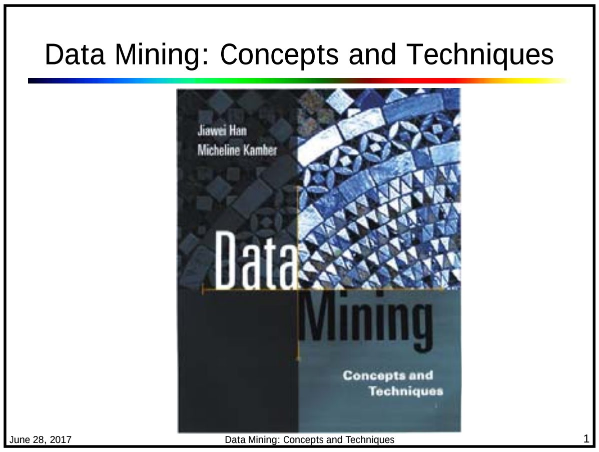
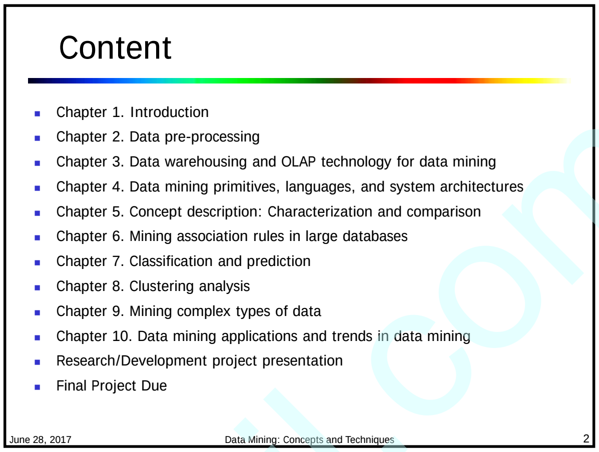
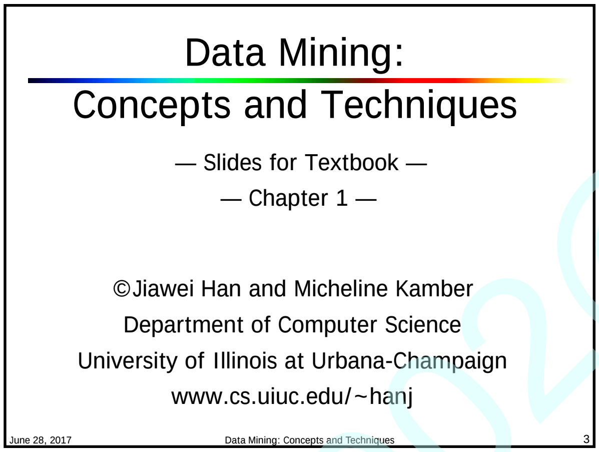


Data Mining: Concepts and Techniques



Content

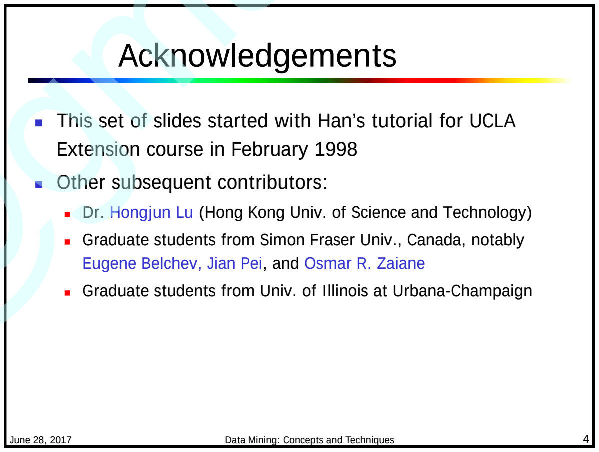
- Chapter 1. Introduction
- Chapter 2. Data pre-processing
- Chapter 3. Data warehousing and OLAP technology for data mining
- Chapter 4. Data mining primitives, languages, and system architectures
- Chapter 5. Concept description: Characterization and comparison
- Chapter 6. Mining association rules in large databases
- Chapter 7. Classification and prediction
- Chapter 8. Clustering analysis
- Chapter 9. Mining complex types of data
- Chapter 10. Data mining applications and trends in data mining
- Research/Development project presentation
- Final Project Due



Data Mining:
Concepts and Techniques

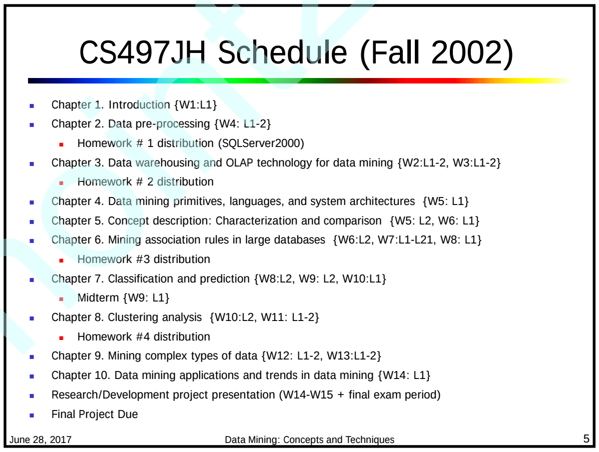
— Slides for Textbook —
— Chapter 1 —

©Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj



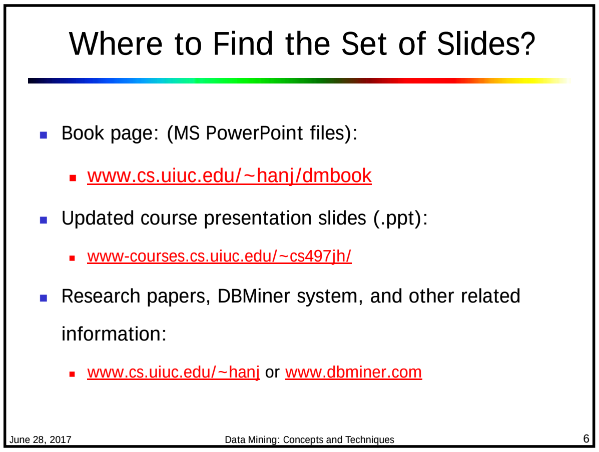
Acknowledgements

- This set of slides started with Han's tutorial for UCLA Extension course in February 1998
- Other subsequent contributors:
  - Dr. Hongjun Lu (Hong Kong Univ. of Science and Technology)
  - Graduate students from Simon Fraser Univ., Canada, notably Eugene Belchev, Jian Pei, and Osmar R. Zaiane
  - Graduate students from Univ. of Illinois at Urbana-Champaign



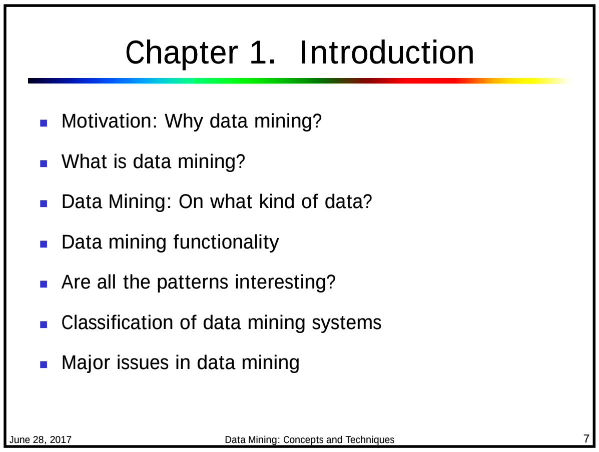
CS497JH Schedule (Fall 2002)

- Chapter 1. Introduction {W1:L1}
- Chapter 2. Data pre-processing {W4: L1-2}
  - Homework # 1 distribution (SQLServer2000)
- Chapter 3. Data warehousing and OLAP technology for data mining {W2:L1-2, W3:L1-2}
  - Homework # 2 distribution
- Chapter 4. Data mining primitives, languages, and system architectures {W5: L1}
- Chapter 5. Concept description: Characterization and comparison {W5: L2, W6: L1}
- Chapter 6. Mining association rules in large databases {W6:L2, W7:L1-L21, W8: L1}
  - Homework #3 distribution
- Chapter 7. Classification and prediction {W8:L2, W9: L2, W10:L1}
  - Midterm {W9: L1}
- Chapter 8. Clustering analysis {W10:L2, W11: L1-2}
  - Homework #4 distribution
- Chapter 9. Mining complex types of data {W12: L1-2, W13:L1-2}
- Chapter 10. Data mining applications and trends in data mining {W14: L1}
- Research/Development project presentation (W14-W15 + final exam period)
- Final Project Due



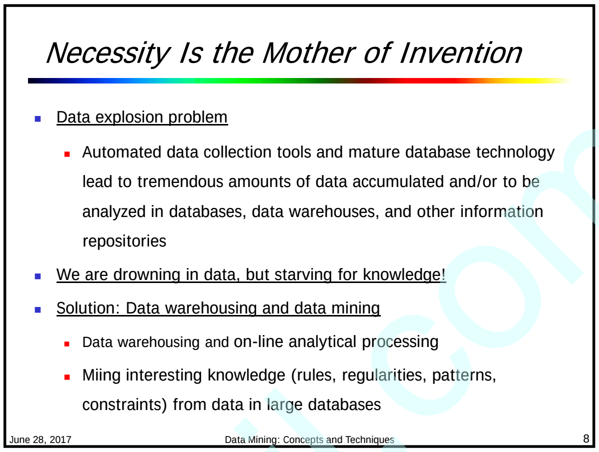
Where to Find the Set of Slides?

- Book page: (MS PowerPoint files):
  - www.cs.uiuc.edu/~hanj/dmbook
- Updated course presentation slides (.ppt):
  - www-courses.cs.uiuc.edu/~cs497jh/
- Research papers, DBMiner system, and other related information:
  - www.cs.uiuc.edu/~hanj or www.dbminer.com



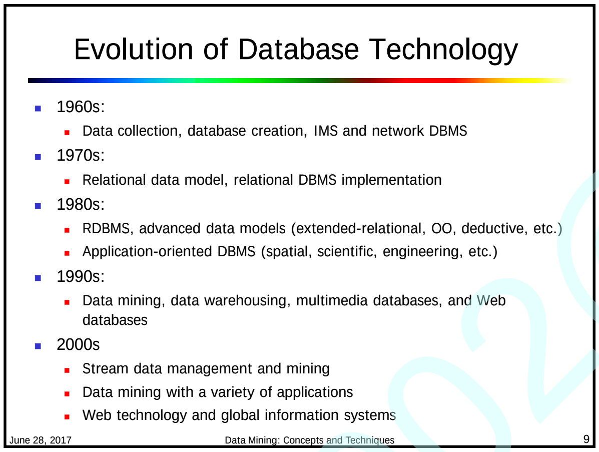
Chapter 1. Introduction

- Motivation: Why data mining?
- What is data mining?
- Data Mining: On what kind of data?
- Data mining functionality
- Are all the patterns interesting?
- Classification of data mining systems
- Major issues in data mining



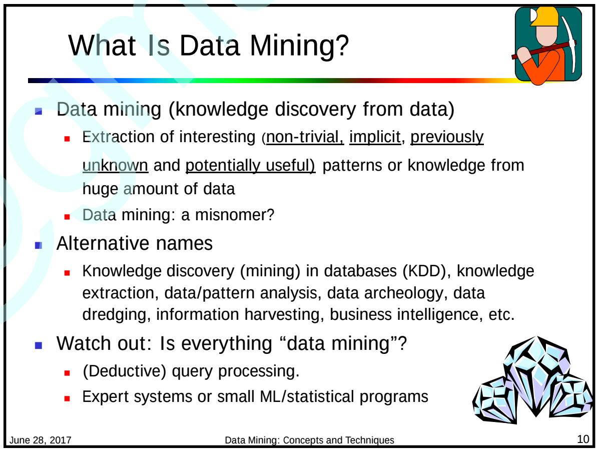
Necessity Is the Mother of Invention

- Data explosion problem
  - Automated data collection tools and mature database technology lead to tremendous amounts of data accumulated and/or to be analyzed in databases, data warehouses, and other information repositories
- We are drowning in data, but starving for knowledge!
- Solution: Data warehousing and data mining
  - Data warehousing and on-line analytical processing
  - Miing interesting knowledge (rules, regularities, patterns, constraints) from data in large databases



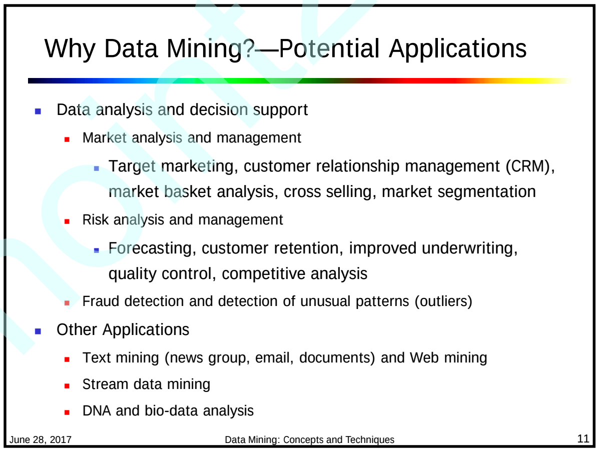
Evolution of Database Technology

- 1960s:
  - Data collection, database creation, IMS and network DBMS
- 1970s:
  - Relational data model, relational DBMS implementation
- 1980s:
  - RDBMS, advanced data models (extended-relational, OO, deductive, etc.)
  - Application-oriented DBMS (spatial, scientific, engineering, etc.)
- 1990s:
  - Data mining, data warehousing, multimedia databases, and Web databases
- 2000s
  - Stream data management and mining
  - Data mining with a variety of applications
  - Web technology and global information systems



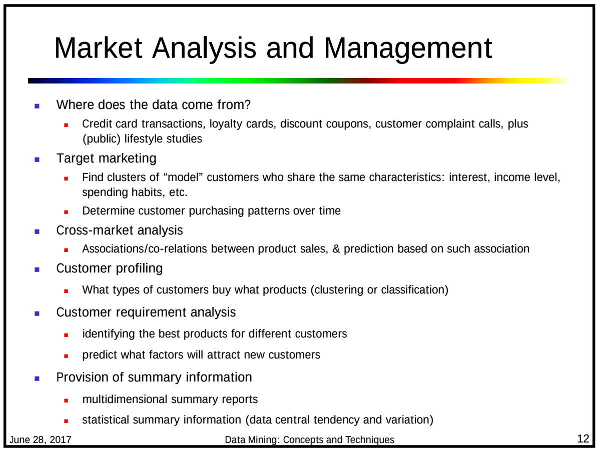
What Is Data Mining?

- Data mining (knowledge discovery from data)
  - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data
  - Data mining: a misnomer?
- Alternative names
  - Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.
- Watch out: Is everything "data mining"?
  - (Deductive) query processing.
  - Expert systems or small ML/statistical programs



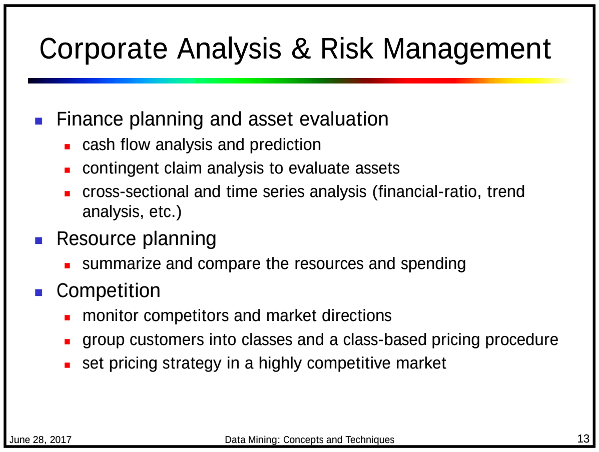
Why Data Mining?—Potential Applications

- Data analysis and decision support
  - Market analysis and management
    - Target marketing, customer relationship management (CRM), market basket analysis, cross selling, market segmentation
  - Risk analysis and management
    - Forecasting, customer retention, improved underwriting, quality control, competitive analysis
  - Fraud detection and detection of unusual patterns (outliers)
- Other Applications
  - Text mining (news group, email, documents) and Web mining
  - Stream data mining
  - DNA and bio-data analysis



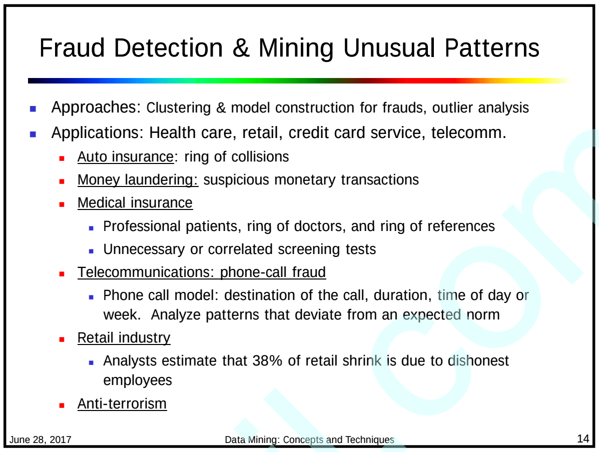
Market Analysis and Management

- Where does the data come from?
  - Credit card transactions, loyalty cards, discount coupons, customer complaint calls, plus (public) lifestyle studies
- Target marketing
  - Find clusters of "model" customers who share the same characteristics: interest, income level, spending habits, etc.
  - Determine customer purchasing patterns over time
- Cross-market analysis
  - Associations/co-relations between product sales, & prediction based on such association
- Customer profiling
  - What types of customers buy what products (clustering or classification)
- Customer requirement analysis
  - identifying the best products for different customers
  - predict what factors will attract new customers
- Provision of summary information
  - multidimensional summary reports
  - statistical summary information (data central tendency and variation)



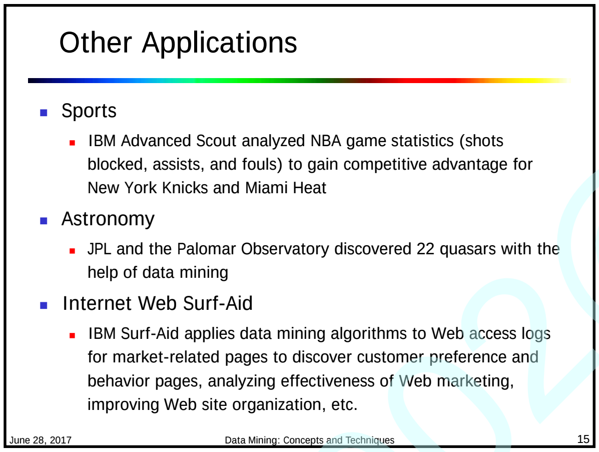
Corporate Analysis & Risk Management

- Finance planning and asset evaluation
  - cash flow analysis and prediction
  - contingent claim analysis to evaluate assets
  - cross-sectional and time series analysis (financial-ratio, trend analysis, etc.)
- Resource planning
  - summarize and compare the resources and spending
- Competition
  - monitor competitors and market directions
  - group customers into classes and a class-based pricing procedure
  - set pricing strategy in a highly competitive market



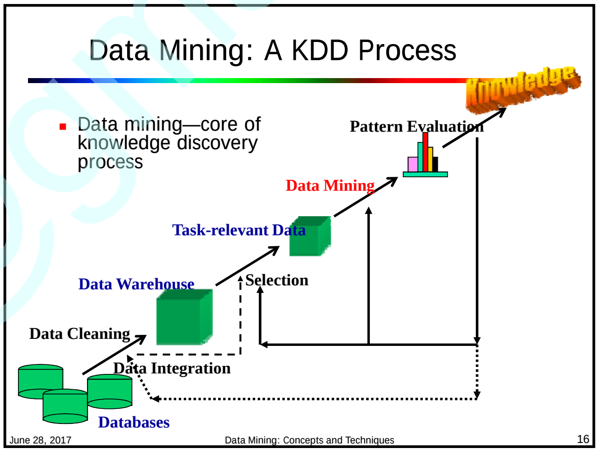
Fraud Detection & Mining Unusual Patterns

- Approaches: Clustering & model construction for frauds, outlier analysis
- Applications: Health care, retail, credit card service, telecomm.
  - Auto insurance: ring of collisions
  - Money laundering: suspicious monetary transactions
  - Medical insurance
    - Professional patients, ring of doctors, and ring of references
    - Unnecessary or correlated screening tests
  - Telecommunications: phone-call fraud
    - Phone call model: destination of the call, duration, time of day or week. Analyze patterns that deviate from an expected norm
  - Retail industry
    - Analysts estimate that 38% of retail shrink is due to dishonest employees
  - Anti-terrorism



Other Applications

- Sports
  - IBM Advanced Scout analyzed NBA game statistics (shots blocked, assists, and fouls) to gain competitive advantage for New York Knicks and Miami Heat
- Astronomy
  - JPL and the Palomar Observatory discovered 22 quasars with the help of data mining
- Internet Web Surf-Aid
  - IBM Surf-Aid applies data mining algorithms to Web access logs for market-related pages to discover customer preference and behavior pages, analyzing effectiveness of Web marketing, improving Web site organization, etc.



Data Mining: A KDD Process

- Data mining—core of knowledge discovery process

Knowledge

Pattern Evaluation

Data Mining

Task-relevant Data

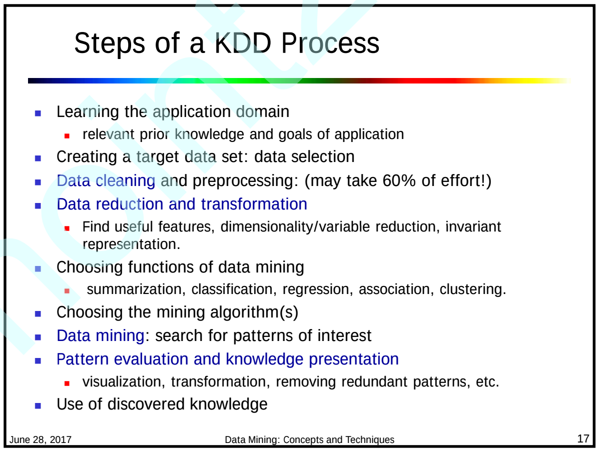
Data Warehouse

Selection

Data Cleaning

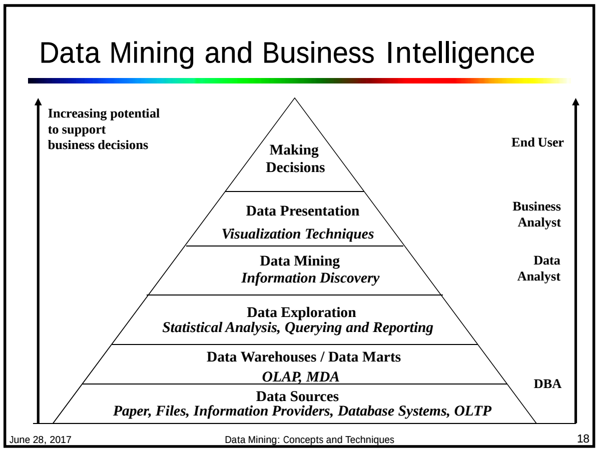
Data Integration

Databases



Steps of a KDD Process

- Learning the application domain
  - relevant prior knowledge and goals of application
- Creating a target data set: data selection
- Data cleaning and preprocessing: (may take 60% of effort!)
- Data reduction and transformation
  - Find useful features, dimensionality/variable reduction, invariant representation.
- Choosing functions of data mining
  - summarization, classification, regression, association, clustering.
- Choosing the mining algorithm(s)
- Data mining: search for patterns of interest
- Pattern evaluation and knowledge presentation
  - visualization, transformation, removing redundant patterns, etc.
- Use of discovered knowledge



Data Mining and Business Intelligence

Increasing potential to support business decisions

Making Decisions — End User

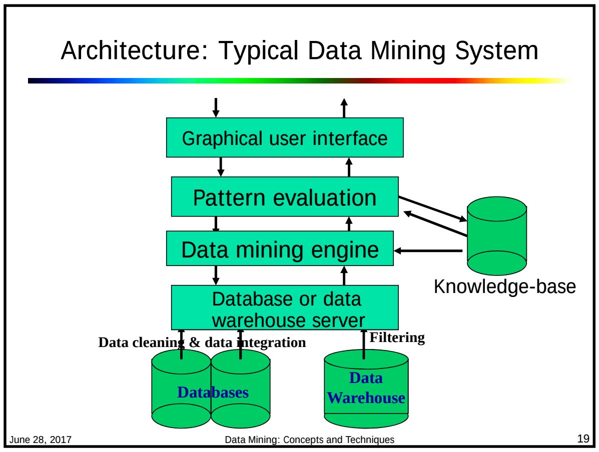
Data Presentation *Visualization Techniques* — Business Analyst

Data Mining *Information Discovery* — Data Analyst

Data Exploration *Statistical Analysis, Querying and Reporting*

Data Warehouses / Data Marts *OLAP, MDA* — DBA

Data Sources *Paper, Files, Information Providers, Database Systems, OLTP*



Architecture: Typical Data Mining System

Graphical user interface

Pattern evaluation

Data mining engine

Knowledge-base

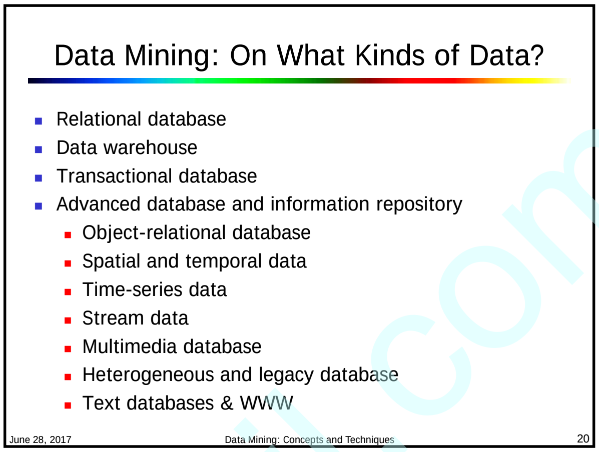
Database or data warehouse server

Data cleaning & data integration

Filtering

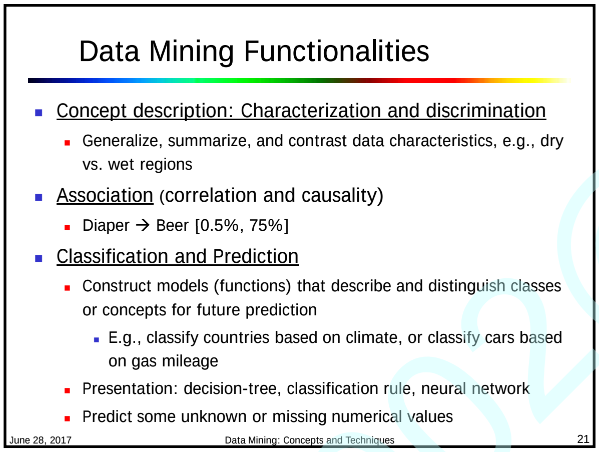
Databases

Data Warehouse



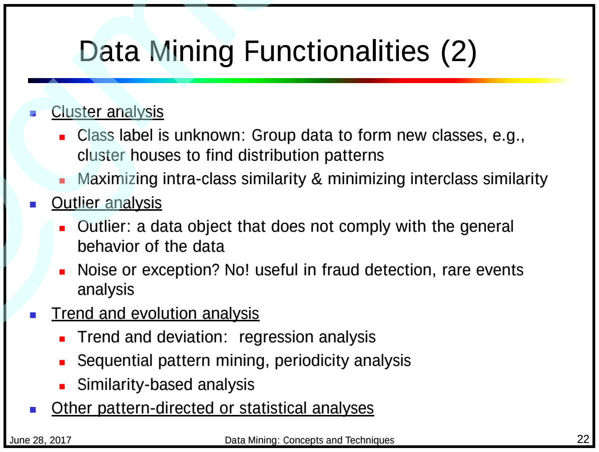
Data Mining: On What Kinds of Data?

- Relational database
- Data warehouse
- Transactional database
- Advanced database and information repository
  - Object-relational database
  - Spatial and temporal data
  - Time-series data
  - Stream data
  - Multimedia database
  - Heterogeneous and legacy database
  - Text databases & WWW



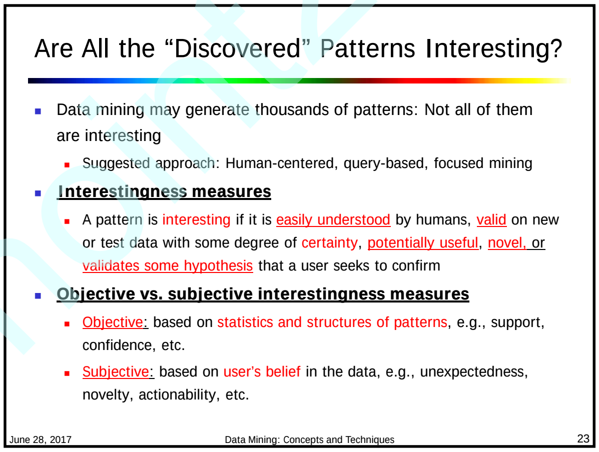
Data Mining Functionalities

- Concept description: Characterization and discrimination
  - Generalize, summarize, and contrast data characteristics, e.g., dry vs. wet regions
- Association (correlation and causality)
  - Diaper → Beer [0.5%, 75%]
- Classification and Prediction
  - Construct models (functions) that describe and distinguish classes or concepts for future prediction
    - E.g., classify countries based on climate, or classify cars based on gas mileage
  - Presentation: decision-tree, classification rule, neural network
  - Predict some unknown or missing numerical values



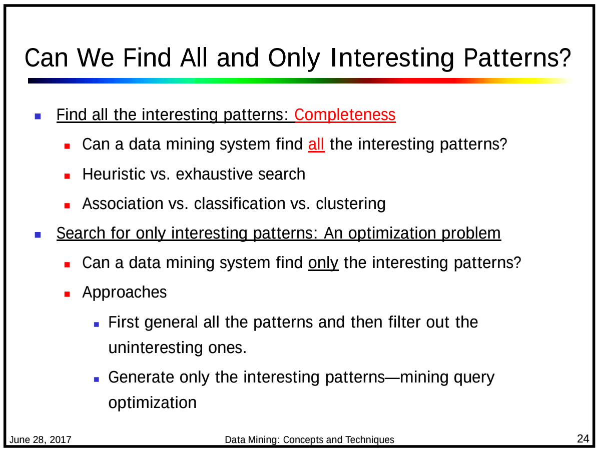
Data Mining Functionalities (2)

- Cluster analysis
  - Class label is unknown: Group data to form new classes, e.g., cluster houses to find distribution patterns
  - Maximizing intra-class similarity & minimizing interclass similarity
- Outlier analysis
  - Outlier: a data object that does not comply with the general behavior of the data
  - Noise or exception? No! useful in fraud detection, rare events analysis
- Trend and evolution analysis
  - Trend and deviation: regression analysis
  - Sequential pattern mining, periodicity analysis
  - Similarity-based analysis
- Other pattern-directed or statistical analyses



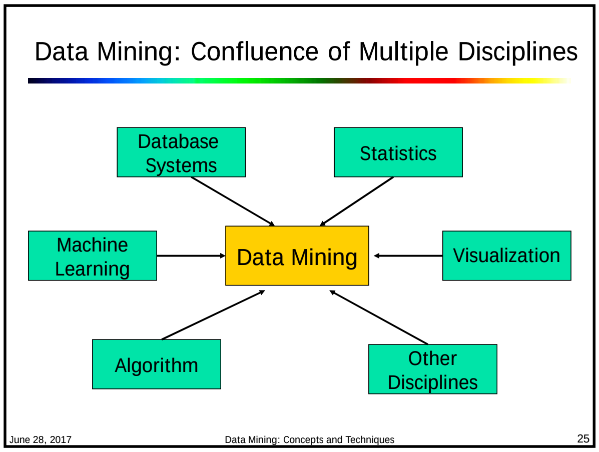
Are All the "Discovered" Patterns Interesting?

- Data mining may generate thousands of patterns: Not all of them are interesting
  - Suggested approach: Human-centered, query-based, focused mining
- **Interestingness measures**
  - A pattern is interesting if it is easily understood by humans, valid on new or test data with some degree of certainty, potentially useful, novel, or validates some hypothesis that a user seeks to confirm
- **Objective vs. subjective interestingness measures**
  - Objective: based on statistics and structures of patterns, e.g., support, confidence, etc.
  - Subjective: based on user's belief in the data, e.g., unexpectedness, novelty, actionability, etc.

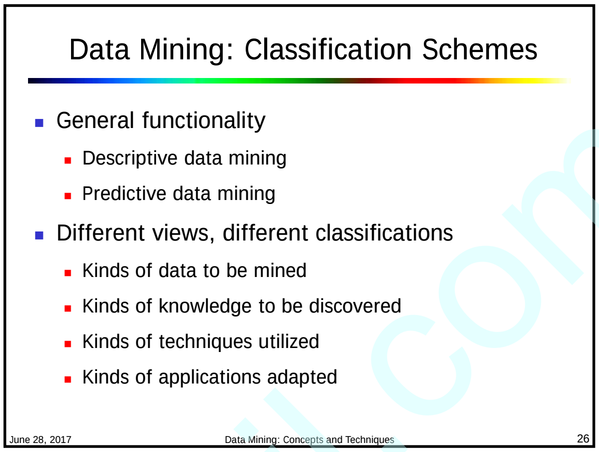


Can We Find All and Only Interesting Patterns?

- Find all the interesting patterns: Completeness
  - Can a data mining system find all the interesting patterns?
  - Heuristic vs. exhaustive search
  - Association vs. classification vs. clustering
- Search for only interesting patterns: An optimization problem
  - Can a data mining system find only the interesting patterns?
  - Approaches
    - First general all the patterns and then filter out the uninteresting ones.
    - Generate only the interesting patterns—mining query optimization

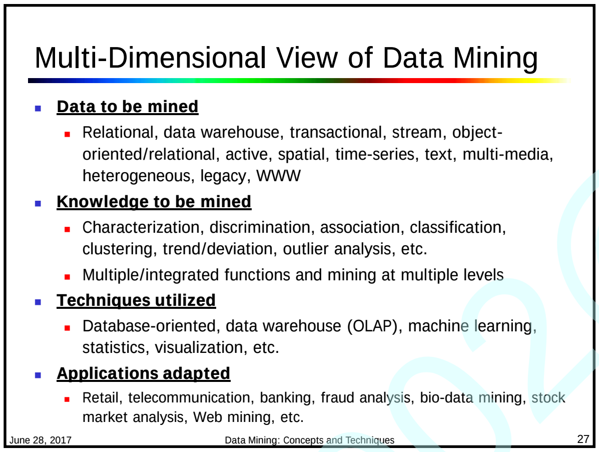


Data Mining: Confluence of Multiple Disciplines



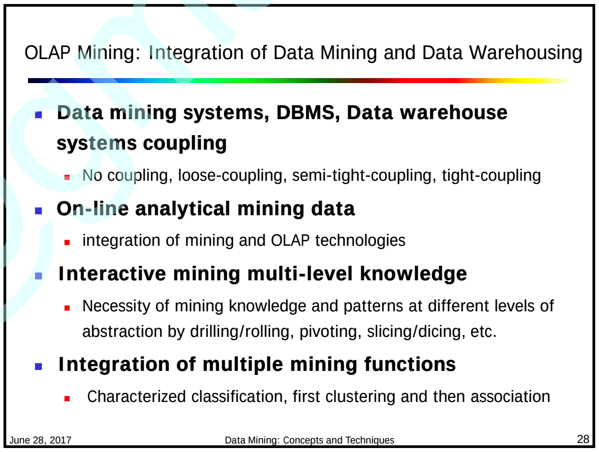
Data Mining: Classification Schemes

- General functionality
  - Descriptive data mining
  - Predictive data mining
- Different views, different classifications
  - Kinds of data to be mined
  - Kinds of knowledge to be discovered
  - Kinds of techniques utilized
  - Kinds of applications adapted



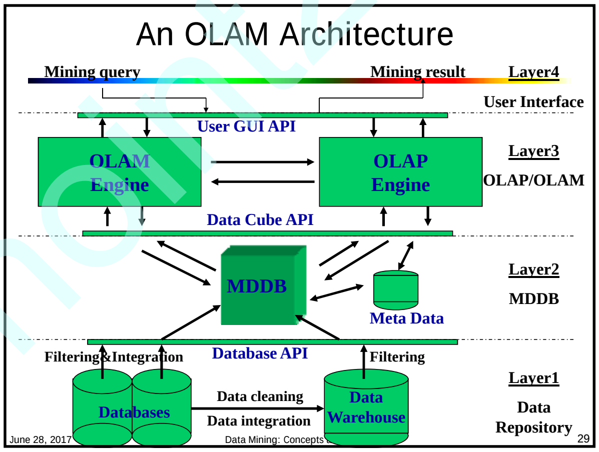
Multi-Dimensional View of Data Mining

- **Data to be mined**
  - Relational, data warehouse, transactional, stream, object-oriented/relational, active, spatial, time-series, text, multi-media, heterogeneous, legacy, WWW
- **Knowledge to be mined**
  - Characterization, discrimination, association, classification, clustering, trend/deviation, outlier analysis, etc.
  - Multiple/integrated functions and mining at multiple levels
- **Techniques utilized**
  - Database-oriented, data warehouse (OLAP), machine learning, statistics, visualization, etc.
- **Applications adapted**
  - Retail, telecommunication, banking, fraud analysis, bio-data mining, stock market analysis, Web mining, etc.

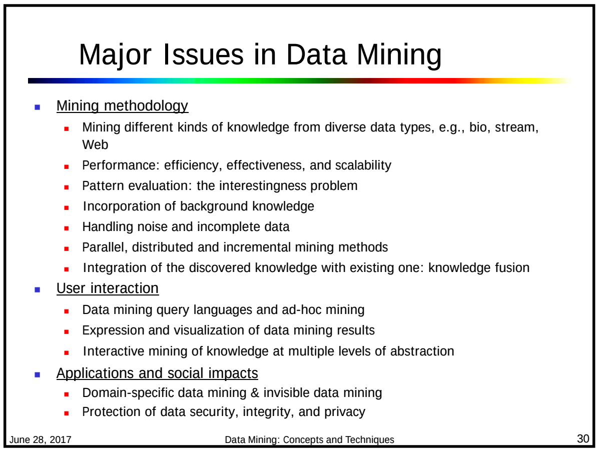


OLAP Mining: Integration of Data Mining and Data Warehousing

- **Data mining systems, DBMS, Data warehouse systems coupling**
  - No coupling, loose-coupling, semi-tight-coupling, tight-coupling
- **On-line analytical mining data**
  - integration of mining and OLAP technologies
- **Interactive mining multi-level knowledge**
  - Necessity of mining knowledge and patterns at different levels of abstraction by drilling/rolling, pivoting, slicing/dicing, etc.
- **Integration of multiple mining functions**
  - Characterized classification, first clustering and then association

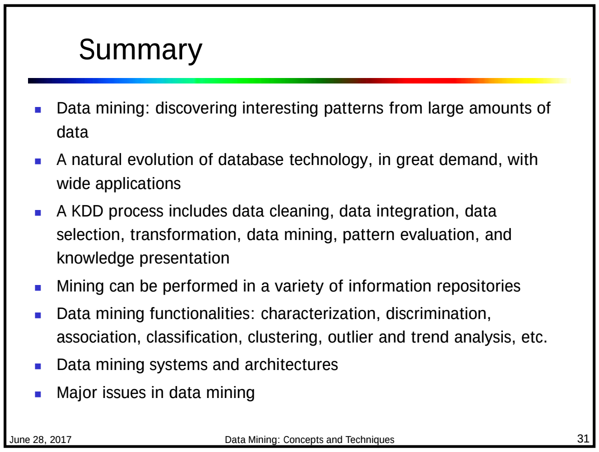


An OLAM Architecture



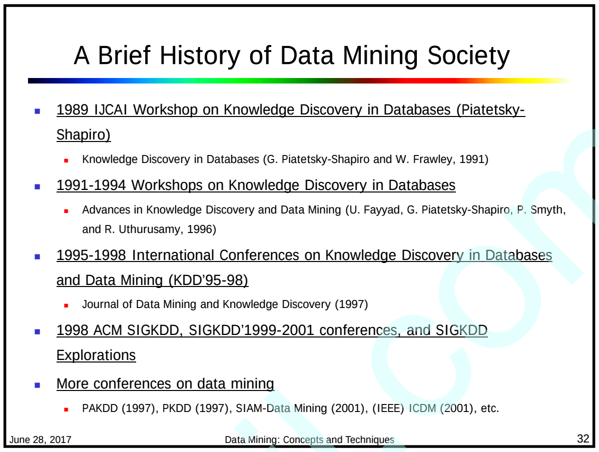
Major Issues in Data Mining

- Mining methodology
  - Mining different kinds of knowledge from diverse data types, e.g., bio, stream, Web
  - Performance: efficiency, effectiveness, and scalability
  - Pattern evaluation: the interestingness problem
  - Incorporation of background knowledge
  - Handling noise and incomplete data
  - Parallel, distributed and incremental mining methods
  - Integration of the discovered knowledge with existing one: knowledge fusion
- User interaction
  - Data mining query languages and ad-hoc mining
  - Expression and visualization of data mining results
  - Interactive mining of knowledge at multiple levels of abstraction
- Applications and social impacts
  - Domain-specific data mining & invisible data mining
  - Protection of data security, integrity, and privacy



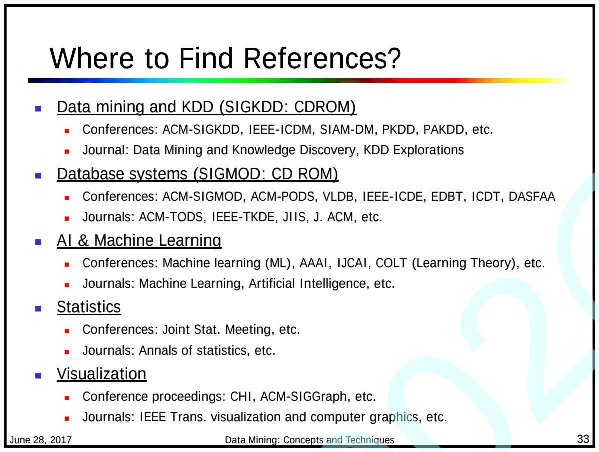
Summary

- Data mining: discovering interesting patterns from large amounts of data
- A natural evolution of database technology, in great demand, with wide applications
- A KDD process includes data cleaning, data integration, data selection, transformation, data mining, pattern evaluation, and knowledge presentation
- Mining can be performed in a variety of information repositories
- Data mining functionalities: characterization, discrimination, association, classification, clustering, outlier and trend analysis, etc.
- Data mining systems and architectures
- Major issues in data mining



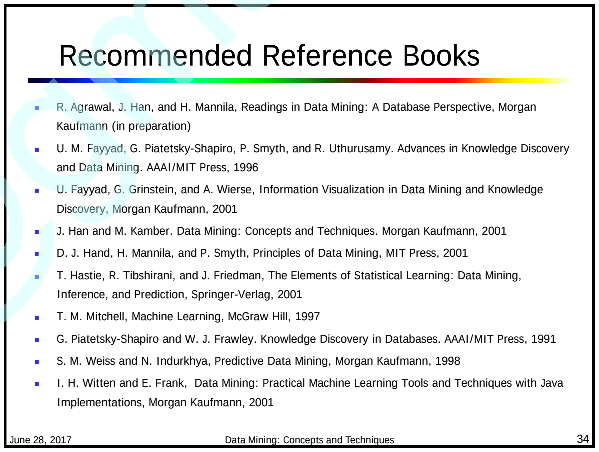
A Brief History of Data Mining Society

- 1989 IJCAI Workshop on Knowledge Discovery in Databases (Piatetsky-Shapiro)
  - Knowledge Discovery in Databases (G. Piatetsky-Shapiro and W. Frawley, 1991)
- 1991-1994 Workshops on Knowledge Discovery in Databases
  - Advances in Knowledge Discovery and Data Mining (U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 1996)
- 1995-1998 International Conferences on Knowledge Discovery in Databases and Data Mining (KDD'95-98)
  - Journal of Data Mining and Knowledge Discovery (1997)
- 1998 ACM SIGKDD, SIGKDD'1999-2001 conferences, and SIGKDD Explorations
- More conferences on data mining
  - PAKDD (1997), PKDD (1997), SIAM-Data Mining (2001), (IEEE) ICDM (2001), etc.



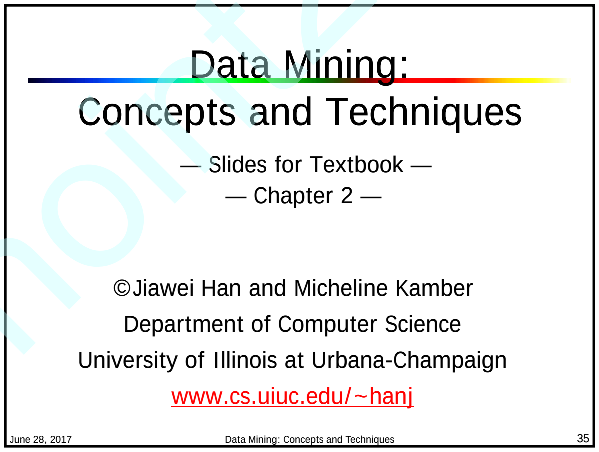
Where to Find References?

- Data mining and KDD (SIGKDD: CDROM)
  - Conferences: ACM-SIGKDD, IEEE-ICDM, SIAM-DM, PKDD, PAKDD, etc.
  - Journal: Data Mining and Knowledge Discovery, KDD Explorations
- Database systems (SIGMOD: CD ROM)
  - Conferences: ACM-SIGMOD, ACM-PODS, VLDB, IEEE-ICDE, EDBT, ICDT, DASFAA
  - Journals: ACM-TODS, IEEE-TKDE, JIIS, J. ACM, etc.
- AI & Machine Learning
  - Conferences: Machine learning (ML), AAAI, IJCAI, COLT (Learning Theory), etc.
  - Journals: Machine Learning, Artificial Intelligence, etc.
- Statistics
  - Conferences: Joint Stat. Meeting, etc.
  - Journals: Annals of statistics, etc.
- Visualization
  - Conference proceedings: CHI, ACM-SIGGraph, etc.
  - Journals: IEEE Trans. visualization and computer graphics, etc.



Recommended Reference Books

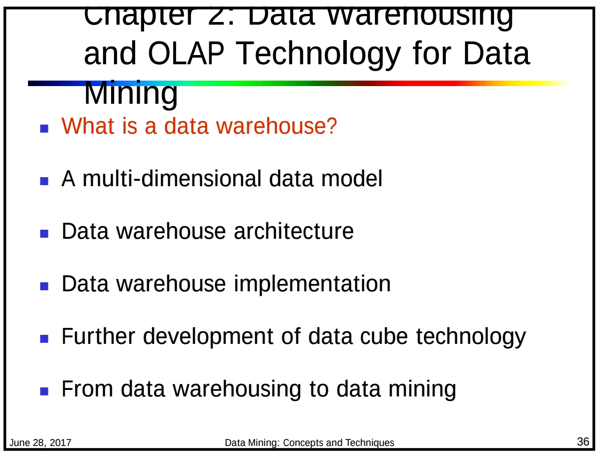
- R. Agrawal, J. Han, and H. Mannila, Readings in Data Mining: A Database Perspective, Morgan Kaufmann (in preparation)
- U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996
- U. Fayyad, G. Grinstein, and A. Wierse, Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann, 2001
- J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2001
- D. J. Hand, H. Mannila, and P. Smyth, Principles of Data Mining, MIT Press, 2001
- T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer-Verlag, 2001
- T. M. Mitchell, Machine Learning, McGraw Hill, 1997
- G. Piatetsky-Shapiro and W. J. Frawley. Knowledge Discovery in Databases. AAAI/MIT Press, 1991
- S. M. Weiss and N. Indurkhya, Predictive Data Mining, Morgan Kaufmann, 1998
- I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 2001



Data Mining: Concepts and Techniques

— Slides for Textbook —
— Chapter 2 —

©Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj



Chapter 2: Data Warehousing and OLAP Technology for Data Mining

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

What is Data Warehouse?

- Defined in many different ways, but not rigorously.
  - A decision support database that is maintained separately from the organization's operational database
  - Support information processing by providing a solid platform of consolidated, historical data for analysis.
- "A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."—W. H. Inmon
- Data warehousing:
  - The process of constructing and using data warehouses

Data Warehouse—Subject-Oriented

- Organized around major subjects, such as customer, product, sales.
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.
- Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

Data Warehouse—Integrated

- Constructed by integrating multiple, heterogeneous data sources
  - relational databases, flat files, on-line transaction records
- Data cleaning and data integration techniques are applied.
  - Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
    - E.g., Hotel price: currency, tax, breakfast covered, etc.
  - When data is moved to the warehouse, it is converted.

Data Warehouse—Time Variant

- The time horizon for the data warehouse is significantly longer than that of operational systems.
  - Operational database: current value data.
  - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)
- Every key structure in the data warehouse
  - Contains an element of time, explicitly or implicitly
  - But the key of operational data may or may not contain "time element".

Data Warehouse—Non-Volatile

- A physically separate store of data transformed from the operational environment.
- Operational update of data does not occur in the data warehouse environment.
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing:
    - *initial loading of data* and *access of data*.

Data Warehouse vs. Heterogeneous DBMS

- Traditional heterogeneous DB integration:
  - Build wrappers/mediators on top of heterogeneous databases
  - Query driven approach
    - When a query is posed to a client site, a meta-dictionary is used to translate the query into queries appropriate for individual heterogeneous sites involved, and the results are integrated into a global answer set
    - Complex information filtering, compete for resources
- Data warehouse: update-driven, high performance
  - Information from heterogeneous sources is integrated in advance and stored in warehouses for direct query and analysis

Data Warehouse vs. Operational DBMS

- OLTP (on-line transaction processing)
  - Major task of traditional relational DBMS
  - Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- OLAP (on-line analytical processing)
  - Major task of data warehouse system
  - Data analysis and decision making
- Distinct features (OLTP vs. OLAP):
  - User and system orientation: customer vs. market
  - Data contents: current, detailed vs. historical, consolidated
  - Database design: ER + application vs. star + subject
  - View: current, local vs. evolutionary, integrated
  - Access patterns: update vs. read-only but complex queries

OLTP vs. OLAP

	OLTP	**OLAP**
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

Why Separate Data Warehouse?

- High performance for both systems
  - DBMS— tuned for OLTP: access methods, indexing, concurrency control, recovery
  - Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation.
- Different functions and different data:
  - missing data: Decision support requires historical data which operational DBs do not typically maintain
  - data consolidation: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
  - data quality: different sources typically use inconsistent data representations, codes and formats which have to be reconciled

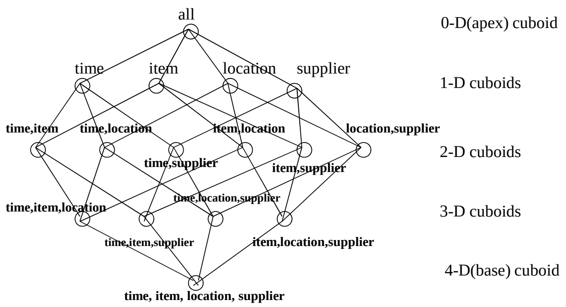
Chapter 2: Data Warehousing and OLAP Technology for Data Mining

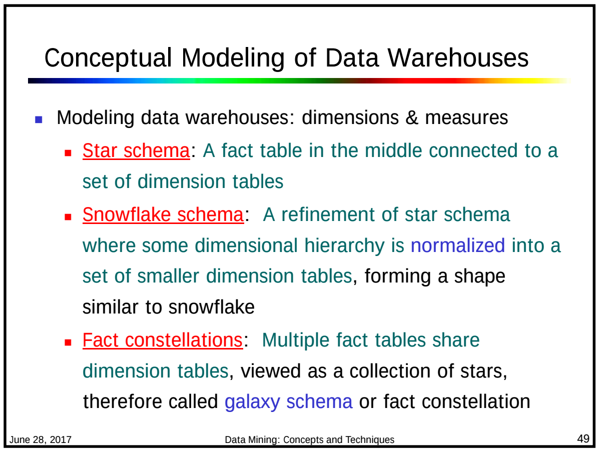
- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

From Tables and Spreadsheets to Data Cubes

- A data warehouse is based on a multidimensional data model which views data in the form of a data cube
- A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
  - Dimension tables, such as item (item_name, brand, type), or time(day, week, month, quarter, year)
  - Fact table contains measures (such as dollars_sold) and keys to each of the related dimension tables
- In data warehousing literature, an n-D base cube is called a base cuboid. The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid. The lattice of cuboids forms a data cube.

Cube: A Lattice of Cuboids

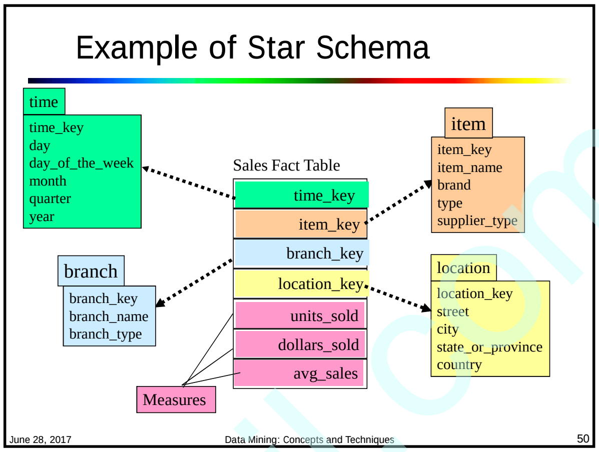




Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - Star schema: A fact table in the middle connected to a set of dimension tables
  - Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

June 28, 2017 Data Mining: Concepts and Techniques 49



Example of Star Schema

time
time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

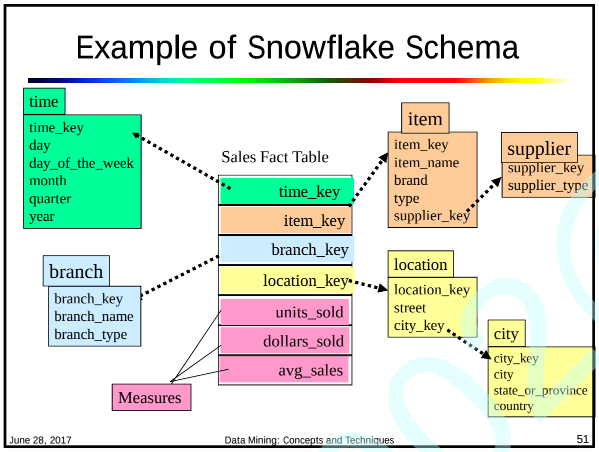
Measures

item
item_key
item_name
brand
type
supplier_type

branch
branch_key
branch_name
branch_type

location
location_key
street
city
state_or_province
country

June 28, 2017 Data Mining: Concepts and Techniques 50



Example of Snowflake Schema

time
time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

Measures

item
item_key
item_name
brand
type
supplier_key

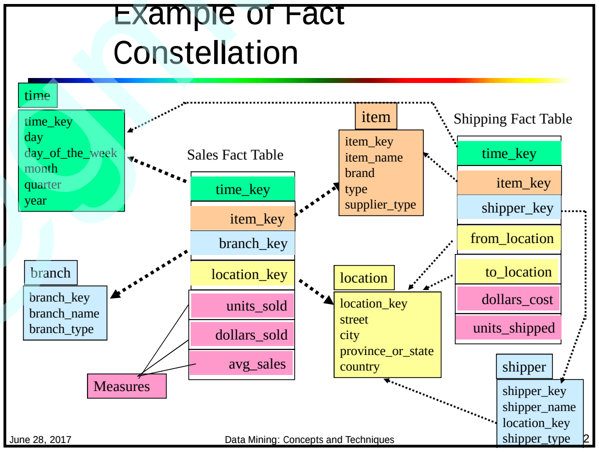
supplier
supplier_key
supplier_type

branch
branch_key
branch_name
branch_type

location
location_key
street
city_key

city
city_key
city
state_or_province
country

June 28, 2017 Data Mining: Concepts and Techniques 51



Example of Fact Constellation

time
time_key
day
day_of_the_week
month
quarter
year

Sales Fact Table
time_key
item_key
branch_key
location_key
units_sold
dollars_sold
avg_sales

Measures

item
item_key
item_name
brand
type
supplier_type

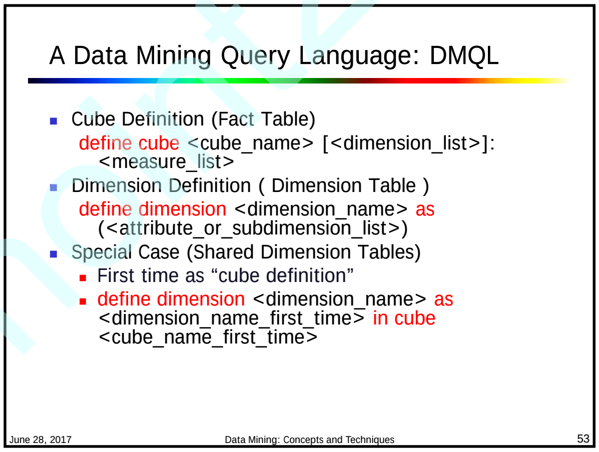
branch
branch_key
branch_name
branch_type

location
location_key
street
city
province_or_state
country

Shipping Fact Table
time_key
item_key
shipper_key
from_location
to_location
dollars_cost
units_shipped

shipper
shipper_key
shipper_name
location_key
shipper_type

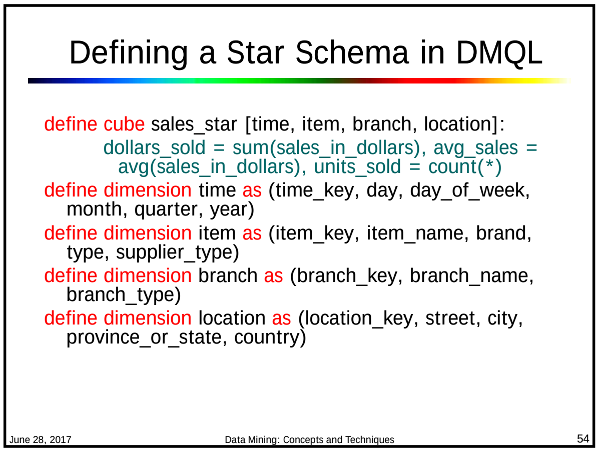
June 28, 2017 Data Mining: Concepts and Techniques 2



A Data Mining Query Language: DMQL

- Cube Definition (Fact Table)
  define cube <cube_name> [<dimension_list>]:
  <measure_list>
- Dimension Definition (Dimension Table)
  define dimension <dimension_name> as
  (<attribute_or_subdimension_list>)
- Special Case (Shared Dimension Tables)
  - First time as "cube definition"
  - define dimension <dimension_name> as
    <dimension_name_first_time> in cube
    <cube_name_first_time>

June 28, 2017 Data Mining: Concepts and Techniques 53



Defining a Star Schema in DMQL

define cube sales_star [time, item, branch, location]:
dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)

define dimension time as (time_key, day, day_of_week, month, quarter, year)

define dimension item as (item_key, item_name, brand, type, supplier_type)

define dimension branch as (branch_key, branch_name, branch_type)

define dimension location as (location_key, street, city, province_or_state, country)

June 28, 2017 Data Mining: Concepts and Techniques 54

Defining a Snowflake Schema in DMQL

define cube sales_snowflake [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)

define dimension time as (time_key, day, day_of_week, month, quarter, year)

define dimension item as (item_key, item_name, brand, type, supplier(supplier_key, supplier_type))

define dimension branch as (branch_key, branch_name, branch_type)

define dimension location as (location_key, street, city(city_key, province_or_state, country))

Defining a Fact Constellation in DMQL

define cube sales [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)

define dimension time as (time_key, day, day_of_week, month, quarter, year)

define dimension item as (item_key, item_name, brand, type, supplier_type)

define dimension branch as (branch_key, branch_name, branch_type)

define dimension location as (location_key, street, city, province_or_state, country)

define cube shipping [time, item, shipper, from_location, to_location]:

dollar_cost = sum(cost_in_dollars), unit_shipped = count(*)

define dimension time as time in cube sales

define dimension item as item in cube sales

define dimension shipper as (shipper_key, shipper_name, location as location in cube sales, shipper_type)

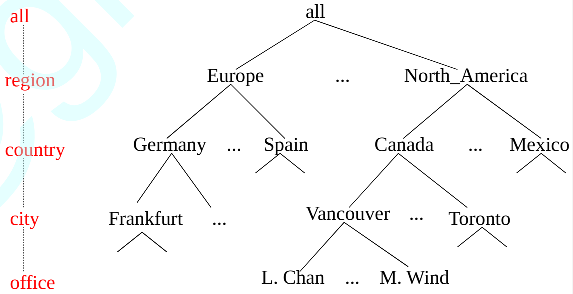
define dimension from_location as location in cube sales

define dimension to_location as location in cube sales

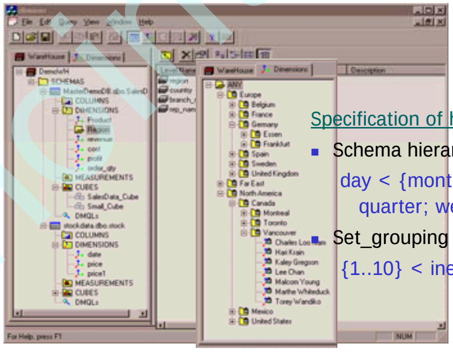
Measures: Three Categories

- distributive: if the result derived by applying the function to *n* aggregate values is the same as that derived by applying the function on all the data without partitioning.
  - E.g., count(), sum(), min(), max().
- algebraic: if it can be computed by an algebraic function with *M* arguments (where *M* is a bounded integer), each of which is obtained by applying a distributive aggregate function.
  - E.g., avg(), min_N(), standard_deviation().
- holistic: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., median(), mode(), rank().

A Concept Hierarchy: Dimension (location)



View of Warehouses and Hierarchies



Specification of hierarchies

- Schema hierarchy

  day < {month < quarter; week} < year

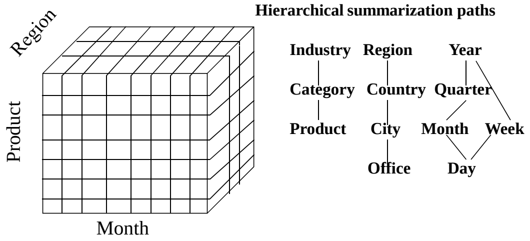
- Set_grouping hierarchy

  {1..10} < inexpensive

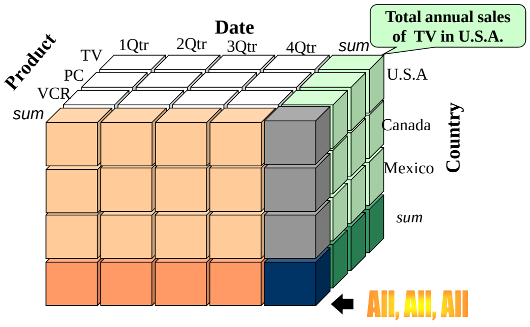
Multidimensional Data

- Sales volume as a function of product, month, and region

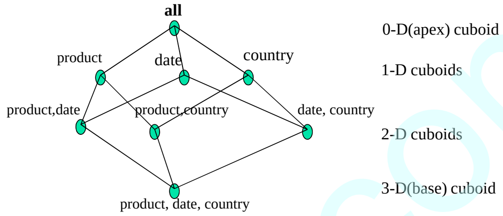
Dimensions: Product, Location, Time
Hierarchical summarization paths



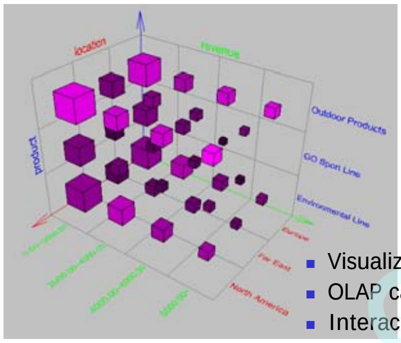
A Sample Data Cube



Cuboids Corresponding to the Cube



Browsing a Data Cube

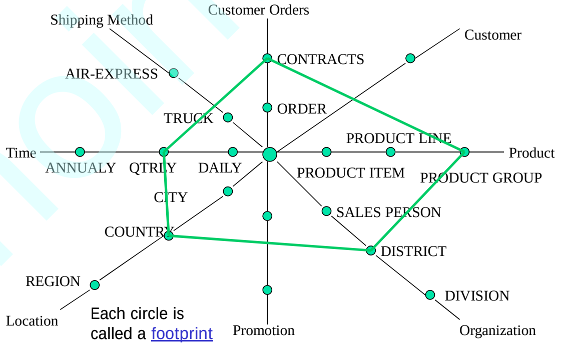


- Visualization
- OLAP capabilities
- Interactive manipulation

Typical OLAP Operations

- Roll up (drill-up): summarize data
  - *by climbing up hierarchy or by dimension reduction*
- Drill down (roll down): reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
- Slice and dice:
  - *project and select*
- Pivot (rotate):
  - *reorient the cube, visualization, 3D to series of 2D planes.*
- Other operations
  - *drill across: involving (across) more than one fact table*
  - *drill through: through the bottom level of the cube to its back-end relational tables (using SQL)*

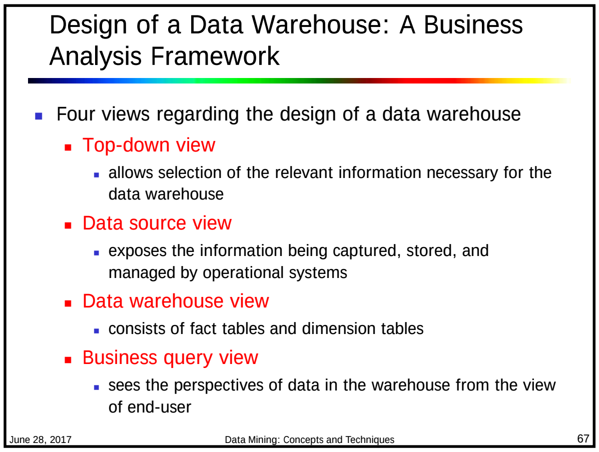
A Star-Net Query Model



Each circle is called a footprint

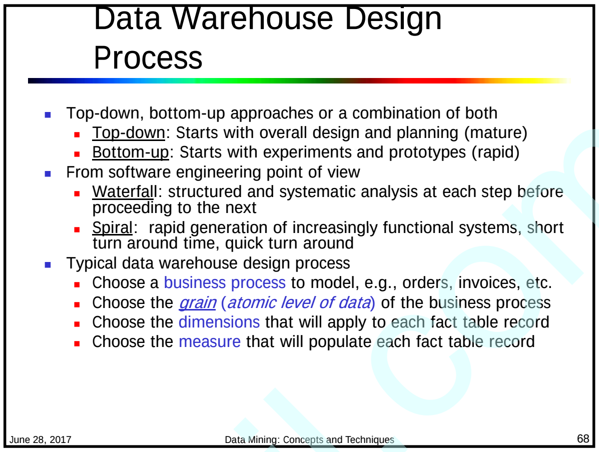
Chapter 2: Data Warehousing and OLAP Technology for Data Mining

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining



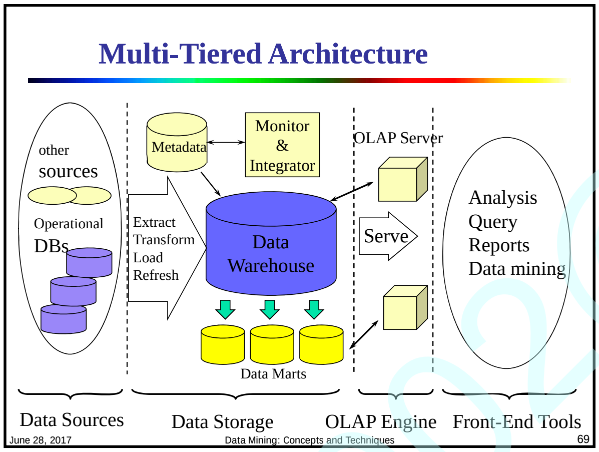
Design of a Data Warehouse: A Business Analysis Framework

- Four views regarding the design of a data warehouse
  - Top-down view
    - allows selection of the relevant information necessary for the data warehouse
  - Data source view
    - exposes the information being captured, stored, and managed by operational systems
  - Data warehouse view
    - consists of fact tables and dimension tables
  - Business query view
    - sees the perspectives of data in the warehouse from the view of end-user

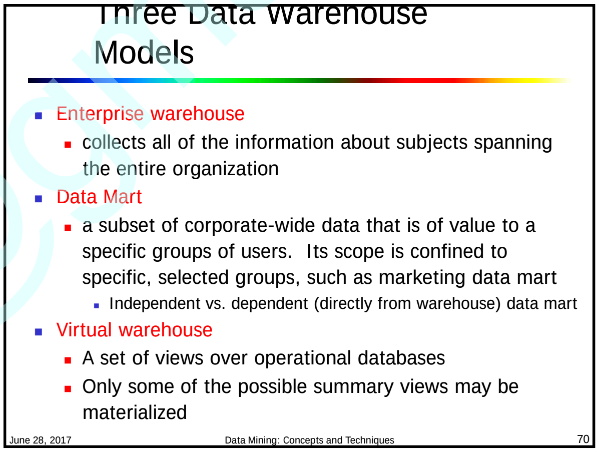


Data Warehouse Design Process

- Top-down, bottom-up approaches or a combination of both
  - Top-down: Starts with overall design and planning (mature)
  - Bottom-up: Starts with experiments and prototypes (rapid)
- From software engineering point of view
  - Waterfall: structured and systematic analysis at each step before proceeding to the next
  - Spiral: rapid generation of increasingly functional systems, short turn around time, quick turn around
- Typical data warehouse design process
  - Choose a business process to model, e.g., orders, invoices, etc.
  - Choose the *grain* (*atomic level of data*) of the business process
  - Choose the dimensions that will apply to each fact table record
  - Choose the measure that will populate each fact table record

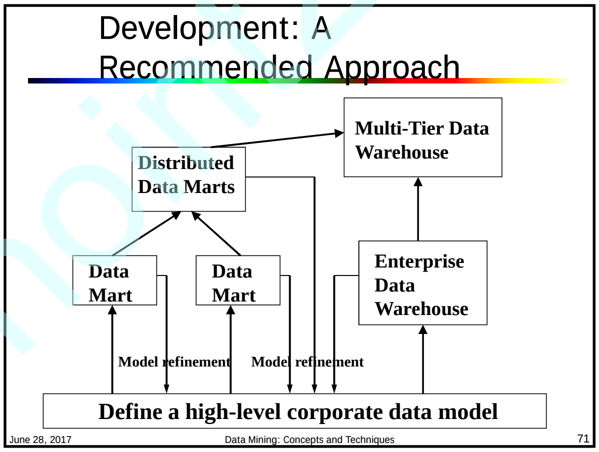


Multi-Tiered Architecture



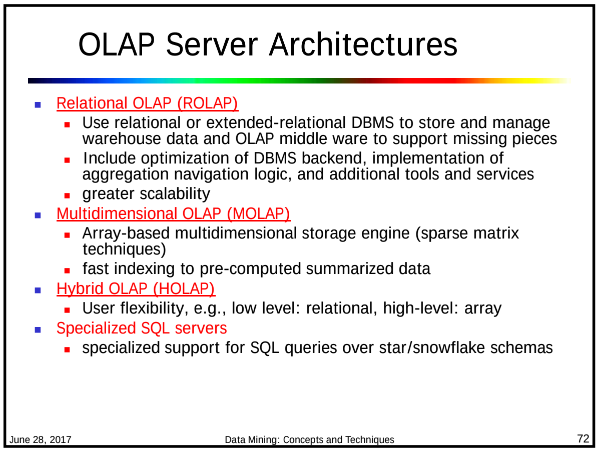
Three Data Warehouse Models

- Enterprise warehouse
  - collects all of the information about subjects spanning the entire organization
- Data Mart
  - a subset of corporate-wide data that is of value to a specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart
    - Independent vs. dependent (directly from warehouse) data mart
- Virtual warehouse
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized



Development: A Recommended Approach

Multi-Tier Data Warehouse
Distributed Data Marts
Data Mart
Data Mart
Enterprise Data Warehouse
Model refinement
Model refinement
Define a high-level corporate data model



OLAP Server Architectures

- Relational OLAP (ROLAP)
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware to support missing pieces
  - Include optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services
  - greater scalability
- Multidimensional OLAP (MOLAP)
  - Array-based multidimensional storage engine (sparse matrix techniques)
  - fast indexing to pre-computed summarized data
- Hybrid OLAP (HOLAP)
  - User flexibility, e.g., low level: relational, high-level: array
- Specialized SQL servers
  - specialized support for SQL queries over star/snowflake schemas

Chapter 2: Data Warehousing and OLAP Technology for Data Mining

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

Efficient Data Cube Computation

- Data cube can be viewed as a lattice of cuboids
  - The bottom-most cuboid is the base cuboid
  - The top-most cuboid (apex) contains only one cell
  - How many cuboids in an n-dimensional cube with L levels?

$$T = \prod_{i=1}^{n} (L_i + 1)$$

- Materialization of data cube
  - Materialize <u>every</u> (cuboid) (full materialization), <u>none</u> (no materialization), or <u>some (partial materialization)</u>
  - Selection of which cuboids to materialize
    - Based on size, sharing, access frequency, etc.

Cube Operation

- Cube definition and computation in DMQL

  define cube sales[item, city, year]: sum(sales_in_dollars)

  compute cube sales

- Transform it into a SQL-like language (with a new operator cube by, introduced by Gray et al.'96)

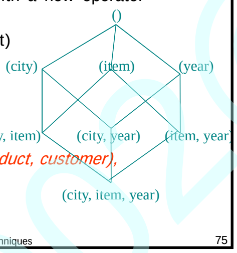
  SELECT item, city, year, SUM (amount)

  FROM SALES

  CUBE BY item, city, year

- Need compute the following Group-Bys

  (date, product, customer),
  (date,product),(date, customer), (product, customer),
  (date), (product), (customer)
  ()



Cube Computation: ROLAP-Based Method

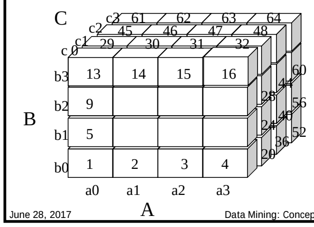
- Efficient cube computation methods
  - ROLAP-based cubing algorithms (Agarwal et al'96)
  - Array-based cubing algorithm (Zhao et al'97)
  - Bottom-up computation method (Beyer & Ramarkrishnan'99)
  - H-cubing technique (Han, Pei, Dong & Wang:SIGMOD'01)
- ROLAP-based cubing algorithms
  - Sorting, hashing, and grouping operations are applied to the dimension attributes in order to reorder and cluster related tuples
  - Grouping is performed on some sub-aggregates as a "partial grouping step"
  - Aggregates may be computed from previously computed aggregates, rather than from the base fact table

Cube Computation: ROLAP-Based Method (2)

- This is not in the textbook but in a research paper
- Hash/sort based methods (Agarwal et. al. VLDB'96)
  - **Smallest-parent:** computing a cuboid from the smallest, previously computed cuboid
  - **Cache-results:** caching results of a cuboid from which other cuboids are computed to reduce disk I/Os
  - **Amortize-scans:** computing as many as possible cuboids at the same time to amortize disk reads
  - **Share-sorts:** sharing sorting costs cross multiple cuboids when sort-based method is used
  - **Share-partitions:** sharing the partitioning cost across multiple cuboids when hash-based algorithms are used

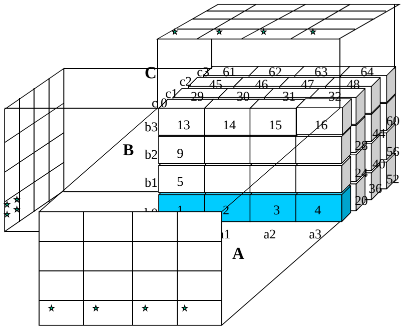
Multi-way Array Aggregation for Cube Computation

- Partition arrays into chunks (a small subcube which fits in memory).
- Compressed sparse array addressing: (chunk_id, offset)
- Compute aggregates in "multiway" by visiting cube cells in the order which minimizes the # of times to visit each cell, and reduces memory access and storage cost.

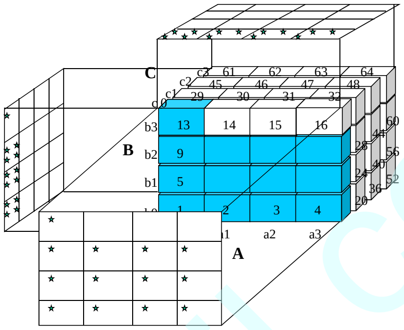


What is the best traversing order to do multi-way aggregation?

Multi-way Array Aggregation for Cube Computation



Multi-way Array Aggregation for Cube Computation



Multi-Way Array Aggregation for Cube Computation (Cont.)

- Method: the planes should be sorted and computed according to their size in ascending order.
  - See the details of Example 2.12 (pp. 75-78)
  - Idea: keep the smallest plane in the main memory, fetch and compute only one chunk at a time for the largest plane
- Limitation of the method: computing well only for a small number of dimensions
  - If there are a large number of dimensions, “bottom-up computation” and iceberg cube computation methods can be explored

Indexing OLAP Data: Bitmap Index

- Index on a particular column
- Each value in the column has a bit vector: bit-op is fast
- The length of the bit vector: # of records in the base table
- The *i*-th bit is set if the *i*-th row of the base table has the value for the indexed column
- not suitable for high cardinality domains

Base table

Cust	Region	Type
C1	Asia	Retail
C2	Europe	Dealer
C3	Asia	Dealer
C4	America	Retail
C5	Europe	Dealer

Index on Region

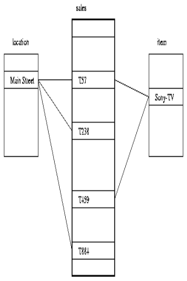
RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

Index on Type

RecID	Retail	Dealer
1	1	0
2	0	1
3	0	1
4	1	0
5	0	1

Indexing OLAP Data: Join Indices

- Join index: JI(R-id, S-id) where R (R-id, …) ▷◁ S (S-id, …)
- Traditional indices map the values to a list of record ids
  - It materializes relational join in JI file and speeds up relational join — a rather costly operation
- In data warehouses, join index relates the values of the dimensions of a start schema to rows in the fact table.
  - E.g. fact table: *Sales* and two dimensions *city* and *product*
    - A join index on *city* maintains for each distinct city a list of R-IDs of the tuples recording the Sales in the city
  - Join indices can span multiple dimensions



Efficient Processing OLAP Queries

- Determine which operations should be performed on the available cuboids:
  - transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g, dice = selection + projection
- Determine to which materialized cuboid(s) the relevant operations should be applied.
- Exploring indexing structures and compressed vs. dense array structures in MOLAP

Metadata Repository

- Meta data is the data defining warehouse objects. It has the following kinds
  - Description of the structure of the warehouse
    - schema, view, dimensions, hierarchies, derived data defn, data mart locations and contents
  - Operational meta-data
    - data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (warehouse usage statistics, error reports, audit trails)
  - The algorithms used for summarization
  - The mapping from operational environment to the data warehouse
  - Data related to system performance
    - warehouse schema, view and derived data definitions
  - Business data
    - business terms and definitions, ownership of data, charging policies

Data Warehouse Back-End Tools and Utilities

- Data extraction:
  - get data from multiple, heterogeneous, and external sources
- Data cleaning:
  - detect errors in the data and rectify them when possible
- Data transformation:
  - convert data from legacy or host format to warehouse format
- Load:
  - sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions
- Refresh
  - propagate the updates from the data sources to the warehouse

Chapter 2: Data Warehousing and OLAP Technology for Data Mining

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

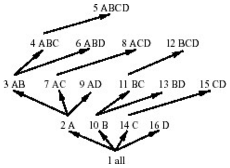
Iceberg Cube

- Computing only the cuboid cells whose count or other aggregates satisfying the condition:

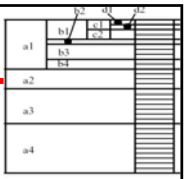
  HAVING COUNT(*) >= *minsup*
- Motivation
  - Only a small portion of cube cells may be "above the water'' in a sparse cube
  - Only calculate "interesting" data—data above certain threshold
  - Suppose 100 dimensions, only 1 base cell. How many aggregate (non-base) cells if count >= 1? What about count >= 2?

Bottom-Up Computation (BUC)

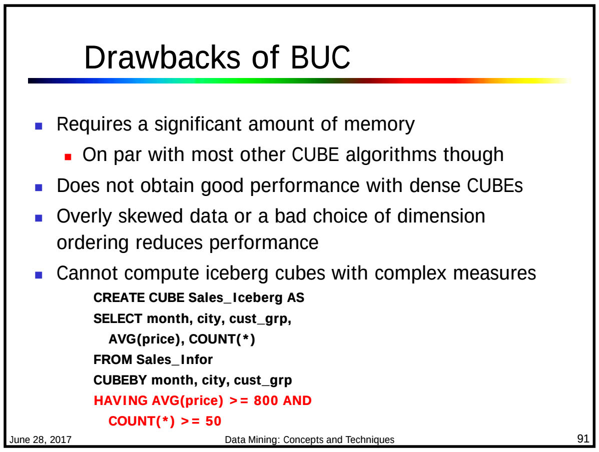
- BUC (Beyer & Ramakrishnan, SIGMOD'99)
- Bottom-up vs. top-down?—depending on how you view it!
- Apriori property:
  - Aggregate the data, then move to the next level
  - If *minsup* is not met, stop!
- If *minsup* = 1 ⇒ compute full CUBE!



Partitioning



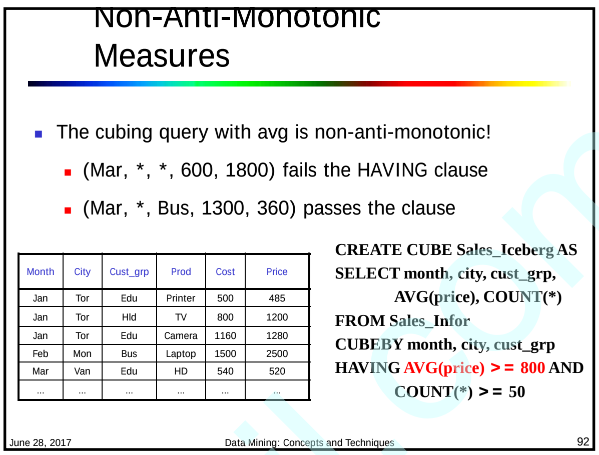
- Usually, entire data set can't fit in main memory
- Sort *distinct* values, partition into blocks that fit
- Continue processing
- Optimizations
  - Partitioning
    - External Sorting, Hashing, Counting Sort
  - Ordering dimensions to encourage pruning
    - Cardinality, Skew, Correlation
  - Collapsing duplicates
    - Can't do holistic aggregates anymore!



Drawbacks of BUC

- Requires a significant amount of memory
  - On par with most other CUBE algorithms though
- Does not obtain good performance with dense CUBEs
- Overly skewed data or a bad choice of dimension ordering reduces performance
- Cannot compute iceberg cubes with complex measures

```
CREATE CUBE Sales_Iceberg AS
SELECT month, city, cust_grp,
   AVG(price), COUNT(*)
FROM Sales_Infor
CUBEBY month, city, cust_grp
HAVING AVG(price) >= 800 AND
   COUNT(*) >= 50
```

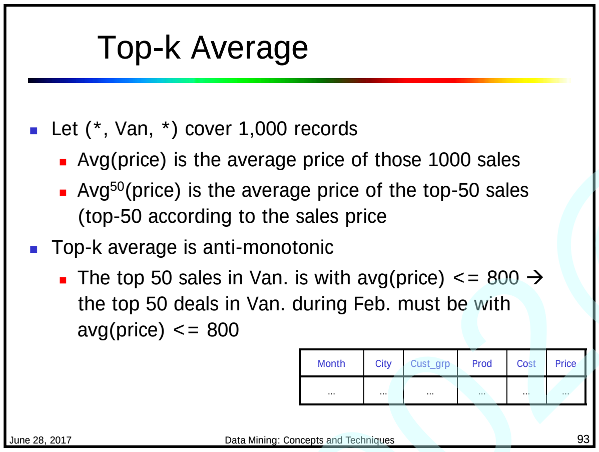


Non-Anti-Monotonic Measures

- The cubing query with avg is non-anti-monotonic!
  - (Mar, *, *, 600, 1800) fails the HAVING clause
  - (Mar, *, Bus, 1300, 360) passes the clause

Month	City	Cust_grp	Prod	Cost	Price
Jan	Tor	Edu	Printer	500	485
Jan	Tor	Hld	TV	800	1200
Jan	Tor	Edu	Camera	1160	1280
Feb	Mon	Bus	Laptop	1500	2500
Mar	Van	Edu	HD	540	520
…	…	…	…	…	…

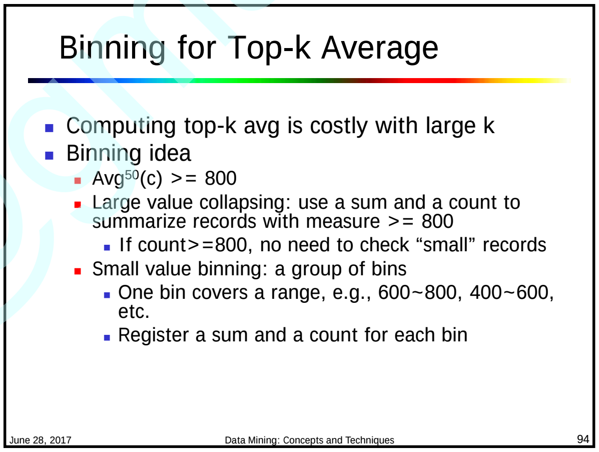
```
CREATE CUBE Sales_Iceberg AS
SELECT month, city, cust_grp,
        AVG(price), COUNT(*)
FROM Sales_Infor
CUBEBY month, city, cust_grp
HAVING AVG(price) >= 800 AND
        COUNT(*) >= 50
```



Top-k Average

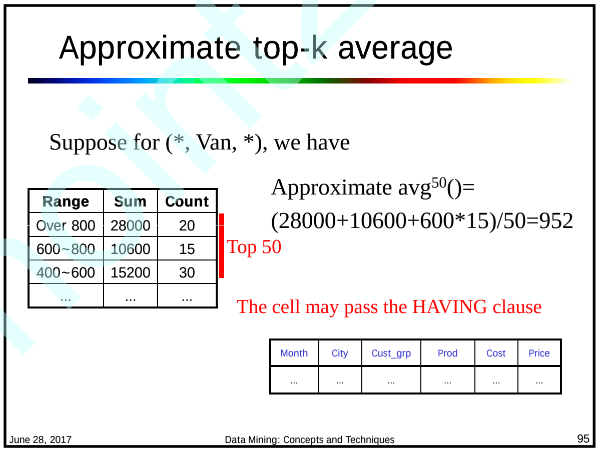
- Let (*, Van, *) cover 1,000 records
  - Avg(price) is the average price of those 1000 sales
  - Avg^{50}(price) is the average price of the top-50 sales (top-50 according to the sales price
- Top-k average is anti-monotonic
  - The top 50 sales in Van. is with avg(price) <= 800 → the top 50 deals in Van. during Feb. must be with avg(price) <= 800

Month	City	Cust_grp	Prod	Cost	Price
…	…	…	…	…	…



Binning for Top-k Average

- Computing top-k avg is costly with large k
- Binning idea
  - Avg^{50}(c) >= 800
  - Large value collapsing: use a sum and a count to summarize records with measure >= 800
    - If count>=800, no need to check "small" records
  - Small value binning: a group of bins
    - One bin covers a range, e.g., 600~800, 400~600, etc.
    - Register a sum and a count for each bin



Approximate top-k average

Suppose for (*, Van, *), we have

Range	Sum	Count
Over 800	28000	20
600~800	10600	15
400~600	15200	30
…	…	…

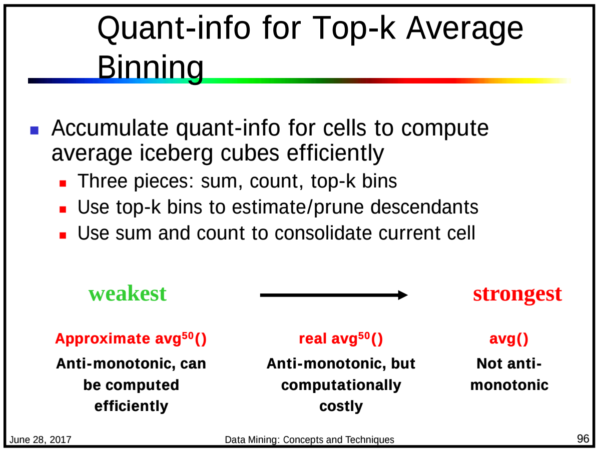
Top 50

Approximate avg^{50}()=

(28000+10600+600*15)/50=952

The cell may pass the HAVING clause

Month	City	Cust_grp	Prod	Cost	Price
…	…	…	…	…	…



Quant-info for Top-k Average Binning

- Accumulate quant-info for cells to compute average iceberg cubes efficiently
  - Three pieces: sum, count, top-k bins
  - Use top-k bins to estimate/prune descendants
  - Use sum and count to consolidate current cell

weakest ⟶ strongest

Approximate avg^{50}()	real avg^{50}()	avg()
Anti-monotonic, can be computed efficiently	Anti-monotonic, but computationally costly	Not anti-monotonic

An Efficient Iceberg Cubing Method: Top-k H-Cubing

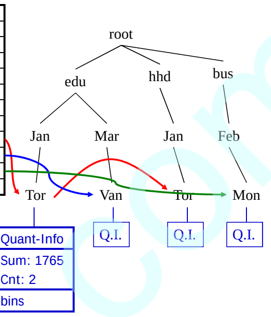
- One can revise Apriori or BUC to compute a top-k avg iceberg cube. This leads to top-k-Apriori and top-k BUC.
- Can we compute iceberg cube more efficiently?
- Top-k H-cubing: an efficient method to compute iceberg cubes with average measure
- H-tree: a hyper-tree structure
- H-cubing: computing iceberg cubes using H-tree

H-tree: A Prefix Hyper-tree

Header table

Attr. Val.	Quant-Info	Side-link
Edu	Sum:2285 ...	
Hhd	...	
Bus	...	
...	...	
Jan	...	
Feb	...	
...	...	
Tor	...	
Van	...	
Mon	...	
...	...	

Month	City	Cust_grp	Prod	Cost	Price
Jan	Tor	Edu	Printer	500	485
Jan	Tor	Hhd	TV	800	1200
Jan	Tor	Edu	Camera	1160	1280
Feb	Mon	Bus	Laptop	1500	2500
Mar	Van	Edu	HD	540	520
...	...	...	...	...	...



Properties of H-tree

- Construction cost: a single database scan
- Completeness: It contains the complete information needed for computing the iceberg cube
- Compactness: # of nodes ☯ n*m+1
  - n: # of tuples in the table
  - m: # of attributes

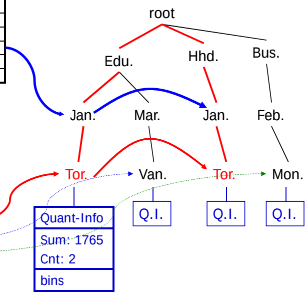
Computing Cells Involving Dimension City

Header Table H_{Tor}

Attr. Val.	Q.I.	Side-link
Edu	...	
Hhd	...	
Bus	...	
...	...	
Jan	...	
Feb	...	
...	...	

From (*, *, Tor) to (*, Jan, Tor)

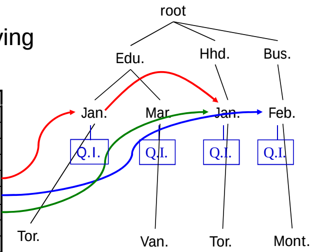
Attr. Val.	Quant-Info	Side-link
Edu	Sum:2285 ...	
Hhd	...	
Bus	...	
...	...	
Jan	...	
Feb	...	
...	...	
Tor	...	
Van	...	
Mon	...	
...	...	



Computing Cells Involving Month But No City

1. Roll up quant-info
2. Compute cells involving month but no city

Attr. Val.	Quant-Info	Side-link
Edu.	Sum:2285 ...	
Hhd.	...	
Bus.	...	
...	...	
Jan.	...	
Feb.	...	
Mar.	...	
...	...	
Tor.	...	
Van.	...	
Mont.	...	
...	...	

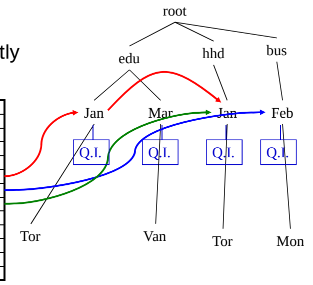


Top-k OK mark: if Q.I. in a child passes top-k avg threshold, so does its parents. No binning is needed!

Computing Cells Involving Only Cust_grp

Check header table directly

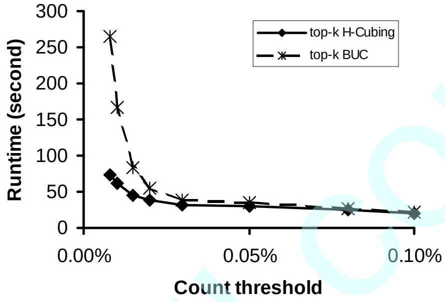
Attr. Val.	Quant-Info	Side-link
Edu	Sum:2285 ...	
Hhd	...	
Bus	...	
...	...	
Jan	...	
Feb	...	
Mar	...	
...	...	
Tor	...	
Van	...	
Mon	...	
...	...	



Properties of H-Cubing

- Space cost
  - an H-tree
  - a stack of up to (m-1) header tables
- One database scan
- Main memory-based tree traversal & side-links updates
- Top-k_OK marking

Scalability w.r.t. Count Threshold (No min_avg Setting)



Computing Iceberg Cubes with Other Complex Measures

- Computing other complex measures
  - Key point: find a function which is weaker but ensures certain anti-monotonicity
- Examples
  - Avg() ≤ v: $avg_k(c) \le v$ (bottom-k avg)
  - Avg() ≥ v only (no count): max(price) ≥ v
  - Sum(profit) (profit can be negative):
    - p_sum(c) ≥ v if p_count(c) ≥ k; or otherwise, $sum^k(c) \ge v$
  - Others: conjunctions of multiple conditions

Discussion: Other Issues

- Computing iceberg cubes with more complex measures?
  - No general answer for holistic measures, e.g., median, mode, rank
  - A research theme even for complex algebraic functions, e.g., standard_dev, variance
- Dynamic vs . static computation of iceberg cubes
  - v and k are only available at query time
  - Setting reasonably low parameters for most nontrivial cases
- Memory-hog? what if the cubing is too big to fit in memory?—projection and then cubing

Condensed Cube

- W. Wang, H. Lu, J. Feng, J. X. Yu, Condensed Cube: An Effective Approach to Reducing Data Cube Size. ICDE'02.
- Iceberg cube cannot solve all the problems
  - Suppose 100 dimensions, only 1 base cell with count = 10. How many aggregate (non-base) cells if count >= 10?
- Condensed cube
  - Only need to store one cell (a_1, a_2, …, a_{100}, 10), which represents all the corresponding aggregate cells
  - Adv.
    - Fully precomputed cube without compression
  - Efficient computation of the minimal condensed cube

Chapter 2: Data Warehousing and OLAP Technology for Data Mining

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

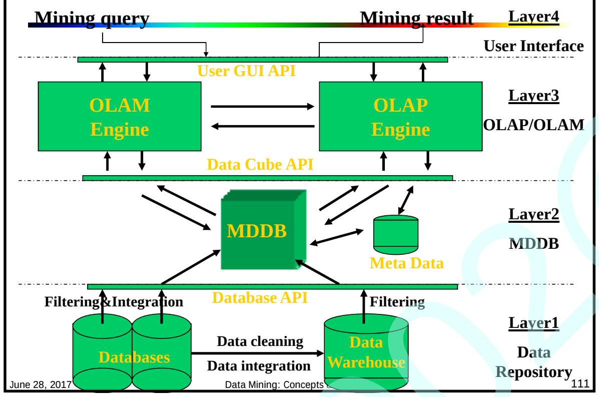
Data Warehouse Usage

- Three kinds of data warehouse applications
  - Information processing
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - Analytical processing
    - multidimensional analysis of data warehouse data
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - Data mining
    - knowledge discovery from hidden patterns
    - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.
- Differences among the three tasks

From On-Line Analytical Processing to On Line Analytical Mining (OLAM)

- Why online analytical mining?
  - High quality of data in data warehouses
    - DW contains integrated, consistent, cleaned data
  - Available information processing structure surrounding data warehouses
    - ODBC, OLEDB, Web accessing, service facilities, reporting and OLAP tools
  - OLAP-based exploratory data analysis
    - mining with drilling, dicing, pivoting, etc.
  - On-line selection of data mining functions
    - integration and swapping of multiple mining functions, algorithms, and tasks.
- Architecture of OLAM

An OLAM Architecture



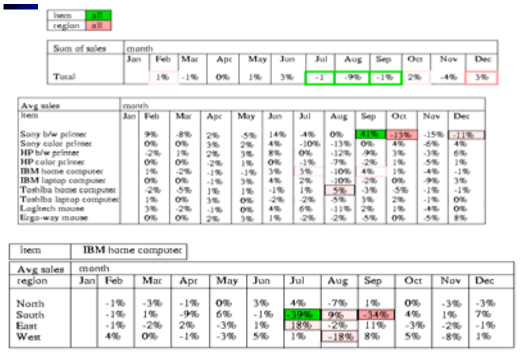
Discovery-Driven Exploration of Data Cubes

- Hypothesis-driven
  - exploration by user, huge search space
- Discovery-driven (Sarawagi, et al.'98)
  - Effective navigation of large OLAP data cubes
  - pre-compute measures indicating exceptions, guide user in the data analysis, at all levels of aggregation
  - Exception: significantly different from the value anticipated, based on a statistical model
  - Visual cues such as background color are used to reflect the degree of exception of each cell

Kinds of Exceptions and their Computation

- Parameters
  - SelfExp: surprise of cell relative to other cells at same level of aggregation
  - InExp: surprise beneath the cell
  - PathExp: surprise beneath cell for each drill-down path
- Computation of exception indicator (modeling fitting and computing SelfExp, InExp, and PathExp values) can be overlapped with cube construction
- Exception themselves can be stored, indexed and retrieved like precomputed aggregates

Examples: Discovery-Driven Data Cubes



item	all
region	all

Sum of sales	month											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Total		1%	-1%	0%	1%	3%	-1	-9%	-1%	2%	-4%	3%

Avg sales	month											
item	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Sony b/w printer		9%	-8%	2%	-5%	14%	-4%	0%	41%	-13%	-15%	-11%
Sony color printer		0%	0%	3%	2%	4%	-10%	-13%	0%	4%	-6%	4%
HP b/w printer		-2%	1%	2%	3%	8%	0%	-12%	-9%	3%	-3%	6%
HP color printer		0%	0%	-2%	1%	0%	-1%	-7%	-2%	1%	-5%	1%
IBM home computer		1%	-2%	-1%	-1%	3%	3%	-10%	4%	1%	-4%	-1%
IBM laptop computer		0%	0%	-1%	3%	4%	2%	-10%	-2%	0%	-9%	3%
Toshiba home computer		-2%	-5%	1%	1%	-1%	1%	5%	-3%	-5%	-1%	-1%
Toshiba laptop computer		1%	0%	3%	0%	-2%	-2%	-5%	3%	2%	-1%	0%
Logitech mouse		3%	-2%	-1%	0%	4%	6%	-11%	2%	1%	-4%	0%
Ergo-way mouse		0%	0%	2%	3%	1%	-2%	-2%	-5%	0%	-5%	8%

item	IBM home computer

Avg sales	month											
region	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
North		-1%	-3%	-1%	0%	3%	4%	-7%	1%	0%	-3%	-3%
South		-1%	1%	-9%	6%	-1%	-39%	9%	-34%	4%	1%	7%
East		-1%	-2%	2%	-3%	1%	18%	-2%	11%	-3%	-2%	-1%
West		4%	0%	-1%	-3%	5%	1%	-18%	8%	5%	-8%	1%

Complex Aggregation at Multiple Granularities: Multi-Feature Cubes

- Multi-feature cubes (Ross, et al. 1998): Compute complex queries involving multiple dependent aggregates at multiple granularities
- Ex. Grouping by all subsets of {item, region, month}, find the maximum price in 1997 for each group, and the total sales among all maximum price tuples

```
select item, region, month, max(price), sum(R.sales)
from purchases
where year = 1997
cube by item, region, month: R
such that R.price = max(price)
```

- Continuing the last example, among the max price tuples, find the min and max shelf live, and find the fraction of the total sales due to tuple that have min shelf life within the set of all max price tuples

Cube-Gradient (Cubegrade)

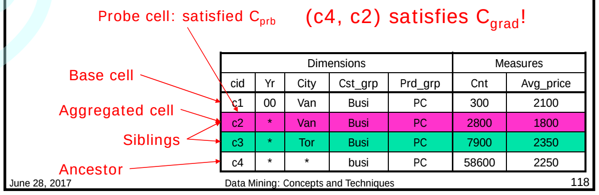
- Analysis of changes of sophisticated measures in multi-dimensional spaces
  - Query: changes of average house price in Vancouver in '00 comparing against '99
  - Answer: Apts in West went down 20%, houses in Metrotown went up 10%
- Cubegrade problem by Imielinski et al.
  - Changes in dimensions → changes in measures
  - Drill-down, roll-up, and mutation

From Cubegrade to Multi-dimensional Constrained Gradients in Data Cubes

- Significantly more expressive than association rules
  - Capture trends in user-specified measures
- Serious challenges
  - Many trivial cells in a cube → "significance constraint" to prune trivial cells
  - Numerate pairs of cells → "probe constraint" to select a subset of cells to examine
  - Only interesting changes wanted→ "gradient constraint" to capture significant changes

MD Constrained Gradient Mining

- Significance constraint C_{sig}: (cnt≥100)
- Probe constraint $C_{prb:}$ (city="Van", cust_grp="busi", prod_grp="*")
- Gradient constraint $C_{grad}(c_g, c_p)$: (avg_price(c_g)/avg_price(c_p)≥1.3)


Probe cell: satisfied C_{prb}
(c4, c2) satisfies C_{grad}!
Base cell
Aggregated cell
Siblings
Ancestor

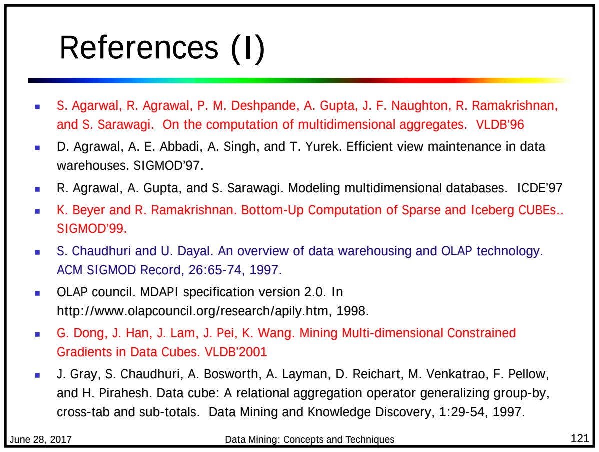
Dimensions					Measures	
cid	Yr	City	Cst_grp	Prd_grp	Cnt	Avg_price
c1	00	Van	Busi	PC	300	2100
c2	*	Van	Busi	PC	2800	1800
c3	*	Tor	Busi	PC	7900	2350
c4	*	*	busi	PC	58600	2250

A LiveSet-Driven Algorithm

- Compute probe cells using C_{sig} and C_{prb}
  - The set of probe cells P is often very small
- Use probe P and constraints to find gradients
  - Pushing selection deeply
  - Set-oriented processing for probe cells
  - Iceberg growing from low to high dimensionalities
  - Dynamic pruning probe cells during growth
  - Incorporating efficient iceberg cubing method

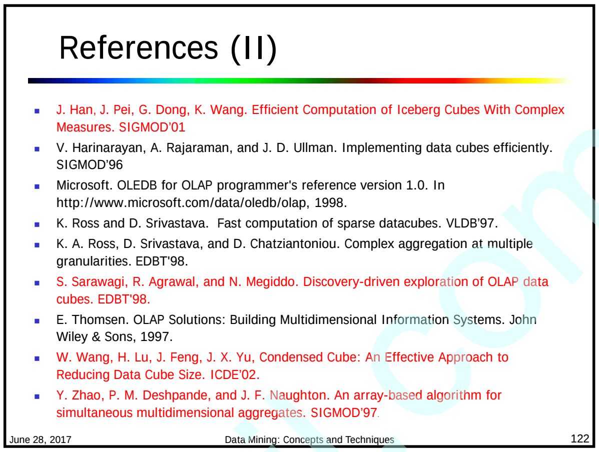
Summary

- Data warehouse
- A multi-dimensional model of a data warehouse
  - Star schema, snowflake schema, fact constellations
  - A data cube consists of dimensions & measures
- OLAP operations: drilling, rolling, slicing, dicing and pivoting
- OLAP servers: ROLAP, MOLAP, HOLAP
- Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - Multiway array aggregation
  - Bitmap index and join index implementations
- Further development of data cube technology
  - Discovery-drive and multi-feature cubes
  - From OLAP to OLAM (on-line analytical mining)



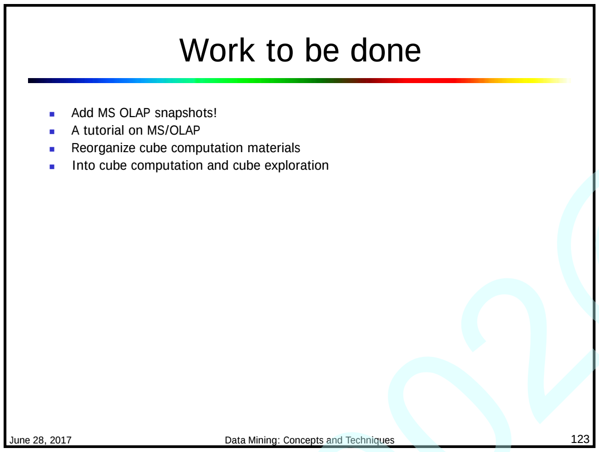
References (I)

- S. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. VLDB'96
- D. Agrawal, A. E. Abbadi, A. Singh, and T. Yurek. Efficient view maintenance in data warehouses. SIGMOD'97.
- R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. ICDE'97
- K. Beyer and R. Ramakrishnan. Bottom-Up Computation of Sparse and Iceberg CUBEs.. SIGMOD'99.
- S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26:65-74, 1997.
- OLAP council. MDAPI specification version 2.0. In http://www.olapcouncil.org/research/apily.htm, 1998.
- G. Dong, J. Han, J. Lam, J. Pei, K. Wang. Mining Multi-dimensional Constrained Gradients in Data Cubes. VLDB'2001
- J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. Data Mining and Knowledge Discovery, 1:29-54, 1997.



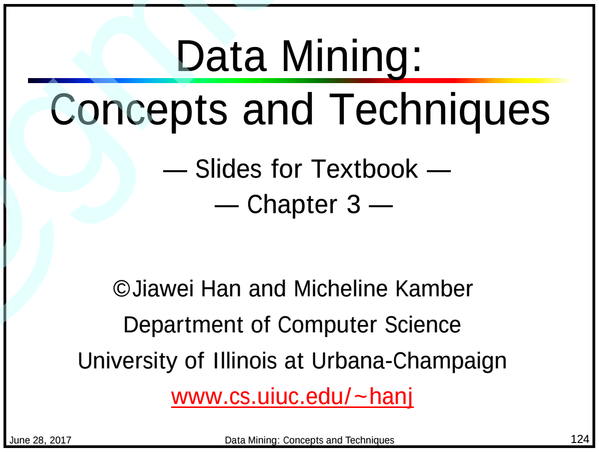
References (II)

- J. Han, J. Pei, G. Dong, K. Wang. Efficient Computation of Iceberg Cubes With Complex Measures. SIGMOD'01
- V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. SIGMOD'96
- Microsoft. OLEDB for OLAP programmer's reference version 1.0. In http://www.microsoft.com/data/oledb/olap, 1998.
- K. Ross and D. Srivastava. Fast computation of sparse datacubes. VLDB'97.
- K. A. Ross, D. Srivastava, and D. Chatziantoniou. Complex aggregation at multiple granularities. EDBT'98.
- S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. EDBT'98.
- E. Thomsen. OLAP Solutions: Building Multidimensional Information Systems. John Wiley & Sons, 1997.
- W. Wang, H. Lu, J. Feng, J. X. Yu, Condensed Cube: An Effective Approach to Reducing Data Cube Size. ICDE'02.
- Y. Zhao, P. M. Deshpande, and J. F. Naughton. An array-based algorithm for simultaneous multidimensional aggregates. SIGMOD'97.



Work to be done

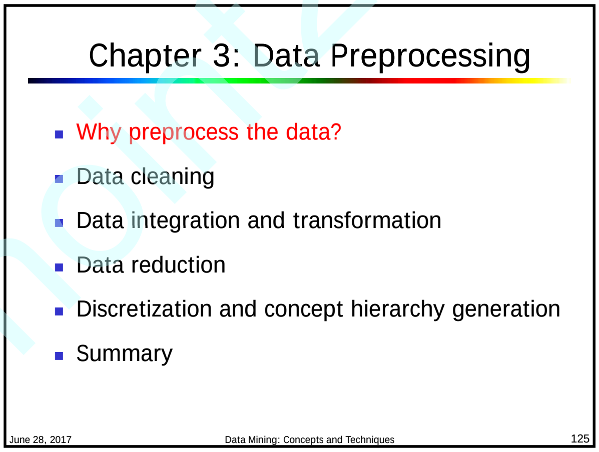
- Add MS OLAP snapshots!
- A tutorial on MS/OLAP
- Reorganize cube computation materials
- Into cube computation and cube exploration



Data Mining: Concepts and Techniques

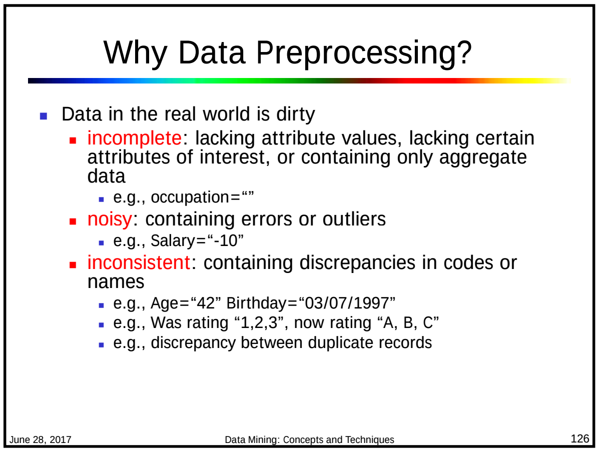
— Slides for Textbook —
— Chapter 3 —

©Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj



Chapter 3: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary



Why Data Preprocessing?

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
    - e.g., occupation=""
  - noisy: containing errors or outliers
    - e.g., Salary="-10"
  - inconsistent: containing discrepancies in codes or names
    - e.g., Age="42" Birthday="03/07/1997"
    - e.g., Was rating "1,2,3", now rating "A, B, C"
    - e.g., discrepancy between duplicate records

Why Is Data Dirty?

- Incomplete data comes from
  - n/a data value when collected
  - different consideration between the time when the data was collected and when it is analyzed.
  - human/hardware/software problems
- Noisy data comes from the process of data
  - collection
  - entry
  - transmission
- Inconsistent data comes from
  - Different data sources
  - Functional dependency violation

Why Is Data Preprocessing Important?

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
    - e.g., duplicate or missing data may cause incorrect or even misleading statistics.
  - Data warehouse needs consistent integration of quality data
- Data extraction, cleaning, and transformation comprises the majority of the work of building a data warehouse. —Bill Inmon

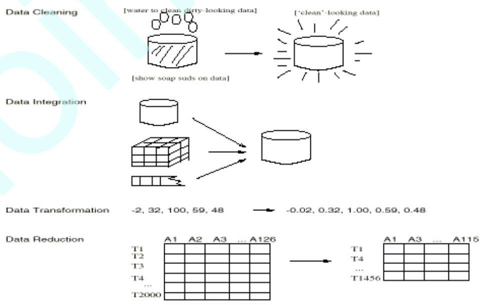
Multi-Dimensional Measure of Data Quality

- A well-accepted multidimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Value added
  - Interpretability
  - Accessibility
- Broad categories:
  - intrinsic, contextual, representational, and accessibility.

Major Tasks in Data Preprocessing

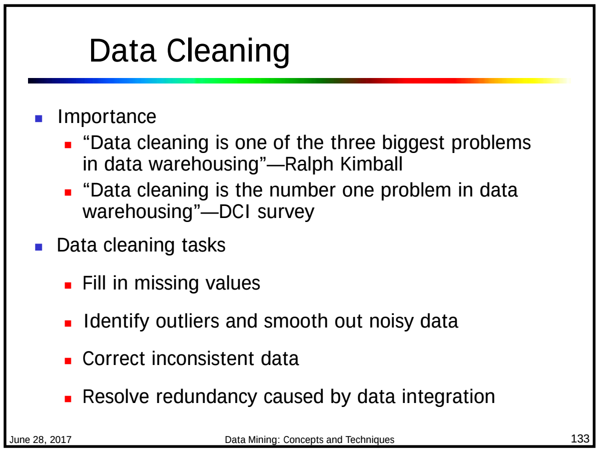
- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
  - Part of data reduction but with particular importance, especially for numerical data

Forms of data preprocessing



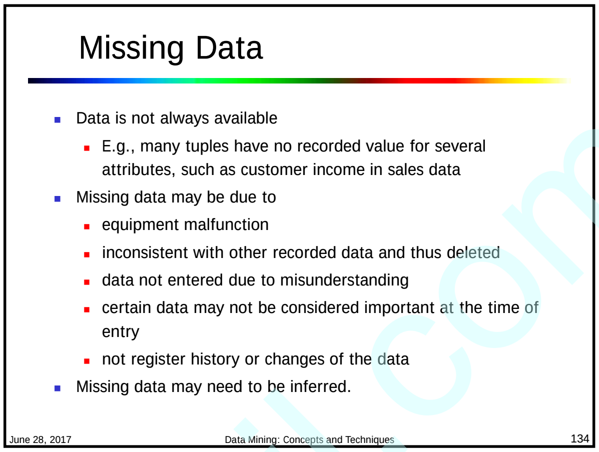
Chapter 3: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary



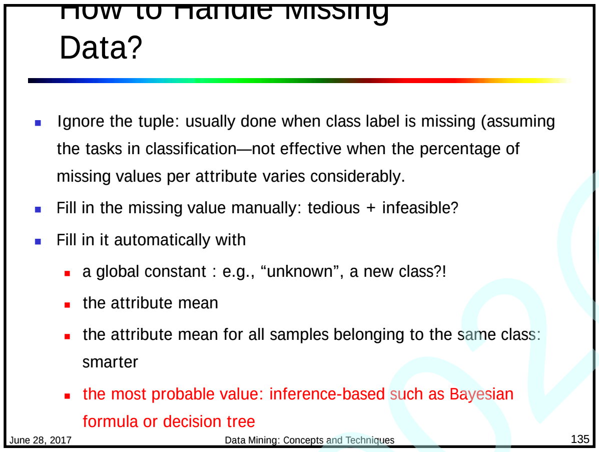
Data Cleaning

- Importance
  - "Data cleaning is one of the three biggest problems in data warehousing"—Ralph Kimball
  - "Data cleaning is the number one problem in data warehousing"—DCI survey
- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data
  - Resolve redundancy caused by data integration



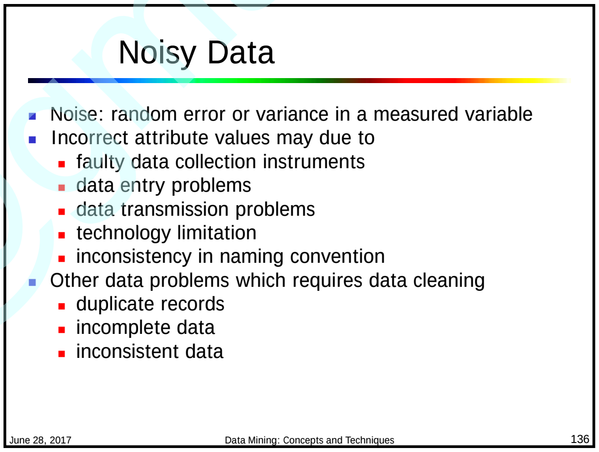
Missing Data

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



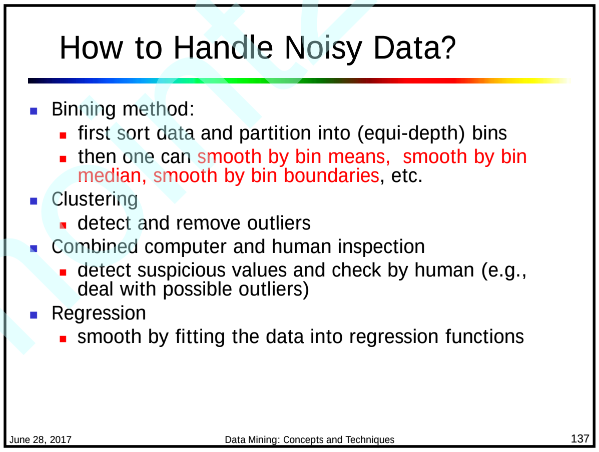
How to Handle Missing Data?

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



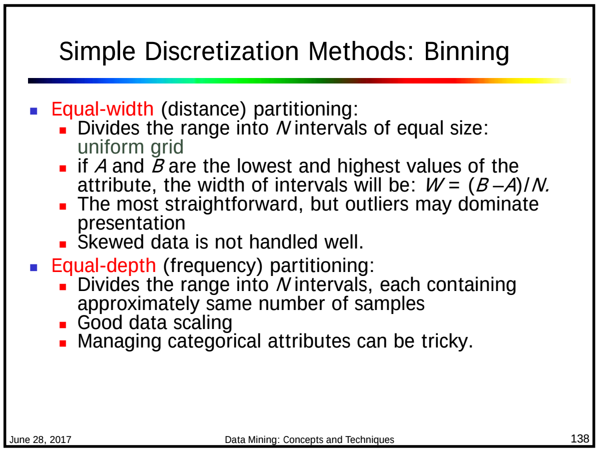
Noisy Data

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



How to Handle Noisy Data?

- Binning method:
  - first sort data and partition into (equi-depth) bins
  - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human (e.g., deal with possible outliers)
- Regression
  - smooth by fitting the data into regression functions



Simple Discretization Methods: Binning

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

Binning Methods for Data Smoothing

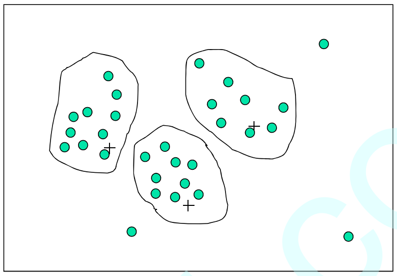
* Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

* Partition into (equi-depth) bins:
- Bin 1: 4, 8, 9, 15
- Bin 2: 21, 21, 24, 25
- Bin 3: 26, 28, 29, 34

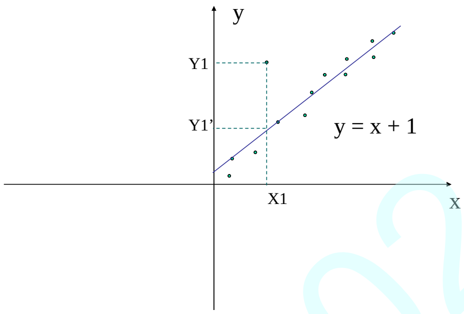
* Smoothing by bin means:
- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

* Smoothing by bin boundaries:
- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

Cluster Analysis



Regression



Chapter 3: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Data Integration

- Data integration:
  - combines data from multiple sources into a coherent store
- Schema integration
  - integrate metadata from different sources
  - Entity identification problem: identify real world entities from multiple data sources, e.g., A.cust-id ≡ B.cust-#
- Detecting and resolving data value conflicts
  - for the same real world entity, attribute values from different sources are different
  - possible reasons: different representations, different scales, e.g., metric vs. British units

Handling Redundancy in Data Integration

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

Data Transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
- Attribute/feature construction
  - New attributes constructed from the given ones

Data Transformation: Normalization

- min-max normalization

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

- z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- normalization by decimal scaling

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

Chapter 3: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary

Data Reduction Strategies

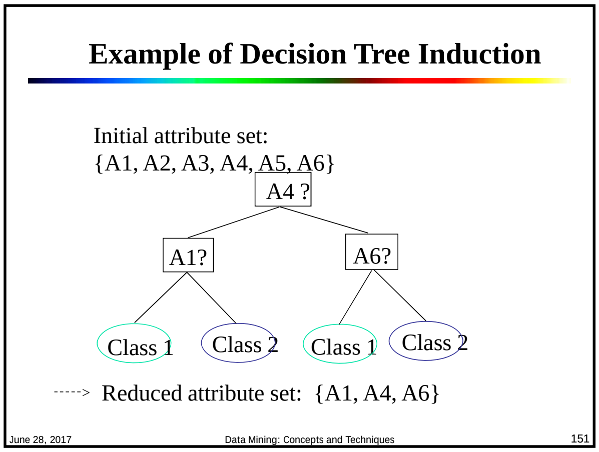
- A data warehouse may store terabytes of data
  - Complex data analysis/mining may take a very long time to run on the complete data set
- Data reduction
  - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results
- Data reduction strategies
  - Data cube aggregation
  - Dimensionality reduction—remove unimportant attributes
  - Data Compression
  - Numerosity reduction—fit data into models
  - Discretization and concept hierarchy generation

Data Cube Aggregation

- The lowest level of a data cube
  - the aggregated data for an individual entity of interest
  - e.g., a customer in a phone calling data warehouse.
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

Dimensionality Reduction

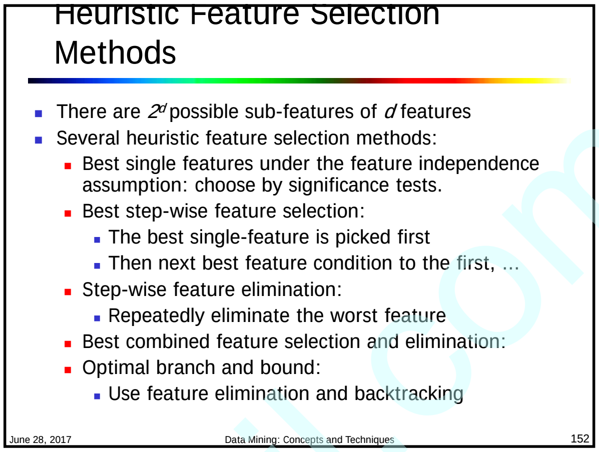
- Feature selection (i.e., attribute subset selection):
  - Select a minimum set of features such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features
  - reduce # of patterns in the patterns, easier to understand
- Heuristic methods (due to exponential # of choices):
  - step-wise forward selection
  - step-wise backward elimination
  - combining forward selection and backward elimination
  - decision-tree induction



Example of Decision Tree Induction

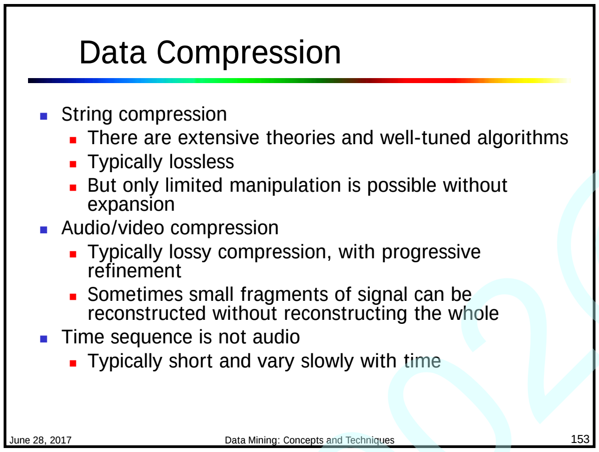
Initial attribute set:
{A1, A2, A3, A4, A5, A6}

-----> Reduced attribute set: {A1, A4, A6}



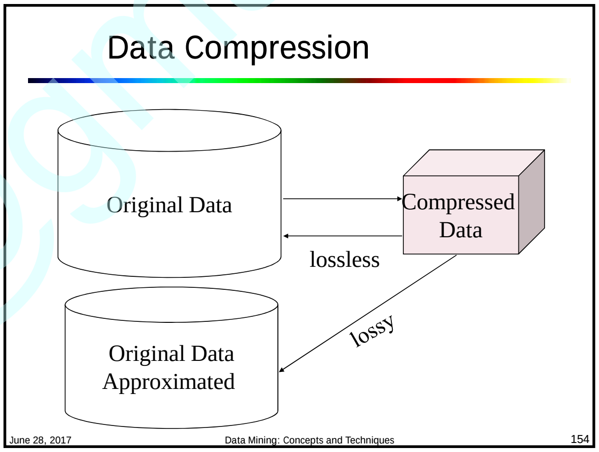
Heuristic Feature Selection Methods

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

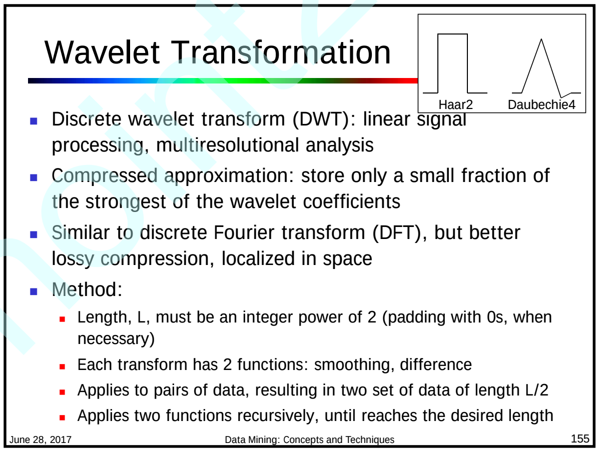


Data Compression

- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically lossless
  - But only limited manipulation is possible without expansion
- Audio/video compression
  - Typically lossy compression, with progressive refinement
  - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio
  - Typically short and vary slowly with time



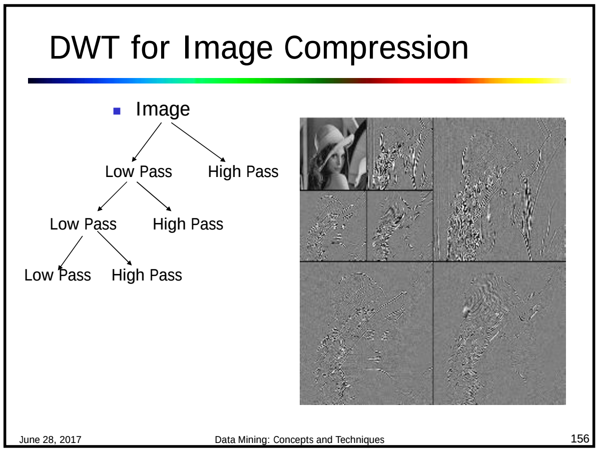
Data Compression



Wavelet Transformation

Haar2
Daubechie4

- Discrete wavelet transform (DWT): linear signal processing, multiresolutional analysis
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Similar to discrete Fourier transform (DFT), but better lossy compression, localized in space
- Method:
  - Length, L, must be an integer power of 2 (padding with 0s, when necessary)
  - Each transform has 2 functions: smoothing, difference
  - Applies to pairs of data, resulting in two set of data of length L/2
  - Applies two functions recursively, until reaches the desired length



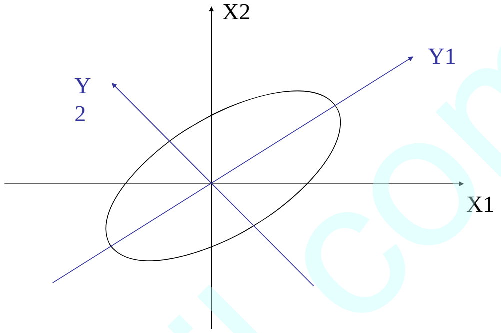
DWT for Image Compression

- Image

Principal Component Analysis

- Given *N* data vectors from *k*-dimensions, find $c <= k$ orthogonal vectors that can be best used to represent data
  - The original data set is reduced to one consisting of N data vectors on c principal components (reduced dimensions)
- Each data vector is a linear combination of the *c* principal component vectors
- Works for numeric data only
- Used when the number of dimensions is large

Principal Component Analysis



Numerosity Reduction

- Parametric methods
  - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
  - Log-linear models: obtain value at a point in m-D space as the product on appropriate marginal subspaces
- Non-parametric methods
  - Do not assume models
  - Major families: histograms, clustering, sampling

Regression and Log-Linear Models

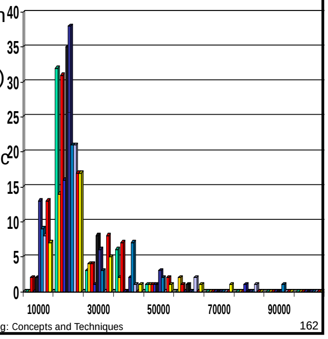
- Linear regression: Data are modeled to fit a straight line
  - Often uses the least-square method to fit the line
- Multiple regression: allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- Log-linear model: approximates discrete multidimensional probability distributions

Regress Analysis and Log-Linear Models

- Linear regression: $Y = \alpha + \beta X$
  - Two parameters , α and β specify the line and are to be estimated by using the data at hand.
  - using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots$.
- Multiple regression: $Y = b0 + b1\ X1 + b2\ X2$.
  - Many nonlinear functions can be transformed into the above.
- Log-linear models:
  - The multi-way table of joint probabilities is approximated by a product of lower-order tables.
  - Probability: $p(a, b, c, d) = \alpha ab\ \beta ac\chi ad\ \delta bcd$

Histograms

- A popular data reduction technique
- Divide data into buckets and store average (sum) for each bucket
- Can be constructed optimally in one dimension using dynamic programming
- Related to quantization problems.



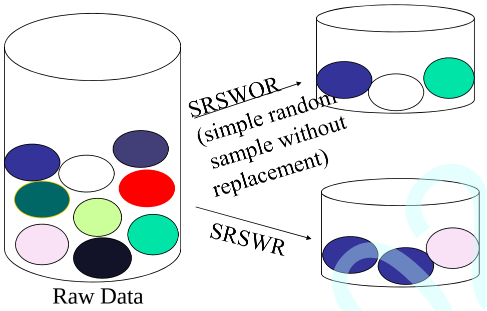
Clustering

- Partition data set into clusters, and one can store cluster representation only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms, further detailed in Chapter 8

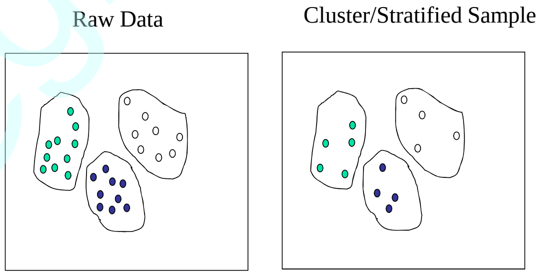
Sampling

- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
  - Stratified sampling:
    - Approximate the percentage of each class (or subpopulation of interest) in the overall database
    - Used in conjunction with skewed data
- Sampling may not reduce database I/Os (page at a time).

Sampling



Sampling

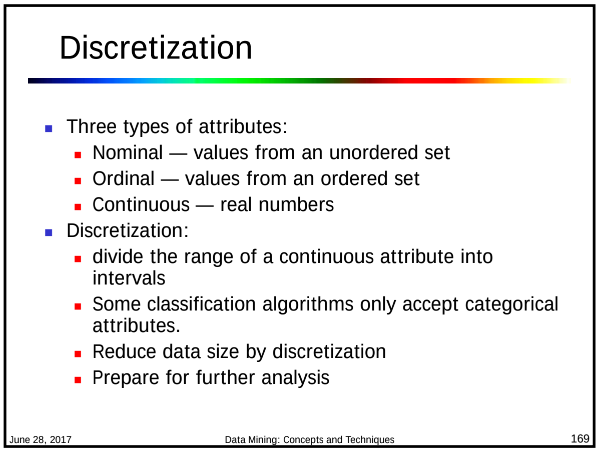


Hierarchical Reduction

- Use multi-resolution structure with different degrees of reduction
- Hierarchical clustering is often performed but tends to define partitions of data sets rather than “clusters”
- Parametric methods are usually not amenable to hierarchical representation
- Hierarchical aggregation
  - An index tree hierarchically divides a data set into partitions by value range of some attributes
  - Each partition can be considered as a bucket
  - Thus an index tree with aggregates stored at each node is a hierarchical histogram

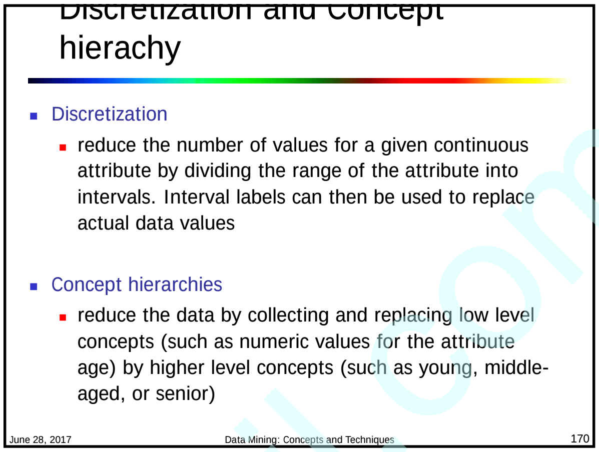
Chapter 3: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary



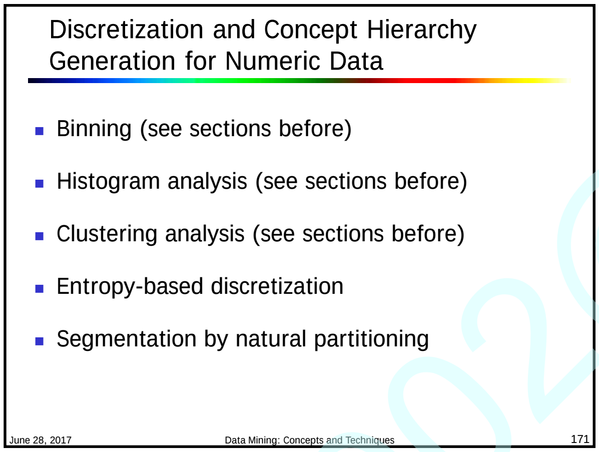
Discretization

- Three types of attributes:
  - Nominal — values from an unordered set
  - Ordinal — values from an ordered set
  - Continuous — real numbers
- Discretization:
  - divide the range of a continuous attribute into intervals
  - Some classification algorithms only accept categorical attributes.
  - Reduce data size by discretization
  - Prepare for further analysis



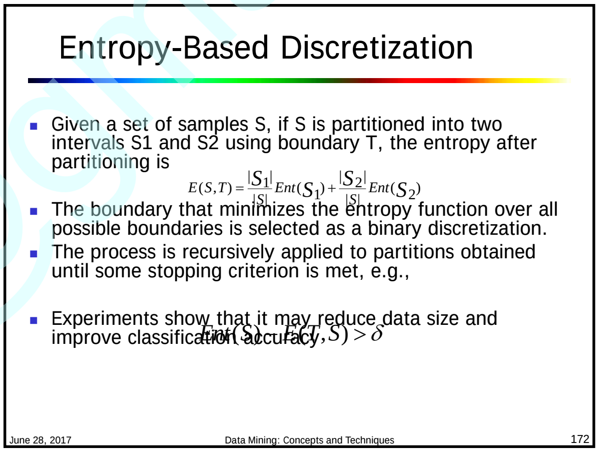
Discretization and Concept hierachy

- Discretization
  - reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values
- Concept hierarchies
  - reduce the data by collecting and replacing low level concepts (such as numeric values for the attribute age) by higher level concepts (such as young, middle-aged, or senior)



Discretization and Concept Hierarchy Generation for Numeric Data

- Binning (see sections before)
- Histogram analysis (see sections before)
- Clustering analysis (see sections before)
- Entropy-based discretization
- Segmentation by natural partitioning



Entropy-Based Discretization

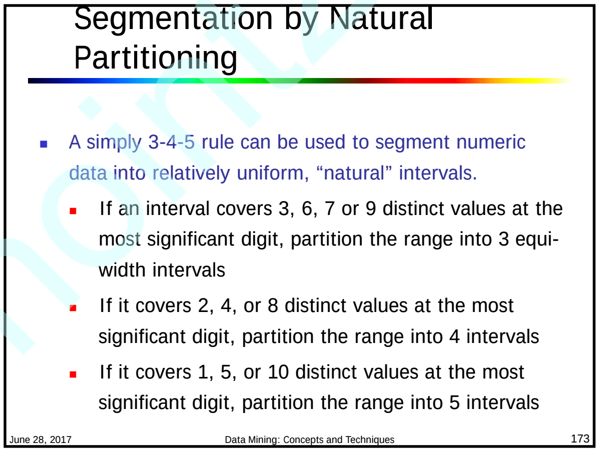
- Given a set of samples S, if S is partitioned into two intervals S1 and S2 using boundary T, the entropy after partitioning is

$$E(S,T) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$$

- The boundary that minimizes the entropy function over all possible boundaries is selected as a binary discretization.
- The process is recursively applied to partitions obtained until some stopping criterion is met, e.g.,

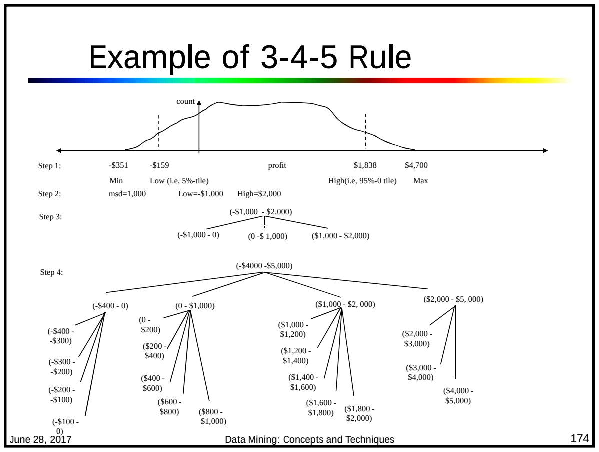
$$Ent(S) - E(T,S) > \delta$$

- Experiments show that it may reduce data size and improve classification accuracy

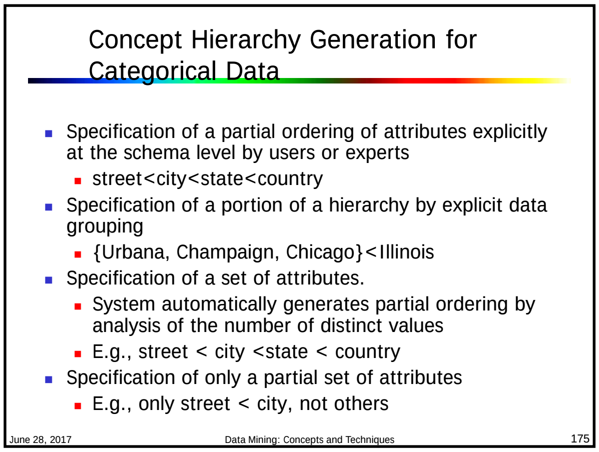


Segmentation by Natural Partitioning

- A simply 3-4-5 rule can be used to segment numeric data into relatively uniform, “natural” intervals.
  - If an interval covers 3, 6, 7 or 9 distinct values at the most significant digit, partition the range into 3 equi-width intervals
  - If it covers 2, 4, or 8 distinct values at the most significant digit, partition the range into 4 intervals
  - If it covers 1, 5, or 10 distinct values at the most significant digit, partition the range into 5 intervals

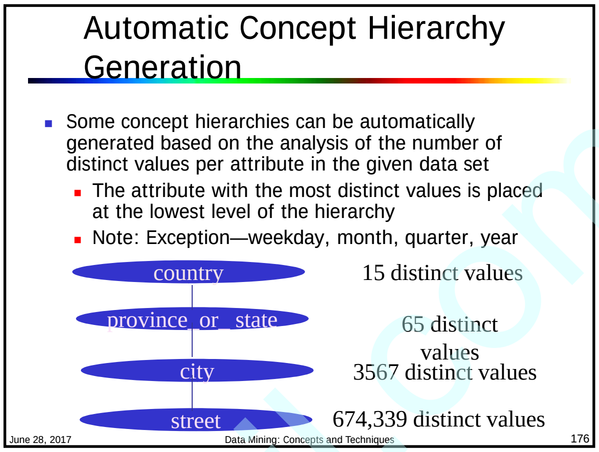


Example of 3-4-5 Rule



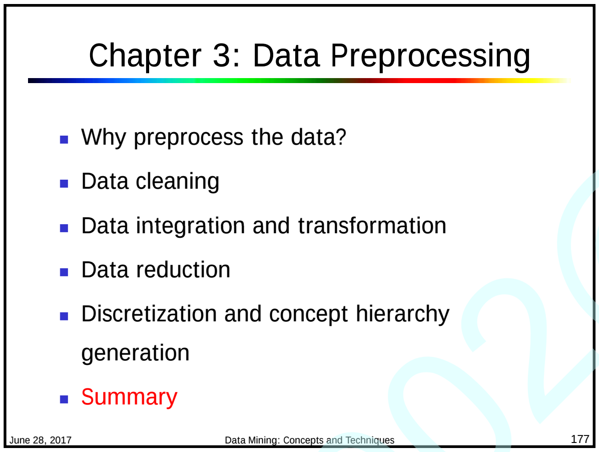
Concept Hierarchy Generation for Categorical Data

- Specification of a partial ordering of attributes explicitly at the schema level by users or experts
  - street<city<state<country
- Specification of a portion of a hierarchy by explicit data grouping
  - {Urbana, Champaign, Chicago}<Illinois
- Specification of a set of attributes.
  - System automatically generates partial ordering by analysis of the number of distinct values
  - E.g., street < city <state < country
- Specification of only a partial set of attributes
  - E.g., only street < city, not others



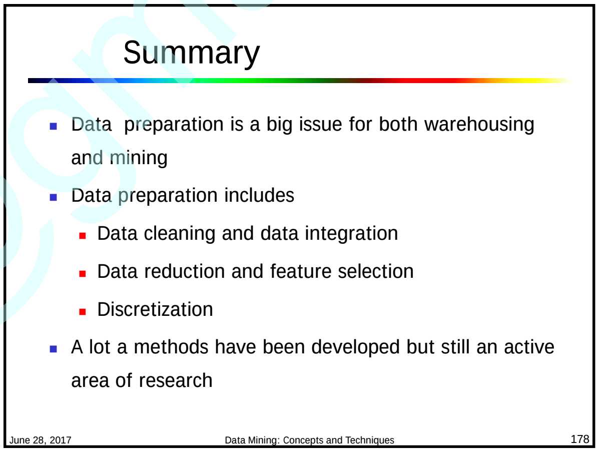
Automatic Concept Hierarchy Generation

- Some concept hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the given data set
  - The attribute with the most distinct values is placed at the lowest level of the hierarchy
  - Note: Exception—weekday, month, quarter, year



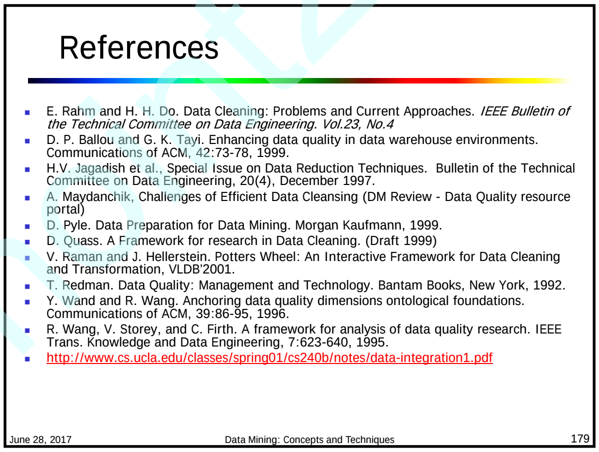
Chapter 3: Data Preprocessing

- Why preprocess the data?
- Data cleaning
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation
- Summary



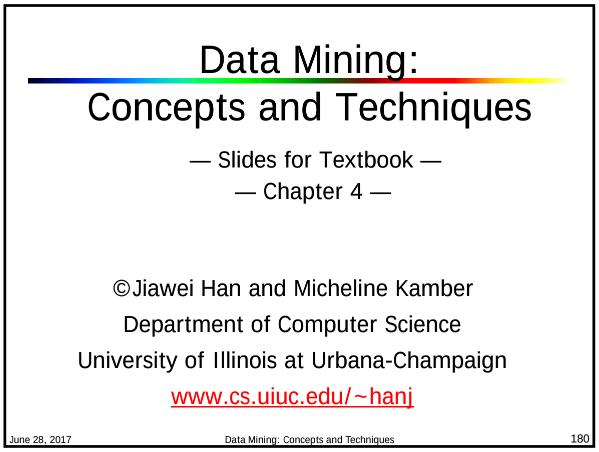
Summary

- Data preparation is a big issue for both warehousing and mining
- Data preparation includes
  - Data cleaning and data integration
  - Data reduction and feature selection
  - Discretization
- A lot a methods have been developed but still an active area of research



References

- E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. *IEEE Bulletin of the Technical Committee on Data Engineering. Vol.23, No.4*
- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. Communications of ACM, 42:73-78, 1999.
- H.V. Jagadish et al., Special Issue on Data Reduction Techniques. Bulletin of the Technical Committee on Data Engineering, 20(4), December 1997.
- A. Maydanchik, Challenges of Efficient Data Cleansing (DM Review - Data Quality resource portal)
- D. Pyle. Data Preparation for Data Mining. Morgan Kaufmann, 1999.
- D. Quass. A Framework for research in Data Cleaning. (Draft 1999)
- V. Raman and J. Hellerstein. Potters Wheel: An Interactive Framework for Data Cleaning and Transformation, VLDB'2001.
- T. Redman. Data Quality: Management and Technology. Bantam Books, New York, 1992.
- Y. Wand and R. Wang. Anchoring data quality dimensions ontological foundations. Communications of ACM, 39:86-95, 1996.
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. IEEE Trans. Knowledge and Data Engineering, 7:623-640, 1995.
- http://www.cs.ucla.edu/classes/spring01/cs240b/notes/data-integration1.pdf



Data Mining:
Concepts and Techniques

— Slides for Textbook —
— Chapter 4 —

©Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj

Chapter 4: Data Mining Primitives, Languages, and System Architectures

- Data mining primitives: What defines a data mining task?
- A data mining query language
- Design graphical user interfaces based on a data mining query language
- Architecture of data mining systems
- Summary

Why Data Mining Primitives and Languages?

- Finding all the patterns autonomously in a database? — unrealistic because the patterns could be too many but uninteresting
- Data mining should be an interactive process
  - User directs what to be mined
- Users must be provided with a set of primitives to be used to communicate with the data mining system
- Incorporating these primitives in a data mining query language
  - More flexible user interaction
  - Foundation for design of graphical user interface
  - Standardization of data mining industry and practice

What Defines a Data Mining Task ?

- Task-relevant data
- Type of knowledge to be mined
- Background knowledge
- Pattern interestingness measurements
- Visualization of discovered patterns

Task-Relevant Data (Minable View)

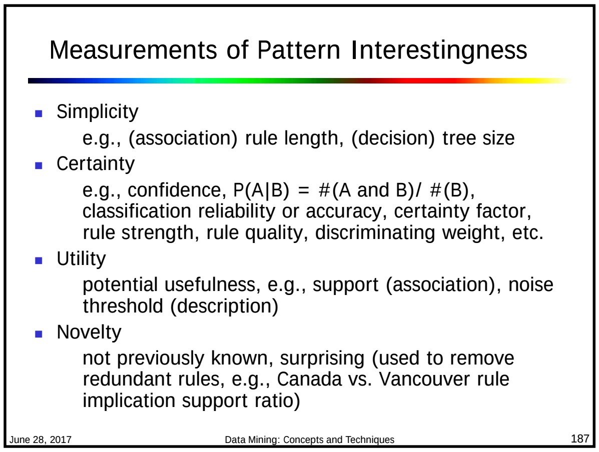
- Database or data warehouse name
- Database tables or data warehouse cubes
- Condition for data selection
- Relevant attributes or dimensions
- Data grouping criteria

Types of knowledge to be mined

- Characterization
- Discrimination
- Association
- Classification/prediction
- Clustering
- Outlier analysis
- Other data mining tasks

Background Knowledge: Concept Hierarchies

- Schema hierarchy
  - E.g., street < city < province_or_state < country
- Set-grouping hierarchy
  - E.g., {20-39} = young, {40-59} = middle_aged
- Operation-derived hierarchy
  - email address: dmbook@cs.sfu.ca
    login-name < department < university < country
- Rule-based hierarchy
  - low_profit_margin (X) <= price(X, P_1) and cost (X, P_2) and (P_1 - P_2) < $50



Measurements of Pattern Interestingness

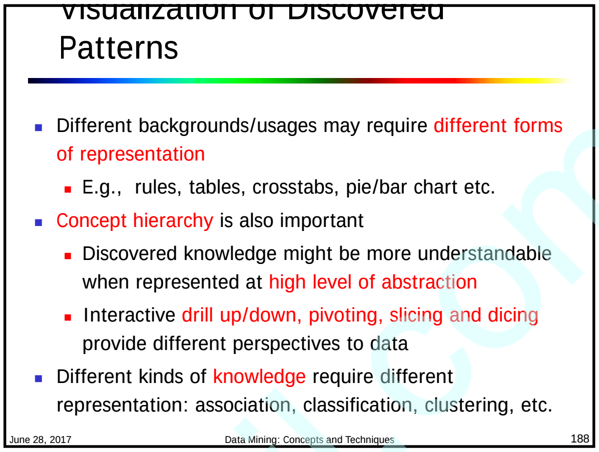
- Simplicity

  e.g., (association) rule length, (decision) tree size
- Certainty

  e.g., confidence, P(A|B) = #(A and B)/ #(B), classification reliability or accuracy, certainty factor, rule strength, rule quality, discriminating weight, etc.
- Utility

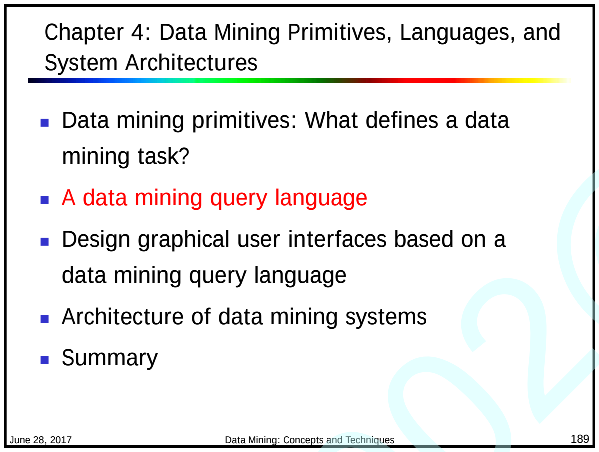
  potential usefulness, e.g., support (association), noise threshold (description)
- Novelty

  not previously known, surprising (used to remove redundant rules, e.g., Canada vs. Vancouver rule implication support ratio)



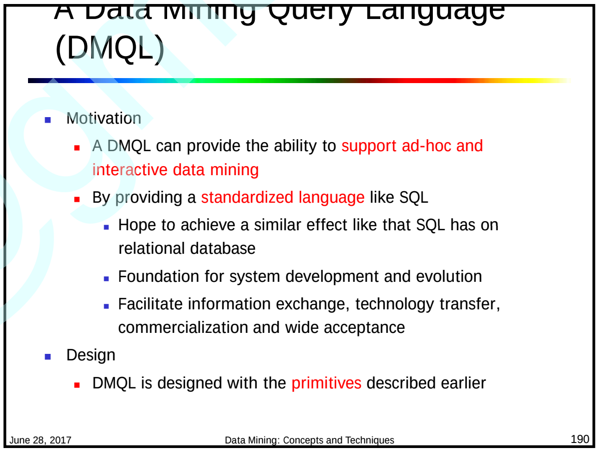
Visualization of Discovered Patterns

- Different backgrounds/usages may require different forms of representation
  - E.g., rules, tables, crosstabs, pie/bar chart etc.
- Concept hierarchy is also important
  - Discovered knowledge might be more understandable when represented at high level of abstraction
  - Interactive drill up/down, pivoting, slicing and dicing provide different perspectives to data
- Different kinds of knowledge require different representation: association, classification, clustering, etc.



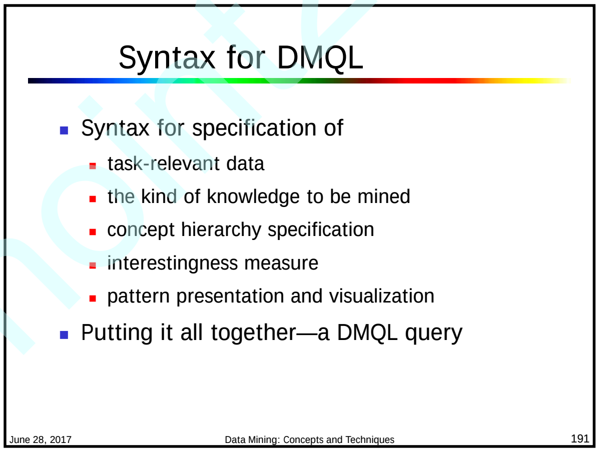
Chapter 4: Data Mining Primitives, Languages, and System Architectures

- Data mining primitives: What defines a data mining task?
- A data mining query language
- Design graphical user interfaces based on a data mining query language
- Architecture of data mining systems
- Summary



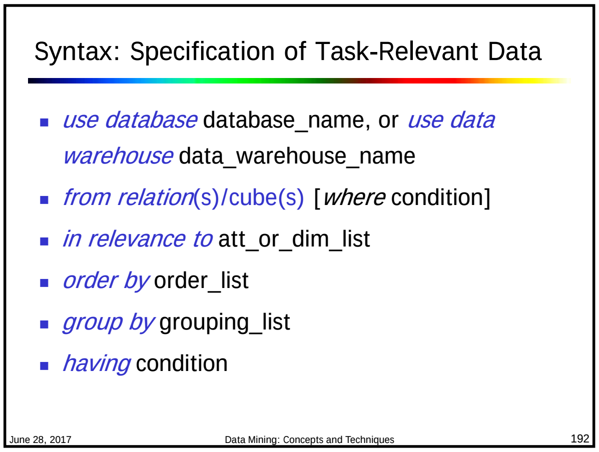
A Data Mining Query Language (DMQL)

- Motivation
  - A DMQL can provide the ability to support ad-hoc and interactive data mining
  - By providing a standardized language like SQL
    - Hope to achieve a similar effect like that SQL has on relational database
    - Foundation for system development and evolution
    - Facilitate information exchange, technology transfer, commercialization and wide acceptance
- Design
  - DMQL is designed with the primitives described earlier



Syntax for DMQL

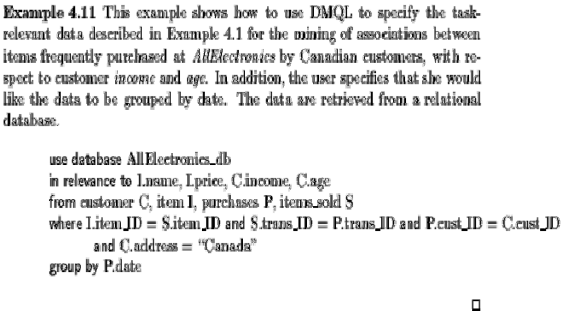
- Syntax for specification of
  - task-relevant data
  - the kind of knowledge to be mined
  - concept hierarchy specification
  - interestingness measure
  - pattern presentation and visualization
- Putting it all together—a DMQL query



Syntax: Specification of Task-Relevant Data

- *use database* database_name, or *use data warehouse* data_warehouse_name
- *from relation*(s)/cube(s) [*where* condition]
- *in relevance to* att_or_dim_list
- *order by* order_list
- *group by* grouping_list
- *having* condition

Specification of task-relevant data



Example 4.11 This example shows how to use DMQL to specify the task-relevant data described in Example 4.1 for the mining of associations between items frequently purchased at *AllElectronics* by Canadian customers, with respect to customer *income* and *age*. In addition, the user specifies that she would like the data to be grouped by date. The data are retrieved from a relational database.

```
use database AllElectronics_db
in relevance to I.name, I.price, C.income, C.age
from customer C, item I, purchases P, items_sold S
where I.item_ID = S.item_ID and S.trans_ID = P.trans_ID and P.cust_ID = C.cust_ID
      and C.address = "Canada"
group by P.date
```

□

Syntax: Kind of knowledge to Be Mined

- Characterization

  Mine_Knowledge_Specification ::=
  mine characteristics [*as* pattern_name]
  analyze measure(s)

- Discrimination

  Mine_Knowledge_Specification ::=
  mine comparison [*as* pattern_name]
  for target_class *where* target_condition
  {*versus* contrast_class_*i* *where* contrast_condition_*i*}
  analyze measure(s)

  E.g. mine comparison as purchaseGroups
  for bigSpenders where avg(I.price) >= $100
  versus budgetSpenders where avg(I.price) < $100
  analyze count

Syntax: Kind of Knowledge to Be Mined (cont.)

- Association

  Mine_Knowledge_Specification ::=
  mine associations [*as* pattern_name]
  [*matching* <metapattern>]

  E.g. mine associations as buyingHabits
  matching P(X:custom, W) ^ Q(X, Y)=>buys(X, Z)

- Classification

  Mine_Knowledge_Specification ::=
  mine classification [*as* pattern_name]
  analyze classifying_attribute_or_dimension

- Other Patterns

  clustering, outlier analysis, prediction ...

Syntax: Concept Hierarchy Specification

- To specify what concept hierarchies to use

  use hierarchy **<hierarchy>** for **<attribute_or_dimension>**

- We use different syntax to define different type of hierarchies
  - schema hierarchies

    define hierarchy **time_hierarchy** on **date** as **[date,month quarter,year]**

  - set-grouping hierarchies

    define hierarchy **age_hierarchy** for **age** on **customer** as

    level1: {*young, middle_aged, senior*} < level0: all

    level2: {20, ..., 39} < level1: *young*

    level2: {40, ..., 59} < level1: *middle_aged*

    level2: {60, ..., 89} < level1: *senior*

Concept Hierarchy Specification (Cont.)

- operation-derived hierarchies

  define hierarchy **age_hierarchy** for **age** on **customer** as

  {age_category(1), ..., age_category(5)} := cluster(default, age, 5) < all**(age)**

- rule-based hierarchies

  define hierarchy **profit_margin_hierarchy** on **item** as

  level_1: low_profit_margin < level_0: all

  if (price - cost)< $50

  level_1: medium-profit_margin < level_0: all

  if ((price - cost) > $50) and ((price - cost) <= $250))

  level_1: high_profit_margin < level_0: all

  if (price - cost) > $250

Specification of Interestingness Measures

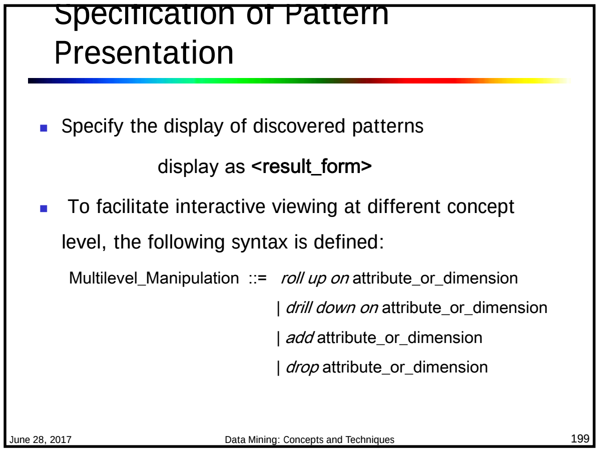
- Interestingness measures and thresholds can be specified by a user with the statement:

  with **<interest_measure_name>** threshold = **threshold_value**

- Example:

  with support threshold = 0.05

  with confidence threshold = 0.7



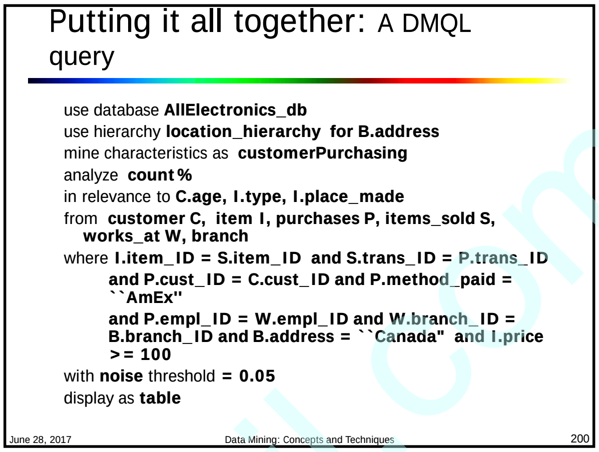
Specification of Pattern Presentation

- Specify the display of discovered patterns

  display as <result_form>

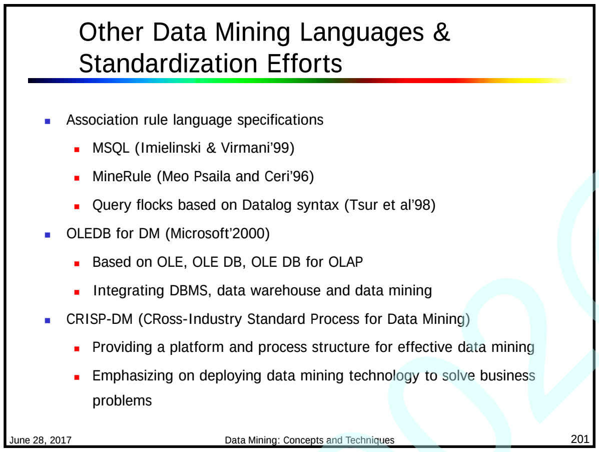
- To facilitate interactive viewing at different concept level, the following syntax is defined:

```
Multilevel_Manipulation  ::=   roll up on attribute_or_dimension
                             | drill down on attribute_or_dimension
                             | add attribute_or_dimension
                             | drop attribute_or_dimension
```



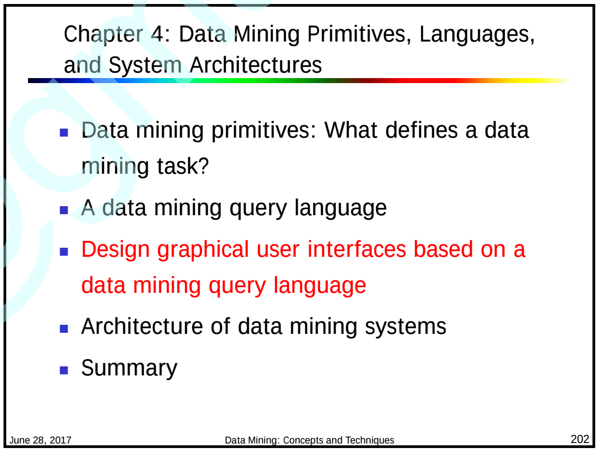
Putting it all together: A DMQL query

```
use database AllElectronics_db
use hierarchy location_hierarchy for B.address
mine characteristics as customerPurchasing
analyze count%
in relevance to C.age, I.type, I.place_made
from customer C, item I, purchases P, items_sold S,
   works_at W, branch
where I.item_ID = S.item_ID and S.trans_ID = P.trans_ID
       and P.cust_ID = C.cust_ID and P.method_paid =
       ``AmEx"
       and P.empl_ID = W.empl_ID and W.branch_ID =
       B.branch_ID and B.address = ``Canada" and I.price
       >= 100
with noise threshold = 0.05
display as table
```



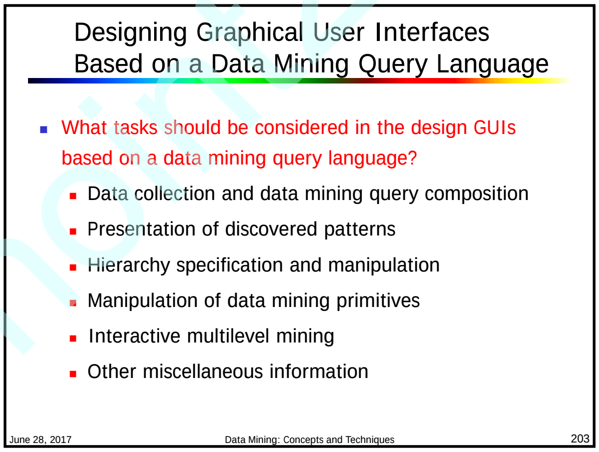
Other Data Mining Languages & Standardization Efforts

- Association rule language specifications
  - MSQL (Imielinski & Virmani'99)
  - MineRule (Meo Psaila and Ceri'96)
  - Query flocks based on Datalog syntax (Tsur et al'98)
- OLEDB for DM (Microsoft'2000)
  - Based on OLE, OLE DB, OLE DB for OLAP
  - Integrating DBMS, data warehouse and data mining
- CRISP-DM (CRoss-Industry Standard Process for Data Mining)
  - Providing a platform and process structure for effective data mining
  - Emphasizing on deploying data mining technology to solve business problems



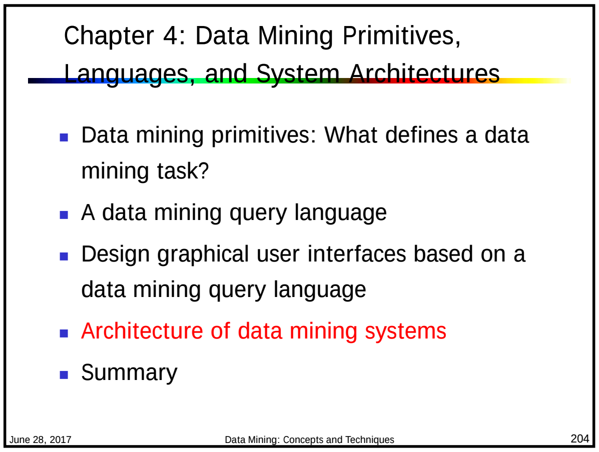
Chapter 4: Data Mining Primitives, Languages, and System Architectures

- Data mining primitives: What defines a data mining task?
- A data mining query language
- Design graphical user interfaces based on a data mining query language
- Architecture of data mining systems
- Summary



Designing Graphical User Interfaces Based on a Data Mining Query Language

- What tasks should be considered in the design GUIs based on a data mining query language?
  - Data collection and data mining query composition
  - Presentation of discovered patterns
  - Hierarchy specification and manipulation
  - Manipulation of data mining primitives
  - Interactive multilevel mining
  - Other miscellaneous information



Chapter 4: Data Mining Primitives, Languages, and System Architectures

- Data mining primitives: What defines a data mining task?
- A data mining query language
- Design graphical user interfaces based on a data mining query language
- Architecture of data mining systems
- Summary

Data Mining System Architectures

- Coupling data mining system with DB/DW system
  - No coupling—flat file processing, not recommended
  - Loose coupling
    - Fetching data from DB/DW
  - Semi-tight coupling—enhanced DM performance
    - Provide efficient implement a few data mining primitives in a DB/DW system, e.g., sorting, indexing, aggregation, histogram analysis, multiway join, precomputation of some stat functions
  - Tight coupling—A uniform information processing environment
    - DM is smoothly integrated into a DB/DW system, mining query is optimized based on mining query, indexing, query processing methods, etc.

Chapter 4: Data Mining Primitives, Languages, and System Architectures

- Data mining primitives: What defines a data mining task?
- A data mining query language
- Design graphical user interfaces based on a data mining query language
- Architecture of data mining systems
- Summary

Summary

- Five primitives for specification of a data mining task
  - task-relevant data
  - kind of knowledge to be mined
  - background knowledge
  - interestingness measures
  - knowledge presentation and visualization techniques to be used for displaying the discovered patterns
- Data mining query languages
  - DMQL, MS/OLEDB for DM, etc.
- Data mining system architecture
  - No coupling, loose coupling, semi-tight coupling, tight coupling

References

- E. Baralis and G. Psaila. Designing templates for mining association rules. Journal of Intelligent Information Systems, 9:7-32, 1997.
- Microsoft Corp., OLEDB for Data Mining, version 1.0, http://www.microsoft.com/data/oledb/dm, Aug. 2000.
- J. Han, Y. Fu, W. Wang, K. Koperski, and O. R. Zaiane, “DMQL: A Data Mining Query Language for Relational Databases”, DMKD'96, Montreal, Canada, June 1996.
- T. Imielinski and A. Virmani. MSQL: A query language for database mining. Data Mining and Knowledge Discovery, 3:373-408, 1999.
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM’94, Gaithersburg, Maryland, Nov. 1994.
- R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. VLDB'96, pages 122-133, Bombay, India, Sept. 1996.
- A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. IEEE Trans. on Knowledge and Data Engineering, 8:970-974, Dec. 1996.
- S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98, Seattle, Washington, June 1998.
- D. Tsur, J. D. Ullman, S. Abitboul, C. Clifton, R. Motwani, and S. Nestorov. Query flocks: A generalization of association-rule mining. SIGMOD'98, Seattle, Washington, June 1998.

Data Mining: Concepts and Techniques

— Slides for Textbook —
— Chapter 5 —

©Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.sfu.ca, www.cs.uiuc.edu/~hanj

Chapter 5: Concept Description: Characterization and Comparison

- What is concept description?
- Data generalization and summarization-based characterization
- Analytical characterization: Analysis of attribute relevance
- Mining class comparisons: Discriminating between different classes
- Mining descriptive statistical measures in large databases
- Discussion
- Summary

What is Concept Description?

- Descriptive vs. predictive data mining
  - Descriptive mining: describes concepts or task-relevant data sets in concise, summarative, informative, discriminative forms
  - Predictive mining: Based on data and analysis, constructs models for the database, and predicts the trend and properties of unknown data
- Concept description:
  - Characterization: provides a concise and succinct summarization of the given collection of data
  - Comparison: provides descriptions comparing two or more collections of data

Concept Description vs. OLAP

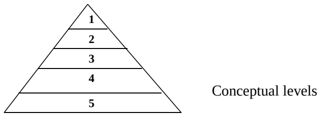
- Concept description:
  - can handle complex data types of the attributes and their aggregations
  - a more automated process
- OLAP:
  - restricted to a small number of dimension and measure types
  - user-controlled process

Chapter 5: Concept Description: Characterization and Comparison

- What is concept description?
- Data generalization and summarization-based characterization
- Analytical characterization: Analysis of attribute relevance
- Mining class comparisons: Discriminating between different classes
- Mining descriptive statistical measures in large databases
- Discussion
- Summary

Data Generalization and Summarization-based Characterization

- Data generalization
  - A process which abstracts a large set of task-relevant data in a database from a low conceptual levels to higher ones.



  - Approaches:
    - Data cube approach(OLAP approach)
    - Attribute-oriented induction approach

Characterization: Data Cube Approach

- Data are stored in *data cube*
- Identify expensive computations
  - e.g., count( ), sum( ), average( ), max( )
- Perform computations and store results in data cubes
- Generalization and specialization can be performed on a data cube by *roll-up* and *drill-down*
- An efficient implementation of data generalization

Data Cube Approach (Cont…)

- Limitations
  - can only handle data types of dimensions to *simple nonnumeric data* and of measures to *simple aggregated numeric values*.
  - Lack of intelligent analysis, can't tell which dimensions should be used and what levels should the generalization reach

Attribute-Oriented Induction

- Proposed in 1989 (KDD '89 workshop)
- Not confined to categorical data nor particular measures.
- How it is done?
  - Collect the task-relevant data (*initial relation*) using a relational database query
  - Perform generalization by attribute removal or attribute generalization.
  - Apply aggregation by merging identical, generalized tuples and accumulating their respective counts
  - Interactive presentation with users

Basic Principles of Attribute-Oriented Induction

- Data focusing: task-relevant data, including dimensions, and the result is the *initial relation*.
- Attribute-removal: remove attribute *A* if there is a large set of distinct values for *A* but (1) there is no generalization operator on *A*, or (2) *A*'s higher level concepts are expressed in terms of other attributes.
- Attribute-generalization: If there is a large set of distinct values for *A*, and there exists a set of generalization operators on *A*, then select an operator and generalize *A*.
- Attribute-threshold control: typical 2-8, specified/default.
- Generalized relation threshold control: control the final relation/rule size. see example

Attribute-Oriented Induction: Basic Algorithm

- InitialRel: Query processing of task-relevant data, deriving the *initial relation*.
- PreGen: Based on the analysis of the number of distinct values in each attribute, determine generalization plan for each attribute: removal? or how high to generalize?
- PrimeGen: Based on the PreGen plan, perform generalization to the right level to derive a "prime generalized relation", accumulating the counts.
- Presentation: User interaction: (1) adjust levels by drilling, (2) pivoting, (3) mapping into rules, cross tabs, visualization presentations.

Example

- DMQL: Describe general characteristics of graduate students in the Big-University database

  use Big_University_DB
  mine characteristics as "Science_Students"
  in relevance to name, gender, major, birth_place, birth_date, residence, phone#, gpa
  from student
  where status in "graduate"

- Corresponding SQL statement:

  Select name, gender, major, birth_place, birth_date, residence, phone#, gpa
  from student
  where status in {"Msc", "MBA", "PhD" }

Class Characterization: An Example

Initial Relation

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone #	GPA
Jim Woodman	M	CS	Vancouver,BC, Canada	8-12-76	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-70	125 Austin Ave., Burnaby	420-5232	3.83
…	…	…	…	…	…	…	…
Removed	Retained	Sci,Eng, Bus	Country	Age range	City	Removed	Excl, VG,..

Prime Generalized Relation

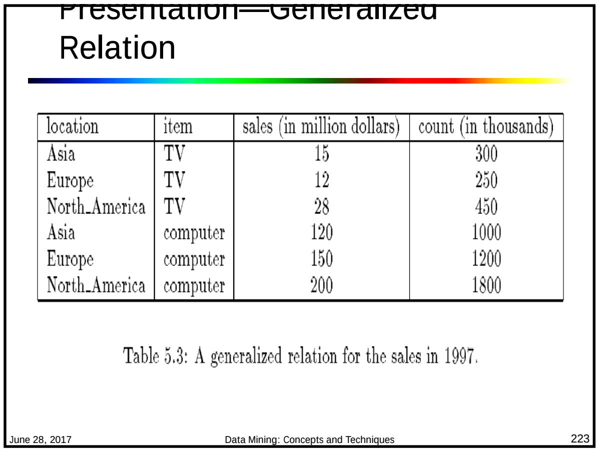
Gender	Major	Birth_region	Age_range	Residence	GPA	Count
M	Science	Canada	20-25	Richmond	Very-good	16
F	Science	Foreign	25-30	Burnaby	Excellent	22
…	…	…	…	…	…	…

Gender \ Birth_Region	Canada	Foreign	Total
M	16	14	30
F	10	22	32
Total	26	36	62

Presentation of Generalized Results

- Generalized relation:
  - Relations where some or all attributes are generalized, with counts or other aggregation values accumulated.
- Cross tabulation:
  - Mapping results into cross tabulation form (similar to contingency tables).
  - Visualization techniques:
  - Pie charts, bar charts, curves, cubes, and other visual forms.
- Quantitative characteristic rules:
  - Mapping generalized result into characteristic rules with quantitative information associated with it, e.g.,

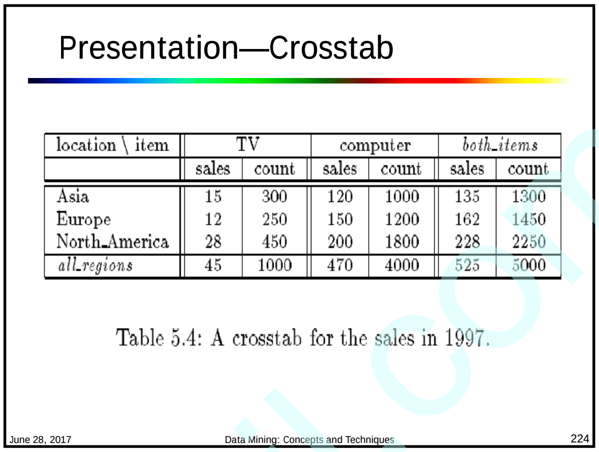
$$grad(x) \land male(x) \Rightarrow$$
$$birth\_region(x) = "Canada"[t:53\%] \lor birth\_region(x) = "foreign"[t:47\%].$$



Presentation—Generalized Relation

location	item	sales (in million dollars)	count (in thousands)
Asia	TV	15	300
Europe	TV	12	250
North_America	TV	28	450
Asia	computer	120	1000
Europe	computer	150	1200
North_America	computer	200	1800

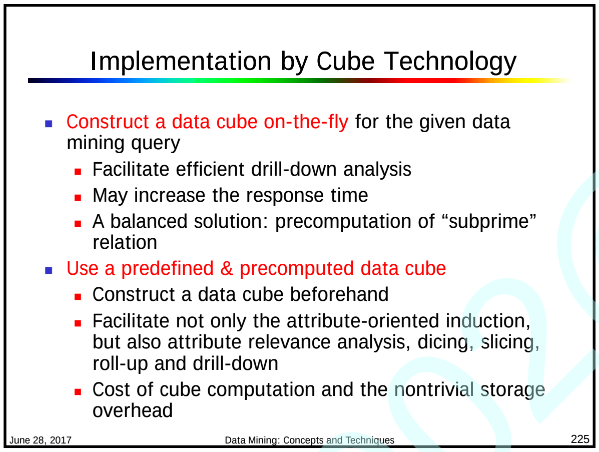
Table 5.3: A generalized relation for the sales in 1997.



Presentation—Crosstab

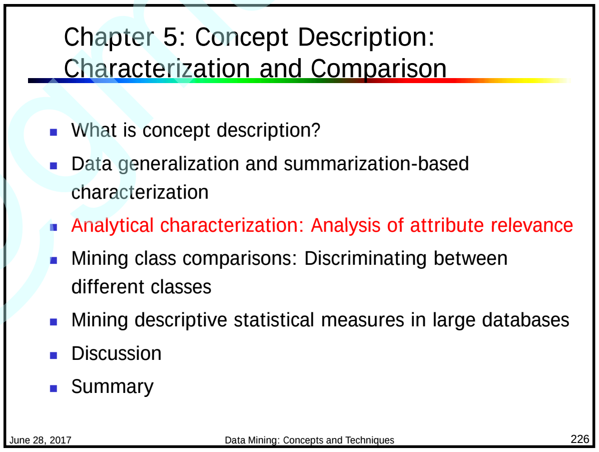
location \ item	TV		computer		*both_items*	
	sales	count	sales	count	sales	count
Asia	15	300	120	1000	135	1300
Europe	12	250	150	1200	162	1450
North_America	28	450	200	1800	228	2250
all_regions	45	1000	470	4000	525	5000

Table 5.4: A crosstab for the sales in 1997.



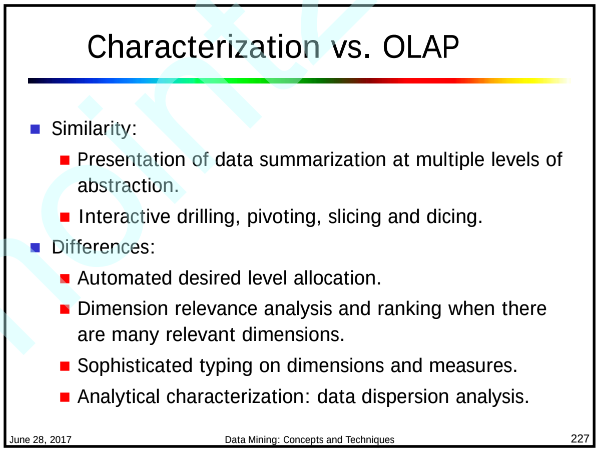
Implementation by Cube Technology

- Construct a data cube on-the-fly for the given data mining query
  - Facilitate efficient drill-down analysis
  - May increase the response time
  - A balanced solution: precomputation of "subprime" relation
- Use a predefined & precomputed data cube
  - Construct a data cube beforehand
  - Facilitate not only the attribute-oriented induction, but also attribute relevance analysis, dicing, slicing, roll-up and drill-down
  - Cost of cube computation and the nontrivial storage overhead



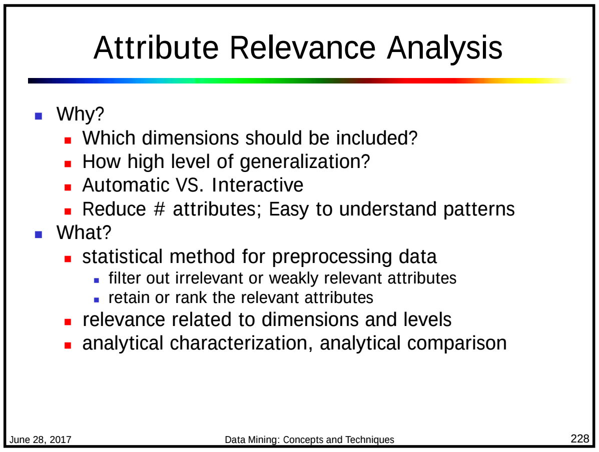
Chapter 5: Concept Description: Characterization and Comparison

- What is concept description?
- Data generalization and summarization-based characterization
- Analytical characterization: Analysis of attribute relevance
- Mining class comparisons: Discriminating between different classes
- Mining descriptive statistical measures in large databases
- Discussion
- Summary



Characterization vs. OLAP

- Similarity:
  - Presentation of data summarization at multiple levels of abstraction.
  - Interactive drilling, pivoting, slicing and dicing.
- Differences:
  - Automated desired level allocation.
  - Dimension relevance analysis and ranking when there are many relevant dimensions.
  - Sophisticated typing on dimensions and measures.
  - Analytical characterization: data dispersion analysis.



Attribute Relevance Analysis

- Why?
  - Which dimensions should be included?
  - How high level of generalization?
  - Automatic VS. Interactive
  - Reduce # attributes; Easy to understand patterns
- What?
  - statistical method for preprocessing data
    - filter out irrelevant or weakly relevant attributes
    - retain or rank the relevant attributes
  - relevance related to dimensions and levels
  - analytical characterization, analytical comparison

Attribute relevance analysis (cont'd)

- How?
  - Data Collection
  - Analytical Generalization
    - Use information gain analysis (e.g., entropy or other measures) to identify highly relevant dimensions and levels.
  - Relevance Analysis
    - Sort and select the most relevant dimensions and levels.
  - Attribute-oriented Induction for class description
    - On selected dimension/level
  - OLAP operations (e.g. drilling, slicing) on relevance rules

Relevance Measures

- Quantitative relevance measure determines the classifying power of an attribute within a set of data.
- Methods
  - information gain (ID3)
  - gain ratio (C4.5)
  - gini index
  - χ^2 contingency table statistics
  - uncertainty coefficient

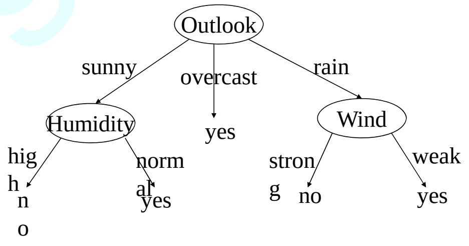
Information-Theoretic Approach

- Decision tree
  - each internal node tests an attribute
  - each branch corresponds to attribute value
  - each leaf node assigns a classification
- ID3 algorithm
  - build decision tree based on training objects with known class labels to classify testing objects
  - rank attributes with information gain measure
  - minimal height
    - the least number of tests to classify an object

Top-Down Induction of Decision Tree

Attributes = {Outlook, Temperature, Humidity, Wind}

PlayTennis = {yes, no}



Entropy and Information Gain

- S contains s_i tuples of class C_i for i = {1, ..., m}
- Information measures info required to classify any arbitrary tuple

$$I(s_1, s_2, \ldots, s_m) = \sum_{i=1}^{m} \frac{s_i}{s} \log \frac{s_i}{s}$$

- Entropy of attribute A with values $\{a_1, a_2, \ldots, a_v\}$

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_{1j}, \ldots, s_{mj})$$

- Information gained by branching on attribute A

$$Gain(A) = I(s_1, s_2, \ldots, s_m) - E(A)$$

Example: Analytical Characterization

- Task
  - Mine general characteristics describing graduate students using analytical characterization
- Given
  - attributes *name, gender, major, birth_place, birth_date, phone#*, and *gpa*
  - *Gen(a_i)* = concept hierarchies on a_i
  - U_i = attribute analytical thresholds for a_i
  - T_i = attribute generalization thresholds for a_i
  - *R* = attribute relevance threshold

Example: Analytical Characterization (cont'd)

- 1. Data collection
  - target class: graduate student
  - contrasting class: undergraduate student
- 2. Analytical generalization using U_i
  - attribute removal
    - remove *name* and *phone#*
  - attribute generalization
    - generalize *major*, *birth_place*, *birth_date* and *gpa*
    - accumulate counts
  - candidate relation: *gender*, *major*, *birth_country*, *age_range* and *gpa*

Example: Analytical characterization (2)

gender	major	birth_country	age_range	gpa	count
M	Science	Canada	20-25	Very_good	16
F	Science	Foreign	25-30	Excellent	22
M	Engineering	Foreign	25-30	Excellent	18
F	Science	Foreign	25-30	Excellent	25
M	Science	Canada	20-25	Excellent	21
F	Engineering	Canada	20-25	Excellent	18

Candidate relation for Target class: Graduate students (Σ=120)

gender	major	birth_country	age_range	gpa	count
M	Science	Foreign	<20	Very_good	18
F	Business	Canada	<20	Fair	20
M	Business	Canada	<20	Fair	22
F	Science	Canada	20-25	Fair	24
M	Engineering	Foreign	20-25	Very_good	22
F	Engineering	Canada	<20	Excellent	24

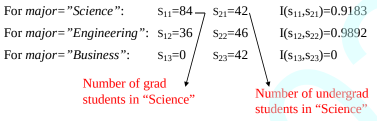
Candidate relation for Contrasting class: Undergraduate students (Σ=130)

Example: Analytical characterization (3)

- 3. Relevance analysis
  - Calculate expected info required to classify an arbitrary tuple

$$I(s_1,s_2)=I(120,130)=-\frac{120}{250}\log_2\frac{120}{250}-\frac{130}{250}\log_2\frac{130}{250}=0.9988$$

  - Calculate entropy of each attribute: e.g. *major*



For *major*="*Science*": s_{11}=84 s_{21}=42 $I(s_{11},s_{21})$=0.9183

For *major*="*Engineering*": s_{12}=36 s_{22}=46 $I(s_{12},s_{22})$=0.9892

For *major*="*Business*": s_{13}=0 s_{23}=42 $I(s_{13},s_{23})$=0

Number of grad students in "Science"

Number of undergrad students in "Science"

Example: Analytical Characterization (4)

- Calculate expected info required to classify a given sample if S is partitioned according to the attribute

$$E(major)=\frac{126}{250}I(s_{11},s_{21})+\frac{82}{250}I(s_{12},s_{22})+\frac{42}{250}I(s_{13},s_{23})=0.7873$$

- Calculate information gain for each attribute

$$Gain(major)=I(s_1,s_2)-E(major)=0.2115$$

  - Information gain for all attributes

Gain(gender)	= 0.0003
Gain(birth_country)	= 0.0407
Gain(major)	= 0.2115
Gain(gpa)	= 0.4490
Gain(age_range)	= 0.5971

Example: Analytical characterization (5)

- 4. Initial working relation (W_0) derivation
  - R = 0.1
  - remove irrelevant/weakly relevant attributes from candidate relation => drop *gender*, *birth_country*
  - remove contrasting class candidate relation

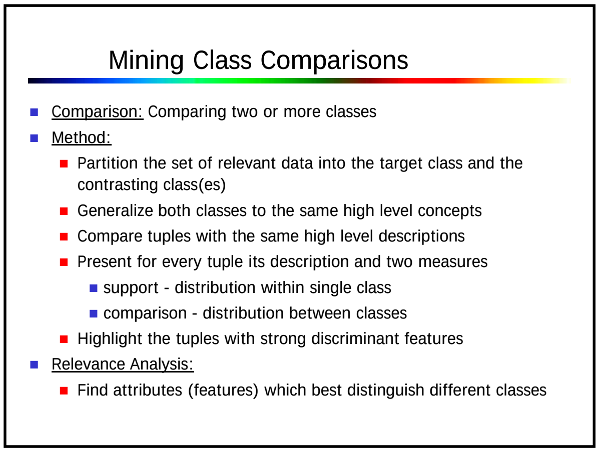
major	age_range	gpa	count
Science	20-25	Very_good	16
Science	25-30	Excellent	47
Science	20-25	Excellent	21
Engineering	20-25	Excellent	18
Engineering	25-30	Excellent	18

Initial target class working relation W_0: Graduate students

- 5. Perform attribute-oriented induction on W_0 using T_i

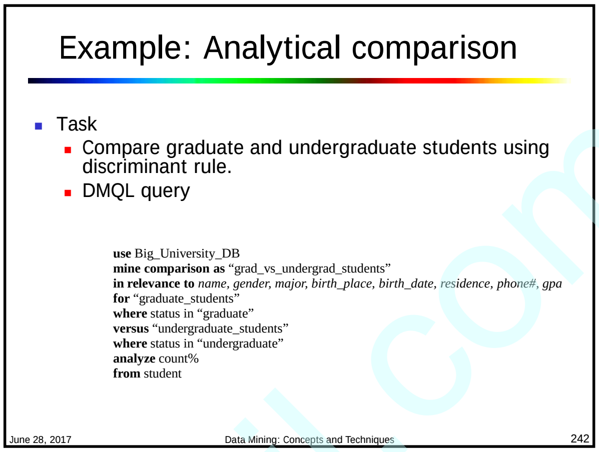
Chapter 5: Concept Description: Characterization and Comparison

- What is concept description?
- Data generalization and summarization-based characterization
- Analytical characterization: Analysis of attribute relevance
- Mining class comparisons: Discriminating between different classes
- Mining descriptive statistical measures in large databases
- Discussion
- Summary



Mining Class Comparisons

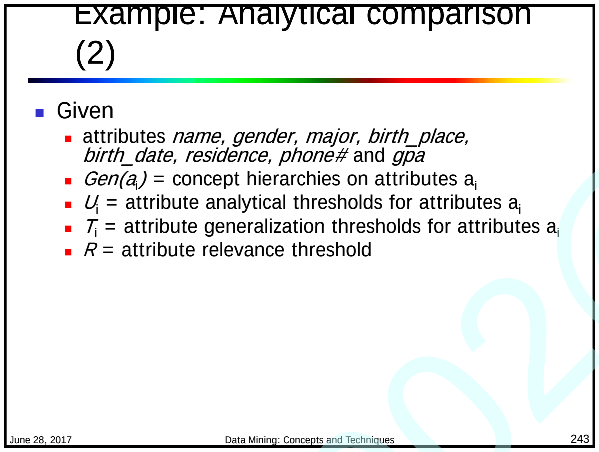
- Comparison: Comparing two or more classes
- Method:
  - Partition the set of relevant data into the target class and the contrasting class(es)
  - Generalize both classes to the same high level concepts
  - Compare tuples with the same high level descriptions
  - Present for every tuple its description and two measures
    - support - distribution within single class
    - comparison - distribution between classes
  - Highlight the tuples with strong discriminant features
- Relevance Analysis:
  - Find attributes (features) which best distinguish different classes



Example: Analytical comparison

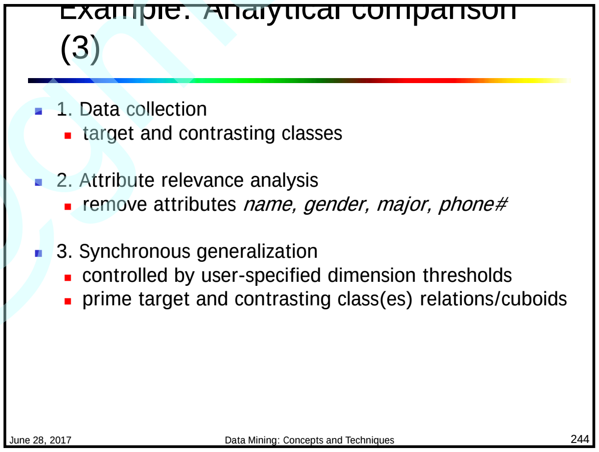
- Task
  - Compare graduate and undergraduate students using discriminant rule.
  - DMQL query

```
use Big_University_DB
mine comparison as "grad_vs_undergrad_students"
in relevance to name, gender, major, birth_place, birth_date, residence, phone#, gpa
for "graduate_students"
where status in "graduate"
versus "undergraduate_students"
where status in "undergraduate"
analyze count%
from student
```



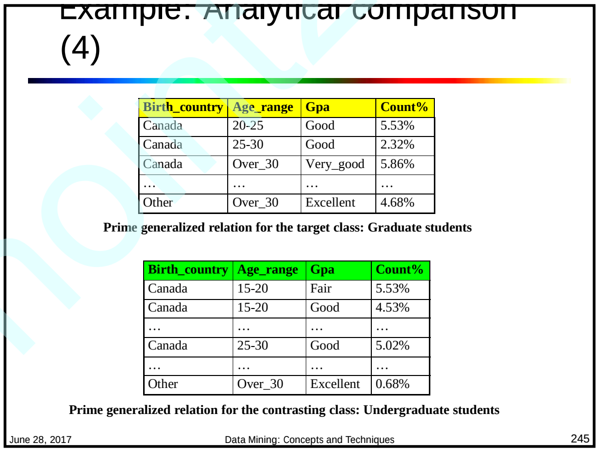
Example: Analytical comparison (2)

- Given
  - attributes *name, gender, major, birth_place, birth_date, residence, phone#* and *gpa*
  - *Gen(a_i)* = concept hierarchies on attributes a_i
  - U_i = attribute analytical thresholds for attributes a_i
  - T_i = attribute generalization thresholds for attributes a_i
  - *R* = attribute relevance threshold



Example: Analytical comparison (3)

- 1. Data collection
  - target and contrasting classes
- 2. Attribute relevance analysis
  - remove attributes *name, gender, major, phone#*
- 3. Synchronous generalization
  - controlled by user-specified dimension thresholds
  - prime target and contrasting class(es) relations/cuboids



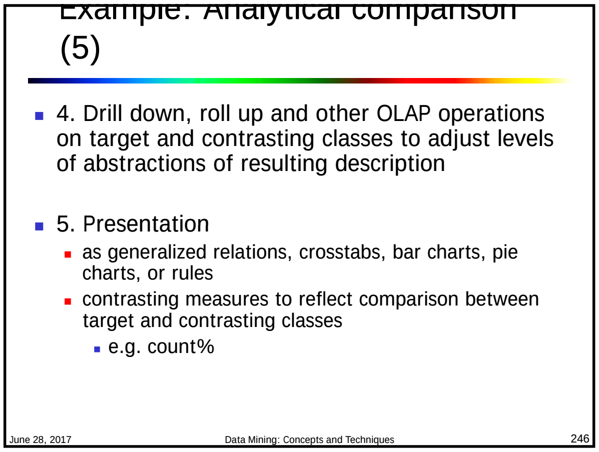
Example: Analytical comparison (4)

Birth_country	Age_range	Gpa	Count%
Canada	20-25	Good	5.53%
Canada	25-30	Good	2.32%
Canada	Over_30	Very_good	5.86%
…	…	…	…
Other	Over_30	Excellent	4.68%

Prime generalized relation for the target class: Graduate students

Birth_country	Age_range	Gpa	Count%
Canada	15-20	Fair	5.53%
Canada	15-20	Good	4.53%
…	…	…	…
Canada	25-30	Good	5.02%
…	…	…	…
Other	Over_30	Excellent	0.68%

Prime generalized relation for the contrasting class: Undergraduate students



Example: Analytical comparison (5)

- 4. Drill down, roll up and other OLAP operations on target and contrasting classes to adjust levels of abstractions of resulting description
- 5. Presentation
  - as generalized relations, crosstabs, bar charts, pie charts, or rules
  - contrasting measures to reflect comparison between target and contrasting classes
    - e.g. count%

Quantitative Discriminant Rules

- Cj = target class
- q_a = a generalized tuple covers some tuples of class
  - but can also cover some tuples of contrasting class
- d-weight
  - range: [0, 1]

$$d-weight = \frac{count(q_a \in C_j)}{\sum_{i=1}^{m} count(q_a \in C_i)}$$

- quantitative discriminant rule form

$$\forall X,\ target\_class(X) \Leftarrow condition(X)\ \ [d : d\_weight]$$

Example: Quantitative Discriminant Rule

Status	Birth_country	Age_range	Gpa	Count
Graduate	Canada	25-30	Good	90
Undergraduate	Canada	25-30	Good	210

Count distribution between graduate and undergraduate students for a generalized tuple

- Quantitative discriminant rule

$$\forall X,\ graduate\_student(X) \Leftarrow$$
$$birth\_country(X) = "Canada" \wedge age\_range(X) = "25-30" \wedge gpa(X) = "good"\ [d : 30\%]$$

  - where 90/(90+210) = 30%

Class Description

- Quantitative characteristic rule

$$\forall X,\ target\_class(X) \Rightarrow condition(X)\ \ [t : t\_weight]$$

  - necessary
- Quantitative discriminant rule

$$\forall X,\ target\_class(X) \Leftarrow condition(X)\ \ [d : d\_weight]$$

  - sufficient
- Quantitative description rule

$$\forall X,\ target\_class(X) \Leftrightarrow$$
$$condition_1(X)[t : w_1, d : w'_1] \vee \ldots \vee condition_n(X)[t : w_n, d : w'_n]$$

  - necessary and sufficient

Example: Quantitative Description Rule

Location/item		TV			Computer			Both_items	
	Count	*t-wt*	*d-wt*	*Count*	*t-wt*	*d-wt*	*Count*	*t-wt*	*d-wt*
Europe	80	25%	40%	240	75%	30%	320	100%	32%
N_Am	120	17.65%	60%	560	82.35%	70%	680	100%	68%
Both_ regions	200	20%	100%	800	80%	100%	1000	100%	100%

Crosstab showing associated t-weight, d-weight values and total number (in thousands) of TVs and computers sold at AllElectronics in 1998

- Quantitative description rule for target class *Europe*

$$\forall X, Europe(X) \Leftrightarrow$$
$$(item(X) = "TV")[t : 25\%, d : 40\%] \vee (item(X) = "computer")[t : 75\%, d : 30\%]$$

Mining Complex Data Objects: Generalization of Structured Data

- Set-valued attribute
  - Generalization of each value in the set into its corresponding higher-level concepts
  - Derivation of the general behavior of the set, such as the number of elements in the set, the types or value ranges in the set, or the weighted average for numerical data
  - E.g., *hobby* = {*tennis, hockey, chess, violin, nintendo_games*} generalizes to {*sports, music, video_games*}
- List-valued or a sequence-valued attribute
  - Same as set-valued attributes except that the order of the elements in the sequence should be observed in the generalization

Generalizing Spatial and Multimedia Data

- Spatial data:
  - Generalize detailed geographic points into clustered regions, such as business, residential, industrial, or agricultural areas, according to land usage
  - Require the merge of a set of geographic areas by spatial operations
- Image data:
  - Extracted by aggregation and/or approximation
  - Size, color, shape, texture, orientation, and relative positions and structures of the contained objects or regions in the image
- Music data:
  - Summarize its melody: based on the approximate patterns that repeatedly occur in the segment
  - Summarized its style: based on its tone, tempo, or the major musical instruments played

Generalizing Object Data

- Object identifier: generalize to the lowest level of class in the class/subclass hierarchies
- Class composition hierarchies
  - generalize nested structured data
  - generalize only objects closely related in semantics to the current one
- Construction and mining of object cubes
  - Extend the attribute-oriented induction method
    - Apply a sequence of class-based generalization operators on different attributes
    - Continue until getting a small number of generalized objects that can be summarized as a concise in high-level terms
  - For efficient implementation
    - Examine each attribute, generalize it to simple-valued data
    - Construct a multidimensional data cube (object cube)
    - Problem: it is not always desirable to generalize a set of values to single-valued data

An Example: Plan Mining by Divide & Conquer

- Plan: a variable sequence of actions
  - E.g., Travel (flight): <traveler, departure, arrival, d-time, a-time, airline, price, seat>
- Plan mining: extraction of important or significant generalized (sequential) patterns from a planbase (a large collection of plans)
  - E.g., Discover travel patterns in an air flight database, or
  - find significant patterns from the sequences of actions in the repair of automobiles
- Method
  - Attribute-oriented induction on sequence data
    - A generalized travel plan: <small-big*-small>
  - Divide & conquer:Mine characteristics for each subsequence
    - E.g., big*: same airline, small-big: nearby region

A Travel Database for Plan Mining

- Example: Mining a travel planbase

Travel plans table

plan#	action#	departure	depart_time	arrival	arrival_time	airline	...
1	1	ALB	800	JFK	900	TWA	...
1	2	JFK	1000	ORD	1230	UA	...
1	3	ORD	1300	LAX	1600	UA	...
1	4	LAX	1710	SAN	1800	DAL	...
2	1	SPI	900	ORD	950	AA	...
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

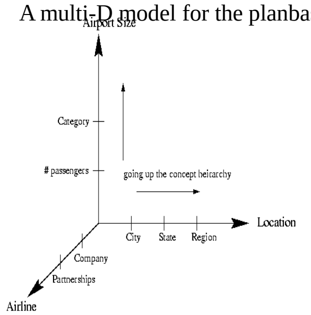
Airport info table

airport_code	city	state	region	airport_size	...
1	1	ALB		800	...
1	2	JFK		1000	...
1	3	ORD		1300	...
1	4	LAX		1710	...
2	1	SPI		900	...
.	.	.		.	.
.	.	.		.	.
.	.	.		.	.

Multidimensional Analysis

- Strategy
  - Generalize the planbase in different directions
  - Look for sequential patterns in the generalized plans
  - Derive high-level plans

A multi-D model for the planbase



Multidimensional Generalization

Multi-D generalization of the planbase

Plan#	Loc_Seq	Size_Seq	State_Seq
1	ALB - JFK - ORD - LAX - SAN	S - L - L - L - S	N - N - I - C - C
2	SPI - ORD - JFK - SYR	S - L - L - S	I - I - N - N
.	.		.
.	.		.
.	.		.

Merging consecutive, identical actions in plans

Plan#	Size_Seq	State_Seq	Region_Seq	...
1	S - L+ - S	N+ - I - C+	E+ - M - P+	...
2	S - L+ - S	I+ - N+	M+ - E+	...
.		.		.
.		.		.
.		.		.

$$flight(x, y,) \wedge airport\_size(x, S) \wedge airport\_size(y, L)$$
$$\Rightarrow region(x) = region(y) \quad [75\%]$$

Generalization-Based Sequence Mining

- Generalize planbase in multidimensional way using dimension tables
- Use # of distinct values (cardinality) at each level to determine the right level of generalization (level-“planning”)
- Use operators *merge* “+”, *option* “[]” to further generalize patterns
- Retain patterns with significant support

Generalized Sequence Patterns

- AirportSize-sequence survives the min threshold (after applying *merge* operator):

  S-L^+-**S** [35%], L^+-**S** [30%], **S**-L^+ [24.5%], L^+ [9%]
- After applying *option* operator:

  [S]-L^+-**[S]** [98.5%]
  - Most of the time, people fly via large airports to get to final destination
- Other plans: 1.5% of chances, there are other patterns: **S**-**S**, **L**-**S**-**L**

Chapter 5: Concept Description: Characterization and Comparison

- What is concept description?
- Data generalization and summarization-based characterization
- Analytical characterization: Analysis of attribute relevance
- Mining class comparisons: Discriminating between different classes
- Mining descriptive statistical measures in large databases
- Discussion
- Summary

Mining Data Dispersion Characteristics

- Motivation
  - To better understand the data: central tendency, variation and spread
- Data dispersion characteristics
  - median, max, min, quantiles, outliers, variance, etc.
- Numerical dimensions correspond to sorted intervals
  - Data dispersion: analyzed with multiple granularities of precision
  - Boxplot or quantile analysis on sorted intervals
- Dispersion analysis on computed measures
  - Folding measures into numerical dimensions
  - Boxplot or quantile analysis on the transformed cube

Measuring the Central Tendency

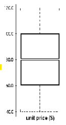
- Mean $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
  - Weighted arithmetic mean $\overline{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$
- Median: A holistic measure
  - Middle value if odd number of values, or average of the middle two values otherwise
  - estimated by interpolation $median = L_1 + (\frac{n/2 - (\sum f)l}{f_{median}})c$
- Mode
  - Value that occurs most frequently in the data
  - Unimodal, bimodal, trimodal
  - Empirical formula: $mean - mode = 3 \times (mean - median)$

Measuring the Dispersion of Data

- Quartiles, outliers and boxplots
  - Quartiles: Q_1 (25th percentile), Q_3 (75th percentile)
  - Inter-quartile range: IQR = $Q_3 - Q_1$
  - Five number summary: min, Q_1, M, Q_3, max
  - Boxplot: ends of the box are the quartiles, median is marked, whiskers, and plot outlier individually
  - Outlier: usually, a value higher/lower than 1.5 x IQR
- Variance and standard deviation
  - Variance s^2: (algebraic, scalable computation)

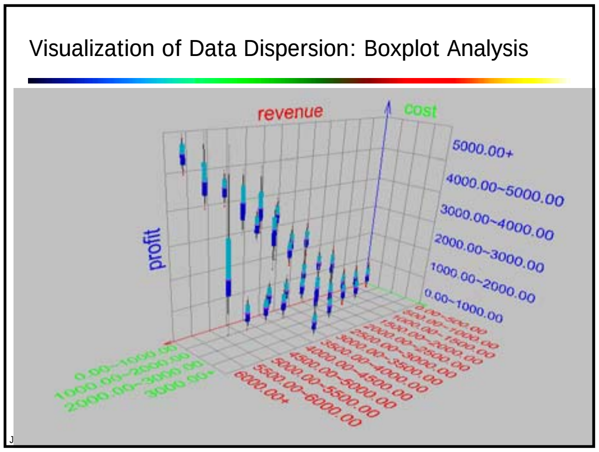
    $$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 = \frac{1}{n-1}[\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2]$$
  - Standard deviation s is the square root of variance s^2

Boxplot Analysis



- Five-number summary of a distribution:

  Minimum, Q1, M, Q3, Maximum
- Boxplot
  - Data is represented with a box
  - The ends of the box are at the first and third quartiles, i.e., the height of the box is IRQ
  - The median is marked by a line within the box
  - Whiskers: two lines outside the box extend to Minimum and Maximum



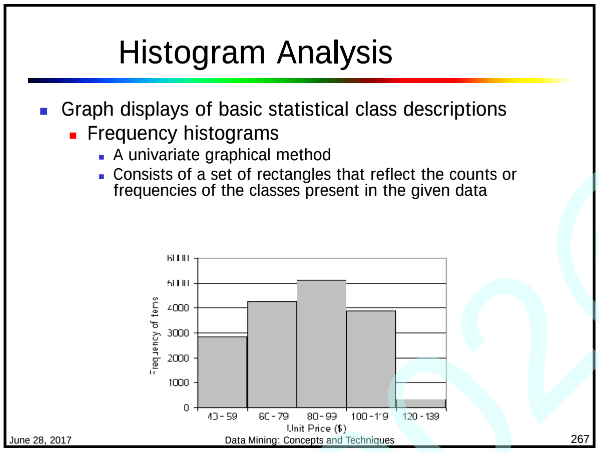
Visualization of Data Dispersion: Boxplot Analysis

Mining Descriptive Statistical Measures in Large Databases

- Variance

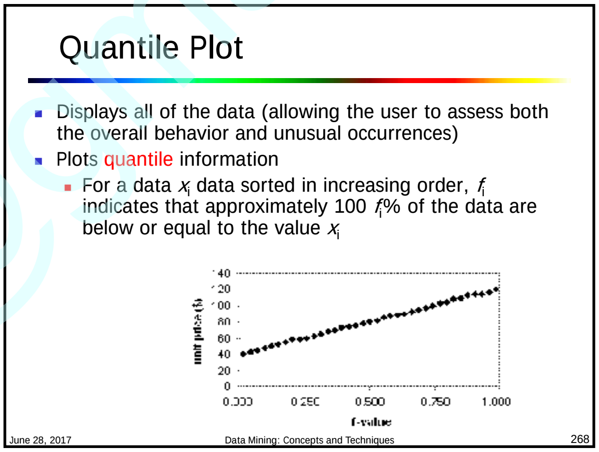
$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}\left[\sum x_i^2 - \frac{1}{n}\left(\sum x_i\right)^2\right]$$

- Standard deviation: the square root of the variance
  - Measures spread about the mean
  - It is zero if and only if all the values are equal
  - Both the deviation and the variance are algebraic



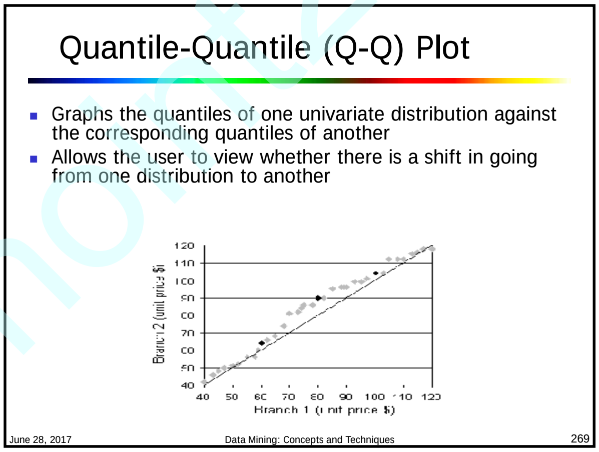
Histogram Analysis

- Graph displays of basic statistical class descriptions
  - Frequency histograms
    - A univariate graphical method
    - Consists of a set of rectangles that reflect the counts or frequencies of the classes present in the given data



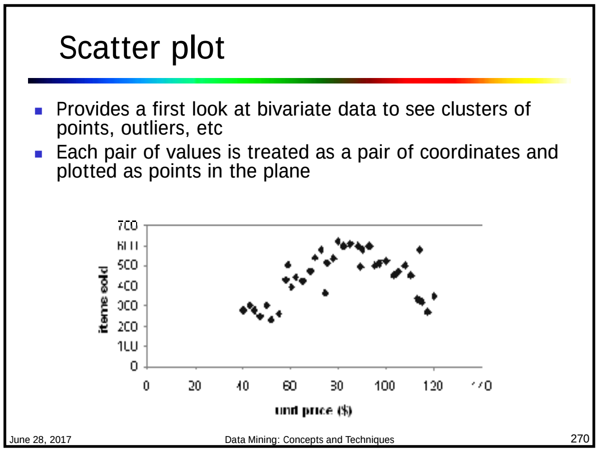
Quantile Plot

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



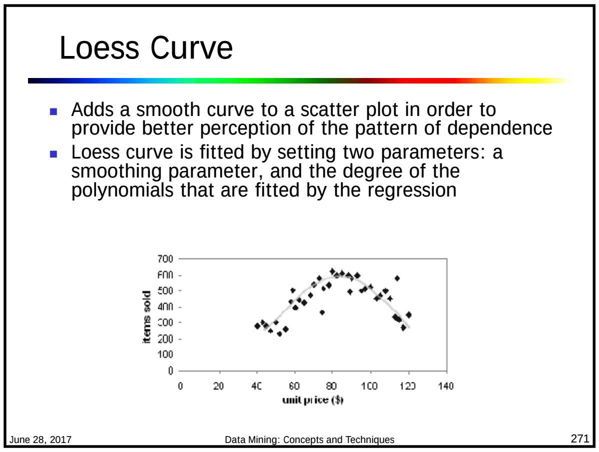
Quantile-Quantile (Q-Q) Plot

- Graphs the quantiles of one univariate distribution against the corresponding quantiles of another
- Allows the user to view whether there is a shift in going from one distribution to another



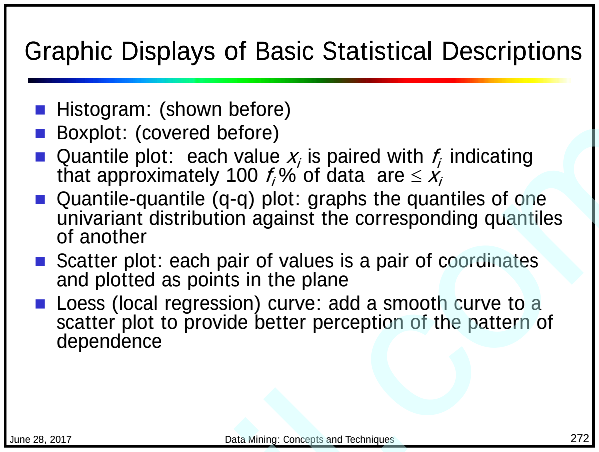
Scatter plot

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



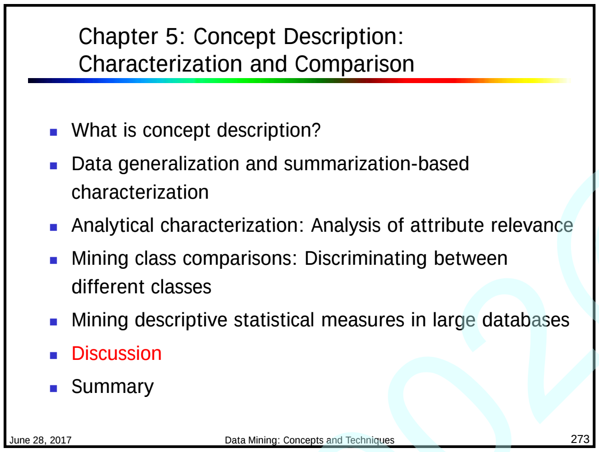
Loess Curve

- Adds a smooth curve to a scatter plot in order to provide better perception of the pattern of dependence
- Loess curve is fitted by setting two parameters: a smoothing parameter, and the degree of the polynomials that are fitted by the regression



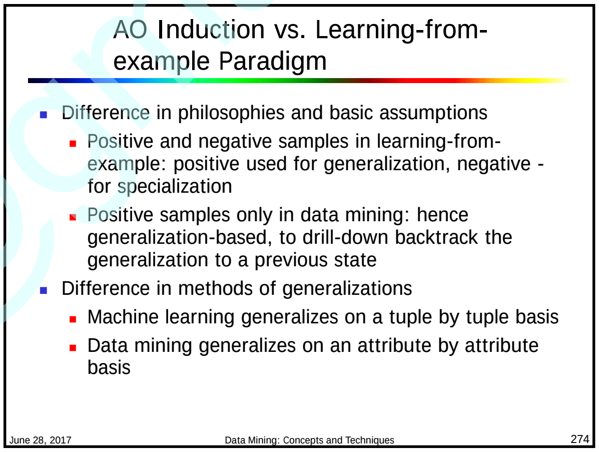
Graphic Displays of Basic Statistical Descriptions

- Histogram: (shown before)
- Boxplot: (covered before)
- Quantile plot: each value x_i is paired with f_i indicating that approximately 100 f_i% of data are $\leq x_i$
- Quantile-quantile (q-q) plot: graphs the quantiles of one univariant distribution against the corresponding quantiles of another
- Scatter plot: each pair of values is a pair of coordinates and plotted as points in the plane
- Loess (local regression) curve: add a smooth curve to a scatter plot to provide better perception of the pattern of dependence



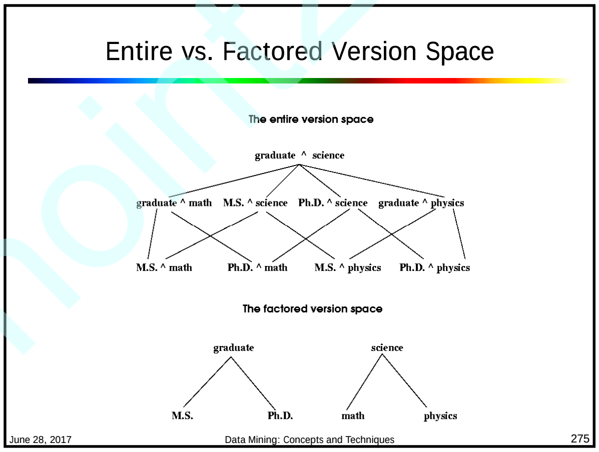
Chapter 5: Concept Description: Characterization and Comparison

- What is concept description?
- Data generalization and summarization-based characterization
- Analytical characterization: Analysis of attribute relevance
- Mining class comparisons: Discriminating between different classes
- Mining descriptive statistical measures in large databases
- Discussion
- Summary

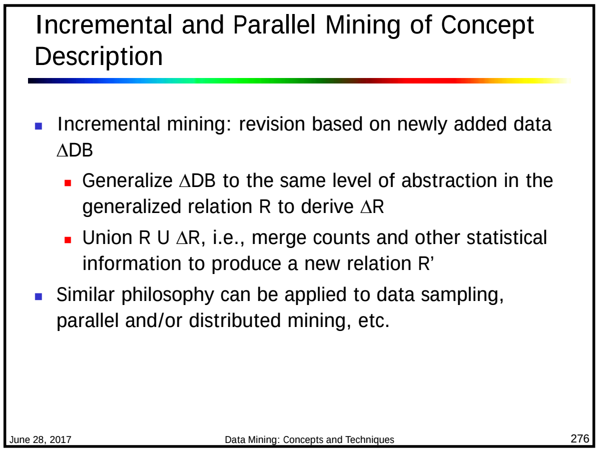


AO Induction vs. Learning-from-example Paradigm

- Difference in philosophies and basic assumptions
  - Positive and negative samples in learning-from-example: positive used for generalization, negative - for specialization
  - Positive samples only in data mining: hence generalization-based, to drill-down backtrack the generalization to a previous state
- Difference in methods of generalizations
  - Machine learning generalizes on a tuple by tuple basis
  - Data mining generalizes on an attribute by attribute basis

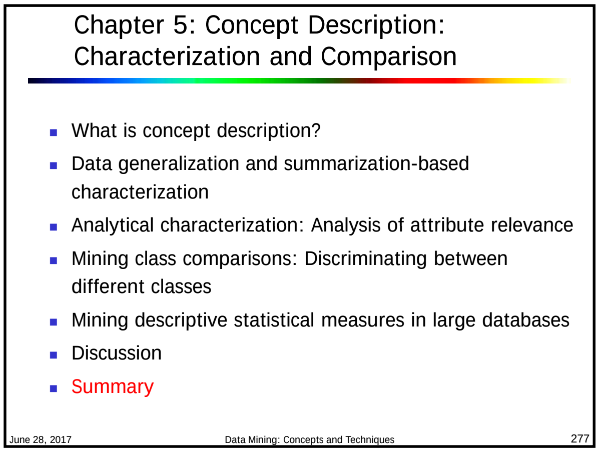


Entire vs. Factored Version Space



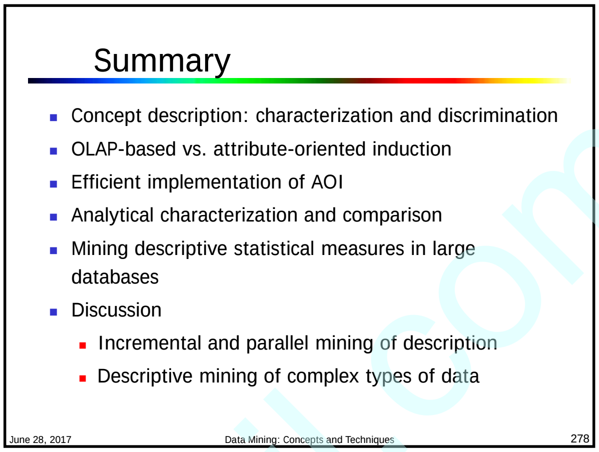
Incremental and Parallel Mining of Concept Description

- Incremental mining: revision based on newly added data ΔDB
  - Generalize ΔDB to the same level of abstraction in the generalized relation R to derive ΔR
  - Union R U ΔR, i.e., merge counts and other statistical information to produce a new relation R'
- Similar philosophy can be applied to data sampling, parallel and/or distributed mining, etc.



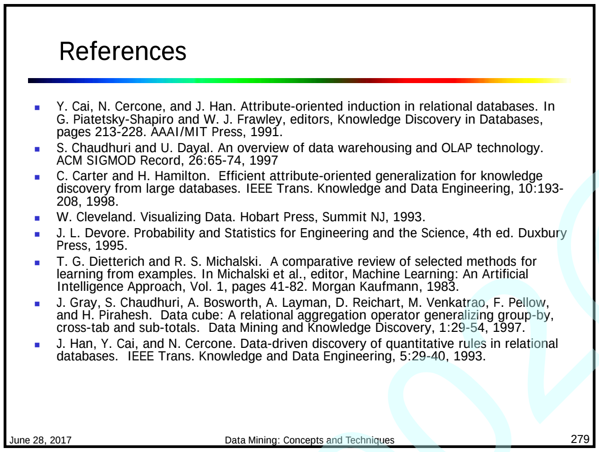
Chapter 5: Concept Description: Characterization and Comparison

- What is concept description?
- Data generalization and summarization-based characterization
- Analytical characterization: Analysis of attribute relevance
- Mining class comparisons: Discriminating between different classes
- Mining descriptive statistical measures in large databases
- Discussion
- Summary



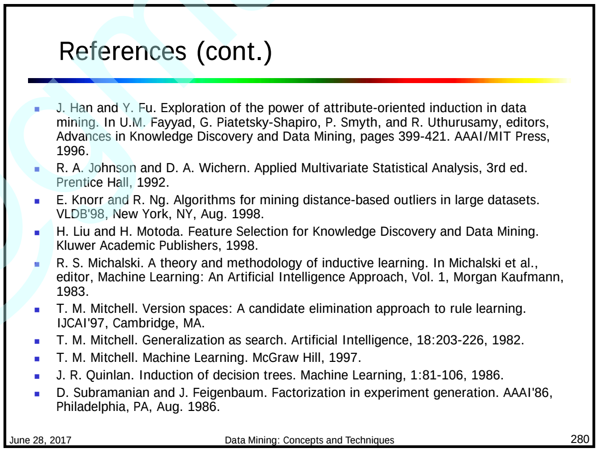
Summary

- Concept description: characterization and discrimination
- OLAP-based vs. attribute-oriented induction
- Efficient implementation of AOI
- Analytical characterization and comparison
- Mining descriptive statistical measures in large databases
- Discussion
  - Incremental and parallel mining of description
  - Descriptive mining of complex types of data



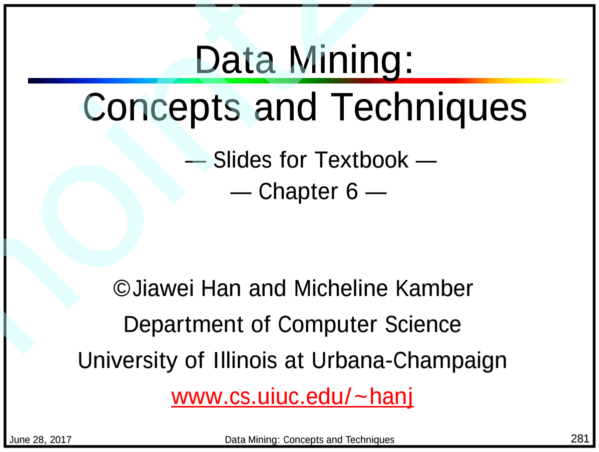
References

- Y. Cai, N. Cercone, and J. Han. Attribute-oriented induction in relational databases. In G. Piatetsky-Shapiro and W. J. Frawley, editors, Knowledge Discovery in Databases, pages 213-228. AAAI/MIT Press, 1991.
- S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26:65-74, 1997
- C. Carter and H. Hamilton. Efficient attribute-oriented generalization for knowledge discovery from large databases. IEEE Trans. Knowledge and Data Engineering, 10:193-208, 1998.
- W. Cleveland. Visualizing Data. Hobart Press, Summit NJ, 1993.
- J. L. Devore. Probability and Statistics for Engineering and the Science, 4th ed. Duxbury Press, 1995.
- T. G. Dietterich and R. S. Michalski. A comparative review of selected methods for learning from examples. In Michalski et al., editor, Machine Learning: An Artificial Intelligence Approach, Vol. 1, pages 41-82. Morgan Kaufmann, 1983.
- J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. Data Mining and Knowledge Discovery, 1:29-54, 1997.
- J. Han, Y. Cai, and N. Cercone. Data-driven discovery of quantitative rules in relational databases. IEEE Trans. Knowledge and Data Engineering, 5:29-40, 1993.



References (cont.)

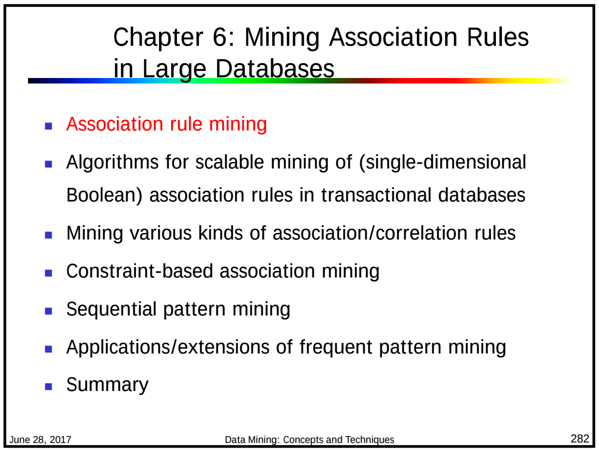
- J. Han and Y. Fu. Exploration of the power of attribute-oriented induction in data mining. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pages 399-421. AAAI/MIT Press, 1996.
- R. A. Johnson and D. A. Wichern. Applied Multivariate Statistical Analysis, 3rd ed. Prentice Hall, 1992.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98, New York, NY, Aug. 1998.
- H. Liu and H. Motoda. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, 1998.
- R. S. Michalski. A theory and methodology of inductive learning. In Michalski et al., editor, Machine Learning: An Artificial Intelligence Approach, Vol. 1, Morgan Kaufmann, 1983.
- T. M. Mitchell. Version spaces: A candidate elimination approach to rule learning. IJCAI'97, Cambridge, MA.
- T. M. Mitchell. Generalization as search. Artificial Intelligence, 18:203-226, 1982.
- T. M. Mitchell. Machine Learning. McGraw Hill, 1997.
- J. R. Quinlan. Induction of decision trees. Machine Learning, 1:81-106, 1986.
- D. Subramanian and J. Feigenbaum. Factorization in experiment generation. AAAI'86, Philadelphia, PA, Aug. 1986.



Data Mining: Concepts and Techniques

— Slides for Textbook —
— Chapter 6 —

©Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj



Chapter 6: Mining Association Rules in Large Databases

- Association rule mining
- Algorithms for scalable mining of (single-dimensional Boolean) association rules in transactional databases
- Mining various kinds of association/correlation rules
- Constraint-based association mining
- Sequential pattern mining
- Applications/extensions of frequent pattern mining
- Summary

What Is Association Mining?

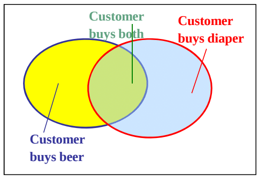
- Association rule mining:
  - Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
  - Frequent pattern: pattern (set of items, sequence, etc.) that occurs frequently in a database [AIS93]
- Motivation: finding regularities in data
  - What products were often purchased together? — Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?

Why Is Frequent Pattern or Assoiciation Mining an Essential Task in Data Mining?

- Foundation for many essential data mining tasks
  - Association, correlation, causality
  - Sequential patterns, temporal or cyclic association, partial periodicity, spatial and multimedia association
  - Associative classification, cluster analysis, iceberg cube, fascicles (semantic data compression)
- Broad applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis
  - Web log (click stream) analysis, DNA sequence analysis, etc.

Basic Concepts: Frequent Patterns and Association Rules

Transaction-id	Items bought
10	A, B, C
20	A, C
30	A, D
40	B, E, F



- Itemset X={x_1, …, x_k}
- Find all the rules *X→Y* with min confidence and support
  - support, *s*, probability that a transaction contains X∪Y
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*.

Let min_support = 50%, min_conf = 50%:

$A \rightarrow C$ (50%, 66.7%)

$C \rightarrow A$ (50%, 100%)

Mining Association Rules—an Example

Transaction-id	Items bought
10	A, B, C
20	A, C
30	A, D
40	B, E, F

Min. support 50%
Min. confidence 50%

Frequent pattern	Support
{A}	75%
{B}	50%
{C}	50%
{A, C}	50%

For rule $A \Rightarrow C$:

support = support($\{A\} \cup \{C\}$) = 50%

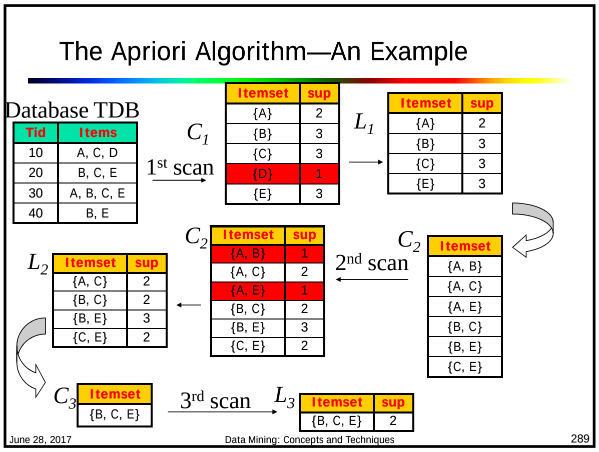
confidence = support($\{A\} \cup \{C\}$)/support($\{A\}$) = 66.6%

Chapter 6: Mining Association Rules in Large Databases

- Association rule mining
- Algorithms for scalable mining of (single-dimensional Boolean) association rules in transactional databases
- Mining various kinds of association/correlation rules
- Constraint-based association mining
- Sequential pattern mining
- Applications/extensions of frequent pattern mining
- Summary

Apriori: A Candidate Generation-and-test Approach

- Any subset of a frequent itemset must be frequent
  - if **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
  - Every transaction having {beer, diaper, nuts} also contains {beer, diaper}
- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested!
- Method:
  - generate length (k+1) candidate itemsets from length k frequent itemsets, and
  - test the candidates against DB
- The performance studies show its efficiency and scalability
- Agrawal & Srikant 1994, Mannila, et al. 1994



The Apriori Algorithm—An Example

Database TDB

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan → C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

→ L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan ← C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

← L_2

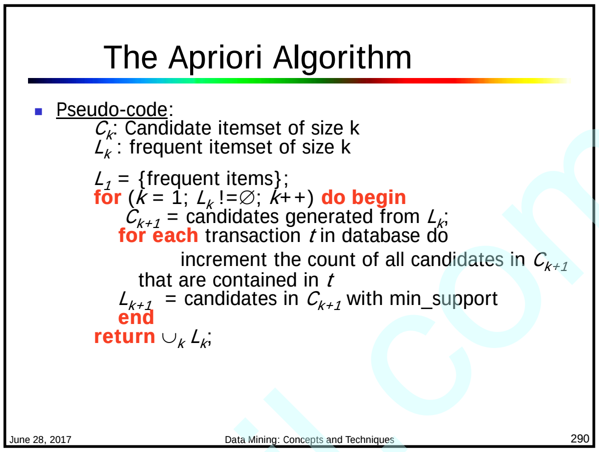
Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3rd scan → L_3

Itemset	sup
{B, C, E}	2

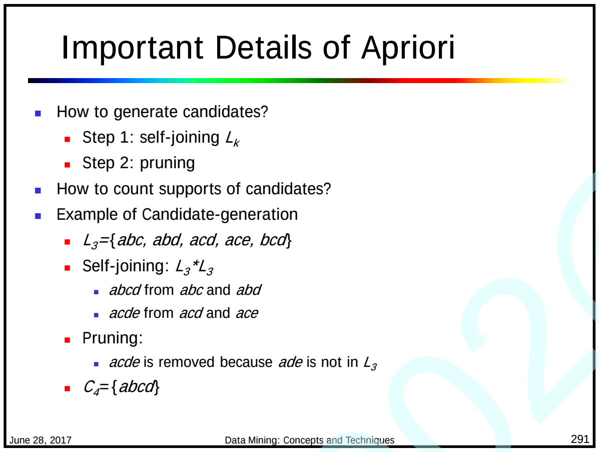


The Apriori Algorithm

- Pseudo-code:

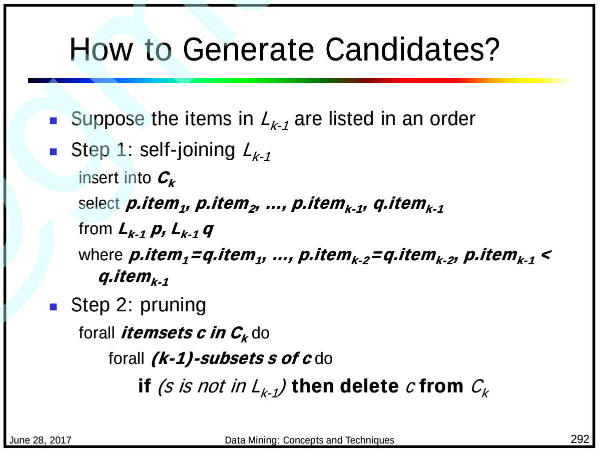
```
$C_k$: Candidate itemset of size k
$L_k$ : frequent itemset of size k

$L_1$ = {frequent items};
for ($k$ = 1; $L_k$ !=∅; $k$++) do begin
    $C_{k+1}$ = candidates generated from $L_k$;
    for each transaction $t$ in database do
            increment the count of all candidates in $C_{k+1}$
        that are contained in $t$
    $L_{k+1}$ = candidates in $C_{k+1}$ with min_support
    end
return $\cup_k L_k$;
```



Important Details of Apriori

- How to generate candidates?
  - Step 1: self-joining L_k
  - Step 2: pruning
- How to count supports of candidates?
- Example of Candidate-generation
  - L_3={*abc, abd, acd, ace, bcd*}
  - Self-joining: L_3*L_3
    - *abcd* from *abc* and *abd*
    - *acde* from *acd* and *ace*
  - Pruning:
    - *acde* is removed because *ade* is not in L_3
  - C_4={*abcd*}



How to Generate Candidates?

- Suppose the items in L_{k-1} are listed in an order
- Step 1: self-joining L_{k-1}

  insert into $\boldsymbol{C_k}$

  select ***p.item$_1$, p.item$_2$, …, p.item$_{k-1}$, q.item$_{k-1}$***

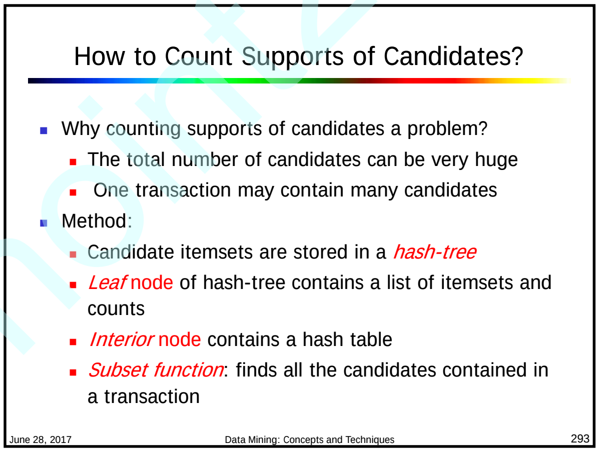
  from ***L_{k-1} p, L_{k-1} q***

  where ***p.item$_1$=q.item$_1$, …, p.item$_{k-2}$=q.item$_{k-2}$, p.item$_{k-1}$ < q.item$_{k-1}$***
- Step 2: pruning

  forall ***itemsets c in C_k*** do

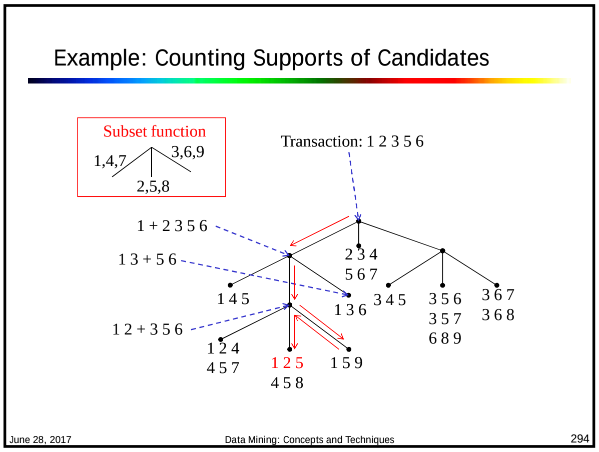
  forall ***(k-1)-subsets s of c*** do

  if *(s is not in L_{k-1})* **then delete** *c* **from** C_k



How to Count Supports of Candidates?

- Why counting supports of candidates a problem?
  - The total number of candidates can be very huge
  - One transaction may contain many candidates
- Method:
  - Candidate itemsets are stored in a *hash-tree*
  - *Leaf* node of hash-tree contains a list of itemsets and counts
  - *Interior* node contains a hash table
  - *Subset function*: finds all the candidates contained in a transaction



Example: Counting Supports of Candidates

Efficient Implementation of Apriori in SQL

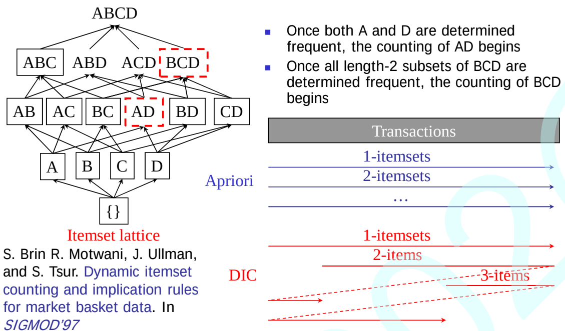
- Hard to get good performance out of pure SQL (SQL-92) based approaches alone
- Make use of object-relational extensions like UDFs, BLOBs, Table functions etc.
  - Get orders of magnitude improvement
- S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. In SIGMOD'98

Challenges of Frequent Pattern Mining

- Challenges
  - Multiple scans of transaction database
  - Huge number of candidates
  - Tedious workload of support counting for candidates
- Improving Apriori: general ideas
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates

DIC: Reduce Number of Scans

- Once both A and D are determined frequent, the counting of AD begins
- Once all length-2 subsets of BCD are determined frequent, the counting of BCD begins



Itemset lattice

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

Partition: Scan Database Only Twice

- Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
  - Scan 1: partition database and find local frequent patterns
  - Scan 2: consolidate global frequent patterns
- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association in large databases. In *VLDB'95*

Sampling for Frequent Patterns

- Select a sample of original database, mine frequent patterns within sample using Apriori
- Scan database once to verify frequent itemsets found in sample, only *borders* of closure of frequent patterns are checked
  - Example: check *abcd* instead of *ab, ac, …, etc.*
- Scan database again to find missed frequent patterns
- H. Toivonen. Sampling large databases for association rules. In *VLDB'96*

DHP: Reduce the Number of Candidates

- A *k*-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent
  - Candidates: a, b, c, d, e
  - Hash entries: {ab, ad, ae} {bd, be, de} …
  - Frequent 1-itemset: a, b, d, e
  - ab is not a candidate 2-itemset if the sum of count of {ab, ad, ae} is below support threshold
- J. Park, M. Chen, and P. Yu. An effective hash-based algorithm for mining association rules. In *SIGMOD'95*

Eclat/MaxEclat and VIPER: Exploring Vertical Data Format

- Use tid-list, the list of transaction-ids containing an itemset
- Compression of tid-lists
  - Itemset A: t1, t2, t3, sup(A)=3
  - Itemset B: t2, t3, t4, sup(B)=3
  - Itemset AB: t2, t3, sup(AB)=2
- Major operation: intersection of tid-lists
- M. Zaki et al. New algorithms for fast discovery of association rules. In KDD'97
- P. Shenoy et al. Turbo-charging vertical mining of large databases. In SIGMOD'00

Bottleneck of Frequent-pattern Mining

- Multiple database scans are costly
- Mining long patterns needs many passes of scanning and generates lots of candidates
  - To find frequent itemset $i_1i_2...i_{100}$
    - # of scans: 100
    - # of Candidates: $(_{100}{}^{1}) + (_{100}{}^{2}) + ... + (_{1}{}^{1}{}_{0}{}^{0}{}_{0}{}^{0}) = 2^{100}-1 = 1.27*10^{30}$!
- Bottleneck: candidate-generation-and-test
- Can we avoid candidate generation?

Mining Frequent Patterns Without Candidate Generation

- Grow long patterns from short ones using local frequent items
  - "abc" is a frequent pattern
  - Get all transactions having "abc": DB|abc
  - "d" is a local frequent item in DB|abc → abcd is a frequent pattern

Construct FP-tree from a Transaction Database

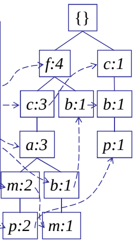
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Sort frequent items in frequency descending order, f-list
3. Scan DB again

Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



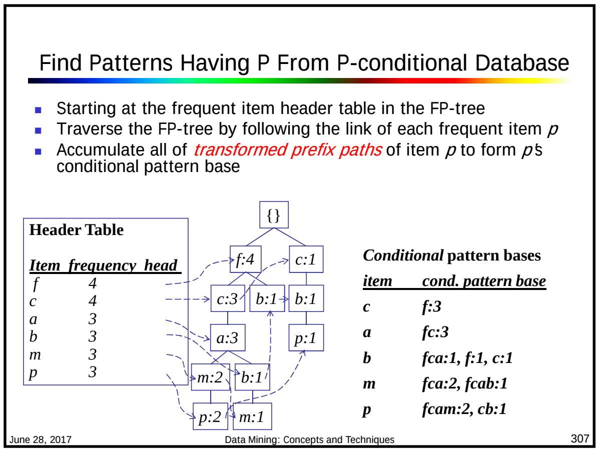
F-list=f-c-a-b-m-p

Benefits of the FP-tree Structure

- Completeness
  - Preserve complete information for frequent pattern mining
  - Never break a long pattern of any transaction
- Compactness
  - Reduce irrelevant info—infrequent items are gone
  - Items in frequency descending order: the more frequently occurring, the more likely to be shared
  - Never be larger than the original database (not count node-links and the *count* field)
  - For Connect-4 DB, compression ratio could be over 100

Partition Patterns and Databases

- Frequent patterns can be partitioned into subsets according to f-list
  - F-list=f-c-a-b-m-p
  - Patterns containing p
  - Patterns having m but no p
  - ...
  - Patterns having c but no a nor b, m, p
  - Pattern f
- Completeness and non-redundency



Find Patterns Having P From P-conditional Database

- Starting at the frequent item header table in the FP-tree
- Traverse the FP-tree by following the link of each frequent item *p*
- Accumulate all of *transformed prefix paths* of item *p* to form *p*'s conditional pattern base

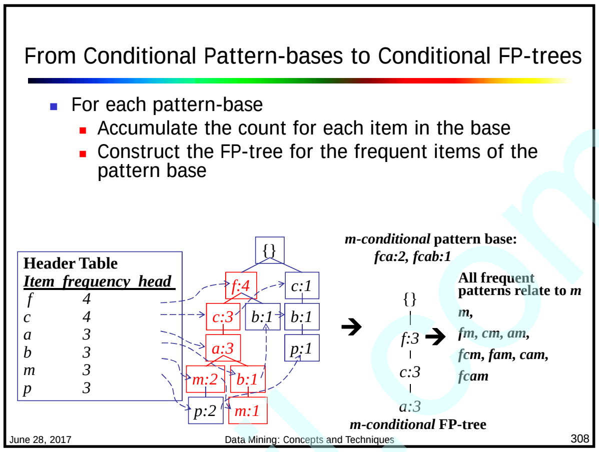
Header Table

Item	*frequency*	*head*
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

***Conditional* pattern bases**

item	*cond. pattern base*
c	*f:3*
a	*fc:3*
b	*fca:1, f:1, c:1*
m	*fca:2, fcab:1*
p	*fcam:2, cb:1*

June 28, 2017 Data Mining: Concepts and Techniques 307



From Conditional Pattern-bases to Conditional FP-trees

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

Header Table

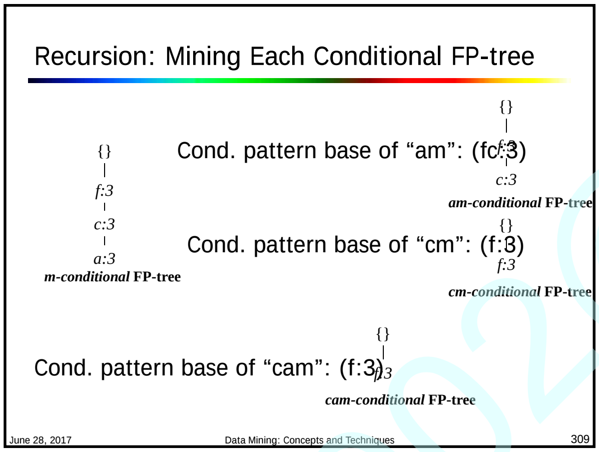
Item	*frequency*	*head*
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

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

All frequent patterns relate to *m*
m,
fm, cm, am,
fcm, fam, cam,
fcam

***m-conditional* FP-tree**

June 28, 2017 Data Mining: Concepts and Techniques 308



Recursion: Mining Each Conditional FP-tree

***m-conditional* FP-tree**

Cond. pattern base of “am”: (fc:3)

***am-conditional* FP-tree**

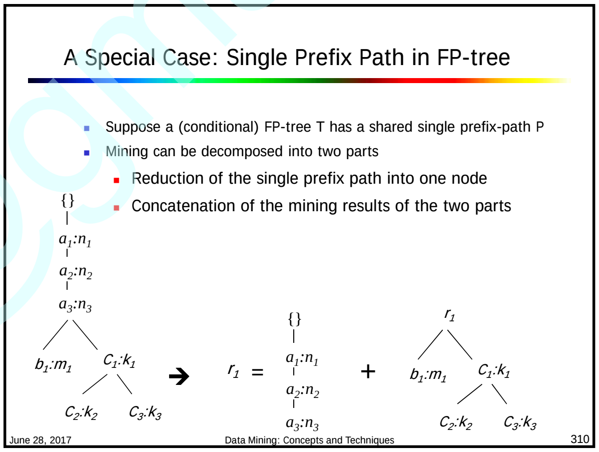
Cond. pattern base of “cm”: (f:3)

***cm-conditional* FP-tree**

Cond. pattern base of “cam”: (f:3)

***cam-conditional* FP-tree**

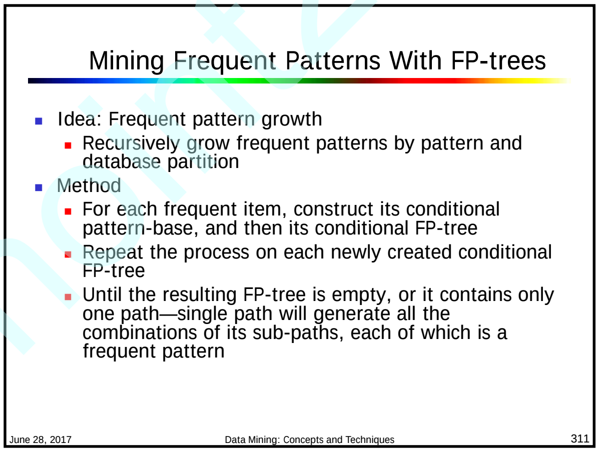
June 28, 2017 Data Mining: Concepts and Techniques 309



A Special Case: Single Prefix Path in FP-tree

- Suppose a (conditional) FP-tree T has a shared single prefix-path P
- Mining can be decomposed into two parts
  - Reduction of the single prefix path into one node
  - Concatenation of the mining results of the two parts

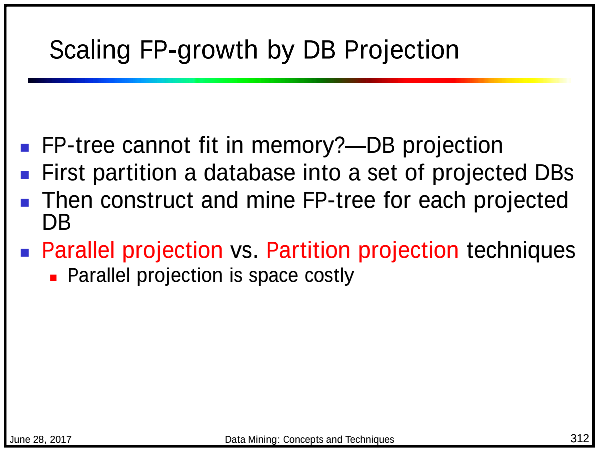
June 28, 2017 Data Mining: Concepts and Techniques 310



Mining Frequent Patterns With FP-trees

- Idea: Frequent pattern growth
  - Recursively grow frequent patterns by pattern and database partition
- Method
  - For each frequent item, construct its conditional pattern-base, and then its conditional FP-tree
  - Repeat the process on each newly created conditional FP-tree
  - Until the resulting FP-tree is empty, or it contains only one path—single path will generate all the combinations of its sub-paths, each of which is a frequent pattern

June 28, 2017 Data Mining: Concepts and Techniques 311



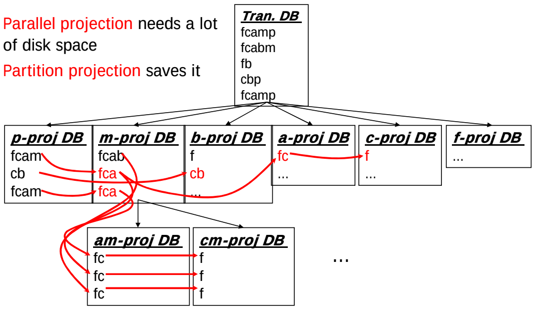
Scaling FP-growth by DB Projection

- FP-tree cannot fit in memory?—DB projection
- First partition a database into a set of projected DBs
- Then construct and mine FP-tree for each projected DB
- Parallel projection vs. Partition projection techniques
  - Parallel projection is space costly

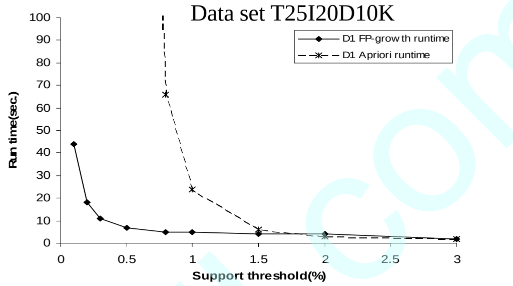
June 28, 2017 Data Mining: Concepts and Techniques 312

Partition-based Projection

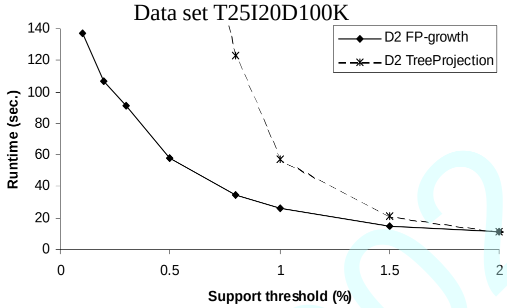
- Parallel projection needs a lot of disk space
- Partition projection saves it



FP-Growth vs. Apriori: Scalability With the Support Threshold



FP-Growth vs. Tree-Projection: Scalability with the Support Threshold



Why Is FP-Growth the Winner?

- Divide-and-conquer:
  - decompose both the mining task and DB according to the frequent patterns obtained so far
  - leads to focused search of smaller databases
- Other factors
  - no candidate generation, no candidate test
  - compressed database: FP-tree structure
  - no repeated scan of entire database
  - basic ops—counting local freq items and building sub FP-tree, no pattern search and matching

Implications of the Methodology

- Mining closed frequent itemsets and max-patterns
  - CLOSET (DMKD'00)
- Mining sequential patterns
  - FreeSpan (KDD'00), PrefixSpan (ICDE'01)
- Constraint-based mining of frequent patterns
  - Convertible constraints (KDD'00, ICDE'01)
- Computing iceberg data cubes with complex measures
  - H-tree and H-cubing algorithm (SIGMOD'01)

Max-patterns

- Frequent pattern $\{a_1, \ldots, a_{100}\} \rightarrow \binom{100}{1} + \binom{100}{2} + \ldots + \binom{100}{100} = 2^{100}-1 = 1.27*10^{30}$ frequent sub-patterns!
- Max-pattern: frequent patterns without proper frequent super pattern
  - BCDE, ACD are max-patterns
  - BCD is not a max-pattern

Min_sup=2

Tid	Items
10	A,B,C,D,E
20	B,C,D,E,
30	A,C,D,F

MaxMiner: Mining Max-patterns

- 1^{st} scan: find frequent items
  - A, B, C, D, E
- 2^{nd} scan: find support for
  - AB, AC, AD, AE, ABCDE
  - BC, BD, BE, BCDE
  - CD, CE, CDE, DE,
- Since BCDE is a max-pattern, no need to check BCD, BDE, CDE in later scan
- R. Bayardo. Efficiently mining long patterns from databases. In *SIGMOD'98*

Tid	Items
10	A,B,C,D,E
20	B,C,D,E,
30	A,C,D,F

Potential max-patterns

Frequent Closed Patterns

- Conf(ac→d)=100% → record acd only
- For frequent itemset X, if there exists no item y s.t. every transaction containing X also contains y, then X is a frequent closed pattern
  - "acd" is a frequent closed pattern
- Concise rep. of freq pats
- Reduce # of patterns and rules
- N. Pasquier et al. In ICDT'99

Min_sup=2

TID	Items
10	a, c, d, e, f
20	a, b, e
30	c, e, f
40	a, c, d, f
50	c, e, f

Mining Frequent Closed Patterns: CLOSET

- Flist: list of all frequent items in support ascending order
  - Flist: d-a-f-e-c
- Divide search space
  - Patterns having d
  - Patterns having d but no a, etc.
- Find frequent closed pattern recursively
  - Every transaction having d also has cfa → cfad is a frequent closed pattern
- J. Pei, J. Han & R. Mao. CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets", DMKD'00.

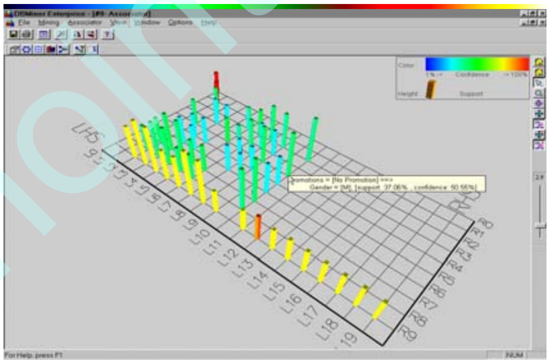
Min_sup=2

TID	Items
10	a, c, d, e, f
20	a, b, e
30	c, e, f
40	a, c, d, f
50	c, e, f

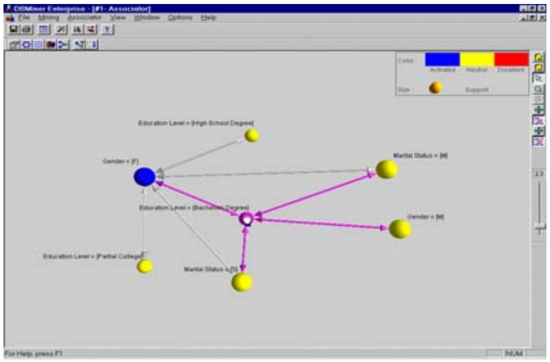
Mining Frequent Closed Patterns: CHARM

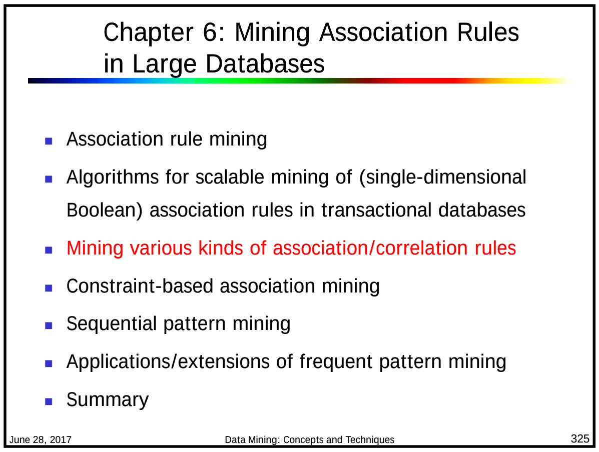
- Use vertical data format: t(AB)={T_1, T_{12}, …}
- Derive closed pattern based on vertical intersections
  - t(X)=t(Y): X and Y always happen together
  - t(X)⊂t(Y): transaction having X always has Y
- Use diffset to accelerate mining
  - Only keep track of difference of tids
  - t(X)={T_1, T_2, T_3}, t(Xy)={T_1, T_3}
  - Diffset(Xy, X)={T_2}
- M. Zaki. CHARM: An Efficient Algorithm for Closed Association Rule Mining, CS-TR99-10, Rensselaer Polytechnic Institute
- M. Zaki, Fast Vertical Mining Using Diffsets, TR01-1, Department of Computer Science, Rensselaer Polytechnic Institute

Visualization of Association Rules: Pane Graph


DBMiner Enterprise - [#0- Associator]
File Mining Associator View Window Options Help
Color
1% -> Confidence -> 100%
Height Support
Promotions = [No Promotion] ==>
Gender = [M] [support: 37.06% , confidence: 50.55%]
LHS
L10
L11
L12
L13
L14
L15
L16
L17
L18
L19
RHS
For Help, press F1
NUM

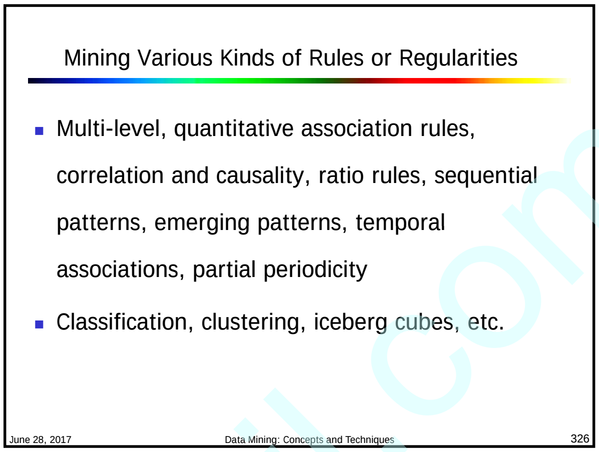
Visualization of Association Rules: Rule Graph


DBMiner Enterprise - [#1- Associator]
File Mining Associator View Window Options Help
Color
Activated Neutral Disabled
Size Support
Education Level = [High School Degree]
Gender = [F]
Marital Status = [M]
Education Level = [Bachelors Degree]
Gender = [M]
Education Level = [Partial College]
Marital Status = [S]
For Help, press F1
NUM



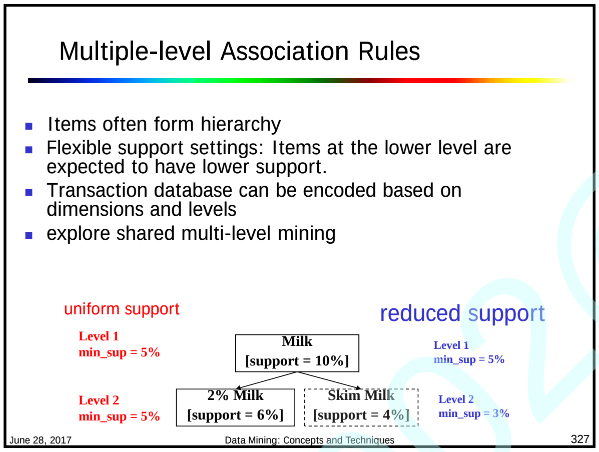
Chapter 6: Mining Association Rules in Large Databases

- Association rule mining
- Algorithms for scalable mining of (single-dimensional Boolean) association rules in transactional databases
- Mining various kinds of association/correlation rules
- Constraint-based association mining
- Sequential pattern mining
- Applications/extensions of frequent pattern mining
- Summary



Mining Various Kinds of Rules or Regularities

- Multi-level, quantitative association rules, correlation and causality, ratio rules, sequential patterns, emerging patterns, temporal associations, partial periodicity
- Classification, clustering, iceberg cubes, etc.



Multiple-level Association Rules

- Items often form hierarchy
- Flexible support settings: Items at the lower level are expected to have lower support.
- Transaction database can be encoded based on dimensions and levels
- explore shared multi-level mining

uniform support

Level 1
min_sup = 5%

Level 2
min_sup = 5%

Milk
[support = 10%]

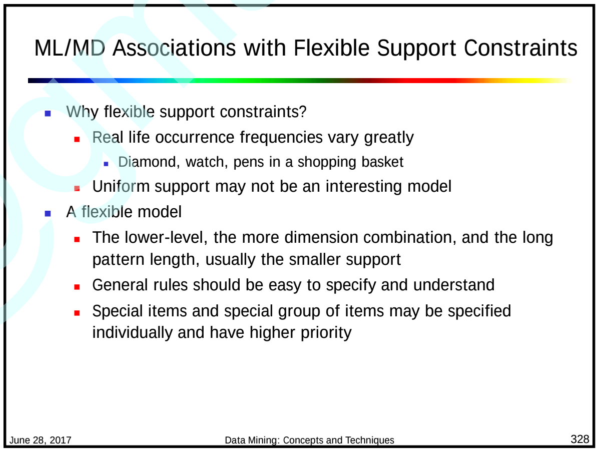
2% Milk
[support = 6%]

Skim Milk
[support = 4%]

reduced support

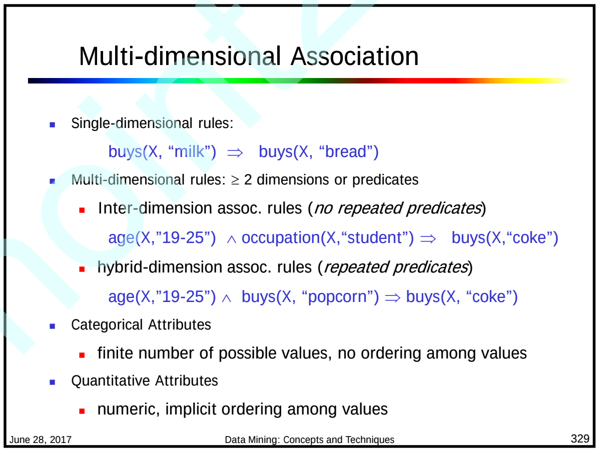
Level 1
min_sup = 5%

Level 2
min_sup = 3%



ML/MD Associations with Flexible Support Constraints

- Why flexible support constraints?
  - Real life occurrence frequencies vary greatly
    - Diamond, watch, pens in a shopping basket
  - Uniform support may not be an interesting model
- A flexible model
  - The lower-level, the more dimension combination, and the long pattern length, usually the smaller support
  - General rules should be easy to specify and understand
  - Special items and special group of items may be specified individually and have higher priority



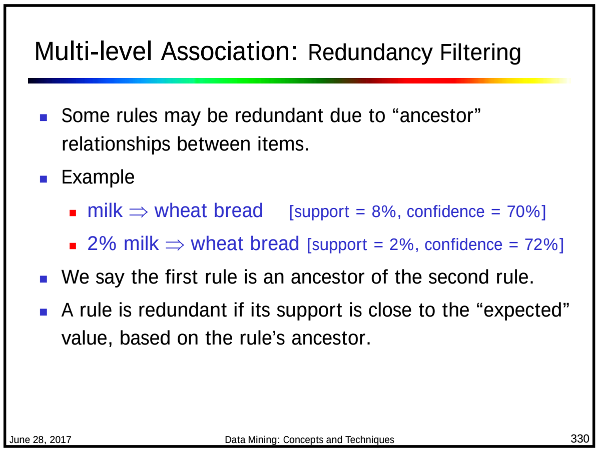
Multi-dimensional Association

- Single-dimensional rules:

  buys(X, "milk") ⇒ buys(X, "bread")
- Multi-dimensional rules: ≥ 2 dimensions or predicates
  - Inter-dimension assoc. rules (*no repeated predicates*)

    age(X,"19-25") ∧ occupation(X,"student") ⇒ buys(X,"coke")
  - hybrid-dimension assoc. rules (*repeated predicates*)

    age(X,"19-25") ∧ buys(X, "popcorn") ⇒ buys(X, "coke")
- Categorical Attributes
  - finite number of possible values, no ordering among values
- Quantitative Attributes
  - numeric, implicit ordering among values



Multi-level Association: Redundancy Filtering

- Some rules may be redundant due to "ancestor" relationships between items.
- Example
  - milk ⇒ wheat bread [support = 8%, confidence = 70%]
  - 2% milk ⇒ wheat bread [support = 2%, confidence = 72%]
- We say the first rule is an ancestor of the second rule.
- A rule is redundant if its support is close to the "expected" value, based on the rule's ancestor.

Multi-Level Mining: Progressive Deepening

- A top-down, progressive deepening approach:
  - First mine high-level frequent items:
    milk (15%), bread (10%)
  - Then mine their lower-level "weaker" frequent itemsets:
    2% milk (5%), wheat bread (4%)
- Different min_support threshold across multi-levels lead to different algorithms:
  - If adopting the same *min_support* across multi-levels
    then toss *t* if any of *t*'s ancestors is infrequent.
  - If adopting reduced *min_support* at lower levels
    then examine only those descendents whose ancestor's support is frequent/non-negligible.

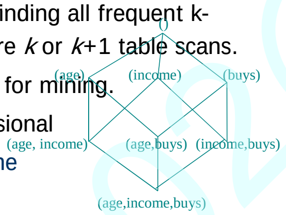
Techniques for Mining MD Associations

- Search for frequent *k*-predicate set:
  - Example: {age, occupation, buys} is a 3-predicate set
  - Techniques can be categorized by how age are treated

1. Using static discretization of quantitative attributes
   - Quantitative attributes are statically discretized by using predefined concept hierarchies
2. Quantitative association rules
   - Quantitative attributes are dynamically discretized into "bins"based on the distribution of the data
3. Distance-based association rules
   - This is a dynamic discretization process that considers the distance between data points

Static Discretization of Quantitative Attributes

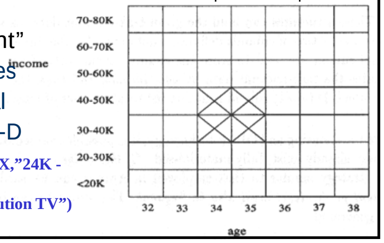
- Discretized prior to mining using concept hierarchy.
- Numeric values are replaced by ranges.
- In relational database, finding all frequent k-predicate sets will require *k* or *k*+1 table scans.
- Data cube is well suited for mining.
- The cells of an n-dimensional cuboid correspond to the predicate sets.



Quantitative Association Rules

- Numeric attributes are *dynamically* discretized
  - Such that the confidence or compactness of the rules mined is maximized
- 2-D quantitative association rules: $A_{quan1} \wedge A_{quan2} \Rightarrow A_{cat}$
- Cluster "adjacent" association rules to form general rules using a 2-D grid
- Example

age(X,"30-34") ∧ income(X,"24K - 48K")
⇒ buys(X,"high resolution TV")



Mining Distance-based Association Rules

- Binning methods do not capture the semantics of interval data

Price($)	Equi-width (width $10)	Equi-depth (depth 2)	Distance-based
7	[0,10]	[7,20]	[7,7]
20	[11,20]	[22,50]	[20,22]
22	[21,30]	[51,53]	[50,53]
50	[31,40]		
51	[41,50]		
53	[51,60]		

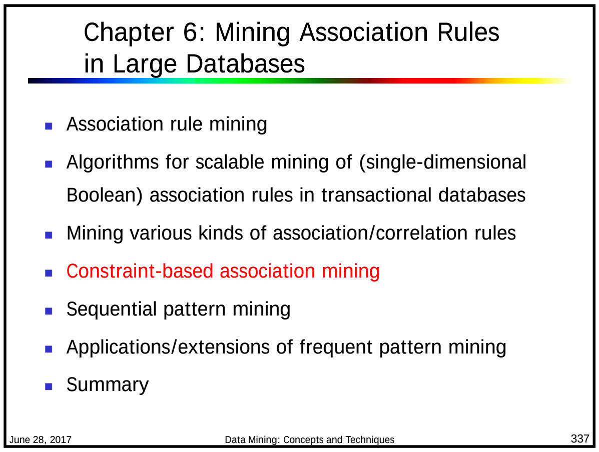
- Distance-based partitioning, more meaningful discretization considering:
  - density/number of points in an interval
  - "closeness" of points in an interval

Interestingness Measure: Correlations (Lift)

- *play basketball* ⇒ *eat cereal* [40%, 66.7%] is misleading
  - The overall percentage of students eating cereal is 75% which is higher than 66.7%.
- *play basketball* ⇒ *not eat cereal* [20%, 33.3%] is more accurate, although with lower support and confidence
- Measure of dependent/correlated events: lift

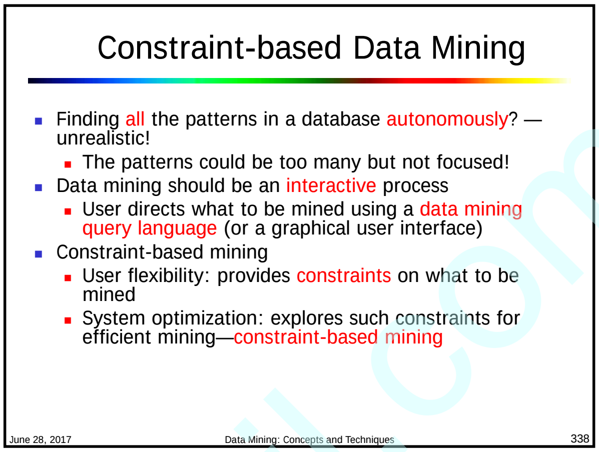
$$corr_{A,B} = \frac{P(A \cup B)}{P(A)P(B)}$$

	Basketball	Not basketball	Sum (row)
Cereal	2000	1750	3750
Not cereal	1000	250	1250
Sum(col.)	3000	2000	5000



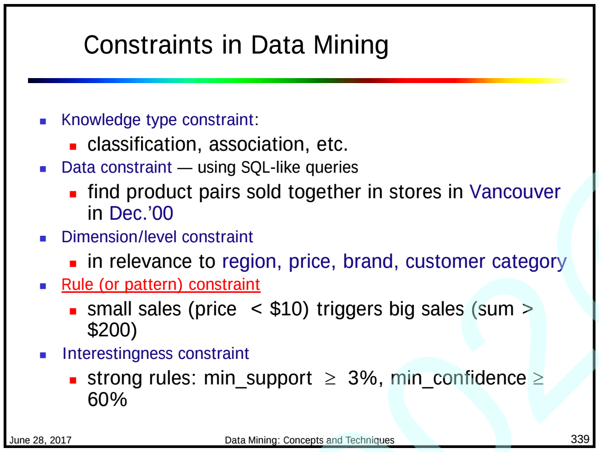
Chapter 6: Mining Association Rules in Large Databases

- Association rule mining
- Algorithms for scalable mining of (single-dimensional Boolean) association rules in transactional databases
- Mining various kinds of association/correlation rules
- Constraint-based association mining
- Sequential pattern mining
- Applications/extensions of frequent pattern mining
- Summary



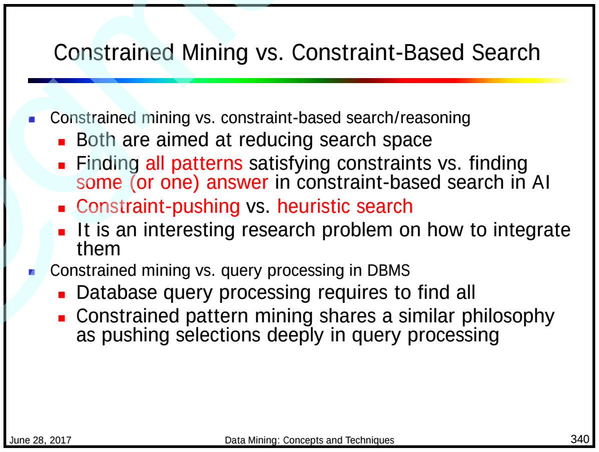
Constraint-based Data Mining

- Finding all the patterns in a database autonomously? — unrealistic!
  - The patterns could be too many but not focused!
- Data mining should be an interactive process
  - User directs what to be mined using a data mining query language (or a graphical user interface)
- Constraint-based mining
  - User flexibility: provides constraints on what to be mined
  - System optimization: explores such constraints for efficient mining—constraint-based mining



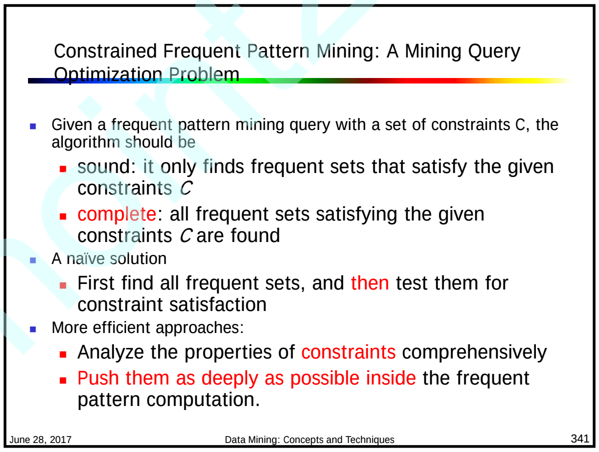
Constraints in Data Mining

- Knowledge type constraint:
  - classification, association, etc.
- Data constraint — using SQL-like queries
  - find product pairs sold together in stores in Vancouver in Dec.'00
- Dimension/level constraint
  - in relevance to region, price, brand, customer category
- Rule (or pattern) constraint
  - small sales (price < \$10) triggers big sales (sum > \$200)
- Interestingness constraint
  - strong rules: min_support ≥ 3%, min_confidence ≥ 60%



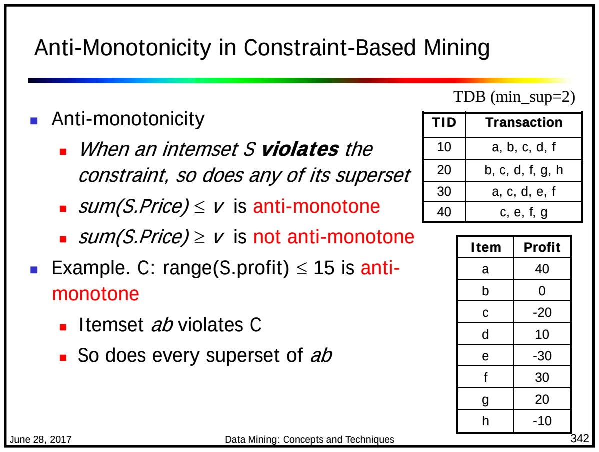
Constrained Mining vs. Constraint-Based Search

- Constrained mining vs. constraint-based search/reasoning
  - Both are aimed at reducing search space
  - Finding all patterns satisfying constraints vs. finding some (or one) answer in constraint-based search in AI
  - Constraint-pushing vs. heuristic search
  - It is an interesting research problem on how to integrate them
- Constrained mining vs. query processing in DBMS
  - Database query processing requires to find all
  - Constrained pattern mining shares a similar philosophy as pushing selections deeply in query processing



Constrained Frequent Pattern Mining: A Mining Query Optimization Problem

- Given a frequent pattern mining query with a set of constraints C, the algorithm should be
  - sound: it only finds frequent sets that satisfy the given constraints *C*
  - complete: all frequent sets satisfying the given constraints *C* are found
- A naïve solution
  - First find all frequent sets, and then test them for constraint satisfaction
- More efficient approaches:
  - Analyze the properties of constraints comprehensively
  - Push them as deeply as possible inside the frequent pattern computation.



Anti-Monotonicity in Constraint-Based Mining

- Anti-monotonicity
  - *When an intemset S **violates** the constraint, so does any of its superset*
  - *sum(S.Price)* ≤ *v* is anti-monotone
  - *sum(S.Price)* ≥ *v* is not anti-monotone
- Example. C: range(S.profit) ≤ 15 is anti-monotone
  - Itemset *ab* violates C
  - So does every superset of *ab*

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

Which Constraints Are Anti-Monotone?

Constraint	Antimonotone
v ∈ S	No
S ⊇ V	no
S ⊆ V	yes
min(S) ≤ v	no
min(S) ≥ v	yes
max(S) ≤ v	yes
max(S) ≥ v	no
count(S) ≤ v	yes
count(S) ≥ v	no
sum(S) ≤ v (a ∈ S, a ≥ 0)	yes
sum(S) ≥ v (a ∈ S, a ≥ 0)	no
range(S) ≤ v	yes
range(S) ≥ v	no
avg(S) θ v, θ ∈ { =, ≤, ≥ }	convertible
support(S) ≥ ξ	yes
support(S) ≤ ξ	no

Monotonicity in Constraint-Based Mining

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

- Monotonicity
  - *When an intemset S* ***satisfies*** *the constraint, so does any of its superset*
  - *sum(S.Price)* ≥ *v* is monotone
  - *min(S.Price)* ≤ *v* is monotone
- Example. C: range(S.profit) ≥ 15
  - Itemset *ab* satisfies C
  - So does every superset of *ab*

Which Constraints Are Monotone?

Constraint	Monotone
v ∈ S	yes
S ⊇ V	yes
S ⊆ V	no
min(S) ≤ v	yes
min(S) ≥ v	no
max(S) ≤ v	no
max(S) ≥ v	yes
count(S) ≤ v	no
count(S) ≥ v	yes
sum(S) ≤ v (a ∈ S, a ≥ 0)	no
sum(S) ≥ v (a ∈ S, a ≥ 0)	yes
range(S) ≤ v	no
range(S) ≥ v	yes
avg(S) θ v, θ ∈ { =, ≤, ≥ }	convertible
support(S) ≥ ξ	no
support(S) ≤ ξ	yes

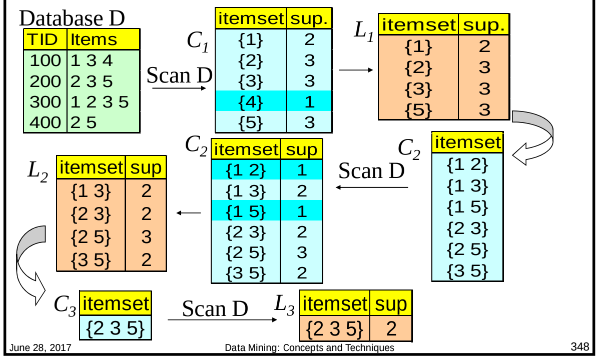
Succinctness

- Succinctness:
  - Given $A_{1,}$ the set of items satisfying a succinctness constraint *C*, then any set *S* satisfying *C* is based on A_1, i.e., *S* contains a subset belonging to A_1
  - Idea: Without looking at the transaction database, whether an itemset *S* satisfies constraint C can be determined based on the selection of items
  - *min*(*S.Price*) ≤ *v* is succinct
  - *sum*(*S.Price*) ≥ *v* is not succinct
- Optimization: If *C* is succinct, *C* is pre-counting pushable

Which Constraints Are Succinct?

Constraint	Succinct
v ∈ S	yes
S ⊇ V	yes
S ⊆ V	yes
min(S) ≤ v	yes
min(S) ≥ v	yes
max(S) ≤ v	yes
max(S) ≥ v	yes
count(S) ≤ v	weakly
count(S) ≥ v	weakly
sum(S) ≤ v (a ∈ S, a ≥ 0)	no
sum(S) ≥ v (a ∈ S, a ≥ 0)	no
range(S) ≤ v	no
range(S) ≥ v	no
avg(S) θ v, θ ∈ { =, ≤, ≥ }	no
support(S) ≥ ξ	no
support(S) ≤ ξ	no

The Apriori Algorithm — Example



Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

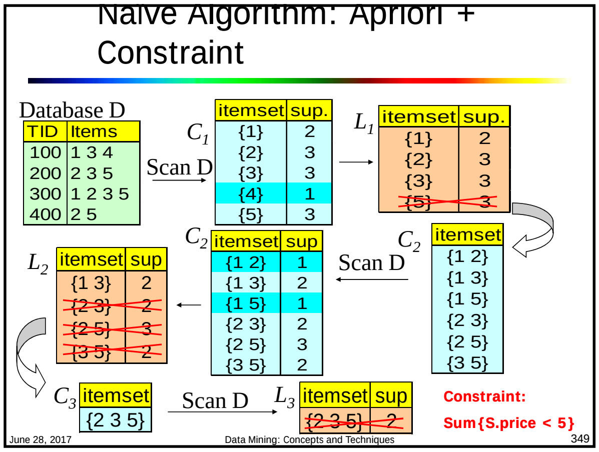
C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2



Naïve Algorithm: Apriori + Constraint

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
~~{5}~~	~~3~~

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

C_3

itemset
{2 3 5}

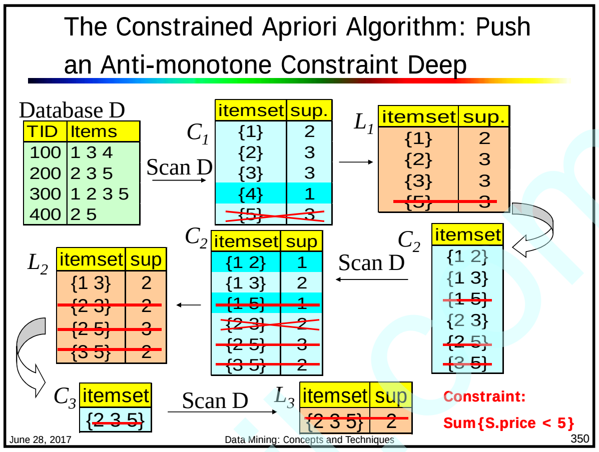
Scan D

L_3

itemset	sup
~~{2 3 5}~~	~~2~~

Constraint:

Sum{S.price < 5}



The Constrained Apriori Algorithm: Push an Anti-monotone Constraint Deep

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
~~{5}~~	~~3~~

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
~~{5}~~	~~3~~

C_2

itemset
{1 2}
{1 3}
~~{1 5}~~
{2 3}
~~{2 5}~~
~~{3 5}~~

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
~~{1 5}~~	~~1~~
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

L_2

itemset	sup
{1 3}	2
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

C_3

itemset
~~{2 3 5}~~

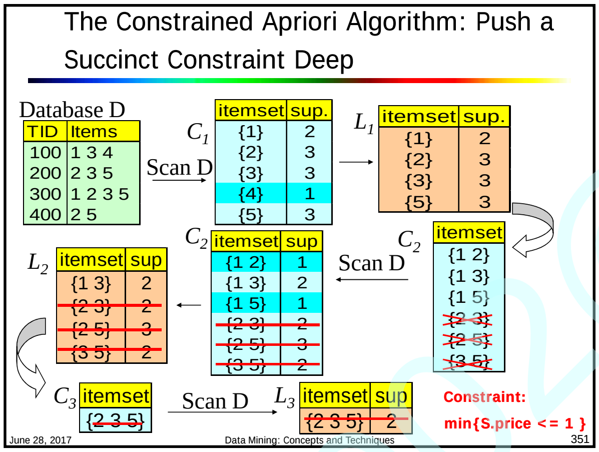
Scan D

L_3

itemset	sup
~~{2 3 5}~~	~~2~~

Constraint:

Sum{S.price < 5}



The Constrained Apriori Algorithm: Push a Succinct Constraint Deep

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
~~{2 3}~~
~~{2 5}~~
~~{3 5}~~

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

L_2

itemset	sup
{1 3}	2
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

C_3

itemset
~~{2 3 5}~~

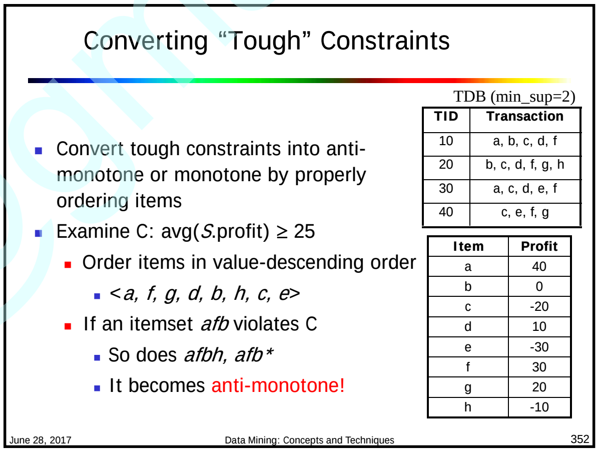
Scan D

L_3

itemset	sup
~~{2 3 5}~~	~~2~~

Constraint:

min{S.price <= 1 }



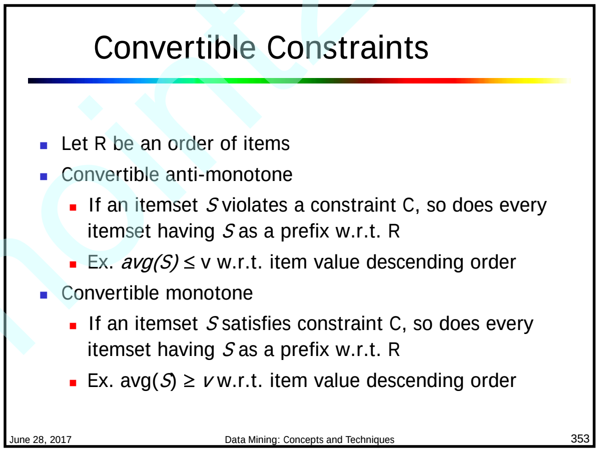
Converting "Tough" Constraints

- Convert tough constraints into anti-monotone or monotone by properly ordering items
- Examine C: avg(*S*.profit) ≥ 25
  - Order items in value-descending order
    - *<a, f, g, d, b, h, c, e>*
  - If an itemset *afb* violates C
    - So does *afbh, afb**
    - It becomes anti-monotone!

TDB (min_sup=2)

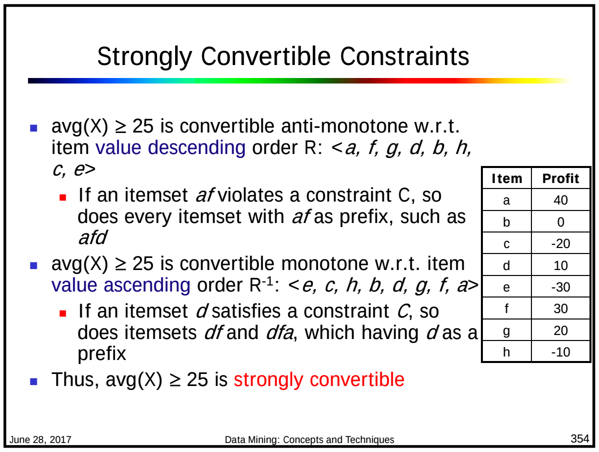
TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10



Convertible Constraints

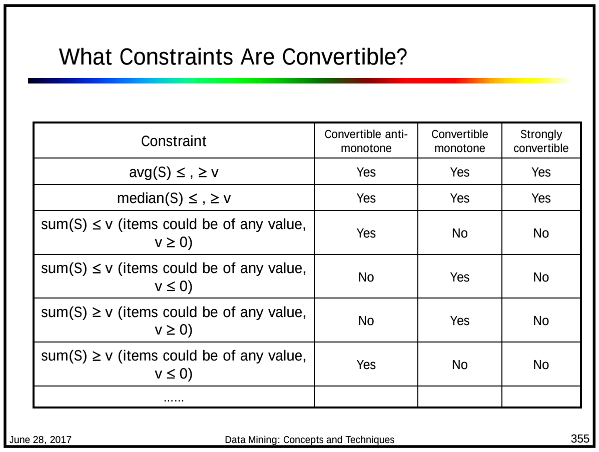
- Let R be an order of items
- Convertible anti-monotone
  - If an itemset *S* violates a constraint C, so does every itemset having *S* as a prefix w.r.t. R
  - Ex. *avg(S)* ≤ v w.r.t. item value descending order
- Convertible monotone
  - If an itemset *S* satisfies constraint C, so does every itemset having *S* as a prefix w.r.t. R
  - Ex. avg(*S*) ≥ *v* w.r.t. item value descending order



Strongly Convertible Constraints

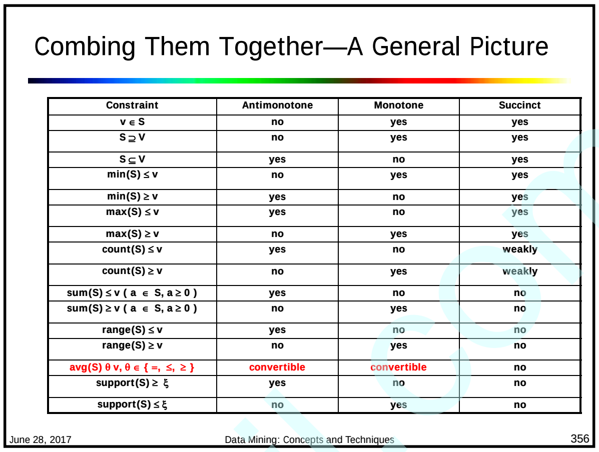
- avg(X) ≥ 25 is convertible anti-monotone w.r.t. item value descending order R: *<a, f, g, d, b, h, c, e>*
  - If an itemset *af* violates a constraint C, so does every itemset with *af* as prefix, such as *afd*
- avg(X) ≥ 25 is convertible monotone w.r.t. item value ascending order R^{-1}: *<e, c, h, b, d, g, f, a>*
  - If an itemset *d* satisfies a constraint *C*, so does itemsets *df* and *dfa*, which having *d* as a prefix
- Thus, avg(X) ≥ 25 is strongly convertible

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10



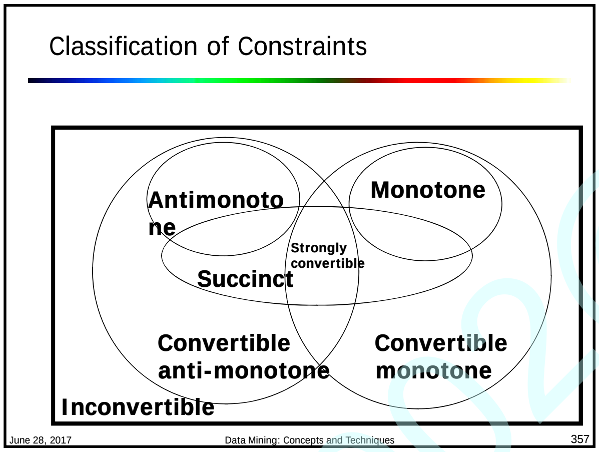
What Constraints Are Convertible?

Constraint	Convertible anti-monotone	Convertible monotone	Strongly convertible
avg(S) ≤ , ≥ v	Yes	Yes	Yes
median(S) ≤ , ≥ v	Yes	Yes	Yes
sum(S) ≤ v (items could be of any value, v ≥ 0)	Yes	No	No
sum(S) ≤ v (items could be of any value, v ≤ 0)	No	Yes	No
sum(S) ≥ v (items could be of any value, v ≥ 0)	No	Yes	No
sum(S) ≥ v (items could be of any value, v ≤ 0)	Yes	No	No
……			



Combing Them Together—A General Picture

Constraint	Antimonotone	Monotone	Succinct
v ∈ S	no	yes	yes
S ⊇ V	no	yes	yes
S ⊆ V	yes	no	yes
min(S) ≤ v	no	yes	yes
min(S) ≥ v	yes	no	yes
max(S) ≤ v	yes	no	yes
max(S) ≥ v	no	yes	yes
count(S) ≤ v	yes	no	weakly
count(S) ≥ v	no	yes	weakly
sum(S) ≤ v (a ∈ S, a ≥ 0)	yes	no	no
sum(S) ≥ v (a ∈ S, a ≥ 0)	no	yes	no
range(S) ≤ v	yes	no	no
range(S) ≥ v	no	yes	no
avg(S) θ v, θ ∈ { =, ≤, ≥ }	convertible	convertible	no
support(S) ≥ ξ	yes	no	no
support(S) ≤ ξ	no	yes	no



Classification of Constraints

Antimonotone

Monotone

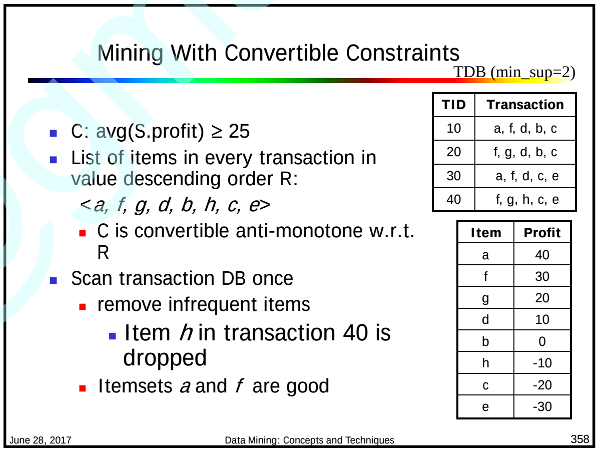
Strongly convertible

Succinct

Convertible anti-monotone

Convertible monotone

Inconvertible



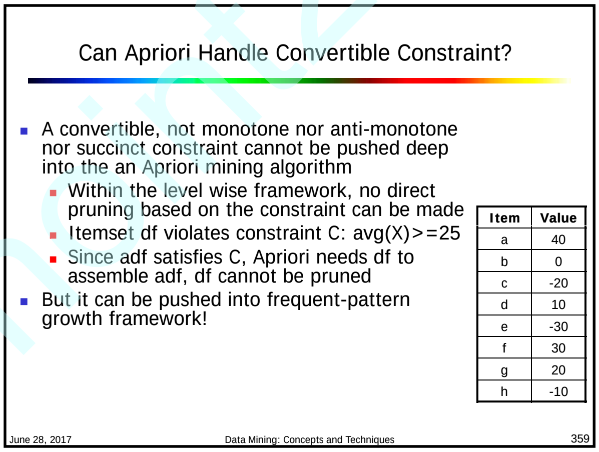
Mining With Convertible Constraints

- C: avg(S.profit) ≥ 25
- List of items in every transaction in value descending order R:
  <a, f, g, d, b, h, c, e>
  - C is convertible anti-monotone w.r.t. R
- Scan transaction DB once
  - remove infrequent items
    - Item *h* in transaction 40 is dropped
  - Itemsets *a* and *f* are good

TDB (min_sup=2)

TID	Transaction
10	a, f, d, b, c
20	f, g, d, b, c
30	a, f, d, c, e
40	f, g, h, c, e

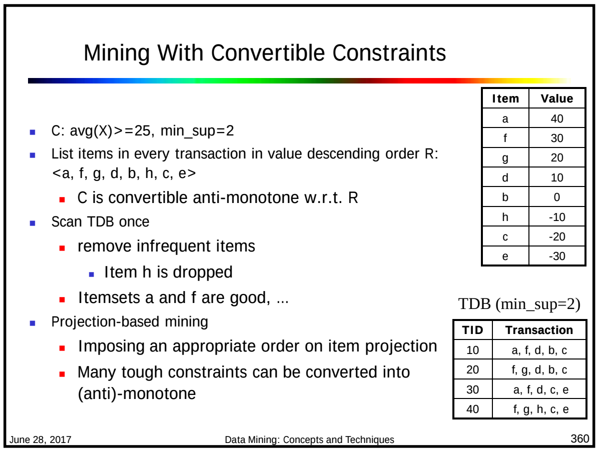
Item	Profit
a	40
f	30
g	20
d	10
b	0
h	-10
c	-20
e	-30



Can Apriori Handle Convertible Constraint?

- A convertible, not monotone nor anti-monotone nor succinct constraint cannot be pushed deep into the an Apriori mining algorithm
  - Within the level wise framework, no direct pruning based on the constraint can be made
  - Itemset df violates constraint C: avg(X)>=25
  - Since adf satisfies C, Apriori needs df to assemble adf, df cannot be pruned
- But it can be pushed into frequent-pattern growth framework!

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10



Mining With Convertible Constraints

- C: avg(X)>=25, min_sup=2
- List items in every transaction in value descending order R: <a, f, g, d, b, h, c, e>
  - C is convertible anti-monotone w.r.t. R
- Scan TDB once
  - remove infrequent items
    - Item h is dropped
  - Itemsets a and f are good, …
- Projection-based mining
  - Imposing an appropriate order on item projection
  - Many tough constraints can be converted into (anti)-monotone

Item	Value
a	40
f	30
g	20
d	10
b	0
h	-10
c	-20
e	-30

TDB (min_sup=2)

TID	Transaction
10	a, f, d, b, c
20	f, g, d, b, c
30	a, f, d, c, e
40	f, g, h, c, e

Handling Multiple Constraints

- Different constraints may require different or even conflicting item-ordering
- If there exists an order *R* s.t. both C_1 and C_2 are convertible w.r.t. *R*, then there is no conflict between the two convertible constraints
- If there exists conflict on order of items
  - Try to satisfy one constraint first
  - Then using the order for the other constraint to mine frequent itemsets in the corresponding projected database

Chapter 6: Mining Association Rules in Large Databases

- Association rule mining
- Algorithms for scalable mining of (single-dimensional Boolean) association rules in transactional databases
- Mining various kinds of association/correlation rules
- Constraint-based association mining
- Sequential pattern mining
- Applications/extensions of frequent pattern mining
- Summary

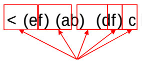
Sequence Databases and Sequential Pattern Analysis

- Transaction databases, time-series databases vs. sequence databases
- Frequent patterns vs. (frequent) sequential patterns
- Applications of sequential pattern mining
  - Customer shopping sequences:
    - First buy computer, then CD-ROM, and then digital camera, within 3 months.
  - Medical treatment, natural disasters (e.g., earthquakes), science & engineering processes, stocks and markets, etc.
  - Telephone calling patterns, Weblog click streams
  - DNA sequences and gene structures

What Is Sequential Pattern Mining?

- Given a set of sequences, find the complete set of *frequent* subsequences

A *sequence* : < (ef) (ab) (df) c b >



A *sequence database*

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

An element may contain a set of items. Items within an element are unordered and we list them alphabetically.

<a(bc)dc> is a *subsequence* of <a(abc)(ac)d(cf)>

Given *support threshold* *min_sup* =2, <(ab)c> is a *sequential pattern*

Challenges on Sequential Pattern Mining

- A huge number of possible sequential patterns are hidden in databases
- A mining algorithm should
  - find the complete set of patterns, when possible, satisfying the minimum support (frequency) threshold
  - be highly efficient, scalable, involving only a small number of database scans
  - be able to incorporate various kinds of user-specific constraints

Studies on Sequential Pattern Mining

- Concept introduction and an initial Apriori-like algorithm
  - R. Agrawal & R. Srikant. "Mining sequential patterns," ICDE'95
- GSP—An Apriori-based, influential mining method (developed at IBM Almaden)
  - R. Srikant & R. Agrawal. "Mining sequential patterns: Generalizations and performance improvements," EDBT'96
- From sequential patterns to episodes (Apriori-like + constraints)
  - H. Mannila, H. Toivonen & A.I. Verkamo. "Discovery of frequent episodes in event sequences," Data Mining and Knowledge Discovery, 1997
- Mining sequential patterns with constraints
  - M.N. Garofalakis, R. Rastogi, K. Shim: SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. VLDB 1999

A Basic Property of Sequential Patterns: Apriori

- A basic property: Apriori (Agrawal & Sirkant'94)
  - If a sequence S is not frequent
  - Then none of the super-sequences of S is frequent
  - E.g, <hb> is infrequent → so do <hab> and <(ah)b>

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Given *support threshold*
min_sup =2

GSP—A Generalized Sequential Pattern Mining Algorithm

- GSP (Generalized Sequential Pattern) mining algorithm
  - proposed by Agrawal and Srikant, EDBT'96
- Outline of the method
  - Initially, every item in DB is a candidate of length-1
  - for each level (i.e., sequences of length-k) do
    - scan database to collect support count for each candidate sequence
    - generate candidate length-(k+1) sequences from length-k frequent sequences using Apriori
  - repeat until no frequent sequence or no candidate can be found
- Major strength: Candidate pruning by Apriori

Finding Length-1 Sequential Patterns

- Examine GSP using an example
- Initial candidates: all singleton sequences
  - <a>, <b>, <c>, <d>, <e>, <f>, <g>, <h>
- Scan database once, count support for candidates

Cand	Sup
<a>	**3**
<b>	**5**
<c>	**4**
<d>	**3**
<e>	**3**
<f>	**2**
~~<g>~~	1
~~<h>~~	1

min_sup =2

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

Generating Length-2 Candidates

51 length-2 Candidates

	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

	<a>	<b>	<c>	<d>	<e>	<f>
<a>		<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
<b>			<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>				<(cd)>	<(ce)>	<(cf)>
<d>					<(de)>	<(df)>
<e>						<(ef)>
<f>						

Without Apriori property, 8*8+8*7/2=92 candidates

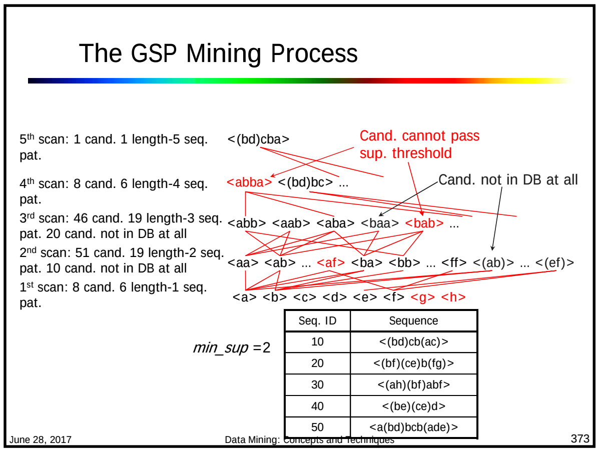
Apriori prunes 44.57% candidates

Finding Length-2 Sequential Patterns

- Scan database one more time, collect support count for each length-2 candidate
- There are 19 length-2 candidates which pass the minimum support threshold
  - They are length-2 sequential patterns

Generating Length-3 Candidates and Finding Length-3 Patterns

- Generate Length-3 Candidates
  - Self-join length-2 sequential patterns
    - Based on the Apriori property
    - <ab>, <aa> and <ba> are all length-2 sequential patterns → <aba> is a length-3 candidate
    - <(bd)>, <bb> and <db> are all length-2 sequential patterns → <(bd)b> is a length-3 candidate
  - 46 candidates are generated
- Find Length-3 Sequential Patterns
  - Scan database once more, collect support counts for candidates
  - 19 out of 46 candidates pass support threshold



The GSP Mining Process

5th scan: 1 cand. 1 length-5 seq. pat.

4th scan: 8 cand. 6 length-4 seq. pat.

3rd scan: 46 cand. 19 length-3 seq. pat. 20 cand. not in DB at all

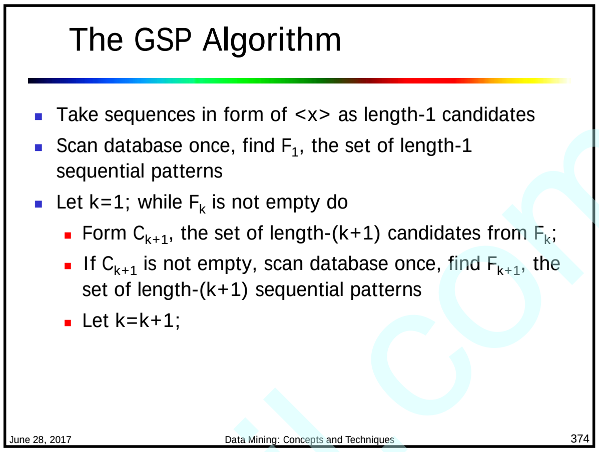
2nd scan: 51 cand. 19 length-2 seq. pat. 10 cand. not in DB at all

1st scan: 8 cand. 6 length-1 seq. pat.

<(bd)cba>
Cand. cannot pass sup. threshold
<abba> <(bd)bc> ...
Cand. not in DB at all
<abb> <aab> <aba> <baa> <bab> ...
<aa> <ab> ... <af> <ba> <bb> ... <ff> <(ab)> ... <(ef)>
<a> <b> <c> <d> <e> <f> <g> <h>

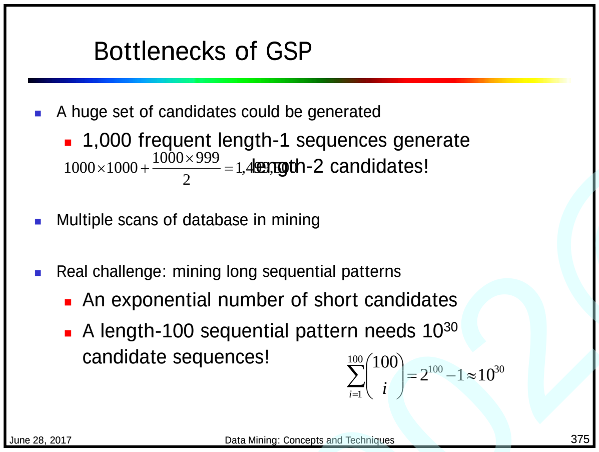
min_sup =2

Seq. ID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>



The GSP Algorithm

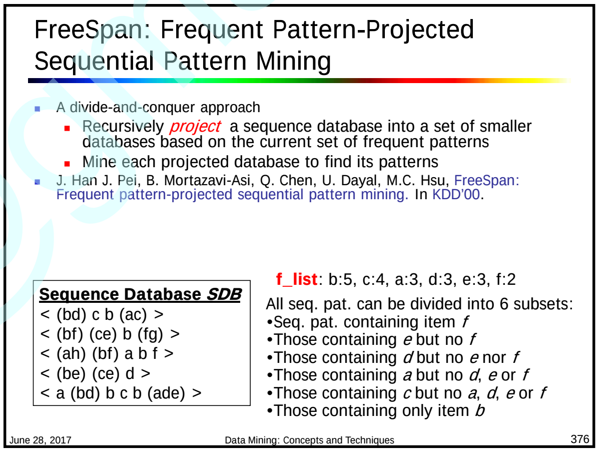
- Take sequences in form of <x> as length-1 candidates
- Scan database once, find F_1, the set of length-1 sequential patterns
- Let k=1; while F_k is not empty do
  - Form C_{k+1}, the set of length-(k+1) candidates from F_k;
  - If C_{k+1} is not empty, scan database once, find F_{k+1}, the set of length-(k+1) sequential patterns
  - Let k=k+1;



Bottlenecks of GSP

- A huge set of candidates could be generated
  - 1,000 frequent length-1 sequences generate $1000\times1000+\frac{1000\times999}{2}=1,499,500$ length-2 candidates!
- Multiple scans of database in mining
- Real challenge: mining long sequential patterns
  - An exponential number of short candidates
  - A length-100 sequential pattern needs 10^{30} candidate sequences!

$$\sum_{i=1}^{100}\binom{100}{i}=2^{100}-1\approx10^{30}$$



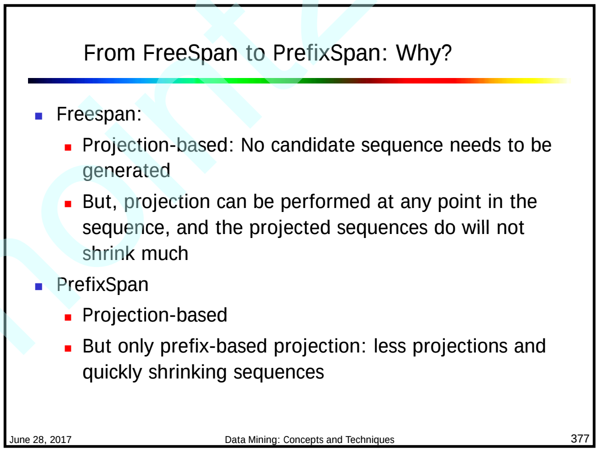
FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining

- A divide-and-conquer approach
  - Recursively *project* a sequence database into a set of smaller databases based on the current set of frequent patterns
  - Mine each projected database to find its patterns
- J. Han J. Pei, B. Mortazavi-Asi, Q. Chen, U. Dayal, M.C. Hsu, FreeSpan: Frequent pattern-projected sequential pattern mining. In KDD'00.

Sequence Database *SDB*
< (bd) c b (ac) >
< (bf) (ce) b (fg) >
< (ah) (bf) a b f >
< (be) (ce) d >
< a (bd) b c b (ade) >

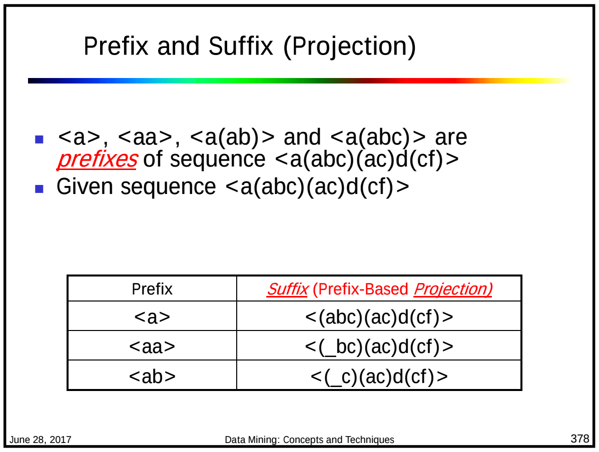
f_list: b:5, c:4, a:3, d:3, e:3, f:2

All seq. pat. can be divided into 6 subsets:
•Seq. pat. containing item *f*
•Those containing *e* but no *f*
•Those containing *d* but no *e* nor *f*
•Those containing *a* but no *d*, *e* or *f*
•Those containing *c* but no *a*, *d*, *e* or *f*
•Those containing only item *b*



From FreeSpan to PrefixSpan: Why?

- Freespan:
  - Projection-based: No candidate sequence needs to be generated
  - But, projection can be performed at any point in the sequence, and the projected sequences do will not shrink much
- PrefixSpan
  - Projection-based
  - But only prefix-based projection: less projections and quickly shrinking sequences



Prefix and Suffix (Projection)

- <a>, <aa>, <a(ab)> and <a(abc)> are *prefixes* of sequence <a(abc)(ac)d(cf)>
- Given sequence <a(abc)(ac)d(cf)>

Prefix	*Suffix* (Prefix-Based *Projection)*
<a>	<(abc)(ac)d(cf)>
<aa>	<(_bc)(ac)d(cf)>
<ab>	<(_c)(ac)d(cf)>

Mining Sequential Patterns by Prefix Projections

- Step 1: find length-1 sequential patterns
  - <a>, <b>, <c>, <d>, <e>, <f>
- Step 2: divide search space. The complete set of seq. pat. can be partitioned into 6 subsets:
  - The ones having prefix <a>;
  - The ones having prefix <b>;
  - …
  - The ones having prefix <f>

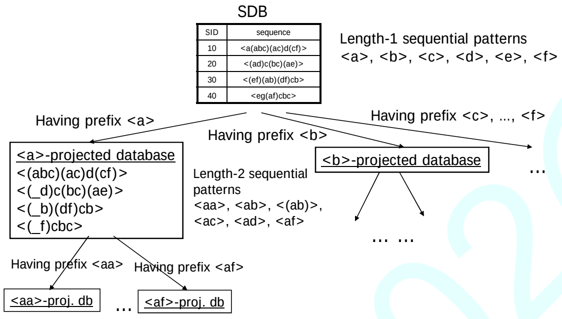
SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

Finding Seq. Patterns with Prefix <a>

- Only need to consider projections w.r.t. <a>
  - <a>-projected database: <(abc)(ac)d(cf)>, <(_d)c(bc)(ae)>, <(_b)(df)cb>, <(_f)cbc>
- Find all the length-2 seq. pat. Having prefix <a>: <aa>, <ab>, <(ab)>, <ac>, <ad>, <af>
  - Further partition into 6 subsets
    - Having prefix <aa>;
    - …
    - Having prefix <af>

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

Completeness of PrefixSpan



SDB

SID	sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

Length-1 sequential patterns
<a>, <b>, <c>, <d>, <e>, <f>

Having prefix <a>

<a>-projected database
<(abc)(ac)d(cf)>
<(_d)c(bc)(ae)>
<(_b)(df)cb>
<(_f)cbc>

Having prefix <b>

<b>-projected database

Having prefix <c>, ..., <f>

…

Length-2 sequential patterns
<aa>, <ab>, <(ab)>, <ac>, <ad>, <af>

… …

Having prefix <aa>

Having prefix <af>

<aa>-proj. db … <af>-proj. db

Efficiency of PrefixSpan

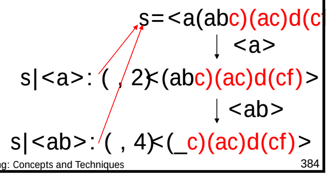
- No candidate sequence needs to be generated
- Projected databases keep shrinking
- Major cost of PrefixSpan: constructing projected databases
  - Can be improved by bi-level projections

Optimization Techniques in PrefixSpan

- Physical projection vs. pseudo-projection
  - Pseudo-projection may reduce the effort of projection when the projected database fits in main memory
- Parallel projection vs. partition projection
  - Partition projection may avoid the blowup of disk space

Speed-up by Pseudo-projection

- Major cost of PrefixSpan: projection
  - Postfixes of sequences often appear repeatedly in recursive projected databases
- When (projected) database can be held in main memory, use pointers to form projections
  - Pointer to the sequence
  - Offset of the postfix



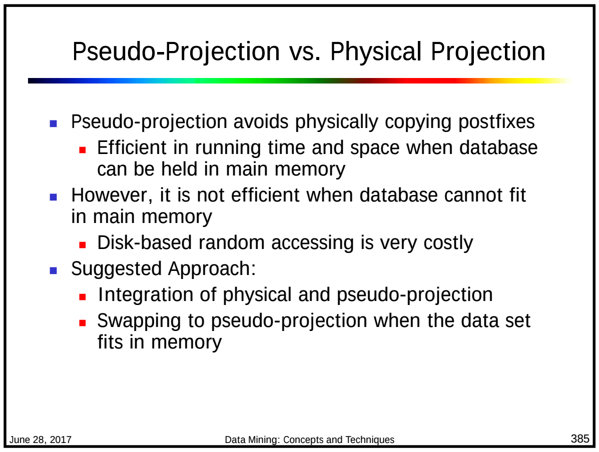
s=<a(abc)(ac)d(c

<a>

s|<a>: (, 2) <(abc)(ac)d(cf)>

<ab>

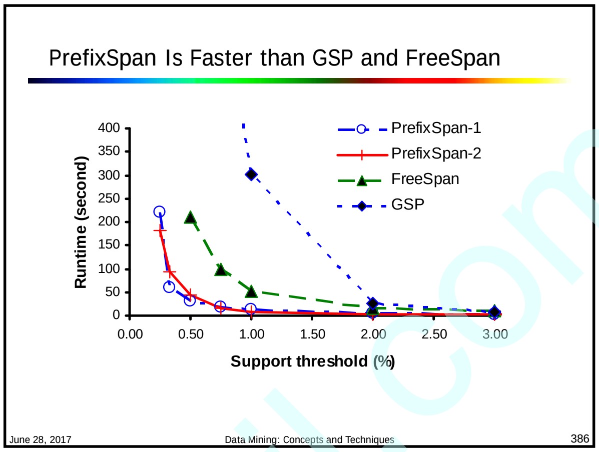
s|<ab>: (, 4) <(_c)(ac)d(cf)>



Pseudo-Projection vs. Physical Projection

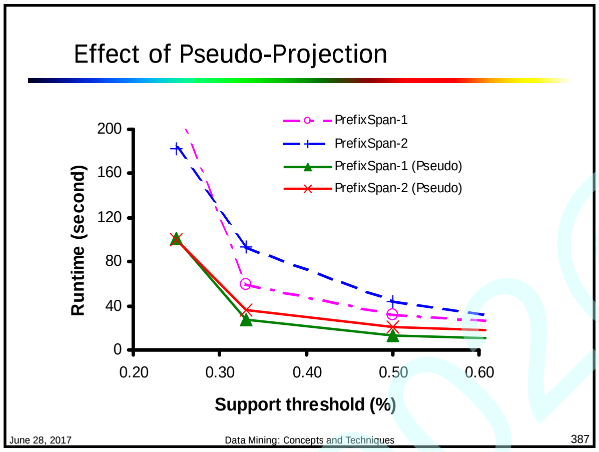
- Pseudo-projection avoids physically copying postfixes
  - Efficient in running time and space when database can be held in main memory
- However, it is not efficient when database cannot fit in main memory
  - Disk-based random accessing is very costly
- Suggested Approach:
  - Integration of physical and pseudo-projection
  - Swapping to pseudo-projection when the data set fits in memory

June 28, 2017 Data Mining: Concepts and Techniques 385



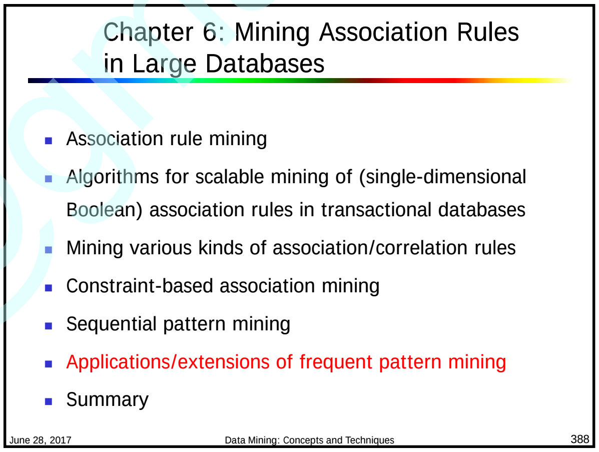
PrefixSpan Is Faster than GSP and FreeSpan

June 28, 2017 Data Mining: Concepts and Techniques 386



Effect of Pseudo-Projection

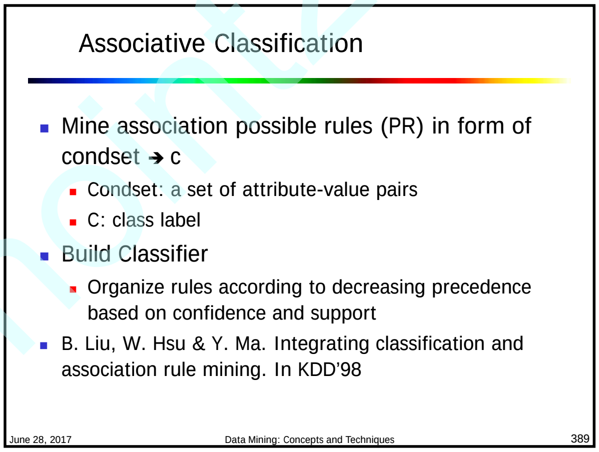
June 28, 2017 Data Mining: Concepts and Techniques 387



Chapter 6: Mining Association Rules in Large Databases

- Association rule mining
- Algorithms for scalable mining of (single-dimensional Boolean) association rules in transactional databases
- Mining various kinds of association/correlation rules
- Constraint-based association mining
- Sequential pattern mining
- Applications/extensions of frequent pattern mining
- Summary

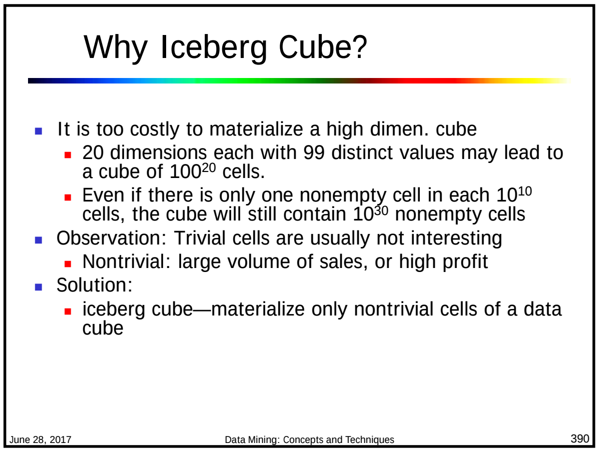
June 28, 2017 Data Mining: Concepts and Techniques 388



Associative Classification

- Mine association possible rules (PR) in form of condset → c
  - Condset: a set of attribute-value pairs
  - C: class label
- Build Classifier
  - Organize rules according to decreasing precedence based on confidence and support
- B. Liu, W. Hsu & Y. Ma. Integrating classification and association rule mining. In KDD'98

June 28, 2017 Data Mining: Concepts and Techniques 389



Why Iceberg Cube?

- It is too costly to materialize a high dimen. cube
  - 20 dimensions each with 99 distinct values may lead to a cube of 100^{20} cells.
  - Even if there is only one nonempty cell in each 10^{10} cells, the cube will still contain 10^{30} nonempty cells
- Observation: Trivial cells are usually not interesting
  - Nontrivial: large volume of sales, or high profit
- Solution:
  - iceberg cube—materialize only nontrivial cells of a data cube

June 28, 2017 Data Mining: Concepts and Techniques 390

Anti-Monotonicity in Iceberg Cubes

- If a cell *c* violates the HAVING clause, so do all more specific cells
- Example. Let Having COUNT(*)>=50
  - (*, *, Edu, 1000, 30) violates the HAVING clause
    - (Feb, *, Edu), (*, Van, Edu), (Mar, Tor, Edu): each must have count no more than 30

```
CREATE CUBE Sales_Iceberg AS
SELECT month, city, cust_grp,
        AVG(price), COUNT(*)
FROM Sales_Infor
CUBEBY month, city, cust_grp
HAVING COUNT(*)>=50
```

Month	City	Cust_grp	Prod	Cost	Price
Jan	Tor	Edu	Printer	500	485
Mar	Van	Edu	HD	540	520
…	…	…	…	…	…

Computing Iceberg Cubes Efficiently

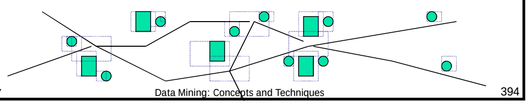
- Based on Apriori-like pruning
- BUC [Bayer & Ramakrishnan, 99]
  - bottom-up cubing, efficient bucket-sort alg.
  - Only handles anti-monotonic iceberg cubes, e.g., measures confined to count and p+_sum (e.g., price)
- Computing non-anti-monotonic iceberg cubes
  - Finding a weaker but anti-monotonic measure (e.g., avg to top-k-avg) for dynamic pruning in computation
  - Use special data structure (H-tree) and perform H-cubing (SIGMOD'01)

Spatial and Multi-Media Association: A Progressive Refinement Method

- Why progressive refinement?
  - Mining operator can be expensive or cheap, fine or rough
  - Trade speed with quality: step-by-step refinement.
- Superset coverage property:
  - Preserve all the positive answers—allow a positive false test but not a false negative test.
- Two- or multi-step mining:
  - First apply rough/cheap operator (superset coverage)
  - Then apply expensive algorithm on a substantially reduced candidate set (Koperski & Han, SSD'95).

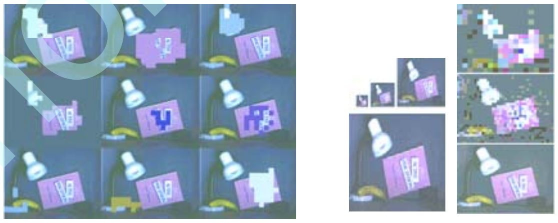
Progressive Refinement Mining of Spatial Associations

- Hierarchy of spatial relationship:
  - "g_close_to": near_by, touch, intersect, contain, etc.
  - First search for rough relationship and then refine it.
- Two-step mining of spatial association:
  - Step 1: rough spatial computation (as a filter)
    - Using MBR or R-tree for rough estimation.
  - Step2: Detailed spatial algorithm (as refinement)
    - Apply only to those objects which have passed the rough spatial association test (no less than *min_support*)



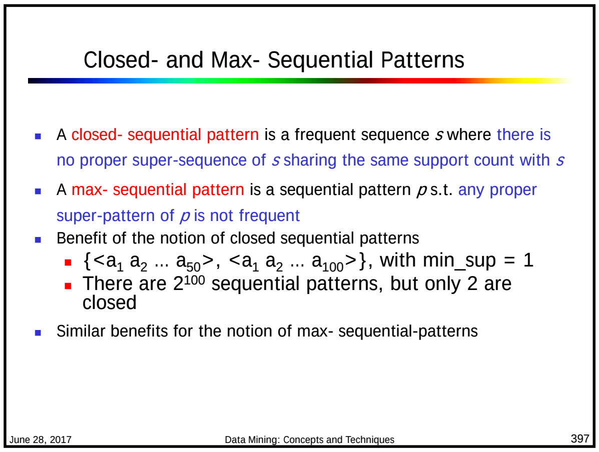
Mining Multimedia Associations

Correlations with color, spatial relationships, etc.
From coarse to Fine Resolution mining



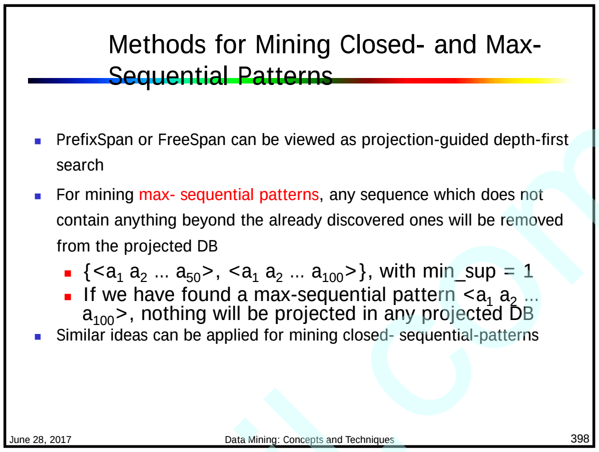
Further Evolution of PrefixSpan

- Closed- and max- sequential patterns
  - Finding only the most meaningful (longest) sequential patterns
- Constraint-based sequential pattern growth
  - Adding user-specific constraints
- From sequential patterns to structured patterns
  - Beyond sequential patterns, mining structured patterns in XML documents



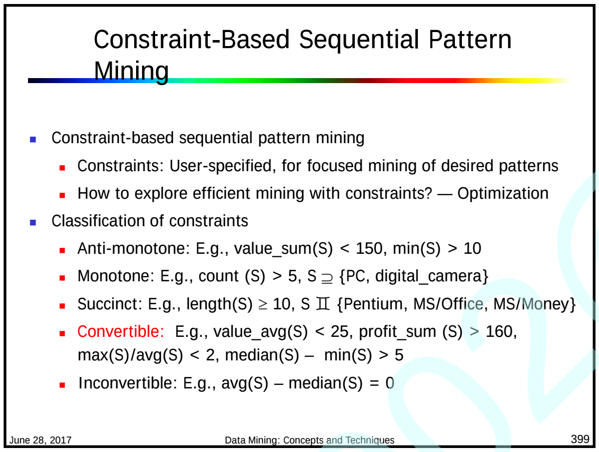
Closed- and Max- Sequential Patterns

- A closed- sequential pattern is a frequent sequence *s* where there is no proper super-sequence of *s* sharing the same support count with *s*
- A max- sequential pattern is a sequential pattern *p* s.t. any proper super-pattern of *p* is not frequent
- Benefit of the notion of closed sequential patterns
  - {<a_1 a_2 … a_{50}>, <a_1 a_2 … a_{100}>}, with min_sup = 1
  - There are 2^{100} sequential patterns, but only 2 are closed
- Similar benefits for the notion of max- sequential-patterns



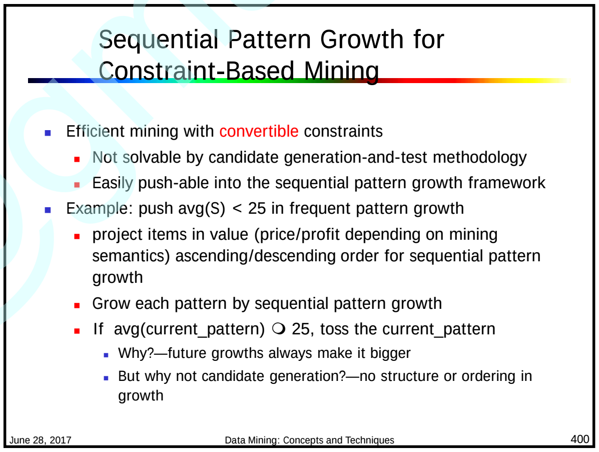
Methods for Mining Closed- and Max- Sequential Patterns

- PrefixSpan or FreeSpan can be viewed as projection-guided depth-first search
- For mining max- sequential patterns, any sequence which does not contain anything beyond the already discovered ones will be removed from the projected DB
  - {<a_1 a_2 … a_{50}>, <a_1 a_2 … a_{100}>}, with min_sup = 1
  - If we have found a max-sequential pattern <a_1 a_2 … a_{100}>, nothing will be projected in any projected DB
- Similar ideas can be applied for mining closed- sequential-patterns



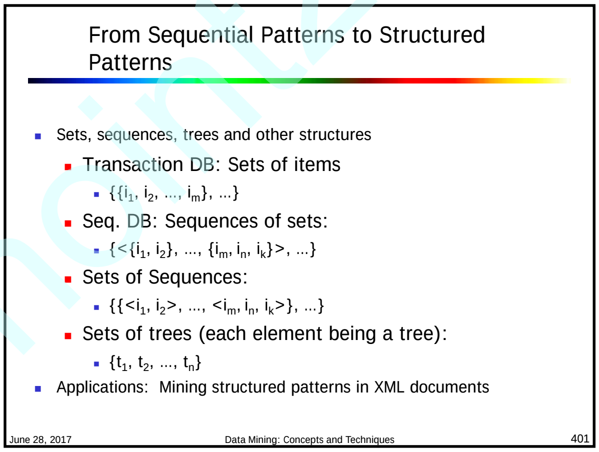
Constraint-Based Sequential Pattern Mining

- Constraint-based sequential pattern mining
  - Constraints: User-specified, for focused mining of desired patterns
  - How to explore efficient mining with constraints? — Optimization
- Classification of constraints
  - Anti-monotone: E.g., value_sum(S) < 150, min(S) > 10
  - Monotone: E.g., count (S) > 5, S ⊇ {PC, digital_camera}
  - Succinct: E.g., length(S) ≥ 10, S Ⅱ {Pentium, MS/Office, MS/Money}
  - Convertible: E.g., value_avg(S) < 25, profit_sum (S) > 160, max(S)/avg(S) < 2, median(S) – min(S) > 5
  - Inconvertible: E.g., avg(S) – median(S) = 0



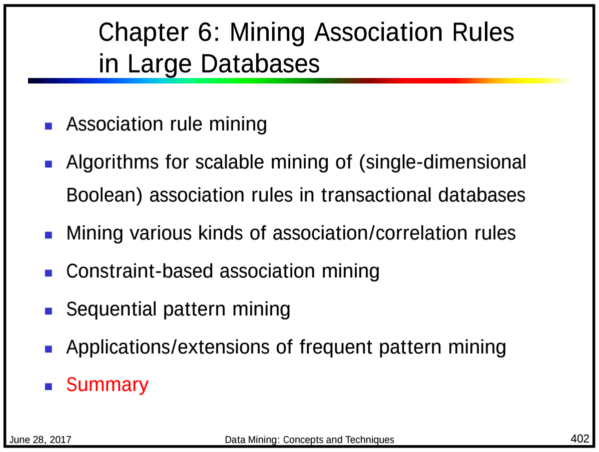
Sequential Pattern Growth for Constraint-Based Mining

- Efficient mining with convertible constraints
  - Not solvable by candidate generation-and-test methodology
  - Easily push-able into the sequential pattern growth framework
- Example: push avg(S) < 25 in frequent pattern growth
  - project items in value (price/profit depending on mining semantics) ascending/descending order for sequential pattern growth
  - Grow each pattern by sequential pattern growth
  - If avg(current_pattern) ❍ 25, toss the current_pattern
    - Why?—future growths always make it bigger
    - But why not candidate generation?—no structure or ordering in growth



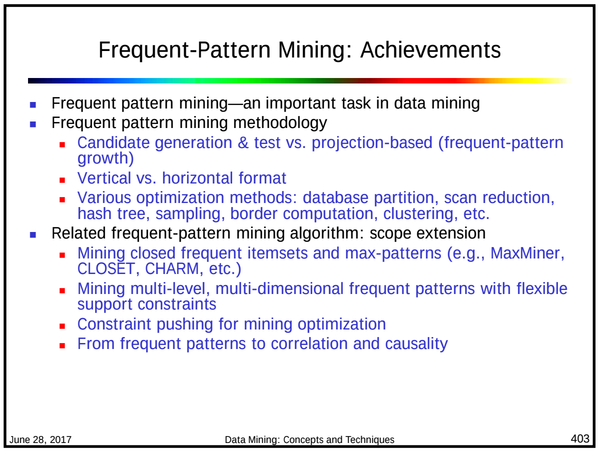
From Sequential Patterns to Structured Patterns

- Sets, sequences, trees and other structures
  - Transaction DB: Sets of items
    - {{i_1, i_2, …, i_m}, …}
  - Seq. DB: Sequences of sets:
    - {<{i_1, i_2}, …, {i_m, i_n, i_k}>, …}
  - Sets of Sequences:
    - {{<i_1, i_2>, …, <i_m, i_n, i_k>}, …}
  - Sets of trees (each element being a tree):
    - {t_1, t_2, …, t_n}
- Applications: Mining structured patterns in XML documents



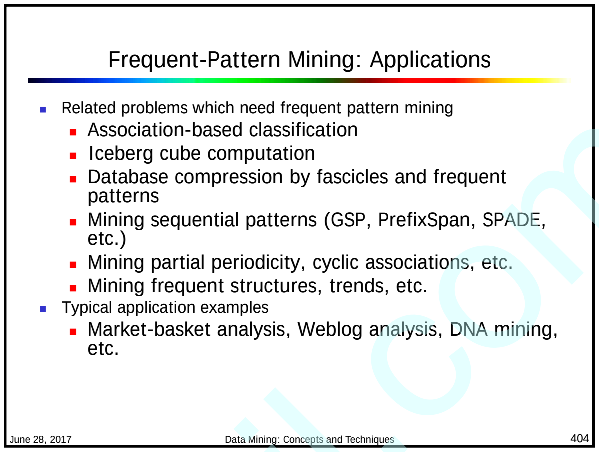
Chapter 6: Mining Association Rules in Large Databases

- Association rule mining
- Algorithms for scalable mining of (single-dimensional Boolean) association rules in transactional databases
- Mining various kinds of association/correlation rules
- Constraint-based association mining
- Sequential pattern mining
- Applications/extensions of frequent pattern mining
- Summary



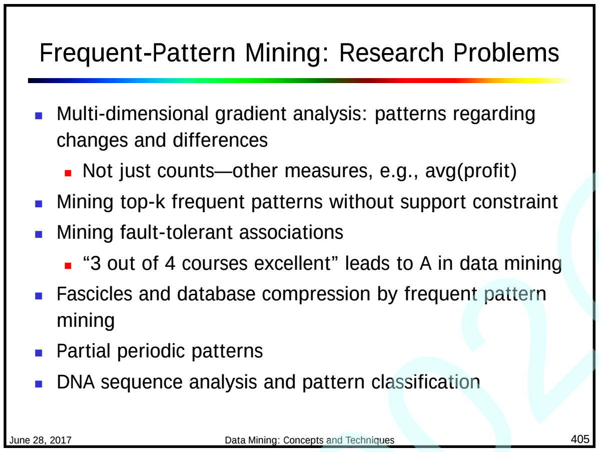
Frequent-Pattern Mining: Achievements

- Frequent pattern mining—an important task in data mining
- Frequent pattern mining methodology
  - Candidate generation & test vs. projection-based (frequent-pattern growth)
  - Vertical vs. horizontal format
  - Various optimization methods: database partition, scan reduction, hash tree, sampling, border computation, clustering, etc.
- Related frequent-pattern mining algorithm: scope extension
  - Mining closed frequent itemsets and max-patterns (e.g., MaxMiner, CLOSET, CHARM, etc.)
  - Mining multi-level, multi-dimensional frequent patterns with flexible support constraints
  - Constraint pushing for mining optimization
  - From frequent patterns to correlation and causality



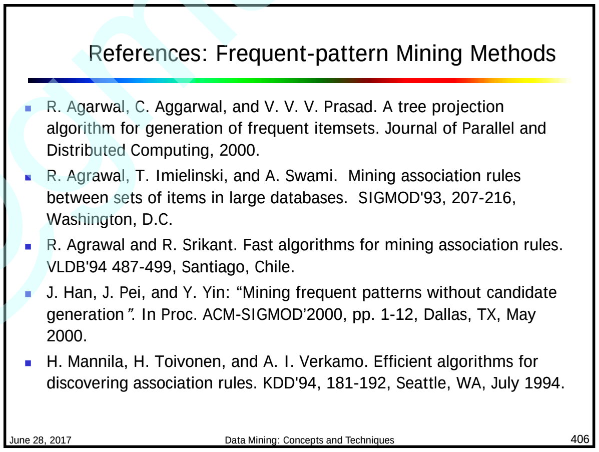
Frequent-Pattern Mining: Applications

- Related problems which need frequent pattern mining
  - Association-based classification
  - Iceberg cube computation
  - Database compression by fascicles and frequent patterns
  - Mining sequential patterns (GSP, PrefixSpan, SPADE, etc.)
  - Mining partial periodicity, cyclic associations, etc.
  - Mining frequent structures, trends, etc.
- Typical application examples
  - Market-basket analysis, Weblog analysis, DNA mining, etc.



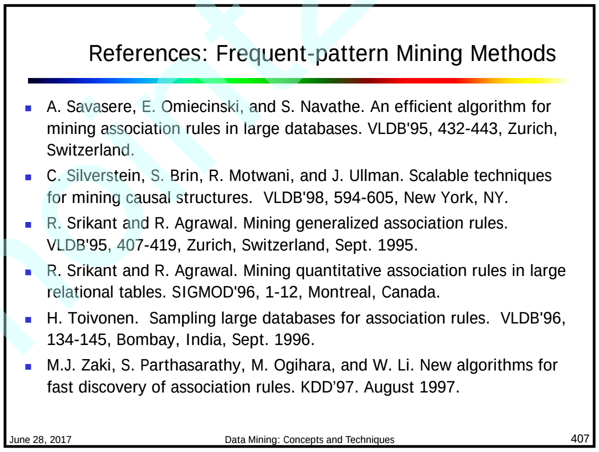
Frequent-Pattern Mining: Research Problems

- Multi-dimensional gradient analysis: patterns regarding changes and differences
  - Not just counts—other measures, e.g., avg(profit)
- Mining top-k frequent patterns without support constraint
- Mining fault-tolerant associations
  - "3 out of 4 courses excellent" leads to A in data mining
- Fascicles and database compression by frequent pattern mining
- Partial periodic patterns
- DNA sequence analysis and pattern classification



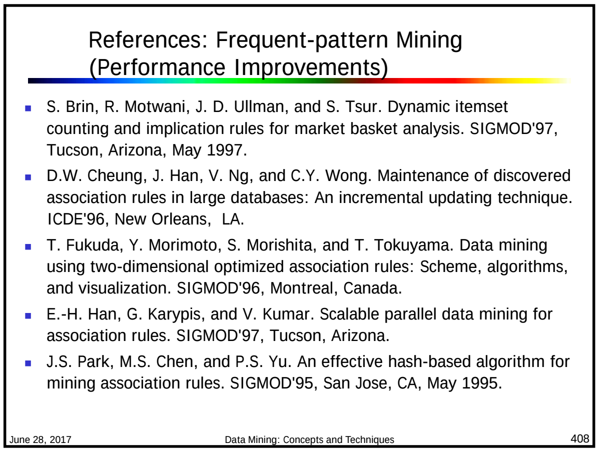
References: Frequent-pattern Mining Methods

- R. Agarwal, C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. Journal of Parallel and Distributed Computing, 2000.
- R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93, 207-216, Washington, D.C.
- R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94 487-499, Santiago, Chile.
- J. Han, J. Pei, and Y. Yin: "Mining frequent patterns without candidate generation". In Proc. ACM-SIGMOD'2000, pp. 1-12, Dallas, TX, May 2000.
- H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. KDD'94, 181-192, Seattle, WA, July 1994.



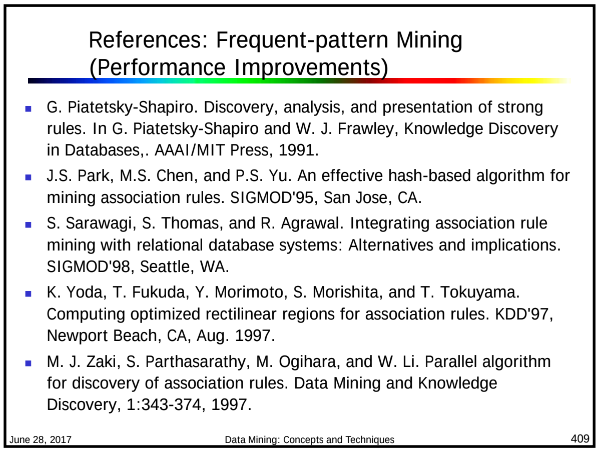
References: Frequent-pattern Mining Methods

- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. VLDB'95, 432-443, Zurich, Switzerland.
- C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. VLDB'98, 594-605, New York, NY.
- R. Srikant and R. Agrawal. Mining generalized association rules. VLDB'95, 407-419, Zurich, Switzerland, Sept. 1995.
- R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. SIGMOD'96, 1-12, Montreal, Canada.
- H. Toivonen. Sampling large databases for association rules. VLDB'96, 134-145, Bombay, India, Sept. 1996.
- M.J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. KDD'97. August 1997.



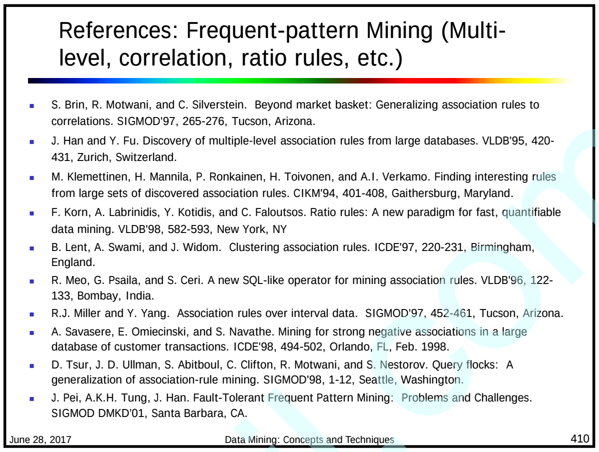
References: Frequent-pattern Mining (Performance Improvements)

- S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. SIGMOD'97, Tucson, Arizona, May 1997.
- D.W. Cheung, J. Han, V. Ng, and C.Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. ICDE'96, New Orleans, LA.
- T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. SIGMOD'96, Montreal, Canada.
- E.-H. Han, G. Karypis, and V. Kumar. Scalable parallel data mining for association rules. SIGMOD'97, Tucson, Arizona.
- J.S. Park, M.S. Chen, and P.S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95, San Jose, CA, May 1995.



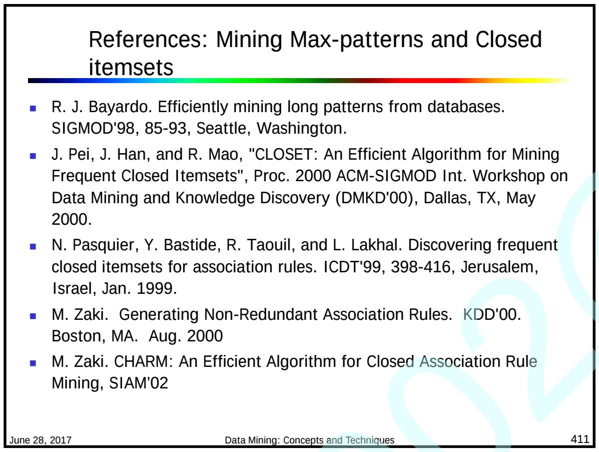
References: Frequent-pattern Mining (Performance Improvements)

- G. Piatetsky-Shapiro. Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro and W. J. Frawley, Knowledge Discovery in Databases,. AAAI/MIT Press, 1991.
- J.S. Park, M.S. Chen, and P.S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95, San Jose, CA.
- S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98, Seattle, WA.
- K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Computing optimized rectilinear regions for association rules. KDD'97, Newport Beach, CA, Aug. 1997.
- M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. Parallel algorithm for discovery of association rules. Data Mining and Knowledge Discovery, 1:343-374, 1997.



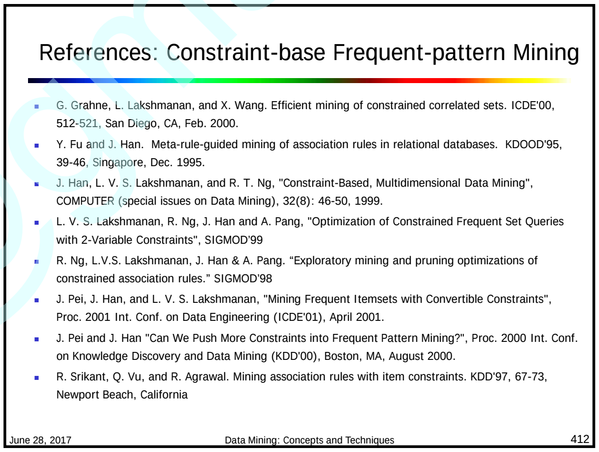
References: Frequent-pattern Mining (Multi-level, correlation, ratio rules, etc.)

- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97, 265-276, Tucson, Arizona.
- J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. VLDB'95, 420-431, Zurich, Switzerland.
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94, 401-408, Gaithersburg, Maryland.
- F. Korn, A. Labrinidis, Y. Kotidis, and C. Faloutsos. Ratio rules: A new paradigm for fast, quantifiable data mining. VLDB'98, 582-593, New York, NY
- B. Lent, A. Swami, and J. Widom. Clustering association rules. ICDE'97, 220-231, Birmingham, England.
- R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. VLDB'96, 122-133, Bombay, India.
- R.J. Miller and Y. Yang. Association rules over interval data. SIGMOD'97, 452-461, Tucson, Arizona.
- A. Savasere, E. Omiecinski, and S. Navathe. Mining for strong negative associations in a large database of customer transactions. ICDE'98, 494-502, Orlando, FL, Feb. 1998.
- D. Tsur, J. D. Ullman, S. Abitboul, C. Clifton, R. Motwani, and S. Nestorov. Query flocks: A generalization of association-rule mining. SIGMOD'98, 1-12, Seattle, Washington.
- J. Pei, A.K.H. Tung, J. Han. Fault-Tolerant Frequent Pattern Mining: Problems and Challenges. SIGMOD DMKD'01, Santa Barbara, CA.



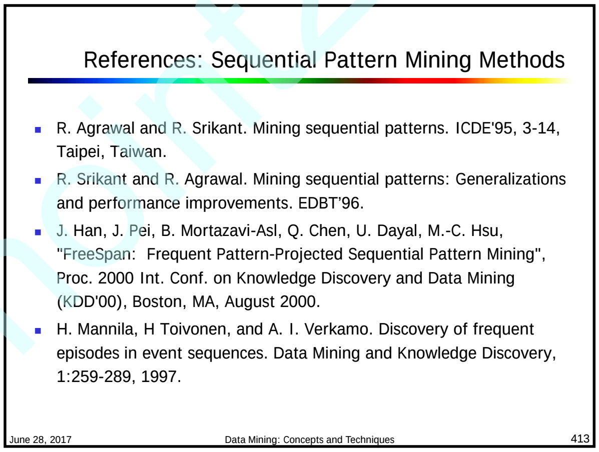
References: Mining Max-patterns and Closed itemsets

- R. J. Bayardo. Efficiently mining long patterns from databases. SIGMOD'98, 85-93, Seattle, Washington.
- J. Pei, J. Han, and R. Mao, "CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets", Proc. 2000 ACM-SIGMOD Int. Workshop on Data Mining and Knowledge Discovery (DMKD'00), Dallas, TX, May 2000.
- N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. ICDT'99, 398-416, Jerusalem, Israel, Jan. 1999.
- M. Zaki. Generating Non-Redundant Association Rules. KDD'00. Boston, MA. Aug. 2000
- M. Zaki. CHARM: An Efficient Algorithm for Closed Association Rule Mining, SIAM'02



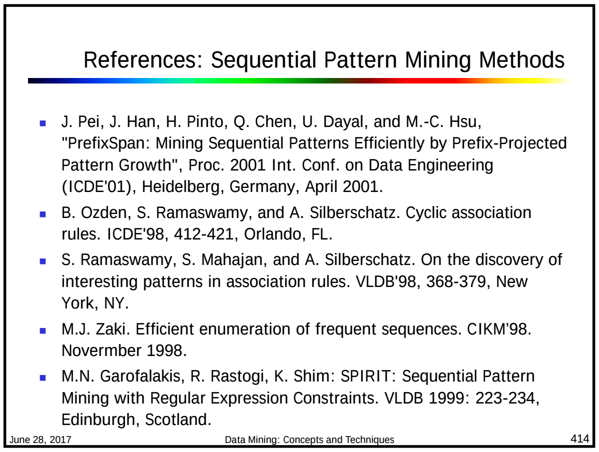
References: Constraint-base Frequent-pattern Mining

- G. Grahne, L. Lakshmanan, and X. Wang. Efficient mining of constrained correlated sets. ICDE'00, 512-521, San Diego, CA, Feb. 2000.
- Y. Fu and J. Han. Meta-rule-guided mining of association rules in relational databases. KDOOD'95, 39-46, Singapore, Dec. 1995.
- J. Han, L. V. S. Lakshmanan, and R. T. Ng, "Constraint-Based, Multidimensional Data Mining", COMPUTER (special issues on Data Mining), 32(8): 46-50, 1999.
- L. V. S. Lakshmanan, R. Ng, J. Han and A. Pang, "Optimization of Constrained Frequent Set Queries with 2-Variable Constraints", SIGMOD'99
- R. Ng, L.V.S. Lakshmanan, J. Han & A. Pang. "Exploratory mining and pruning optimizations of constrained association rules." SIGMOD'98
- J. Pei, J. Han, and L. V. S. Lakshmanan, "Mining Frequent Itemsets with Convertible Constraints", Proc. 2001 Int. Conf. on Data Engineering (ICDE'01), April 2001.
- J. Pei and J. Han "Can We Push More Constraints into Frequent Pattern Mining?", Proc. 2000 Int. Conf. on Knowledge Discovery and Data Mining (KDD'00), Boston, MA, August 2000.
- R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. KDD'97, 67-73, Newport Beach, California



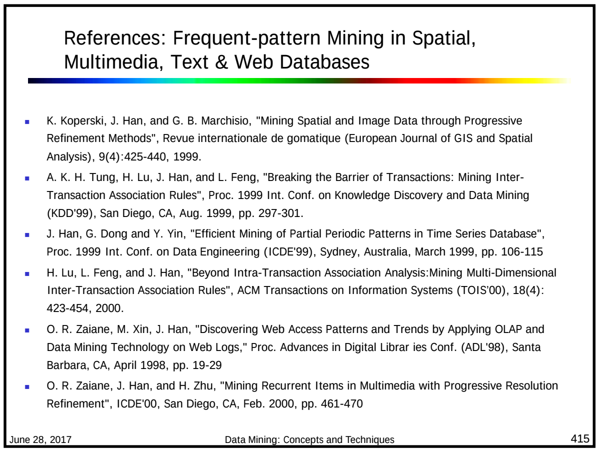
References: Sequential Pattern Mining Methods

- R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95, 3-14, Taipei, Taiwan.
- R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. EDBT'96.
- J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, M.-C. Hsu, "FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining", Proc. 2000 Int. Conf. on Knowledge Discovery and Data Mining (KDD'00), Boston, MA, August 2000.
- H. Mannila, H Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1:259-289, 1997.



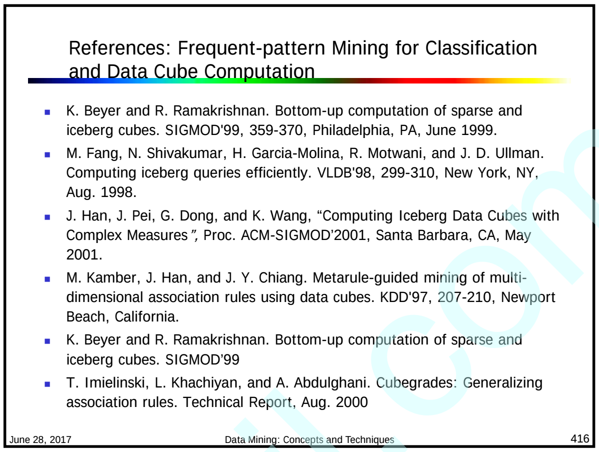
References: Sequential Pattern Mining Methods

- J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", Proc. 2001 Int. Conf. on Data Engineering (ICDE'01), Heidelberg, Germany, April 2001.
- B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. ICDE'98, 412-421, Orlando, FL.
- S. Ramaswamy, S. Mahajan, and A. Silberschatz. On the discovery of interesting patterns in association rules. VLDB'98, 368-379, New York, NY.
- M.J. Zaki. Efficient enumeration of frequent sequences. CIKM'98. Novermber 1998.
- M.N. Garofalakis, R. Rastogi, K. Shim: SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. VLDB 1999: 223-234, Edinburgh, Scotland.



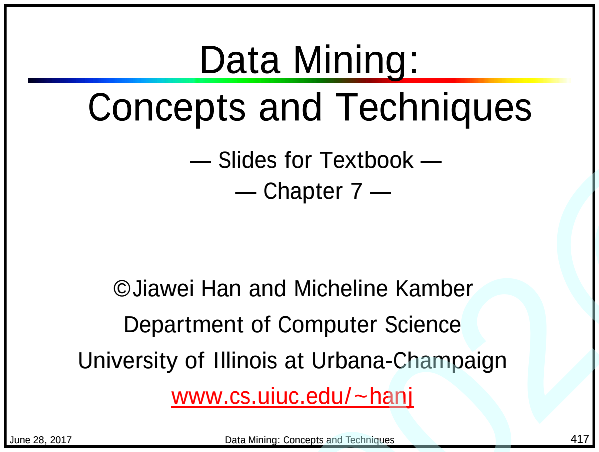
References: Frequent-pattern Mining in Spatial, Multimedia, Text & Web Databases

- K. Koperski, J. Han, and G. B. Marchisio, "Mining Spatial and Image Data through Progressive Refinement Methods", Revue internationale de gomatique (European Journal of GIS and Spatial Analysis), 9(4):425-440, 1999.
- A. K. H. Tung, H. Lu, J. Han, and L. Feng, "Breaking the Barrier of Transactions: Mining Inter-Transaction Association Rules", Proc. 1999 Int. Conf. on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA, Aug. 1999, pp. 297-301.
- J. Han, G. Dong and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database", Proc. 1999 Int. Conf. on Data Engineering (ICDE'99), Sydney, Australia, March 1999, pp. 106-115
- H. Lu, L. Feng, and J. Han, "Beyond Intra-Transaction Association Analysis:Mining Multi-Dimensional Inter-Transaction Association Rules", ACM Transactions on Information Systems (TOIS'00), 18(4): 423-454, 2000.
- O. R. Zaiane, M. Xin, J. Han, "Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs," Proc. Advances in Digital Librar ies Conf. (ADL'98), Santa Barbara, CA, April 1998, pp. 19-29
- O. R. Zaiane, J. Han, and H. Zhu, "Mining Recurrent Items in Multimedia with Progressive Resolution Refinement", ICDE'00, San Diego, CA, Feb. 2000, pp. 461-470



References: Frequent-pattern Mining for Classification and Data Cube Computation

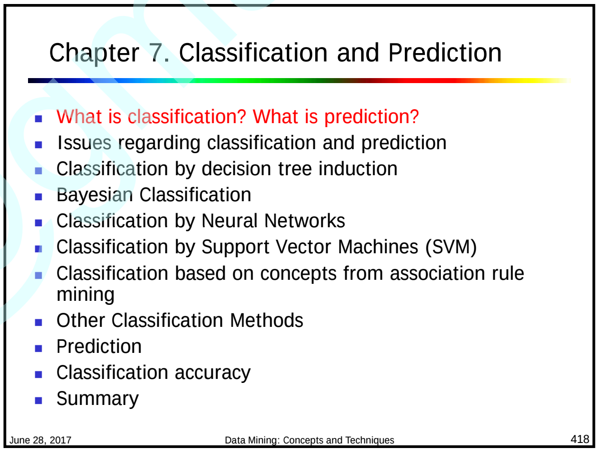
- K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cubes. SIGMOD'99, 359-370, Philadelphia, PA, June 1999.
- M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J. D. Ullman. Computing iceberg queries efficiently. VLDB'98, 299-310, New York, NY, Aug. 1998.
- J. Han, J. Pei, G. Dong, and K. Wang, "Computing Iceberg Data Cubes with Complex Measures", Proc. ACM-SIGMOD'2001, Santa Barbara, CA, May 2001.
- M. Kamber, J. Han, and J. Y. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. KDD'97, 207-210, Newport Beach, California.
- K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cubes. SIGMOD'99
- T. Imielinski, L. Khachiyan, and A. Abdulghani. Cubegrades: Generalizing association rules. Technical Report, Aug. 2000



Data Mining:
Concepts and Techniques

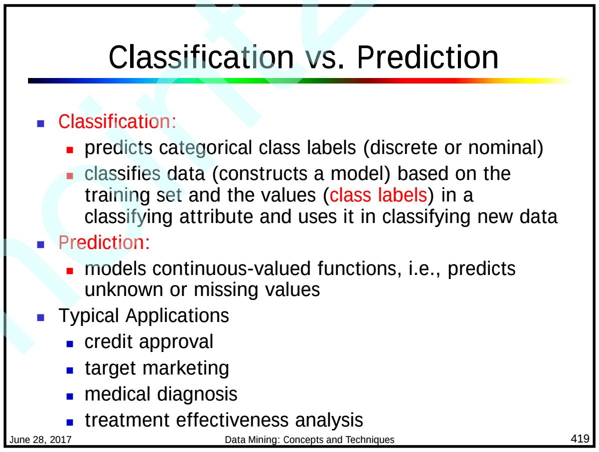
— Slides for Textbook —
— Chapter 7 —

©Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj



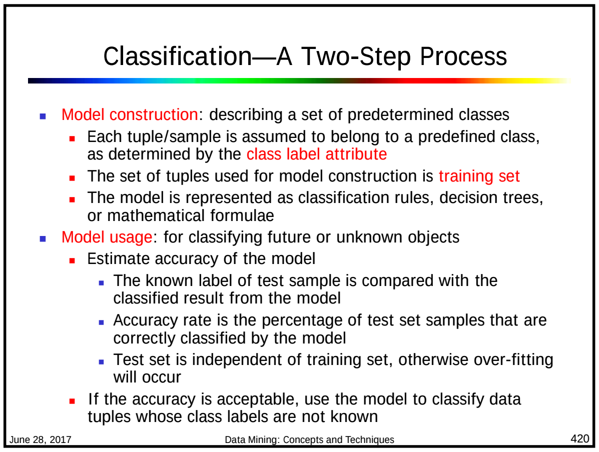
Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary



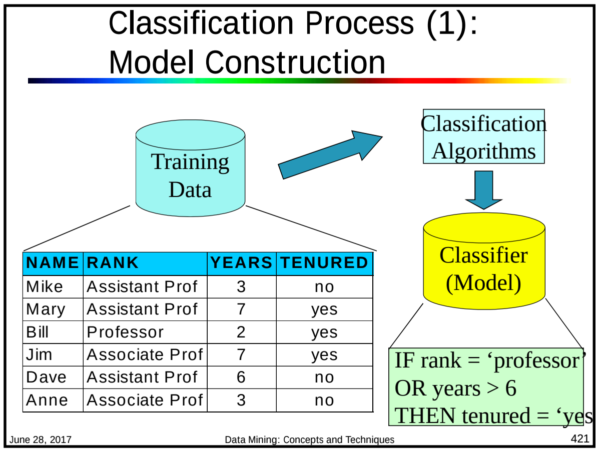
Classification vs. Prediction

- Classification:
  - predicts categorical class labels (discrete or nominal)
  - classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data
- Prediction:
  - models continuous-valued functions, i.e., predicts unknown or missing values
- Typical Applications
  - credit approval
  - target marketing
  - medical diagnosis
  - treatment effectiveness analysis



Classification—A Two-Step Process

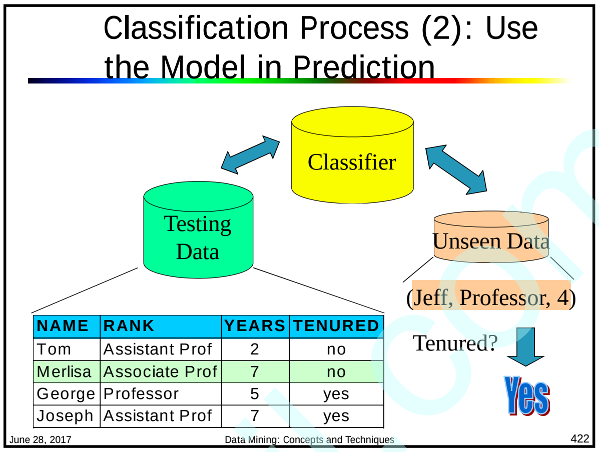
- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction is training set
  - The model is represented as classification rules, decision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



Classification Process (1): Model Construction

NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

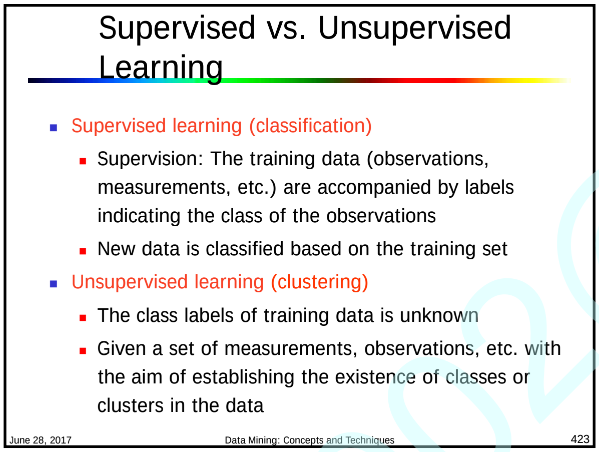


Classification Process (2): Use the Model in Prediction

NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

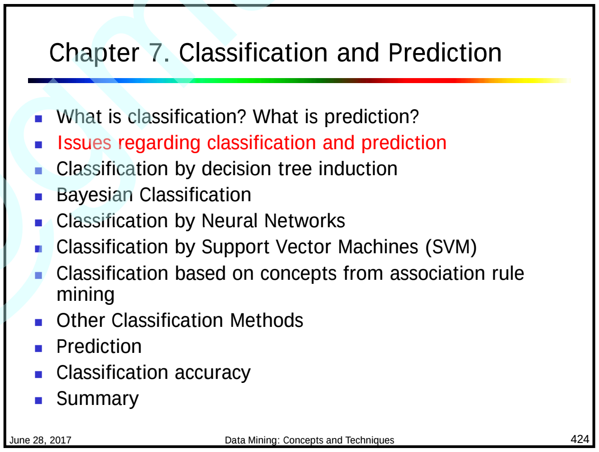
(Jeff, Professor, 4)

Tenured?



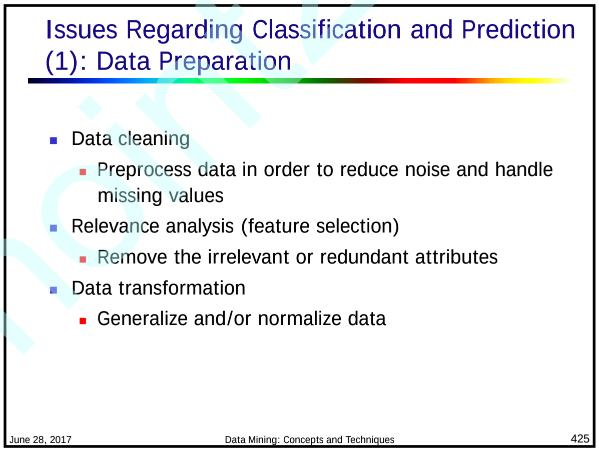
Supervised vs. Unsupervised Learning

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



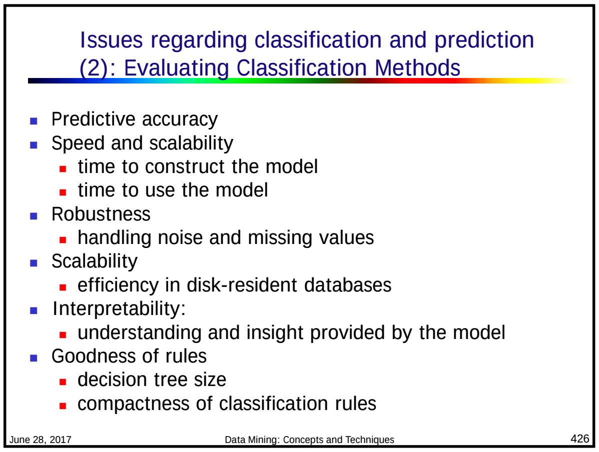
Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary



Issues Regarding Classification and Prediction (1): Data Preparation

- Data cleaning
  - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
  - Remove the irrelevant or redundant attributes
- Data transformation
  - Generalize and/or normalize data



Issues regarding classification and prediction (2): Evaluating Classification Methods

- Predictive accuracy
- Speed and scalability
  - time to construct the model
  - time to use the model
- Robustness
  - handling noise and missing values
- Scalability
  - efficiency in disk-resident databases
- Interpretability:
  - understanding and insight provided by the model
- Goodness of rules
  - decision tree size
  - compactness of classification rules

Chapter 7. Classification and Prediction

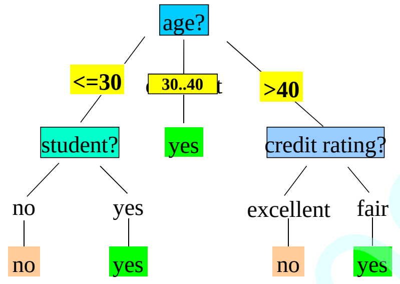
- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Training Dataset

This follows an example from Quinlan's ID3

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Output: A Decision Tree for "*buys_computer*"



Algorithm for Decision Tree Induction

- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - At start, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

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

- Select the attribute with the highest information gain
- S contains s_i tuples of class C_i for i = {1, …, m}
- information measures info required to classify any arbitrary tuple

$$I(s_1,s_2,\ldots,s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

- entropy of attribute A with values {$a_1,a_2,\ldots,a_v$}

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j}+\ldots+s_{mj}}{s} I(s_{1j},\ldots,s_{mj})$$

- information gained by branching on attribute A

$$Gain(A) = I(s_1, s_2,\ldots, s_m) - E(A)$$

Attribute Selection by Information Gain Computation

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"
- I(p, n) = I(9, 5) =0.940
- Compute the entropy for *age*:

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
30…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694$$

$\frac{5}{14} I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = I(p,n) - E(age) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Other Attribute Selection Measures

- Gini index (CART, IBM IntelligentMiner)
  - All attributes are assumed continuous-valued
  - Assume there exist several possible split values for each attribute
  - May need other tools, such as clustering, to get the possible split values
  - Can be modified for categorical attributes

Gini index (IBM IntelligentMiner)

- If a data set *T* contains examples from *n* classes, gini index, *gini*(*T*) is defined as

  $$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

  where p_j is the relative frequency of class *j* in *T.*
- If a data set *T* is split into two subsets T_1 and T_2 with sizes N_1 and N_2 respectively, the *gini* index of the split data contains examples from *n* classes, the *gini* index *gini*(*T*) is defined as

  $$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

- The attribute split the nod *points for each attribute*).

Extracting Classification Rules from Trees

- Represent the knowledge in the form of IF-THEN rules
- One rule is created for each path from the root to a leaf
- Each attribute-value pair along a path forms a conjunction
- The leaf node holds the class prediction
- Rules are easier for humans to understand
- Example

IF *age* = "<=30" AND *student* = "*no*" THEN *buys_computer* = "*no*"

IF *age* = "<=30" AND *student* = "*yes*" THEN *buys_computer* = "*yes*"

IF *age* = "31…40" THEN *buys_computer* = "*yes*"

IF *age* = ">40" AND *credit_rating* = "*excellent*" THEN *buys_computer* = "*yes*"

IF *age* = "<=30" AND *credit_rating* = "*fair*" THEN *buys_computer* = "*no*"

Avoid Overfitting in Classification

- Overfitting: An induced tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: Remove branches from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

Approaches to Determine the Final Tree Size

- Separate training (2/3) and testing (1/3) sets
- Use cross validation, e.g., 10-fold cross validation
- Use all the data for training
  - but apply a statistical test (e.g., chi-square) to estimate whether expanding or pruning a node may improve the entire distribution
- Use minimum description length (MDL) principle
  - halting growth of the tree when the encoding is minimized

Enhancements to basic decision tree induction

- Allow for continuous-valued attributes
  - Dynamically define new discrete-valued attributes that partition the continuous attribute value into a discrete set of intervals
- Handle missing attribute values
  - Assign the most common value of the attribute
  - Assign probability to each of the possible values
- Attribute construction
  - Create new attributes based on existing ones that are sparsely represented
  - This reduces fragmentation, repetition, and replication

Classification in Large Databases

- Classification—a classical problem extensively studied by statisticians and machine learning researchers
- Scalability: Classifying data sets with millions of examples and hundreds of attributes with reasonable speed
- Why decision tree induction in data mining?
  - relatively faster learning speed (than other classification methods)
  - convertible to simple and easy to understand classification rules
  - can use SQL queries for accessing databases
  - comparable classification accuracy with other methods

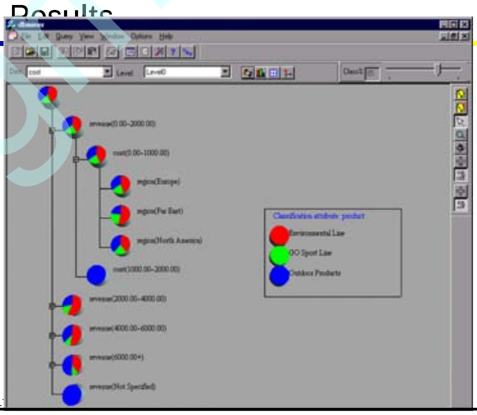
Induction Methods in Data Mining Studies

- SLIQ (EDBT'96 — Mehta et al.)
  - builds an index for each attribute and only class list and the current attribute list reside in memory
- SPRINT (VLDB'96 — J. Shafer et al.)
  - constructs an attribute list data structure
- PUBLIC (VLDB'98 — Rastogi & Shim)
  - integrates tree splitting and tree pruning: stop growing the tree earlier
- RainForest (VLDB'98 — Gehrke, Ramakrishnan & Ganti)
  - separates the scalability aspects from the criteria that determine the quality of the tree
  - builds an AVC-list (attribute, value, class label)

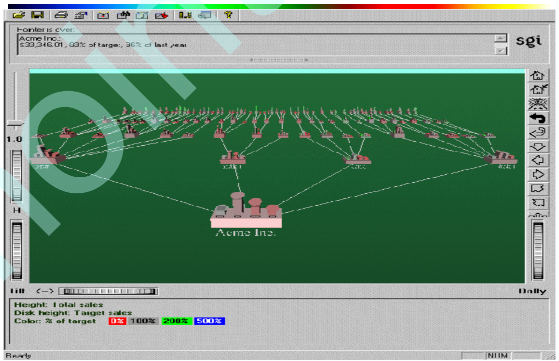
Data Cube-Based Decision-Tree Induction

- Integration of generalization with decision-tree induction (Kamber et al'97).
- Classification at primitive concept levels
  - E.g., precise temperature, humidity, outlook, etc.
  - Low-level concepts, scattered classes, bushy classification-trees
  - Semantic interpretation problems.
- Cube-based multi-level classification
  - Relevance analysis at multi-levels.
  - Information-gain analysis with dimension + level.

Presentation of Classification Results



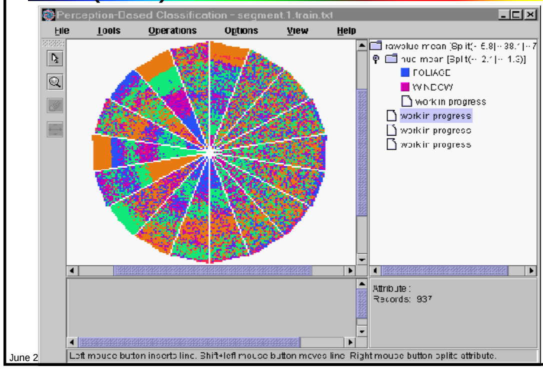
Visualization of a Decision Tree in SGI/MineSet 3.0


Pointer is over:
Acme Inc.:
$33,316.01, 83% of target, 96% of last year

Height: Total sales
Disk height: Target sales
Color: % of target 0% 100% 200% 500%

Ready

Perception-Based Classification (PBC)


Perception-Based Classification - segment.1.train.txt

rawvalue mean [Split(-- 6.8|-- 38.1|-- 7
hue mean [Split(-- 2.1|-- 1.3)]
FOLIAGE
WINDOW
work in progress
work in progress
work in progress
work in progress

Attribute :
Records: 937

Left mouse button inserts line. Shift+left mouse button moves line. Right mouse button splits attribute.

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Bayesian Classification: Why?

- Probabilistic learning: Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct. Prior knowledge can be combined with observed data.
- Probabilistic prediction: Predict multiple hypotheses, weighted by their probabilities
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Bayesian Theorem: Basics

- Let X be a data sample whose class label is unknown
- Let H be a hypothesis that X belongs to class C
- For classification problems, determine P(H/X): the probability that the hypothesis holds given the observed data sample X
- P(H): prior probability of hypothesis H (i.e. the initial probability before we observe any data, reflects the background knowledge)
- P(X): probability that sample data is observed
- P(X|H) : probability of observing the sample X, given that the hypothesis holds

Bayesian Theorem

- Given training data *X, posteriori probability of a hypothesis H, P(H|X)* follows the Bayes theorem

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

- Informally, this can be written as

  posterior =likelihood x prior / evidence
- MAP (maximum posteriori) hypothesis

$$h_{MAP} \equiv \arg\max_{h\in H} P(h|D) = \arg\max_{h\in H} P(D|h)P(h)$$

- Practical difficulty: require initial knowledge of many probabilities, significant computational cost

Naïve Bayes Classifier

- A simplified assumption: attributes are conditionally independent:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

- The product of occurrence of say 2 elements x_1 and x_2, given the current class is C, is the product of the probabilities of each element taken separately, given the same class P([y_1,y_2],C) = P(y_1,C) * P(y_2,C)
- No dependence relation between attributes
- Greatly reduces the computation cost, only count the class distribution.
- Once the probability P(X|C_i) is known, assign X to the class with maximum P(X|C_i)*P(C_i)

Training dataset

Class:
C1:buys_computer= 'yes'
C2:buys_computer= 'no'

Data sample
X =(age<=30,
Income=medium,
Student=yes
Credit_rating=
Fair)

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
30…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayesian Classifier: Example

- Compute P(X/Ci) for each class

P(age="<30" | buys_computer="yes") = 2/9=0.222
P(age="<30" | buys_computer="no") = 3/5 =0.6
P(income="medium" | buys_computer="yes")= 4/9 =0.444
P(income="medium" | buys_computer="no") = 2/5 = 0.4
P(student="yes" | buys_computer="yes)= 6/9 =0.667
P(student="yes" | buys_computer="no")= 1/5=0.2
P(credit_rating="fair" | buys_computer="yes")=6/9=0.667
P(credit_rating="fair" | buys_computer="no")=2/5=0.4

X=(age<=30 ,income =medium, student=yes,credit_rating=fair)

P(X|Ci) : P(X|buys_computer="yes")= 0.222 x 0.444 x 0.667 x 0.0.667 =0.044
P(X|buys_computer="no")= 0.6 x 0.4 x 0.2 x 0.4 =0.019
P(X|Ci)*P(Ci) : P(X|buys_computer="yes") * P(buys_computer="yes")=0.028
P(X|buys_computer="yes") * P(buys_computer="yes")=0.007

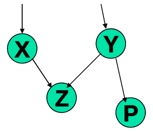
X belongs to class "buys_computer=yes"

Naïve Bayesian Classifier: Comments

- Advantages :
  - Easy to implement
  - Good results obtained in most of the cases
- Disadvantages
  - Assumption: class conditional independence , therefore loss of accuracy
  - Practically, dependencies exist among variables
  - E.g., hospitals: patients: Profile: age, family history etc
    Symptoms: fever, cough etc., Disease: lung cancer, diabetes etc
  - Dependencies among these cannot be modeled by Naïve Bayesian Classifier
- How to deal with these dependencies?
  - Bayesian Belief Networks

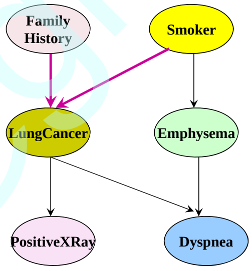
Bayesian Networks

- Bayesian belief network allows a *subset* of the variables conditionally independent
- A graphical model of causal relationships
  - Represents dependency among the variables
  - Gives a specification of joint probability distribution



❑Nodes: random variables
❑Links: dependency
❑X,Y are the parents of Z, and Y is the parent of P
❑No dependency between Z and P
❑Has no loops or cycles

Bayesian Belief Network: An Example



Bayesian Belief Networks

	(FH, S)	(FH, ~S)	(~FH, S)	(~FH, ~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

The conditional probability table for the variable LungCancer:
Shows the conditional probability for each possible combination of its parents

$$P(z1,...,zn) = \prod_{i=1}^{n} P(z_i \mid Parents(Z_i))$$

Learning Bayesian Networks

- Several cases
  - Given both the network structure and all variables observable: learn only the CPTs
  - Network structure known, some hidden variables: method of gradient descent, analogous to neural network learning
  - Network structure unknown, all variables observable: search through the model space to reconstruct graph topology
  - Unknown structure, all hidden variables: no good algorithms known for this purpose
- D. Heckerman, Bayesian networks for data mining

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Classification

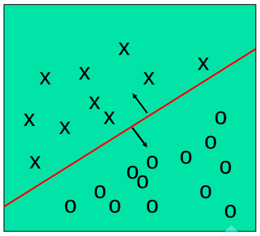
- Classification:
  - predicts categorical class labels
- Typical Applications
  - {credit history, salary}-> credit approval (Yes/No)
  - {Temp, Humidity} --> Rain (Yes/No)

$$x \in X = \{0,1\}^n, y \in Y = \{0,1\}$$

Mathematically $h : X \rightarrow Y$

$$y = h(x)$$

Linear Classification



- Binary Classification problem
- The data above the red line belongs to class 'x'
- The data below red line belongs to class 'o'
- Examples – SVM, Perceptron, Probabilistic Classifiers

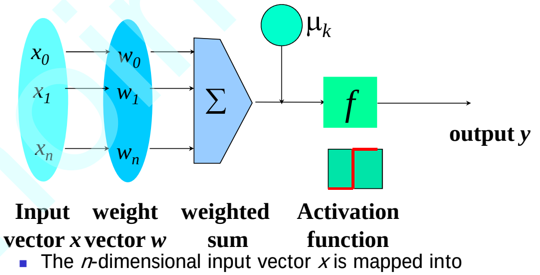
Discriminative Classifiers

- Advantages
  - prediction accuracy is generally high
    - (as compared to Bayesian methods – in general)
  - robust, works when training examples contain errors
  - fast evaluation of the learned target function
    - (Bayesian networks are normally slow)
- Criticism
  - long training time
  - difficult to understand the learned function (weights)
    - (Bayesian networks can be used easily for pattern discovery)
  - not easy to incorporate domain knowledge
    - (easy in the form of priors on the data or distributions)

Neural Networks

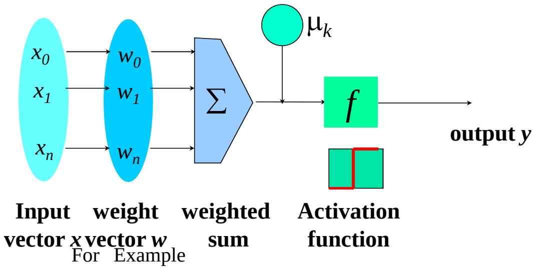
- Analogy to Biological Systems (Indeed a great example of a good learning system)
- Massive Parallelism allowing for computational efficiency
- The first learning algorithm came in 1959 (Rosenblatt) who suggested that if a target output value is provided for a single neuron with fixed inputs, one can incrementally change weights to learn to produce these outputs using the perceptron learning rule

A Neuron



- The n-dimensional input vector x is mapped into variable y by means of the scalar product and a nonlinear function mapping

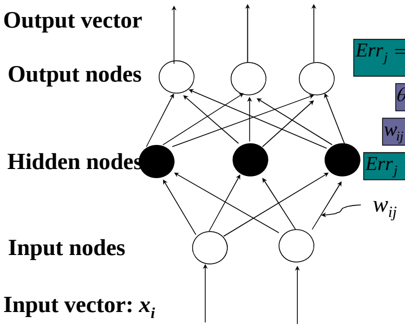
A Neuron



For Example

$$y = \text{sign}(\sum_{i=0}^{n} w_i x_i + \mu_k)$$

Multi-Layer Perceptron



$$Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$$

$$\theta_j = \theta_j + (l)Err_j$$

$$w_{ij} = w_{ij} + (l)Err_j O_i$$

$$Err_j = O_j(1-O_j)(T_j - O_j)$$

$$O_j = \frac{1}{1+e^{-I_j}}$$

$$I_j = \sum_i w_{ij} O_i + \theta_j$$

Network Training

- The ultimate objective of training
  - obtain a set of weights that makes almost all the tuples in the training data classified correctly
- Steps
  - Initialize weights with random values
  - Feed the input tuples into the network one by one
  - For each unit
    - Compute the net input to the unit as a linear combination of all the inputs to the unit
    - Compute the output value using the activation function
    - Compute the error
    - Update the weights and the bias

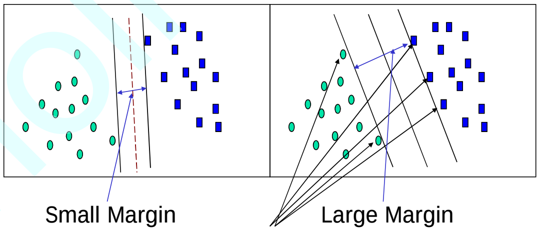
Network Pruning and Rule Extraction

- Network pruning
  - Fully connected network will be hard to articulate
  - *N* input nodes, *h* hidden nodes and *m* output nodes lead to *h(m+N)* weights
  - Pruning: Remove some of the links without affecting classification accuracy of the network
- Extracting rules from a trained network
  - Discretize activation values; replace individual activation value by the cluster average maintaining the network accuracy
  - Enumerate the output from the discretized activation values to find rules between activation value and output
  - Find the relationship between the input and activation value
  - Combine the above two to have rules relating the output to input

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

SVM – Support Vector Machines



SVM – Cont.

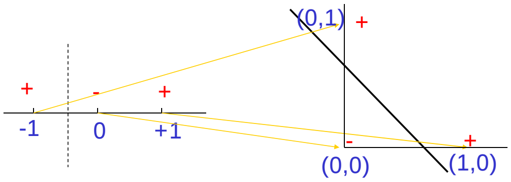
- Linear Support Vector Machine

Given a set of points $x_i \in \Re^n$ with label $y_i \in \{-1,1\}$
The SVM finds a hyperplane defined by the pair (w,b)
(where w is the normal to the plane and b is the distance from the origin)
s.t. $y_i(x_i \cdot w + b) \geq +1 \quad i = 1,..., N$

x – feature vector, b- bias, y- class label, ||w|| - margin

SVM – Cont.

- What if the data is not linearly separable?
- Project the data to high dimensional space where it is linearly separable and then we can use linear SVM – (Using Kernels)



Non-Linear SVM

Classification using SVM (w,b)

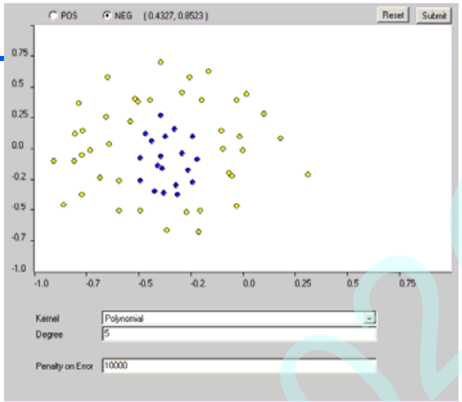
$$x_i \cdot w + b \overset{?}{>} 0$$

In non linear case we can see this as

$$K(x_i, w) + b \overset{?}{>} 0$$

Kernel – Can be thought of as doing dot product in some high dimensional space

Example of Non-linear SVM



(0.4327, 0.8523)

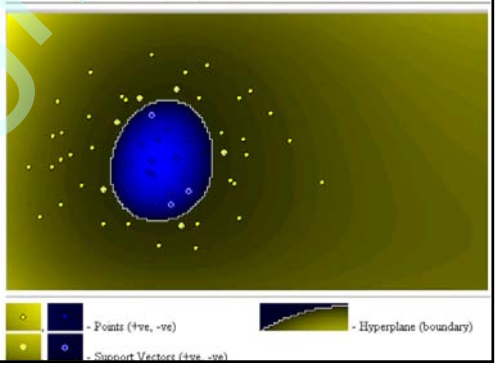
Kernel: Polynomial

Degree: 5

Penalty on Error: 10000

Results

Number of Support Vectors: **9** (-ve: 3, +ve: 6) Total number of points: 60



SVM vs. Neural Network

- SVM
  - Relatively new concept
  - Nice Generalization properties
  - Hard to learn – learned in batch mode using quadratic programming techniques
  - Using kernels can learn very complex functions
- Neural Network
  - Quiet Old
  - Generalizes well but doesn't have strong mathematical foundation
  - Can easily be learned in incremental fashion
  - To learn complex functions – use multilayer perceptron (not that trivial)

SVM Related Links

- http://svm.dcs.rhbnc.ac.uk/
- http://www.kernel-machines.org/
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.
- SVM^{light} – Software (in C) http://ais.gmd.de/~thorsten/svm_light
- BOOK: An Introduction to Support Vector Machines
  N. Cristianini and J. Shawe-Taylor
  Cambridge University Press

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Association-Based Classification

- Several methods for association-based classification
  - ARCS: Quantitative association mining and clustering of association rules (Lent et al'97)
    - It beats C4.5 in (mainly) scalability and also accuracy
  - Associative classification: (Liu et al'98)
    - It mines high support and high confidence rules in the form of "cond_set => y", where y is a class label
  - CAEP (Classification by aggregating emerging patterns) (Dong et al'99)
    - Emerging patterns (EPs): the itemsets whose support increases significantly from one class to another
    - Mine Eps based on minimum support and growth rate

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Other Classification Methods

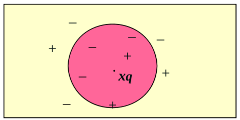
- k-nearest neighbor classifier
- case-based reasoning
- Genetic algorithm
- Rough set approach
- Fuzzy set approaches

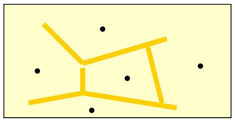
Instance-Based Methods

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

The *k*-Nearest Neighbor Algorithm

- All instances correspond to points in the n-D space.
- The nearest neighbor are defined in terms of Euclidean distance.
- The target function could be discrete- or real- valued.
- For discrete-valued, the *k*-NN returns the most common value among the k training examples nearest to *xq*.
- Vonoroi diagram: the decision surface induced by 1-NN for a typical set of training examples.





Discussion on the *k*-NN Algorithm

- The k-NN algorithm for continuous-valued target functions
  - Calculate the mean values of the *k* nearest neighbors
- Distance-weighted nearest neighbor algorithm
  - Weight the contribution of each of the k neighbors according to their distance to the query point x_q
    - giving greater weight to closer neighbors
  - Similarly, for real-valued target functions $w \equiv \frac{1}{d(x_q, x_i)^2}$
- Robust to noisy data by averaging k-nearest neighbors
- Curse of dimensionality: distance between neighbors could be dominated by irrelevant attributes.
  - To overcome it, axes stretch or elimination of the least relevant attributes.

Case-Based Reasoning

- Also uses: lazy evaluation + analyze similar instances
- Difference: Instances are not "points in a Euclidean space"
- Example: Water faucet problem in CADET (Sycara et al'92)
- Methodology
  - Instances represented by rich symbolic descriptions (e.g., function graphs)
  - Multiple retrieved cases may be combined
  - Tight coupling between case retrieval, knowledge-based reasoning, and problem solving
- Research issues
  - Indexing based on syntactic similarity measure, and when failure, backtracking, and adapting to additional cases

Remarks on Lazy vs. Eager Learning

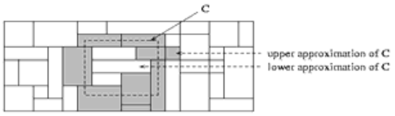
- Instance-based learning: lazy evaluation
- Decision-tree and Bayesian classification: eager evaluation
- Key differences
  - Lazy method may consider query instance *xq* when deciding how to generalize beyond the training data *D*
  - Eager method cannot since they have already chosen global approximation when seeing the query
- Efficiency: Lazy - less time training but more time predicting
- Accuracy
  - Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
  - Eager: must commit to a single hypothesis that covers the entire instance space

Genetic Algorithms

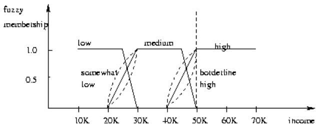
- GA: based on an analogy to biological evolution
- Each rule is represented by a string of bits
- An initial population is created consisting of randomly generated rules
  - e.g., IF A_1 and Not A_2 then C_2 can be encoded as 100
- Based on the notion of survival of the fittest, a new population is formed to consists of the fittest rules and their offsprings
- The fitness of a rule is represented by its classification accuracy on a set of training examples
- Offsprings are generated by crossover and mutation

Rough Set Approach

- Rough sets are used to approximately or "roughly" define equivalent classes
- A rough set for a given class C is approximated by two sets: a lower approximation (certain to be in C) and an upper approximation (cannot be described as not belonging to C)
- Finding the minimal subsets (reducts) of attributes (for feature reduction) is NP-hard but a discernibility matrix is used to reduce the computation intensity



Fuzzy Set Approaches



- Fuzzy logic uses truth values between 0.0 and 1.0 to represent the degree of membership (such as using fuzzy membership graph)
- Attribute values are converted to fuzzy values
  - e.g., income is mapped into the discrete categories {low, medium, high} with fuzzy values calculated
- For a given new sample, more than one fuzzy value may apply
- Each applicable rule contributes a vote for membership in the categories
- Typically, the truth values for each predicted category are summed

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

What Is Prediction?

- Prediction is similar to classification
  - First, construct a model
  - Second, use model to predict unknown value
    - Major method for prediction is regression
      - Linear and multiple regression
      - Non-linear regression
- Prediction is different from classification
  - Classification refers to predict categorical class label
  - Prediction models continuous-valued functions

Predictive Modeling in Databases

- Predictive modeling: Predict data values or construct generalized linear models based on the database data.
- One can only predict value ranges or category distributions
- Method outline:
  - Minimal generalization
  - Attribute relevance analysis
  - Generalized linear model construction
  - Prediction
- Determine the major factors which influence the prediction
  - Data relevance analysis: uncertainty measurement, entropy analysis, expert judgement, etc.
- Multi-level prediction: drill-down and roll-up analysis

Regress Analysis and Log-Linear Models in Prediction

- Linear regression: Y = α + β X
  - Two parameters , α and β specify the line and are to be estimated by using the data at hand.
  - using the least squares criterion to the known values of Y1, Y2, …, X1, X2, ….
- Multiple regression: Y = b0 + b1 X1 + b2 X2.
  - Many nonlinear functions can be transformed into the above.
- Log-linear models:
  - The multi-way table of joint probabilities is approximated by a product of lower-order tables.
  - Probability: *p(a, b, c, d) = αab βacχad δbcd*

Locally Weighted Regression

- Construct an explicit approximation to *f* over a local region surrounding query instance *xq*.
- Locally weighted linear regression:
  - The target function *f* is approximated near *xq* using the linear function:

$$\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$$

  - minimize the squared error: distance-decreasing weight *K*

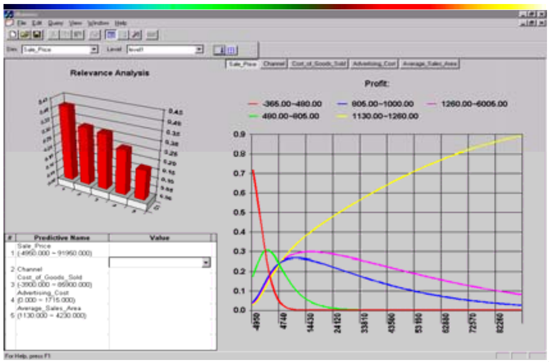
$$E(x_q) \equiv \frac{1}{2} \sum_{x \in k\_nearest\_neighbors\_of\_x_q} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$$

  - the gradient descent training rule:

$$\Delta w_j \equiv \eta \sum_{x \in k\_nearest\_neighbors\_of\_x_q} K(d(x_q, x))((f(x) - \hat{f}(x))a_j(x)$$

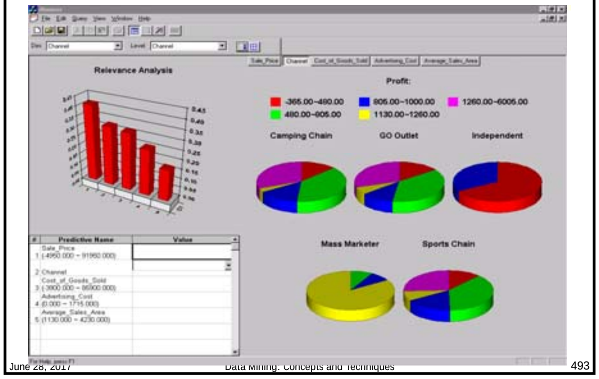
- In most cases, the target function is approximated by a constant, linear, or quadratic function.

Prediction: Numerical Data



#	Predictive Name	Value
1	Sale_Price (-4950.000 ~ 91950.000)	
2	Channel	
3	Cost_of_Goods_Sold (-3900.000 ~ 86900.000)	
4	Advertising_Cost (0.000 ~ 1715.000)	
5	Average_Sales_Area (1130.000 ~ 4230.000)	

Prediction: Categorical Data



#	Predictive Name	Value
1	Sale_Price (-4950.000 ~ 91950.000)	
2	Channel	
3	Cost_of_Goods_Sold (-3900.000 ~ 86900.000)	
4	Advertising_Cost (0.000 ~ 1715.000)	
5	Average_Sales_Area (1130.000 ~ 4230.000)	

Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Classification Accuracy: Estimating Error Rates

- Partition: Training-and-testing
  - use two independent data sets, e.g., training set (2/3), test set(1/3)
  - used for data set with large number of samples
- Cross-validation
  - divide the data set into *k* subsamples
  - use *k-1* subsamples as training data and one sub-sample as test data—*k*-fold cross-validation
  - for data set with moderate size
- Bootstrapping (leave-one-out)
  - for small size data

Bagging and Boosting

- General idea

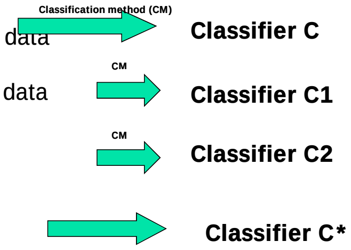
Training data

Altered Training data

Altered Training data

……..

Aggregation ….



Classification method (CM)

Classifier C

CM

Classifier C1

CM

Classifier C2

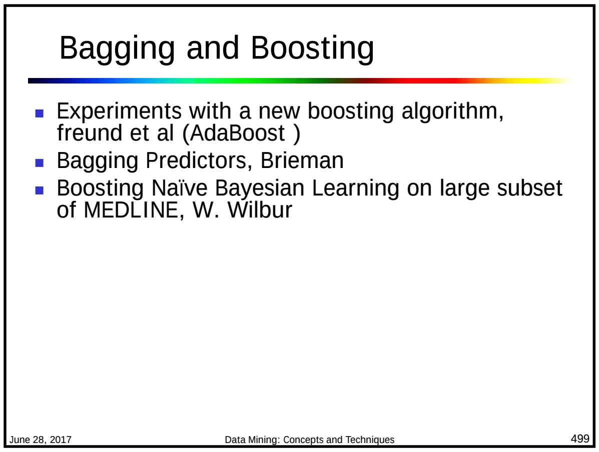
Classifier C*

Bagging

- Given a set S of s samples
- Generate a bootstrap sample T from S. Cases in S may not appear in T or may appear more than once.
- Repeat this sampling procedure, getting a sequence of k independent training sets
- A corresponding sequence of classifiers C1,C2,…,Ck is constructed for each of these training sets, by using the same classification algorithm
- To classify an unknown sample X,let each classifier predict or vote
- The Bagged Classifier C* counts the votes and assigns X to the class with the “most” votes

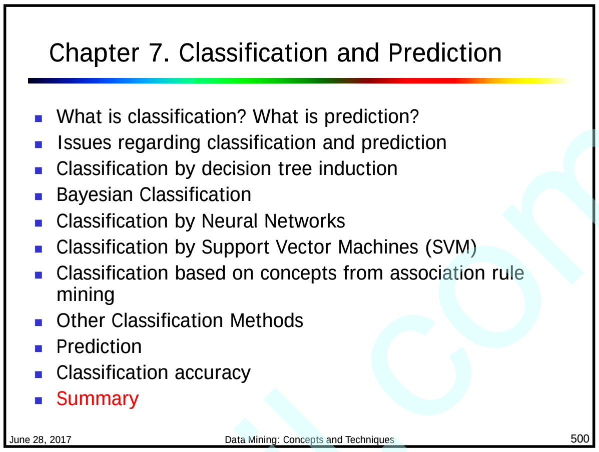
Boosting Technique — Algorithm

- Assign every example an equal weight *1/N*
- *For t = 1, 2, …, T Do*
  - Obtain a hypothesis (classifier) $h^{(t)}$ under $w^{(t)}$
  - Calculate the error of *h(t)* and re-weight the examples based on the error . Each classifier is dependent on the previous ones. Samples that are incorrectly predicted are weighted more heavily
  - Normalize $w^{(t+1)}$ to sum to 1 (weights assigned to different classifiers sum to 1)
- Output a weighted sum of all the hypothesis, with each hypothesis weighted according to its accuracy on the training set



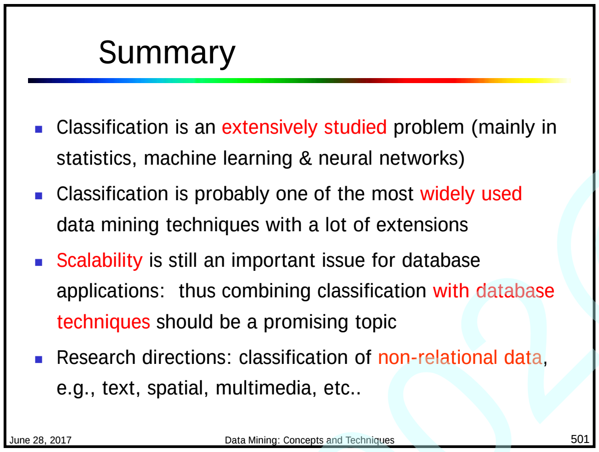
Bagging and Boosting

- Experiments with a new boosting algorithm, freund et al (AdaBoost)
- Bagging Predictors, Brieman
- Boosting Naïve Bayesian Learning on large subset of MEDLINE, W. Wilbur



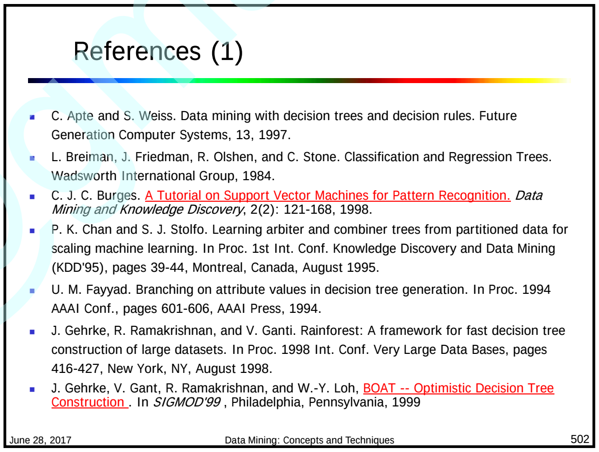
Chapter 7. Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by Neural Networks
- Classification by Support Vector Machines (SVM)
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary



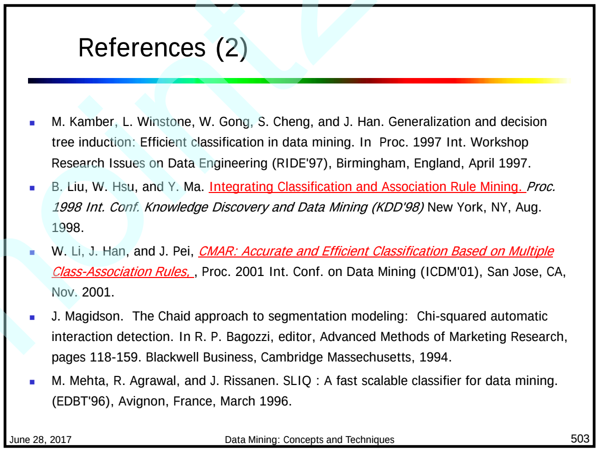
Summary

- Classification is an extensively studied problem (mainly in statistics, machine learning & neural networks)
- Classification is probably one of the most widely used data mining techniques with a lot of extensions
- Scalability is still an important issue for database applications: thus combining classification with database techniques should be a promising topic
- Research directions: classification of non-relational data, e.g., text, spatial, multimedia, etc..



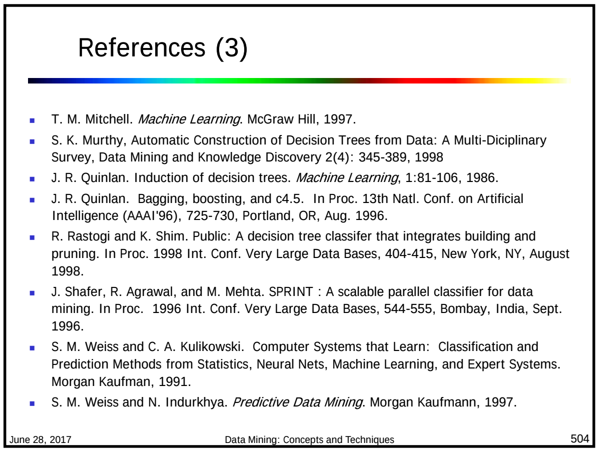
References (1)

- C. Apte and S. Weiss. Data mining with decision trees and decision rules. Future Generation Computer Systems, 13, 1997.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth International Group, 1984.
- C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2): 121-168, 1998.
- P. K. Chan and S. J. Stolfo. Learning arbiter and combiner trees from partitioned data for scaling machine learning. In Proc. 1st Int. Conf. Knowledge Discovery and Data Mining (KDD'95), pages 39-44, Montreal, Canada, August 1995.
- U. M. Fayyad. Branching on attribute values in decision tree generation. In Proc. 1994 AAAI Conf., pages 601-606, AAAI Press, 1994.
- J. Gehrke, R. Ramakrishnan, and V. Ganti. Rainforest: A framework for fast decision tree construction of large datasets. In Proc. 1998 Int. Conf. Very Large Data Bases, pages 416-427, New York, NY, August 1998.
- J. Gehrke, V. Gant, R. Ramakrishnan, and W.-Y. Loh, BOAT -- Optimistic Decision Tree Construction . In *SIGMOD'99* , Philadelphia, Pennsylvania, 1999



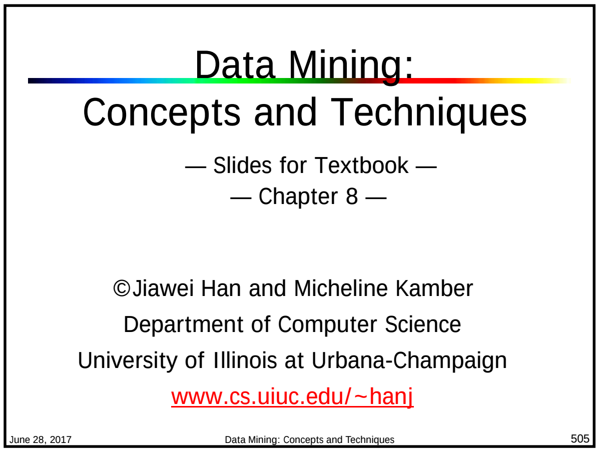
References (2)

- M. Kamber, L. Winstone, W. Gong, S. Cheng, and J. Han. Generalization and decision tree induction: Efficient classification in data mining. In Proc. 1997 Int. Workshop Research Issues on Data Engineering (RIDE'97), Birmingham, England, April 1997.
- B. Liu, W. Hsu, and Y. Ma. Integrating Classification and Association Rule Mining. *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining (KDD'98)* New York, NY, Aug. 1998.
- W. Li, J. Han, and J. Pei, *CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules,* , Proc. 2001 Int. Conf. on Data Mining (ICDM'01), San Jose, CA, Nov. 2001.
- J. Magidson. The Chaid approach to segmentation modeling: Chi-squared automatic interaction detection. In R. P. Bagozzi, editor, Advanced Methods of Marketing Research, pages 118-159. Blackwell Business, Cambridge Massechusetts, 1994.
- M. Mehta, R. Agrawal, and J. Rissanen. SLIQ : A fast scalable classifier for data mining. (EDBT'96), Avignon, France, March 1996.



References (3)

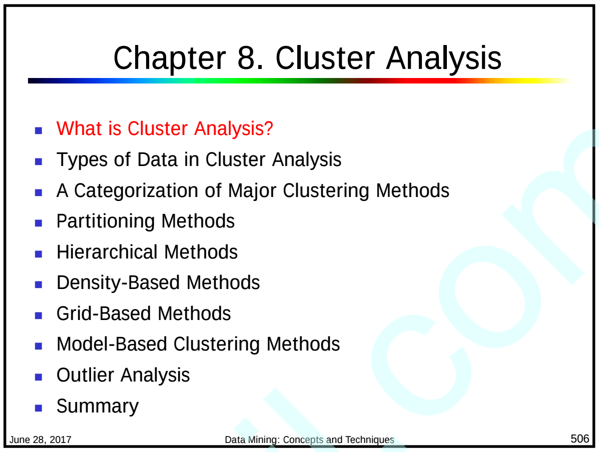
- T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- S. K. Murthy, Automatic Construction of Decision Trees from Data: A Multi-Diciplinary Survey, Data Mining and Knowledge Discovery 2(4): 345-389, 1998
- J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81-106, 1986.
- J. R. Quinlan. Bagging, boosting, and c4.5. In Proc. 13th Natl. Conf. on Artificial Intelligence (AAAI'96), 725-730, Portland, OR, Aug. 1996.
- R. Rastogi and K. Shim. Public: A decision tree classifer that integrates building and pruning. In Proc. 1998 Int. Conf. Very Large Data Bases, 404-415, New York, NY, August 1998.
- J. Shafer, R. Agrawal, and M. Mehta. SPRINT : A scalable parallel classifier for data mining. In Proc. 1996 Int. Conf. Very Large Data Bases, 544-555, Bombay, India, Sept. 1996.
- S. M. Weiss and C. A. Kulikowski. Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems. Morgan Kaufman, 1991.
- S. M. Weiss and N. Indurkhya. *Predictive Data Mining*. Morgan Kaufmann, 1997.



Data Mining:
Concepts and Techniques

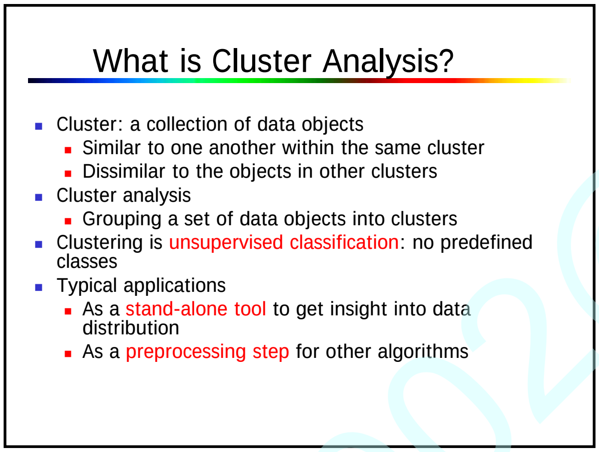
— Slides for Textbook —
— Chapter 8 —

©Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj



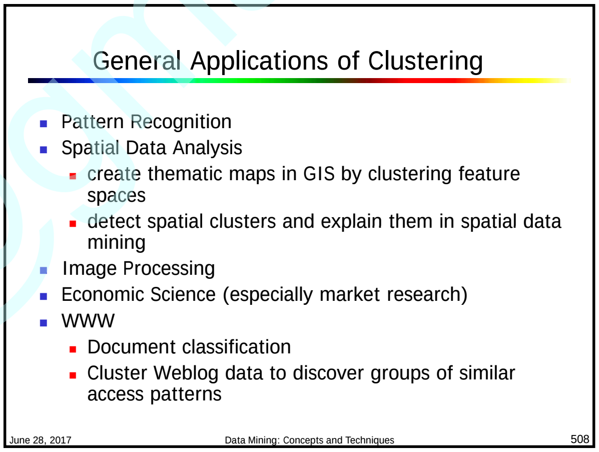
Chapter 8. Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary



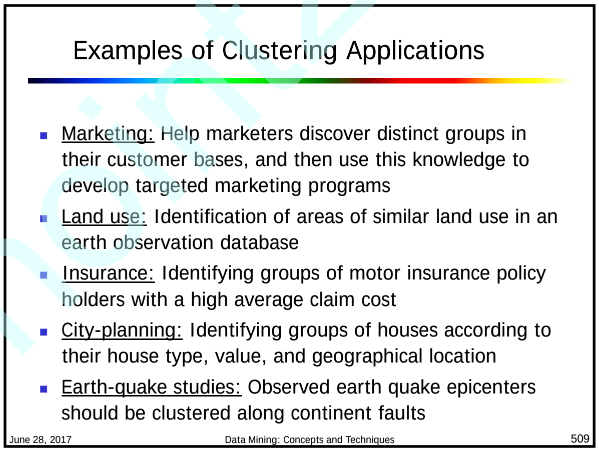
What is Cluster Analysis?

- Cluster: a collection of data objects
  - Similar to one another within the same cluster
  - Dissimilar to the objects in other clusters
- Cluster analysis
  - Grouping a set of data objects into clusters
- Clustering is unsupervised classification: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms



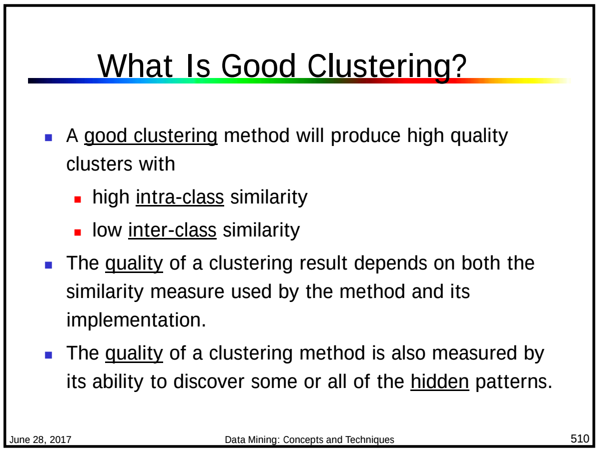
General Applications of Clustering

- Pattern Recognition
- Spatial Data Analysis
  - create thematic maps in GIS by clustering feature spaces
  - detect spatial clusters and explain them in spatial data mining
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns



Examples of Clustering Applications

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



What Is Good Clustering?

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

Requirements of Clustering in Data Mining

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

Chapter 8. Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Data Structures

- Data matrix
  - (two modes)

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

- Dissimilarity matrix
  - (one mode)

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

Measure the Quality of Clustering

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

Type of data in clustering analysis

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

Interval-valued variables

- Standardize data
  - Calculate the mean absolute deviation:

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

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

  - Calculate the standardized measurement (*z-score*)

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

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

Similarity and Dissimilarity Between Objects

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

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+...+|x_{i_p}-x_{j_p}|^q)}$$

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

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

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

Similarity and Dissimilarity Between Objects (Cont.)

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

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

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

Binary Variables

- A contingency table for binary data

		Object *j*		
		1	0	*sum*
	1	*a*	*b*	*a*+*b*
Object *i*	0	*c*	*d*	*c*+*d*
	sum	*a*+*c*	*b*+*d*	*p*

- Simple matching coefficient (invariant, if the binary variable is *symmetric*): $d(i,j)=\frac{b+c}{a+b+c+d}$
- Jaccard coefficient (noninvariant if the binary variable is *asymmetric*): $d(i,j)=\frac{b+c}{a+b+c}$

Dissimilarity between Binary Variables

- Example

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

  - gender is a symmetric attribute
  - the remaining attributes are asymmetric binary
  - let the values Y and P be set to 1, and the value N be set to 0

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

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

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

Nominal Variables

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

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

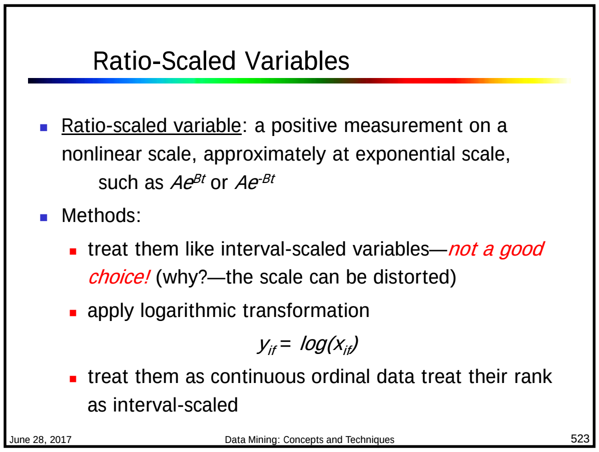
- Method 2: use a large number of binary variables
  - creating a new binary variable for each of the *M* nominal states

Ordinal Variables

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

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

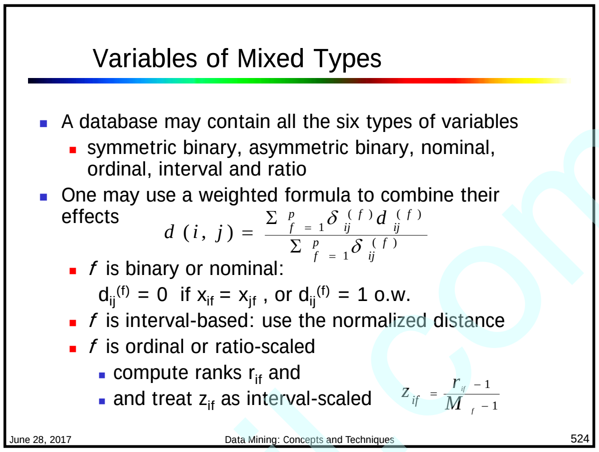
  - compute the dissimilarity using methods for interval-scaled variables



Ratio-Scaled Variables

- Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt}
- Methods:
  - treat them like interval-scaled variables—*not a good choice!* (why?—the scale can be distorted)
  - apply logarithmic transformation

    $y_{if} = log(x_{if})$
  - treat them as continuous ordinal data treat their rank as interval-scaled



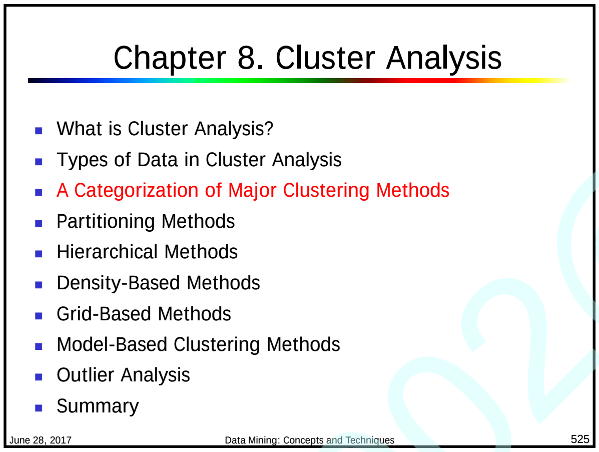
Variables of Mixed Types

- A database may contain all the six types of variables
  - symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio
- One may use a weighted formula to combine their effects

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

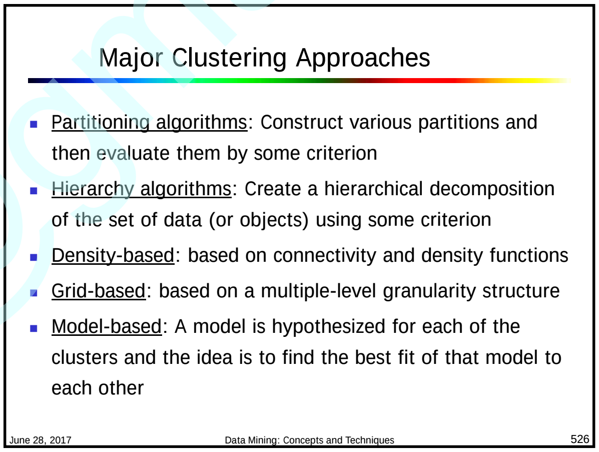
  - *f* is binary or nominal:

    $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ o.w.
  - *f* is interval-based: use the normalized distance
  - *f* is ordinal or ratio-scaled
    - compute ranks r_{if} and
    - and treat z_{if} as interval-scaled $z_{if} = \frac{r_{if} - 1}{M_f - 1}$



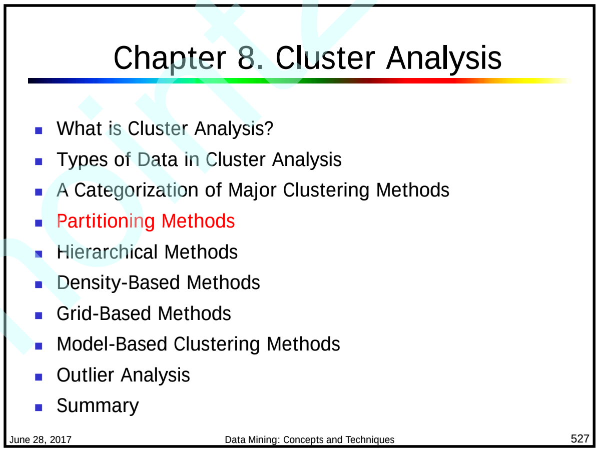
Chapter 8. Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary



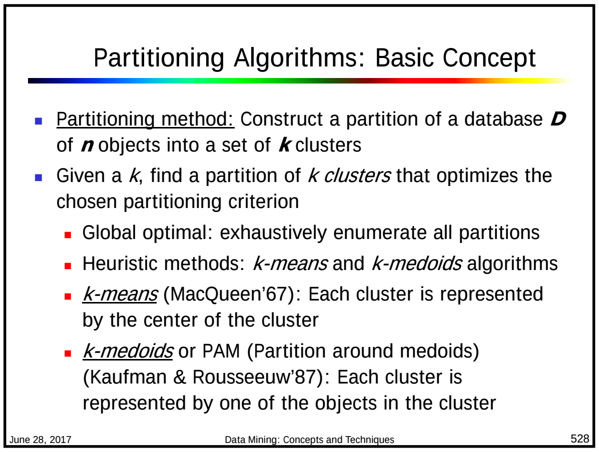
Major Clustering Approaches

- Partitioning algorithms: Construct various partitions and then evaluate them by some criterion
- Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion
- Density-based: based on connectivity and density functions
- Grid-based: based on a multiple-level granularity structure
- Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other



Chapter 8. Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary



Partitioning Algorithms: Basic Concept

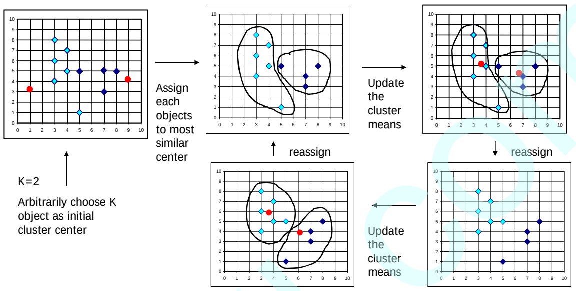
- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when no more new assignment

The *K-Means* Clustering Method

- Example



K=2

Arbitrarily choose K object as initial cluster center

Assign each objects to most similar center

Update the cluster means

reassign

reassign

Update the cluster means

Comments on the *K-Means* Method

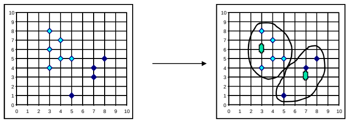
- Strength: *Relatively efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Comparing: PAM: O(k(n-k)^2), CLARA: O(ks^2 + k(n-k))
- Comment: Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined, then what about categorical data?
  - Need to specify *k,* the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

What is the problem of k-Means Method?

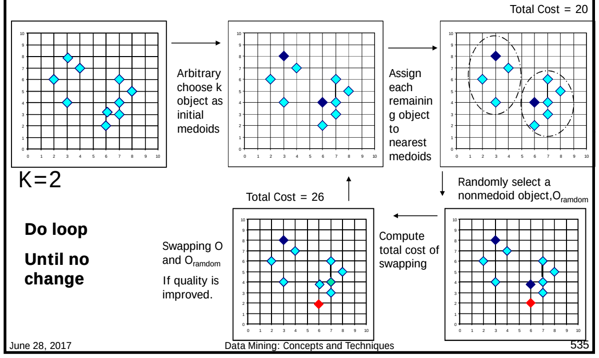
- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



The *K-Medoids* Clustering Method

- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids, 1987)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets
- *CLARA* (Kaufmann & Rousseeuw, 1990)
- *CLARANS* (Ng & Han, 1994): Randomized sampling
- Focusing + spatial data structure (Ester et al., 1995)

Typical k-medoids algorithm (PAM)


Total Cost = 20

K=2

Arbitrary choose k object as initial medoids

Assign each remaining object to nearest medoids

Randomly select a nonmedoid object,O_{ramdom}

Compute total cost of swapping

Total Cost = 26

Swapping O and O_{ramdom}

If quality is improved.

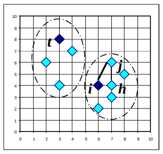
Do loop

Until no change

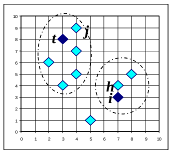
PAM (Partitioning Around Medoids) (1987)

- PAM (Kaufman and Rousseeuw, 1987), built in Splus
- Use real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - For each pair of non-selected object ***h*** and selected object ***i***, calculate the total swapping cost TC_{ih}
  - For each pair of ***i*** and ***h***,
    - If $TC_{ih} < 0$, ***i*** is replaced by ***h***
    - Then assign each non-selected object to the most similar representative object
  - repeat steps 2-3 until there is no change

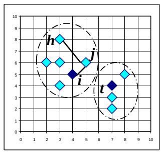
PAM Clustering: Total swapping cost $TC_{ih}=\sum_j C_{jih}$



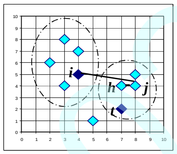
$C_{jih} = d(j, h) - d(j, i)$



$C_{jih} = 0$



$C_{jih} = d(j, t) - d(j, i)$



$C_{jih} = d(j, h) - d(j, t)$

What is the problem with PAM?

- Pam is more robust than k-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean
- Pam works efficiently for small data sets but does not **scale well** for large data sets.
  - $O(k(n-k)^2)$ for each iteration

    where n is # of data,k is # of clusters

➔Sampling based method,

CLARA(Clustering LARge Applications)

CLARA (Clustering Large Applications) (1990)

- *CLARA* (Kaufmann and Rousseeuw in 1990)
  - Built in statistical analysis packages, such as S+
- It draws *multiple samples* of the data set, applies *PAM* on each sample, and gives the best clustering as the output
- Strength: deals with larger data sets than *PAM*
- Weakness:
  - Efficiency depends on the sample size
  - A good clustering based on samples will not necessarily represent a good clustering of the whole data set if the sample is biased

CLARANS ("Randomized" CLARA) *(1994)*

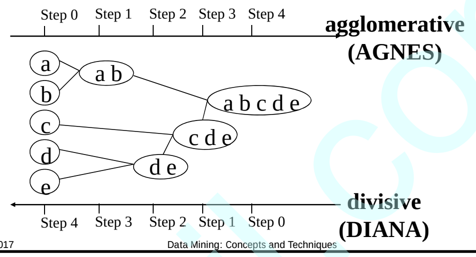
- *CLARANS* (A Clustering Algorithm based on Randomized Search) (Ng and Han'94)
- CLARANS draws sample of neighbors dynamically
- The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of *k* medoids
- If the local optimum is found, *CLARANS* starts with new randomly selected node in search for a new local optimum
- It is more efficient and scalable than both *PAM* and *CLARA*
- Focusing techniques and spatial access structures may further improve its performance (Ester et al.'95)

Chapter 8. Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

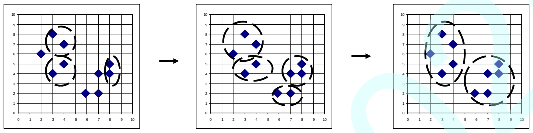
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition

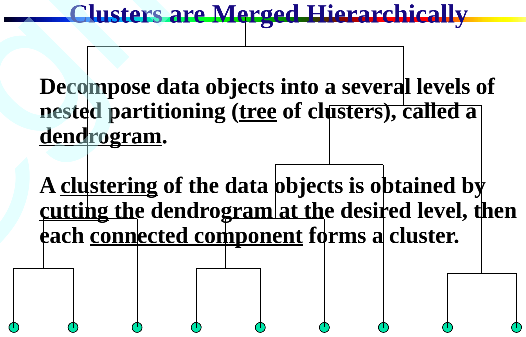


AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



A *Dendrogram* Shows How the Clusters are Merged Hierarchically

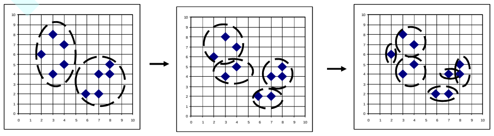


Decompose data objects into a several levels of nested partitioning (<u>tree</u> of clusters), called a <u>dendrogram</u>.

A <u>clustering</u> of the data objects is obtained by <u>cutting the dendrogram</u> at the desired level, then each <u>connected component</u> forms a cluster.

DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



More on Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - <u>do not scale</u> well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - <u>BIRCH (1996)</u>: uses CF-tree and incrementally adjusts the quality of sub-clusters
  - <u>CURE (1998)</u>: selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - <u>CHAMELEON (1999)</u>: hierarchical clustering using dynamic modeling

BIRCH (1996)

- Birch: Balanced Iterative Reducing and Clustering using Hierarchies, by Zhang, Ramakrishnan, Livny (SIGMOD'96)
- Incrementally construct a CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
  - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data)
  - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree
- *Scales linearly*: finds a good clustering with a single scan and improves the quality with a few additional scans
- *Weakness:* handles only numeric data, and sensitive to the order of the data record.

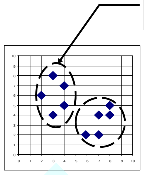
Clustering Feature Vector

Clustering Feature: $\boldsymbol{CF = (N, \overrightarrow{LS}, SS)}$

N: **Number of data points**

$LS: \sum_{i=1}^{N} \overrightarrow{X_i}$

$SS: \sum_{i=1}^{N} X_i^2$



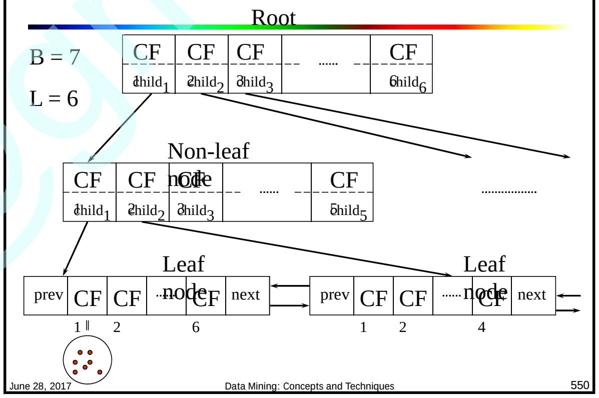
CF = (5, (16,30),(54,190))

(3,4)
(2,6)
(4,5)
(4,7)
(3,8)

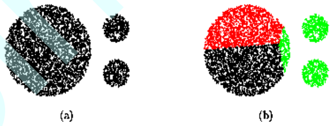
CF-Tree in BIRCH

- Clustering feature:
  - summary of the statistics for a given subcluster: the 0-th, 1st and 2nd moments of the subcluster from the statistical point of view.
  - registers crucial measurements for computing cluster and utilizes storage efficiently
- A CF tree is a height-balanced tree that stores the clustering features for a hierarchical clustering
  - A nonleaf node in a tree has descendants or "children"
  - The nonleaf nodes store sums of the CFs of their children
- A CF tree has two parameters
  - Branching factor: specify the maximum number of children.
  - threshold: max diameter of sub-clusters stored at the leaf nodes

CF Tree

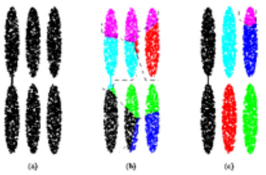

Root
B = 7
L = 6
CF1 CF2 CF3 ...... CF6
child1 child2 child3 child6
Non-leaf node
CF1 CF2 CF3 ...... CF5
child1 child2 child3 child5
....
Leaf node
prev CF1 CF2 ...... CF6 next
Leaf node
prev CF1 CF2 ...... CF4 next

CURE (Clustering Using REpresentatives)

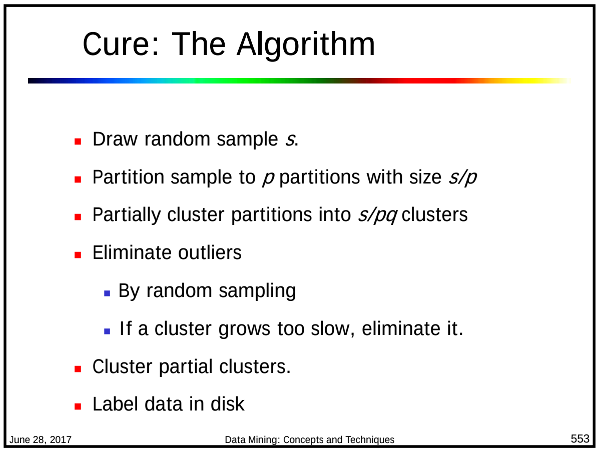


- CURE: proposed by Guha, Rastogi & Shim, 1998
  - Stops the creation of a cluster hierarchy if a level consists of *k* clusters
  - Uses multiple representative points to evaluate the distance between clusters, adjusts well to arbitrary shaped clusters and avoids single-link effect

Drawbacks of Distance-Based Method

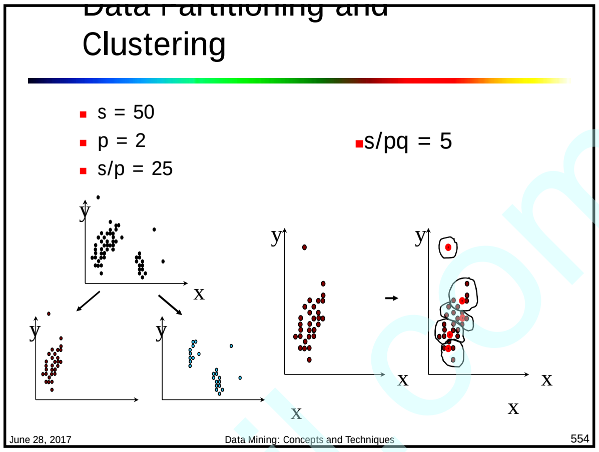


- Drawbacks of square-error based clustering method
  - Consider only one point as representative of a cluster
  - Good only for convex shaped, similar size and density, and if *k* can be reasonably estimated



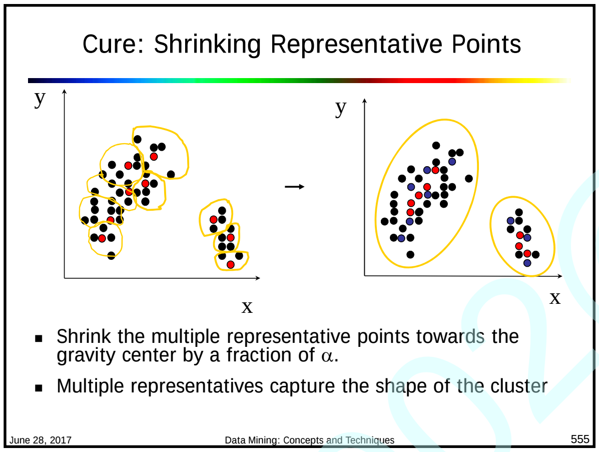
Cure: The Algorithm

- Draw random sample *s.*
- Partition sample to *p* partitions with size *s/p*
- Partially cluster partitions into *s/pq* clusters
- Eliminate outliers
  - By random sampling
  - If a cluster grows too slow, eliminate it.
- Cluster partial clusters.
- Label data in disk



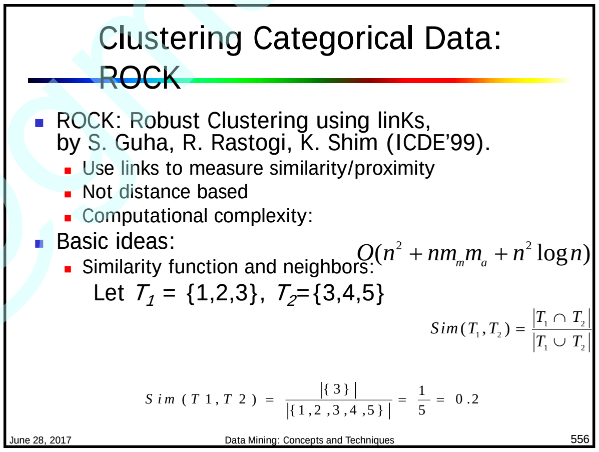
Data Partitioning and Clustering

- s = 50
- p = 2
- s/p = 25
- s/pq = 5



Cure: Shrinking Representative Points

- Shrink the multiple representative points towards the gravity center by a fraction of α.
- Multiple representatives capture the shape of the cluster



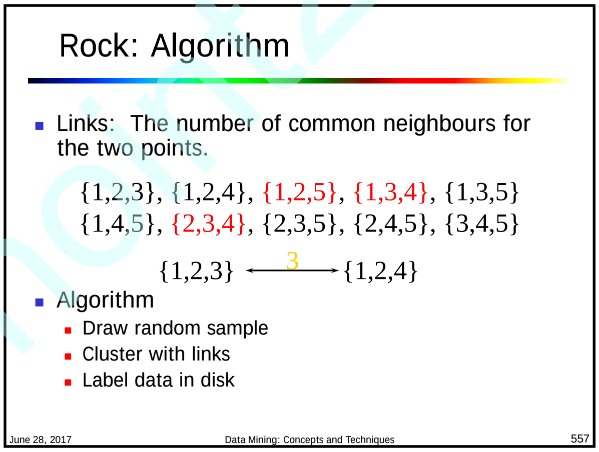
Clustering Categorical Data: ROCK

- ROCK: Robust Clustering using linKs, by S. Guha, R. Rastogi, K. Shim (ICDE'99).
  - Use links to measure similarity/proximity
  - Not distance based
  - Computational complexity: $O(n^2 + nm_m m_a + n^2 \log n)$
- Basic ideas:
  - Similarity function and neighbors:

    Let T_1 = {1,2,3}, T_2={3,4,5}

$$Sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

$$Sim(T1, T2) = \frac{|\{3\}|}{|\{1,2,3,4,5\}|} = \frac{1}{5} = 0.2$$



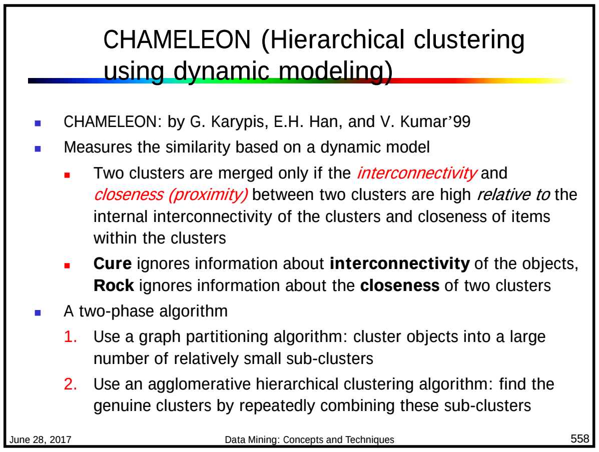
Rock: Algorithm

- Links: The number of common neighbours for the two points.

{1,2,3}, {1,2,4}, {1,2,5}, {1,3,4}, {1,3,5}
{1,4,5}, {2,3,4}, {2,3,5}, {2,4,5}, {3,4,5}

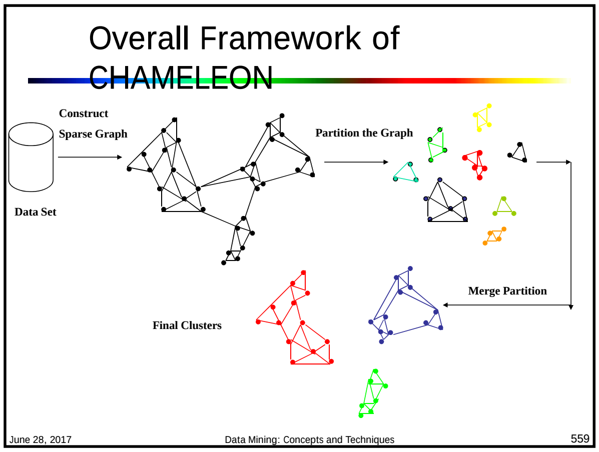
$$\{1,2,3\} \xleftrightarrow{3} \{1,2,4\}$$

- Algorithm
  - Draw random sample
  - Cluster with links
  - Label data in disk



CHAMELEON (Hierarchical clustering using dynamic modeling)

- CHAMELEON: by G. Karypis, E.H. Han, and V. Kumar'99
- Measures the similarity based on a dynamic model
  - Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters
  - **Cure** ignores information about **interconnectivity** of the objects, **Rock** ignores information about the **closeness** of two clusters
- A two-phase algorithm
  1. Use a graph partitioning algorithm: cluster objects into a large number of relatively small sub-clusters
  2. Use an agglomerative hierarchical clustering algorithm: find the genuine clusters by repeatedly combining these sub-clusters



Overall Framework of CHAMELEON

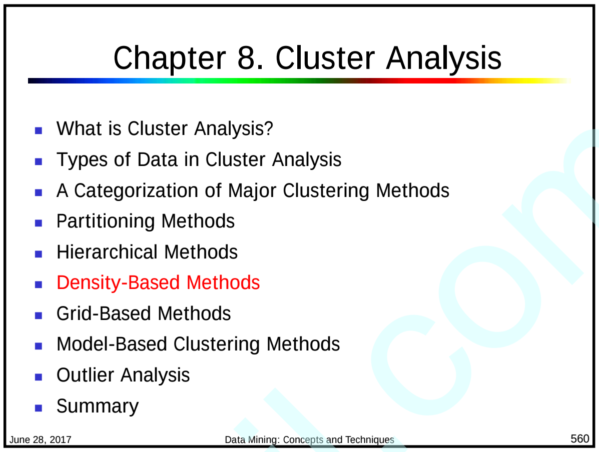
Construct Sparse Graph

Data Set

Partition the Graph

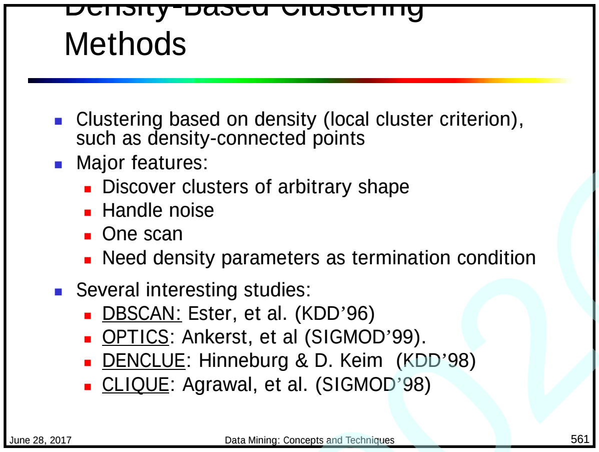
Merge Partition

Final Clusters



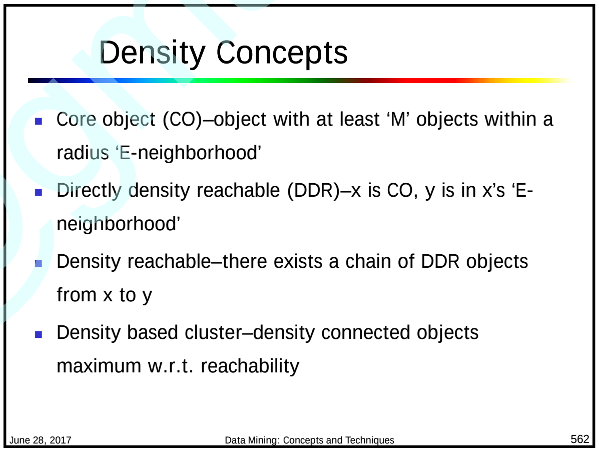
Chapter 8. Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary



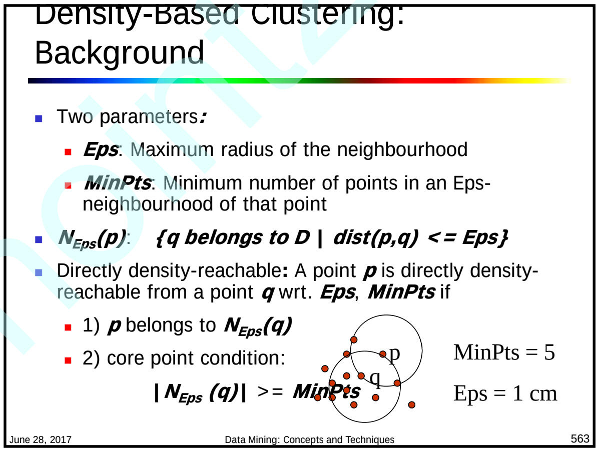
Density-Based Clustering Methods

- Clustering based on density (local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)



Density Concepts

- Core object (CO)–object with at least 'M' objects within a radius 'E-neighborhood'
- Directly density reachable (DDR)–x is CO, y is in x's 'E-neighborhood'
- Density reachable–there exists a chain of DDR objects from x to y
- Density based cluster–density connected objects maximum w.r.t. reachability



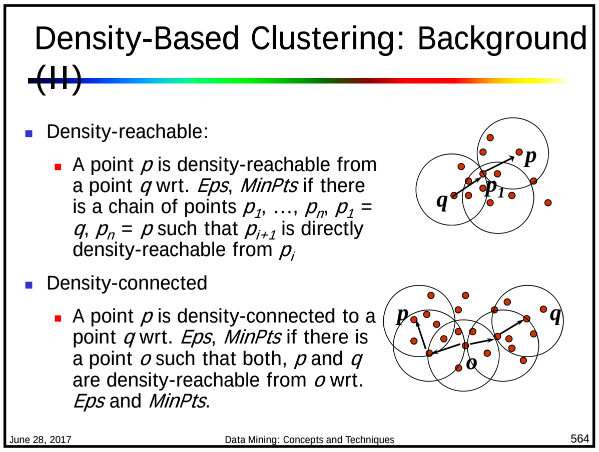
Density-Based Clustering: Background

- Two parameters**:**
  - ***Eps***: Maximum radius of the neighbourhood
  - ***MinPts***: Minimum number of points in an Eps-neighbourhood of that point
- ***$N_{Eps}(p)$***: ***{q belongs to D | dist(p,q) <= Eps}***
- Directly density-reachable**:** A point ***p*** is directly density-reachable from a point ***q*** wrt. ***Eps***, ***MinPts*** if
  - 1) ***p*** belongs to ***$N_{Eps}(q)$***
  - 2) core point condition:

    $|N_{Eps}(q)| >= MinPts$

MinPts = 5

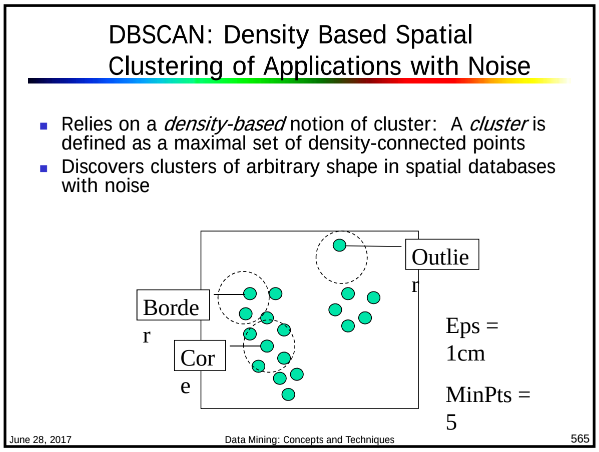
Eps = 1 cm



Density-Based Clustering: Background (II)

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

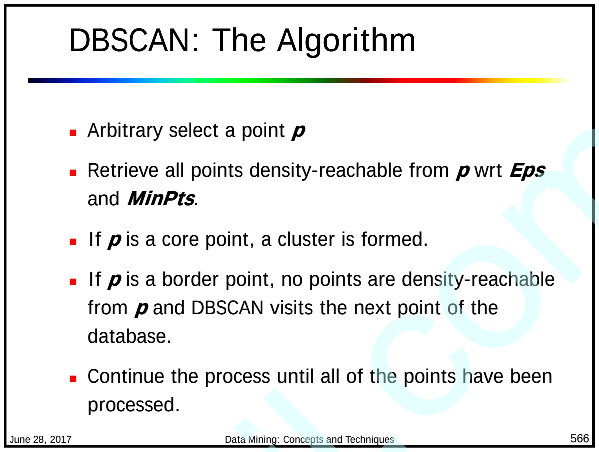
- Density-connected
  - A point *p* is density-connected to a point *q* wrt. *Eps*, *MinPts* if there is a point *o* such that both, *p* and *q* are density-reachable from *o* wrt. *Eps* and *MinPts*.



DBSCAN: Density Based Spatial Clustering of Applications with Noise

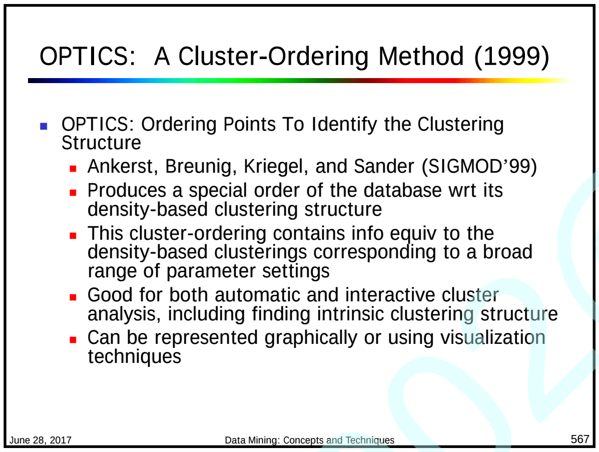
- Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape in spatial databases with noise

Outlie
r
Borde
r
Cor
e
Eps =
1cm
MinPts =
5



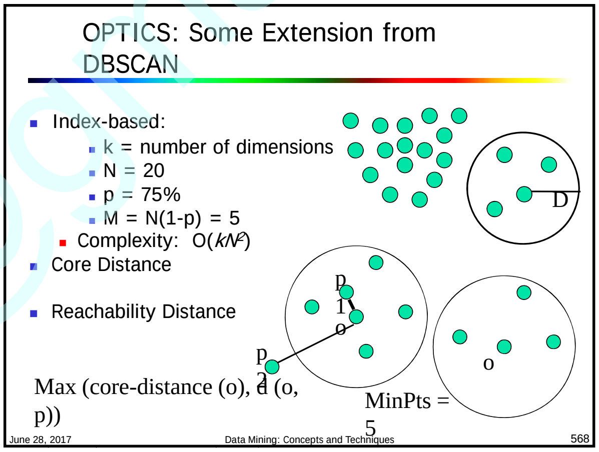
DBSCAN: The Algorithm

- Arbitrary select a point ***p***
- Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.
- If ***p*** is a core point, a cluster is formed.
- If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.
- Continue the process until all of the points have been processed.



OPTICS: A Cluster-Ordering Method (1999)

- OPTICS: Ordering Points To Identify the Clustering Structure
  - Ankerst, Breunig, Kriegel, and Sander (SIGMOD'99)
  - Produces a special order of the database wrt its density-based clustering structure
  - This cluster-ordering contains info equiv to the density-based clusterings corresponding to a broad range of parameter settings
  - Good for both automatic and interactive cluster analysis, including finding intrinsic clustering structure
  - Can be represented graphically or using visualization techniques

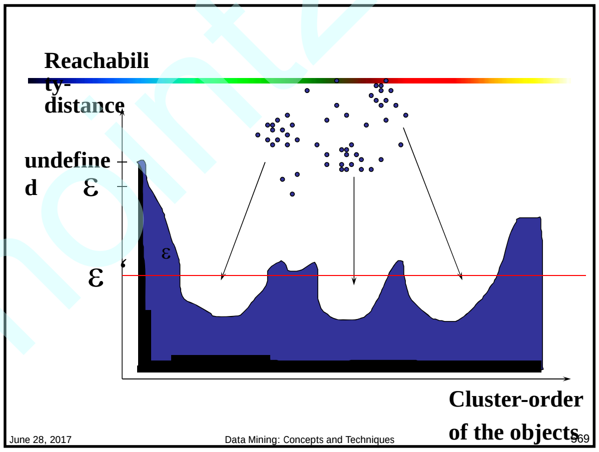


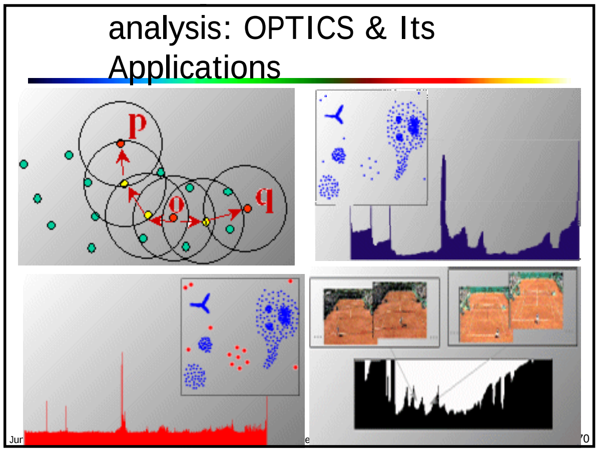
OPTICS: Some Extension from DBSCAN

- Index-based:
  - k = number of dimensions
  - N = 20
  - p = 75%
  - M = N(1-p) = 5
  - Complexity: $O(kN^2)$
- Core Distance
- Reachability Distance

Max (core-distance (o), d (o, p))

D
p
1
o
p
2
o
MinPts =
5





analysis: OPTICS & Its Applications

DENCLUE: Using density functions

- DENsity-based CLUstEring by Hinneburg & Keim (KDD'98)
- Major features
  - Solid mathematical foundation
  - Good for data sets with large amounts of noise
  - Allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets
  - Significant faster than existing algorithm (faster than DBSCAN by a factor of up to 45)
  - But needs a large number of parameters

Denclue: Technical Essence

- Uses grid cells but only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure.
- Influence function: describes the impact of a data point within its neighborhood.
- Overall density of the data space can be calculated as the sum of the influence function of all data points.
- Clusters can be determined mathematically by identifying density attractors.
- Density attractors are local maximal of the overall density function.

Gradient: The steepness of a slope

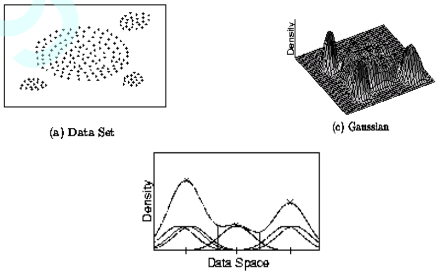
- Example

$$f_{Gaussian}(x,y) = e^{-\frac{d(x,y)^2}{2\sigma^2}}$$

$$f^D_{Gaussian}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

$$\nabla f^D_{Gaussian}(x,x_i) = \sum_{i=1}^{N} (x_i - x) \cdot e^{-\frac{d(x,x_i)^2}{2\sigma^2}}$$

Density Attractor



Center-Defined and Arbitrary

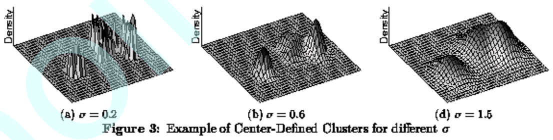


Figure 3: Example of Center-Defined Clusters for different σ

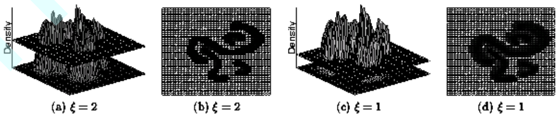


Figure 4: Example of Arbitray-Shape Clusters for different ξ

Chapter 8. Cluster Analysis

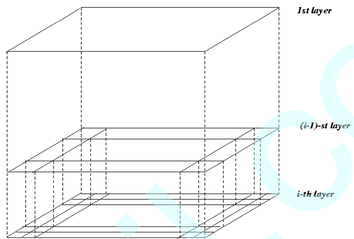
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Grid-Based Clustering Method

- Using multi-resolution grid data structure
- Several interesting methods
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
    - A multi-resolution clustering approach using wavelet method
  - CLIQUE: Agrawal, et al. (SIGMOD'98)

STING: A Statistical Information Grid Approach

- Wang, Yang and Muntz (VLDB'97)
- The spatial area area is divided into rectangular cells
- There are several levels of cells corresponding to different levels of resolution



STING: A Statistical Information Grid Approach (2)

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from parameters of lower level cell
  - *count*, *mean*, *s*, *min*, *max*
  - type of distribution—normal, *uniform*, etc.
- Use a top-down approach to answer spatial data queries
- Start from a pre-selected layer—typically with a small number of cells
- For each cell in the current level compute the confidence interval

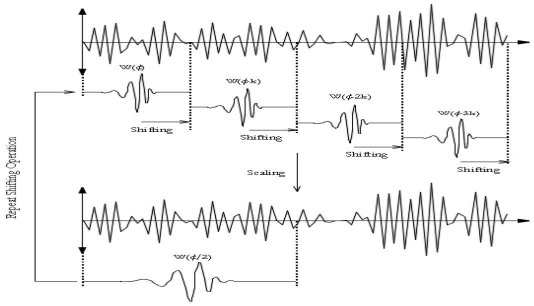
STING: A Statistical Information Grid Approach (3)

- Remove the irrelevant cells from further consideration
- When finish examining the current layer, proceed to the next lower level
- Repeat this process until the bottom layer is reached
- Advantages:
  - Query-independent, easy to parallelize, incremental update
  - *O(K),* where *K* is the number of grid cells at the lowest level
- Disadvantages:
  - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected

WaveCluster (1998)

- Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
- A multi-resolution clustering approach which applies wavelet transform to the feature space
  - A wavelet transform is a signal processing technique that decomposes a signal into different frequency sub-band.
- Both grid-based and density-based
- Input parameters:
  - # of grid cells for each dimension
  - the wavelet, and the # of applications of wavelet transform.

What is Wavelet (1)?



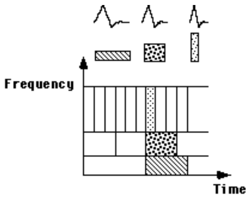
WaveCluster (1998)

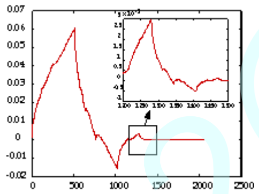
- How to apply wavelet transform to find clusters
  - Summaries the data by imposing a multidimensional grid structure onto data space
  - These multidimensional spatial data objects are represented in a n-dimensional feature space
  - Apply wavelet transform on feature space to find the dense regions in the feature space
  - Apply wavelet transform multiple times which result in clusters at different scales from fine to coarse

Wavelet Transform

- Decomposes a signal into different frequency subbands. (can be applied to n-dimensional signals)
- Data are transformed to preserve relative distance between objects at different levels of resolution.
- Allows natural clusters to become more distinguishable

What Is Wavelet (2)?





Quantization

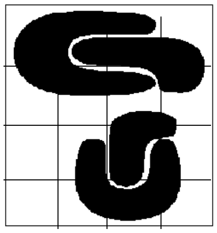
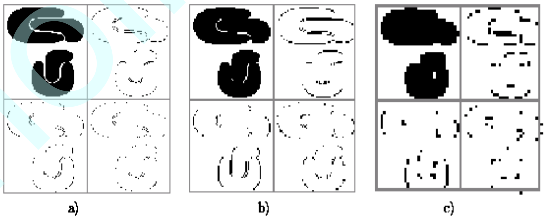


Figure 1: A sample 2-dimensional feature space.

Transformation



WaveCluster (1998)

- Why is wavelet transformation useful for clustering
  - Unsupervised clustering

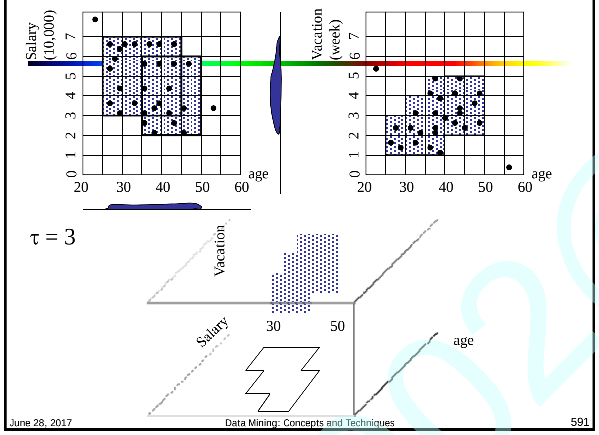
    It uses hat-shape filters to emphasize region where points cluster, but simultaneously to suppress weaker information in their boundary
  - Effective removal of outliers
  - Multi-resolution
  - Cost efficiency
- Major features:
  - Complexity O(N)
  - Detect arbitrary shaped clusters at different scales
  - Not sensitive to noise, not sensitive to input order
  - Only applicable to low dimensional data

CLIQUE (Clustering In QUEst)

- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98).
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space
- CLIQUE can be considered as both density-based and grid-based
  - It partitions each dimension into the same number of equal length interval
  - It partitions an m-dimensional data space into non-overlapping rectangular units
  - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - A cluster is a maximal set of connected dense units within a subspace

CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.
- Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters:
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests.
- Generate minimal description for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determination of minimal cover for each cluster



Strength and Weakness of *CLIQUE*

- Strength
  - It *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces
  - It is *insensitive* to the order of records in input and does not presume some canonical data distribution
  - It scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases
- Weakness
  - The accuracy of the clustering result may be degraded at the expense of simplicity of the method

Chapter 8. Cluster Analysis

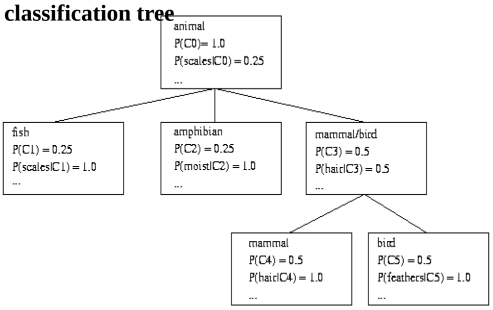
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Model-Based Clustering Methods

- Attempt to optimize the fit between the data and some mathematical model
- Statistical and AI approach
  - Conceptual clustering
    - A form of clustering in machine learning
    - Produces a classification scheme for a set of unlabeled objects
    - Finds characteristic description for each concept (class)
  - COBWEB (Fisher'87)
    - A popular a simple method of incremental conceptual learning
    - Creates a hierarchical clustering in the form of a classification tree
    - Each node refers to a concept and contains a probabilistic description of that concept

COBWEB Clustering Method

A classification tree


animal
P(C0)= 1.0
P(scales|C0) = 0.25
...

fish
P(C1) = 0.25
P(scales|C1) = 1.0
...

amphibian
P(C2) = 0.25
P(moist|C2) = 1.0
...

mammal/bird
P(C3) = 0.5
P(hair|C3) = 0.5
...

mammal
P(C4) = 0.5
P(hair|C4) = 1.0
...

bird
P(C5) = 0.5
P(feathers|C5) = 1.0
...

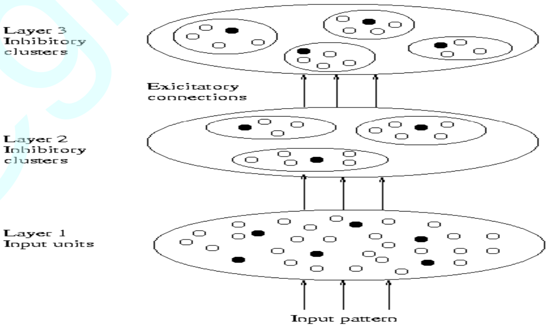
More on Statistical-Based Clustering

- Limitations of COBWEB
  - The assumption that the attributes are independent of each other is often too strong because correlation may exist
  - Not suitable for clustering large database data – skewed tree and expensive probability distributions
- CLASSIT
  - an extension of COBWEB for incremental clustering of continuous data
  - suffers similar problems as COBWEB
- AutoClass (Cheeseman and Stutz, 1996)
  - Uses Bayesian statistical analysis to estimate the number of clusters
  - Popular in industry

Other Model-Based Clustering Methods

- Neural network approaches
  - Represent each cluster as an exemplar, acting as a "prototype" of the cluster
  - New objects are distributed to the cluster whose exemplar is the most similar according to some dostance measure
- Competitive learning
  - Involves a hierarchical architecture of several units (neurons)
  - Neurons compete in a "winner-takes-all" fashion for the object currently being presented

Model-Based Clustering Methods



Self-organizing feature maps (SOMs)

- Clustering is also performed by having several units competing for the current object
- The unit whose weight vector is closest to the current object wins
- The winner and its neighbors learn by having their weights adjusted
- SOMs are believed to resemble processing that can occur in the brain
- Useful for visualizing high-dimensional data in 2- or 3-D space

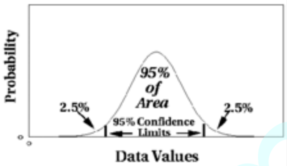
Chapter 8. Cluster Analysis

- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

What Is Outlier Discovery?

- What are outliers?
  - The set of objects are considerably dissimilar from the remainder of the data
  - Example: Sports: Michael Jordon, Wayne Gretzky, ...
- Problem
  - Find top n outlier points
- Applications:
  - Credit card fraud detection
  - Telecom fraud detection
  - Customer segmentation
  - Medical analysis

Statistical Approaches



- Assume a model underlying distribution that generates data set (e.g. normal distribution)
- Use discordancy tests depending on
  - data distribution
  - distribution parameter (e.g., mean, variance)
  - number of expected outliers
- Drawbacks
  - most tests are for single attribute
  - In many cases, data distribution may not be known

Outlier Discovery: Distance-Based Approach

- Introduced to counter the main limitations imposed by statistical methods
  - We need multi-dimensional analysis without knowing data distribution.
- Distance-based outlier: A DB(p, D)-outlier is an object O in a dataset T such that at least a fraction p of the objects in T lies at a distance greater than D from O
- Algorithms for mining distance-based outliers
  - Index-based algorithm
  - Nested-loop algorithm
  - Cell-based algorithm

Outlier Discovery: Deviation-Based Approach

- Identifies outliers by examining the main characteristics of objects in a group
- Objects that "deviate" from this description are considered outliers
- sequential exception technique
  - simulates the way in which humans can distinguish unusual objects from among a series of supposedly like objects
- OLAP data cube technique
  - uses data cubes to identify regions of anomalies in large multidimensional data

Chapter 8. Cluster Analysis

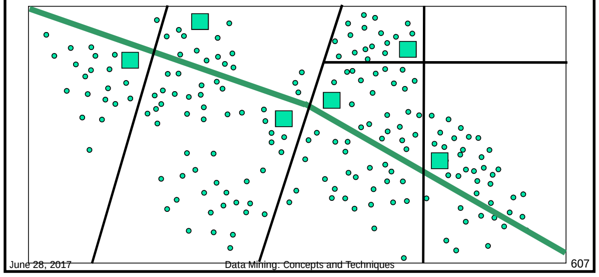
- What is Cluster Analysis?
- Types of Data in Cluster Analysis
- A Categorization of Major Clustering Methods
- Partitioning Methods
- Hierarchical Methods
- Density-Based Methods
- Grid-Based Methods
- Model-Based Clustering Methods
- Outlier Analysis
- Summary

Problems and Challenges

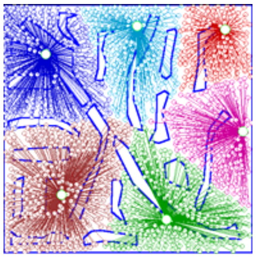
- Considerable progress has been made in scalable clustering methods
  - Partitioning: k-means, k-medoids, CLARANS
  - Hierarchical: BIRCH, CURE
  - Density-based: DBSCAN, CLIQUE, OPTICS
  - Grid-based: STING, WaveCluster
  - Model-based: Autoclass, Denclue, Cobweb
- Current clustering techniques do not address all the requirements adequately
- Constraint-based clustering analysis: Constraints exist in data space (bridges and highways) or in user queries

Constraint-Based Clustering Analysis

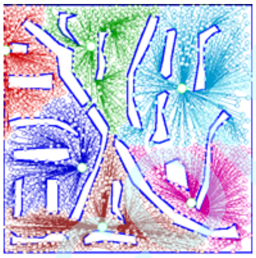
- Clustering analysis: less parameters but more user-desired constraints, e.g., an ATM allocation problem



Clustering With Obstacle Objects



Not Taking obstacles into account



Taking obstacles into account

Summary

- Cluster analysis groups objects based on their similarity and has wide applications
- Measure of similarity can be computed for various types of data
- Clustering algorithms can be categorized into partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model-based methods
- Outlier detection and analysis are very useful for fraud detection, etc. and can be performed by statistical, distance-based or deviation-based approaches
- There are still lots of research issues on cluster analysis, such as constraint-based clustering

References (1)

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. SIGMOD'98
- M. R. Anderberg. Cluster Analysis for Applications. Academic Press, 1973.
- M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, SIGMOD'99.
- P. Arabie, L. J. Hubert, and G. De Soete. Clustering and Classification. World Scietific, 1996
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases. KDD'96.
- M. Ester, H.-P. Kriegel, and X. Xu. Knowledge discovery in large spatial databases: Focusing techniques for efficient class identification. SSD'95.
- D. Fisher. Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2:139-172, 1987.
- D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamic systems. In Proc. VLDB'98.
- S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.
- A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Printice Hall, 1988.

References (2)

- L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley & Sons, 1990.
- E. Knorr and R. Ng. Algorithms for mining distance-based outliers in large datasets. VLDB'98.
- G. J. McLachlan and K.E. Bkasford. Mixture Models: Inference and Applications to Clustering. John Wiley and Sons, 1988.
- P. Michaud. Clustering techniques. Future Generation Computer systems, 13, 1997.
- R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. VLDB'94.
- E. Schikuta. Grid clustering: An efficient hierarchical clustering method for very large data sets. Proc. 1996 Int. Conf. on Pattern Recognition, 101-105.
- G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. VLDB'98.
- W. Wang, Yang, R. Muntz, STING: A Statistical Information grid Approach to Spatial Data Mining, VLDB'97.
- T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH : an efficient data clustering method for very large databases. SIGMOD'96.

Data Mining: Concepts and Techniques

— Slides for Textbook —

— Chapter 9 —

©Jawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj

Chapter 9. Mining Complex Types of Data

- Mining spatial databases
- Mining multimedia databases
- Mining time-series and sequence data
- Mining stream data
- Mining text databases
- Mining the World-Wide Web
- Summary

Spatial Data Warehousing

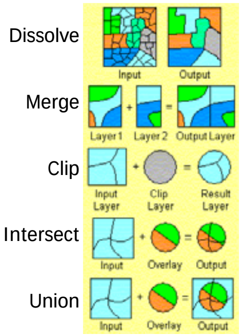
- Spatial data warehouse: Integrated, subject-oriented, time-variant, and nonvolatile spatial data repository
- Spatial data integration: a big issue
  - Structure-specific formats (raster- vs. vector-based, OO vs. relational models, different storage and indexing, etc.)
  - Vendor-specific formats (ESRI, MapInfo, Integraph, IDRISI, etc.)
  - Geo-specific formats (geographic vs. equal area projection, etc.)
- Spatial data cube: multidimensional spatial database
  - Both dimensions and measures may contain spatial components

Dimensions and Measures in Spatial Data Warehouse

- Dimensions
  - non-spatial
    - e.g. *"25-30 degrees"* generalizes to *"hot"* (both are strings)
  - spatial-to-nonspatial
    - e.g. *Seattle* generalizes to description *"Pacific Northwest"* (as a string)
  - spatial-to-spatial
    - e.g. *Seattle* generalizes to *Pacific Northwest* (as a spatial region)
- Measures
  - numerical (e.g. monthly revenue of a region)
    - distributive (e.g. count, sum)
    - algebraic (e.g. average)
    - holistic (e.g. median, rank)
  - spatial
    - collection of spatial pointers (e.g. pointers to all regions with temperature of 25-30 degrees in July)

Spatial-to-Spatial Generalization

- Generalize detailed geographic points into clustered regions, such as businesses, residential, industrial, or agricultural areas, according to land usage
- Requires the merging of a set of geographic areas by spatial operations

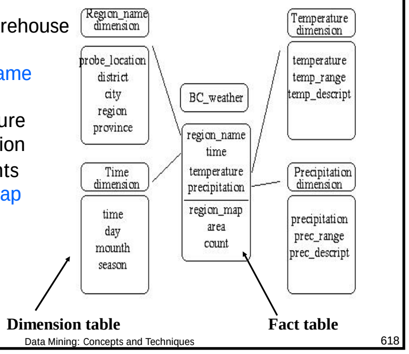


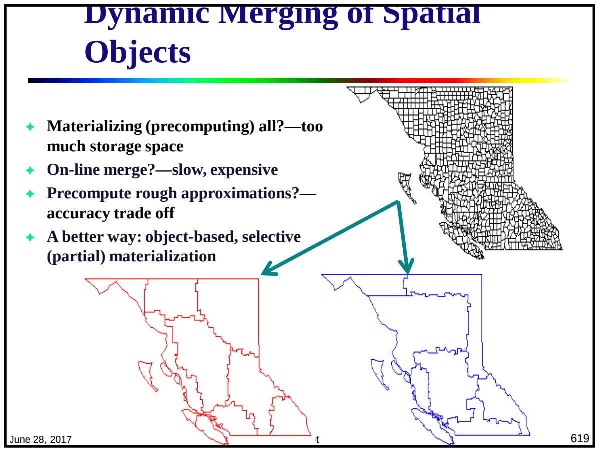
Example: British Columbia Weather Pattern Analysis

- Input
  - A map with about 3,000 weather probes scattered in B.C.
  - Daily data for temperature, precipitation, wind velocity, etc.
  - Data warehouse using star schema
- Output
  - A map that reveals patterns: merged (similar) regions
- Goals
  - Interactive analysis (drill-down, slice, dice, pivot, roll-up)
  - Fast response time
  - Minimizing storage space used
- Challenge
  - A merged region may contain hundreds of "primitive" regions (polygons)

Star Schema of the BC Weather Warehouse

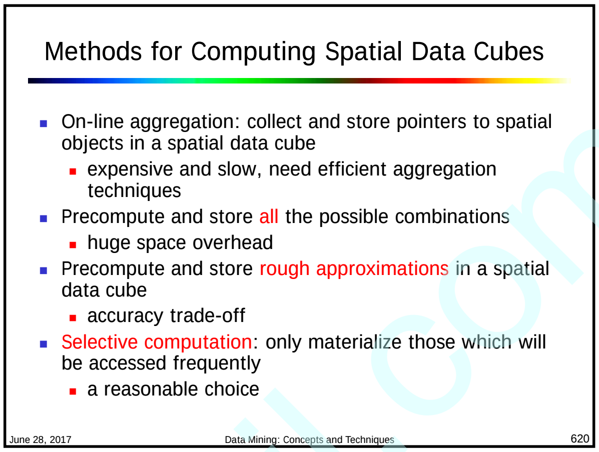
- Spatial data warehouse
  - Dimensions
    - region_name
    - time
    - temperature
    - precipitation
  - Measurements
    - region_map
    - area
    - count





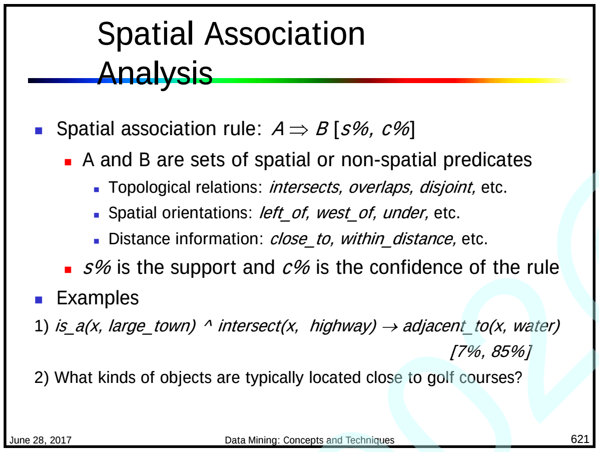
Dynamic Merging of Spatial Objects

- Materializing (precomputing) all?—too much storage space
- On-line merge?—slow, expensive
- Precompute rough approximations?—accuracy trade off
- A better way: object-based, selective (partial) materialization



Methods for Computing Spatial Data Cubes

- On-line aggregation: collect and store pointers to spatial objects in a spatial data cube
  - expensive and slow, need efficient aggregation techniques
- Precompute and store all the possible combinations
  - huge space overhead
- Precompute and store rough approximations in a spatial data cube
  - accuracy trade-off
- Selective computation: only materialize those which will be accessed frequently
  - a reasonable choice

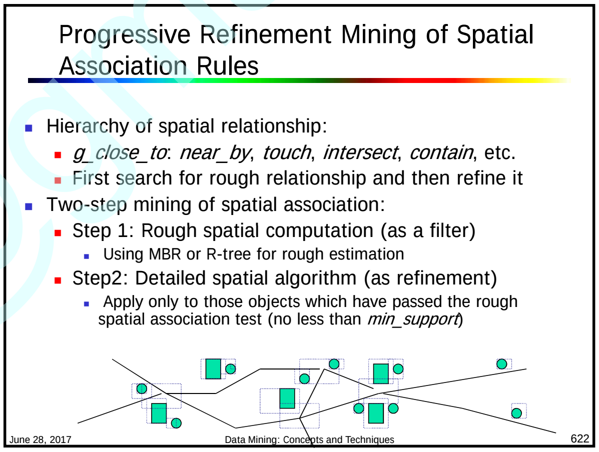


Spatial Association Analysis

- Spatial association rule: $A \Rightarrow B$ [$s\%$, $c\%$]
  - A and B are sets of spatial or non-spatial predicates
    - Topological relations: *intersects, overlaps, disjoint,* etc.
    - Spatial orientations: *left_of, west_of, under,* etc.
    - Distance information: *close_to, within_distance,* etc.
  - *s%* is the support and *c%* is the confidence of the rule
- Examples

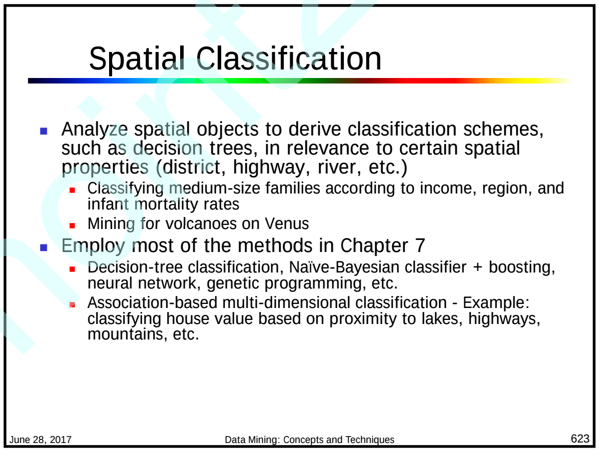
1) *is_a(x, large_town) ^ intersect(x, highway) → adjacent_to(x, water) [7%, 85%]*

2) What kinds of objects are typically located close to golf courses?



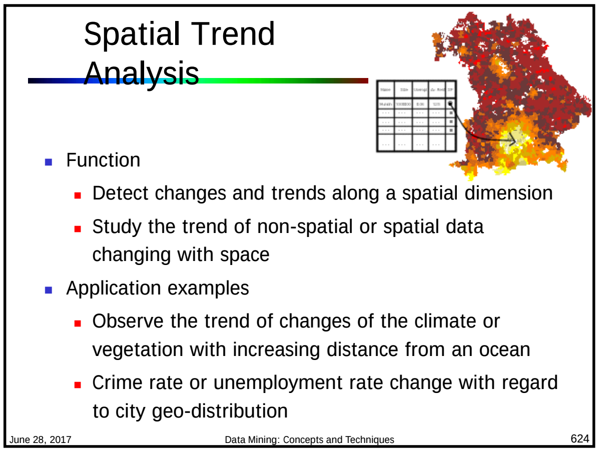
Progressive Refinement Mining of Spatial Association Rules

- Hierarchy of spatial relationship:
  - *g_close_to*: *near_by*, *touch*, *intersect*, *contain*, etc.
  - First search for rough relationship and then refine it
- Two-step mining of spatial association:
  - Step 1: Rough spatial computation (as a filter)
    - Using MBR or R-tree for rough estimation
  - Step2: Detailed spatial algorithm (as refinement)
    - Apply only to those objects which have passed the rough spatial association test (no less than *min_support*)



Spatial Classification

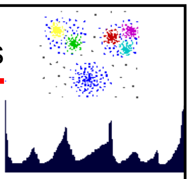
- Analyze spatial objects to derive classification schemes, such as decision trees, in relevance to certain spatial properties (district, highway, river, etc.)
  - Classifying medium-size families according to income, region, and infant mortality rates
  - Mining for volcanoes on Venus
- Employ most of the methods in Chapter 7
  - Decision-tree classification, Naïve-Bayesian classifier + boosting, neural network, genetic programming, etc.
  - Association-based multi-dimensional classification - Example: classifying house value based on proximity to lakes, highways, mountains, etc.



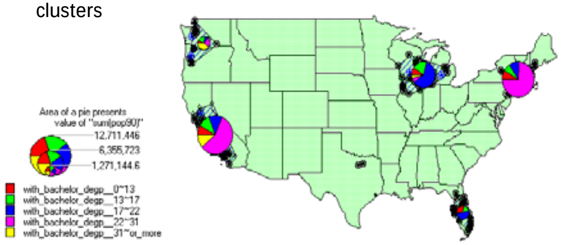
Spatial Trend Analysis

- Function
  - Detect changes and trends along a spatial dimension
  - Study the trend of non-spatial or spatial data changing with space
- Application examples
  - Observe the trend of changes of the climate or vegetation with increasing distance from an ocean
  - Crime rate or unemployment rate change with regard to city geo-distribution

Spatial Cluster Analysis



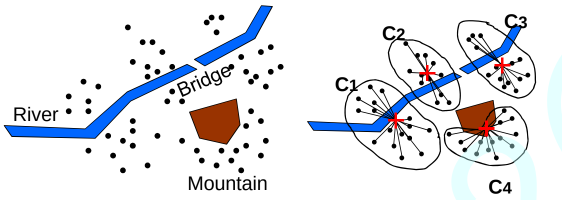
- Mining clusters—k-means, k-medoids, hierarchical, density-based, etc.
- Analysis of distinct features of the clusters


Area of a pie presents value of "sum(pop90)"
12,711,446
6,355,723
1,271,144.6
🟥 with_bachelor_degp__0~13
🟩 with_bachelor_degp__13~17
🟦 with_bachelor_degp__17~22
🟪 with_bachelor_degp__22~31
🟨 with_bachelor_degp__31~or_more

Constraints-Based Clustering

- Constraints on individual objects
  - Simple selection of relevant objects before clustering
- Clustering parameters as constraints
  - K-means, density-based: radius, min-# of points
- Constraints specified on clusters using SQL aggregates
  - Sum of the profits in each cluster > $1 million
- Constraints imposed by physical obstacles
  - Clustering with obstructed distance

Constraint-Based Clustering: Planning ATM Locations



Spatial data with obstacles

Clustering *without* taking obstacles into consideration

Chapter 9. Mining Complex Types of Data

- Mining spatial databases
- Mining multimedia databases
- Mining time-series and sequence data
- Mining stream data
- Mining text databases
- Mining the World-Wide Web
- Summary

Similarity Search in Multimedia Data

- Description-based retrieval systems
  - Build indices and perform object retrieval based on image descriptions, such as keywords, captions, size, and time of creation
  - Labor-intensive if performed manually
  - Results are typically of poor quality if automated
- Content-based retrieval systems
  - Support retrieval based on the image content, such as color histogram, texture, shape, objects, and wavelet transforms

Queries in Content-Based Retrieval Systems

- Image sample-based queries
  - Find all of the images that are similar to the given image sample
  - Compare the feature vector (signature) extracted from the sample with the feature vectors of images that have already been extracted and indexed in the image database
- Image feature specification queries
  - Specify or sketch image features like color, texture, or shape, which are translated into a feature vector
  - Match the feature vector with the feature vectors of the images in the database

Approaches Based on Image Signature

- Color histogram-based signature
  - The signature includes color histograms based on color composition of an image regardless of its scale or orientation
  - No information about shape, location, or texture
  - Two images with similar color composition may contain very different shapes or textures, and thus could be completely unrelated in semantics
- Multifeature composed signature
  - Define different distance functions for color, shape, location, and texture, and subsequently combine them to derive the overall result.

Wavelet Analysis

- Wavelet-based signature
  - Use the dominant wavelet coefficients of an image as its signature
  - Wavelets capture shape, texture, and location information in a single unified framework
  - Improved efficiency and reduced the need for providing multiple search primitives
  - May fail to identify images containing similar in location or size objects
- Wavelet-based signature with region-based granularity
  - Similar images may contain similar regions, but a region in one image could be a translation or scaling of a matching region in the other
  - Compute and compare signatures at the granularity of regions, not the entire image

Wavelet Analysis

- Wavelet-based signature
  - Use the dominant wavelet coefficients of an image as its signature
  - Wavelets capture shape, texture, and location information in a single unified framework
  - Improved efficiency and reduced the need for providing multiple search primitives
  - May fail to identify images containing similar objects that are in different locations.

One Signature for the Entire Image?

- Walnus: [NRS99] by Natsev, Rastogi, and Shim
- Similar images may contain similar regions, but a region in one image could be a translation or scaling of a matching region in the other

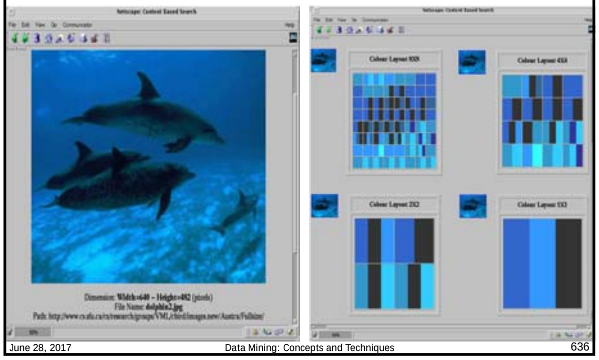
- Wavelet-based signature with region-based granularity
  - Define regions by clustering signatures of windows of varying sizes within the image
  - Signature of a region is the centroid of the cluster
  - Similarity is defined in terms of the fraction of the area of the two images covered by matching pairs of regions from two images

Analysis of Multimedia Data

- Multimedia data cube
  - Design and construction similar to that of traditional data cubes from relational data
  - Contain additional dimensions and measures for multimedia information, such as color, texture, and shape
- The database does not store images but their descriptors
  - Feature descriptor: a set of vectors for each visual characteristic
    - Color vector: contains the color histogram
    - MFC (Most Frequent Color) vector: five color centroids
    - MFO (Most Frequent Orientation) vector: five edge orientation centroids
  - Layout descriptor: contains a color layout vector and an edge layout vector

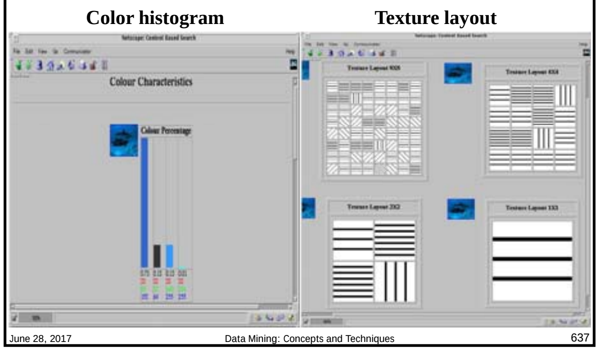
Multi-Dimensional Search in Multimedia Databases

Color layout



Dimension: Width=640 – Height=482 (pixels)
File Name: dolphin2.jpg
Path: http://www.cs.sfu.ca/cs/research/groups/VML/cbird/images.new/Austra/Fullsize/

Multi-Dimensional Analysis in Multimedia Databases

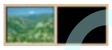


Mining Multimedia Databases

Refining or combining searches

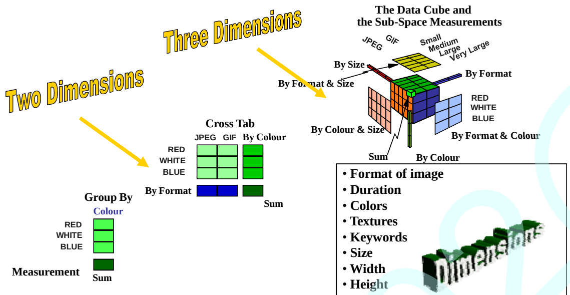
Search for "blue sky"
(top layout grid is blue)

Search for "airplane in blue sky"
(top layout grid is blue and
keyword = "airplane")



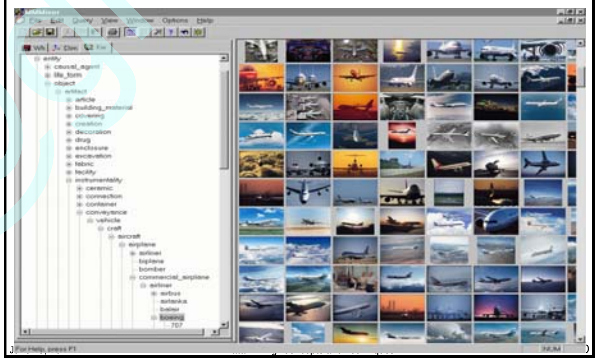
Search for "blue sky and
green meadows"
(top layout grid is blue
and bottom is green)

Mining Multimedia Databases

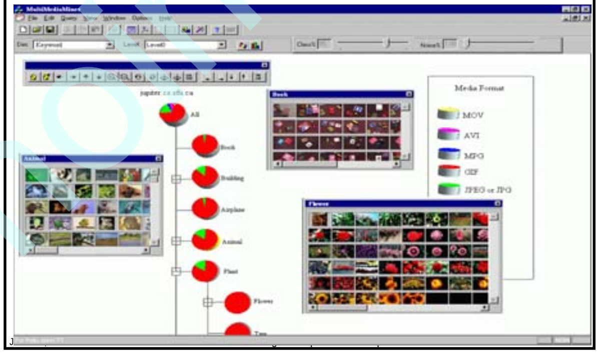


- Format of image
- Duration
- Colors
- Textures
- Keywords
- Size
- Width
- Height
- Internet domain of image
- Internet domain of parent pages
- Image popularity

Mining Multimedia Databases in MultiMediaMiner



Classification in MultiMediaMiner



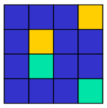
Mining Associations in Multimedia Data

- Special features:
  - Need # of occurrences besides Boolean existence, e.g.,
    - "Two red square and one blue circle" implies theme "air-show"
  - Need spatial relationships
    - Blue on top of white squared object is associated with brown bottom
  - Need multi-resolution and progressive refinement mining
    - It is expensive to explore detailed associations among objects at high resolution
    - It is crucial to ensure the completeness of search at multi-resolution space

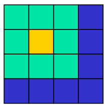
Mining Multimedia Databases

Spatial Relationships from Layout

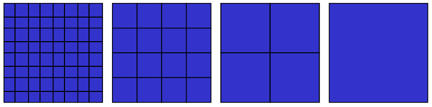
property **P1** ***on-top-of*** property **P2**



property **P1** ***next-to*** property **P2**

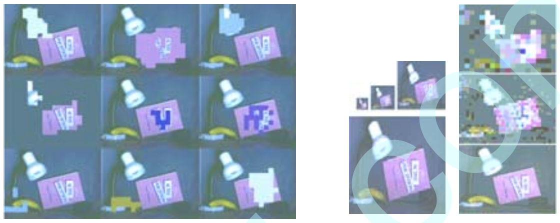


Different Resolution Hierarchy



Mining Multimedia Databases

From Coarse to Fine Resolution Mining



Challenge: Curse of Dimensionality

- Difficult to implement a data cube efficiently given a large number of dimensions, especially serious in the case of multimedia data cubes
- Many of these attributes are set-oriented instead of single-valued
- Restricting number of dimensions may lead to the modeling of an image at a rather rough, limited, and imprecise scale
- More research is needed to strike a balance between efficiency and power of representation

Chapter 9. Mining Complex Types of Data

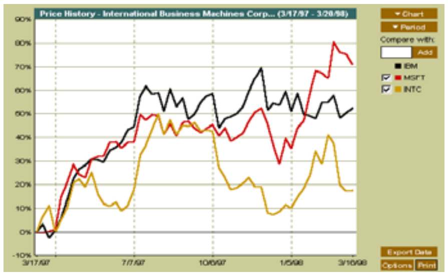
- Mining spatial databases
- Mining multimedia databases
- Mining time-series and sequence data
- Mining stream data
- Mining text databases
- Mining the World-Wide Web
- Summary

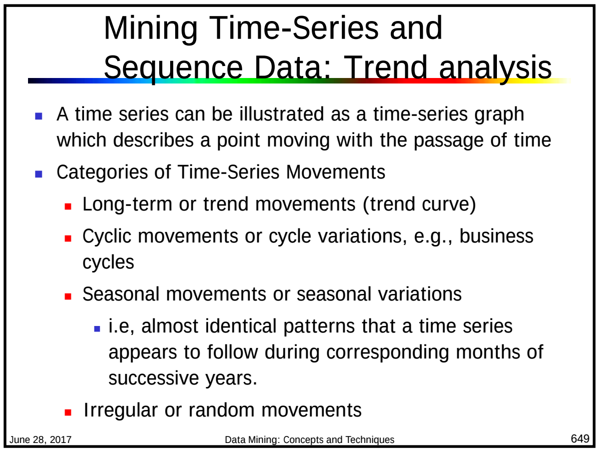
Mining Time-Series and Sequence Data

- Time-series database
  - Consists of sequences of values or events changing with time
  - Data is recorded at regular intervals
  - Characteristic time-series components
    - Trend, cycle, seasonal, irregular
- Applications
  - Financial: stock price, inflation
  - Biomedical: blood pressure
  - Meteorological: precipitation

Mining Time-Series and Sequence Data

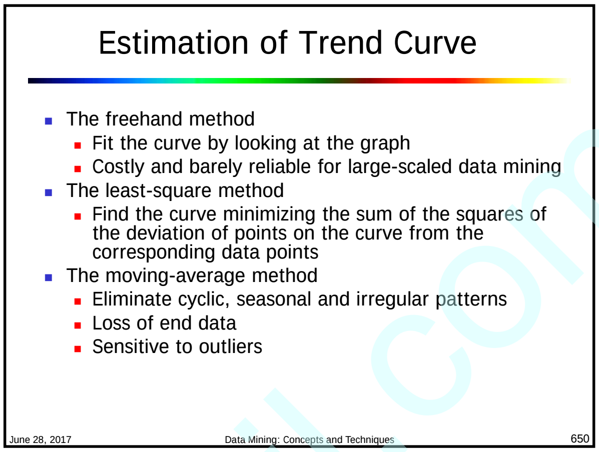
Time-series plot





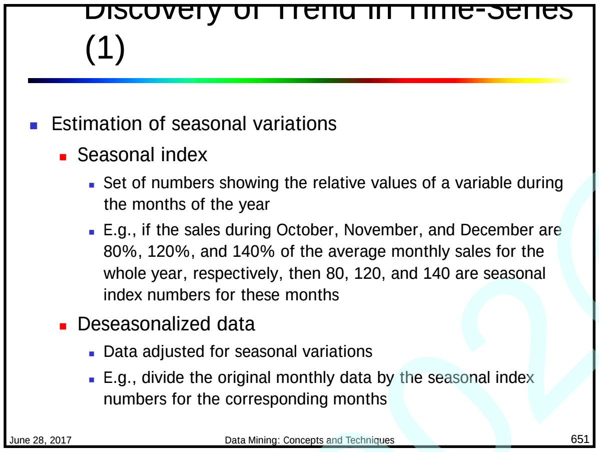
Mining Time-Series and Sequence Data: Trend analysis

- A time series can be illustrated as a time-series graph which describes a point moving with the passage of time
- Categories of Time-Series Movements
  - Long-term or trend movements (trend curve)
  - Cyclic movements or cycle variations, e.g., business cycles
  - Seasonal movements or seasonal variations
    - i.e, almost identical patterns that a time series appears to follow during corresponding months of successive years.
  - Irregular or random movements



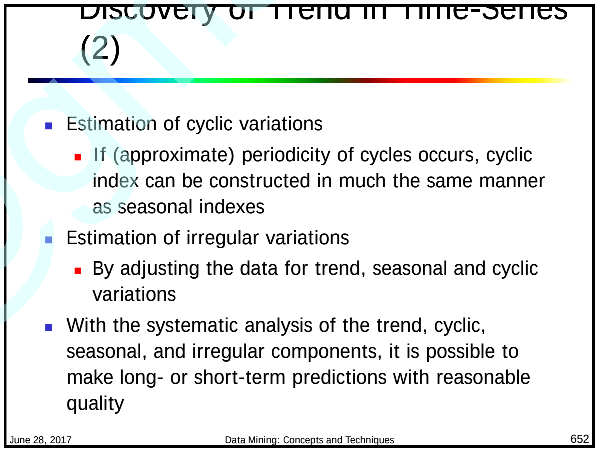
Estimation of Trend Curve

- The freehand method
  - Fit the curve by looking at the graph
  - Costly and barely reliable for large-scaled data mining
- The least-square method
  - Find the curve minimizing the sum of the squares of the deviation of points on the curve from the corresponding data points
- The moving-average method
  - Eliminate cyclic, seasonal and irregular patterns
  - Loss of end data
  - Sensitive to outliers



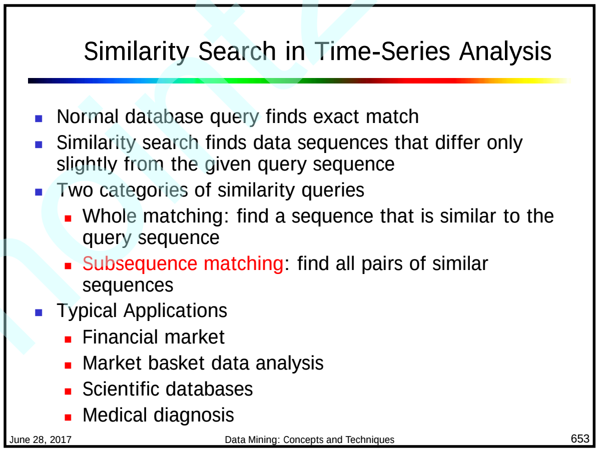
Discovery of Trend in Time-Series (1)

- Estimation of seasonal variations
  - Seasonal index
    - Set of numbers showing the relative values of a variable during the months of the year
    - E.g., if the sales during October, November, and December are 80%, 120%, and 140% of the average monthly sales for the whole year, respectively, then 80, 120, and 140 are seasonal index numbers for these months
  - Deseasonalized data
    - Data adjusted for seasonal variations
    - E.g., divide the original monthly data by the seasonal index numbers for the corresponding months



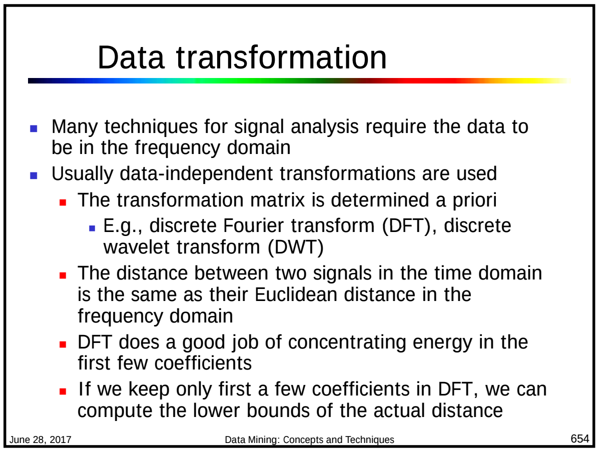
Discovery of Trend in Time-Series (2)

- Estimation of cyclic variations
  - If (approximate) periodicity of cycles occurs, cyclic index can be constructed in much the same manner as seasonal indexes
- Estimation of irregular variations
  - By adjusting the data for trend, seasonal and cyclic variations
- With the systematic analysis of the trend, cyclic, seasonal, and irregular components, it is possible to make long- or short-term predictions with reasonable quality



Similarity Search in Time-Series Analysis

- Normal database query finds exact match
- Similarity search finds data sequences that differ only slightly from the given query sequence
- Two categories of similarity queries
  - Whole matching: find a sequence that is similar to the query sequence
  - Subsequence matching: find all pairs of similar sequences
- Typical Applications
  - Financial market
  - Market basket data analysis
  - Scientific databases
  - Medical diagnosis



Data transformation

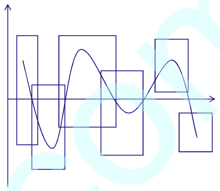
- Many techniques for signal analysis require the data to be in the frequency domain
- Usually data-independent transformations are used
  - The transformation matrix is determined a priori
    - E.g., discrete Fourier transform (DFT), discrete wavelet transform (DWT)
  - The distance between two signals in the time domain is the same as their Euclidean distance in the frequency domain
  - DFT does a good job of concentrating energy in the first few coefficients
  - If we keep only first a few coefficients in DFT, we can compute the lower bounds of the actual distance

Multidimensional Indexing

- Multidimensional index
  - Constructed for efficient accessing using the first few Fourier coefficients
- Use the index can to retrieve the sequences that are at most a certain small distance away from the query sequence
- Perform post-processing by computing the actual distance between sequences in the time domain and discard any false matches

Subsequence Matching

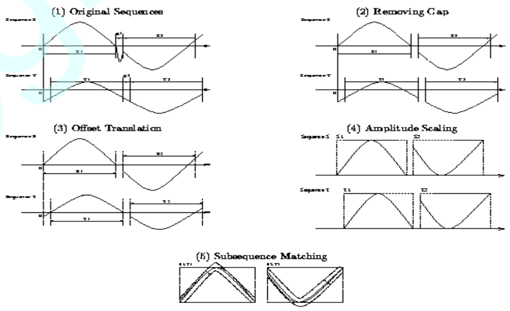
- Break each sequence into a set of pieces of window with length *w*
- Extract the features of the subsequence inside the window
- Map each sequence to a "trail" in the feature space
- Divide the trail of each sequence into "subtrails" and represent each of them with minimum bounding rectangle
- Use a multipiece assembly algorithm to search for longer sequence matches



Enhanced similarity search methods

- Allow for gaps within a sequence or differences in offsets or amplitudes
- Normalize sequences with amplitude scaling and offset translation
- Two subsequences are considered similar if one lies within an envelope of ε width around the other, ignoring outliers
- Two sequences are said to be similar if they have enough non-overlapping time-ordered pairs of similar subsequences
- Parameters specified by a user or expert: sliding window size, width of an envelope for similarity, maximum gap, and matching fraction

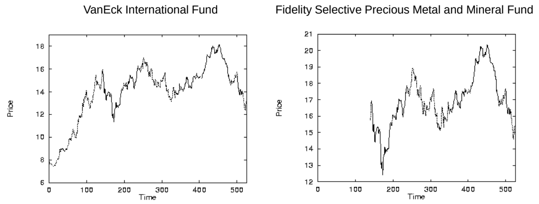
Similar time series analysis



Steps for Performing a Similarity Search

- Atomic matching
  - Find all pairs of gap-free windows of a small length that are similar
- Window stitching
  - Stitch similar windows to form pairs of large similar subsequences allowing gaps between atomic matches
- Subsequence Ordering
  - Linearly order the subsequence matches to determine whether enough similar pieces exist

Similar time series analysis


VanEck International Fund

Fidelity Selective Precious Metal and Mineral Fund

Two similar mutual funds in the different fund group

Query Languages for Time Sequences

- Time-sequence query language
  - Should be able to specify sophisticated queries like

Find all of the sequences that are similar to some sequence in class *A*, but not similar to any sequence in class *B*

  - Should be able to support various kinds of queries: range queries, all-pair queries, and nearest neighbor queries
- Shape definition language
  - Allows users to define and query the overall shape of time sequences
  - Uses human readable series of sequence transitions or macros
  - Ignores the specific details
    - E.g., the pattern up, Up, UP can be used to describe increasing degrees of rising slopes
    - Macros: spike, valley, etc.

Sequential Pattern Mining

- Mining of frequently occurring patterns related to time or other sequences
- Sequential pattern mining usually concentrate on symbolic patterns
- Examples
  - Renting “Star Wars”, then “Empire Strikes Back”, then “Return of the Jedi” in that order
  - Collection of ordered events within an interval
- Applications
  - Targeted marketing
  - Customer retention
  - Weather prediction

Mining Sequences (cont.)

Customer-sequence

CustId	Video sequence
1	{(C), (H)}
2	{(AB), (C), (DFG)}
3	{(CEG)}
4	{(C), (DG), (H)}
5	{(H)}

Map Large Itemsets

Large Itemsets	MappedID
(C)	1
(D)	2
(G)	3
(DG)	4
(H)	5

Sequential patterns with support > 0.25
{(C), (H)}
{(C), (DG)}

Sequential pattern mining: Cases and Parameters

- Duration of a time sequence *T*
  - Sequential pattern mining can then be confined to the data within a specified duration
  - Ex. Subsequence corresponding to the year of 1999
  - Ex. Partitioned sequences, such as every year, or every week after stock crashes, or every two weeks before and after a volcano eruption
- Event folding window *w*
  - If *w = T*, time-insensitive frequent patterns are found
  - If *w = 0* (no event sequence folding), sequential patterns are found where each event occurs at a distinct time instant
  - If *0 < w < T*, sequences occurring within the same period *w* are folded in the analysis

Sequential pattern mining: Cases and Parameters (2)

- Time interval, *int*, between events in the discovered pattern
  - *int* = 0: no interval gap is allowed, i.e., only strictly consecutive sequences are found
    - Ex. “Find frequent patterns occurring in consecutive weeks”
  - *min_int* ≤ *int* ≤ *max_int*: find patterns that are separated by at least *min_int* but at most *max_int*
    - Ex. “If a person rents movie A, it is likely she will rent movie B within 30 days” (*int* ≤ 30)
  - *int* = c ≠ 0: find patterns carrying an exact interval
    - Ex. “Every time when Dow Jones drops more than 5%, what will happen exactly two days later?” (*int* = 2)

Episodes and Sequential Pattern Mining Methods

- Other methods for specifying the kinds of patterns
  - Serial episodes: A → B
  - Parallel episodes: A & B
  - Regular expressions: (A | B)C*(D → E)
- Methods for sequential pattern mining
  - Variations of Apriori-like algorithms, e.g., GSP
  - Database projection-based pattern growth
    - Similar to the frequent pattern growth without candidate generation

Periodicity Analysis

- Periodicity is everywhere: tides, seasons, daily power consumption, etc.
- Full periodicity
  - Every point in time contributes (precisely or approximately) to the periodicity
- Partial periodicit: A more general notion
  - Only some segments contribute to the periodicity
    - Jim reads NY Times 7:00-7:30 am every week day
- Cyclic association rules
  - Associations which form cycles
- Methods
  - Full periodicity: FFT, other statistical analysis methods
  - Partial and cyclic periodicity: Variations of Apriori-like mining methods

Chapter 9. Mining Complex Types of Data

- Mining spatial databases
- Mining multimedia databases
- Mining time-series and sequence data
- Mining stream data
- Mining text databases
- Mining the World-Wide Web
- Summary

Chapter 9. Mining Complex Types of Data

- Mining spatial databases
- Mining multimedia databases
- Mining time-series and sequence data
- Mining stream data
- Mining text databases
- Mining the World-Wide Web
- Summary

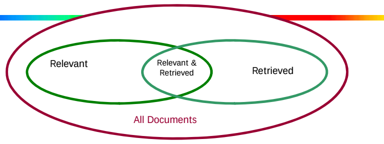
Text Databases and IR

- Text databases (document databases)
  - Large collections of documents from various sources: news articles, research papers, books, digital libraries, e-mail messages, and Web pages, library database, etc.
  - Data stored is usually *semi-structured*
  - Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data
- Information retrieval
  - A field developed in parallel with database systems
  - Information is organized into (a large number of) documents
  - Information retrieval problem: locating relevant documents based on user input, such as keywords or example documents

Information Retrieval

- Typical IR systems
  - Online library catalogs
  - Online document management systems
- Information retrieval vs. database systems
  - Some DB problems are not present in IR, e.g., update, transaction management, complex objects
  - Some IR problems are not addressed well in DBMS, e.g., unstructured documents, approximate search using keywords and relevance

Basic Measures for Text Retrieval



- Precision: the percentage of retrieved documents that are in fact relevant to the query (i.e., “correct” responses)

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|}$$

- Recall: the percentage of documents that are relevant to the query and were, in fact, retrieved

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}$$

Information Retrieval Techniques(1

- Basic Concepts
  - A document can be described by a set of representative keywords called index terms.
  - Different index terms have varying relevance when used to describe document contents.
  - This effect is captured through the assignment of numerical weights to each index term of a document. (e.g.: frequency, tf-idf)
- DBMS Analogy
  - Index Terms → Attributes
  - Weights → Attribute Values

Information Retrieval Techniques(2

- Index Terms (Attribute) Selection:
  - Stop list
  - Word stem
  - Index terms weighting methods
- Terms ✕ Documents Frequency Matrices
- Information Retrieval Models:
  - Boolean Model
  - Vector Model
  - Probabilistic Model

Boolean Model

- Consider that index terms are either present or absent in a document
- As a result, the index term weights are assumed to be all binaries
- A query is composed of index terms linked by three connectives: not, and, and or
  - e.g.: car *and* repair, plane *or* airplane
- The Boolean model predicts that each document is either relevant or non-relevant based on the match of a document to the query

Boolean Model: Keyword-Based Retrieval

- A document is represented by a string, which can be identified by a set of keywords
- Queries may use expressions of keywords
  - E.g., car *and* repair shop, tea *or* coffee, DBMS *but not* Oracle
  - Queries and retrieval should consider synonyms, e.g., repair and maintenance
- Major difficulties of the model
  - Synonymy: A keyword *T* does not appear anywhere in the document, even though the document is closely related to *T*, e.g., data mining
  - Polysemy: The same keyword may mean different things in different contexts, e.g., mining

Similarity-Based Retrieval in Text Databases

- Finds similar documents based on a set of common keywords
- Answer should be based on the degree of relevance based on the nearness of the keywords, relative frequency of the keywords, etc.
- Basic techniques
- Stop list
  - Set of words that are deemed "irrelevant", even though they may appear frequently
  - E.g., *a, the, of, for, to, with*, etc.
  - Stop lists may vary when document set varies

Similarity-Based Retrieval in Text Databases (2)

- Word stem
  - Several words are small syntactic variants of each other since they share a common word stem
  - E.g., *drug, drugs, drugged*
- A term frequency table
  - Each entry *frequent_table(i, j)* = # of occurrences of the word t_i in document d_i
  - Usually, the *ratio* instead of the absolute number of occurrences is used
- Similarity metrics: measure the closeness of a document to a query (a set of keywords)
  - Relative term occurrences
  - Cosine distance:

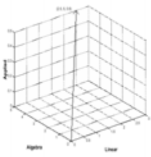
$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{|v_1||v_2|}$$

Indexing Techniques

- Inverted index
  - Maintains two hash- or B+-tree indexed tables:
    - document_table: a set of document records <doc_id, postings_list>
    - term_table: a set of term records, <term, postings_list>
  - Answer query: Find all docs associated with one or a set of terms
  - + easy to implement
  - – do not handle well synonymy and polysemy, and posting lists could be too long (storage could be very large)
- Signature file
  - Associate a signature with each document
  - A signature is a representation of an ordered list of terms that describe the document
  - Order is obtained by frequency analysis, stemming and stop lists

Vector Model

- Documents and user queries are represented as m-dimensional vectors, where m is the total number of index terms in the document collection.
- The degree of similarity of the document d with regard to the query q is calculated as the correlation between the vectors that represent them, using measures such as the Euclidian distance or the cosine of the angle between these two vectors.



Latent Semantic Indexing (1)

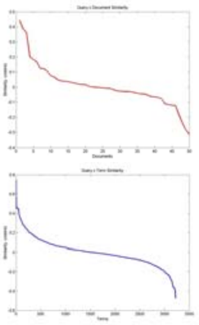
- Basic idea
  - Similar documents have similar word frequencies
  - Difficulty: the size of the term frequency matrix is very large
  - Use a singular value decomposition (SVD) techniques to reduce the size of frequency table
  - Retain the *K* most significant rows of the frequency table
- Method
  - Create a term x document weighted frequency matrix A
  - SVD construction: A = U * S * V'
  - Define K and obtain U_k , S_k , and V_k.
  - Create query vector q' .
  - Project q' into the term-document space: $Dq = q' * U_k * S_k^{-1}$
  - Calculate similarities: cos α = Dq . D / ||Dq|| * ||D||

Latent Semantic Indexing (2)

Weighted Frequency Matrix

Query Terms:
- Insulation
- Joint

```
DOCUMENTS:
  'CM031.txt'
  'CM046.txt'
  'CM001.txt'
  'CM029.txt'
  'CM040.txt'
K>> return
TERMS:
  'joint'
  'insulation'
  'roofing'
  'expansion'
  'saw'
```



Probabilistic Model

- Basic assumption: Given a user query, there is a set of documents which contains exactly the relevant documents and no other (ideal answer set)
- Querying process as a process of specifying the properties of an ideal answer set. Since these properties are not known at query time, an initial guess is made
- This initial guess allows the generation of a preliminary probabilistic description of the ideal answer set which is used to retrieve the first set of documents
- An interaction with the user is then initiated with the purpose of improving the probabilistic description of the answer set

Types of Text Data Mining

- Keyword-based association analysis
- Automatic document classification
- Similarity detection
  - Cluster documents by a common author
  - Cluster documents containing information from a common source
- Link analysis: unusual correlation between entities
- Sequence analysis: predicting a recurring event
- Anomaly detection: find information that violates usual patterns
- Hypertext analysis
  - Patterns in anchors/links
    - Anchor text correlations with linked objects

Keyword-Based Association Analysis

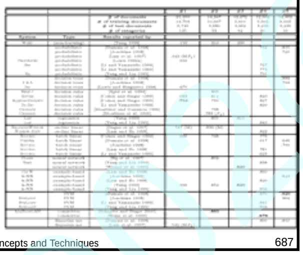
- Motivation
  - Collect sets of keywords or terms that occur frequently together and then find the association or correlation relationships among them
- Association Analysis Process
  - Preprocess the text data by parsing, stemming, removing stop words, etc.
  - Evoke association mining algorithms
    - Consider each document as a transaction
    - View a set of keywords in the document as a set of items in the transaction
  - Term level association mining
    - No need for human effort in tagging documents
    - The number of meaningless results and the execution time is greatly reduced

Text Classification(1)

- Motivation
  - Automatic classification for the large number of on-line text documents (Web pages, e-mails, corporate intranets, etc.)
- Classification Process
  - Data preprocessing
  - Definition of training set and test sets
  - Creation of the classification model using the selected classification algorithm
  - Classification model validation
  - Classification of new/unknown text documents
- Text document classification differs from the classification of relational data
  - Document databases are not structured according to attribute-value pairs

Text Classification(2)

- Classification Algorithms:
  - Support Vector Machines
  - K-Nearest Neighbors
  - Naïve Bayes
  - Neural Networks
  - Decision Trees
  - Association rule-based
  - Boosting



Document Clustering

- Motivation
  - Automatically group related documents based on their contents
  - No predetermined training sets or taxonomies
  - Generate a taxonomy at runtime
- Clustering Process
  - Data preprocessing: remove stop words, stem, feature extraction, lexical analysis, etc.
  - Hierarchical clustering: compute similarities applying clustering algorithms.
  - Model-Based clustering (Neural Network Approach): clusters are represented by "exemplars". (e.g.: SOM)

Chapter 9. Mining Complex Types of Data

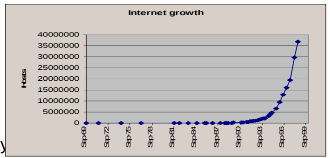
- Mining spatial databases
- Mining multimedia databases
- Mining time-series and sequence data
- Mining stream data
- Mining text databases
- Mining the World-Wide Web
- Summary

Mining the World-Wide Web

- The WWW is huge, widely distributed, global information service center for
  - Information services: news, advertisements, consumer information, financial management, education, government, e-commerce, etc.
  - Hyper-link information
  - Access and usage information
- WWW provides rich sources for data mining
- Challenges
  - Too huge for effective data warehousing and data mining
  - Too complex and heterogeneous: no standards and structure

Mining the World-Wide Web

- Growing and changing very rapidly



- Broad diversity
- Only a small portion of the information on the Web is truly relevant or useful
  - 99% of the Web information is useless to 99% of Web users
  - How can we find high-quality Web pages on a specified topic?

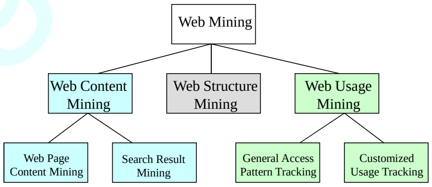
Web search engines

- Index-based: search the Web, index Web pages, and build and store huge keyword-based indices
- Help locate sets of Web pages containing certain keywords
- Deficiencies
  - A topic of any breadth may easily contain hundreds of thousands of documents
  - Many documents that are highly relevant to a topic may not contain keywords defining them (polysemy)

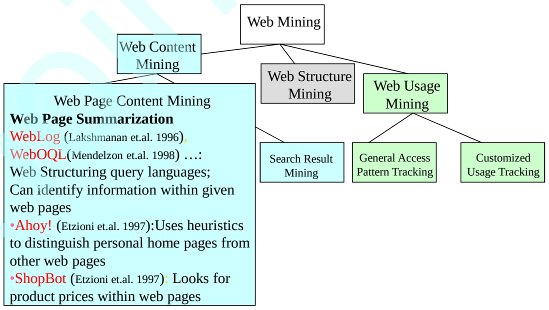
Web Mining: A more challenging task

- Searches for
  - Web access patterns
  - Web structures
  - Regularity and dynamics of Web contents
- Problems
  - The "abundance" problem
  - Limited coverage of the Web: hidden Web sources, majority of data in DBMS
  - Limited query interface based on keyword-oriented search
  - Limited customization to individual users

Web Mining Taxonomy

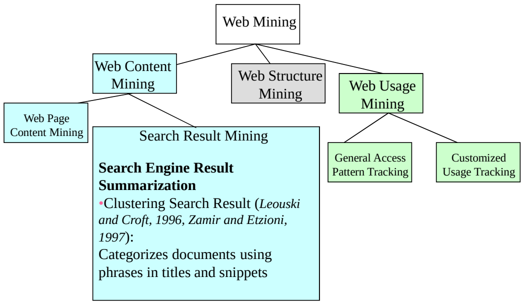


Mining the World-Wide Web



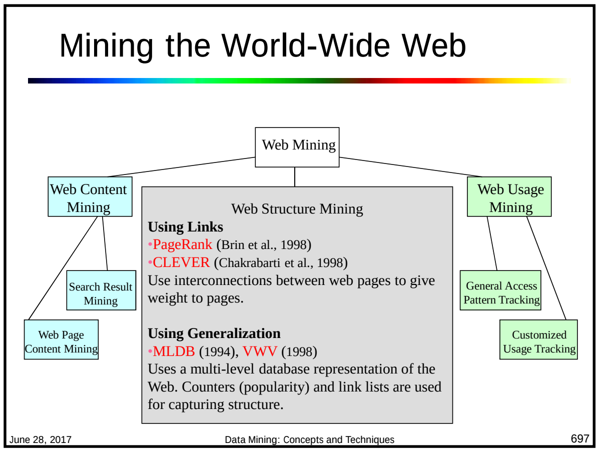
Web Page Content Mining
Web Page Summarization
WebLog (Lakshmanan et.al. 1996),
WebOQL(Mendelzon et.al. 1998) …:
Web Structuring query languages;
Can identify information within given web pages
•Ahoy! (Etzioni et.al. 1997):Uses heuristics to distinguish personal home pages from other web pages
•ShopBot (Etzioni et.al. 1997): Looks for product prices within web pages

Mining the World-Wide Web



Search Result Mining

Search Engine Result Summarization
•Clustering Search Result (*Leouski and Croft, 1996, Zamir and Etzioni, 1997*):
Categorizes documents using phrases in titles and snippets



Mining the World-Wide Web

Web Mining

Web Content Mining

Web Page Content Mining

Search Result Mining

Web Structure Mining

Using Links

•PageRank (Brin et al., 1998)

•CLEVER (Chakrabarti et al., 1998)

Use interconnections between web pages to give weight to pages.

Using Generalization

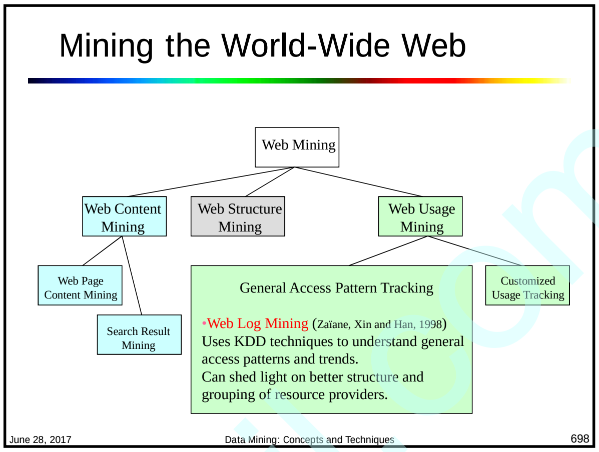
•MLDB (1994), VWV (1998)

Uses a multi-level database representation of the Web. Counters (popularity) and link lists are used for capturing structure.

Web Usage Mining

General Access Pattern Tracking

Customized Usage Tracking



Mining the World-Wide Web

Web Mining

Web Content Mining

Web Page Content Mining

Search Result Mining

Web Structure Mining

Web Usage Mining

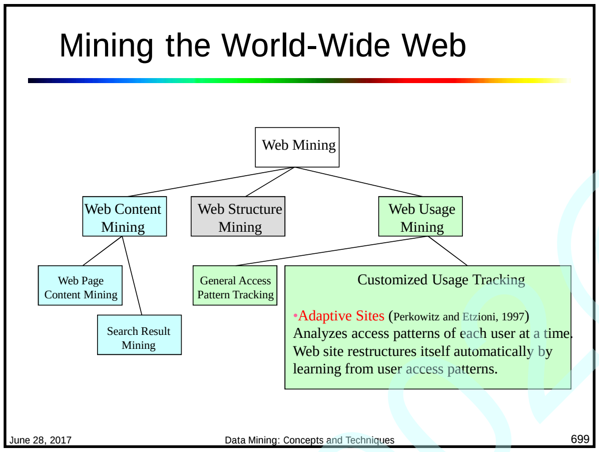
General Access Pattern Tracking

•Web Log Mining (Zaïane, Xin and Han, 1998)

Uses KDD techniques to understand general access patterns and trends.

Can shed light on better structure and grouping of resource providers.

Customized Usage Tracking



Mining the World-Wide Web

Web Mining

Web Content Mining

Web Page Content Mining

Search Result Mining

Web Structure Mining

Web Usage Mining

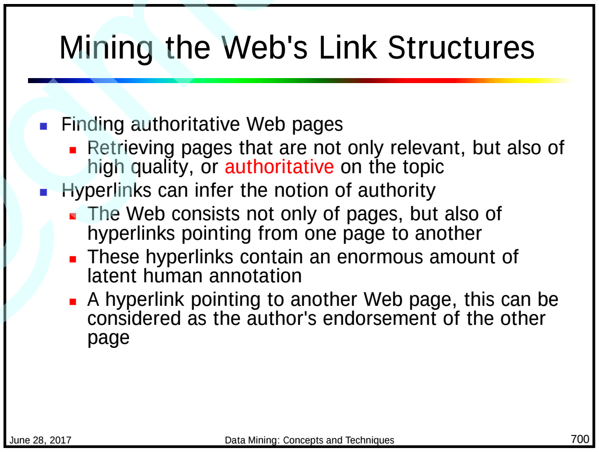
General Access Pattern Tracking

Customized Usage Tracking

•Adaptive Sites (Perkowitz and Etzioni, 1997)

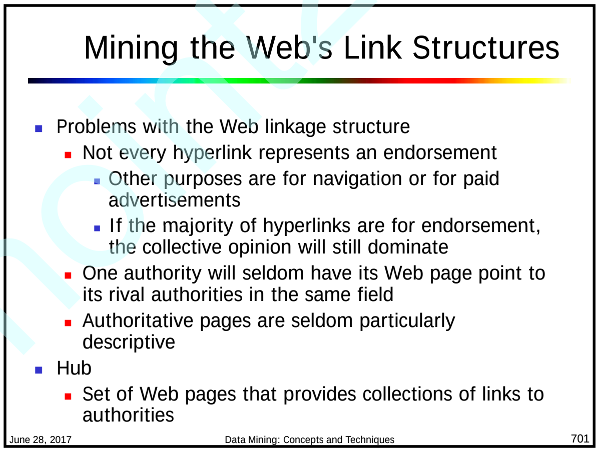
Analyzes access patterns of each user at a time.

Web site restructures itself automatically by learning from user access patterns.



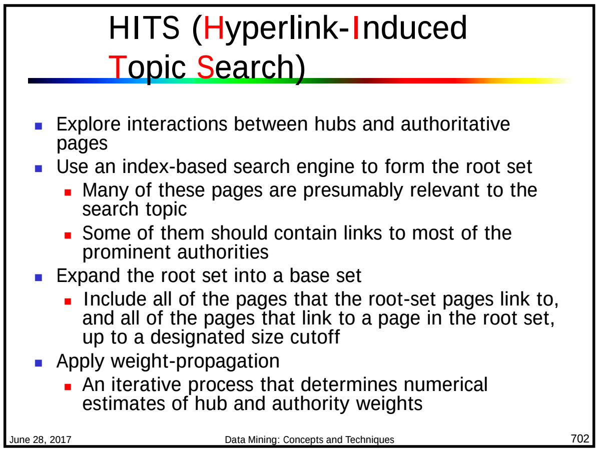
Mining the Web's Link Structures

- Finding authoritative Web pages
  - Retrieving pages that are not only relevant, but also of high quality, or authoritative on the topic
- Hyperlinks can infer the notion of authority
  - The Web consists not only of pages, but also of hyperlinks pointing from one page to another
  - These hyperlinks contain an enormous amount of latent human annotation
  - A hyperlink pointing to another Web page, this can be considered as the author's endorsement of the other page



Mining the Web's Link Structures

- Problems with the Web linkage structure
  - Not every hyperlink represents an endorsement
    - Other purposes are for navigation or for paid advertisements
    - If the majority of hyperlinks are for endorsement, the collective opinion will still dominate
  - One authority will seldom have its Web page point to its rival authorities in the same field
  - Authoritative pages are seldom particularly descriptive
- Hub
  - Set of Web pages that provides collections of links to authorities



HITS (Hyperlink-Induced Topic Search)

- Explore interactions between hubs and authoritative pages
- Use an index-based search engine to form the root set
  - Many of these pages are presumably relevant to the search topic
  - Some of them should contain links to most of the prominent authorities
- Expand the root set into a base set
  - Include all of the pages that the root-set pages link to, and all of the pages that link to a page in the root set, up to a designated size cutoff
- Apply weight-propagation
  - An iterative process that determines numerical estimates of hub and authority weights

Systems Based on HITS

  - Output a short list of the pages with large hub weights, and the pages with large authority weights for the given search topic
- Systems based on the HITS algorithm
  - Clever, Google: achieve better quality search results than those generated by term-index engines such as AltaVista and those created by human ontologists such as Yahoo!
- Difficulties from ignoring textual contexts
  - Drifting: when hubs contain multiple topics
  - Topic hijacking: when many pages from a single Web site point to the same single popular site

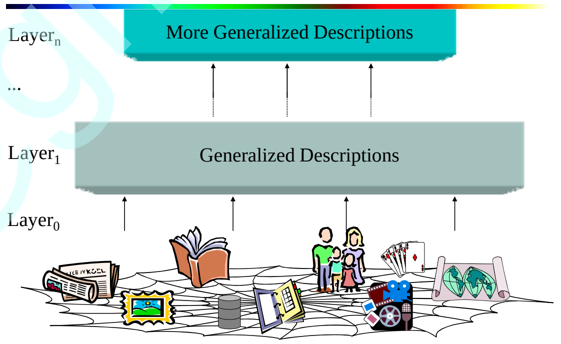
Automatic Classification of Web Documents

- Assign a class label to each document from a set of predefined topic categories
- Based on a set of examples of preclassified documents
- Example
  - Use Yahoo!'s taxonomy and its associated documents as training and test sets
  - Derive a Web document classification scheme
  - Use the scheme classify new Web documents by assigning categories from the same taxonomy
- Keyword-based document classification methods
- Statistical models

Multilayered Web Information Base

- Layer$_0$: the Web itself
- Layer$_1$: the Web page descriptor layer
  - Contains descriptive information for pages on the Web
  - An abstraction of Layer$_0$: substantially smaller but still rich enough to preserve most of the interesting, general information
  - Organized into dozens of semistructured classes
    - *document, person, organization, ads, directory, sales, software, game, stocks, library_catalog, geographic_data, scientific_data*, etc.
- Layer$_2$ and up: various Web directory services constructed on top of Layer$_1$
  - provide multidimensional, application-specific services

Multiple Layered Web Architecture



Mining the World-Wide Web

Layer-0: Primitive data

Layer-1: dozen database relations representing types of objects (metadata)

document, organization, person, software, game, map, image,…

- **document**(file_addr, authors, title, publication, publication_date, abstract, language, table_of_contents, category_description, keywords, index, multimedia_attached, num_pages, format, first_paragraphs, size_doc, timestamp, access_frequency, links_out,...)

- **person**(last_name, first_name, home_page_addr, position, picture_attached, phone, e-mail, office_address, education, research_interests, publications, size_of_home_page, timestamp, access_frequency, ...)

- **image**(image_addr, author, title, publication_date, category_description, keywords, size, width, height, duration, format, parent_pages, colour_histogram, Colour_layout, Texture_layout, Movement_vector, localisation_vector, timestamp, access_frequency, ...)

Mining the World-Wide Web

Layer-2: simplification of layer-1

- **doc_brief**(file_addr, authors, title, publication, publication_date, abstract, language, category_description, key_words, major_index, num_pages, format, size_doc, access_frequency, links_out)

- **person_brief** (last_name, first_name, publications,affiliation, e-mail, research_interests, size_home_page, access_frequency)

Layer-3: generalization of layer-2

- **cs_doc**(file_addr, authors, title, publication, publication_date, abstract, language, category_description, keywords, num_pages, form, size_doc, links_out)

- **doc_summary**(affiliation, field, publication_year, count, first_author_list, file_addr_list)

- **doc_author_brief**(file_addr, authors, affiliation, title, publication, pub_date, category_description, keywords, num_pages, format, size_doc, links_out)

- **person_summary**(affiliation, research_interest, year, num_publications, count)

XML and Web Mining

- XML can help to extract the correct descriptors
  - Standardization would greatly facilitate information extraction

    <NAME> eXtensible Markup Language</NAME>
    <RECOM>World-Wide Web Consortium</RECOM>
    <SINCE>1998</SINCE>
    <VERSION>1.0</VERSION>
    <DESC>Meta language that facilitates more meaningful and precise declarations of document content</DESC>
    <HOW>Definition of new tags and DTDs</HOW>

  - Potential problem
    - XML can help solve heterogeneity for vertical applications, but the freedom to define tags can make horizontal applications on the Web more heterogeneous

Benefits of Multi-Layer Meta-Web

- Benefits:
  - Multi-dimensional Web info summary analysis
  - Approximate and intelligent query answering
  - Web high-level query answering (WebSQL, WebML)
  - Web content and structure mining
  - Observing the dynamics/evolution of the Web
- Is it realistic to construct such a meta-Web?
  - Benefits even if it is partially constructed
  - Benefits may justify the cost of tool development, standardization and partial restructuring

Web Usage Mining

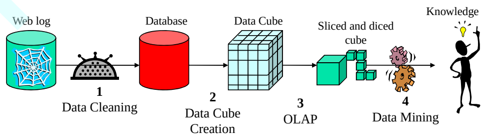
- Mining Web log records to discover user access patterns of Web pages
- Applications
  - Target potential customers for electronic commerce
  - Enhance the quality and delivery of Internet information services to the end user
  - Improve Web server system performance
  - Identify potential prime advertisement locations
- Web logs provide rich information about Web dynamics
  - Typical Web log entry includes the URL requested, the IP address from which the request originated, and a timestamp

Techniques for Web usage mining

- Construct multidimensional view on the Weblog database
  - Perform multidimensional OLAP analysis to find the top *N* users, top *N* accessed Web pages, most frequently accessed time periods, etc.
- Perform data mining on Weblog records
  - Find association patterns, sequential patterns, and trends of Web accessing
  - May need additional information,e.g., user browsing sequences of the Web pages in the Web server buffer
- Conduct studies to
  - Analyze system performance, improve system design by Web caching, Web page prefetching, and Web page swapping

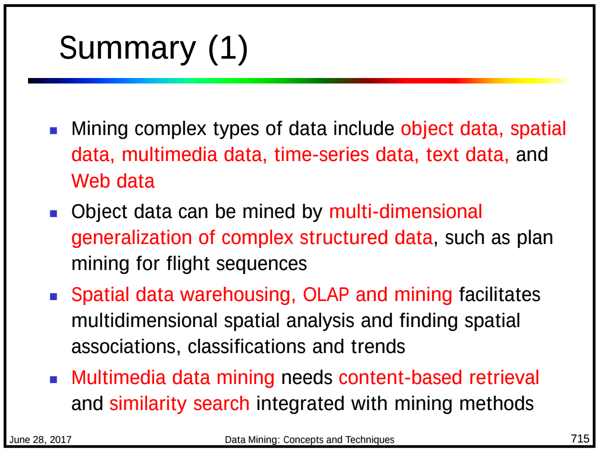
Mining the World-Wide Web

- Design of a Web Log Miner
  - Web log is filtered to generate a relational database
  - A data cube is generated form database
  - OLAP is used to drill-down and roll-up in the cube
  - OLAM is used for mining interesting knowledge



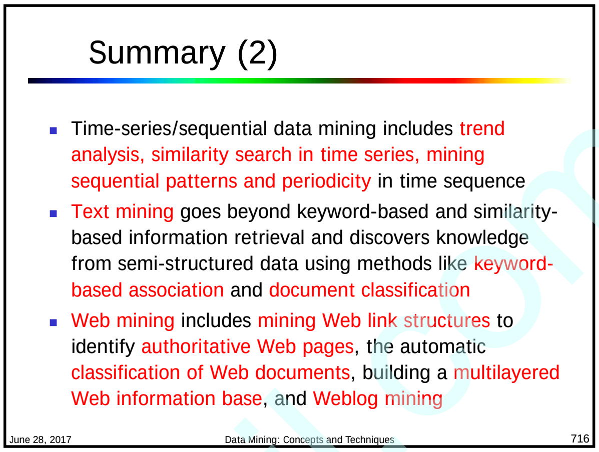
Chapter 9. Mining Complex Types of Data

- Mining spatial databases
- Mining multimedia databases
- Mining time-series and sequence data
- Mining stream data
- Mining text databases
- Mining the World-Wide Web
- Summary



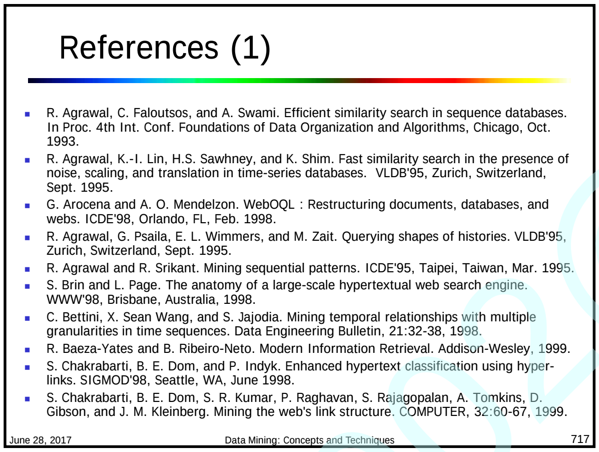
Summary (1)

- Mining complex types of data include object data, spatial data, multimedia data, time-series data, text data, and Web data
- Object data can be mined by multi-dimensional generalization of complex structured data, such as plan mining for flight sequences
- Spatial data warehousing, OLAP and mining facilitates multidimensional spatial analysis and finding spatial associations, classifications and trends
- Multimedia data mining needs content-based retrieval and similarity search integrated with mining methods



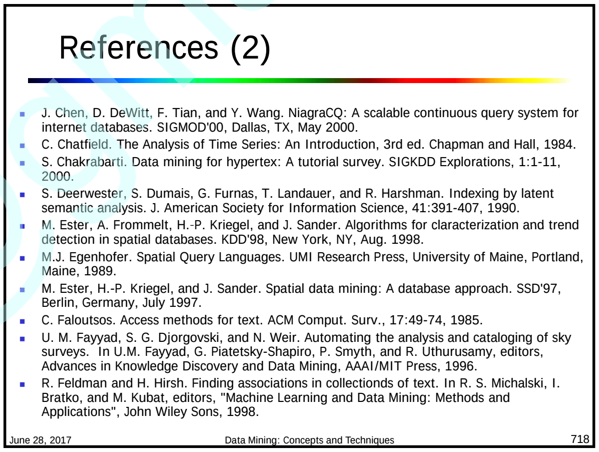
Summary (2)

- Time-series/sequential data mining includes trend analysis, similarity search in time series, mining sequential patterns and periodicity in time sequence
- Text mining goes beyond keyword-based and similarity-based information retrieval and discovers knowledge from semi-structured data using methods like keyword-based association and document classification
- Web mining includes mining Web link structures to identify authoritative Web pages, the automatic classification of Web documents, building a multilayered Web information base, and Weblog mining



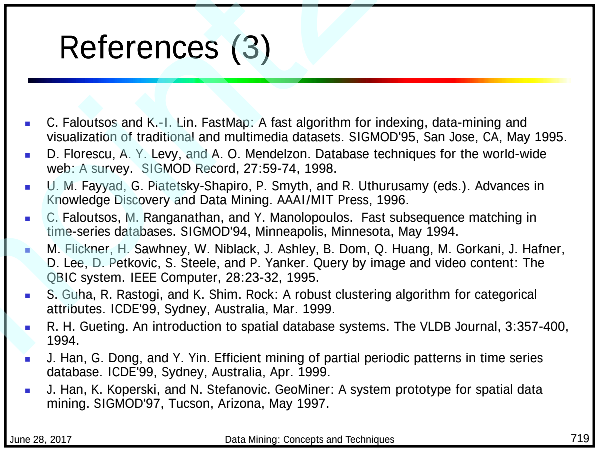
References (1)

- R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In Proc. 4th Int. Conf. Foundations of Data Organization and Algorithms, Chicago, Oct. 1993.
- R. Agrawal, K.-I. Lin, H.S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. VLDB'95, Zurich, Switzerland, Sept. 1995.
- G. Arocena and A. O. Mendelzon. WebOQL : Restructuring documents, databases, and webs. ICDE'98, Orlando, FL, Feb. 1998.
- R. Agrawal, G. Psaila, E. L. Wimmers, and M. Zait. Querying shapes of histories. VLDB'95, Zurich, Switzerland, Sept. 1995.
- R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95, Taipei, Taiwan, Mar. 1995.
- S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. WWW'98, Brisbane, Australia, 1998.
- C. Bettini, X. Sean Wang, and S. Jajodia. Mining temporal relationships with multiple granularities in time sequences. Data Engineering Bulletin, 21:32-38, 1998.
- R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, 1999.
- S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext classification using hyper-links. SIGMOD'98, Seattle, WA, June 1998.
- S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. M. Kleinberg. Mining the web's link structure. COMPUTER, 32:60-67, 1999.



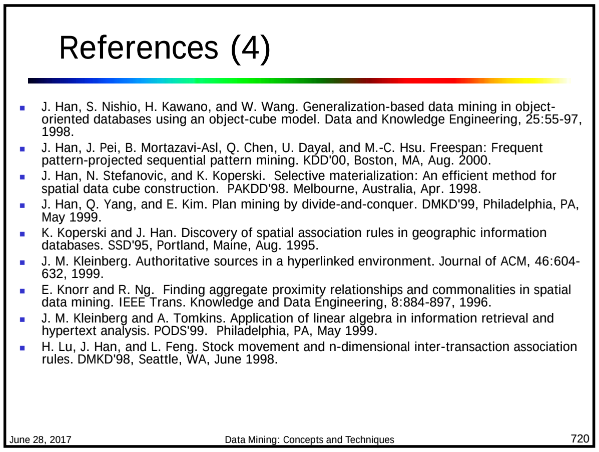
References (2)

- J. Chen, D. DeWitt, F. Tian, and Y. Wang. NiagraCQ: A scalable continuous query system for internet databases. SIGMOD'00, Dallas, TX, May 2000.
- C. Chatfield. The Analysis of Time Series: An Introduction, 3rd ed. Chapman and Hall, 1984.
- S. Chakrabarti. Data mining for hypertex: A tutorial survey. SIGKDD Explorations, 1:1-11, 2000.
- S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. J. American Society for Information Science, 41:391-407, 1990.
- M. Ester, A. Frommelt, H.-P. Kriegel, and J. Sander. Algorithms for claracterization and trend detection in spatial databases. KDD'98, New York, NY, Aug. 1998.
- M.J. Egenhofer. Spatial Query Languages. UMI Research Press, University of Maine, Portland, Maine, 1989.
- M. Ester, H.-P. Kriegel, and J. Sander. Spatial data mining: A database approach. SSD'97, Berlin, Germany, July 1997.
- C. Faloutsos. Access methods for text. ACM Comput. Surv., 17:49-74, 1985.
- U. M. Fayyad, S. G. Djorgovski, and N. Weir. Automating the analysis and cataloging of sky surveys. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996.
- R. Feldman and H. Hirsh. Finding associations in collectionds of text. In R. S. Michalski, I. Bratko, and M. Kubat, editors, "Machine Learning and Data Mining: Methods and Applications", John Wiley Sons, 1998.



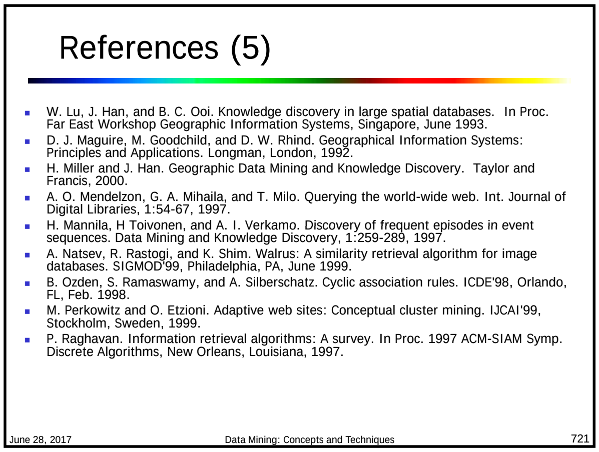
References (3)

- C. Faloutsos and K.-I. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. SIGMOD'95, San Jose, CA, May 1995.
- D. Florescu, A. Y. Levy, and A. O. Mendelzon. Database techniques for the world-wide web: A survey. SIGMOD Record, 27:59-74, 1998.
- U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.). Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.
- C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. SIGMOD'94, Minneapolis, Minnesota, May 1994.
- M. Flickner, H. Sawhney, W. Niblack, J. Ashley, B. Dom, Q. Huang, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, S. Steele, and P. Yanker. Query by image and video content: The QBIC system. IEEE Computer, 28:23-32, 1995.
- S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. ICDE'99, Sydney, Australia, Mar. 1999.
- R. H. Gueting. An introduction to spatial database systems. The VLDB Journal, 3:357-400, 1994.
- J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. ICDE'99, Sydney, Australia, Apr. 1999.
- J. Han, K. Koperski, and N. Stefanovic. GeoMiner: A system prototype for spatial data mining. SIGMOD'97, Tucson, Arizona, May 1997.



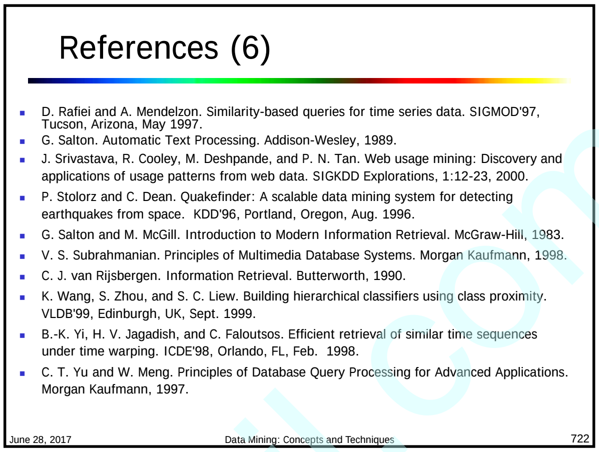
References (4)

- J. Han, S. Nishio, H. Kawano, and W. Wang. Generalization-based data mining in object-oriented databases using an object-cube model. Data and Knowledge Engineering, 25:55-97, 1998.
- J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M.-C. Hsu. Freespan: Frequent pattern-projected sequential pattern mining. KDD'00, Boston, MA, Aug. 2000.
- J. Han, N. Stefanovic, and K. Koperski. Selective materialization: An efficient method for spatial data cube construction. PAKDD'98. Melbourne, Australia, Apr. 1998.
- J. Han, Q. Yang, and E. Kim. Plan mining by divide-and-conquer. DMKD'99, Philadelphia, PA, May 1999.
- K. Koperski and J. Han. Discovery of spatial association rules in geographic information databases. SSD'95, Portland, Maine, Aug. 1995.
- J. M. Kleinberg. Authoritative sources in a hyperlinked environment. Journal of ACM, 46:604-632, 1999.
- E. Knorr and R. Ng. Finding aggregate proximity relationships and commonalities in spatial data mining. IEEE Trans. Knowledge and Data Engineering, 8:884-897, 1996.
- J. M. Kleinberg and A. Tomkins. Application of linear algebra in information retrieval and hypertext analysis. PODS'99. Philadelphia, PA, May 1999.
- H. Lu, J. Han, and L. Feng. Stock movement and n-dimensional inter-transaction association rules. DMKD'98, Seattle, WA, June 1998.



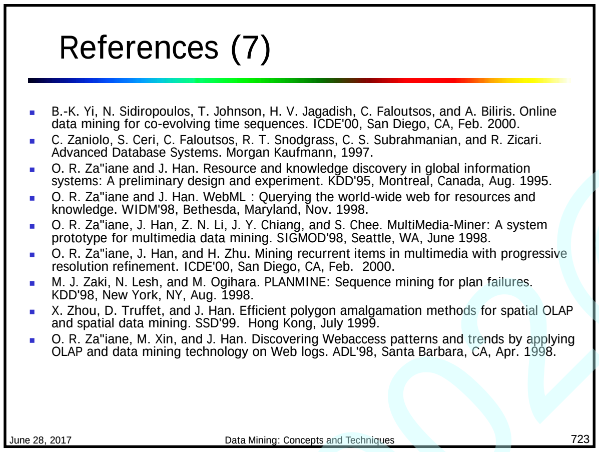
References (5)

- W. Lu, J. Han, and B. C. Ooi. Knowledge discovery in large spatial databases. In Proc. Far East Workshop Geographic Information Systems, Singapore, June 1993.
- D. J. Maguire, M. Goodchild, and D. W. Rhind. Geographical Information Systems: Principles and Applications. Longman, London, 1992.
- H. Miller and J. Han. Geographic Data Mining and Knowledge Discovery. Taylor and Francis, 2000.
- A. O. Mendelzon, G. A. Mihaila, and T. Milo. Querying the world-wide web. Int. Journal of Digital Libraries, 1:54-67, 1997.
- H. Mannila, H Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. Data Mining and Knowledge Discovery, 1:259-289, 1997.
- A. Natsev, R. Rastogi, and K. Shim. Walrus: A similarity retrieval algorithm for image databases. SIGMOD'99, Philadelphia, PA, June 1999.
- B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. ICDE'98, Orlando, FL, Feb. 1998.
- M. Perkowitz and O. Etzioni. Adaptive web sites: Conceptual cluster mining. IJCAI'99, Stockholm, Sweden, 1999.
- P. Raghavan. Information retrieval algorithms: A survey. In Proc. 1997 ACM-SIAM Symp. Discrete Algorithms, New Orleans, Louisiana, 1997.



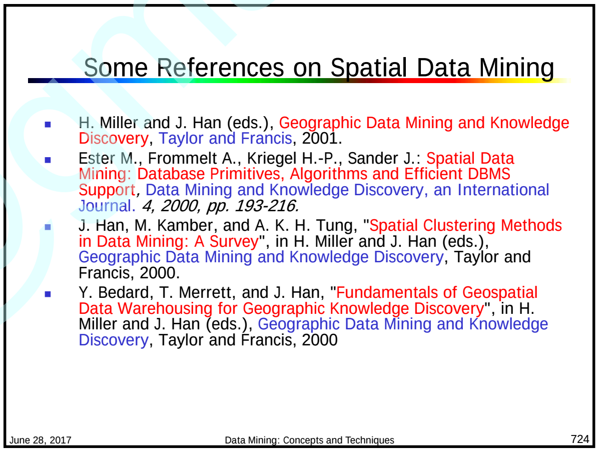
References (6)

- D. Rafiei and A. Mendelzon. Similarity-based queries for time series data. SIGMOD'97, Tucson, Arizona, May 1997.
- G. Salton. Automatic Text Processing. Addison-Wesley, 1989.
- J. Srivastava, R. Cooley, M. Deshpande, and P. N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations, 1:12-23, 2000.
- P. Stolorz and C. Dean. Quakefinder: A scalable data mining system for detecting earthquakes from space. KDD'96, Portland, Oregon, Aug. 1996.
- G. Salton and M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
- V. S. Subrahmanian. Principles of Multimedia Database Systems. Morgan Kaufmann, 1998.
- C. J. van Rijsbergen. Information Retrieval. Butterworth, 1990.
- K. Wang, S. Zhou, and S. C. Liew. Building hierarchical classifiers using class proximity. VLDB'99, Edinburgh, UK, Sept. 1999.
- B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. ICDE'98, Orlando, FL, Feb. 1998.
- C. T. Yu and W. Meng. Principles of Database Query Processing for Advanced Applications. Morgan Kaufmann, 1997.



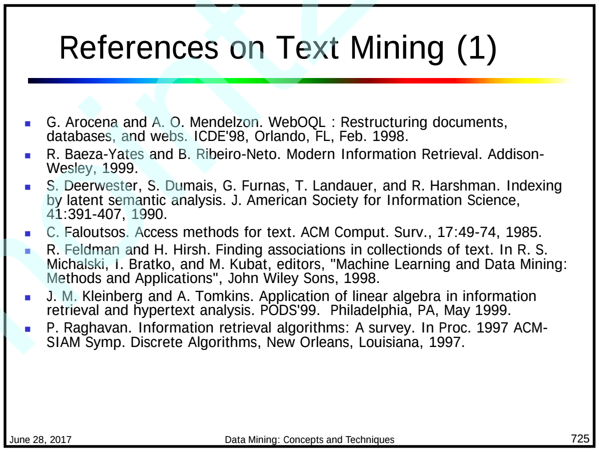
References (7)

- B.-K. Yi, N. Sidiropoulos, T. Johnson, H. V. Jagadish, C. Faloutsos, and A. Biliris. Online data mining for co-evolving time sequences. ICDE'00, San Diego, CA, Feb. 2000.
- C. Zaniolo, S. Ceri, C. Faloutsos, R. T. Snodgrass, C. S. Subrahmanian, and R. Zicari. Advanced Database Systems. Morgan Kaufmann, 1997.
- O. R. Za"iane and J. Han. Resource and knowledge discovery in global information systems: A preliminary design and experiment. KDD'95, Montreal, Canada, Aug. 1995.
- O. R. Za"iane and J. Han. WebML : Querying the world-wide web for resources and knowledge. WIDM'98, Bethesda, Maryland, Nov. 1998.
- O. R. Za"iane, J. Han, Z. N. Li, J. Y. Chiang, and S. Chee. MultiMedia-Miner: A system prototype for multimedia data mining. SIGMOD'98, Seattle, WA, June 1998.
- O. R. Za"iane, J. Han, and H. Zhu. Mining recurrent items in multimedia with progressive resolution refinement. ICDE'00, San Diego, CA, Feb. 2000.
- M. J. Zaki, N. Lesh, and M. Ogihara. PLANMINE: Sequence mining for plan failures. KDD'98, New York, NY, Aug. 1998.
- X. Zhou, D. Truffet, and J. Han. Efficient polygon amalgamation methods for spatial OLAP and spatial data mining. SSD'99. Hong Kong, July 1999.
- O. R. Za"iane, M. Xin, and J. Han. Discovering Webaccess patterns and trends by applying OLAP and data mining technology on Web logs. ADL'98, Santa Barbara, CA, Apr. 1998.



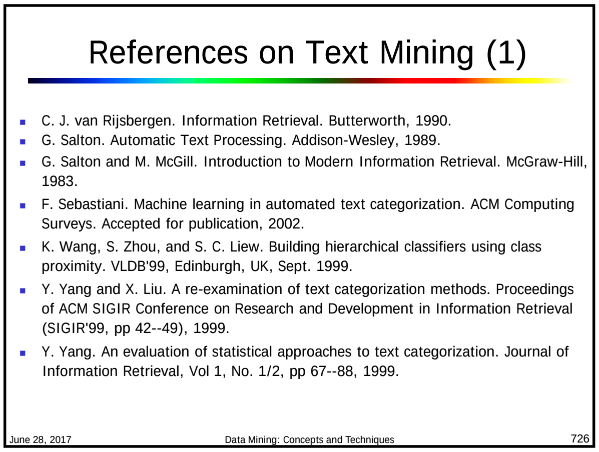
Some References on Spatial Data Mining

- H. Miller and J. Han (eds.), Geographic Data Mining and Knowledge Discovery, Taylor and Francis, 2001.
- Ester M., Frommelt A., Kriegel H.-P., Sander J.: Spatial Data Mining: Database Primitives, Algorithms and Efficient DBMS Support, Data Mining and Knowledge Discovery, an International Journal. *4, 2000, pp. 193-216.*
- J. Han, M. Kamber, and A. K. H. Tung, "Spatial Clustering Methods in Data Mining: A Survey", in H. Miller and J. Han (eds.), Geographic Data Mining and Knowledge Discovery, Taylor and Francis, 2000.
- Y. Bedard, T. Merrett, and J. Han, "Fundamentals of Geospatial Data Warehousing for Geographic Knowledge Discovery", in H. Miller and J. Han (eds.), Geographic Data Mining and Knowledge Discovery, Taylor and Francis, 2000



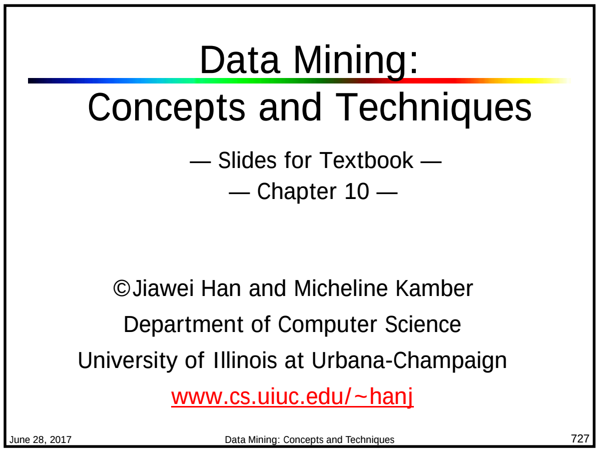
References on Text Mining (1)

- G. Arocena and A. O. Mendelzon. WebOQL : Restructuring documents, databases, and webs. ICDE'98, Orlando, FL, Feb. 1998.
- R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley, 1999.
- S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. J. American Society for Information Science, 41:391-407, 1990.
- C. Faloutsos. Access methods for text. ACM Comput. Surv., 17:49-74, 1985.
- R. Feldman and H. Hirsh. Finding associations in collectionds of text. In R. S. Michalski, I. Bratko, and M. Kubat, editors, "Machine Learning and Data Mining: Methods and Applications", John Wiley Sons, 1998.
- J. M. Kleinberg and A. Tomkins. Application of linear algebra in information retrieval and hypertext analysis. PODS'99. Philadelphia, PA, May 1999.
- P. Raghavan. Information retrieval algorithms: A survey. In Proc. 1997 ACM-SIAM Symp. Discrete Algorithms, New Orleans, Louisiana, 1997.



References on Text Mining (1)

- C. J. van Rijsbergen. Information Retrieval. Butterworth, 1990.
- G. Salton. Automatic Text Processing. Addison-Wesley, 1989.
- G. Salton and M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.
- F. Sebastiani. Machine learning in automated text categorization. ACM Computing Surveys. Accepted for publication, 2002.
- K. Wang, S. Zhou, and S. C. Liew. Building hierarchical classifiers using class proximity. VLDB'99, Edinburgh, UK, Sept. 1999.
- Y. Yang and X. Liu. A re-examination of text categorization methods. Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99, pp 42--49), 1999.
- Y. Yang. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, Vol 1, No. 1/2, pp 67--88, 1999.

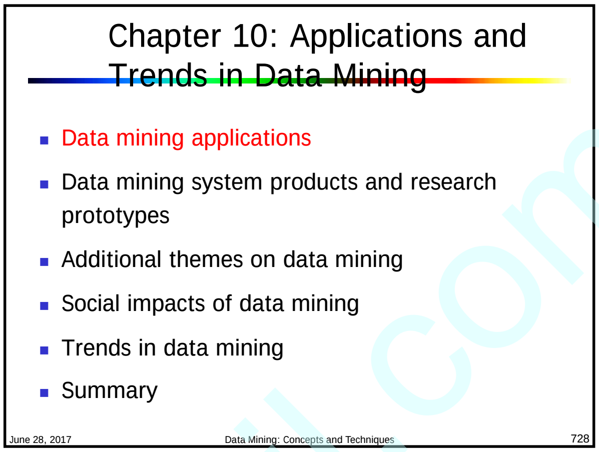


Data Mining: Concepts and Techniques

— Slides for Textbook —
— Chapter 10 —

©Jiawei Han and Micheline Kamber
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj

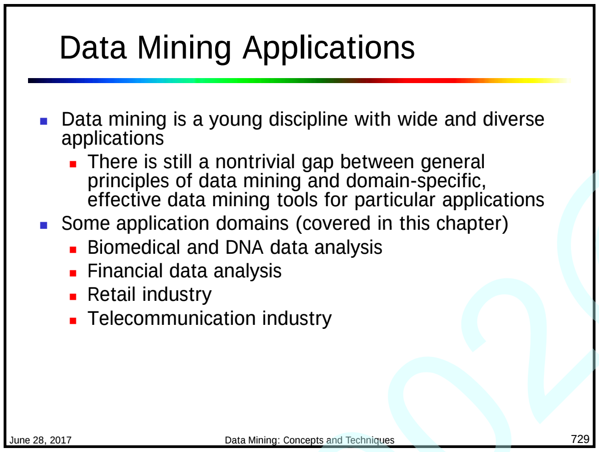
June 28, 2017 Data Mining: Concepts and Techniques 727



Chapter 10: Applications and Trends in Data Mining

- Data mining applications
- Data mining system products and research prototypes
- Additional themes on data mining
- Social impacts of data mining
- Trends in data mining
- Summary

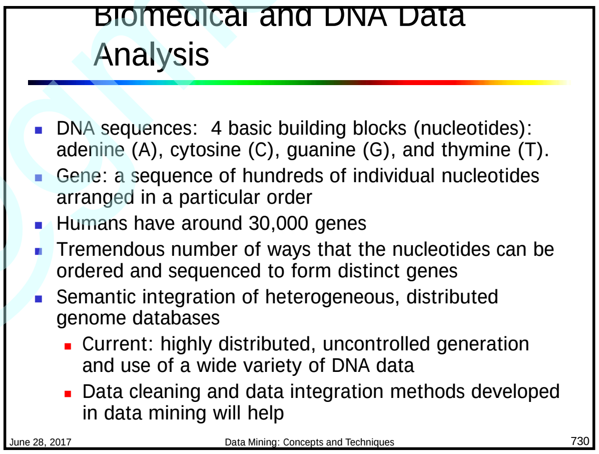
June 28, 2017 Data Mining: Concepts and Techniques 728



Data Mining Applications

- Data mining is a young discipline with wide and diverse applications
  - There is still a nontrivial gap between general principles of data mining and domain-specific, effective data mining tools for particular applications
- Some application domains (covered in this chapter)
  - Biomedical and DNA data analysis
  - Financial data analysis
  - Retail industry
  - Telecommunication industry

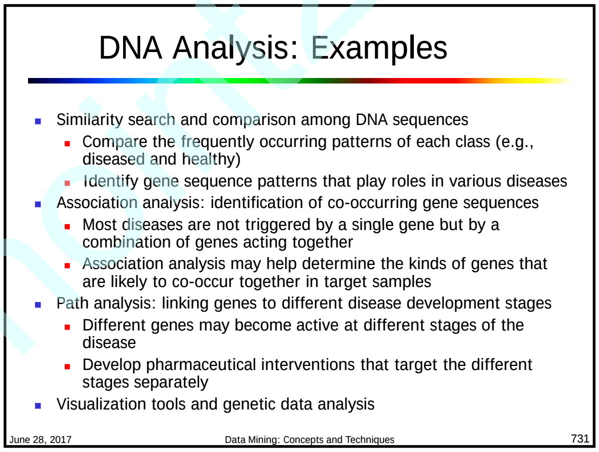
June 28, 2017 Data Mining: Concepts and Techniques 729



Biomedical and DNA Data Analysis

- DNA sequences: 4 basic building blocks (nucleotides): adenine (A), cytosine (C), guanine (G), and thymine (T).
- Gene: a sequence of hundreds of individual nucleotides arranged in a particular order
- Humans have around 30,000 genes
- Tremendous number of ways that the nucleotides can be ordered and sequenced to form distinct genes
- Semantic integration of heterogeneous, distributed genome databases
  - Current: highly distributed, uncontrolled generation and use of a wide variety of DNA data
  - Data cleaning and data integration methods developed in data mining will help

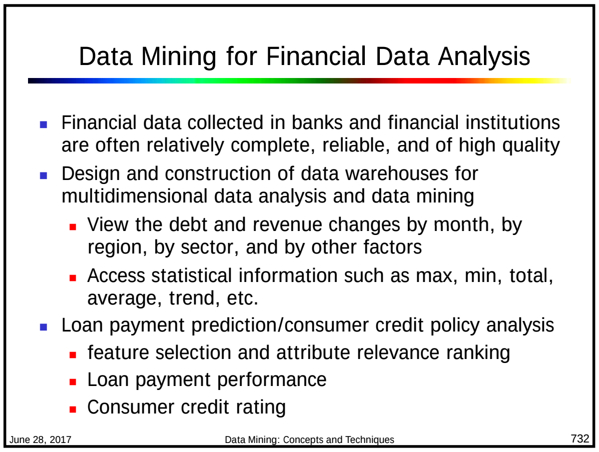
June 28, 2017 Data Mining: Concepts and Techniques 730



DNA Analysis: Examples

- Similarity search and comparison among DNA sequences
  - Compare the frequently occurring patterns of each class (e.g., diseased and healthy)
  - Identify gene sequence patterns that play roles in various diseases
- Association analysis: identification of co-occurring gene sequences
  - Most diseases are not triggered by a single gene but by a combination of genes acting together
  - Association analysis may help determine the kinds of genes that are likely to co-occur together in target samples
- Path analysis: linking genes to different disease development stages
  - Different genes may become active at different stages of the disease
  - Develop pharmaceutical interventions that target the different stages separately
- Visualization tools and genetic data analysis

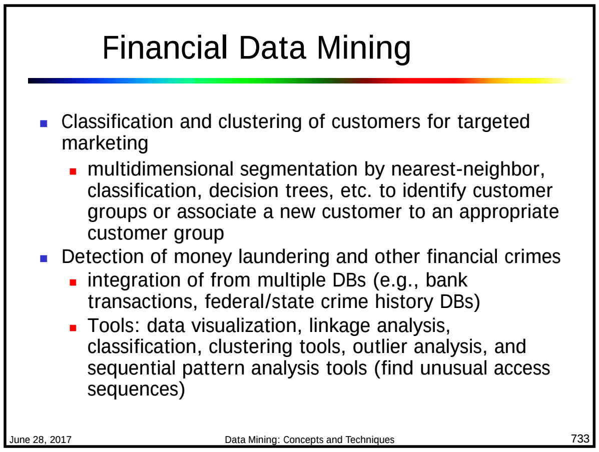
June 28, 2017 Data Mining: Concepts and Techniques 731



Data Mining for Financial Data Analysis

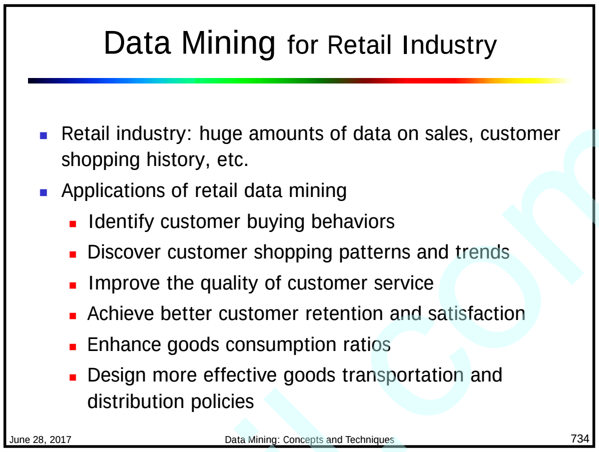
- Financial data collected in banks and financial institutions are often relatively complete, reliable, and of high quality
- Design and construction of data warehouses for multidimensional data analysis and data mining
  - View the debt and revenue changes by month, by region, by sector, and by other factors
  - Access statistical information such as max, min, total, average, trend, etc.
- Loan payment prediction/consumer credit policy analysis
  - feature selection and attribute relevance ranking
  - Loan payment performance
  - Consumer credit rating

June 28, 2017 Data Mining: Concepts and Techniques 732



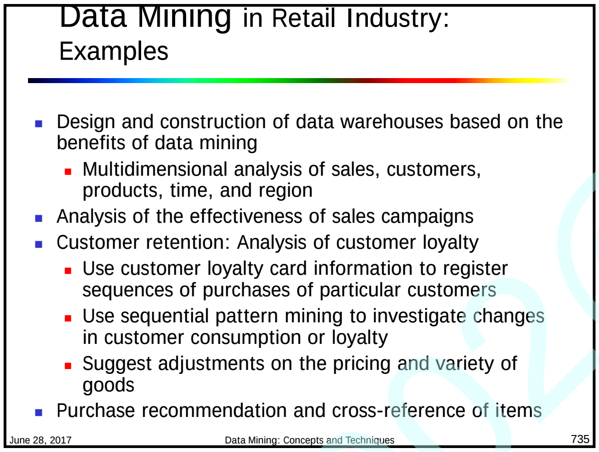
Financial Data Mining

- Classification and clustering of customers for targeted marketing
  - multidimensional segmentation by nearest-neighbor, classification, decision trees, etc. to identify customer groups or associate a new customer to an appropriate customer group
- Detection of money laundering and other financial crimes
  - integration of from multiple DBs (e.g., bank transactions, federal/state crime history DBs)
  - Tools: data visualization, linkage analysis, classification, clustering tools, outlier analysis, and sequential pattern analysis tools (find unusual access sequences)



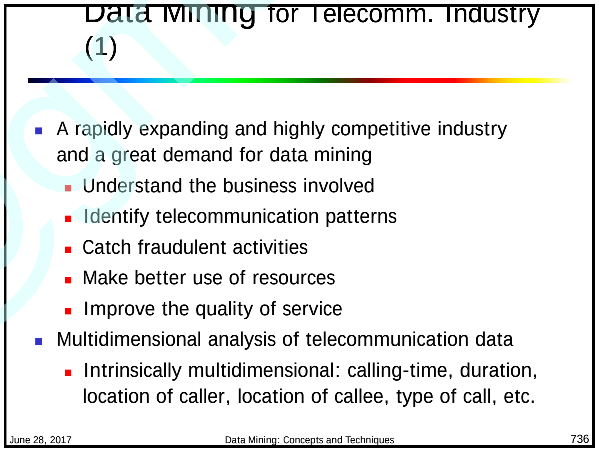
Data Mining for Retail Industry

- Retail industry: huge amounts of data on sales, customer shopping history, etc.
- Applications of retail data mining
  - Identify customer buying behaviors
  - Discover customer shopping patterns and trends
  - Improve the quality of customer service
  - Achieve better customer retention and satisfaction
  - Enhance goods consumption ratios
  - Design more effective goods transportation and distribution policies



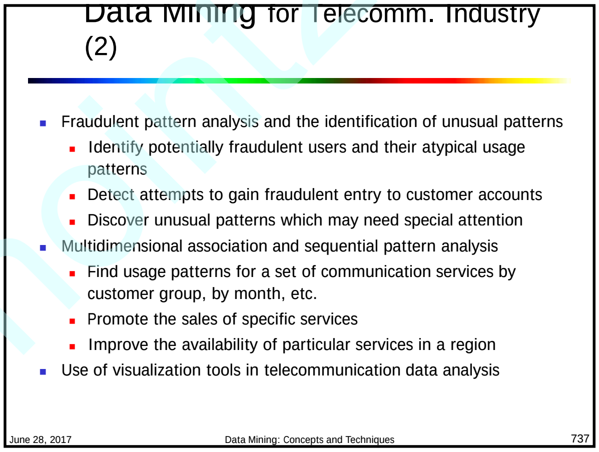
Data Mining in Retail Industry: Examples

- Design and construction of data warehouses based on the benefits of data mining
  - Multidimensional analysis of sales, customers, products, time, and region
- Analysis of the effectiveness of sales campaigns
- Customer retention: Analysis of customer loyalty
  - Use customer loyalty card information to register sequences of purchases of particular customers
  - Use sequential pattern mining to investigate changes in customer consumption or loyalty
  - Suggest adjustments on the pricing and variety of goods
- Purchase recommendation and cross-reference of items



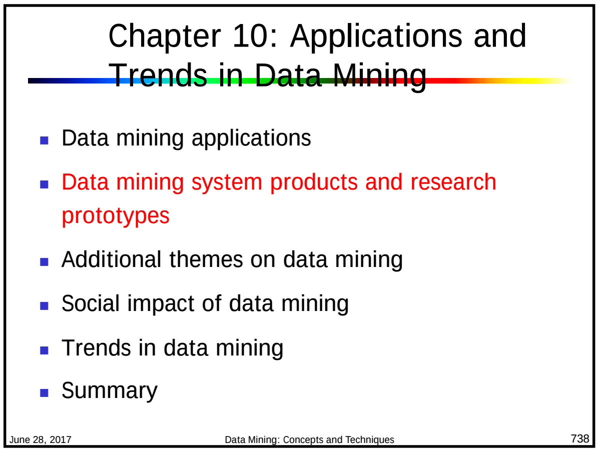
Data Mining for Telecomm. Industry (1)

- A rapidly expanding and highly competitive industry and a great demand for data mining
  - Understand the business involved
  - Identify telecommunication patterns
  - Catch fraudulent activities
  - Make better use of resources
  - Improve the quality of service
- Multidimensional analysis of telecommunication data
  - Intrinsically multidimensional: calling-time, duration, location of caller, location of callee, type of call, etc.



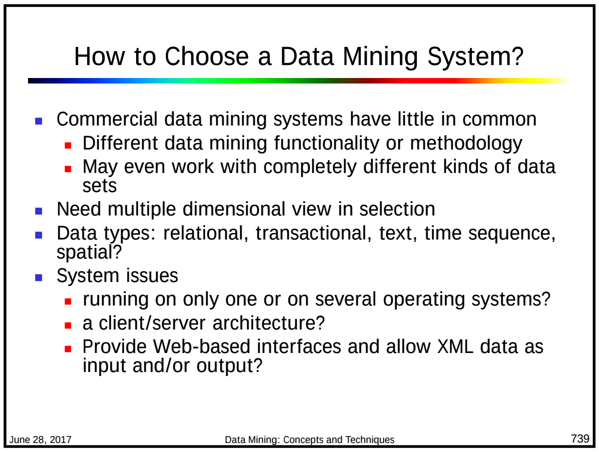
Data Mining for Telecomm. Industry (2)

- Fraudulent pattern analysis and the identification of unusual patterns
  - Identify potentially fraudulent users and their atypical usage patterns
  - Detect attempts to gain fraudulent entry to customer accounts
  - Discover unusual patterns which may need special attention
- Multidimensional association and sequential pattern analysis
  - Find usage patterns for a set of communication services by customer group, by month, etc.
  - Promote the sales of specific services
  - Improve the availability of particular services in a region
- Use of visualization tools in telecommunication data analysis



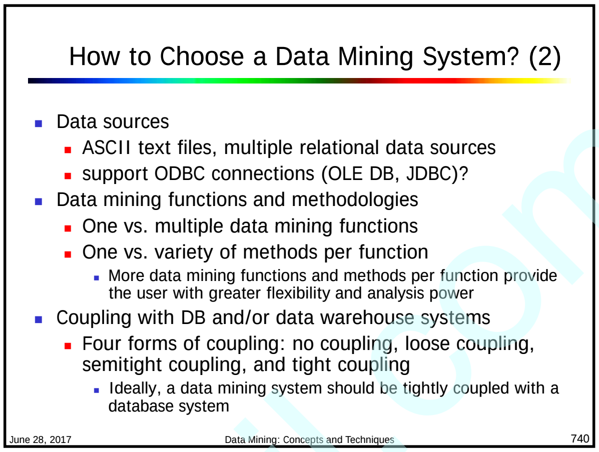
Chapter 10: Applications and Trends in Data Mining

- Data mining applications
- Data mining system products and research prototypes
- Additional themes on data mining
- Social impact of data mining
- Trends in data mining
- Summary



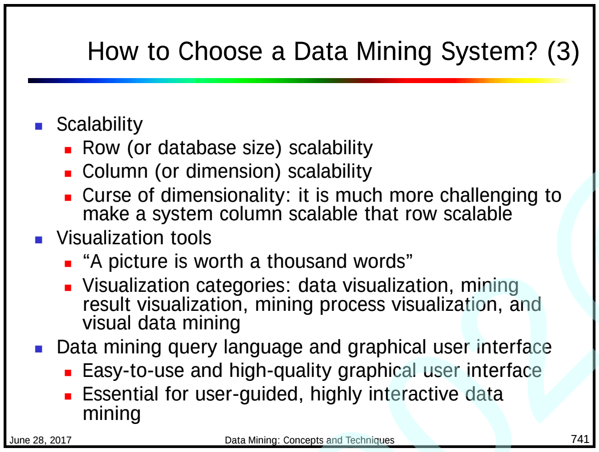
How to Choose a Data Mining System?

- Commercial data mining systems have little in common
  - Different data mining functionality or methodology
  - May even work with completely different kinds of data sets
- Need multiple dimensional view in selection
- Data types: relational, transactional, text, time sequence, spatial?
- System issues
  - running on only one or on several operating systems?
  - a client/server architecture?
  - Provide Web-based interfaces and allow XML data as input and/or output?



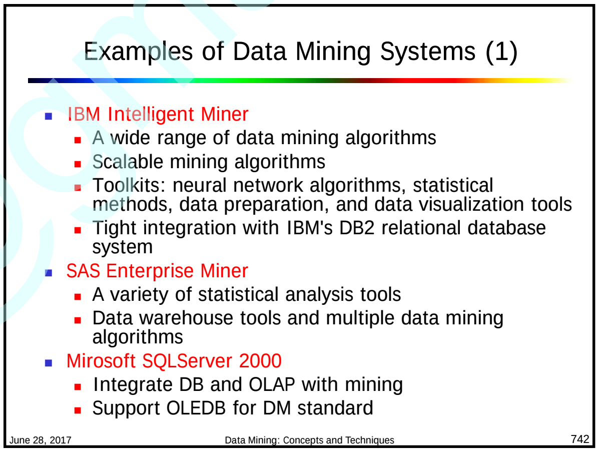
How to Choose a Data Mining System? (2)

- Data sources
  - ASCII text files, multiple relational data sources
  - support ODBC connections (OLE DB, JDBC)?
- Data mining functions and methodologies
  - One vs. multiple data mining functions
  - One vs. variety of methods per function
    - More data mining functions and methods per function provide the user with greater flexibility and analysis power
- Coupling with DB and/or data warehouse systems
  - Four forms of coupling: no coupling, loose coupling, semitight coupling, and tight coupling
    - Ideally, a data mining system should be tightly coupled with a database system



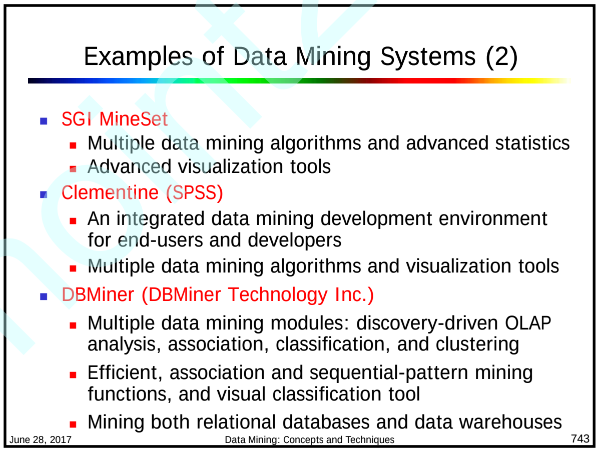
How to Choose a Data Mining System? (3)

- Scalability
  - Row (or database size) scalability
  - Column (or dimension) scalability
  - Curse of dimensionality: it is much more challenging to make a system column scalable that row scalable
- Visualization tools
  - “A picture is worth a thousand words”
  - Visualization categories: data visualization, mining result visualization, mining process visualization, and visual data mining
- Data mining query language and graphical user interface
  - Easy-to-use and high-quality graphical user interface
  - Essential for user-guided, highly interactive data mining



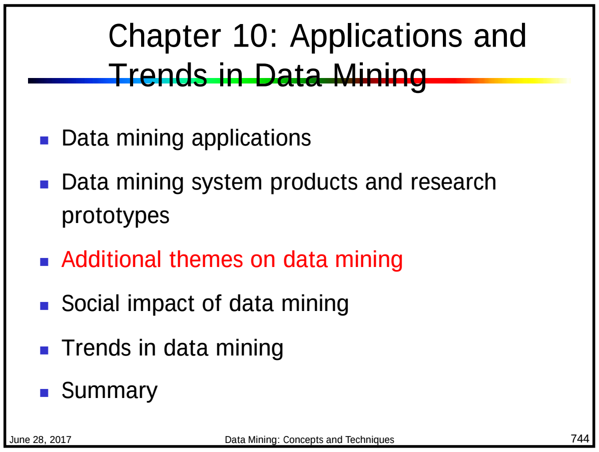
Examples of Data Mining Systems (1)

- IBM Intelligent Miner
  - A wide range of data mining algorithms
  - Scalable mining algorithms
  - Toolkits: neural network algorithms, statistical methods, data preparation, and data visualization tools
  - Tight integration with IBM's DB2 relational database system
- SAS Enterprise Miner
  - A variety of statistical analysis tools
  - Data warehouse tools and multiple data mining algorithms
- Mirosoft SQLServer 2000
  - Integrate DB and OLAP with mining
  - Support OLEDB for DM standard



Examples of Data Mining Systems (2)

- SGI MineSet
  - Multiple data mining algorithms and advanced statistics
  - Advanced visualization tools
- Clementine (SPSS)
  - An integrated data mining development environment for end-users and developers
  - Multiple data mining algorithms and visualization tools
- DBMiner (DBMiner Technology Inc.)
  - Multiple data mining modules: discovery-driven OLAP analysis, association, classification, and clustering
  - Efficient, association and sequential-pattern mining functions, and visual classification tool
  - Mining both relational databases and data warehouses

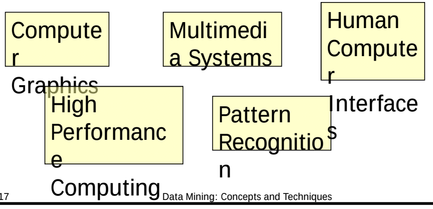


Chapter 10: Applications and Trends in Data Mining

- Data mining applications
- Data mining system products and research prototypes
- Additional themes on data mining
- Social impact of data mining
- Trends in data mining
- Summary

Visual Data Mining

- Visualization: use of computer graphics to create visual images which aid in the understanding of complex, often massive representations of data
- Visual Data Mining: the process of discovering implicit but useful knowledge from large data sets using visualization techniques



Computer Graphics

Multimedia Systems

Human Computer Interfaces

High Performance Computing

Pattern Recognition

Visualization

- Purpose of Visualization
  - Gain insight into an information space by mapping data onto graphical primitives
  - Provide qualitative overview of large data sets
  - Search for patterns, trends, structure, irregularities, relationships among data.
  - Help find interesting regions and suitable parameters for further quantitative analysis.
  - Provide a visual proof of computer representations derived

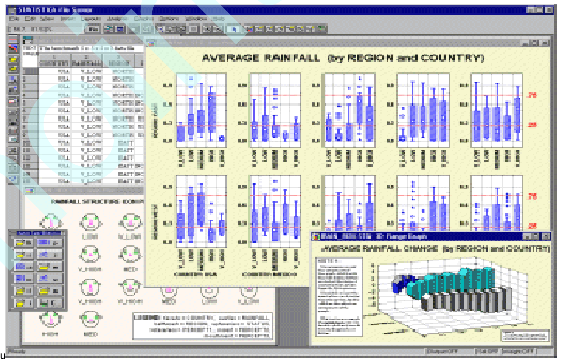
Visual Data Mining & Data Visualization

- Integration of visualization and data mining
  - data visualization
  - data mining result visualization
  - data mining process visualization
  - interactive visual data mining
- Data visualization
  - Data in a database or data warehouse can be viewed
    - at different levels of abstraction
    - as different combinations of attributes or dimensions
  - Data can be presented in various visual forms

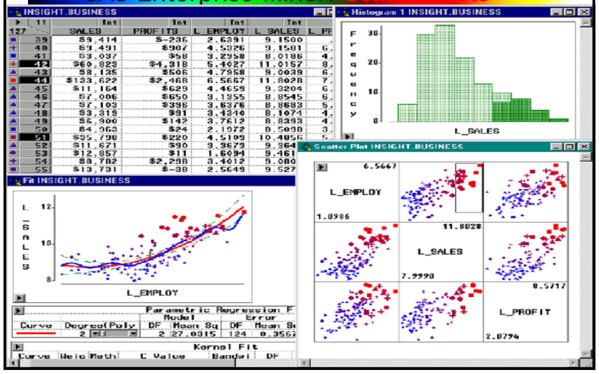
Data Mining Result Visualization

- Presentation of the results or knowledge obtained from data mining in visual forms
- Examples
  - Scatter plots and boxplots (obtained from descriptive data mining)
  - Decision trees
  - Association rules
  - Clusters
  - Outliers
  - Generalized rules

Boxplots from Statsoft: Multiple Variable Combinations


AVERAGE RAINFALL (by REGION and COUNTRY)
AVERAGE RAINFALL CHANGE (by REGION and COUNTRY)

Visualization of Data Mining Results in SAS Enterprise Miner: Scatter Plots



11 / 127	Int SALES	Int PROFITS	Int L_EMPLOY	Int L_SALES	L_PF
39	$9,414	$-236	2.6391	9.1500	.
40	$9,491	$907	4.5326	9.1581	6.
41	$3,037	$58	3.2958	8.0186	4.
42	$60,823	$4,318	5.4027	11.0157	8.
43	$8,135	$506	4.7958	9.0039	6.
44	$133,622	$2,468	6.5667	11.8028	7.
45	$11,164	$629	4.4659	9.3204	6.
46	$7,006	$650	3.1355	8.8545	6.
47	$7,103	$396	3.6376	8.8683	5.
48	$3,319	$91	3.4340	8.1074	4.
49	$6,900	$142	3.7612	8.8393	4.
50	$4,963	$24	2.1972	8.5090	3.
51	$35,790	$220	4.5109	10.4856	5.
52	$11,671	$90	3.3673	9.364	
53	$12,857	$11	1.6094	9.461	
54	$8,782	$2,298	3.4012	9.080	
55	$13,731	$-38	2.5649	9.527	

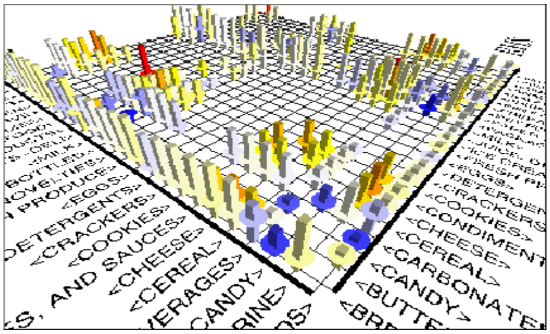
Parametric Regression F

Curve	Degree(Poly	Model DF	Model Mean Sq	Error DF	Error Mean Sq
	2	2	27.0315	124	0.3567

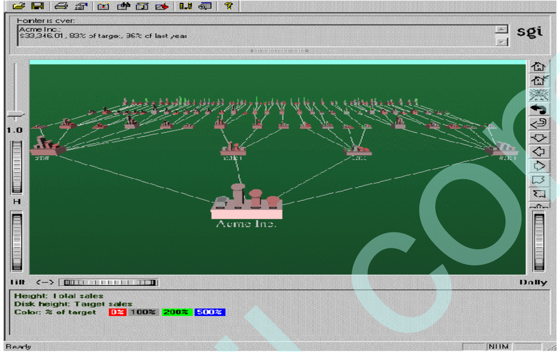
Kernel Fit

Curve	Weig	Meth	C Value	Bandwi	DF

Visualization of Association Rules in SGI/MineSet 3.0

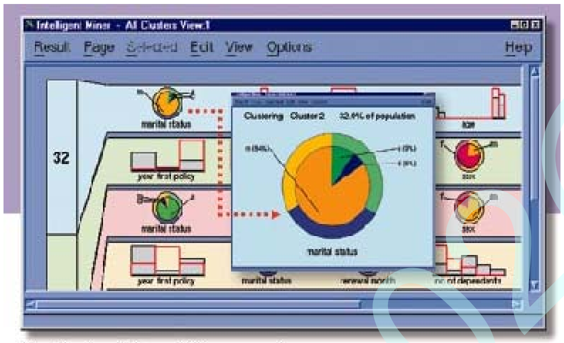


Visualization of a Decision Tree in SGI/MineSet 3.0



Pointer is over:
Acme Inc.:
$33,346.01, 83% of target, 36% of last year
sgi
1.0
H
Acme Inc.
Tilt
Dolly
Height: Total sales
Disk height: Target sales
Color: % of target 0% 100% 200% 500%
Ready
NUM

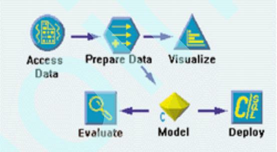
Visualization of Cluster Grouping in IBM Intelligent Miner



Data Mining Process Visualization

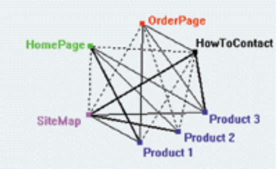
- Presentation of the various processes of data mining in visual forms so that users can see
  - Data extraction process
  - Where the data is extracted
  - How the data is cleaned, integrated, preprocessed, and mined
  - Method selected for data mining
  - Where the results are stored
  - How they may be viewed

Visualization of Data Mining Processes by Clementine



See your solution discovery process clearly

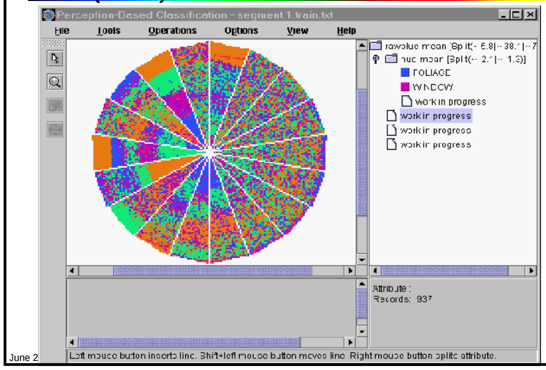
Understand variations with visualized data



Interactive Visual Data Mining

- Using visualization tools in the data mining process to help users make smart data mining decisions
- Example
  - Display the data distribution in a set of attributes using colored sectors or columns (depending on whether the whole space is represented by either a circle or a set of columns)
  - Use the display to which sector should first be selected for classification and where a good split point for this sector may be

Perception-Based Classification (PBC)



rawvalue mean [Split(- 6.8|-- 38.1|-- 7
nuc mear [Split(-- 2.1|-- 1.3)]
🟦 FOLIAGE
🟪 WINDOW
work in progress
work in progress
work in progress
work in progress

Attribute :
Records: 937

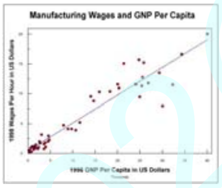
Left mouse button inserts line. Shift+left mouse button moves line. Right mouse button splits attribute.

Audio Data Mining

- Uses audio signals to indicate the patterns of data or the features of data mining results
- An interesting alternative to visual mining
- An inverse task of mining audio (such as music) databases which is to find patterns from audio data
- Visual data mining may disclose interesting patterns using graphical displays, but requires users to concentrate on watching patterns
- Instead, transform patterns into sound and music and listen to pitches, rhythms, tune, and melody in order to identify anything interesting or unusual

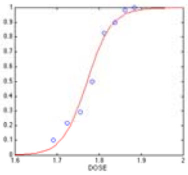
Scientific and Statistical Data Mining (1)

- There are many well-established statistical techniques for data analysis, particularly for numeric data
  - applied extensively to data from scientific experiments and data from economics and the social sciences
- **Regression**
  - predict the value of a response (dependent) variable from one or more predictor (independent) variables where the variables are numeric
  - forms of regression: linear, multiple, weighted, polynomial, nonparametric, and robust



Scientific and Statistical Data Mining (2)

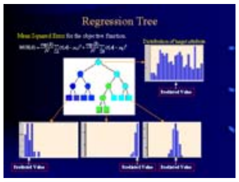
- **Generalized linear models**
  - allow a categorical response variable (or some transformation of it) to be related to a set of predictor variables
  - similar to the modeling of a numeric response variable using linear regression
  - include logistic regression and Poisson regression

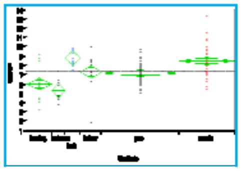


- **Mixed-effect models**
  - For analyzing grouped data, i.e. data that can be classified according to one or more grouping variables
  - Typically describe relationships between a response variable and some covariates in data grouped according to one or more factors

Scientific and Statistical Data Mining (3)

- **Regression trees**
  - Binary trees used for classification and prediction
  - Similar to decision trees:Tests are performed at the internal nodes
  - In a regression tree the mean of the objective attribute is computed and used as the predicted value
- **Analysis of variance**
  - Analyze experimental data for two or more populations described by a numeric response variable and one or more categorical variables (factors)

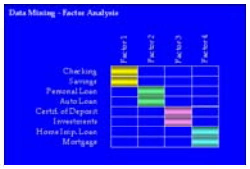

Regression Tree
Mean Squared Error for the objective function
Distribution of target attribute
Predicted Value
Predicted Value
Predicted Value
Predicted Value

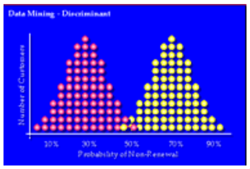


Scientific and Statistical Data Mining (4)

http://www.spss.com/datamine/factor.htm

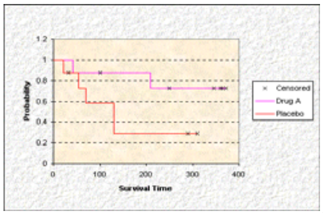
- **Factor analysis**
  - determine which variables are combined to generate a given factor
  - e.g., for many psychiatric data, one can indirectly measure other quantities (such as test scores) that reflect the factor of interest
- **Discriminant analysis**
  - predict a categorical response variable, commonly used in social science
  - Attempts to determine several discriminant functions (linear combinations of the independent variables) that discriminate among the groups defined by the response variable





Scientific and Statistical Data Mining (5)

- **Time series**: many methods such as autoregression, ARIMA (Autoregressive integrated moving-average modeling), long memory time-series modeling
- **Quality control:** displays group summary charts
- **Survival analysis**
  - predicts the probability that a patient undergoing a medical treatment would survive



Theoretical Foundations of Data Mining (1)

- Data reduction
  - The basis of data mining is to reduce the data representation
  - Trades accuracy for speed in response
- Data compression
  - The basis of data mining is to compress the given data by encoding in terms of bits, association rules, decision trees, clusters, etc.
- Pattern discovery
  - The basis of data mining is to discover patterns occurring in the database, such as associations, classification models, sequential patterns, etc.

Theoretical Foundations of Data Mining (2)

- Probability theory
  - The basis of data mining is to discover joint probability distributions of random variables
- Microeconomic view
  - A view of utility: the task of data mining is finding patterns that are interesting only to the extent in that they can be used in the decision-making process of some enterprise
- Inductive databases
  - Data mining is the problem of performing inductive logic on databases,
  - The task is to query the data and the theory (i.e., patterns) of the database
  - Popular among many researchers in database systems

Data Mining and Intelligent Query Answering

- A general framework for the integration of data mining and intelligent query answering
  - Data query: finds concrete data stored in a database; returns exactly what is being asked
  - Knowledge query: finds rules, patterns, and other kinds of knowledge in a database
    - Intelligent (or cooperative) query answering: analyzes the intent of the query and provides generalized, neighborhood or associated information relevant to the query

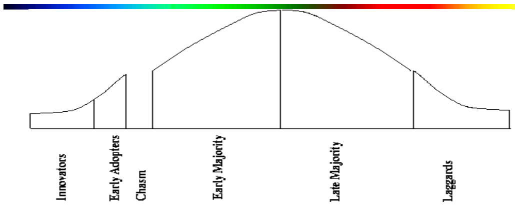
Chapter 10: Applications and Trends in Data Mining

- Data mining applications
- Data mining system products and research prototypes
- Additional themes on data mining
- Social impacts of data mining
- Trends in data mining
- Summary

Is Data Mining a Hype or Will It Be Persistent?

- Data mining is a technology
- Technological life cycle
  - Innovators
  - Early adopters
  - Chasm
  - Early majority
  - Late majority
  - Laggards

Life Cycle of Technology Adoption



- Data mining is at Chasm!?
  - Existing data mining systems are too generic
  - Need business-specific data mining solutions and smooth integration of business logic with data mining functions

Data Mining: Merely Managers' Business or Everyone's?

- Data mining will surely be an important tool for managers' decision making
  - Bill Gates: "Business @ the speed of thought"
- The amount of the available data is increasing, and data mining systems will be more affordable
- Multiple personal uses
  - Mine your family's medical history to identify genetically-related medical conditions
  - Mine the records of the companies you deal with
  - Mine data on stocks and company performance, etc.
- Invisible data mining
  - Build data mining functions into many intelligent tools

Social Impacts: Threat to Privacy and Data Security?

- Is data mining a threat to privacy and data security?
  - "Big Brother", "Big Banker", and "Big Business" are carefully watching you
  - Profiling information is collected every time
    - credit card, debit card, supermarket loyalty card, or frequent flyer card, or apply for any of the above
    - You surf the Web, rent a video, fill out a contest entry form,
    - You pay for prescription drugs, or present you medical care number when visiting the doctor
  - Collection of personal data may be beneficial for companies and consumers, there is also potential for misuse
    - Medical Records, Employee Evaluations, Etc.

Protect Privacy and Data Security

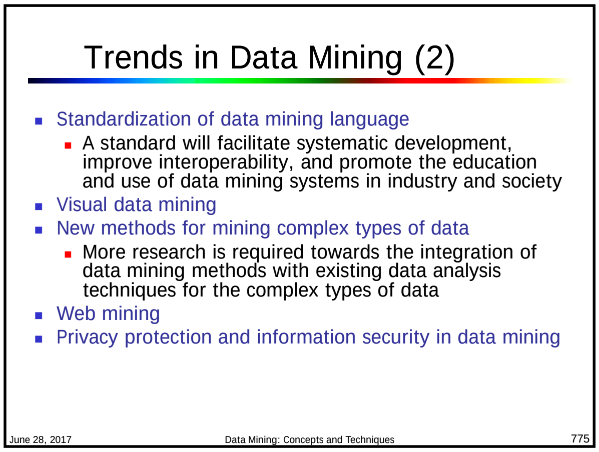
- Fair information practices
  - International guidelines for data privacy protection
  - Cover aspects relating to data collection, purpose, use, quality, openness, individual participation, and accountability
  - Purpose specification and use limitation
  - Openness: Individuals have the right to know what information is collected about them, who has access to the data, and how the data are being used
- Develop and use data security-enhancing techniques
  - Blind signatures
  - Biometric encryption
  - Anonymous databases

Chapter 10: Applications and Trends in Data Mining

- Data mining applications
- Data mining system products and research prototypes
- Additional themes on data mining
- Social impact of data mining
- Trends in data mining
- Summary

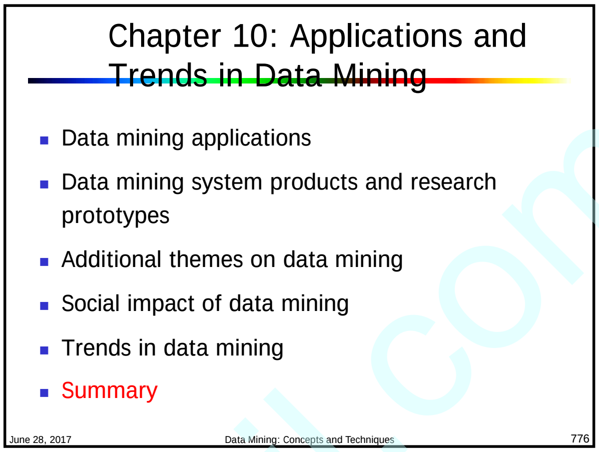
Trends in Data Mining (1)

- Application exploration
  - development of application-specific data mining system
  - Invisible data mining (mining as built-in function)
- Scalable data mining methods
  - Constraint-based mining: use of constraints to guide data mining systems in their search for interesting patterns
- Integration of data mining with database systems, data warehouse systems, and Web database systems
- Invisible data mining



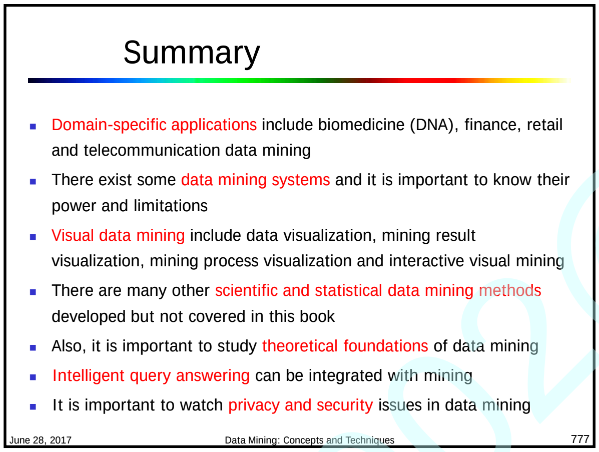
Trends in Data Mining (2)

- Standardization of data mining language
  - A standard will facilitate systematic development, improve interoperability, and promote the education and use of data mining systems in industry and society
- Visual data mining
- New methods for mining complex types of data
  - More research is required towards the integration of data mining methods with existing data analysis techniques for the complex types of data
- Web mining
- Privacy protection and information security in data mining



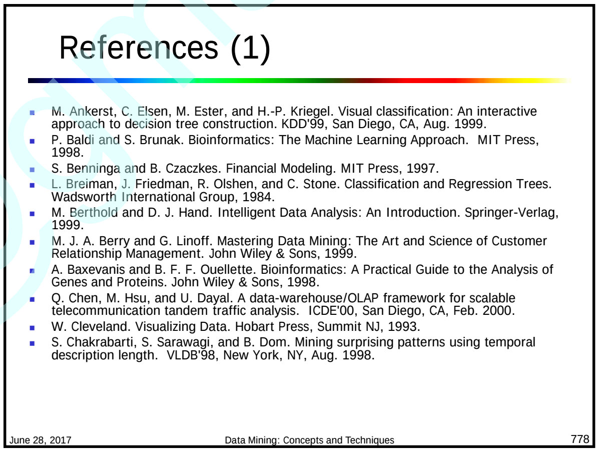
Chapter 10: Applications and Trends in Data Mining

- Data mining applications
- Data mining system products and research prototypes
- Additional themes on data mining
- Social impact of data mining
- Trends in data mining
- Summary



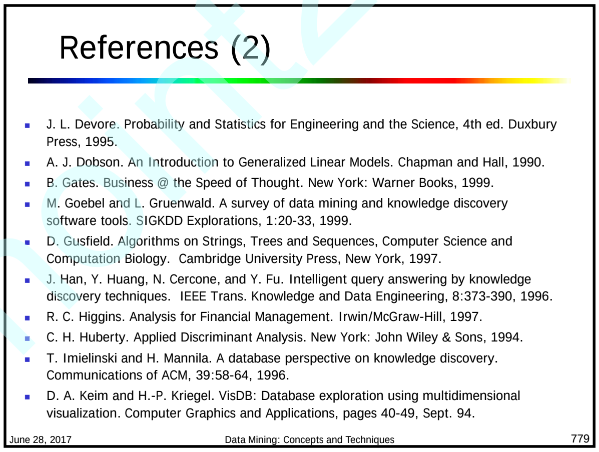
Summary

- Domain-specific applications include biomedicine (DNA), finance, retail and telecommunication data mining
- There exist some data mining systems and it is important to know their power and limitations
- Visual data mining include data visualization, mining result visualization, mining process visualization and interactive visual mining
- There are many other scientific and statistical data mining methods developed but not covered in this book
- Also, it is important to study theoretical foundations of data mining
- Intelligent query answering can be integrated with mining
- It is important to watch privacy and security issues in data mining



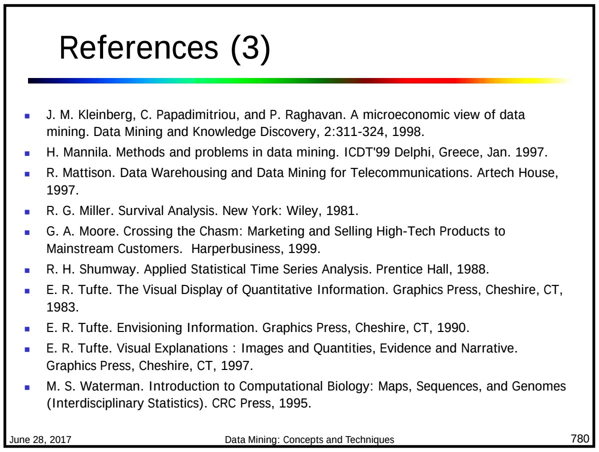
References (1)

- M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel. Visual classification: An interactive approach to decision tree construction. KDD'99, San Diego, CA, Aug. 1999.
- P. Baldi and S. Brunak. Bioinformatics: The Machine Learning Approach. MIT Press, 1998.
- S. Benninga and B. Czaczkes. Financial Modeling. MIT Press, 1997.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth International Group, 1984.
- M. Berthold and D. J. Hand. Intelligent Data Analysis: An Introduction. Springer-Verlag, 1999.
- M. J. A. Berry and G. Linoff. Mastering Data Mining: The Art and Science of Customer Relationship Management. John Wiley & Sons, 1999.
- A. Baxevanis and B. F. F. Ouellette. Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins. John Wiley & Sons, 1998.
- Q. Chen, M. Hsu, and U. Dayal. A data-warehouse/OLAP framework for scalable telecommunication tandem traffic analysis. ICDE'00, San Diego, CA, Feb. 2000.
- W. Cleveland. Visualizing Data. Hobart Press, Summit NJ, 1993.
- S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising patterns using temporal description length. VLDB'98, New York, NY, Aug. 1998.



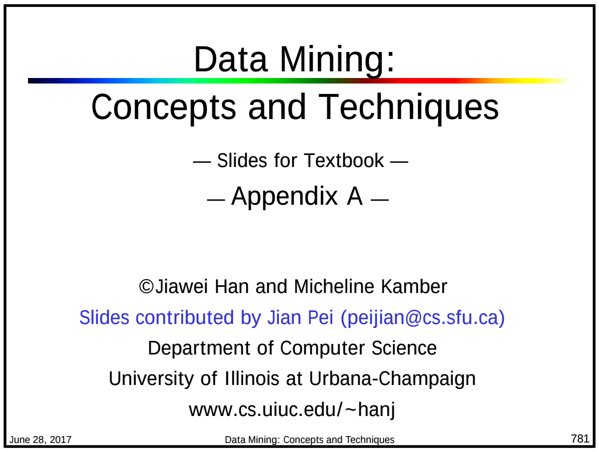
References (2)

- J. L. Devore. Probability and Statistics for Engineering and the Science, 4th ed. Duxbury Press, 1995.
- A. J. Dobson. An Introduction to Generalized Linear Models. Chapman and Hall, 1990.
- B. Gates. Business @ the Speed of Thought. New York: Warner Books, 1999.
- M. Goebel and L. Gruenwald. A survey of data mining and knowledge discovery software tools. SIGKDD Explorations, 1:20-33, 1999.
- D. Gusfield. Algorithms on Strings, Trees and Sequences, Computer Science and Computation Biology. Cambridge University Press, New York, 1997.
- J. Han, Y. Huang, N. Cercone, and Y. Fu. Intelligent query answering by knowledge discovery techniques. IEEE Trans. Knowledge and Data Engineering, 8:373-390, 1996.
- R. C. Higgins. Analysis for Financial Management. Irwin/McGraw-Hill, 1997.
- C. H. Huberty. Applied Discriminant Analysis. New York: John Wiley & Sons, 1994.
- T. Imielinski and H. Mannila. A database perspective on knowledge discovery. Communications of ACM, 39:58-64, 1996.
- D. A. Keim and H.-P. Kriegel. VisDB: Database exploration using multidimensional visualization. Computer Graphics and Applications, pages 40-49, Sept. 94.



References (3)

- J. M. Kleinberg, C. Papadimitriou, and P. Raghavan. A microeconomic view of data mining. Data Mining and Knowledge Discovery, 2:311-324, 1998.
- H. Mannila. Methods and problems in data mining. ICDT'99 Delphi, Greece, Jan. 1997.
- R. Mattison. Data Warehousing and Data Mining for Telecommunications. Artech House, 1997.
- R. G. Miller. Survival Analysis. New York: Wiley, 1981.
- G. A. Moore. Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers. Harperbusiness, 1999.
- R. H. Shumway. Applied Statistical Time Series Analysis. Prentice Hall, 1988.
- E. R. Tufte. The Visual Display of Quantitative Information. Graphics Press, Cheshire, CT, 1983.
- E. R. Tufte. Envisioning Information. Graphics Press, Cheshire, CT, 1990.
- E. R. Tufte. Visual Explanations : Images and Quantities, Evidence and Narrative. Graphics Press, Cheshire, CT, 1997.
- M. S. Waterman. Introduction to Computational Biology: Maps, Sequences, and Genomes (Interdisciplinary Statistics). CRC Press, 1995.

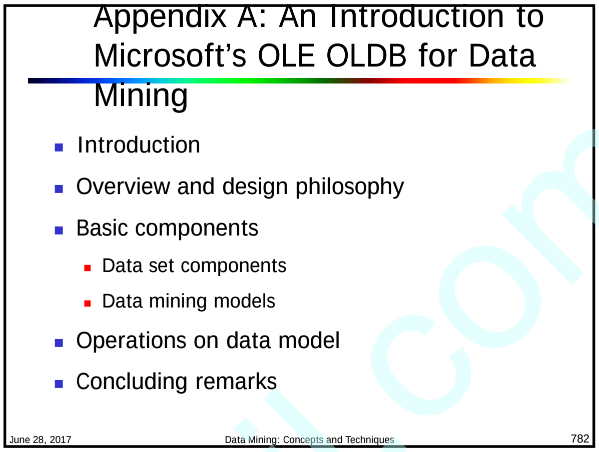


Data Mining:
Concepts and Techniques

— Slides for Textbook —

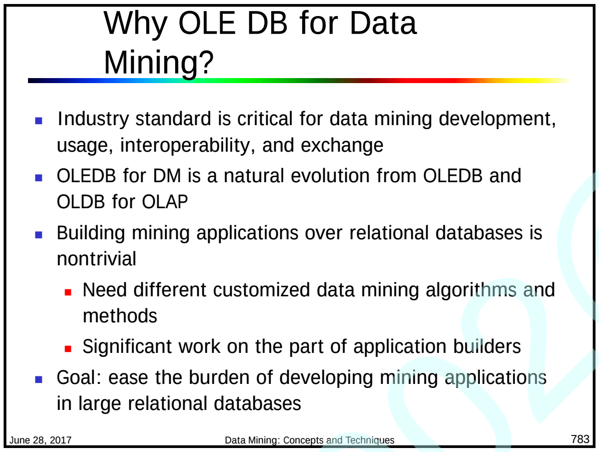
— Appendix A —

©Jiawei Han and Micheline Kamber
Slides contributed by Jian Pei (peijian@cs.sfu.ca)
Department of Computer Science
University of Illinois at Urbana-Champaign
www.cs.uiuc.edu/~hanj



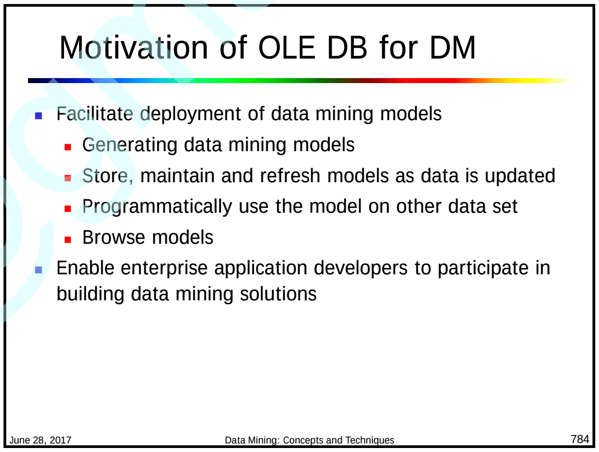
Appendix A: An Introduction to Microsoft's OLE OLDB for Data Mining

- Introduction
- Overview and design philosophy
- Basic components
  - Data set components
  - Data mining models
- Operations on data model
- Concluding remarks



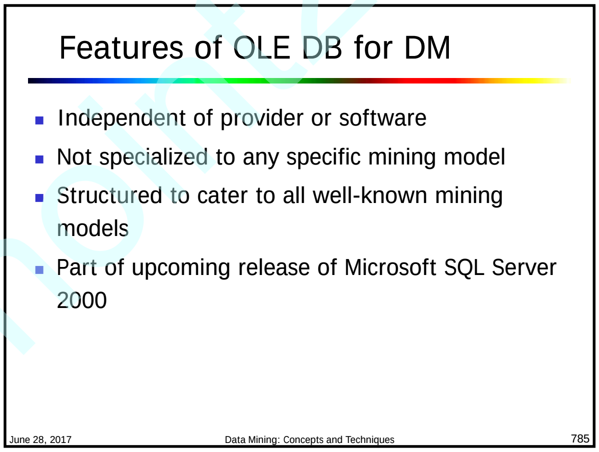
Why OLE DB for Data Mining?

- Industry standard is critical for data mining development, usage, interoperability, and exchange
- OLEDB for DM is a natural evolution from OLEDB and OLDB for OLAP
- Building mining applications over relational databases is nontrivial
  - Need different customized data mining algorithms and methods
  - Significant work on the part of application builders
- Goal: ease the burden of developing mining applications in large relational databases



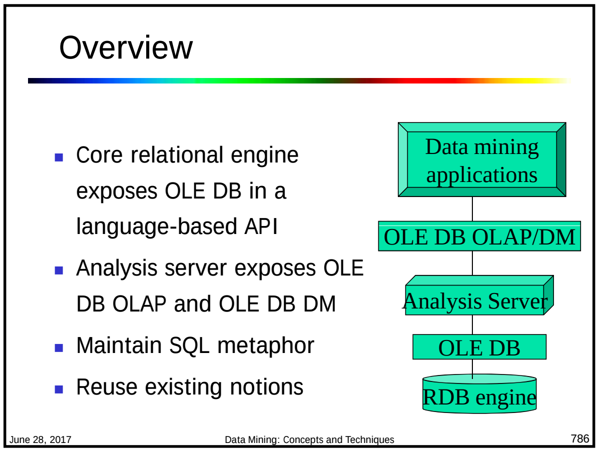
Motivation of OLE DB for DM

- Facilitate deployment of data mining models
  - Generating data mining models
  - Store, maintain and refresh models as data is updated
  - Programmatically use the model on other data set
  - Browse models
- Enable enterprise application developers to participate in building data mining solutions



Features of OLE DB for DM

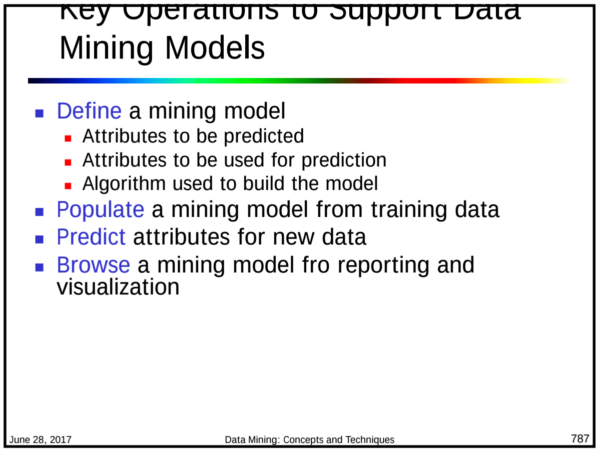
- Independent of provider or software
- Not specialized to any specific mining model
- Structured to cater to all well-known mining models
- Part of upcoming release of Microsoft SQL Server 2000



Overview

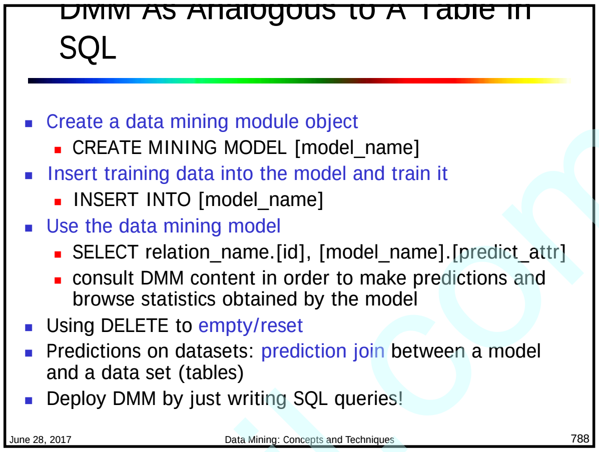
- Core relational engine exposes OLE DB in a language-based API
- Analysis server exposes OLE DB OLAP and OLE DB DM
- Maintain SQL metaphor
- Reuse existing notions

Data mining applications
OLE DB OLAP/DM
Analysis Server
OLE DB
RDB engine



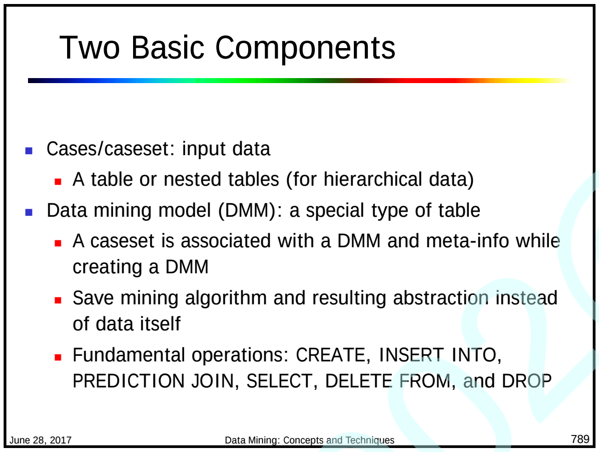
Key Operations to Support Data Mining Models

- Define a mining model
  - Attributes to be predicted
  - Attributes to be used for prediction
  - Algorithm used to build the model
- Populate a mining model from training data
- Predict attributes for new data
- Browse a mining model fro reporting and visualization



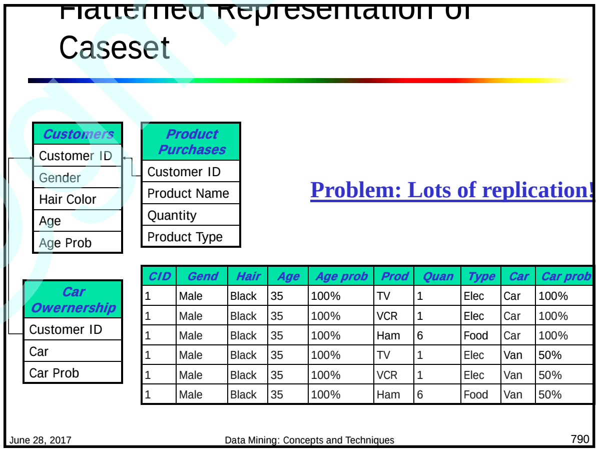
DMM As Analogous to A Table in SQL

- Create a data mining module object
  - CREATE MINING MODEL [model_name]
- Insert training data into the model and train it
  - INSERT INTO [model_name]
- Use the data mining model
  - SELECT relation_name.[id], [model_name].[predict_attr]
  - consult DMM content in order to make predictions and browse statistics obtained by the model
- Using DELETE to empty/reset
- Predictions on datasets: prediction join between a model and a data set (tables)
- Deploy DMM by just writing SQL queries!



Two Basic Components

- Cases/caseset: input data
  - A table or nested tables (for hierarchical data)
- Data mining model (DMM): a special type of table
  - A caseset is associated with a DMM and meta-info while creating a DMM
  - Save mining algorithm and resulting abstraction instead of data itself
  - Fundamental operations: CREATE, INSERT INTO, PREDICTION JOIN, SELECT, DELETE FROM, and DROP



Flattened Representation of Caseset

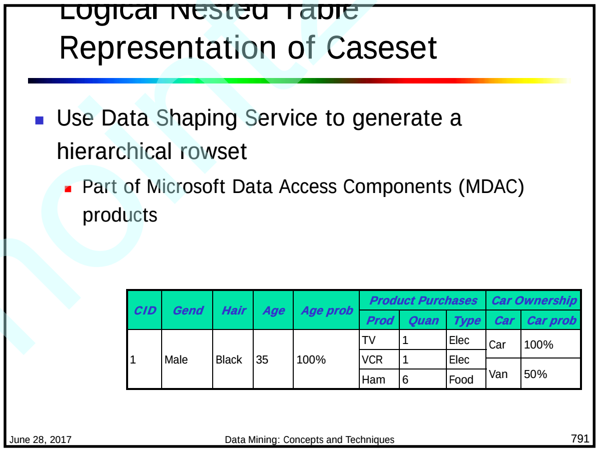
Customers
Customer ID
Gender
Hair Color
Age
Age Prob

Product Purchases
Customer ID
Product Name
Quantity
Product Type

Car Owernership
Customer ID
Car
Car Prob

Problem: Lots of replication!

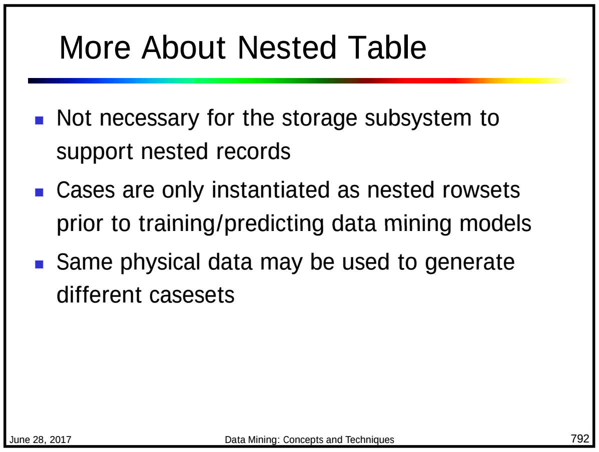
CID	Gend	Hair	Age	Age prob	Prod	Quan	Type	Car	Car prob
1	Male	Black	35	100%	TV	1	Elec	Car	100%
1	Male	Black	35	100%	VCR	1	Elec	Car	100%
1	Male	Black	35	100%	Ham	6	Food	Car	100%
1	Male	Black	35	100%	TV	1	Elec	Van	50%
1	Male	Black	35	100%	VCR	1	Elec	Van	50%
1	Male	Black	35	100%	Ham	6	Food	Van	50%



Logical Nested Table Representation of Caseset

- Use Data Shaping Service to generate a hierarchical rowset
  - Part of Microsoft Data Access Components (MDAC) products

CID	Gend	Hair	Age	Age prob	Product Purchases			Car Ownership	
					Prod	Quan	Type	Car	Car prob
1	Male	Black	35	100%	TV	1	Elec	Car	100%
					VCR	1	Elec	Van	50%
					Ham	6	Food		



More About Nested Table

- Not necessary for the storage subsystem to support nested records
- Cases are only instantiated as nested rowsets prior to training/predicting data mining models
- Same physical data may be used to generate different casesets

Defining A Data Mining Model

- The name of the model
- The algorithm and parameters
- The columns of caseset and the relationships among columns
- “Source columns” and “prediction columns”

Example

```
CREATE MINING MODEL [Age Prediction]                                    %Name of Model
(
[Customer ID]       LONG      KEY,                                      %source column
[Gender]            TEXT      DISCRETE,                                 %source column
[Age]               Double    DISCRETIZED() PREDICT,                    %prediction column
[Product Purchases]           TABLE                                     %source column
(
[Product Name]      TEXT      KEY,                                      %source column
[Quantity]          DOUBLE    NORMAL CONTINUOUS,                        %source column
[Product Type]      TEXT      DISCRETE RELATED TO [Product Name]
                                                                        %source column
))
USING [Decision_Trees_101]                                              %Mining algorithm used
```

Column Specifiers

- KEY
- ATTRIBUTE
- RELATION (RELATED TO clause)
- QUALIFIER (OF clause)
  - PROBABILITY: [0, 1]
  - VARIANCE
  - SUPPORT
  - PROBABILITY-VARIANCE
  - ORDER
  - TABLE

Attribute Types

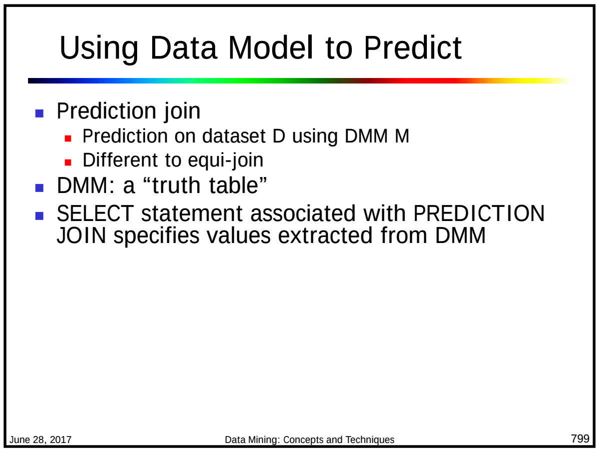
- DISCRETE
- ORDERED
- CYCLICAL
- CONTINOUS
- DISCRETIZED
- SEQUENCE_TIME

Populating A DMM

- Use INSERT INTO statement
- Consuming a case using the data mining model
- Use SHAPE statement to create the nested table from the input data

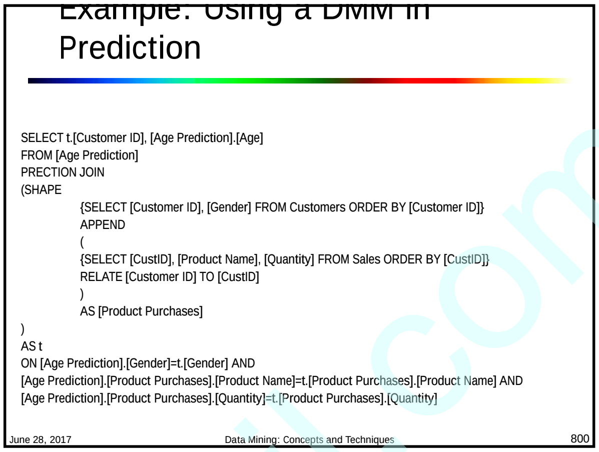
Example: Populating a DMM

```
INSERT INTO [Age Prediction]
(
[Customer ID], [Gender], [Age],
[Product Purchases](SKIP, [Product Name], [Quantity], [Product Type])
)
SHAPE
{SELECT [Customer ID], [Gender], [Age] FROM Customers ORDER BY [Customer ID]}
APPEND
{SELECT [CustID], {product Name], [Quantity], [Product Type] FROM Sales
ORDER BY [CustID]}
RELATE [Customer ID] TO [CustID]
)
AS [Product Purchases]
```



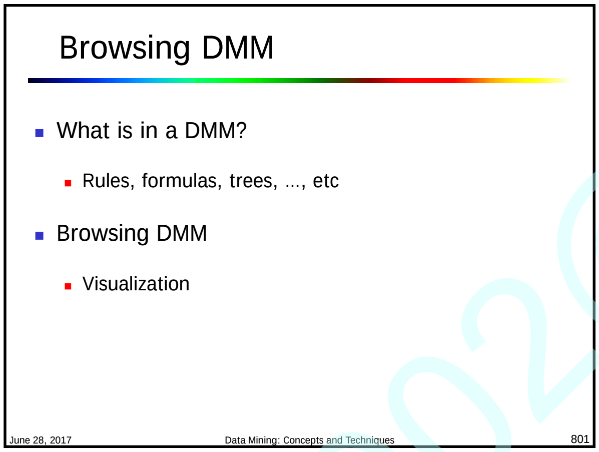
Using Data Model to Predict

- Prediction join
  - Prediction on dataset D using DMM M
  - Different to equi-join
- DMM: a “truth table”
- SELECT statement associated with PREDICTION JOIN specifies values extracted from DMM



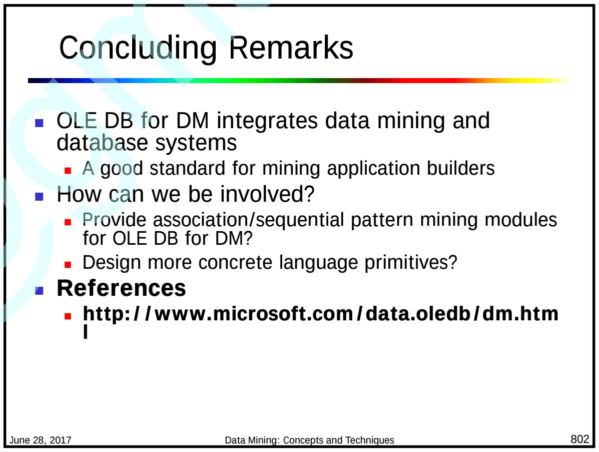
Example: Using a DMM in Prediction

```
SELECT t.[Customer ID], [Age Prediction].[Age]
FROM [Age Prediction]
PRECTION JOIN
(SHAPE
            {SELECT [Customer ID], [Gender] FROM Customers ORDER BY [Customer ID]}
            APPEND
            (
            {SELECT [CustID], [Product Name], [Quantity] FROM Sales ORDER BY [CustID]}
            RELATE [Customer ID] TO [CustID]
            )
            AS [Product Purchases]
)
AS t
ON [Age Prediction].[Gender]=t.[Gender] AND
[Age Prediction].[Product Purchases].[Product Name]=t.[Product Purchases].[Product Name] AND
[Age Prediction].[Product Purchases].[Quantity]=t.[Product Purchases].[Quantity]
```



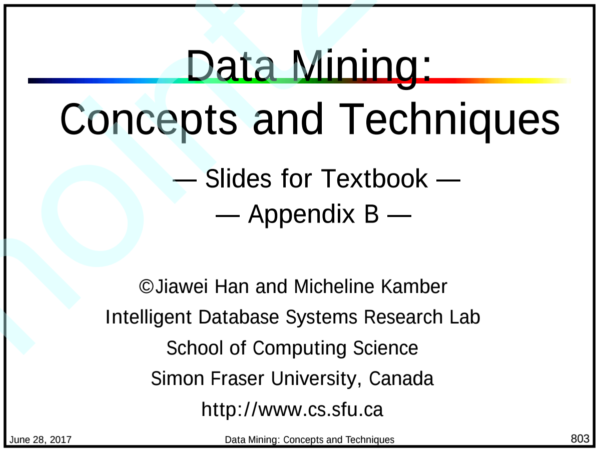
Browsing DMM

- What is in a DMM?
  - Rules, formulas, trees, …, etc
- Browsing DMM
  - Visualization



Concluding Remarks

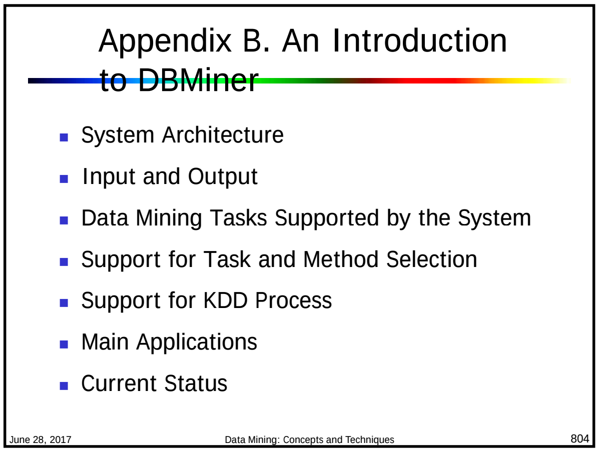
- OLE DB for DM integrates data mining and database systems
  - A good standard for mining application builders
- How can we be involved?
  - Provide association/sequential pattern mining modules for OLE DB for DM?
  - Design more concrete language primitives?
- **References**
  - **http://www.microsoft.com/data.oledb/dm.html**



Data Mining: Concepts and Techniques

— Slides for Textbook —
— Appendix B —

©Jiawei Han and Micheline Kamber
Intelligent Database Systems Research Lab
School of Computing Science
Simon Fraser University, Canada
http://www.cs.sfu.ca



Appendix B. An Introduction to DBMiner

- System Architecture
- Input and Output
- Data Mining Tasks Supported by the System
- Support for Task and Method Selection
- Support for KDD Process
- Main Applications
- Current Status

System Architecture

- DBMiner: A data mining system originated in Intelligent Database Systems Lab and further developed by DBMiner Technology Inc.
- OLAM (on-line analytical mining) architecture for interactive mining of multi-level knowledge in both RDBMS and data warehouses
- Mining knowledge on Microsoft SQLServer 7.0 databases and/or data warehouses
- Multiple mining functions: discovery-driven OLAP, association, classification and clustering

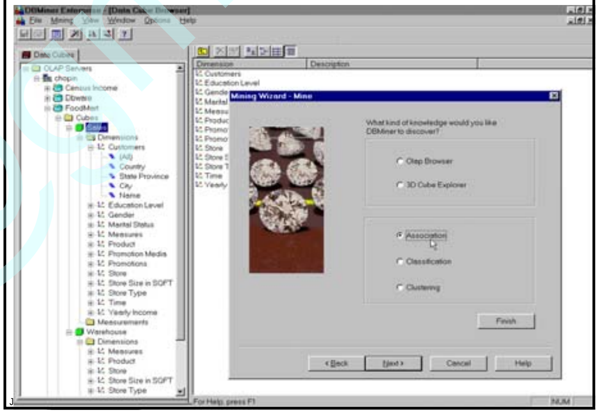
Input and Output

- Input: SQLServer 7.0 data cubes which are constructed from single or multiple relational tables, data warehouses or spread sheets (with OLEDB and RDBMS connections)
- Multiple outputs
  - Summarization and discovery-driven OLAP: crosstabs and graphical outputs using MS/Excel2000
  - Association: rule tables, rule planes and ball graphs
  - Classification: decision trees and decision tables
  - Clustering: maps and summarization graphs
  - Others:
    - Data and cube views
    - Visualization of concept hierarchies
    - Visualization for task management
    - Visualization of 2-D and 3-D boxplots

Data Mining Tasks

- DBMiner covers the following functions
  - Discovery-driven, OLAP-based multi-dimensional analysis
  - Association and frequent pattern analysis
  - Classification (decision tree analysis)
  - Cluster analysis
  - 3-D cube viewer and analyzer
- Other function
  - OLAP service, cube exploration, statistical analysis
  - Sequential pattern analysis (under development)
  - Visual classification (under development)

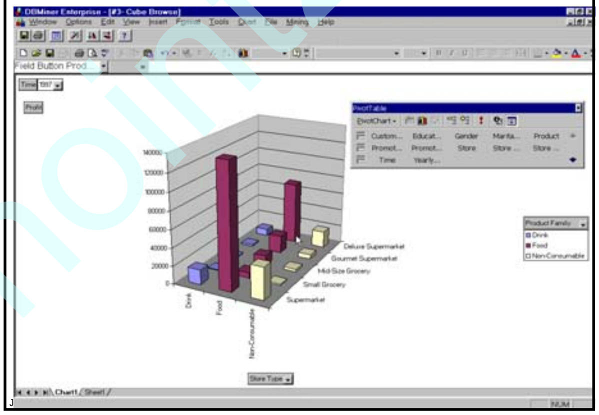
DBMiner Data and Mining Views (Working Panel)



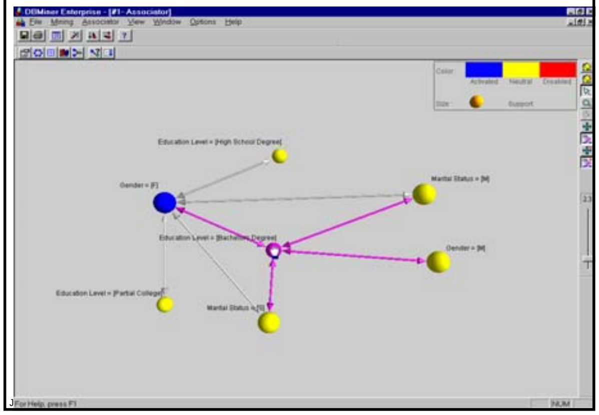
What kind of knowledge would you like DBMiner to discover?

○ Olap Browser
○ 3D Cube Explorer
● Association
○ Classification
○ Clustering

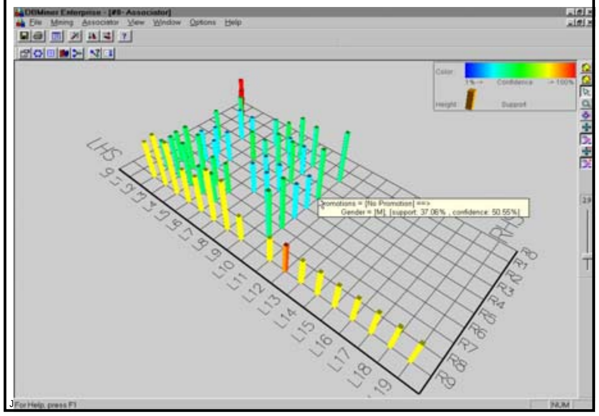
OLAP (Summarization) Display Using MS/Excel 2000



Market-Basket-Analysis (Association)—Ball graph

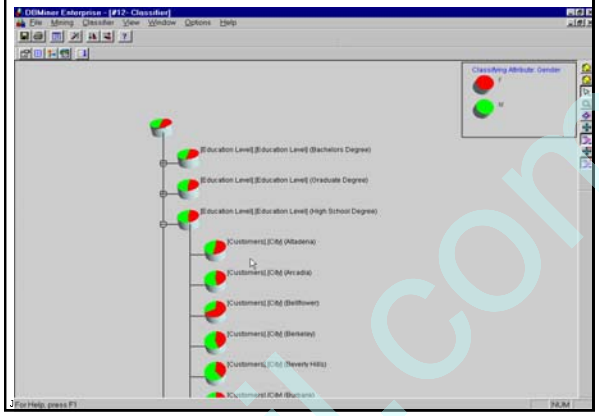

DBMiner Enterprise - [#1- Associator]
File Mining Associator View Window Options Help
Color
Activated
Neutral
Disabled
Size
Support
Education Level = [High School Degree]
Gender = [F]
Marital Status = [M]
Education Level = [Bachelors Degree]
Gender = [M]
Education Level = [Partial College]
Marital Status = [S]
For Help, press F1
NUM

Display of Association Rules in Rule Plane Form

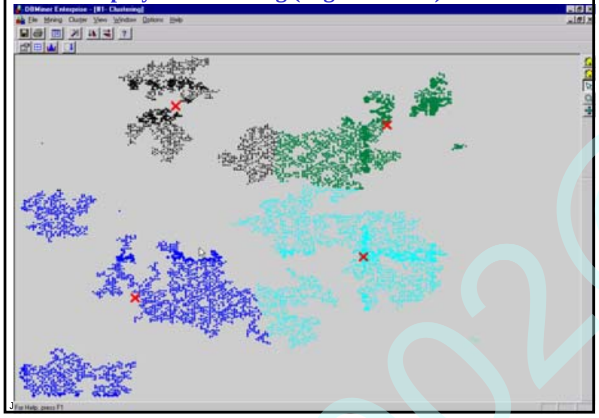

DBMiner Enterprise - [#8- Associator]
File Mining Associator View Window Options Help
Color
1% -> Confidence -> 100%
Height Support
LHS
L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14 L15 L16 L17 L18 L19
RHS
R0 R1 R2 R3 R4 R5 R6 R7 R8 R9
For Help, press F1
NUM

Promotions = [No Promotion] ==>
Gender = [M], [support: 37.06% , confidence: 50.55%]

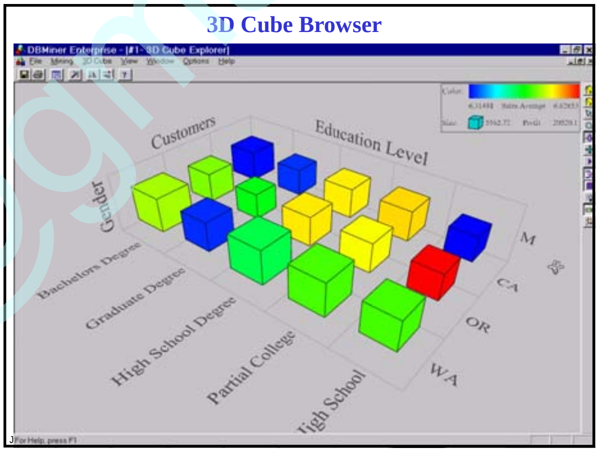
Display of Decision Tree (Classification Results)


DBMiner Enterprise - [#12- Classifier]
File Mining Classifier View Window Options Help
Classifying Attribute: Gender
F
M
[Education Level].[Education Level] (Bachelors Degree)
[Education Level].[Education Level] (Graduate Degree)
[Education Level].[Education Level] (High School Degree)
[Customers].[City] (Altadena)
[Customers].[City] (Arcadia)
[Customers].[City] (Bellflower)
[Customers].[City] (Berkeley)
[Customers].[City] (Beverly Hills)
[Customers].[City] (Burbank)
For Help, press F1
NUM

Display of Clustering (Segmentation) Results


DBMiner Enterprise - [#1- Clustering]
File Mining Cluster View Window Options Help
For Help, press F1

3D Cube Browser

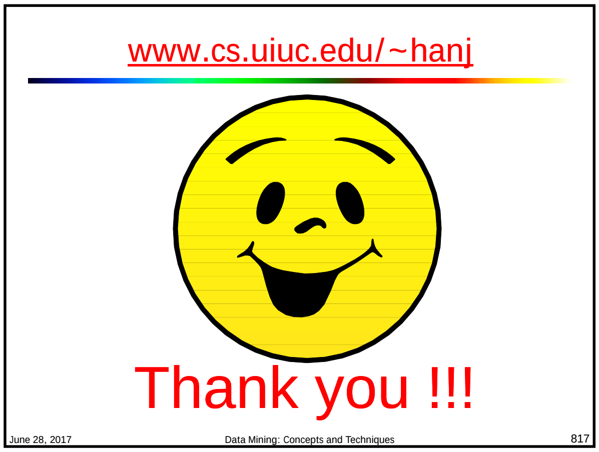


Current Status

- Evolving to DBMiner 3.0
  - Smooth integration of relational database and data warehouse systems
  - Support Microsoft OLEDB for Data Mining
  - Integrates naturally with Microsoft SQLServer 2000 Analysis Service, as one of Microsoft SQLServer 2000 Analysis Service providers
  - Adding fast association mining, sequential pattern mining and gradient mining methods
  - Adding predictive associative classification method
- Towards RetailMiner, WebMiner, GeoMiner, and Bio-Miner

Contact

- For licensing, purchasing and other issues
  - Please consult and contact www.dbminer.com
- Welcome application-oriented in-depth development contract
- Welcome R&D collaborations, joint research and development, technology licensing, and product/company acquisition



www.cs.uiuc.edu/~hanj

Thank you !!!