last element of w_2 (which is absent in the last element of w_1) to the last element of w_1

Candidate Generation Examples

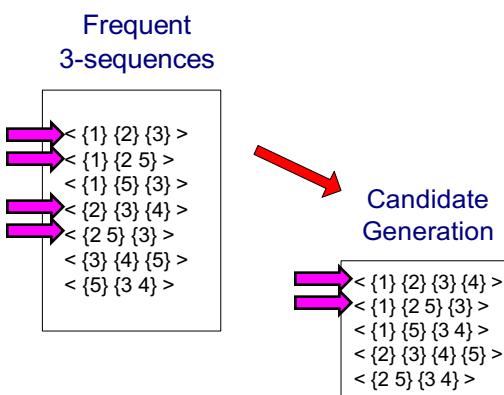
- Merging $w_1 = \langle \{1\} \{2\} \{3\} \{4\} \{6\} \rangle$ and $w_2 = \langle \{2\} \{3\} \{4\} \{6\} \{5\} \rangle$ produces the candidate sequence $\langle \{1\} \{2\} \{3\} \{4\} \{6\} \{5\} \rangle$ because the last element of w_2 has only one event
- Merging $w_1 = \langle \{1\} \{2\} \{3\} \{4\} \rangle$ and $w_2 = \langle \{2\} \{3\} \{4\} \{5\} \rangle$ produces the candidate sequence $\langle \{1\} \{2\} \{3\} \{4\} \{5\} \rangle$ because the last element in w_2 has more than one event
- Merging $w_1 = \langle \{1\} \{2\} \{3\} \rangle$ and $w_2 = \langle \{2\} \{3\} \{4\} \rangle$ produces the candidate sequence $\langle \{1\} \{2\} \{3\} \{4\} \rangle$ because the last element in w_2 has more than one event
- 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 $\langle \{2\} \{6\} \{4\} \{5\} \rangle$

11/19/2012

Introduction to Data Mining

45

GSP Example

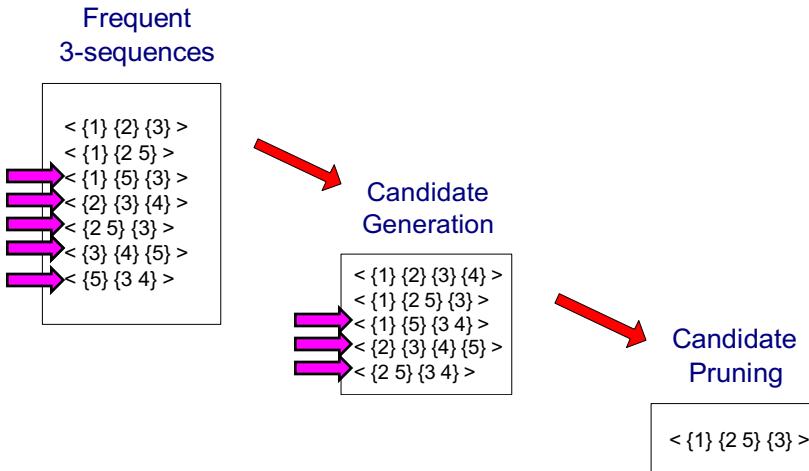


11/19/2012

Introduction to Data Mining

46

GSP Example

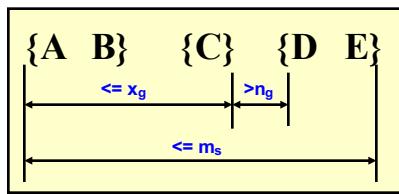


11/19/2012

Introduction to Data Mining

47

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, d	Sequential Pattern, s	d contains s?
$\langle \{2,4\} \{3,5,6\} \{4,7\} \{4,5\} \{8\} \rangle$	$\langle \{6\} \{5\} \rangle$	Yes
$\langle \{1\} \{2\} \{3\} \{4\} \{5\} \rangle$	$\langle \{1\} \{4\} \rangle$	No
$\langle \{1\} \{2,3\} \{3,4\} \{4,5\} \rangle$	$\langle \{2\} \{3\} \{5\} \rangle$	Yes
$\langle \{1,2\} \{3\} \{2,3\} \{3,4\} \{2,4\} \{4,5\} \rangle$	$\langle \{1,2\} \{5\} \rangle$	No

02/03/2018

Introduction to Data Mining

48

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:

$x_g = 1$ (max-gap)
 $n_g = 0$ (min-gap)
 $m_s = 5$ (maximum span)
 $minsup = 60\%$

$\langle \{2\} \{5\} \rangle$ support = 40%

but

$\langle \{2\} \{3\} \{5\} \rangle$ support = 60%

Problem exists because of max-gap constraint

No such problem if max-gap is infinite

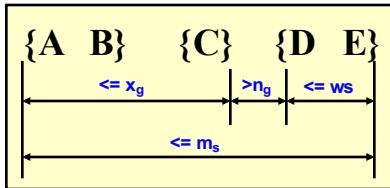
Contiguous Subsequences

- 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 at least 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 $\langle \{1\} \{2\} 3 \rangle$, $\langle \{1\} 2 \rangle \langle \{2\} \{3\} \rangle$, and $\langle \{3\} 4 \rangle \langle \{1\} 2 \rangle \langle \{2\} 3 \rangle \langle \{4\} \rangle$
 - 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, d	Sequential Pattern, s	d contains s?
$< \{2,4\} \{3,5,6\} \{4,7\} \{4,5\} \{8\} >$	$< \{3,4,5\} >$	Yes
$< \{1\} \{2\} \{3\} \{4\} \{5\} >$	$< \{1,2\} \{3,4\} >$	No
$< \{1,2\} \{2,3\} \{3,4\} \{4,5\} >$	$< \{1,2\} \{3,4\} >$	Yes

Modified Support Counting Step

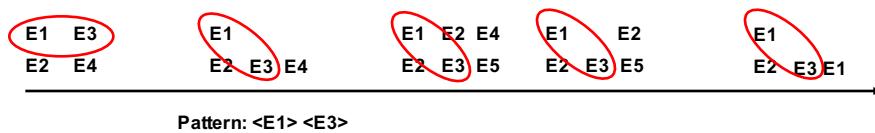
- Given a candidate sequential pattern: $<\{a, c\}>$
 - Any data sequences that contain

$< \dots \{a\} \{c\} \dots >$,
 $< \dots \{a\} \dots \{c\} \dots >$ (where $\text{time}(\{c\}) - \text{time}(\{a\}) \leq ws$)
 $< \dots \{c\} \dots \{a\} \dots >$ (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



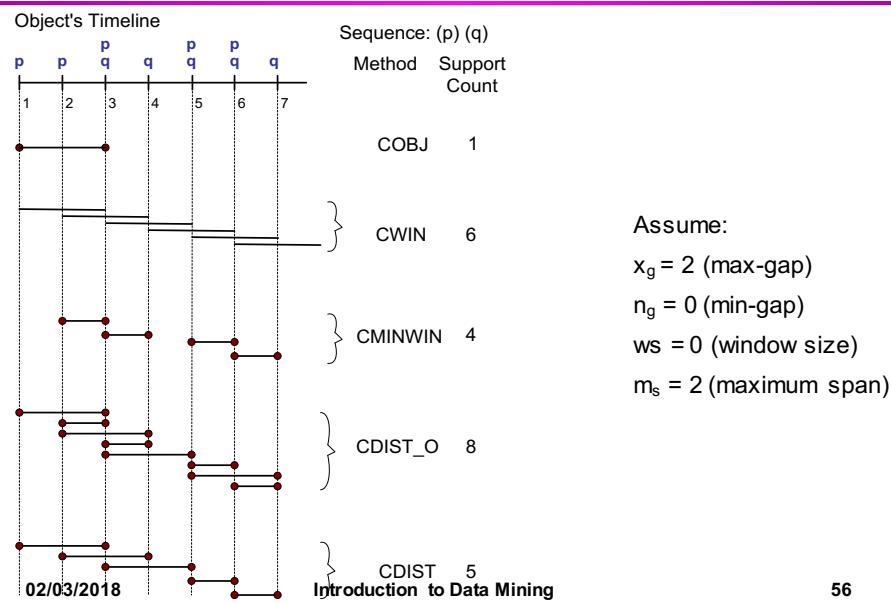
Pattern: <E1> <E3>

02/03/2018

Introduction to Data Mining

55

General Support Counting Schemes



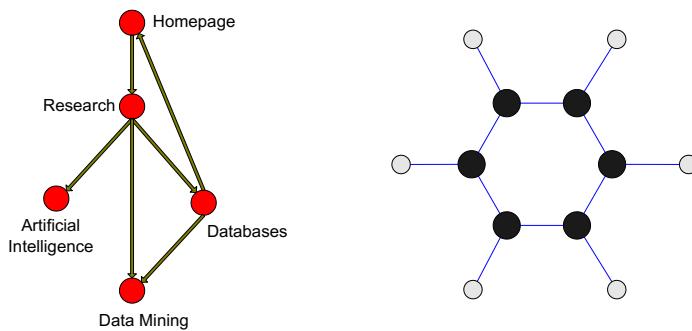
Data Mining

Association Analysis: Advanced Concepts

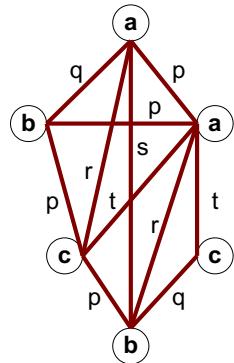
Subgraph Mining

Frequent Subgraph Mining

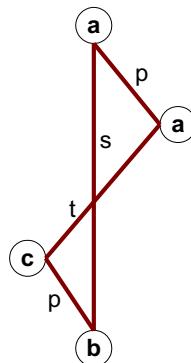
- Extends association analysis to finding frequent subgraphs
- Useful for Web Mining, computational chemistry, bioinformatics, spatial data sets, etc



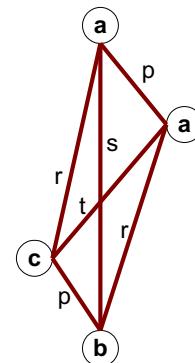
Graph Definitions



(a) Labeled Graph



(b) Subgraph

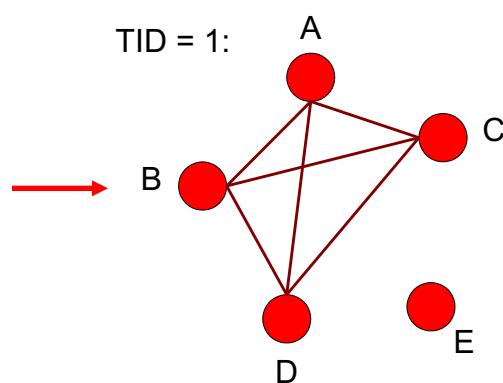


(c) Induced Subgraph

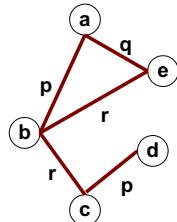
Representing Transactions as Graphs

- Each transaction is a clique of items

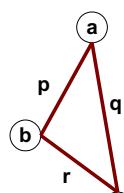
Transaction Id	Items
1	{A,B,C,D}
2	{A,B,E}
3	{B,C}
4	{A,B,D,E}
5	{B,C,D}



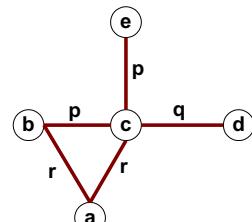
Representing Graphs as Transactions



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

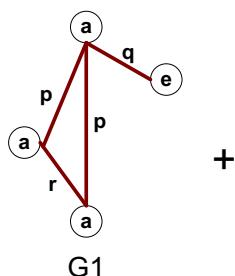
Challenges

- 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?

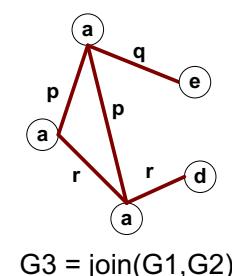
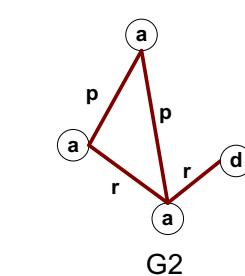
Challenges...

- 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



+



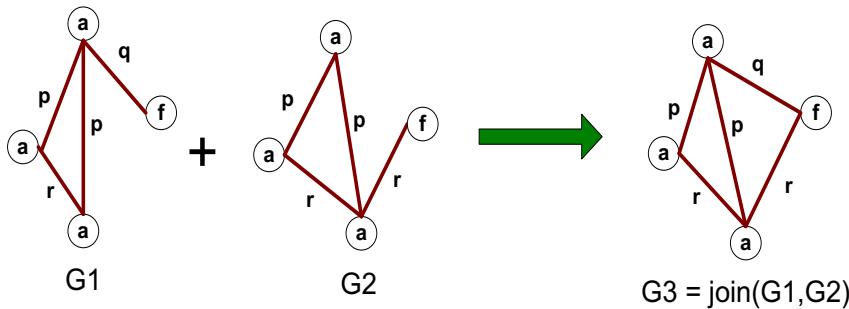
G3 = join(G1, G2)

$$M_{G1} = \begin{pmatrix} 0 & p & p & q \\ p & 0 & r & 0 \\ p & r & 0 & 0 \\ q & 0 & 0 & 0 \end{pmatrix}$$

$$M_{G2} = \begin{pmatrix} 0 & p & p & 0 \\ p & 0 & r & 0 \\ p & r & 0 & r \\ 0 & 0 & r & 0 \end{pmatrix}$$

$$M_{G3} = \begin{pmatrix} 0 & p & p & q & 0 \\ p & 0 & r & 0 & 0 \\ p & r & 0 & 0 & r \\ q & 0 & 0 & 0 & ? \\ 0 & 0 & r & ? & 0 \end{pmatrix}$$

Edge Growing



02/03/2018

Introduction to Data Mining

65

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

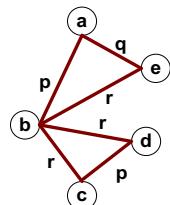
In practice, it is not as easy. There are many other issues

02/03/2018

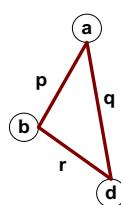
Introduction to Data Mining

66

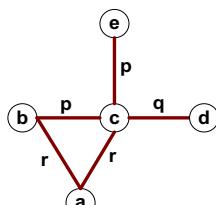
Example: Dataset



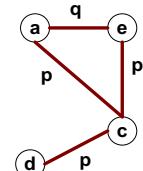
G1



G2



G3



G4

	(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
G4	0	0	0	0	0	0	...	0

02/03/2018

Introduction to Data Mining

67

Example

Minimum support count = 2

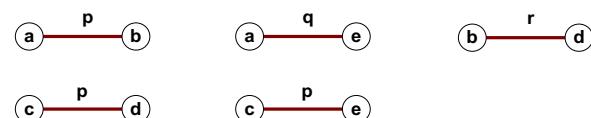
k=1

Frequent Subgraphs



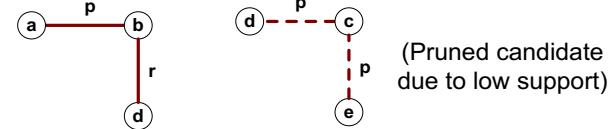
k=2

Frequent Subgraphs



k=3

Candidate Subgraphs



02/03/2018

Introduction to Data Mining

68

Candidate Generation

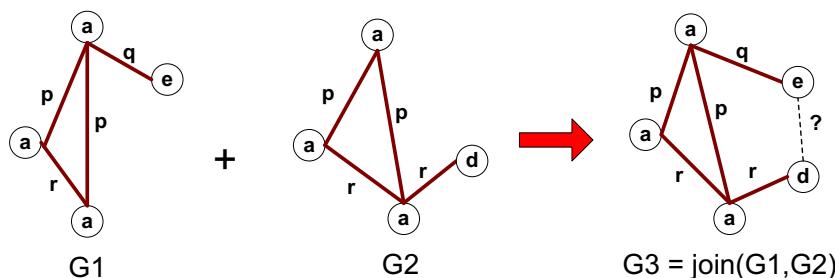
- In Apriori:

- 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

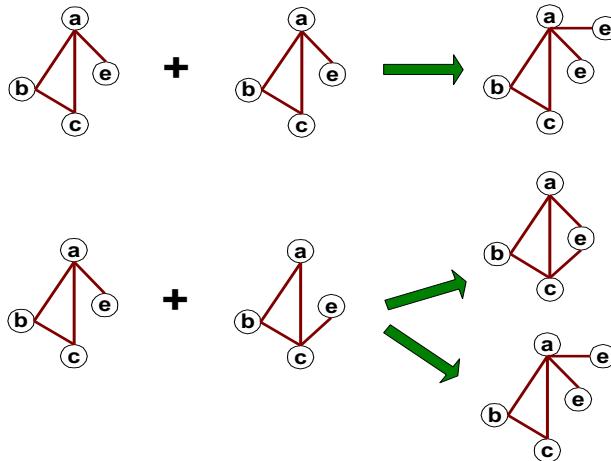
Multiplicity of Candidates (Vertex Growing)



$$M_{G_1} = \begin{pmatrix} 0 & p & p & q \\ p & 0 & r & 0 \\ p & r & 0 & 0 \\ q & 0 & 0 & 0 \end{pmatrix} \quad M_{G_2} = \begin{pmatrix} 0 & p & p & 0 \\ p & 0 & r & 0 \\ p & r & 0 & r \\ 0 & 0 & r & 0 \end{pmatrix} \quad M_{G_3} = \begin{pmatrix} 0 & p & p & 0 & q \\ p & 0 & r & 0 & 0 \\ p & r & 0 & r & 0 \\ 0 & 0 & r & 0 & ? \\ q & 0 & 0 & ? & 0 \end{pmatrix}$$

Multiplicity of Candidates (Edge growing)

- Case 1: identical vertex labels



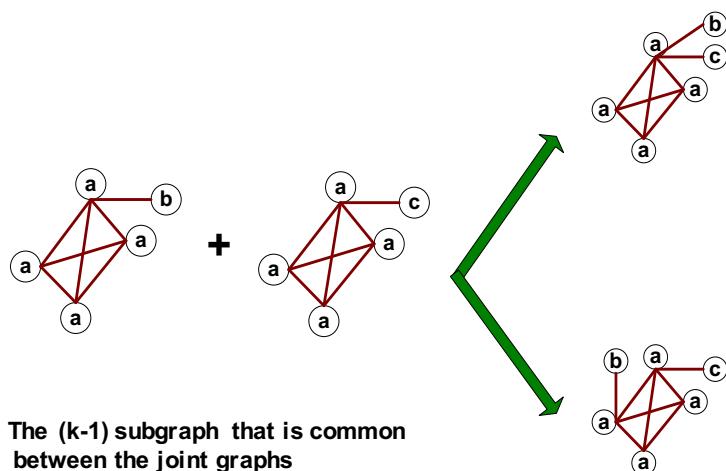
02/03/2018

Introduction to Data Mining

71

Multiplicity of Candidates (Edge growing)

- Case 2: Core contains identical labels



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

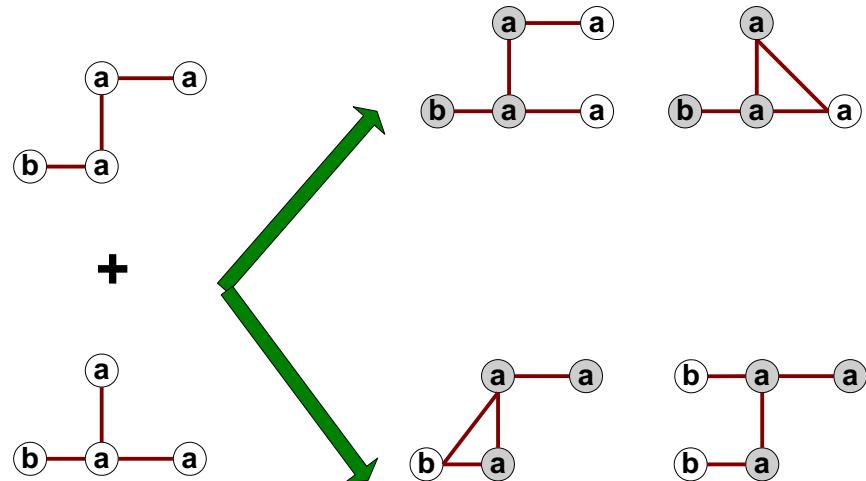
02/03/2018

Introduction to Data Mining

72

Multiplicity of Candidates (Edge growing)

- Case 3: Core multiplicity

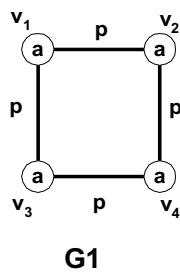


02/03/2018

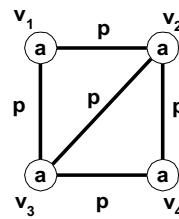
Introduction to Data Mining

73

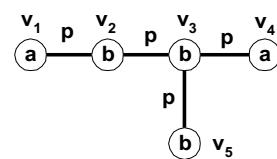
Topological Equivalence



G1



G2



G3

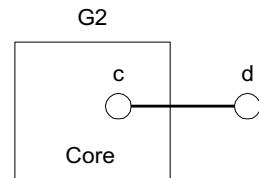
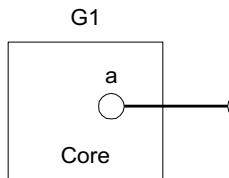
02/03/2018

Introduction to Data Mining

74

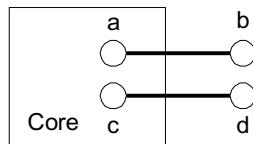
Candidate Generation by Edge Growing

- Given:



- Case 1: $a \neq c$ and $b \neq d$

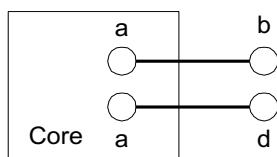
$G3 = \text{Merge}(G1, G2)$



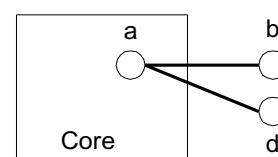
Candidate Generation by Edge Growing

- Case 2: $a = c$ and $b \neq d$

$G3 = \text{Merge}(G1, G2)$



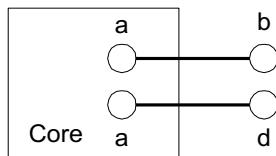
$G3 = \text{Merge}(G1, G2)$



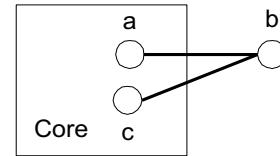
Candidate Generation by Edge Growing

- Case 3: $a \neq c$ and $b = d$

$G3 = \text{Merge}(G1, G2)$



$G3 = \text{Merge}(G1, G2)$



02/03/2018

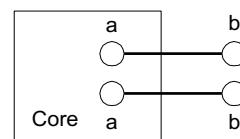
Introduction to Data Mining

77

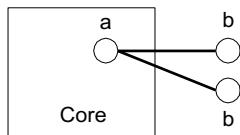
Candidate Generation by Edge Growing

- Case 4: $a = c$ and $b = d$

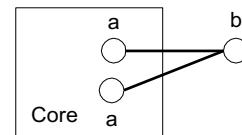
$G3 = \text{Merge}(G1, G2)$



$G3 = \text{Merge}(G1, G2)$



$G3 = \text{Merge}(G1, G2)$



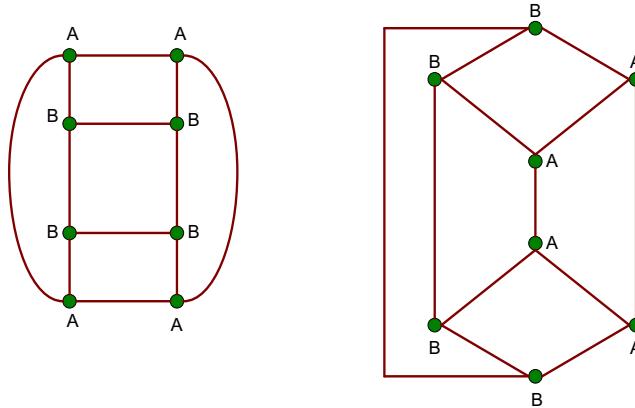
02/03/2018

Introduction to Data Mining

78

Graph Isomorphism

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



02/03/2018

Introduction to Data Mining

79

Graph Isomorphism

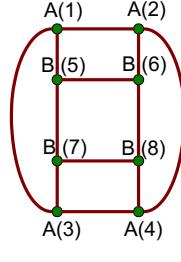
- Test for graph isomorphism is needed:
 - During candidate generation step, to determine whether a candidate has been generated
 - During candidate pruning step, to check whether its $(k-1)$ -subgraphs are frequent
 - During candidate counting, to check whether a candidate is contained within another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

02/03/2018

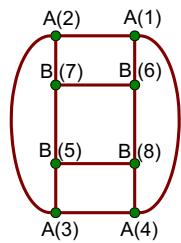
Introduction to Data Mining

80

Graph Isomorphism



	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

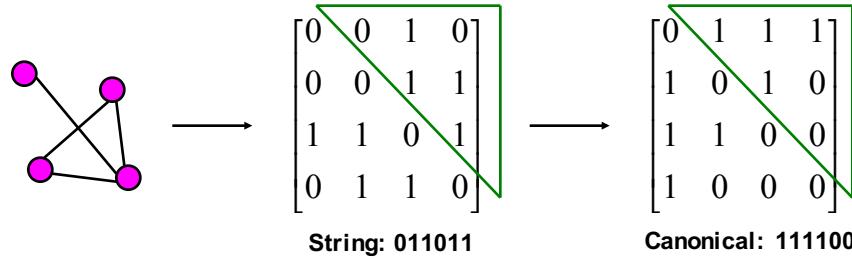
02/03/2018

Introduction to Data Mining

81

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



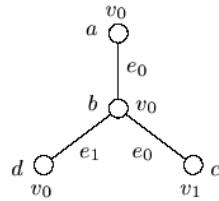
02/03/2018

Introduction to Data Mining

82

Example of Canonical Labeling (Kuramochi & Karypis, ICDM 2001)

- Graph:



- Adjacency matrix representation:

id	a	b	c	d
label	v ₀	v ₀	v ₁	v ₀
a	0	e ₀	0	0
b	e ₀	0	e ₀	e ₁
c	0	e ₀	0	0
d	0	e ₁	0	0

Example of Canonical Labeling (Kuramochi & Karypis, ICDM 2001)

- Order based on vertex degree:

id	a	c	d	b
label	v ₀	v ₁	v ₀	v ₀
partition	0	0	0	1
a	0	0	0	e ₀
c	0	0	0	e ₀
d	0	0	0	e ₁
b	e ₀	e ₀	e ₁	0

- Order based on vertex labels:

id	d	a	c	b
label	v ₀	v ₀	v ₁	v ₀
partition	0	0	1	2
d	0	0	0	e ₁
a	0	0	0	e ₀
c	0	0	0	e ₀
b	e ₁	e ₀	e ₀	0

Example of Canonical Labeling (Kuramochi & Karypis, ICDM 2001)

- Find canonical label:

	id	<i>d</i>	<i>a</i>	<i>c</i>	<i>b</i>
label		v_0	v_0	v_1	v_0
partition		0		1	2

<i>d</i>	0	0	0	e_1
<i>a</i>	0	0	0	e_0
<i>c</i>	0	0	0	e_0
<i>b</i>	e_0	e_1	e_0	0

	id	<i>a</i>	<i>d</i>	<i>c</i>	<i>b</i>
label		v_0	v_0	v_1	v_0
partition		0		1	2

<i>a</i>	0	0	0	e_0
<i>d</i>	0	0	0	e_1
<i>c</i>	0	0	0	e_0
<i>b</i>	e_0	e_1	e_0	0

0 0 0 $e_1 e_0 e_0$

>

0 0 0 $e_0 e_1 e_0$

(Canonical Label)

Data Mining Cluster Analysis: Advanced Concepts and Algorithms

Lecture Notes for Chapter 8

Introduction to Data Mining, 2nd Edition
by
Tan, Steinbach, Karpatne, Kumar

Outline

- Prototype-based
 - Fuzzy c-means
 - Mixture Model Clustering
 - Self-Organizing Maps
- Density-based
 - Grid-based clustering
 - Subspace clustering
- Graph-based
 - Chameleon
 - Jarvis-Patrick
 - Shared Nearest Neighbor (SNN)
- Characteristics of Clustering Algorithms

Hard (Crisp) vs Soft (Fuzzy) Clustering

- Hard (Crisp) vs. Soft (Fuzzy) clustering

- For soft clustering allow points to belong to more than one cluster

- For K-means, generalize objective function

$$SSE = \sum_{j=1}^k \sum_{i=1}^m w_{ij} dist(\mathbf{x}_i, \mathbf{c}_j)^2 \quad \sum_{j=1}^k w_{ij} = 1$$

w_{ij} : weight with which object \mathbf{x}_i belongs to cluster \mathbf{c}_j

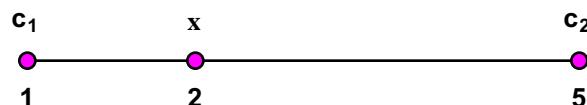
- To minimize SSE, repeat the following steps:

- ◆ Fix \mathbf{c}_j and determine w_{ij} (cluster assignment)

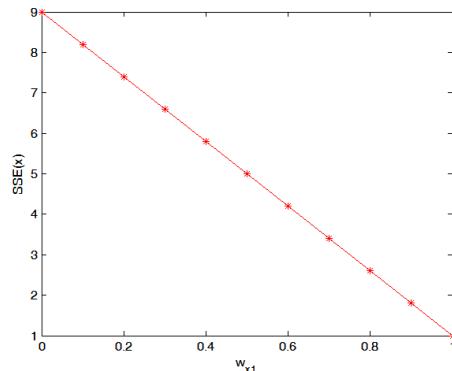
- ◆ Fix w_{ij} and recompute \mathbf{c}_j

- Hard clustering: $w_{ij} \in \{0,1\}$

Soft (Fuzzy) Clustering: Estimating Weights



$$\begin{aligned} SSE(x) &= w_{x1}(2-1)^2 + w_{x2}(5-2)^2 \\ &= w_{x1} + 9w_{x2} \end{aligned}$$



$SSE(x)$ is minimized when $w_{x1} = 1, w_{x2} = 0$

Fuzzy C-means

- Objective function

$$SSE = \sum_{j=1}^k \sum_{i=1}^m w_{ij}^p dist(x_i, c_j)^2 \quad \sum_{j=1}^k w_{ij} = 1$$

p: fuzzifier ($p > 1$)

- ◆ w_{ij} : weight with which object x_i belongs to cluster c_j
- ◆ p: a power for the weight not a superscript and controls how "fuzzy" the clustering is

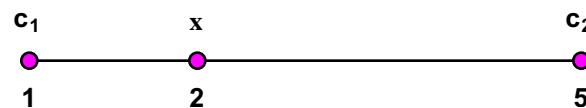
— To minimize objective function, repeat the following:

- ◆ Fix c_j and determine w_{ij}
- ◆ Fix w_{ij} and recompute c

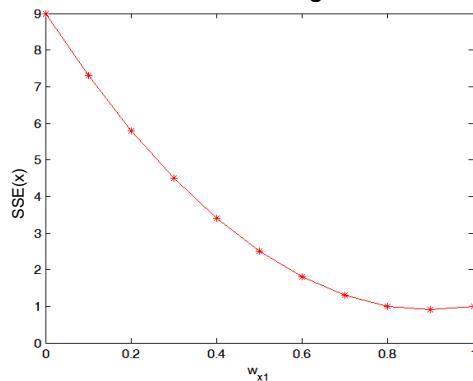
— Fuzzy c-means clustering: $w_{ij} \in [0, 1]$

Bezdek, James C. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981.

Fuzzy C-means



$$\begin{aligned} SSE(x) &= w_{x1}^2 (2-1)^2 + w_{x2}^2 (5-2)^2 \\ &= w_{x1}^2 + 9w_{x2}^2 \end{aligned}$$



$SSE(x)$ is minimized when $w_{x1} = 0.9$, $w_{x2} = 0.1$

Fuzzy C-means

- Objective function:

$$SSE = \sum_{j=1}^k \sum_{i=1}^m w_{ij}^p dist(\mathbf{x}_i, \mathbf{c}_j)^2 \quad \sum_{j=1}^k w_{ij} = 1$$

p: fuzzifier ($p > 1$)

- Initialization: choose the weights w_{ij} randomly

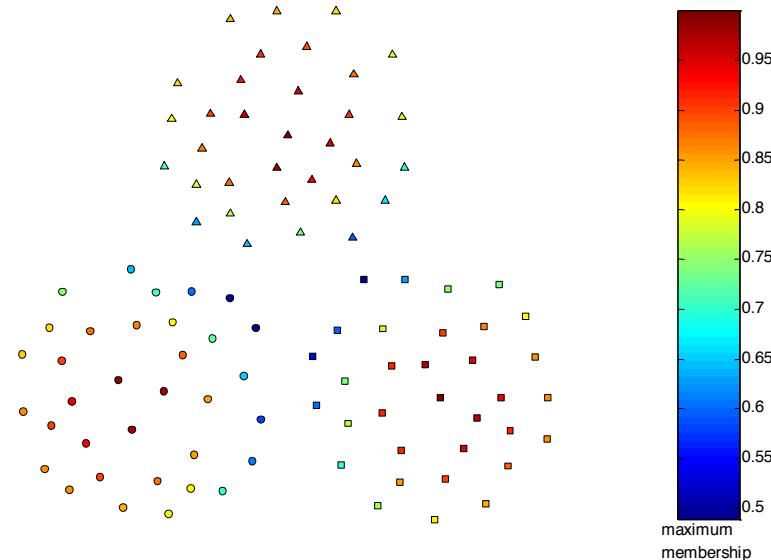
- Repeat:

- Update centroids: $\mathbf{c}_j = \sum_{i=1}^m w_{ij} \mathbf{x}_i / \sum_{i=1}^m w_{ij}$

- Update weights:

$$w_{ij} = (1/dist(\mathbf{x}_i, \mathbf{c}_j)^2)^{\frac{1}{p-1}} / \sum_{j=1}^k (1/dist(\mathbf{x}_i, \mathbf{c}_j)^2)^{\frac{1}{p-1}}$$

Fuzzy K-means Applied to Sample Data



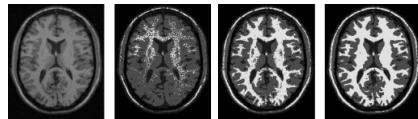
An Example Application: Image Segmentation

- Modified versions of fuzzy c-means have been used for image segmentation

- Especially fMRI images (functional magnetic resonance images)

● References

- Gong, Maoguo, Yan Liang, Jiao Shi, Wenping Ma, and Jingjing Ma. "Fuzzy c-means clustering with local information and kernel metric for image segmentation." *Image Processing, IEEE Transactions on* 22, no. 2 (2013): 573-584.



From left to right: original images, fuzzy c-means, EM, BCFCM

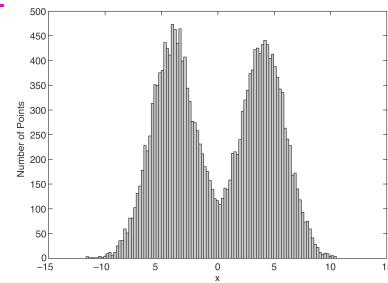
- Ahmed, Mohamed N., Sameh M. Yamany, Nevin Mohamed, Aly A. Farag, and Thomas Moriarty. "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data." *Medical Imaging, IEEE Transactions on* 21, no. 3 (2002): 193-199.

Hard (Crisp) vs Soft (Probabilistic) Clustering

- Idea is to model the set of data points as arising from a mixture of distributions
 - Typically, normal (Gaussian) distribution is used
 - But other distributions have been very profitably used
- Clusters are found by estimating the parameters of the statistical distributions
 - Can use a k-means like algorithm, called the Expectation-Maximization (EM) algorithm, to estimate these parameters
 - ◆ Actually, k-means is a special case of this approach
 - Provides a compact representation of clusters
 - The probabilities with which point belongs to each cluster provide a functionality similar to fuzzy clustering.

Probabilistic Clustering: Example

- Informal example: consider modeling the points that generate the following histogram.
- Looks like a combination of two normal (Gaussian) distributions
- Suppose we can estimate the mean and standard deviation of each normal distribution.
 - This completely describes the two clusters
 - We can compute the probabilities with which each point belongs to each cluster
 - Can assign each point to the cluster (distribution) for which it $\text{prob}(x_i|\Theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ is most probable.



02/14/2018

Introduction to Data Mining, 2nd Edition

11

Probabilistic Clustering: EM Algorithm

Initialize the parameters

Repeat

 For each point, compute its probability under each distribution

 Using these probabilities, update the parameters of each distribution

Until there is no change

- Very similar to K-means
- Consists of assignment and update steps
- Can use random initialization
 - Problem of local minima
- For normal distributions, typically use K-means to initialize
- If using normal distributions, can find elliptical as well as spherical shapes.

02/14/2018

Introduction to Data Mining, 2nd Edition

12

Probabilistic Clustering: Updating Centroids

Update formula for weights assuming an estimate for statistical parameters

$$c_j = \sum_{i=1}^m x_i p(C_j | x_i) / \sum_{i=1}^m p(C_j | x_i)$$

x_i is a data point
 C_j is a cluster
 c_j is a centroid

- Very similar to the fuzzy k-means formula

- Weights are probabilities
- Weights are not raised to a power
- Probabilities calculated using Bayes rule: $p(C_j | x_i) = \frac{p(x_i | C_j)p(C_j)}{\sum_{l=1}^k p(x_i | C_l)p(C_l)}$

- Need to assign weights to each cluster

- Weights may not be equal
- Similar to prior probabilities

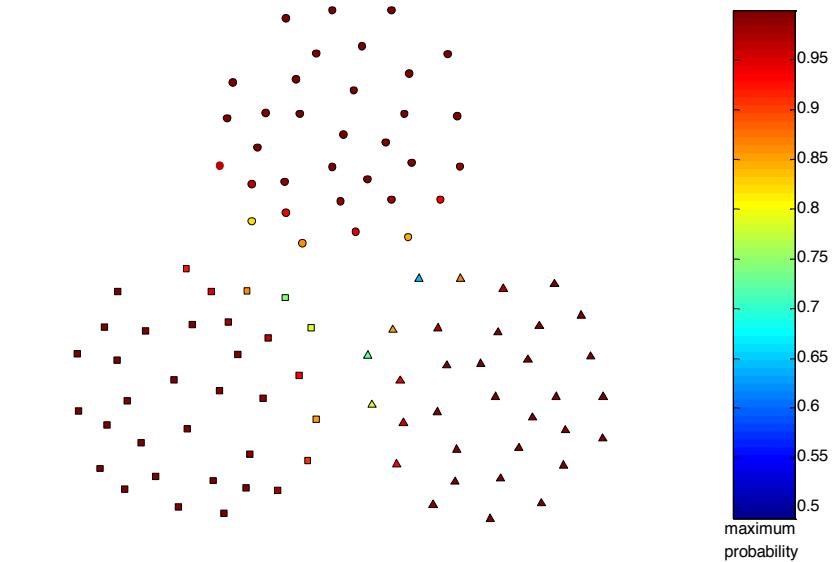
- Can be estimated: $p(C_j) = \frac{1}{m} \sum_{i=1}^m p(C_j | x_i)$

More Detailed EM Algorithm

Algorithm 9.2 EM algorithm.

- 1: Select an initial set of model parameters.
(As with K-means, this can be done randomly or in a variety of ways.)
- 2: **repeat**
- 3: **Expectation Step** For each object, calculate the probability that each object belongs to each distribution, i.e., calculate $\text{prob}(\text{distribution } j | x_i, \Theta)$.
- 4: **Maximization Step** Given the probabilities from the expectation step, find the new estimates of the parameters that maximize the expected likelihood.
- 5: **until** The parameters do not change.
(Alternatively, stop if the change in the parameters is below a specified threshold.)

Probabilistic Clustering Applied to Sample Data

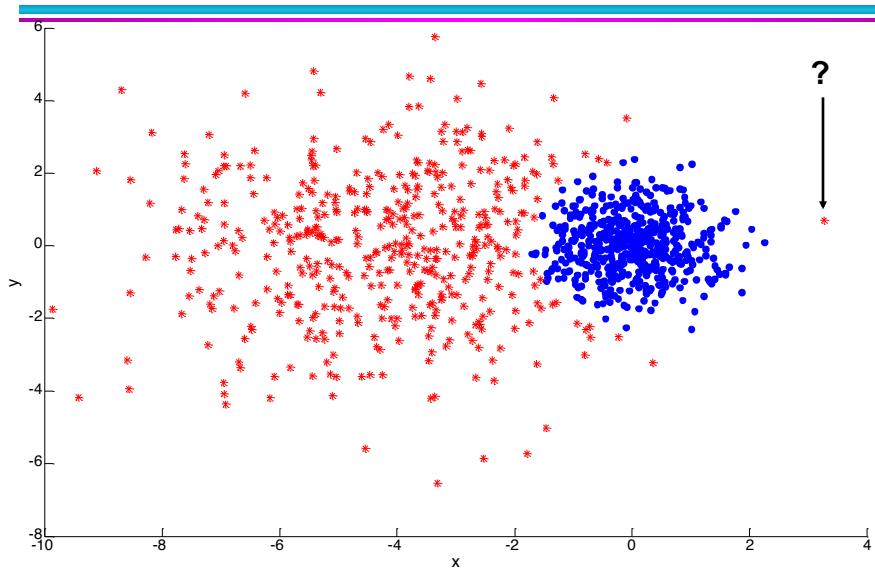


02/14/2018

Introduction to Data Mining, 2nd Edition

15

Probabilistic Clustering: Dense and Sparse Clusters



02/14/2018

Introduction to Data Mining, 2nd Edition

16

Problems with EM

- Convergence can be slow
- Only guarantees finding local maxima
- Makes some significant statistical assumptions
- Number of parameters for Gaussian distribution grows as $O(d^2)$, d the number of dimensions
 - Parameters associated with covariance matrix
 - K-means only estimates cluster means, which grow as $O(d)$

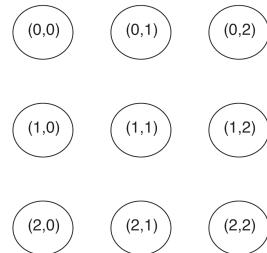
Alternatives to EM

- Method of moments / Spectral methods
 - ICML 2014 workshop bibliography
<https://sites.google.com/site/momentsicml2014/bibliography>
- Markov chain Monte Carlo (MCMC)
- Other approaches

SOM: Self-Organizing Maps

● Self-organizing maps (SOM)

- Centroid based clustering scheme
- Like K-means, a fixed number of clusters are specified
- However, the spatial relationship of clusters is also specified, typically as a grid
- Points are considered one by one
- Each point is assigned to the closest centroid
- Other centroids are updated based on their nearness to the closest centroid



Kohonen, Teuvo, and Self-Organizing Maps. "Springer series in information sciences." *Self-organizing maps* 30 (1995).

SOM: Self-Organizing Maps

Algorithm 9.3 Basic SOM Algorithm.

-
- ```
1: Initialize the centroids.
2: repeat
3: Select the next object.
4: Determine the closest centroid to the object.
5: Update this centroid and the centroids that are close, i.e., in a specified neighborhood.
6: until The centroids don't change much or a threshold is exceeded.
7: Assign each object to its closest centroid and return the centroids and clusters.
```
- 

## ● Updates are weighted by distance

- Centroids farther away are affected less

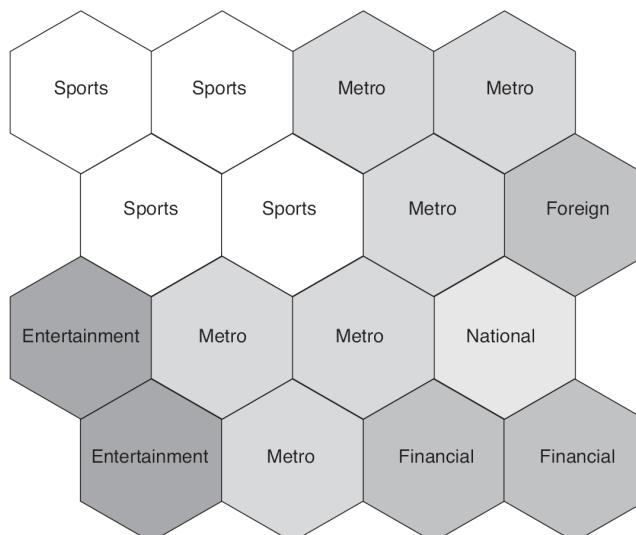
## ● The impact of the updates decreases with each time

- At some point the centroids will not change much

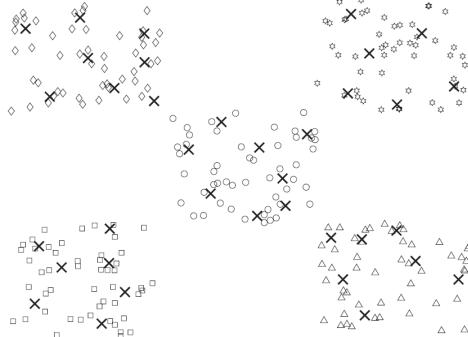
## SOM: Self-Organizing Maps

- SOM can be viewed as a type of dimensionality reduction
- If a two-dimensional grid is used, the results can be visualized

## SOM Clusters of LA Times Document Data



## Another SOM Example: 2D Points



(a) Distribution of SOM reference vectors (X's) for a two-dimensional point set.

|         |         |         |          |          |          |
|---------|---------|---------|----------|----------|----------|
| diamond | diamond | diamond | hexagon  | hexagon  | hexagon  |
| diamond | diamond | diamond | circle   | hexagon  | hexagon  |
| diamond | diamond | circle  | circle   | circle   | hexagon  |
| square  | square  | circle  | circle   | triangle | triangle |
| square  | square  | circle  | circle   | triangle | triangle |
| square  | square  | square  | triangle | triangle | triangle |

(b) Classes of the SOM centroids.

## Issues with SOM

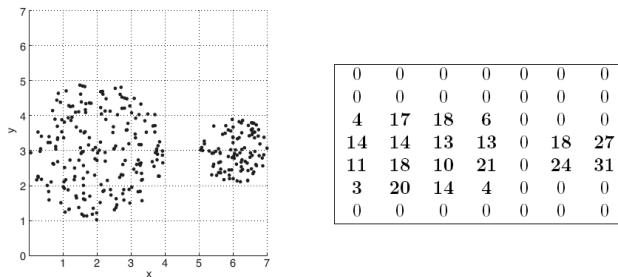
- Computational complexity
- Locally optimal solution
- Grid is somewhat arbitrary

## Grid-based Clustering

- A type of density-based clustering

**Algorithm 9.4** Basic grid-based clustering algorithm.

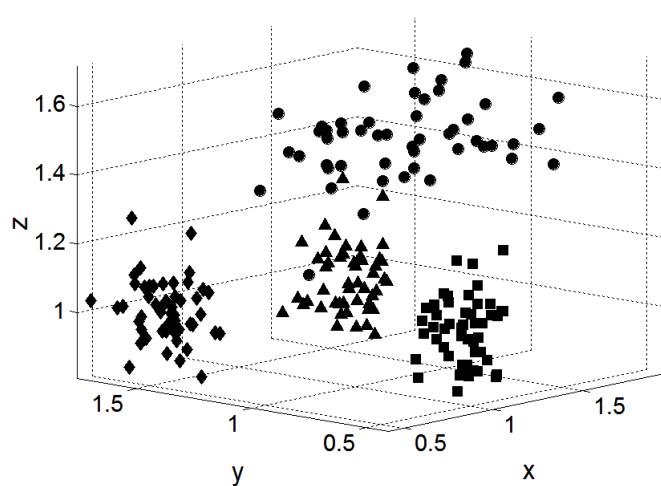
- 1: Define a set of grid cells.
- 2: Assign objects to the appropriate cells and compute the density of each cell.
- 3: Eliminate cells having a density below a specified threshold,  $\tau$ .
- 4: Form clusters from contiguous (adjacent) groups of dense cells.



## Subspace Clustering

- Until now, we found clusters by considering all of the attributes
- Some clusters may involve only a subset of attributes, i.e., subspaces of the data
  - Example:
    - ◆ When k-means is used to find document clusters, the resulting clusters can typically be characterized by 10 or so terms

## Example

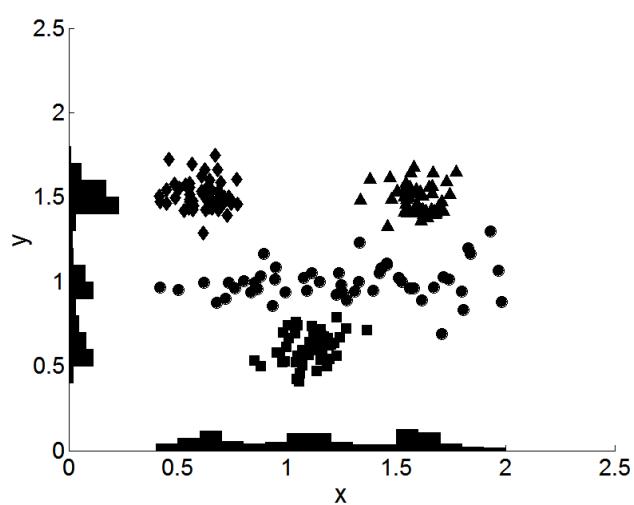


02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

27

## Example

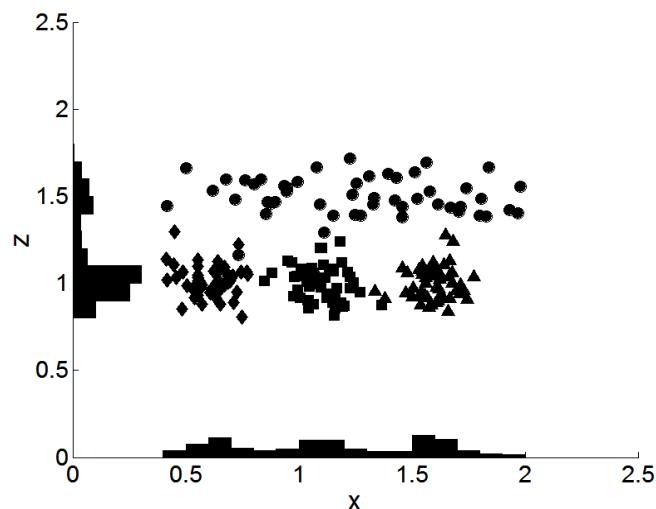


02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

28

## Example

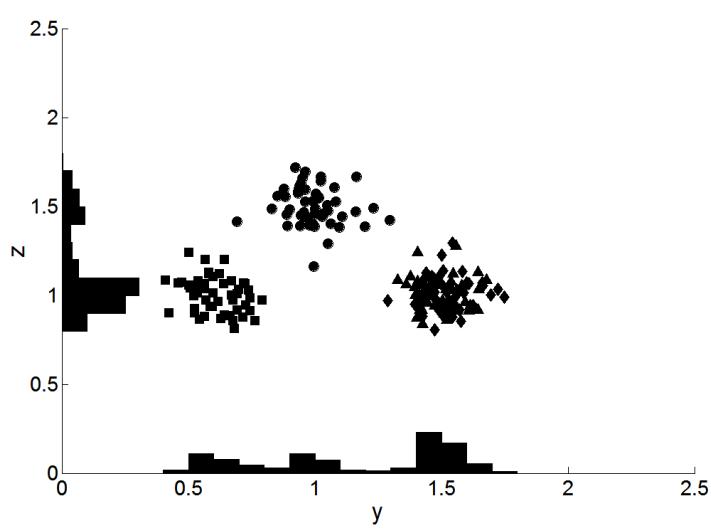


02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

29

## Example



02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

30

## Clique Algorithm - Overview

- A grid-based clustering algorithm that methodically finds subspace clusters
  - Partitions the data space into rectangular units of equal volume
  - Measures the density of each unit by the fraction of points it contains
  - A unit is dense if the fraction of overall points it contains is above a user specified threshold,  $\tau$
  - A cluster is a group of collections of contiguous (touching) dense units

## Clique Algorithm

- It is impractical to check each volume unit to see if it is dense since there is exponential number of such units
- **Monotone property of density-based clusters:**
  - If a set of points forms a density based cluster in  $k$  dimensions, then the same set of points is also part of a density based cluster in all possible subsets of those dimensions
- Very similar to Apriori algorithm
- Can find overlapping clusters

## Clique Algorithm

---

### Algorithm 9.5 CLIQUE.

- 1: Find all the dense areas in the one-dimensional spaces corresponding to each attribute. This is the set of dense one-dimensional cells.
  - 2:  $k \leftarrow 2$
  - 3: **repeat**
  - 4:   Generate all candidate dense  $k$ -dimensional cells from dense  $(k-1)$ -dimensional cells.
  - 5:   Eliminate cells that have fewer than  $\xi$  points.
  - 6:    $k \leftarrow k + 1$
  - 7: **until** There are no candidate dense  $k$ -dimensional cells.
  - 8: Find clusters by taking the union of all adjacent, high-density cells.
  - 9: Summarize each cluster using a small set of inequalities that describe the attribute ranges of the cells in the cluster.
- 

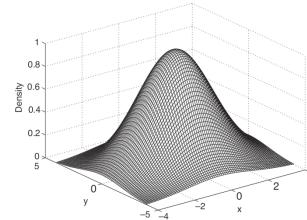
## Limitations of Clique

- Time complexity is exponential in number of dimensions
  - Especially if “too many” dense units are generated at lower stages
- May fail if clusters are of widely differing densities, since the threshold is fixed
  - Determining appropriate threshold and unit interval length can be challenging

## Denclue (DENsity CLUstering)

- Based on the notion of kernel-density estimation
  - Contribution of each point to the density is given by an influence or kernel function

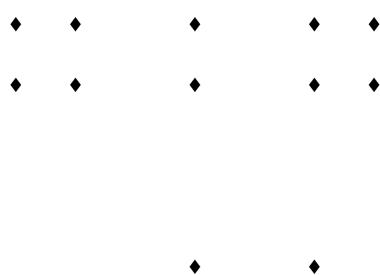
$$K(y) = e^{-\text{distance}(\mathbf{x}, \mathbf{y})^2 / 2\sigma^2}$$



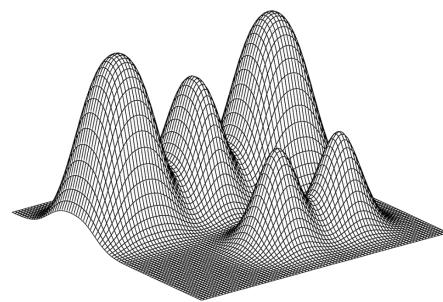
Formula and plot of Gaussian Kernel

- Overall density is the sum of the contributions of all points

## Example of Density from Gaussian Kernel

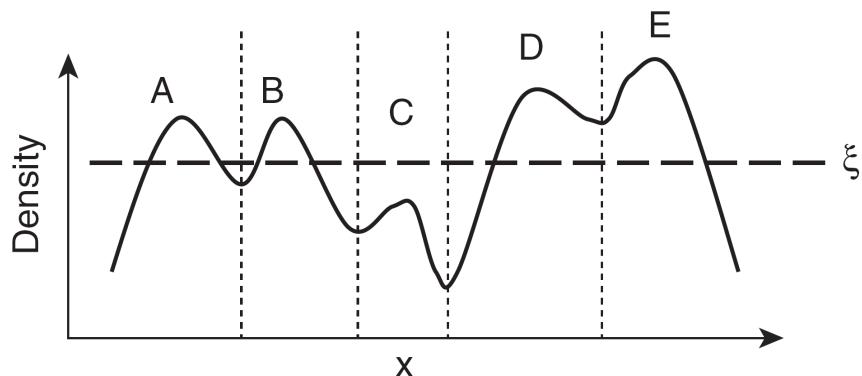


Set of 12 points.



Overall density—surface plot.

## DENCLUE Algorithm



02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

37

## DENCLUE Algorithm

- Find the density function
- Identify local maxima (density attractors)
- Assign each point to the density attractor
  - Follow direction of maximum increase in density
- Define clusters as groups consisting of points associated with density attractor
- Discard clusters whose density attractor has a density less than a user specified minimum,  $\xi$
- Combine clusters connected by paths of points that are connected by points with density above  $\xi$

02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

38

## Graph-Based Clustering: General Concepts

- 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 in graph terms
- In the simplest case, clusters are connected components in the graph.

## CURE Algorithm: Graph-Based Clustering

- Agglomerative hierarchical clustering algorithms vary in terms of how the proximity of two clusters are computed
  - ◆ MIN (single link)
    - susceptible to noise/outliers
  - ◆ MAX (complete link)/GROUP AVERAGE/Centroid/Ward's:
    - may not work well with non-globular clusters
- CURE algorithm tries to handle both problems

## CURE Algorithm

- Represents a cluster using multiple representative points
  - Representative points are found by selecting a constant number of points from a cluster
    - ◆ First representative point is chosen to be the point furthest from the center of the cluster
    - ◆ Remaining representative points are chosen so that they are farthest from all previously chosen points

## CURE Algorithm

- “Shrink” representative points toward the center of the cluster by a factor,  $\alpha$

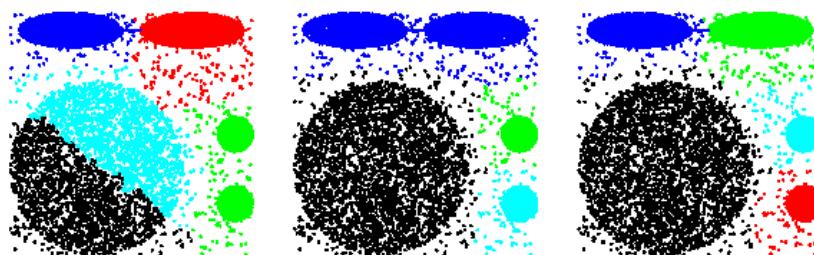


- Shrinking representative points toward the center helps avoid problems with noise and outliers
- Cluster similarity is the similarity of the closest pair of representative points from different clusters

## CURE Algorithm

- Uses agglomerative hierarchical scheme to perform clustering;
  - $\alpha = 0$ : similar to centroid-based
  - $\alpha = 1$ : somewhat similar to single-link
- CURE is better able to handle clusters of arbitrary shapes and sizes

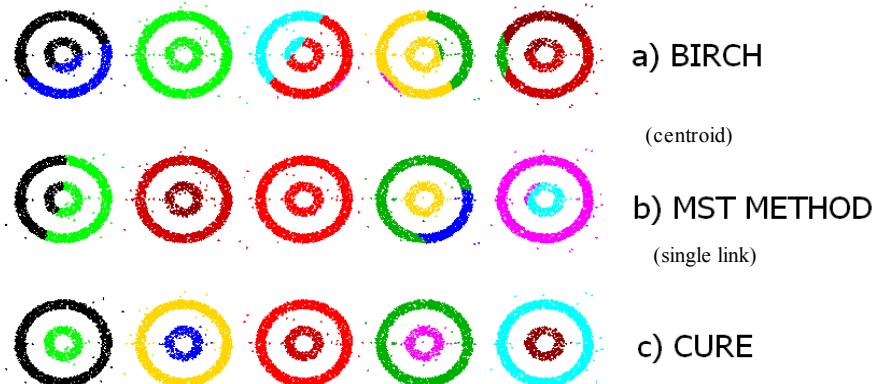
## Experimental Results: CURE



a) BIRCH      b) MST METHOD    c) CURE

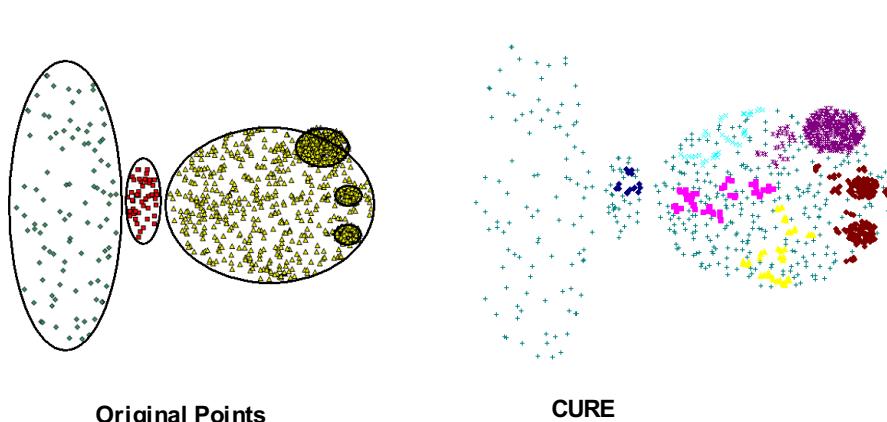
Picture from *CURE*, Guha, Rastogi, Shim.

## Experimental Results: CURE



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

## CURE Cannot Handle Differing Densities



## Graph-Based Clustering: Chameleon

---

- Based on several key ideas
  - Sparsification of the proximity graph
  - Partitioning the data into clusters that are relatively pure subclusters of the “true” clusters
  - Merging based on preserving characteristics of clusters

## 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 ...

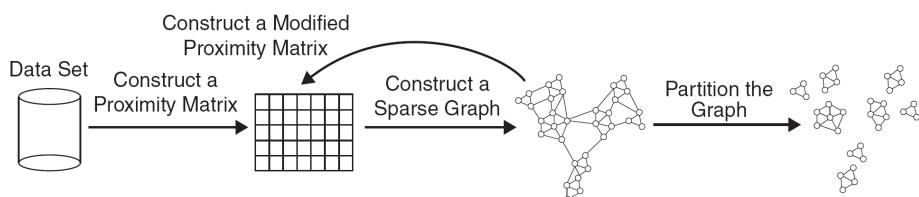
- Clustering may work better
  - Sparsification techniques keep the connections to the most similar (nearest) neighbors of a point while breaking the connections to less similar points.
  - The nearest neighbors of a point tend to belong to the same class as the point itself.
  - This reduces the impact 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

02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

49

## Sparsification in the Clustering Process



02/14/2018

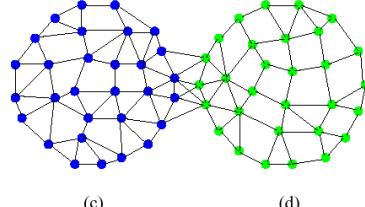
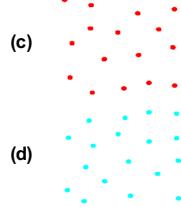
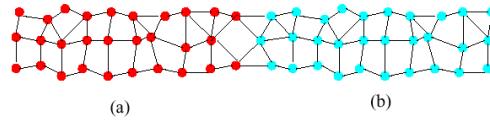
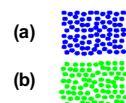
Introduction to Data Mining, 2<sup>nd</sup> Edition

50

## 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 properties are 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*



## Relative Interconnectivity

- **Relative Interconnectivity (RI)** is the absolute interconnectivity of two clusters normalized by the internal connectivity of the clusters. Two clusters are combined if the points in the resulting cluster are almost as strongly connected as points in each of the original clusters. Mathematically,

$$RI = \frac{EC(C_i, C_j)}{\frac{1}{2}(EC(C_i) + EC(C_j))}, \quad (9.18)$$

where  $EC(C_i, C_j)$  is the sum of the edges (of the  $k$ -nearest neighbor graph) that connect clusters  $C_i$  and  $C_j$ ;  $EC(C_i)$  is the minimum sum of the cut edges if we bisect cluster  $C_i$ ; and  $EC(C_j)$  is the minimum sum of the cut edges if we bisect cluster  $C_j$ .

## Relative Closeness

- **Relative Closeness (RC)** is the absolute closeness of two clusters normalized by the internal closeness of the clusters. Two clusters are combined only if the points in the resulting cluster are almost as close to each other as in each of the original clusters. Mathematically,

$$RC = \frac{\bar{S}_{EC}(C_i, C_j)}{\frac{m_i}{m_i+m_j} \bar{S}_{EC}(C_i) + \frac{m_j}{m_i+m_j} \bar{S}_{EC}(C_j)}, \quad (9.17)$$

where  $m_i$  and  $m_j$  are the sizes of clusters  $C_i$  and  $C_j$ , respectively,  $\bar{S}_{EC}(C_i, C_j)$  is the average weight of the edges (of the  $k$ -nearest neighbor graph) that connect clusters  $C_i$  and  $C_j$ ;  $\bar{S}_{EC}(C_i)$  is the average weight of edges if we bisect cluster  $C_i$ ; and  $\bar{S}_{EC}(C_j)$  is the average weight of edges if we bisect cluster  $C_j$ . ( $EC$  stands for edge cut.)

## 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)

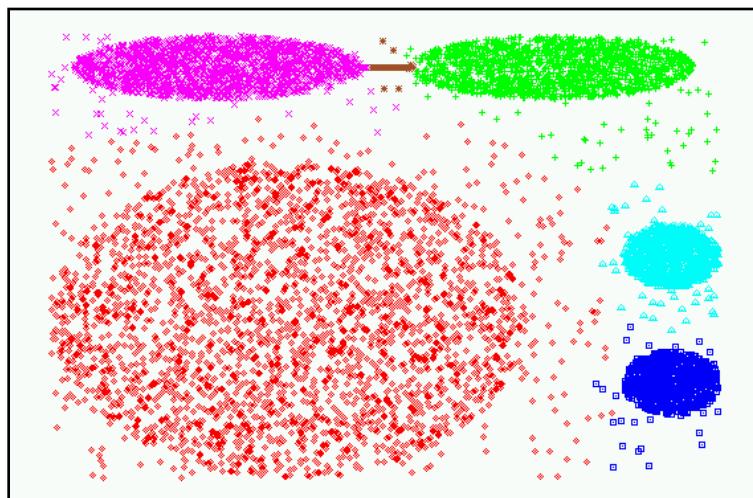
### ● Phase 1: Use a multilevel graph partitioning algorithm on the graph to find a large number of clusters of well-connected vertices

- Each cluster should contain mostly points from one “true” cluster, i.e., be a sub-cluster of a “real” cluster

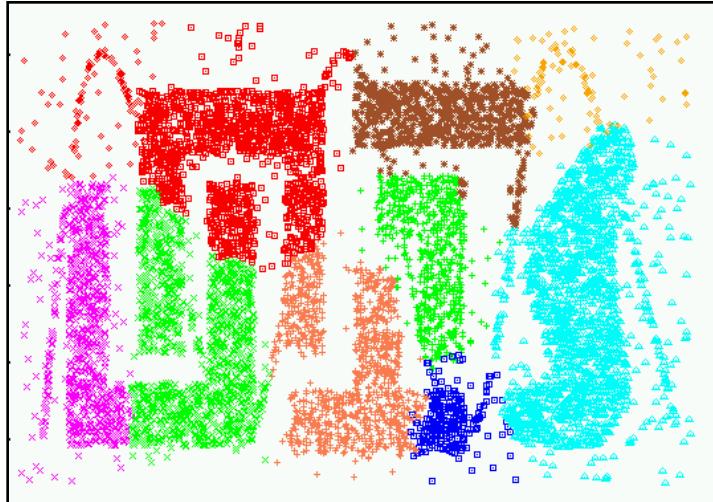
## Chameleon: Steps ...

- **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: CURE (*10 clusters*)

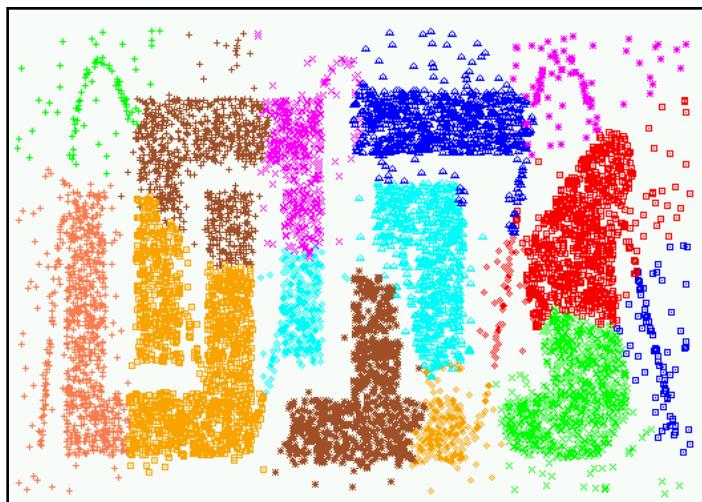


02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

59

## Experimental Results: CURE (*15 clusters*)

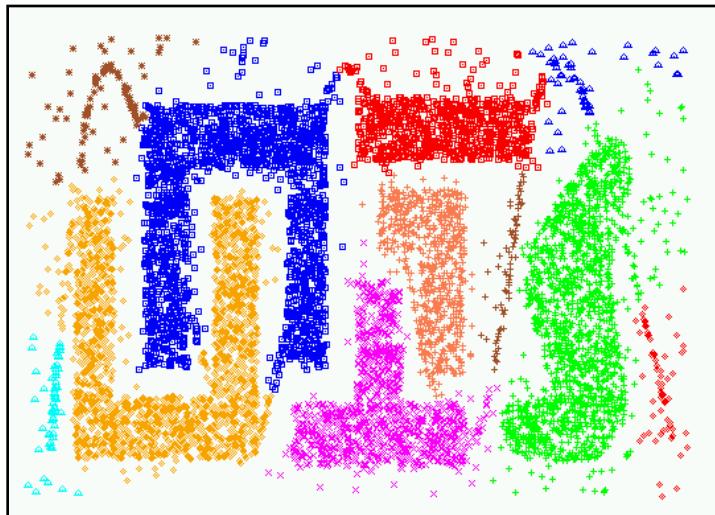


02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

60

## Experimental Results: CHAMELEON

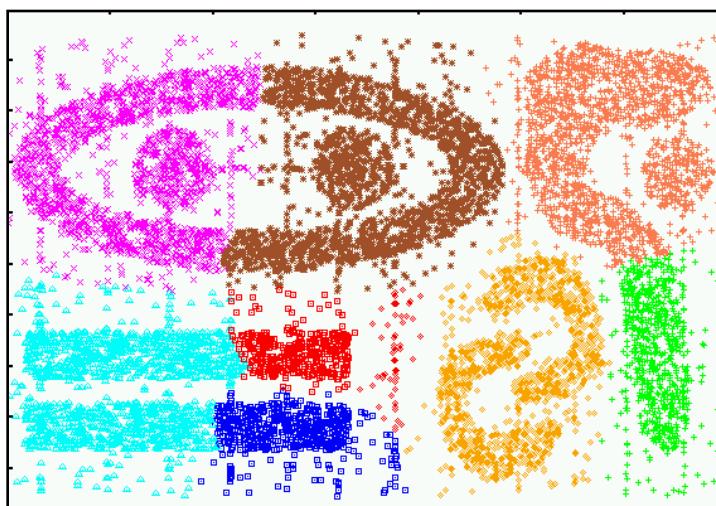


02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

61

## Experimental Results: CURE (9 clusters)

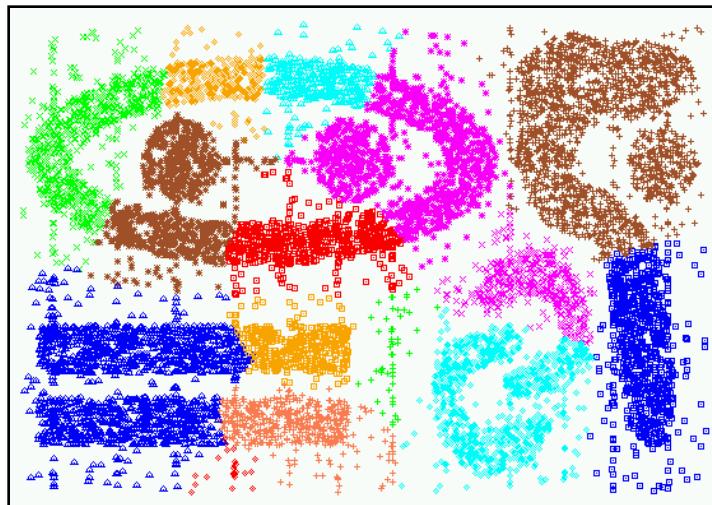


02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

62

## Experimental Results: CURE (15 clusters)

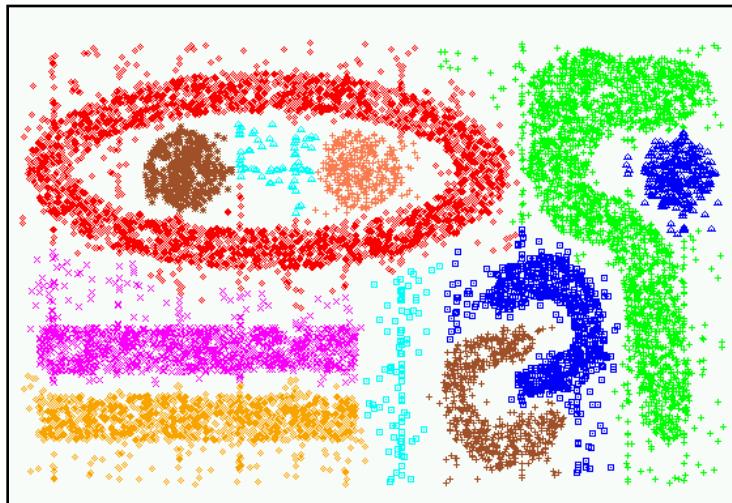


02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

63

## Experimental Results: CHAMELEON



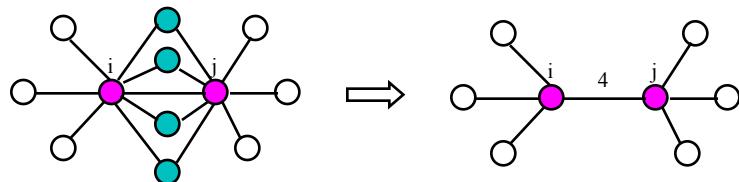
02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

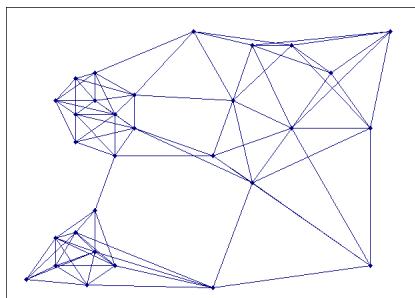
64

## Graph-Based Clustering: SNN Approach

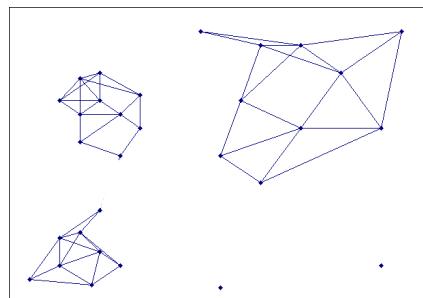
Shared Nearest Neighbor (SNN) graph: the weight of an edge is the number of shared neighbors between vertices given that the vertices are connected



## Creating the SNN Graph



Sparse Graph



Shared Near Neighbor Graph

Link weights are similarities  
between neighboring points

Link weights are number of  
Shared Nearest Neighbors

## Jarvis-Patrick Clustering

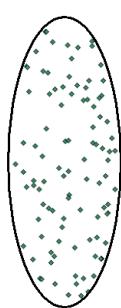
- First, the k-nearest neighbors of all points are found
  - In graph terms this can be regarded as breaking all but the k strongest links from a point to other points in the proximity graph
- A pair of points is put in the same cluster if
  - any two points share more than T neighbors and
  - the two 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

02/14/2018

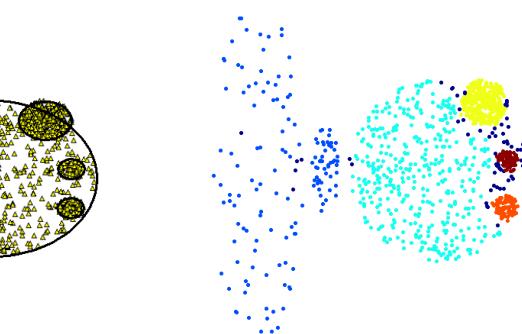
Introduction to Data Mining, 2<sup>nd</sup> Edition

67

## When Jarvis-Patrick Works Reasonably Well



Original Points



Jarvis Patrick Clustering

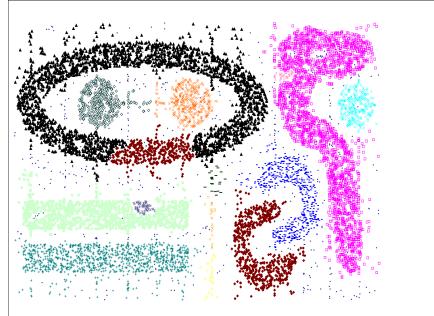
6 shared neighbors out of 20

02/14/2018

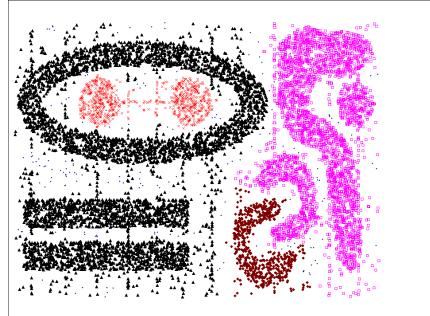
Introduction to Data Mining, 2<sup>nd</sup> Edition

68

## When Jarvis-Patrick Does NOT Work Well



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



Threshold of  $T - 1$

## SNN Density-Based Clustering

- Combines:
  - Graph based clustering (similarity definition based on number of shared nearest neighbors)
  - Density based clustering (DBSCAN-like approach)
  
- SNN density measures whether a point is surrounded by similar points (with respect to its nearest neighbors)

## SNN Clustering Algorithm

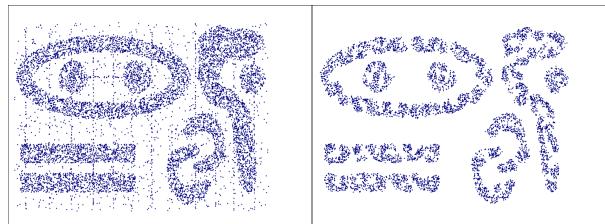
1. **Compute the similarity matrix**  
This corresponds to a similarity graph with data points for nodes and edges whose weights are the similarities between data points
2. **Sparsify the similarity matrix by keeping only the  $k$  most similar neighbors**  
This corresponds to only keeping the  $k$  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

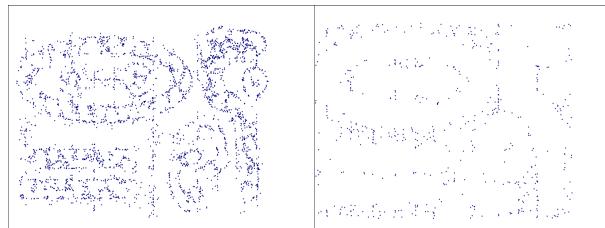
(Note that steps 4-8 are DBSCAN)

## SNN Density



a) All Points

b) High SNN Density



c) Medium SNN Density

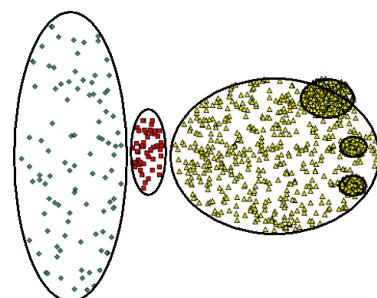
d) Low SNN Density

02/14/2018

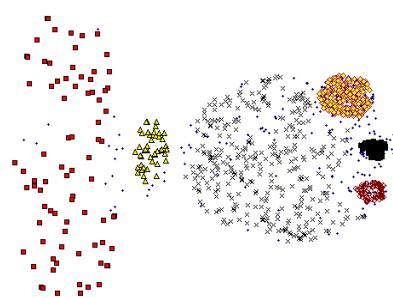
Introduction to Data Mining, 2<sup>nd</sup> Edition

73

## SNN Clustering Can Handle Differing Densities



Original Points



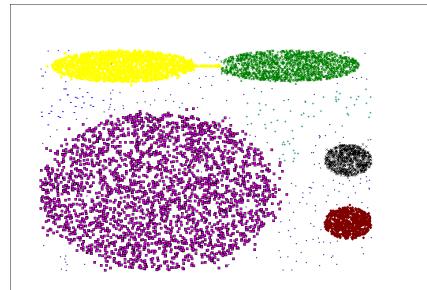
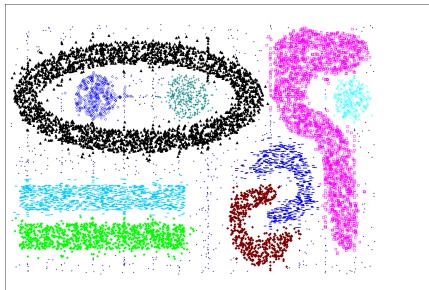
SNN Clustering

02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

74

## SNN Clustering Can Handle Other Difficult Situations

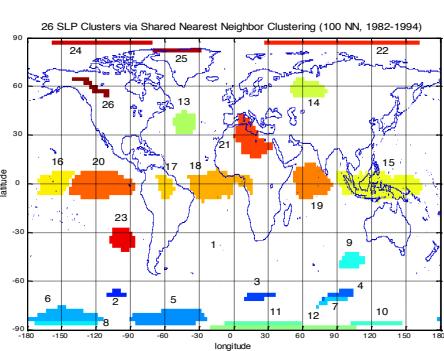


02/14/2018

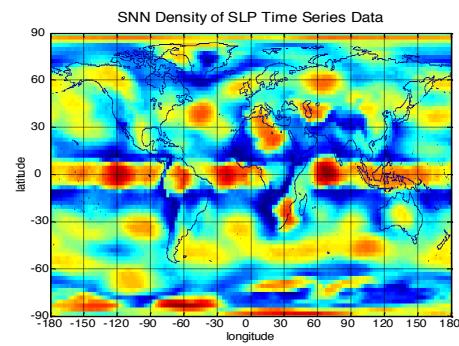
Introduction to Data Mining, 2<sup>nd</sup> Edition

75

## Finding Clusters of Time Series In Spatio-Temporal Data



SNN Clusters of SLP.



02/14/2018

Introduction to Data Mining, 2<sup>nd</sup> Edition

76

## 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
- Parameterization is not easy

## Characteristics of Data, Clusters, and Clustering Algorithms

- A cluster analysis is affected by characteristics of
  - Data
  - Clusters
  - Clustering algorithms
- Looking at these characteristics gives us a number of dimensions that you can use to describe clustering algorithms and the results that they produce

## Characteristics of Data

---

- High dimensionality
- Size of data set
- Sparsity of attribute values
- Noise and Outliers
- Types of attributes and type of data sets
- Differences in attribute scales
- Properties of the data space
  - Can you define a meaningful centroid

## Characteristics of Clusters

---

- Data distribution
- Shape
- Differing sizes
- Differing densities
- Poor separation
- Relationship of clusters
- Types of clusters
  - Center-based, contiguity-based, density-based
- Subspace clusters

## Characteristics of Clustering Algorithms

- Order dependence
- Non-determinism
- Parameter selection
- Scalability
- Underlying model
- Optimization based approach

## Comparison of MIN and EM-Clustering

- We assume EM clustering using the Gaussian (normal) distribution.
- MIN is hierarchical, EM clustering is partitional.
- Both MIN and EM clustering are complete.
- MIN has a graph-based (contiguity-based) notion of a cluster, while EM clustering has a prototype (or model-based) notion of a cluster.
- MIN will not be able to distinguish poorly separated clusters, but EM can manage this in many situations.
- MIN can find clusters of different shapes and sizes; EM clustering prefers globular clusters and can have trouble with clusters of different sizes.
- MIN has trouble with clusters of different densities, while EM can often handle this.
- Neither MIN nor EM clustering finds subspace clusters.

## Comparison of MIN and EM-Clustering

- MIN can handle outliers, but noise can join clusters; EM clustering can tolerate noise, but can be strongly affected by outliers.
- EM can only be applied to data for which a centroid is meaningful; MIN only requires a meaningful definition of proximity.
- EM will have trouble as dimensionality increases and the number of its parameters (the number of entries in the covariance matrix) increases as the square of the number of dimensions; MIN can work well with a suitable definition of proximity.
- EM is designed for Euclidean data, although versions of EM clustering have been developed for other types of data. MIN is shielded from the data type by the fact that it uses a similarity matrix.
- MIN makes no distribution assumptions; the version of EM we are considering assumes Gaussian distributions.

## Comparison of MIN and EM-Clustering

- EM has an  $O(n)$  time complexity; MIN is  $O(n^2 \log(n))$ .
- Because of random initialization, the clusters found by EM can vary from one run to another; MIN produces the same clusters unless there are ties in the similarity matrix.
- Neither MIN nor EM automatically determine the number of clusters.
- MIN does not have any user-specified parameters; EM has the number of clusters and possibly the weights of the clusters.
- EM clustering can be viewed as an optimization problem; MIN uses a graph model of the data.
- Neither EM or MIN are order dependent.

## Comparison of DBSCAN and K-means

- Both are partitional.
- K-means is complete; DBSCAN is not.
- K-means has a prototype-based notion of a cluster; DB uses a density-based notion.
- K-means can find clusters that are not well-separated. DBSCAN will merge clusters that touch.
- DBSCAN handles clusters of different shapes and sizes; K-means prefers globular clusters.

## Comparison of DBSCAN and K-means

- DBSCAN can handle noise and outliers; K-means performs poorly in the presence of outliers
- K-means can only be applied to data for which a centroid is meaningful; DBSCAN requires a meaningful definition of density
- DBSCAN works poorly on high-dimensional data; K-means works well for some types of high-dimensional data
- Both techniques were designed for Euclidean data, but extended to other types of data
- DBSCAN makes no distribution assumptions; K-means is really assuming spherical Gaussian distributions

## Comparison of DBSCAN and K-means

---

- K-means has an  $O(n)$  time complexity; DBSCAN is  $O(n^2)$
- Because of random initialization, the clusters found by K-means can vary from one run to another; DBSCAN always produces the same clusters
- DBSCAN automatically determines the number of clusters; K-means does not
- K-means has only one parameter, DBSCAN has two.
- K-means clustering can be viewed as an optimization problem and as a special case of EM clustering; DBSCAN is not based on a formal model.

# Anomaly Detection

## Lecture Notes for Chapter 9

Introduction to Data Mining, 2<sup>nd</sup> Edition  
by  
Tan, Steinbach, Karpatne, Kumar

2/14/18

Introduction to Data Mining, 2nd Edition

1

## Anomaly/Outlier Detection

- What are anomalies/outliers?
  - The set of data points that are considerably different than the remainder of the data
- Natural implication is that anomalies are relatively rare
  - One in a thousand occurs often if you have lots of data
  - Context is important, e.g., freezing temps in July
- Can be important or a nuisance
  - 10 foot tall 2 year old
  - Unusually high blood pressure

2/14/18

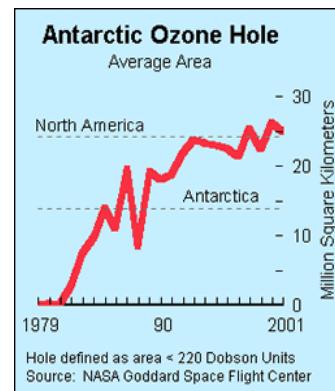
Introduction to Data Mining, 2nd Edition

2

# Importance of Anomaly Detection

## Ozone Depletion History

- In 1985 three researchers (Farman, Gardiner 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>

2/14/18

Introduction to Data Mining, 2nd Edition

3

# Causes of Anomalies

- Data from different classes
  - Measuring the weights of oranges, but a few grapefruit are mixed in
- Natural variation
  - Unusually tall people
- Data errors
  - 200 pound 2 year old

2/14/18

Introduction to Data Mining, 2nd Edition

4

## Distinction Between Noise and Anomalies

---

- Noise is erroneous, perhaps random, values or contaminating objects
  - Weight recorded incorrectly
  - Grapefruit mixed in with the oranges
- Noise doesn't necessarily produce unusual values or objects
- Noise is not interesting
- Anomalies may be interesting if they are not a result of noise
- Noise and anomalies are related but distinct concepts

## General Issues: Number of Attributes

---

- Many anomalies are defined in terms of a single attribute
  - Height
  - Shape
  - Color
- Can be hard to find an anomaly using all attributes
  - Noisy or irrelevant attributes
  - Object is only anomalous with respect to some attributes
- However, an object may not be anomalous in any one attribute

## General Issues: Anomaly Scoring

- Many anomaly detection techniques provide only a binary categorization
  - An object is an anomaly or it isn't
  - This is especially true of classification-based approaches
- Other approaches assign a score to all points
  - This score measures the degree to which an object is an anomaly
  - This allows objects to be ranked
- In the end, you often need a binary decision
  - Should this credit card transaction be flagged?
  - Still useful to have a score
- How many anomalies are there?

2/14/18

Introduction to Data Mining, 2nd Edition

7

## Other Issues for Anomaly Detection

- Find all anomalies at once or one at a time
  - Swamping
  - Masking
- Evaluation
  - How do you measure performance?
  - Supervised vs. unsupervised situations
- Efficiency
- Context
  - Professional basketball team

2/14/18

Introduction to Data Mining, 2nd Edition

8

## Variants of Anomaly Detection Problems

---

- Given a data set D, find all data points  $x \in D$  with anomaly scores greater than some threshold t
- Given a data set D, find all data points  $x \in D$  having the top-n largest anomaly scores
- Given a data set D, containing mostly normal (but unlabeled) data points, and a test point  $x$ , compute the anomaly score of  $x$  with respect to D

## Model-Based Anomaly Detection

---

- Build a model for the data and see
  - Unsupervised
    - ◆ Anomalies are those points that don't fit well
    - ◆ Anomalies are those points that distort the model
    - ◆ Examples:
      - Statistical distribution
      - Clusters
      - Regression
      - Geometric
      - Graph
  - Supervised
    - ◆ Anomalies are regarded as a rare class
    - ◆ Need to have training data

## Additional Anomaly Detection Techniques

- Proximity-based
  - Anomalies are points far away from other points
  - Can detect this graphically in some cases
- Density-based
  - Low density points are outliers
- Pattern matching
  - Create profiles or templates of atypical but important events or objects
  - Algorithms to detect these patterns are usually simple and efficient

2/14/18

Introduction to Data Mining, 2nd Edition

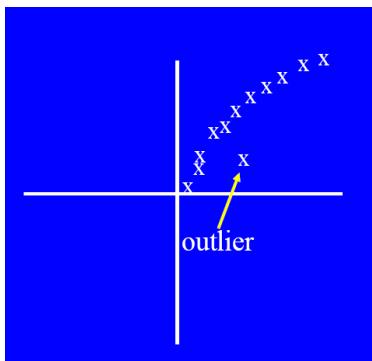
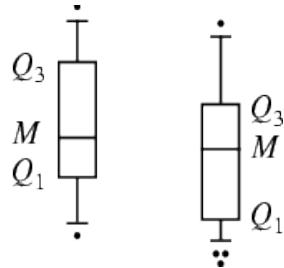
11

## Visual Approaches

- Boxplots or scatter plots

- Limitations

- Not automatic
  - Subjective



2/14/18

Introduction to Data Mining, 2nd Edition

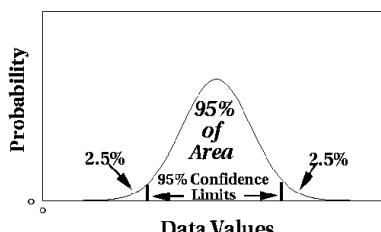
12

## Statistical Approaches

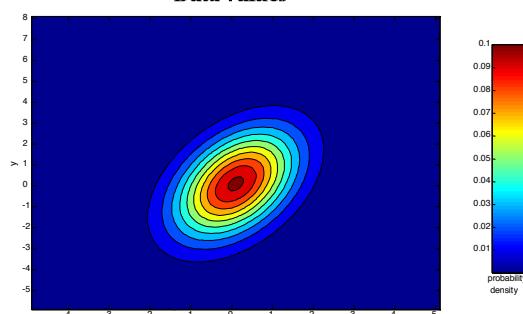
**Probabilistic definition of an outlier:** An outlier is an object that has a low probability with respect to a probability distribution model of the data.

- Usually assume a parametric model describing the distribution of the data (e.g., normal distribution)
- Apply a statistical test that depends on
  - Data distribution
  - Parameters of distribution (e.g., mean, variance)
  - Number of expected outliers (confidence limit)
- Issues
  - Identifying the distribution of a data set
    - ◆ Heavy tailed distribution
  - Number of attributes
  - Is the data a mixture of distributions?

## Normal Distributions



One-dimensional  
Gaussian



Two-dimensional  
Gaussian

## 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:  
$$G = \frac{\max|X - \bar{X}|}{S}$$
- Reject  $H_0$  if:

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{t_{(\alpha/2, N-2)}^2}{N-2 + t_{(\alpha/2, N-2)}^2}}$$

## 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  $t$ :

$$L_t(D) = \prod_{i=1}^N P_D(x_i) = \left( (1 - \lambda)^{|M_t|} \prod_{x_i \in M_t} P_{M_t}(x_i) \right) \left( \lambda^{|A_t|} \prod_{x_i \in A_t} P_{A_t}(x_i) \right)$$
$$LL_t(D) = |M_t| \log(1 - \lambda) + \sum_{x_i \in M_t} \log P_{M_t}(x_i) + |A_t| \log \lambda + \sum_{x_i \in A_t} \log P_{A_t}(x_i)$$

## Strengths/Weaknesses of Statistical Approaches

- Firm mathematical foundation
- Can be very efficient
- Good results if distribution is known
- In many cases, data distribution may not be known
- For high dimensional data, it may be difficult to estimate the true distribution
- Anomalies can distort the parameters of the distribution

## Distance-Based Approaches

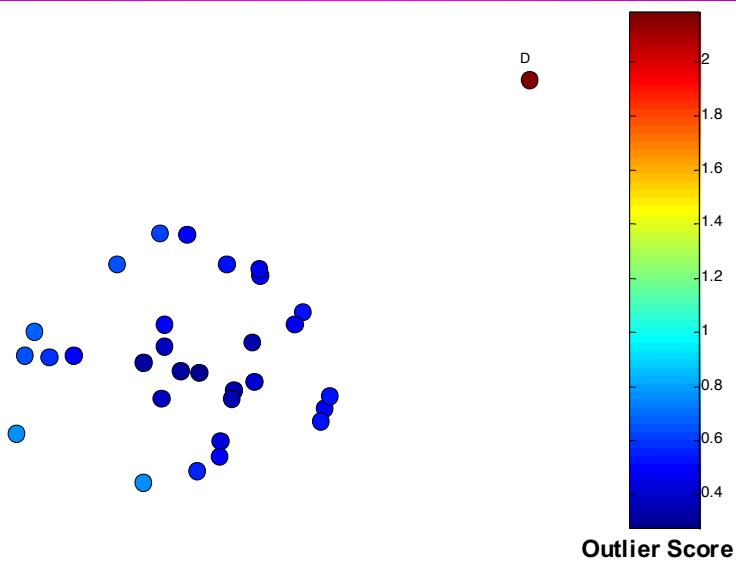
- Several different techniques
- An object is an outlier if a specified fraction of the objects is more than a specified distance away  
(Knorr, Ng 1998)
  - Some statistical definitions are special cases of this
- The outlier score of an object is the distance to its kth nearest neighbor

2/14/18

Introduction to Data Mining, 2nd Edition

19

## One Nearest Neighbor - One Outlier

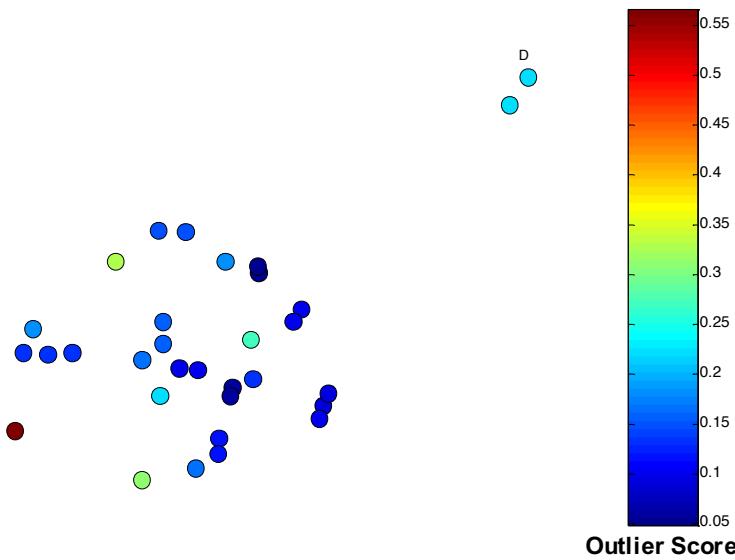


2/14/18

Introduction to Data Mining, 2nd Edition

20

## One Nearest Neighbor - Two Outliers

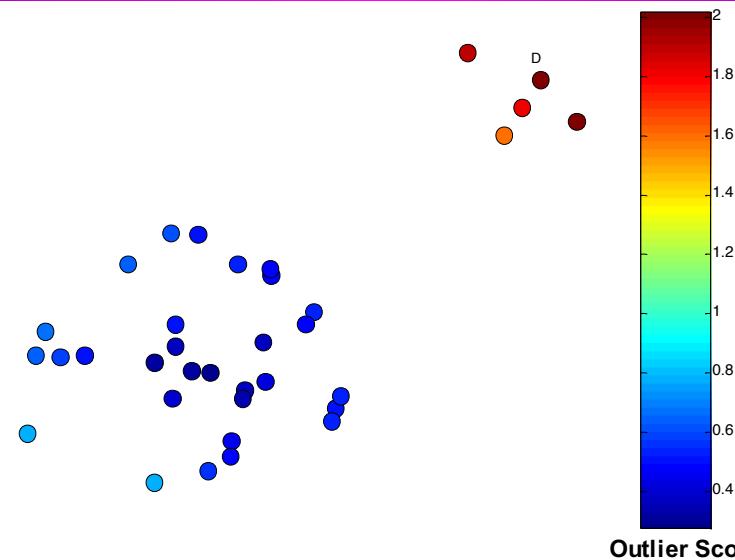


2/14/18

Introduction to Data Mining, 2nd Edition

21

## Five Nearest Neighbors - Small Cluster

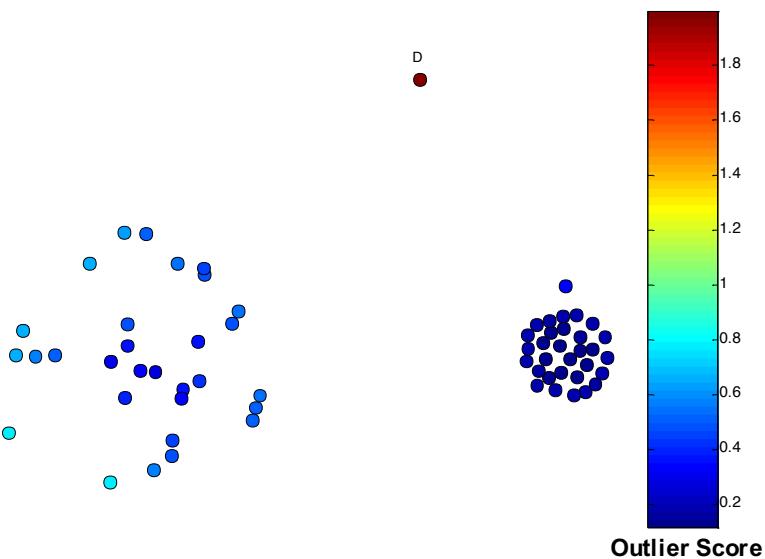


2/14/18

Introduction to Data Mining, 2nd Edition

22

## Five Nearest Neighbors - Differing Density



2/14/18

Introduction to Data Mining, 2nd Edition

23

## Strengths/Weaknesses of Distance-Based Approaches

- Simple
- Expensive –  $O(n^2)$
- Sensitive to parameters
- Sensitive to variations in density
- Distance becomes less meaningful in high-dimensional space

2/14/18

Introduction to Data Mining, 2nd Edition

24

## Density-Based Approaches

- **Density-based Outlier:** The outlier score of an object is the inverse of the density around the object.
  - Can be defined in terms of the  $k$  nearest neighbors
  - One definition: Inverse of distance to  $k$ th neighbor
  - Another definition: Inverse of the average distance to  $k$  neighbors
  - DBSCAN definition
- If there are regions of different density, this approach can have problems

2/14/18

Introduction to Data Mining, 2nd Edition

25

## Relative Density

- Consider the density of a point relative to that of its  $k$  nearest neighbors

$$\text{average relative density}(\mathbf{x}, k) = \frac{\text{density}(\mathbf{x}, k)}{\sum_{\mathbf{y} \in N(\mathbf{x}, k)} \text{density}(\mathbf{y}, k) / |N(\mathbf{x}, k)|}. \quad (10.7)$$

**Algorithm 10.2** Relative density outlier score algorithm.

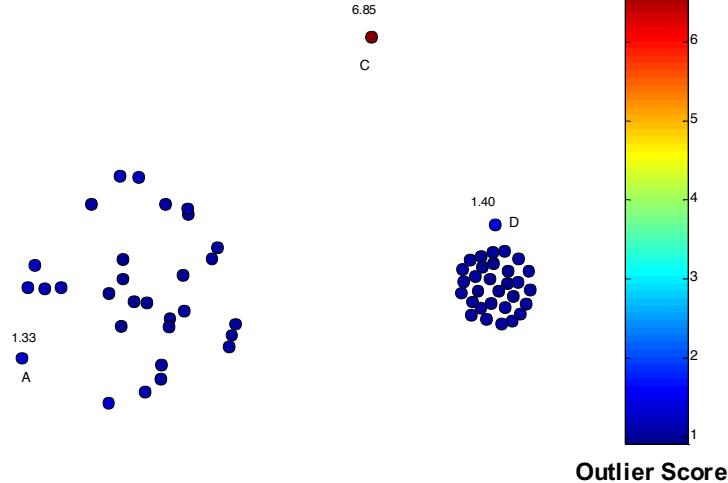
- 
- 1:  $\{k$  is the number of nearest neighbors $\}$
  - 2: **for** all objects  $\mathbf{x}$  **do**
  - 3:   Determine  $N(\mathbf{x}, k)$ , the  $k$ -nearest neighbors of  $\mathbf{x}$ .
  - 4:   Determine  $\text{density}(\mathbf{x}, k)$ , the density of  $\mathbf{x}$ , using its nearest neighbors, i.e., the objects in  $N(\mathbf{x}, k)$ .
  - 5: **end for**
  - 6: **for** all objects  $\mathbf{x}$  **do**
  - 7:   Set the  $\text{outlier score}(\mathbf{x}, k) = \text{average relative density}(\mathbf{x}, k)$  from Equation 10.7.
  - 8: **end for**
- 

2/14/18

Introduction to Data Mining, 2nd Edition

26

## Relative Density Outlier Scores



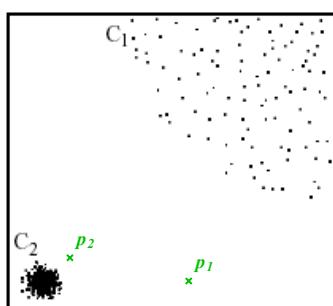
2/14/18

Introduction to Data Mining, 2nd Edition

27

## Density-based: LOF approach

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



In the NN approach,  $p_2$  is not considered as outlier, while LOF approach find both  $p_1$  and  $p_2$  as outliers

2/14/18

Introduction to Data Mining, 2nd Edition

28

## Strengths/Weaknesses of Density-Based Approaches

- Simple
- Expensive –  $O(n^2)$
- Sensitive to parameters
- Density becomes less meaningful in high-dimensional space

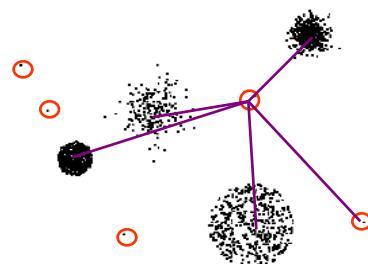
2/14/18

Introduction to Data Mining, 2nd Edition

29

## Clustering-Based Approaches

- **Clustering-based Outlier:** An object is a cluster-based outlier if it does not strongly belong to any cluster
  - For prototype-based clusters, an object is an outlier if it is not close enough to a cluster center
  - For density-based clusters, an object is an outlier if its density is too low
  - For graph-based clusters, an object is an outlier if it is not well connected
- Other issues include the impact of outliers on the clusters and the number of clusters

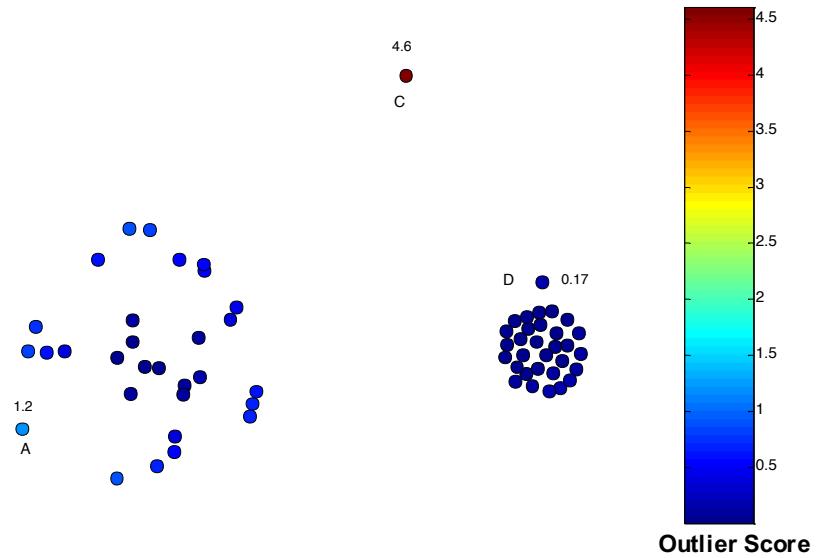


2/14/18

Introduction to Data Mining, 2nd Edition

30

## Distance of Points from Closest Centroids

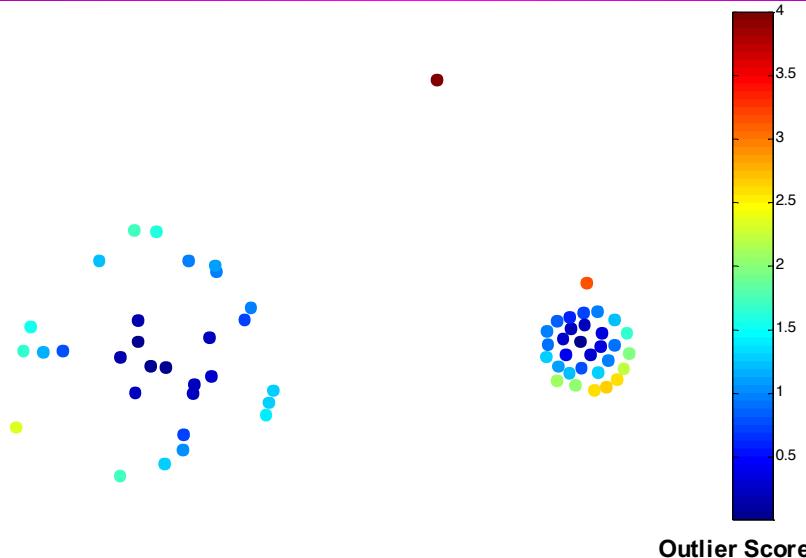


2/14/18

Introduction to Data Mining, 2nd Edition

31

## Relative Distance of Points from Closest Centroid



2/14/18

Introduction to Data Mining, 2nd Edition

32

## **Strengths/Weaknesses of Distance-Based Approaches**

---

- Simple
- Many clustering techniques can be used
- Can be difficult to decide on a clustering technique
- Can be difficult to decide on number of clusters
- Outliers can distort the clusters