Data Mining: Concepts and Techniques

— Slides for Textbook —

— Chapter 1 —

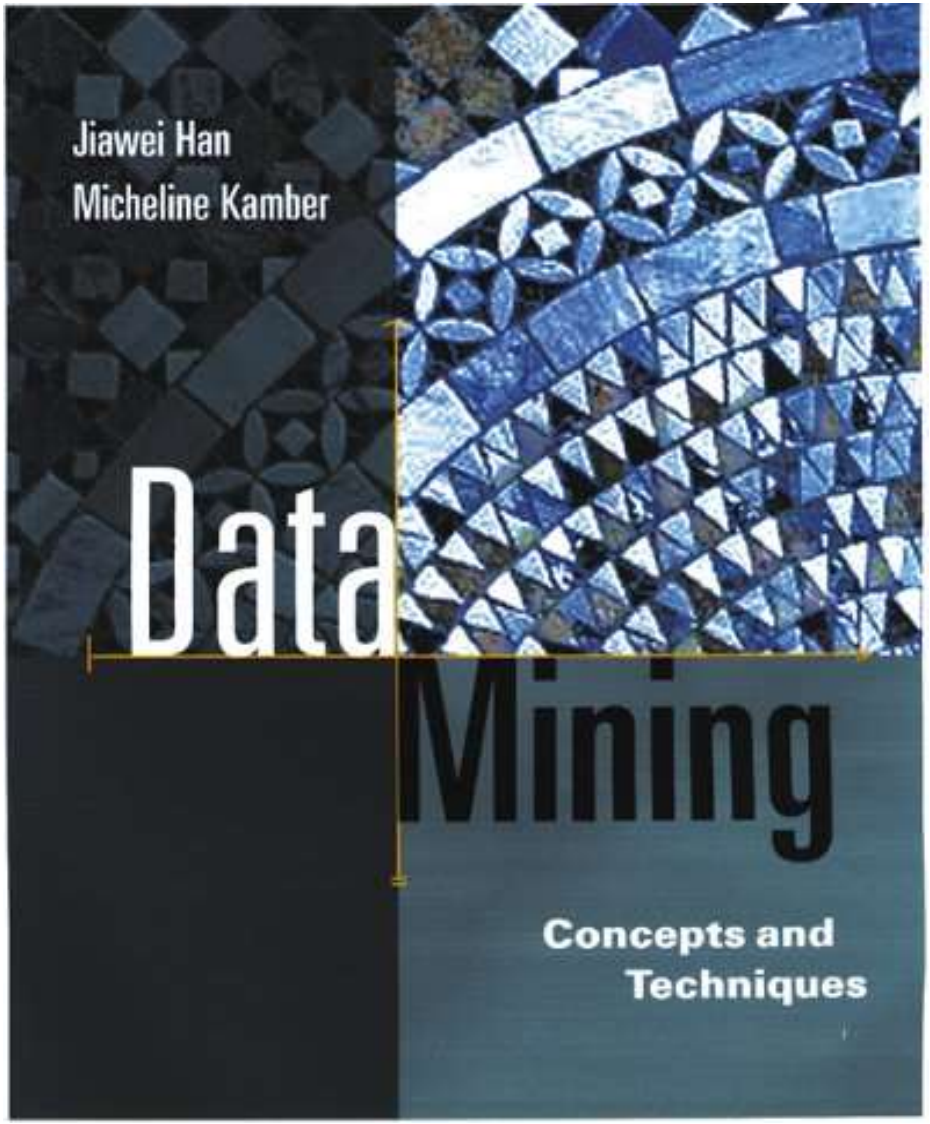
©Jiawei Han and Micheline Kamber

Department of Computer Science

University of Illinois at Urbana-Champaign

www.cs.uiuc.edu/~hanj

Data Mining: Concepts and Techniques



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

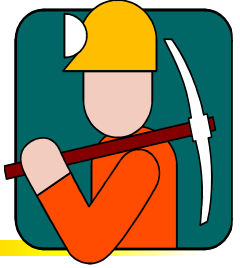
Necessity Is the Mother of Invention

- 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

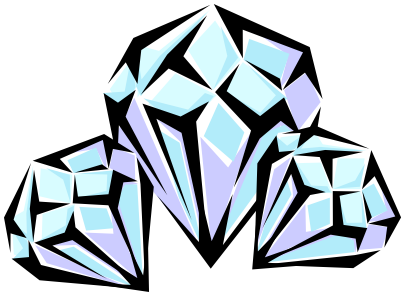
Evolution of Database Technology

- 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

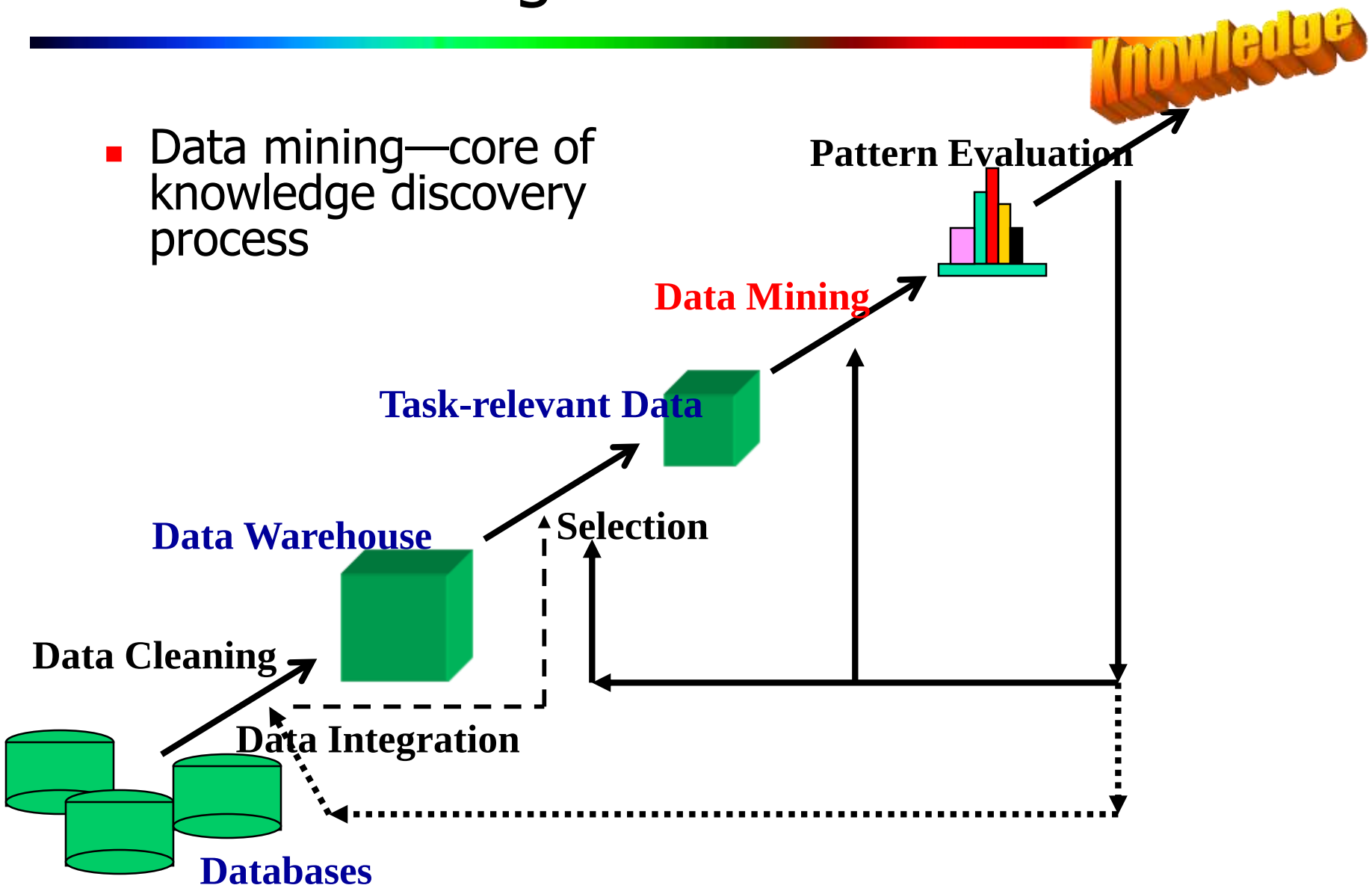
- Approaches: Clustering & model construction for frauds, outlier analysis
- Applications: Health care, retail, credit card service, telecomm.
  - Auto insurance: ring of collisions
  - Money laundering: suspicious monetary transactions
  - Medical insurance
    - Professional patients, ring of doctors, 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

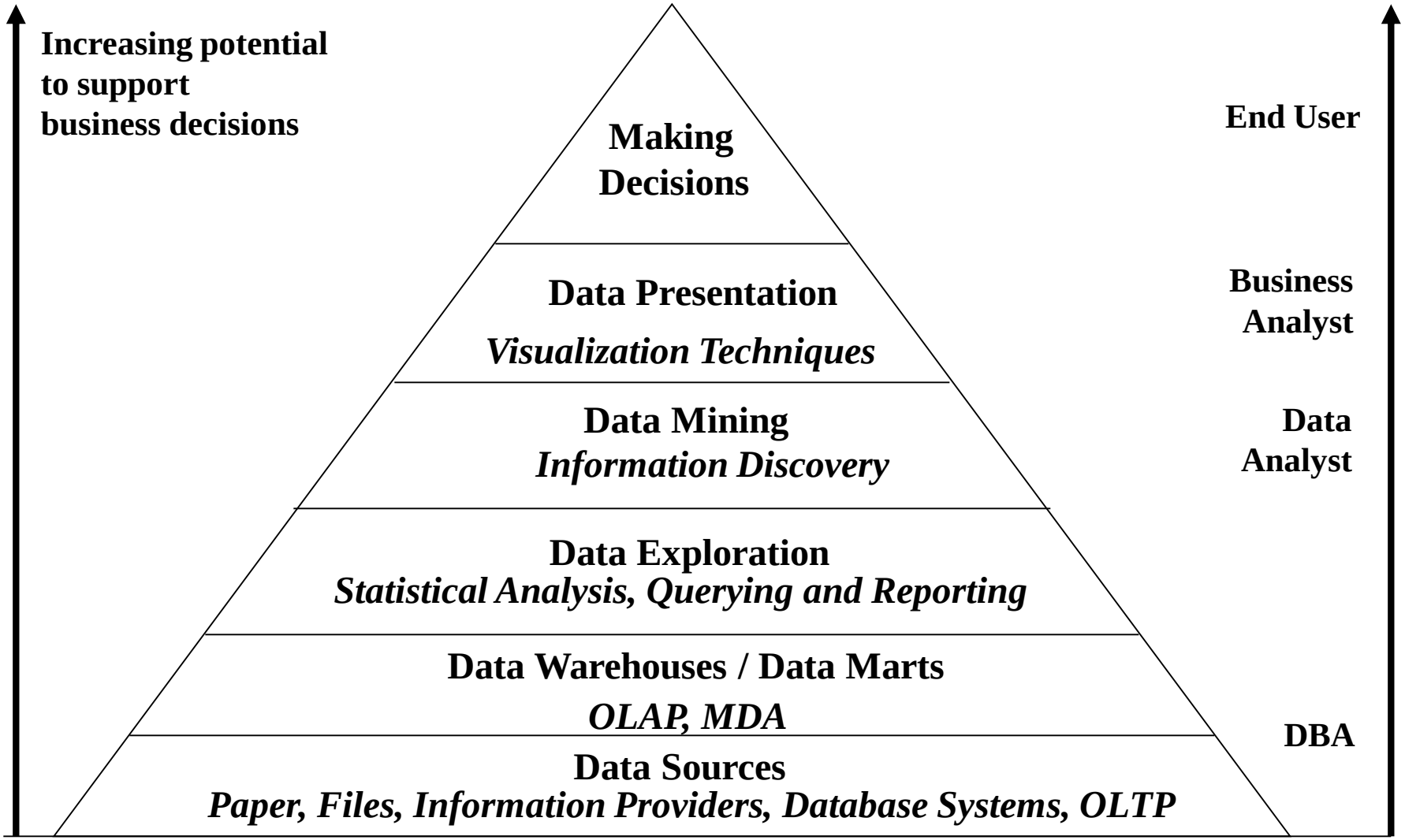
- Data mining—core of knowledge discovery process



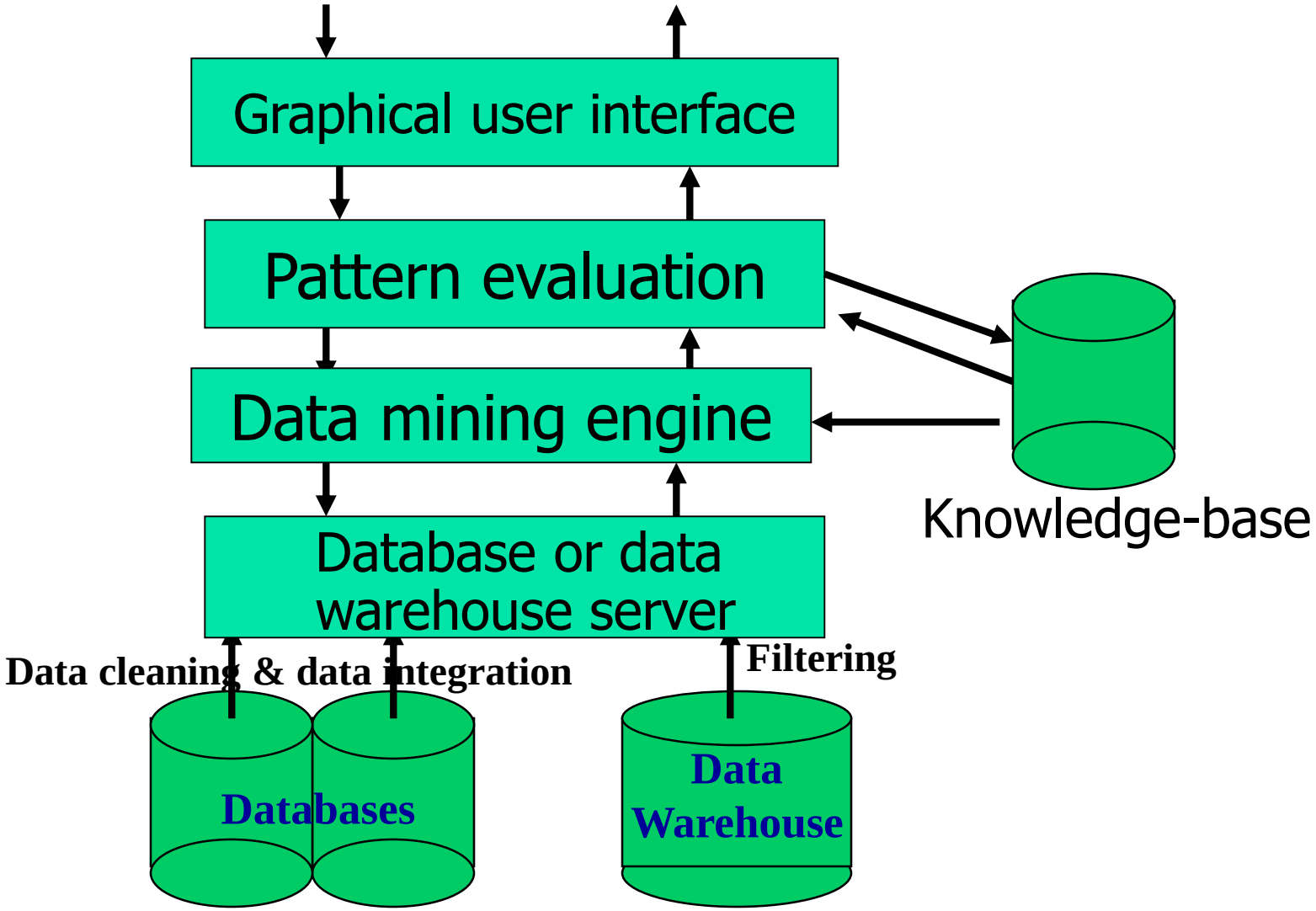
Steps of a KDD Process

- 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



Architecture: Typical Data Mining System



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

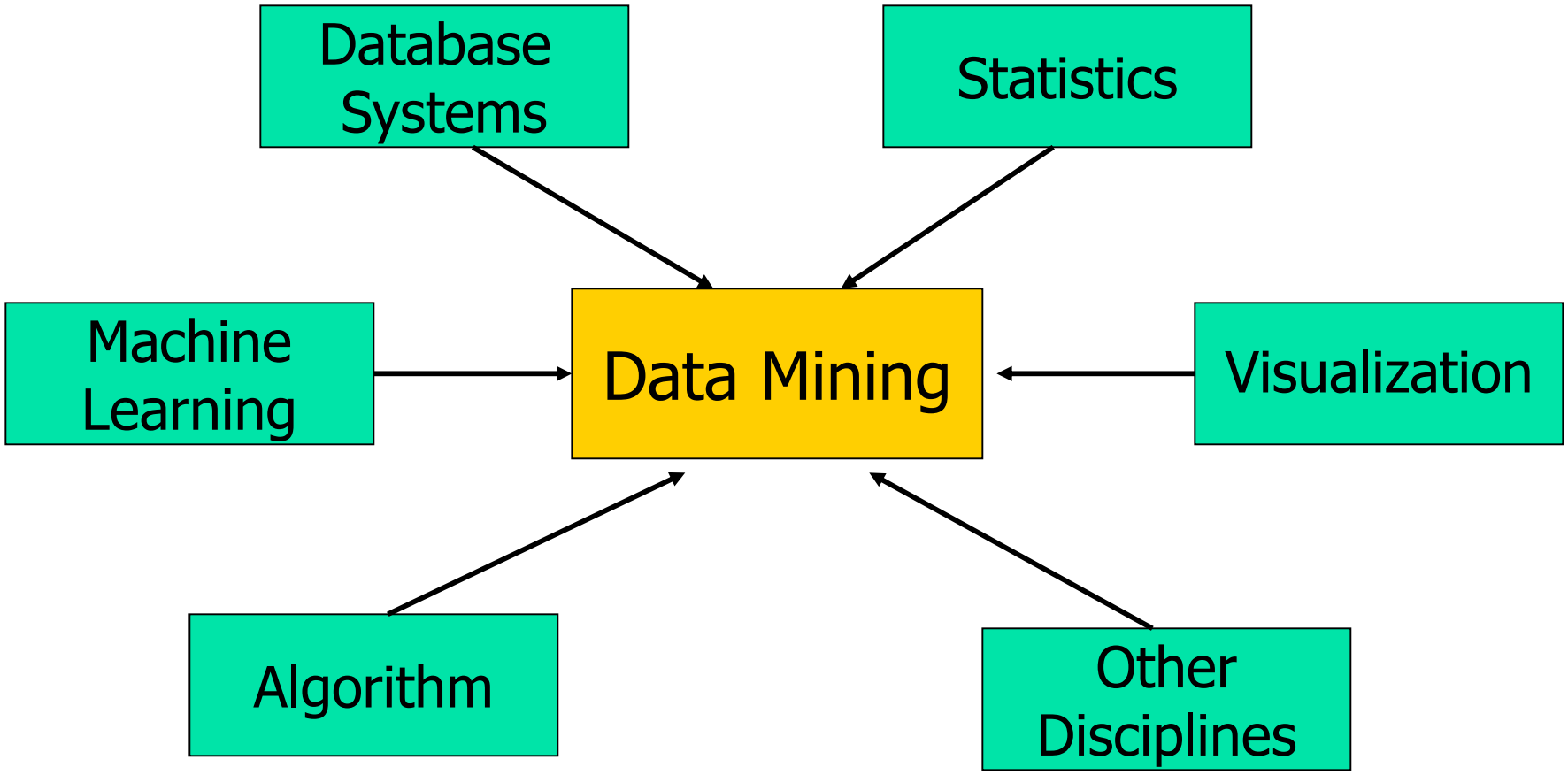
Are All the "Discovered" Patterns Interesting?

- 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

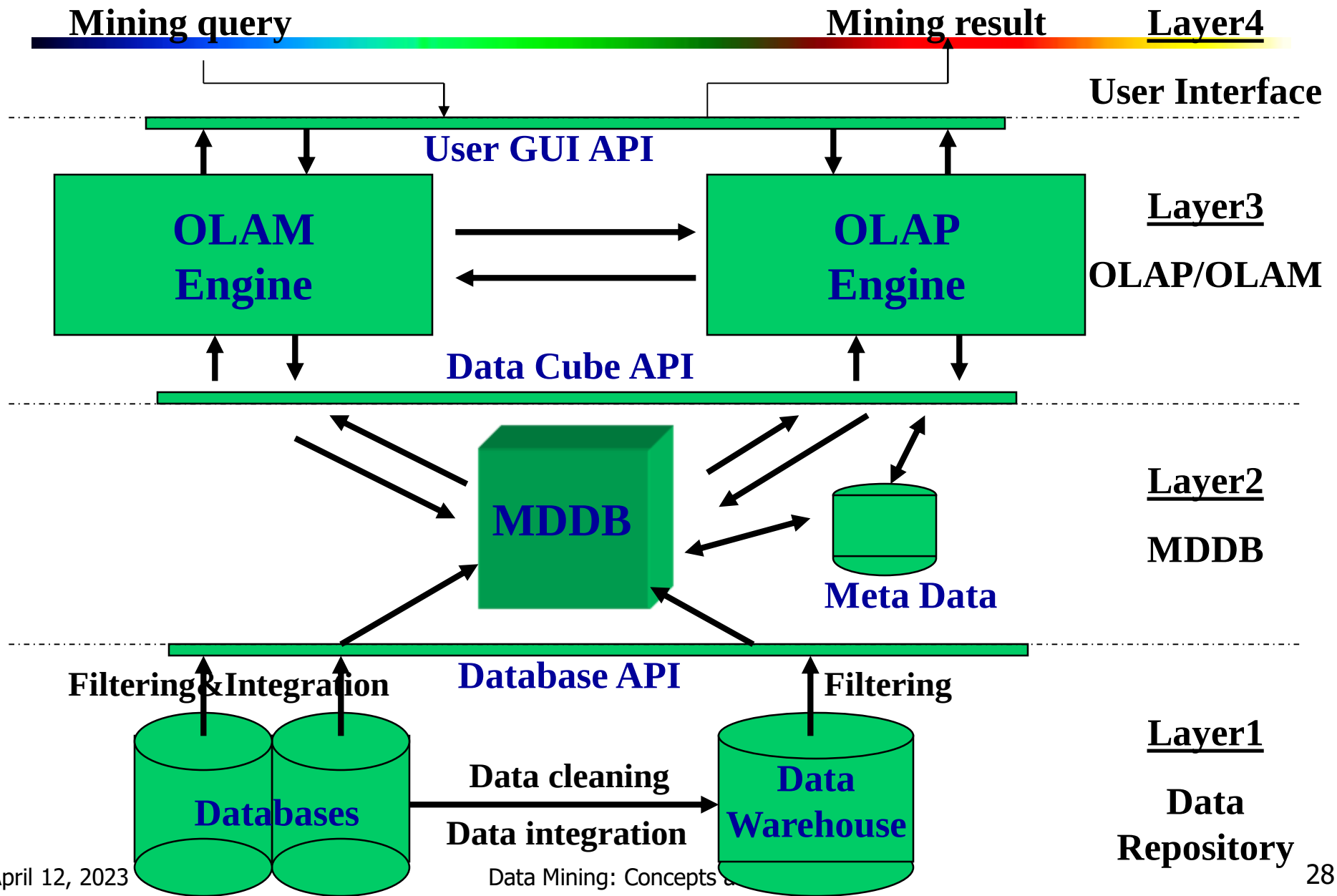
Multi-Dimensional View of Data Mining

- **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

www.cs.uiuc.edu/~hanj

Thank you !!!

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

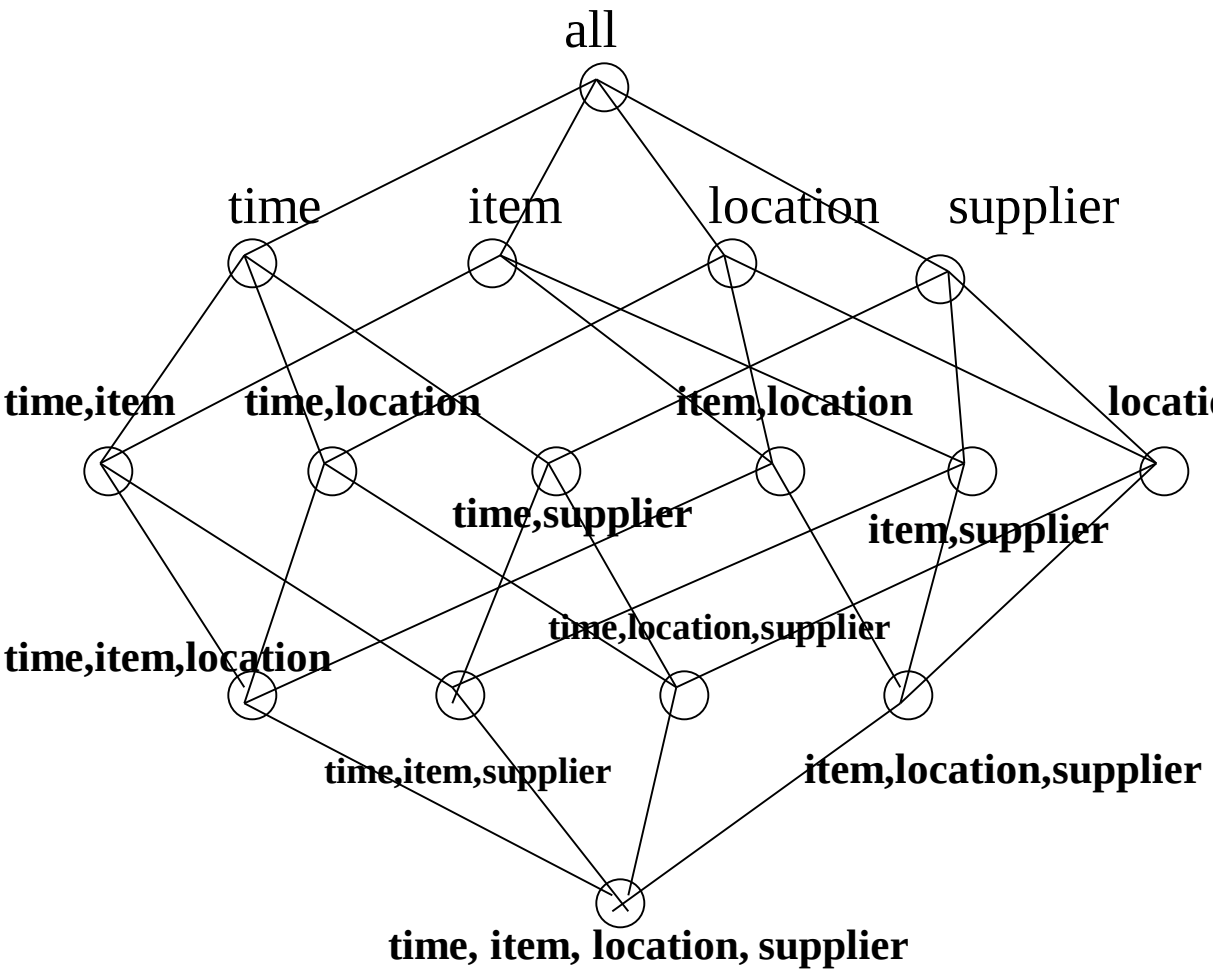
Chapter 2: Data Warehousing and OLAP Technology for Data Mining

- 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



0-D(apex) cuboid

1-D cuboids

2-D cuboids

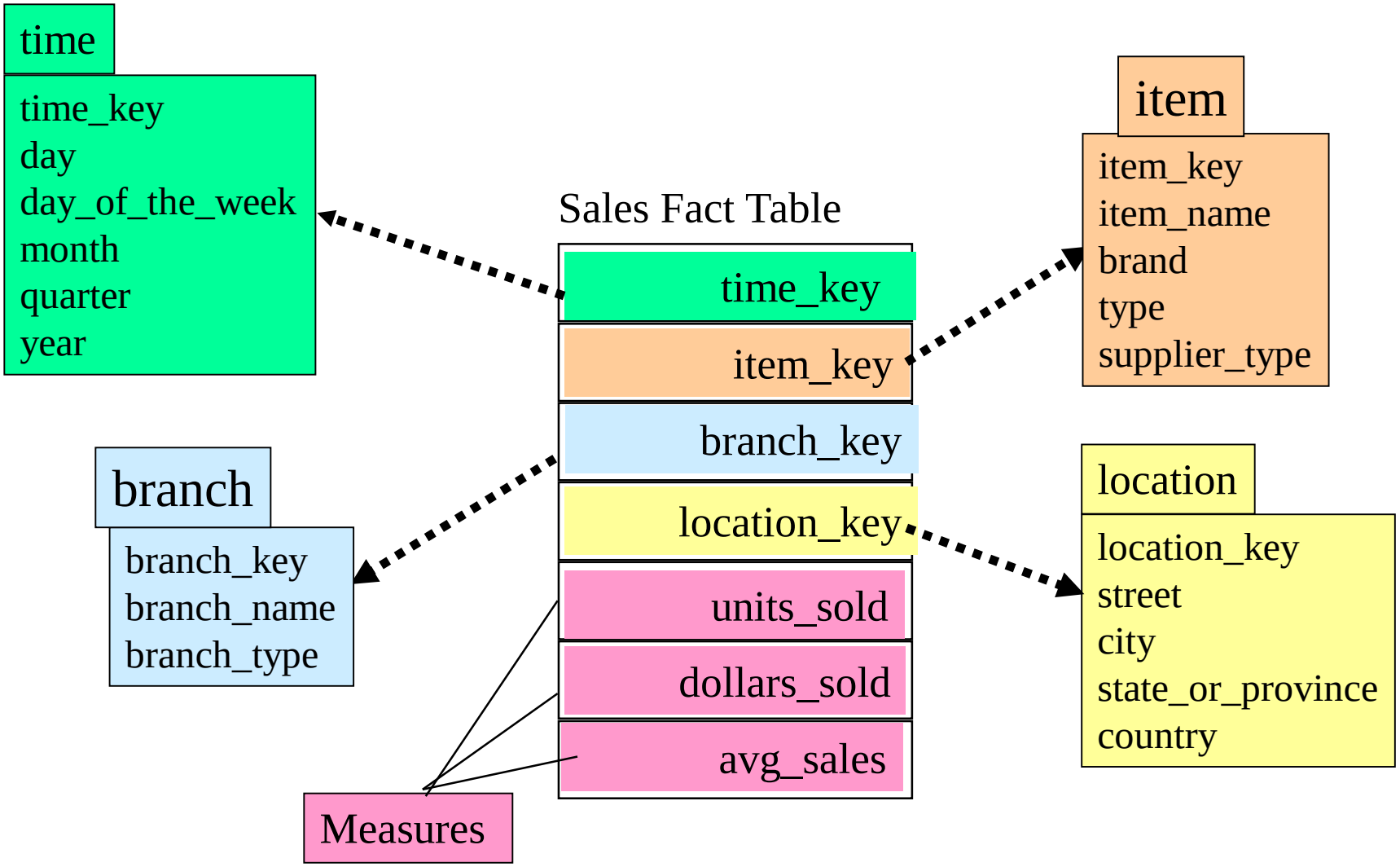
3-D cuboids

4-D(base) cuboid

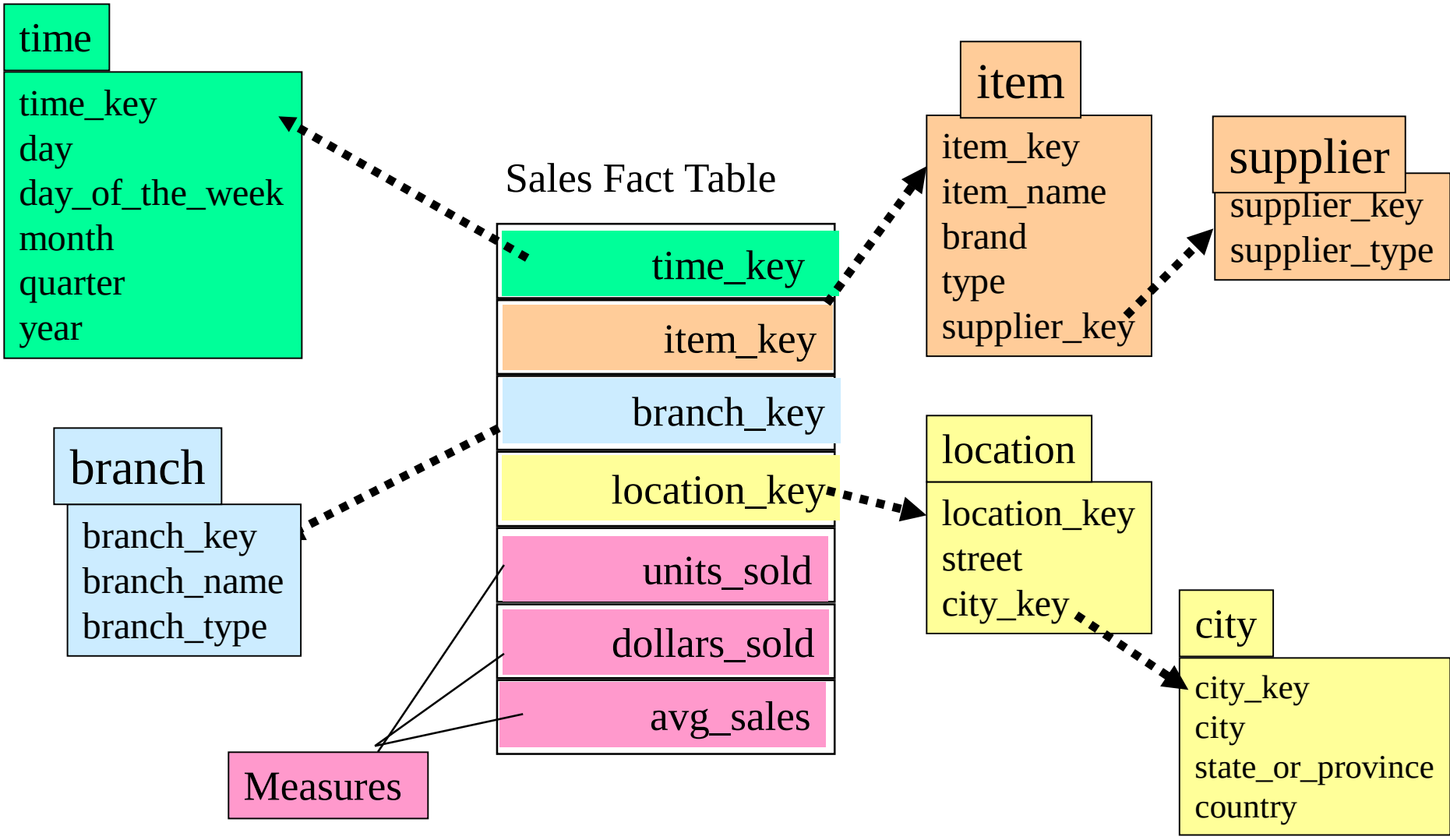
Conceptual Modeling of Data Warehouses

- 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

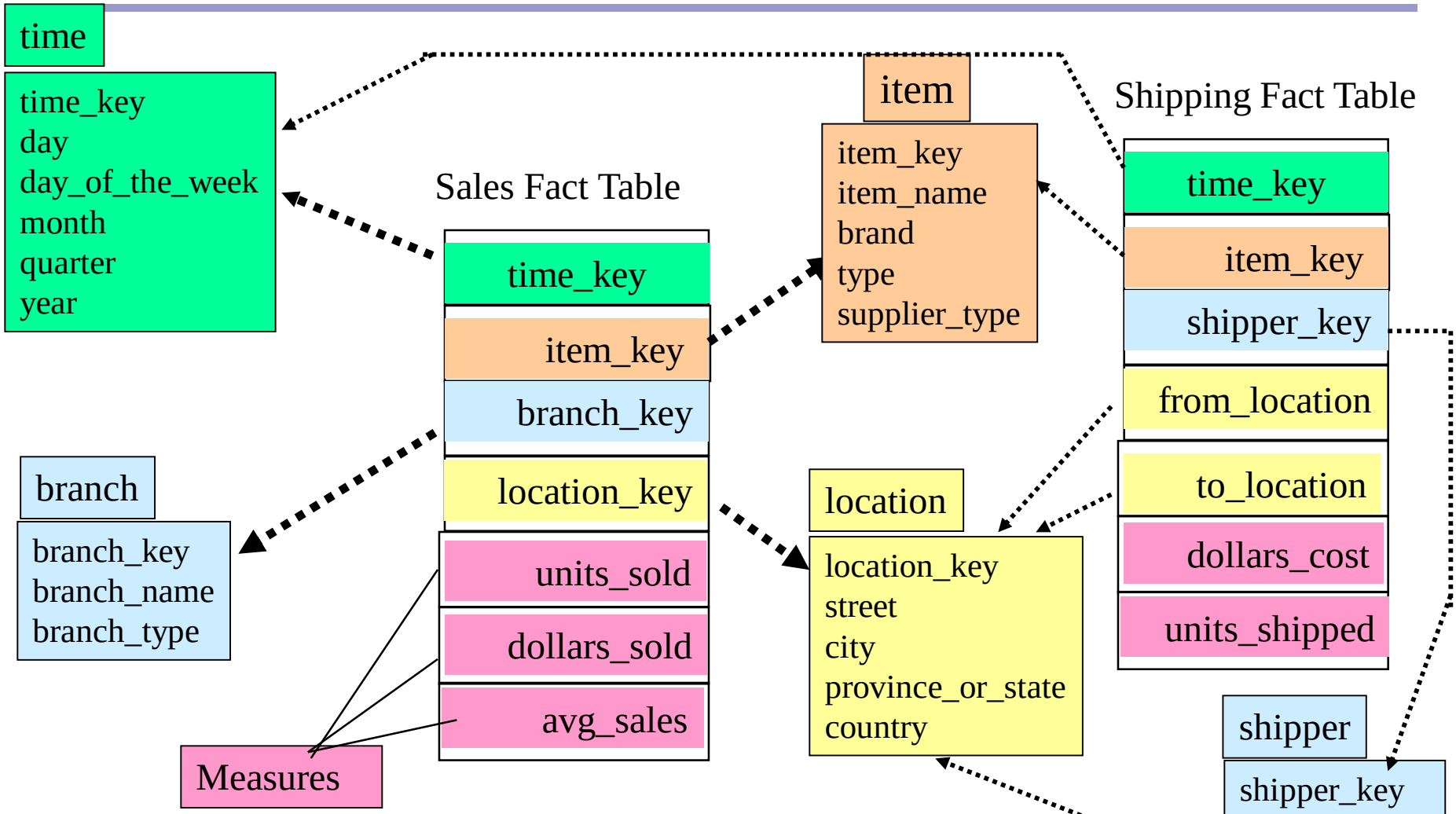
Example of Star Schema



Example of Snowflake Schema



Example of Fact Constellation



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>

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)
```

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))

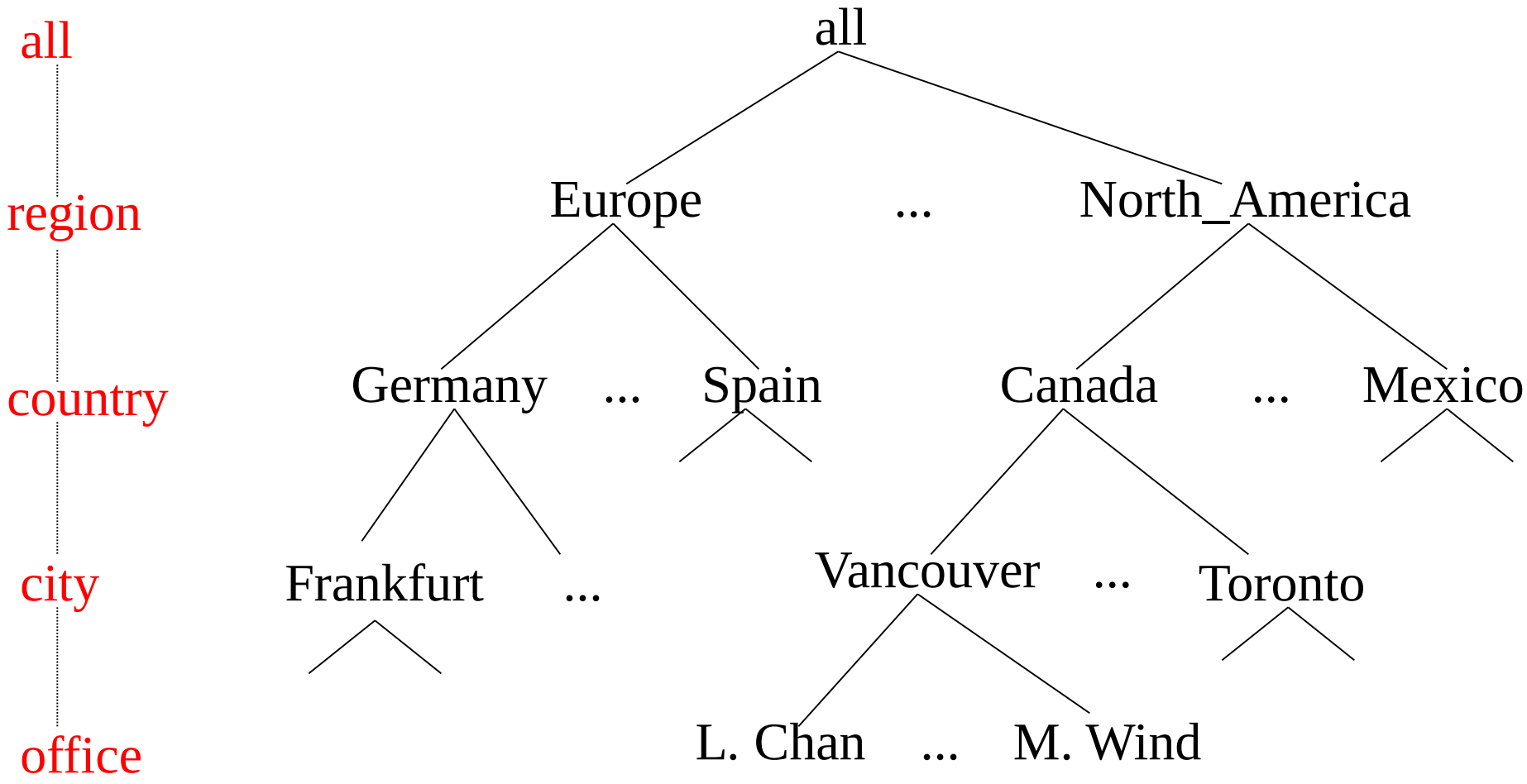
Defining a Fact Constellation in DMQL

```
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
```

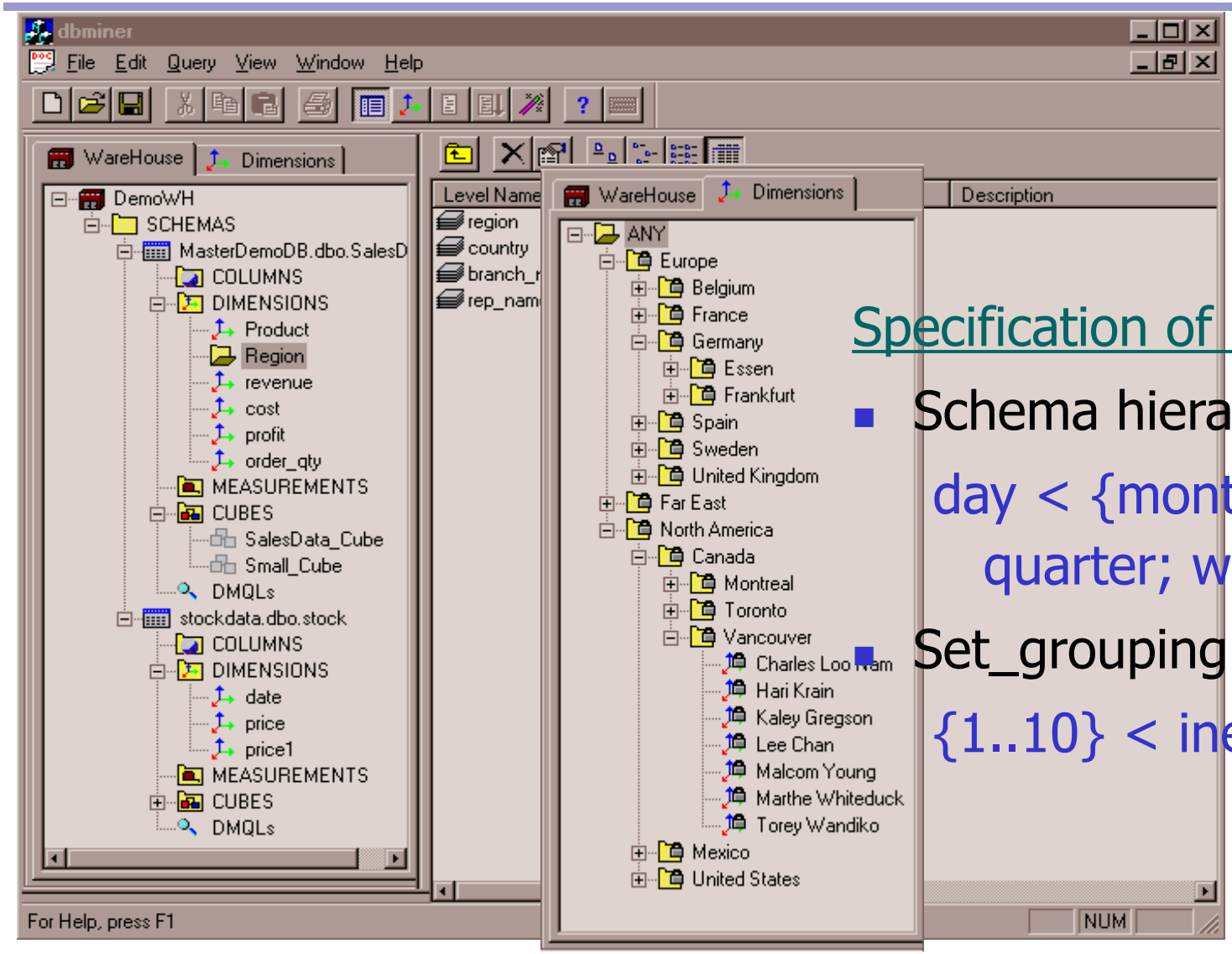
Measures: Three Categories

- 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

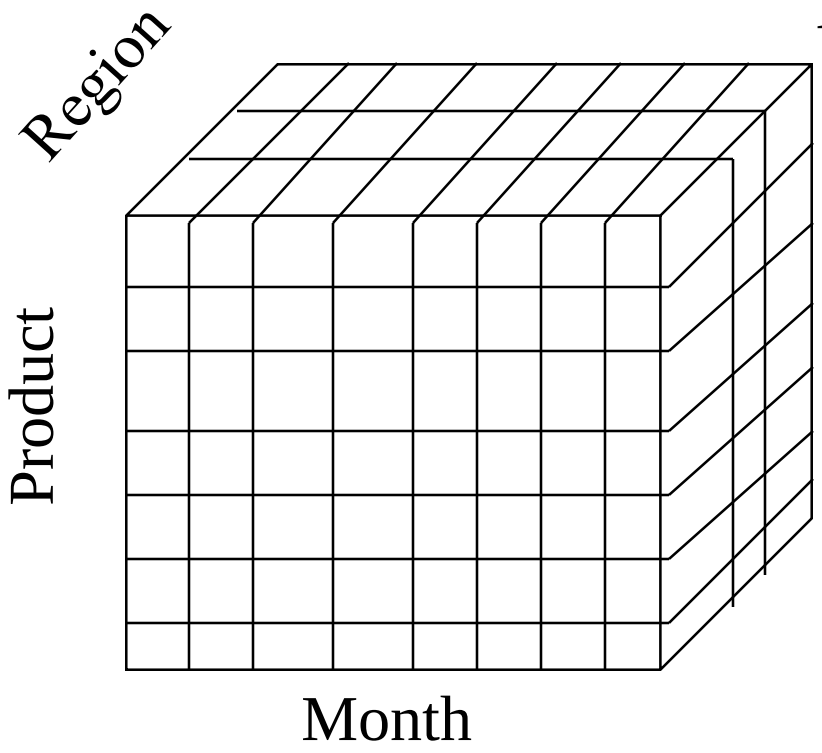
  day < {month < quarter; week} < year

- Set_grouping hierarchy

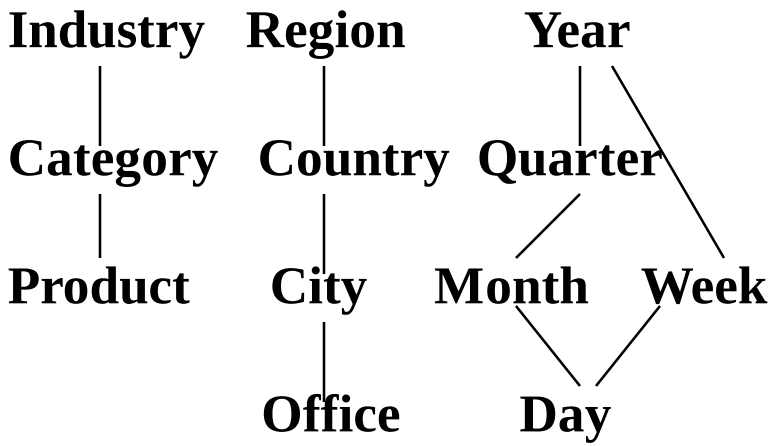
  {1..10} < inexpensive

Multidimensional Data

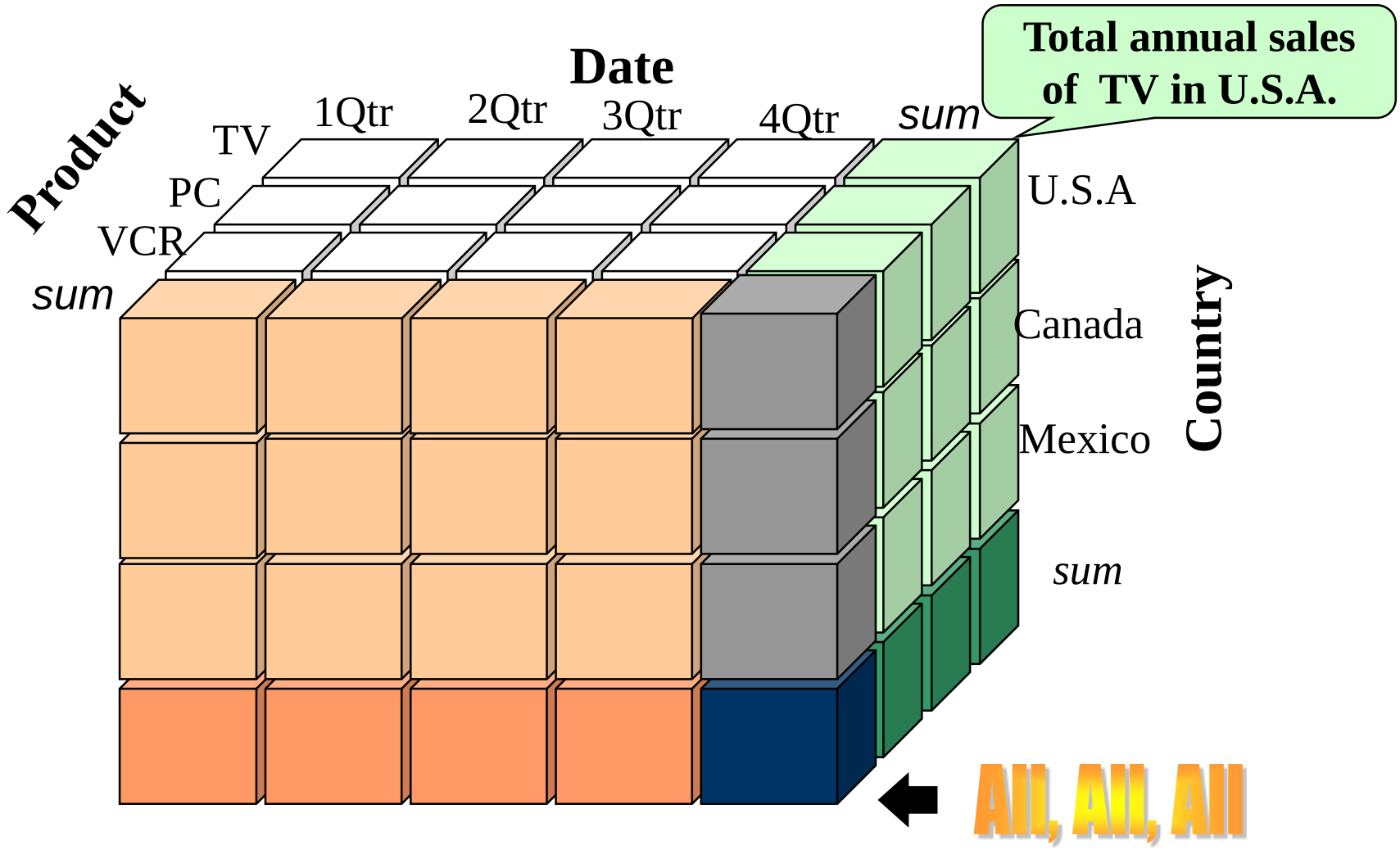
- Sales volume as a function of product, month, and region



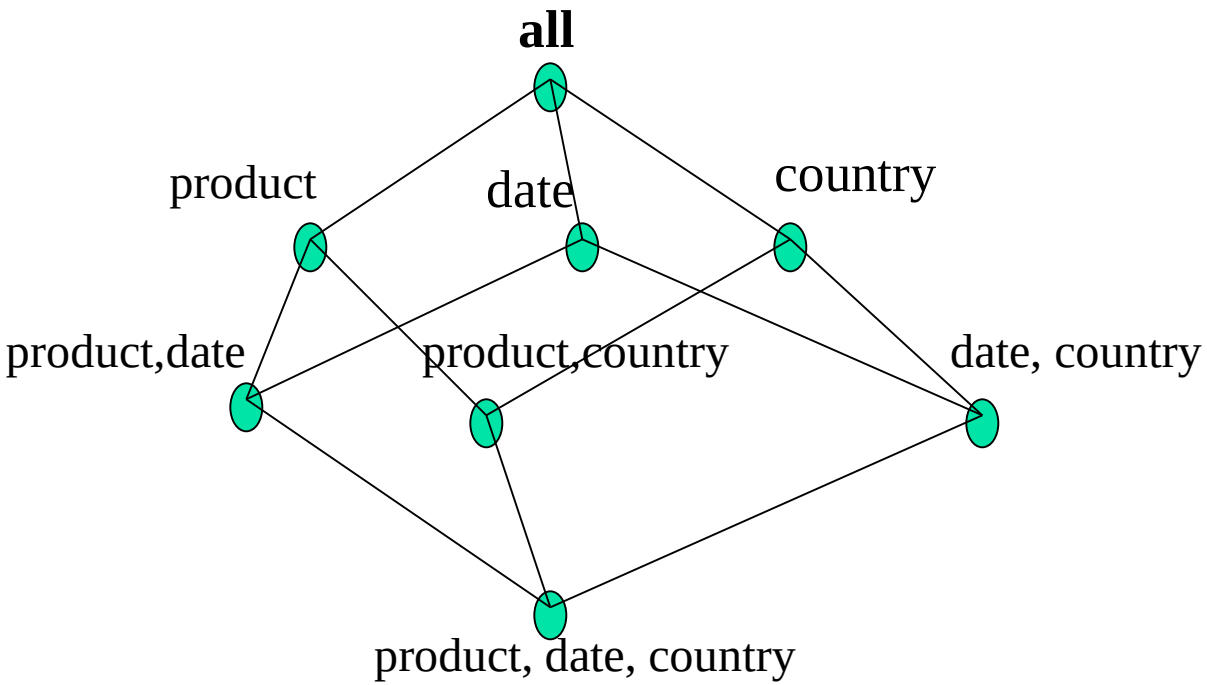
Dimensions: Product, Location, Time
Hierarchical summarization paths



A Sample Data Cube



Cuboids Corresponding to the Cube



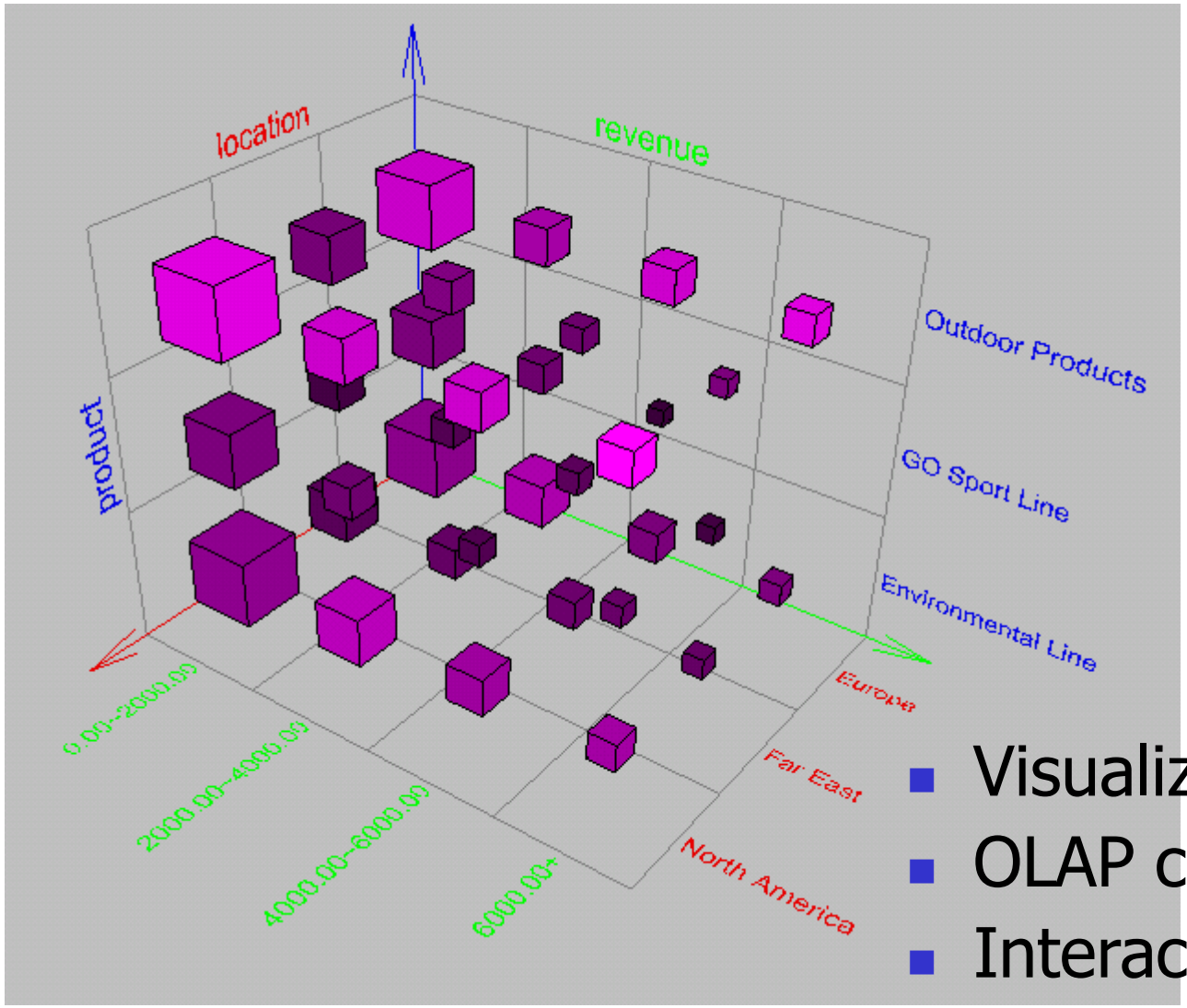
0-D(apex) cuboid

1-D cuboids

2-D cuboids

3-D(base) cuboid

Browsing a Data Cube

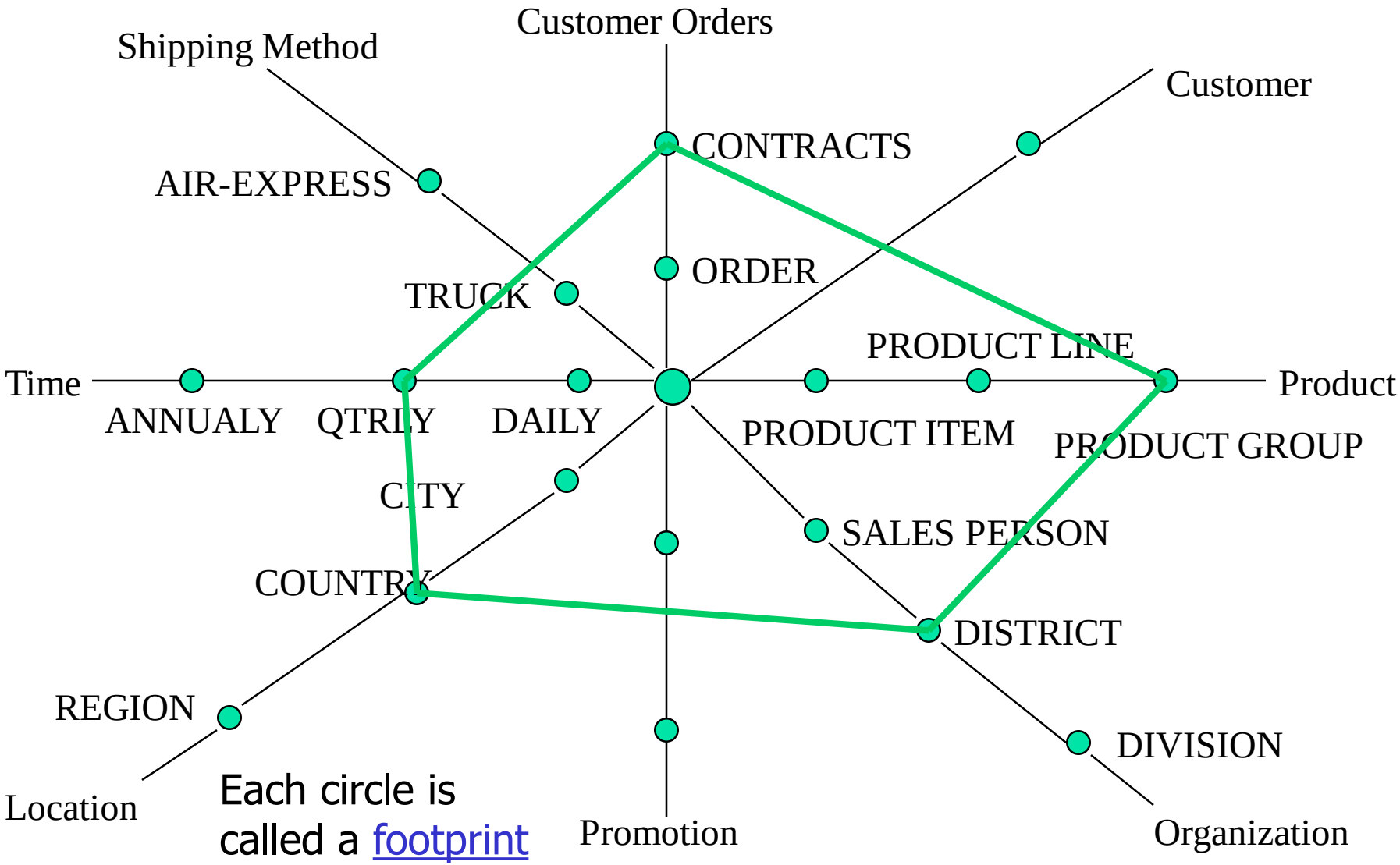


- Visualization
- OLAP capabilities
- Interactive manipulation

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

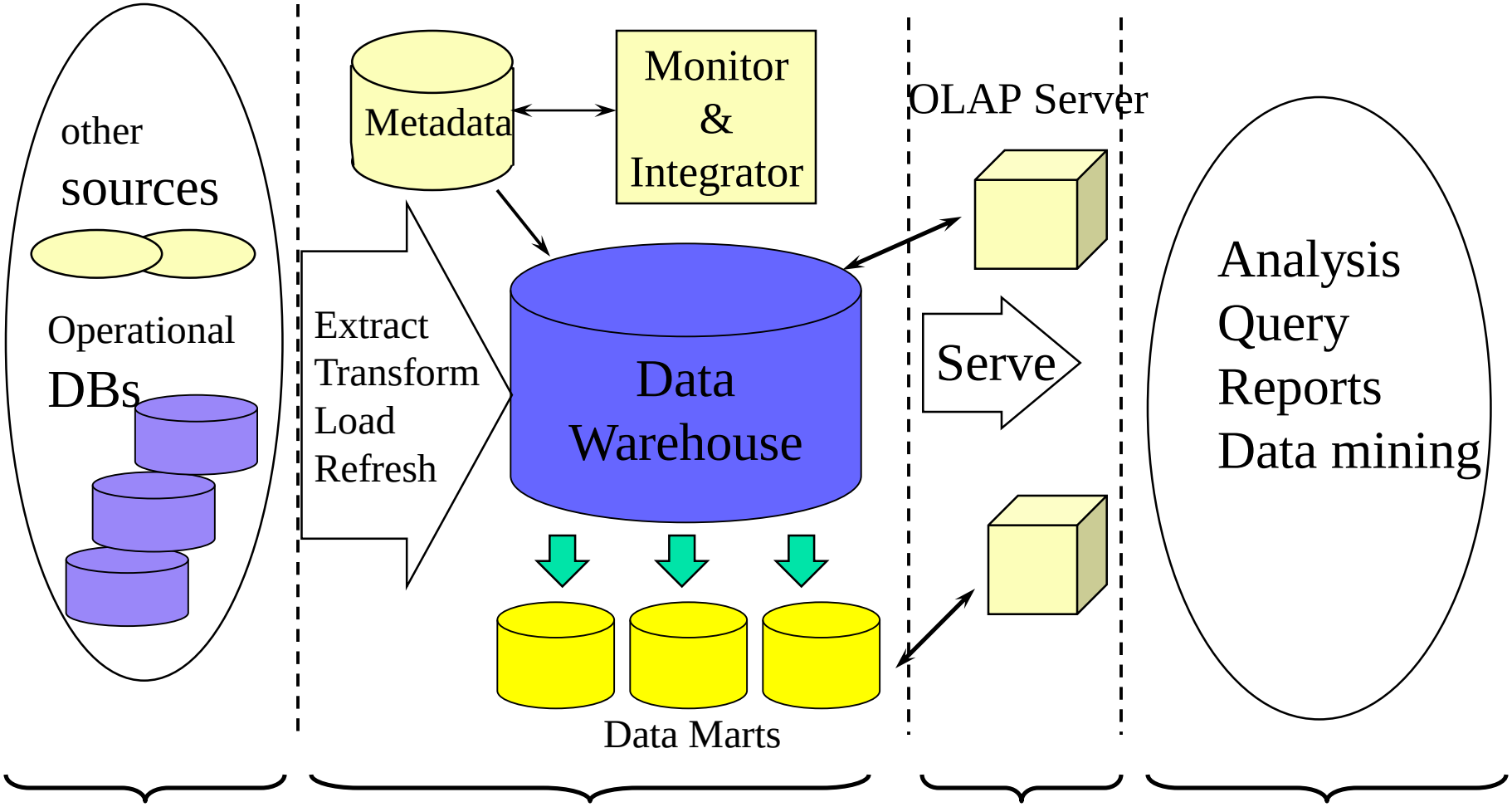
Design of a Data Warehouse: A Business Analysis Framework

- 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

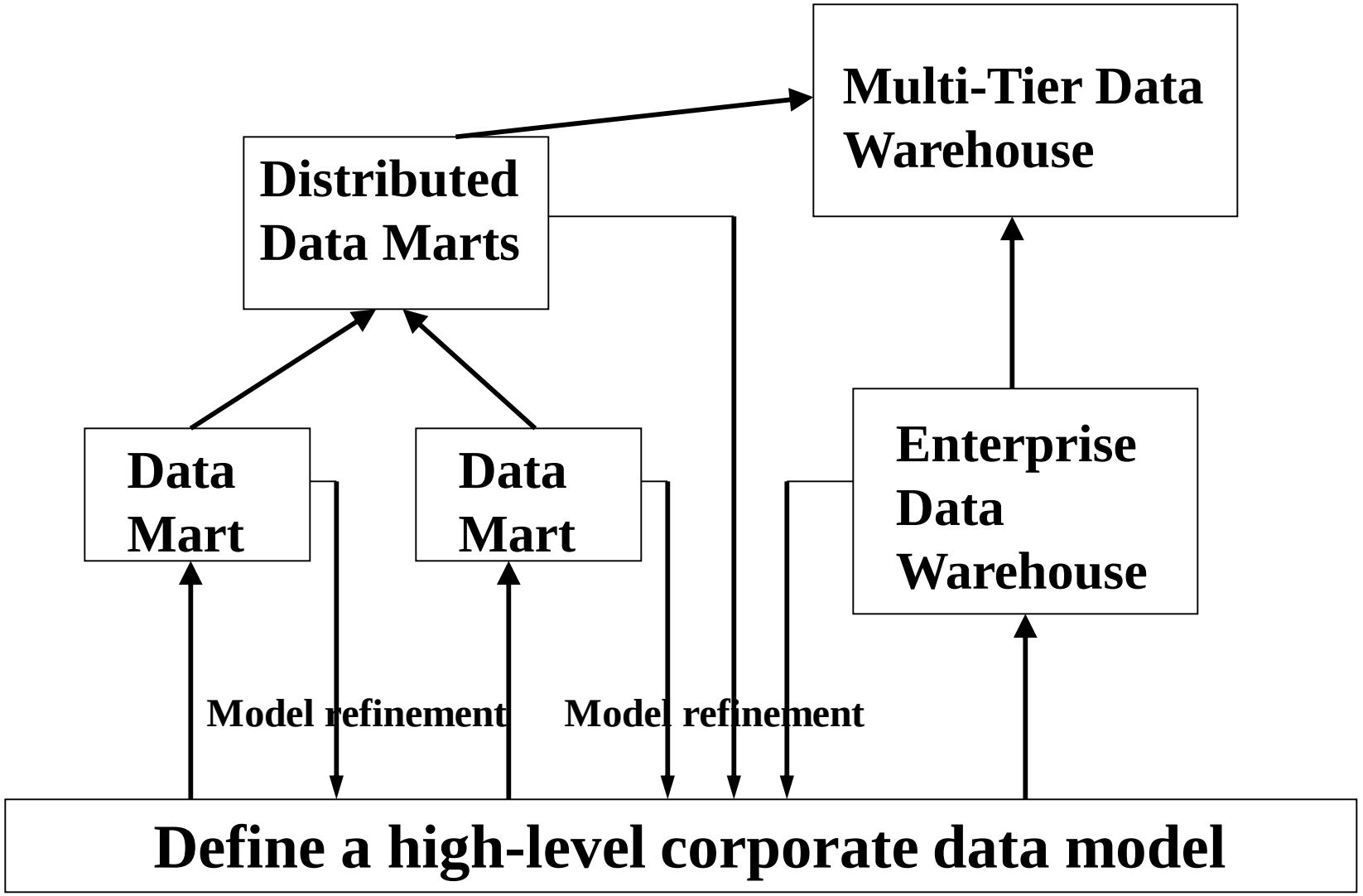
Multi-Tiered Architecture



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

Data Warehouse Development: A Recommended Approach



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 every (cuboid) (full materialization), none (no materialization), or some (partial materialization)
  - 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)

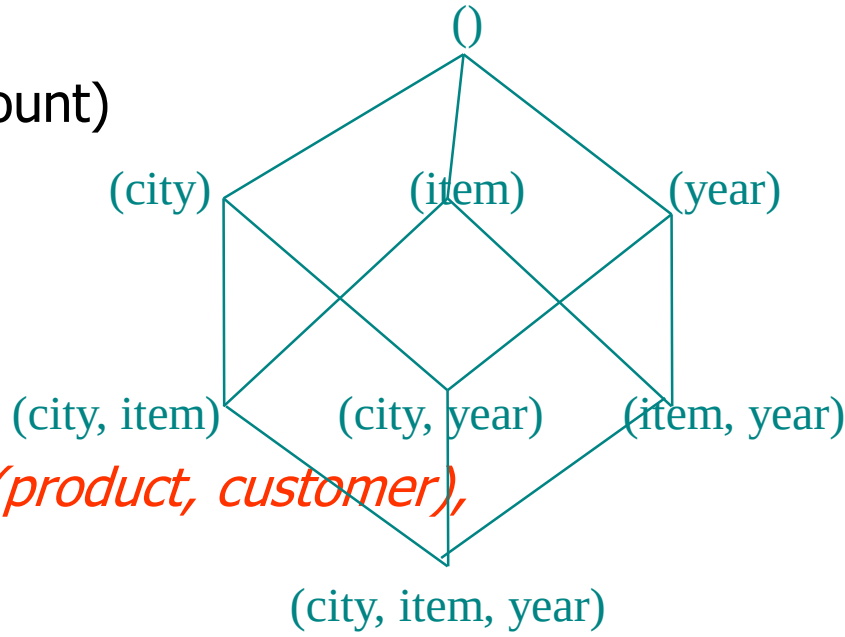
  SELECT item, city, year, SUM (amount)

  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

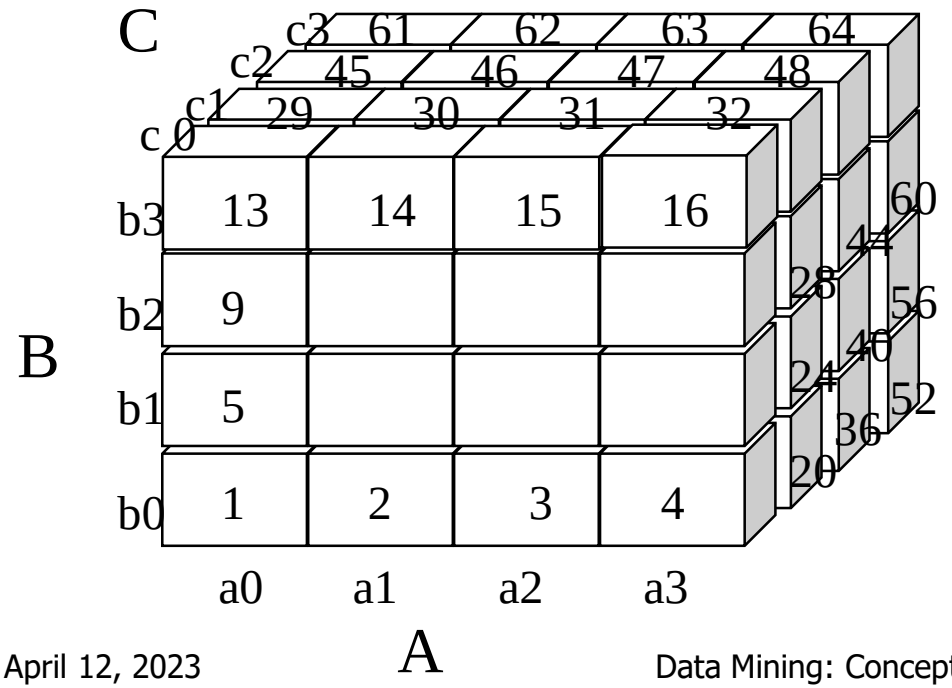
- Efficient cube computation methods
  - ROLAP-based cubing algorithms (Agarwal et al'96)
  - Array-based cubing algorithm (Zhao et al'97)
  - Bottom-up computation method (Beyer & Ramarkrishnan'99)
  - H-cubing technique (Han, Pei, Dong & Wang:SIGMOD'01)
- ROLAP-based cubing algorithms
  - Sorting, 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

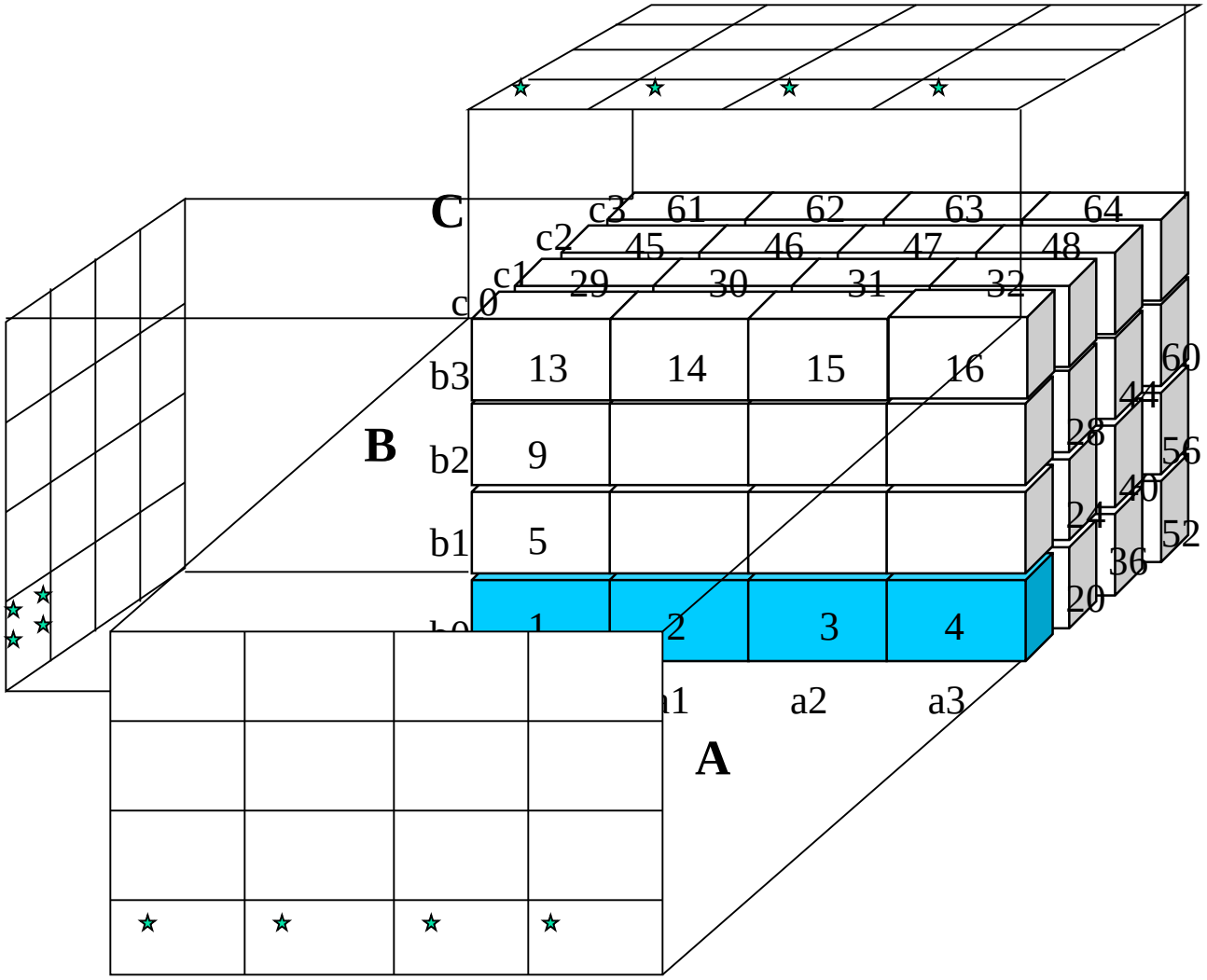
Multi-way Array Aggregation for Cube Computation

- 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.

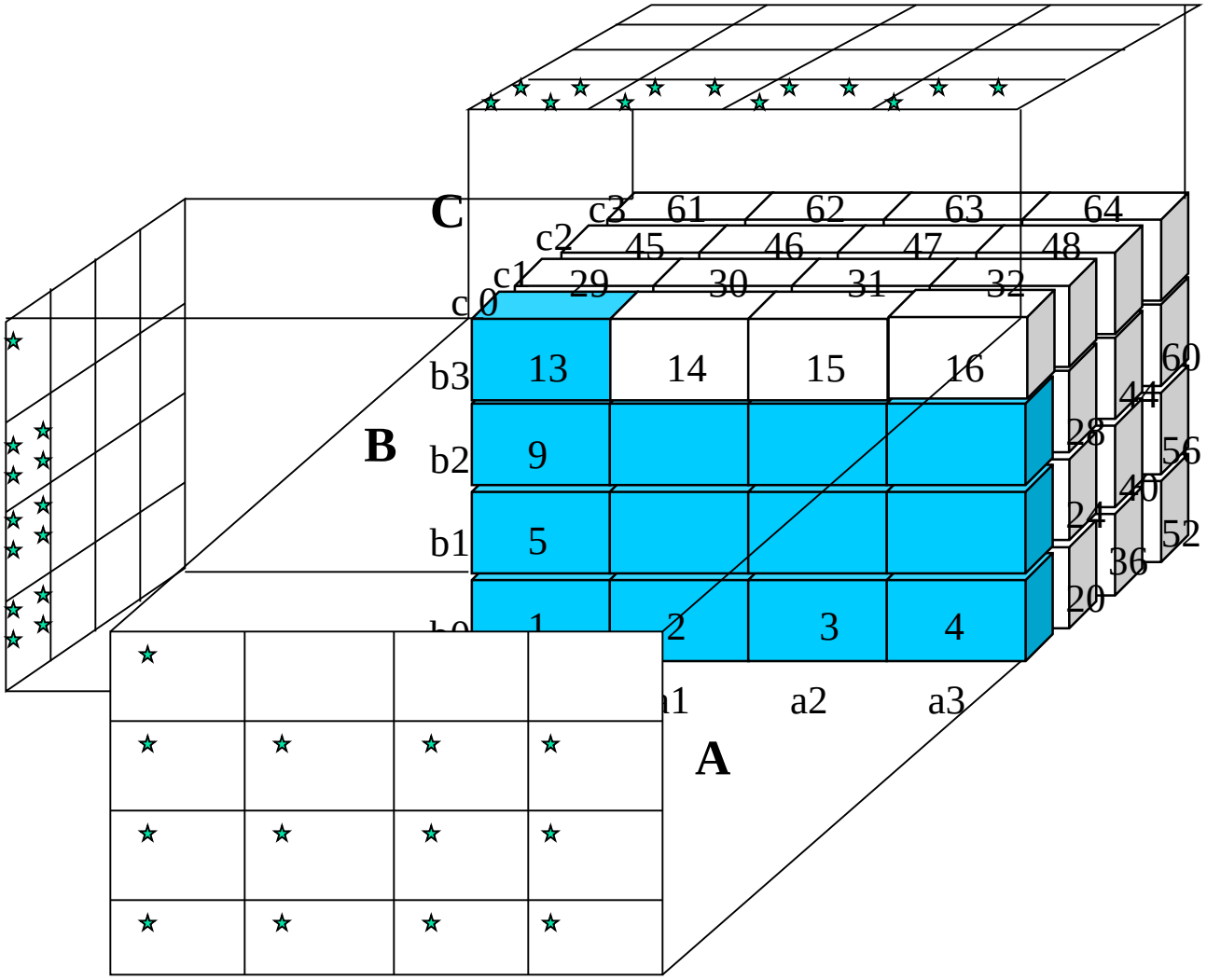


What is the best traversing order to do multi-way aggregation?

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

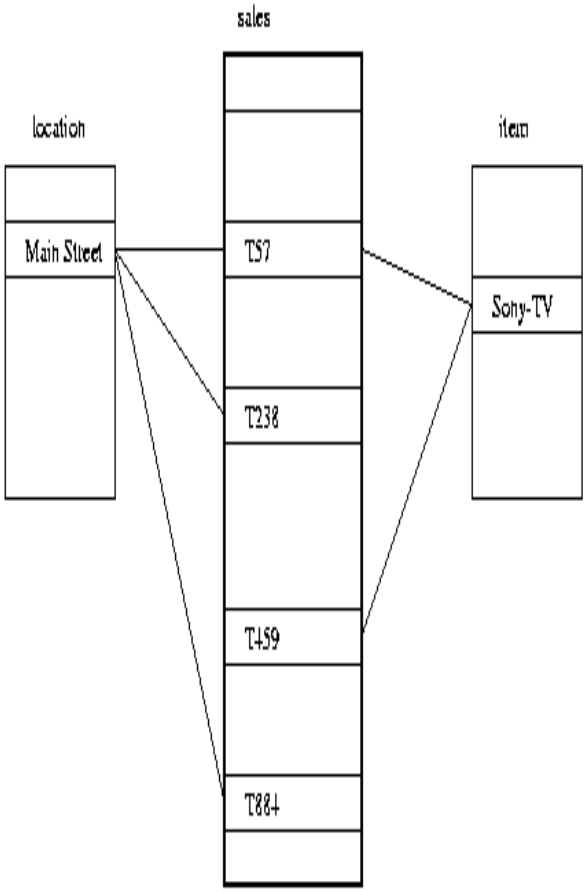
RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

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

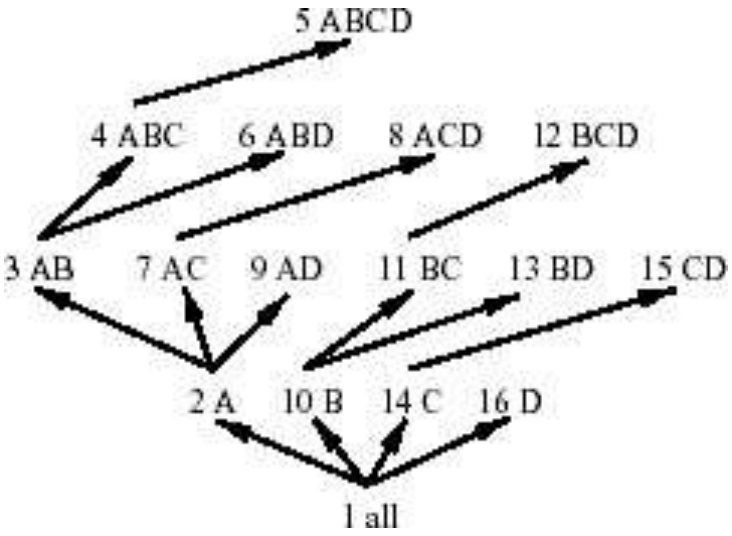
- Computing only the cuboid cells whose count or other aggregates satisfying the condition:

  HAVING COUNT(*) >= *minsup*

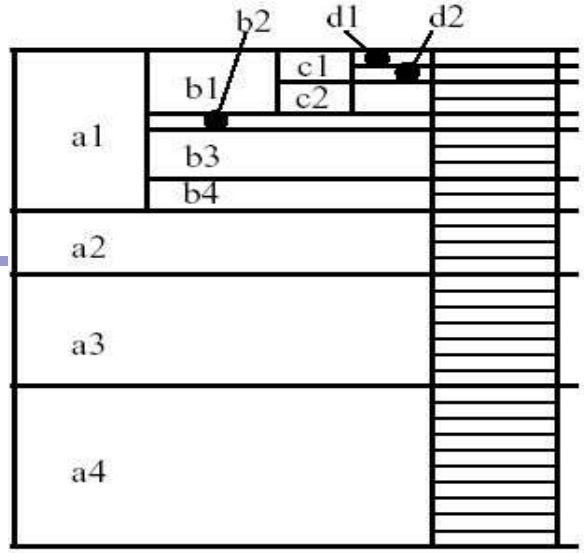
- Motivation
  - Only a small portion of cube cells may be "above the water" in a sparse cube
  - Only calculate "interesting" data—data above certain threshold
  - Suppose 100 dimensions, only 1 base cell. How many aggregate (non-base) cells if count >= 1? What about count >= 2?

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 $\Rightarrow$ 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 $avg^{50}()$=

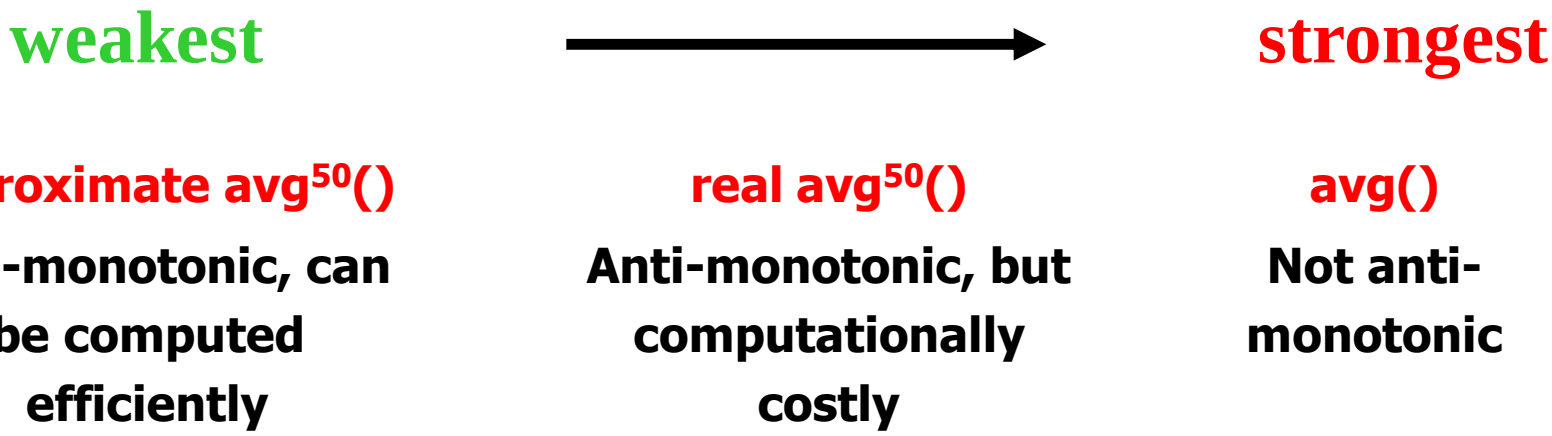
(28000+10600+600*15)/50=952

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



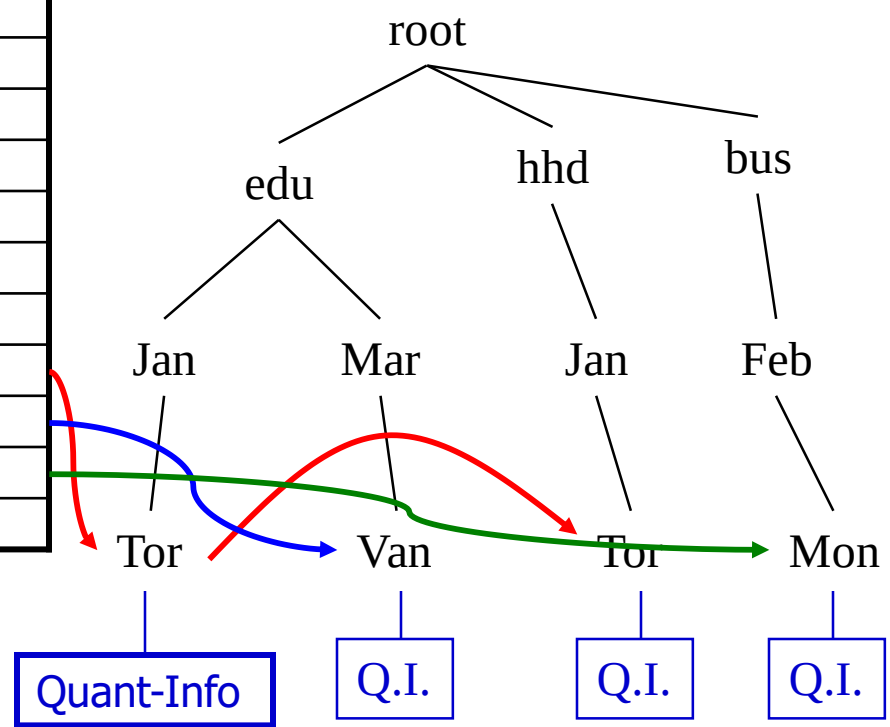
An Efficient Iceberg Cubing Method: Top-k H-Cubing

- 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

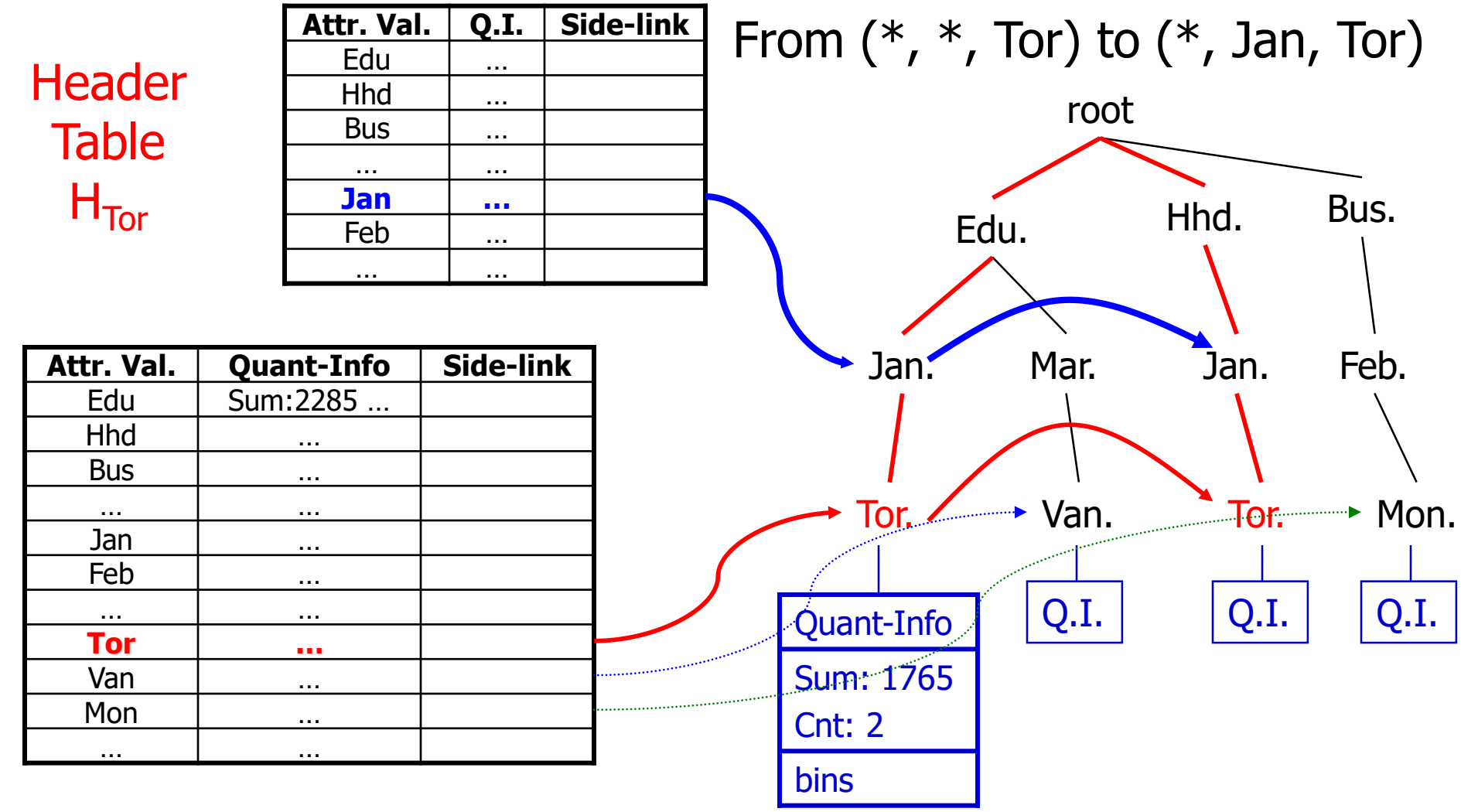
- Construction cost: a single database scan
- Completeness: It contains the complete information needed for computing the iceberg cube
- Compactness: # of nodes ☯ n*m+1
  - n: # of tuples in the table
  - m: # of attributes

Computing Cells Involving Dimension City

Header Table H_{Tor}

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

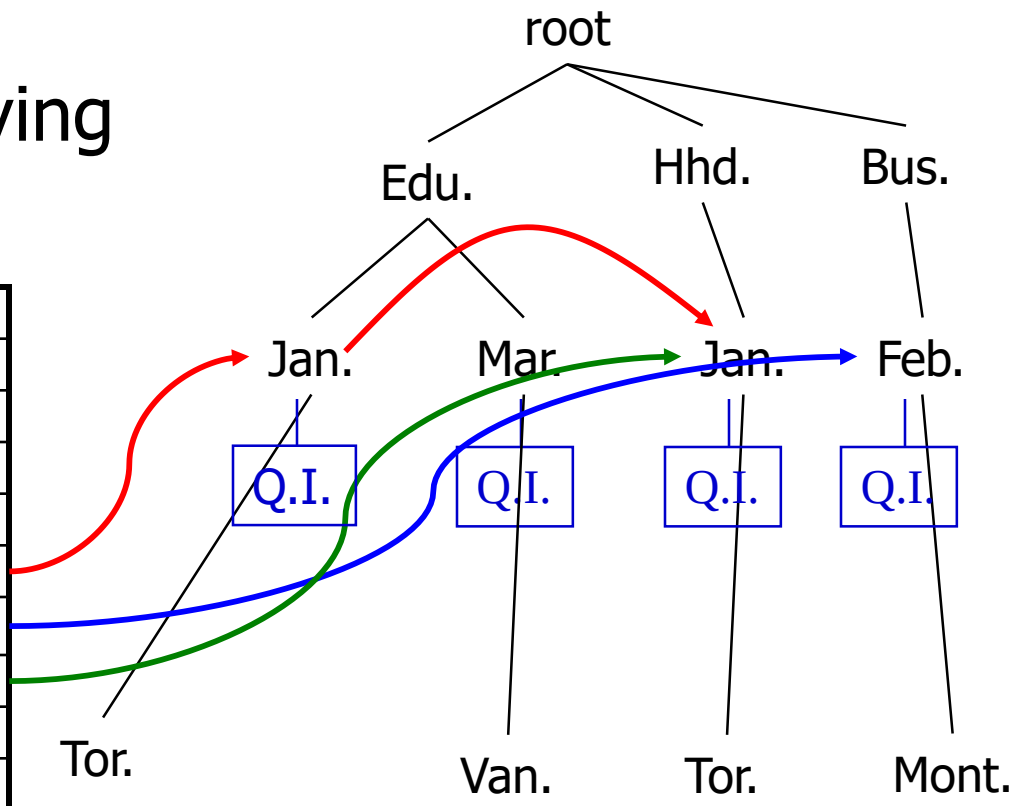
Attr. Val.	Quant-Info	Side-link
Edu	Sum:2285 …	
Hhd	…	
Bus	…	
…	…	
Jan	…	
Feb	…	
…	…	
Tor	**…**	
Van	…	
Mon	…	
…	…	



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.	…	
…	…	

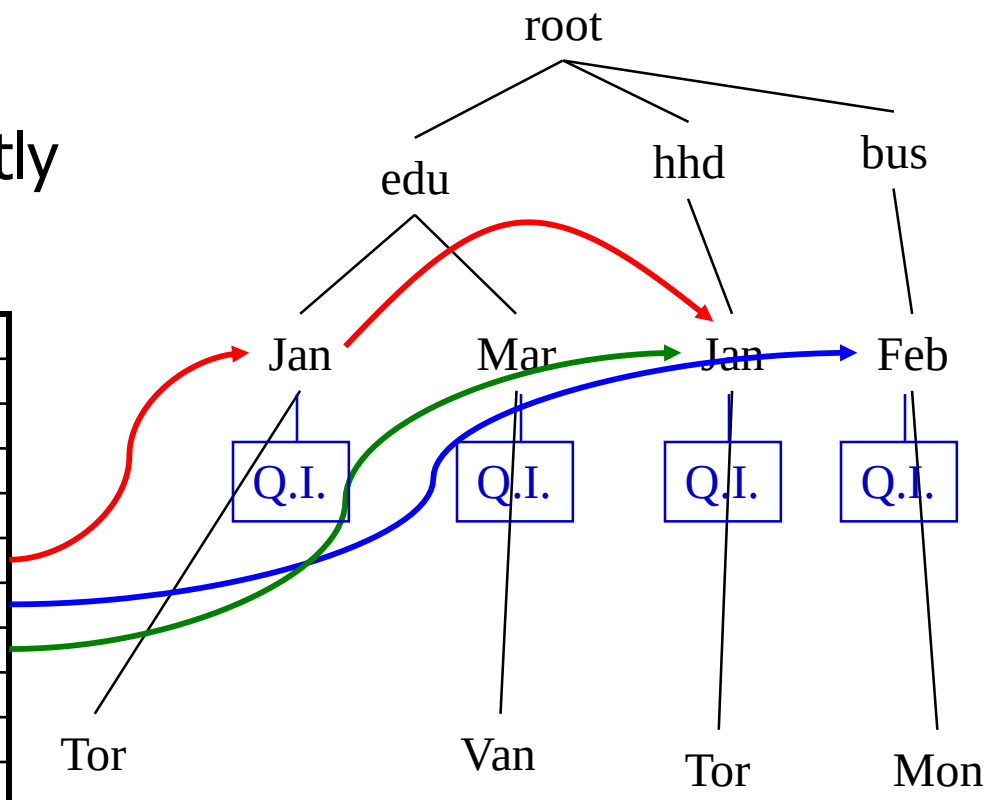


Top-k OK mark: if Q.I. in a child passes top-k avg threshold, so does its parents. No binning is needed!

Computing Cells Involving Only Cust_grp

Check header table directly

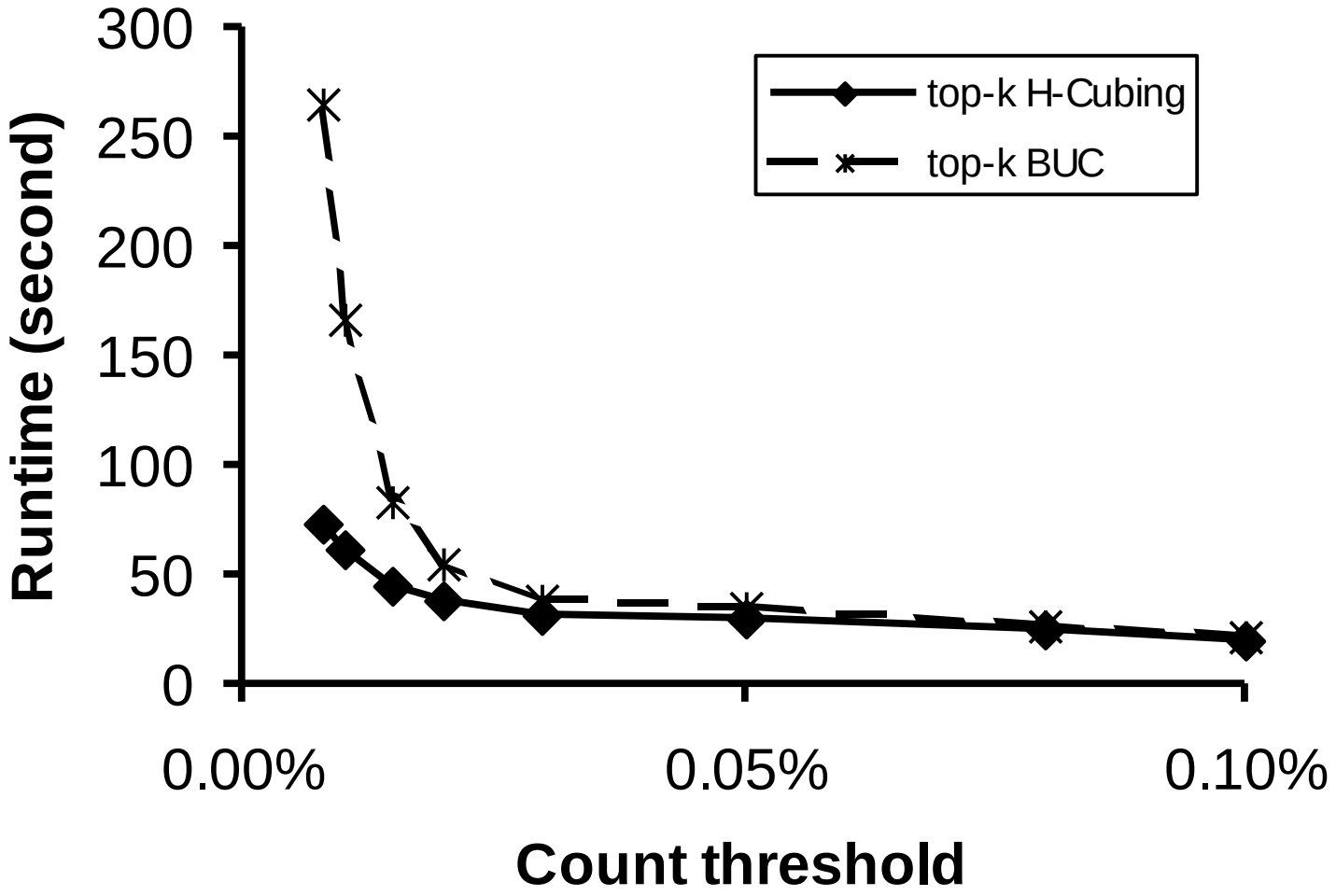
Attr. Val.	Quant-Info	Side-link
Edu	**Sum:2285 …**	
Hhd	**…**	
Bus	**…**	
…	…	
Jan	…	
Feb	…	
Mar	…	
…	…	
Tor	…	
Van	…	
Mon	…	
…	…	



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() $\leq$ v: $avg_k(c) \leq v$ (bottom-k avg)
  - Avg() $\geq$ v only (no count): max(price) $\geq$ v
  - Sum(profit) (profit can be negative):
    - p_sum(c) $\geq$ v if p_count(c) $\geq$ k; or otherwise, $sum^k(c) \geq v$
  - 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.
- Icerberg 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

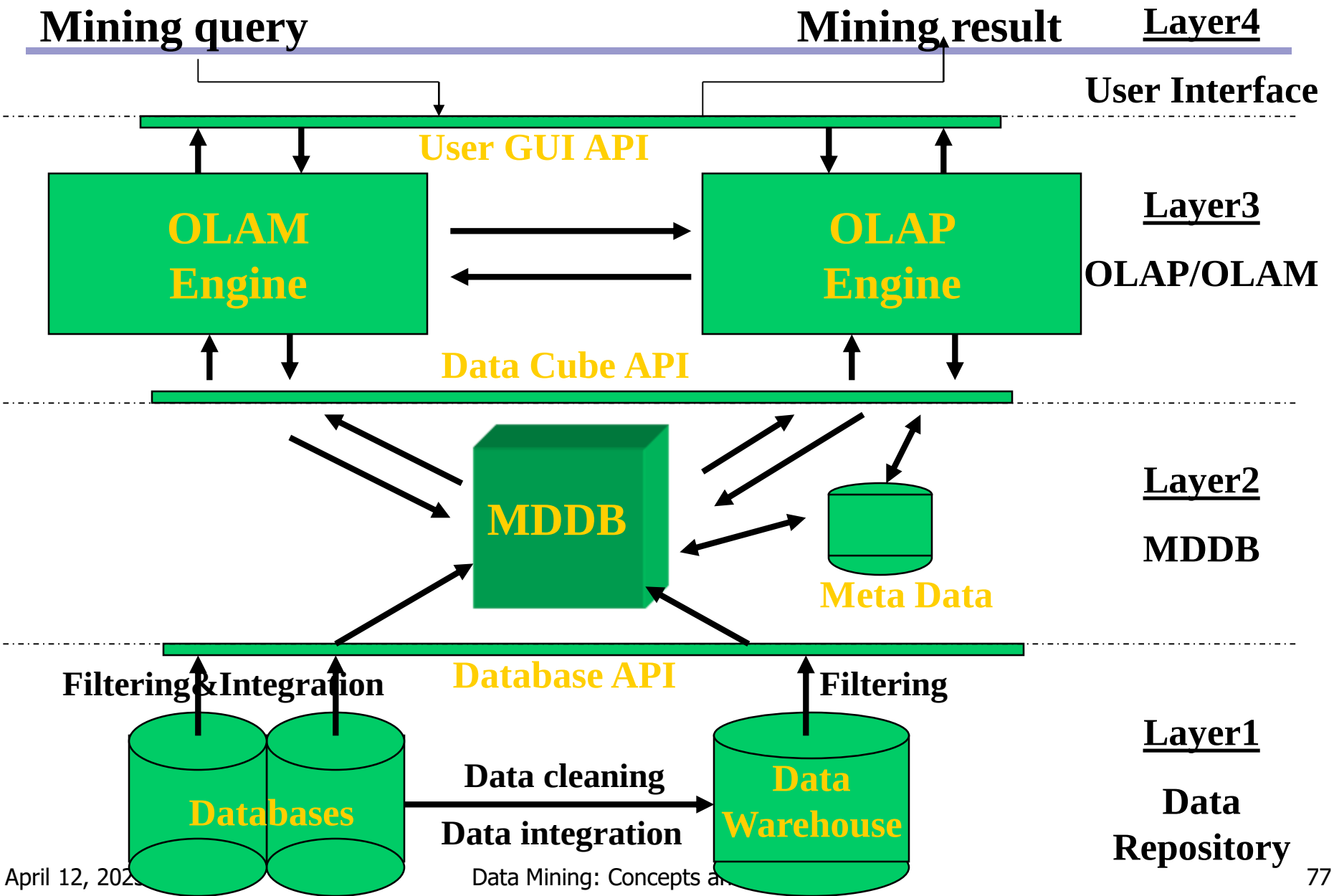
Data Warehouse Usage

- 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.

www.cs.uiuc.edu/~hanj

Thank you !!!

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

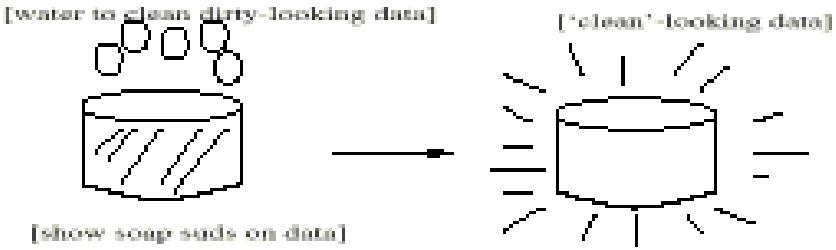
- A well-accepted multidimensional view:
  - Accuracy
  - Completeness
  - Consistency
  - Timeliness
  - Believability
  - Value added
  - Interpretability
  - Accessibility
- Broad categories:
  - intrinsic, contextual, representational, and accessibility.

Major Tasks in Data Preprocessing

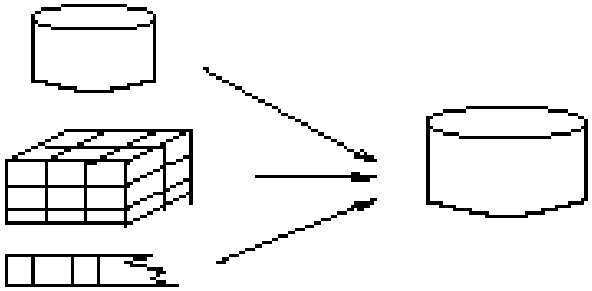
- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
  - Integration of multiple databases, data cubes, or files
- Data transformation
  - Normalization and aggregation
- Data reduction
  - Obtains reduced representation 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

Data Cleaning



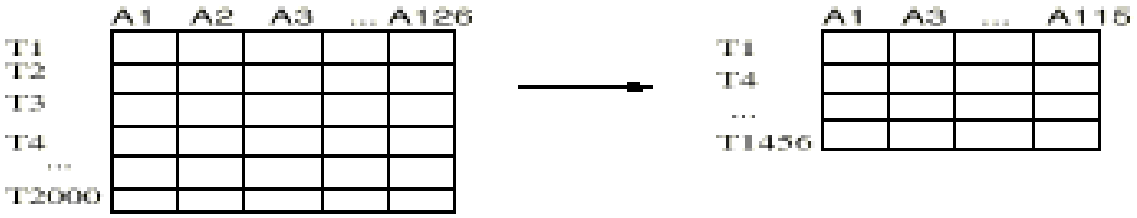
Data Integration



Data Transformation

-2, 32, 100, 59, 48 → -0.02, 0.32, 1.00, 0.59, 0.48

Data Reduction



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

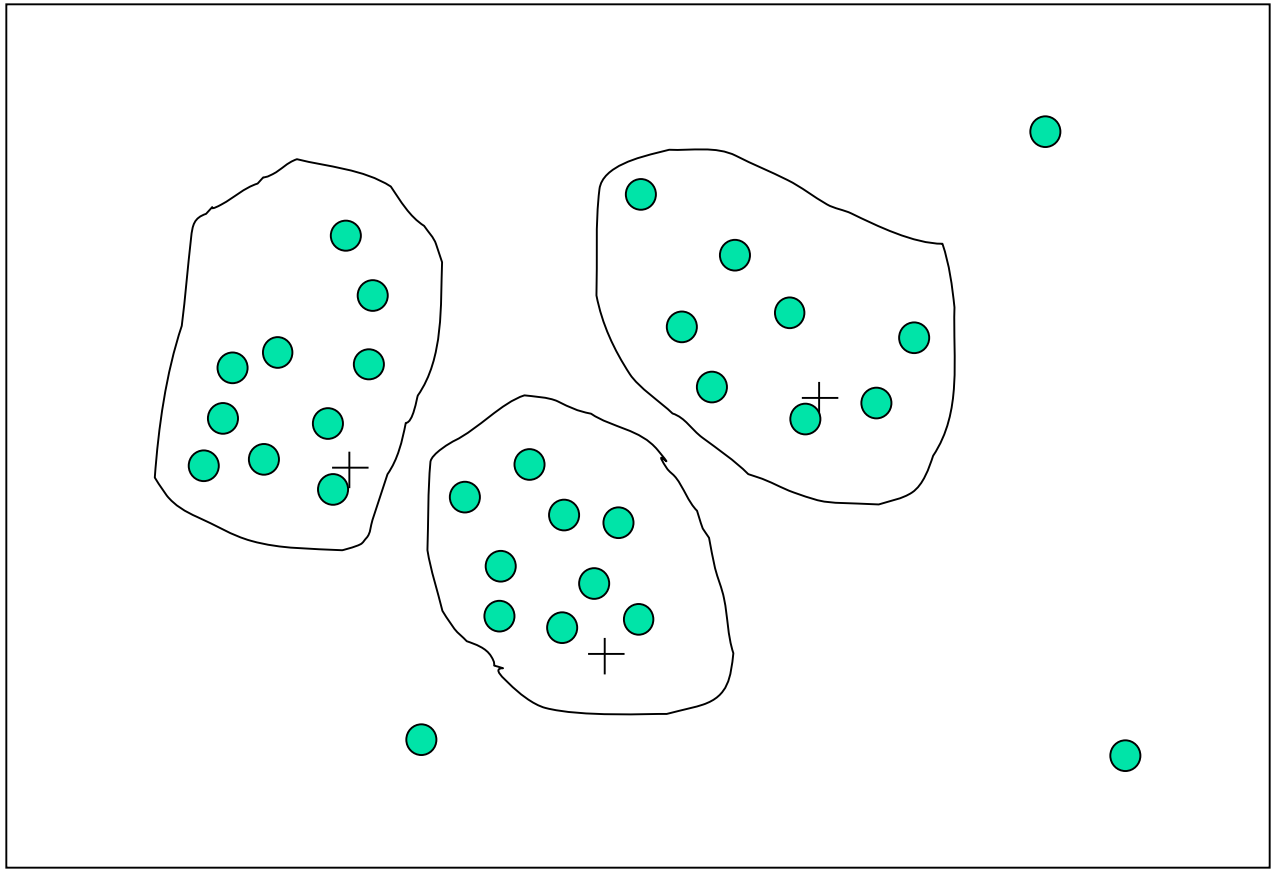
Simple Discretization Methods: Binning

- 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.

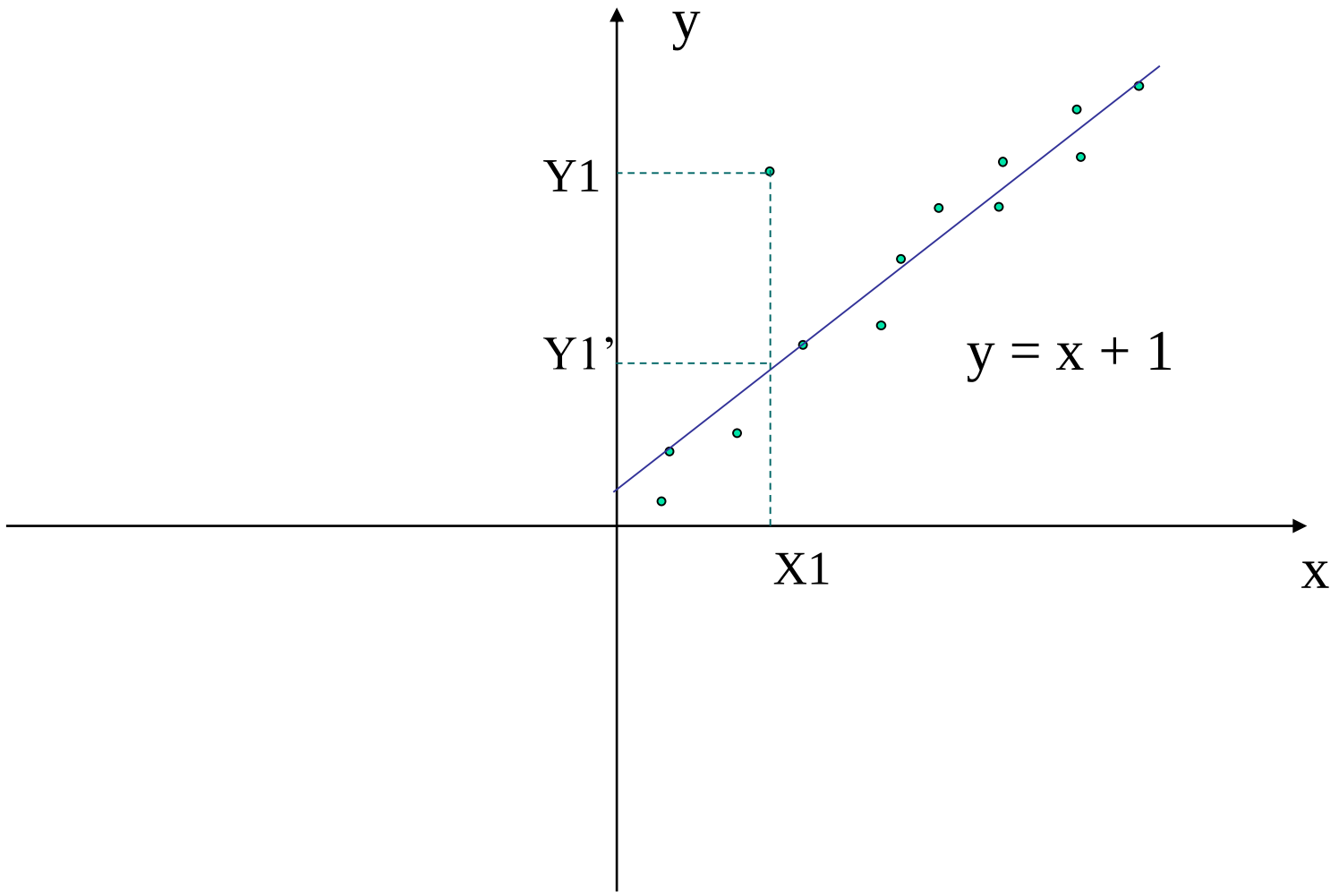
Binning Methods for Data Smoothing

* 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 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

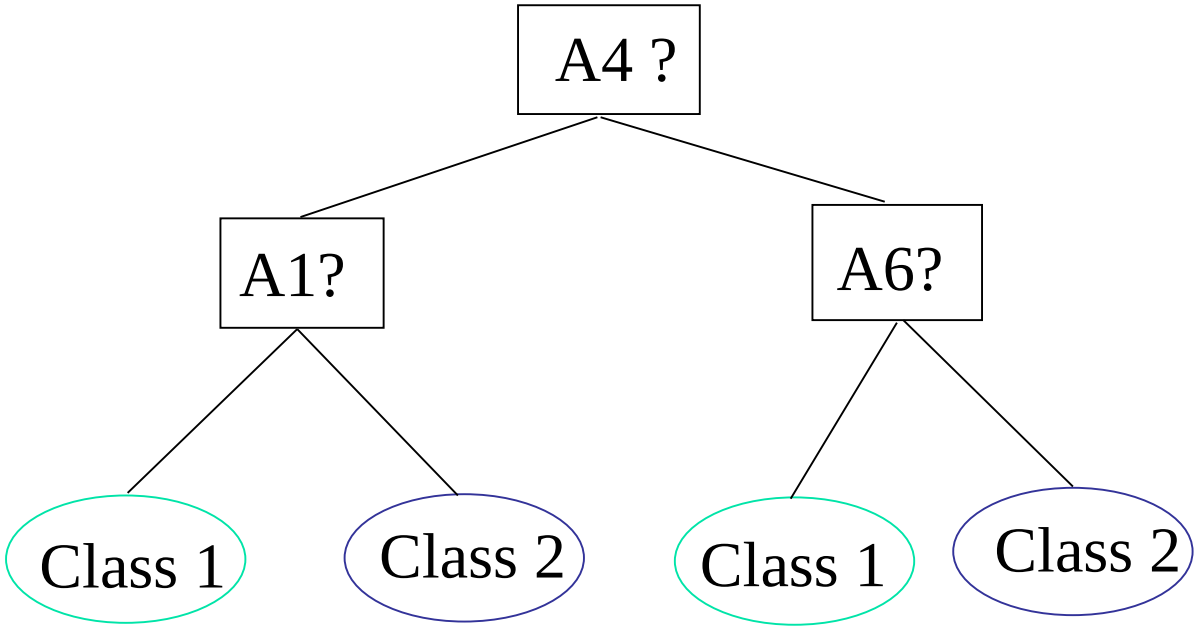
- The lowest level of a data cube
  - the aggregated data for an individual entity of interest
  - e.g., a customer in a phone calling data warehouse.
- Multiple levels of aggregation in data cubes
  - Further reduce the size of data to deal with
- Reference appropriate levels
  - Use the smallest representation 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}

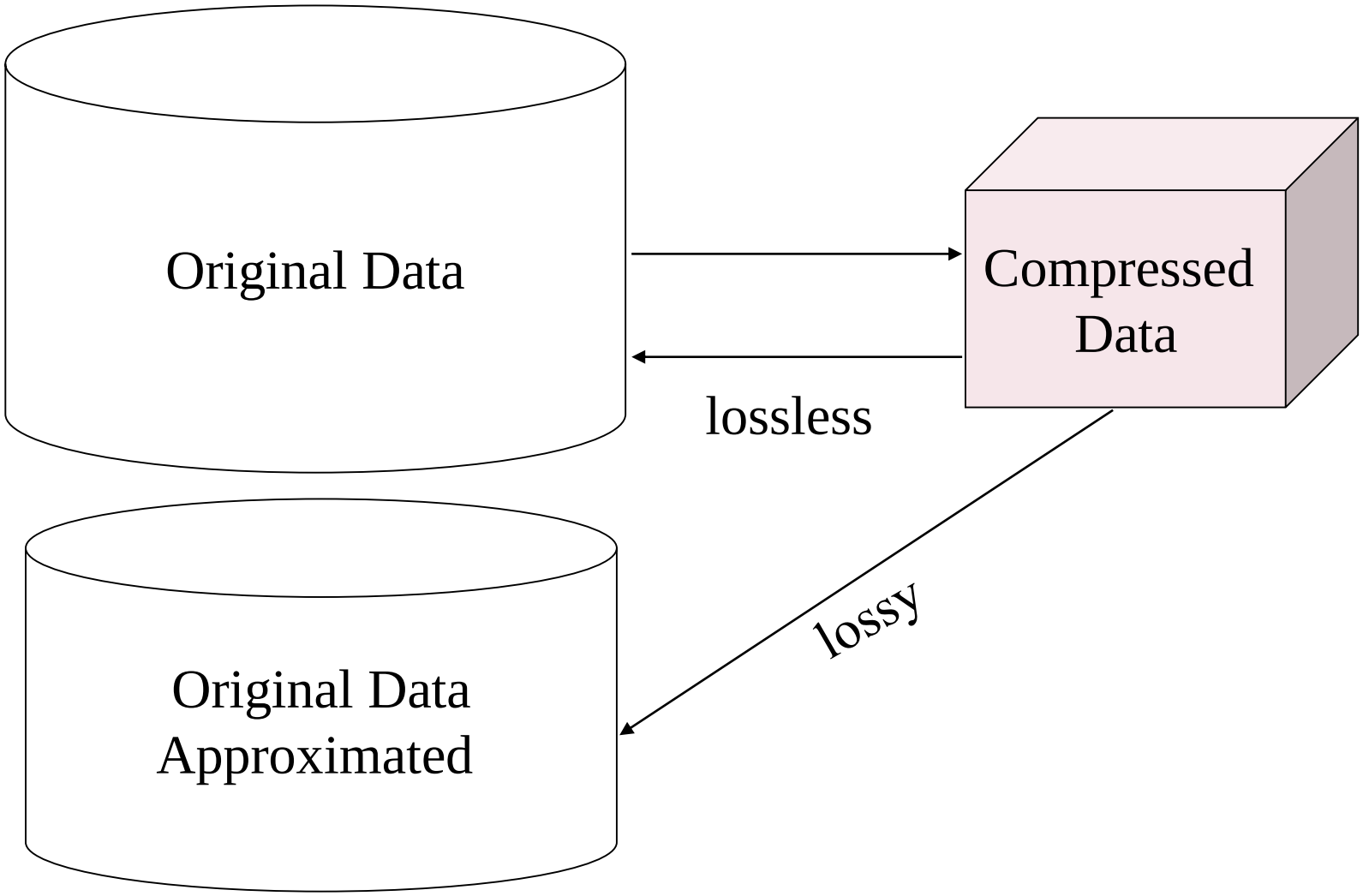


-------> Reduced attribute set: {A1, A4, A6}

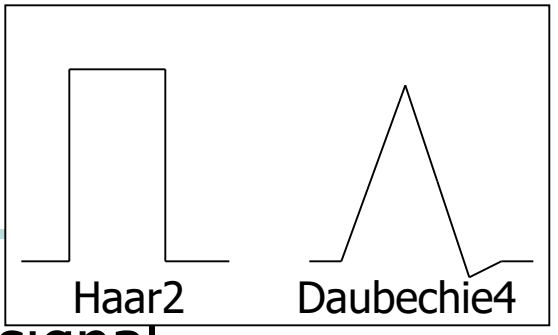
Data Compression

- 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

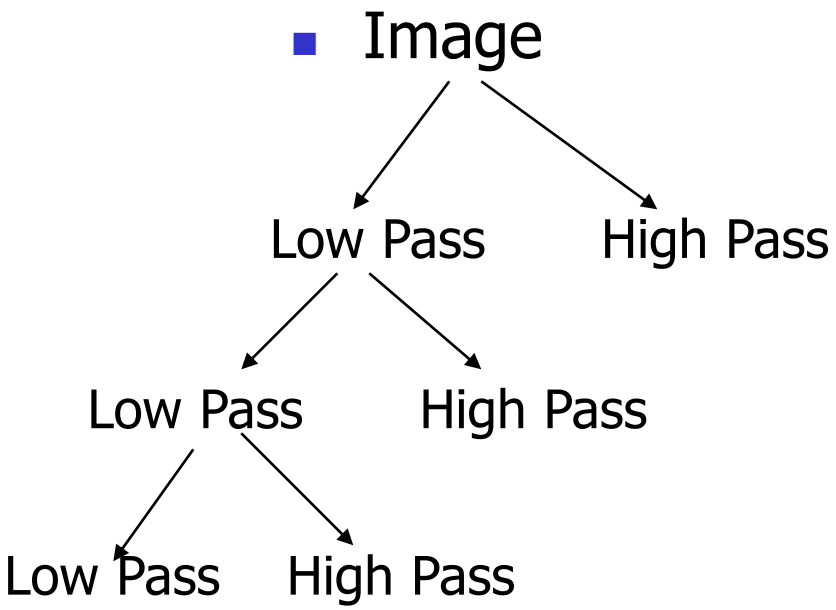


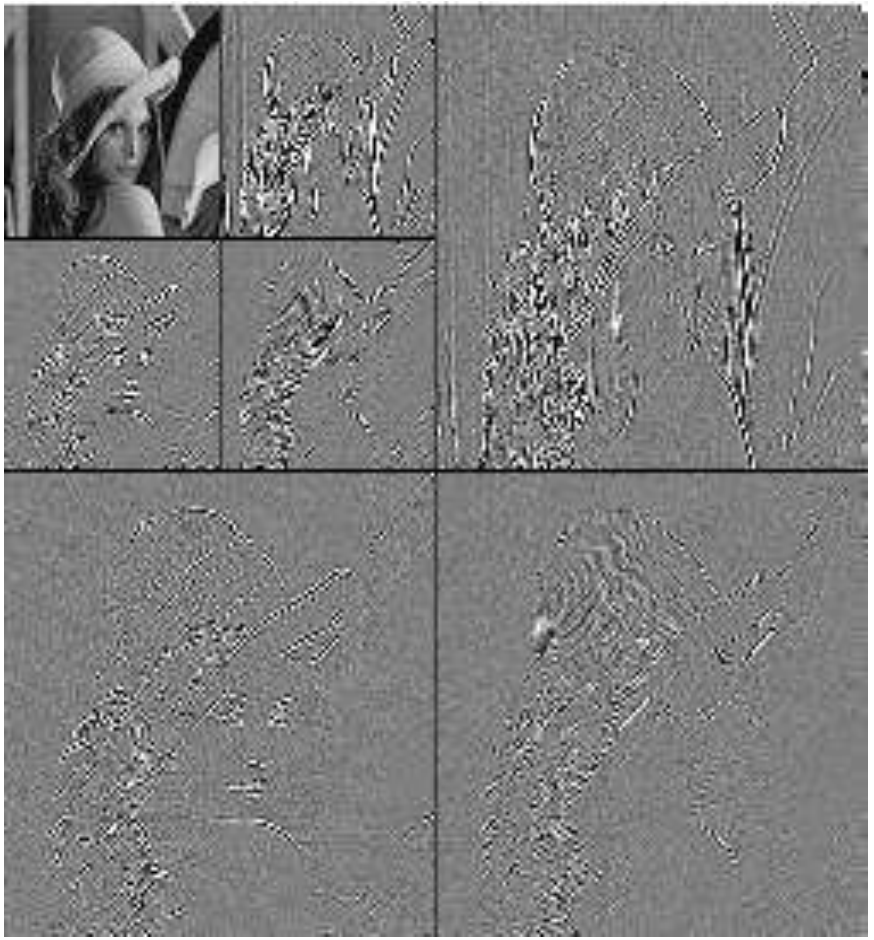
Wavelet Transformation



- 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

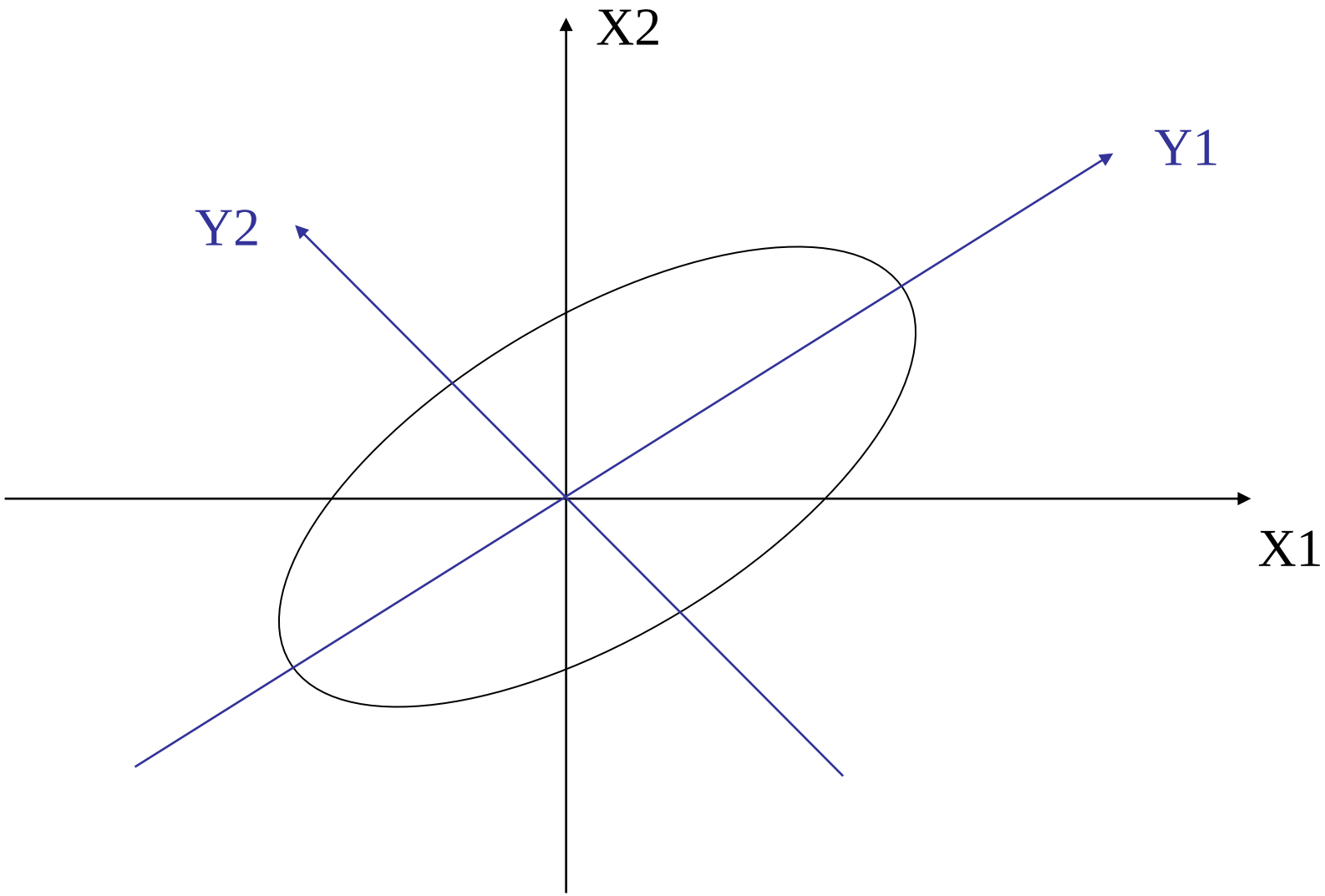




Principal Component Analysis

- 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

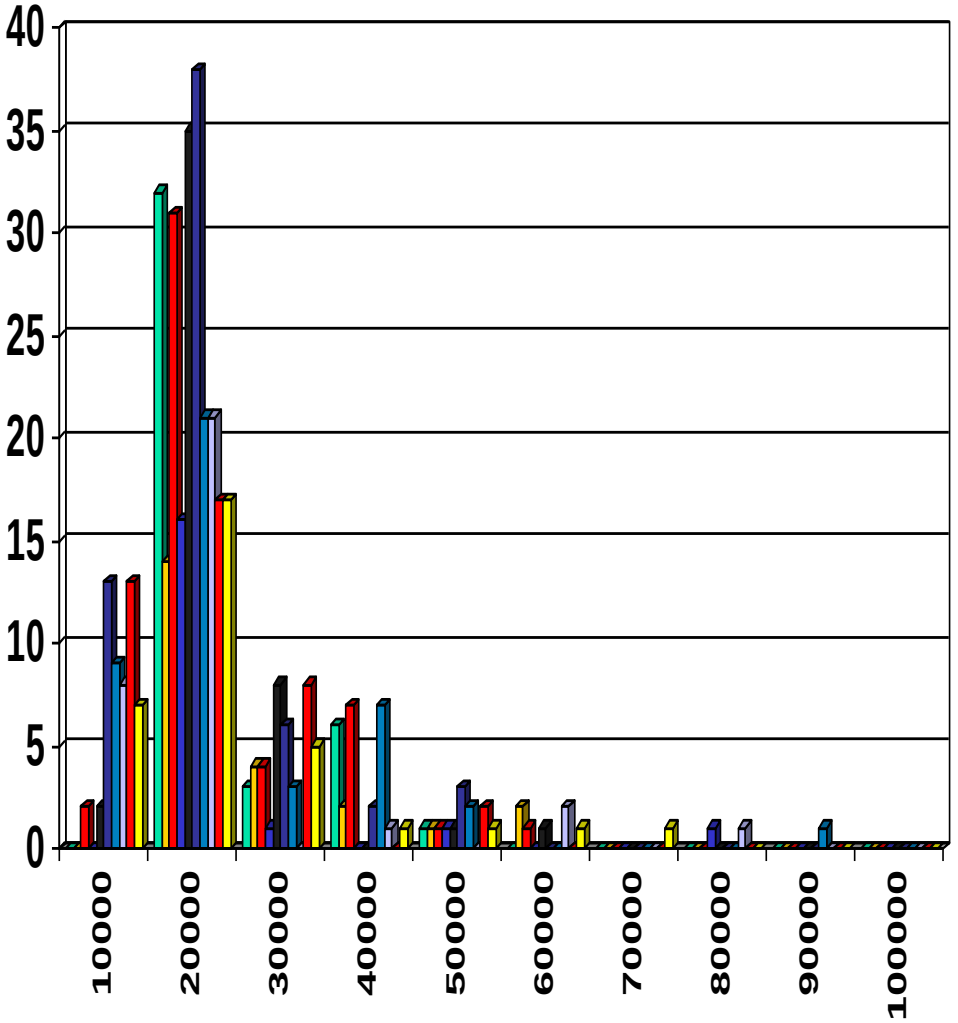
- Linear regression: Data are modeled to fit a straight line
  - Often uses the least-square method to fit the line
- Multiple regression: allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- Log-linear model: approximates discrete multidimensional probability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



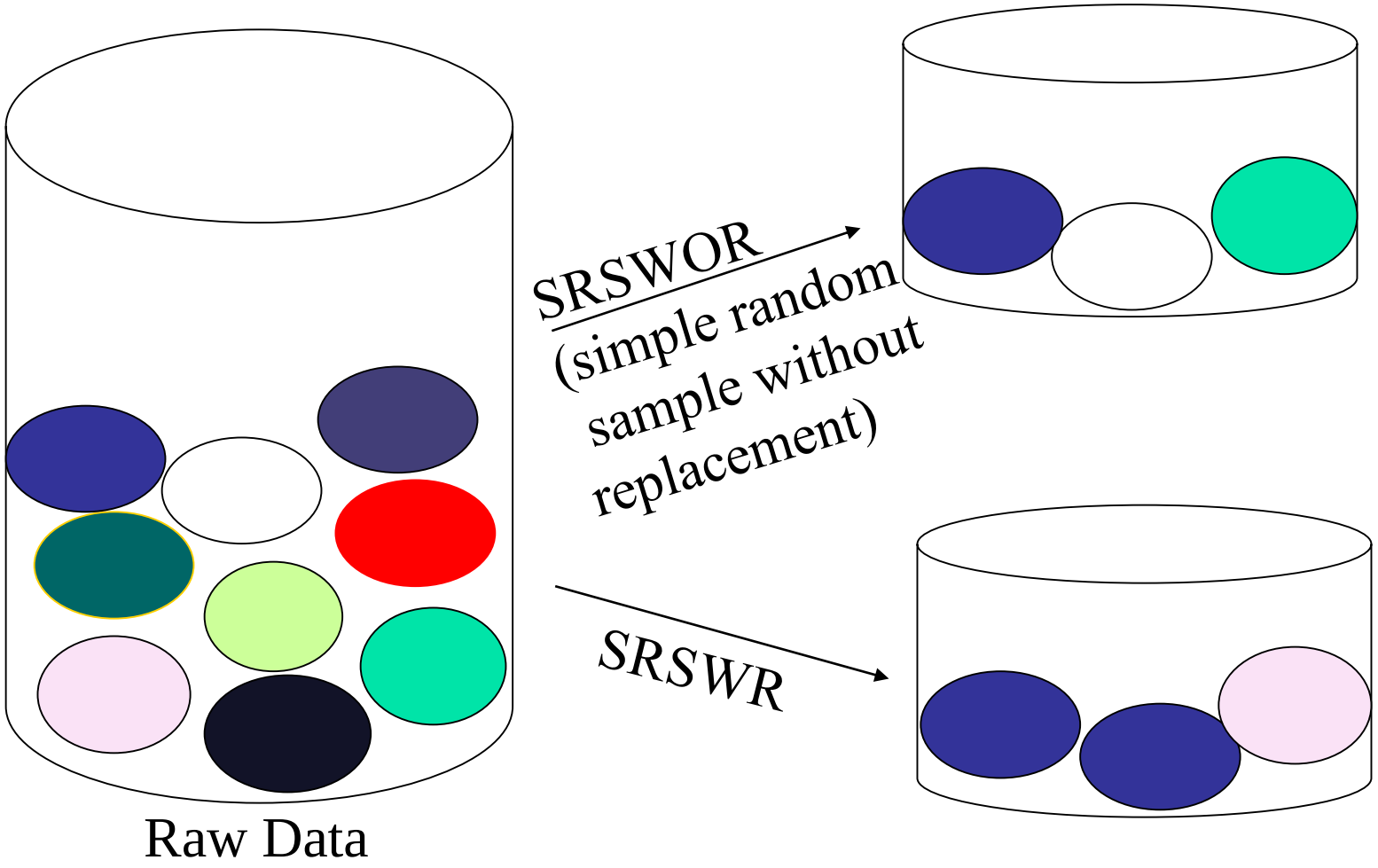
Clustering

- 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

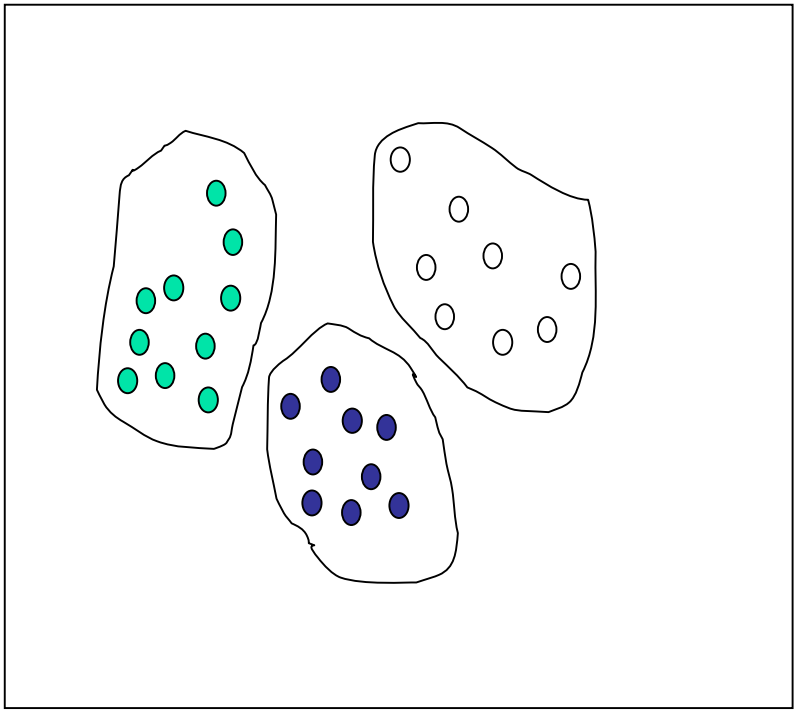
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
  - Stratified sampling:
    - Approximate 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

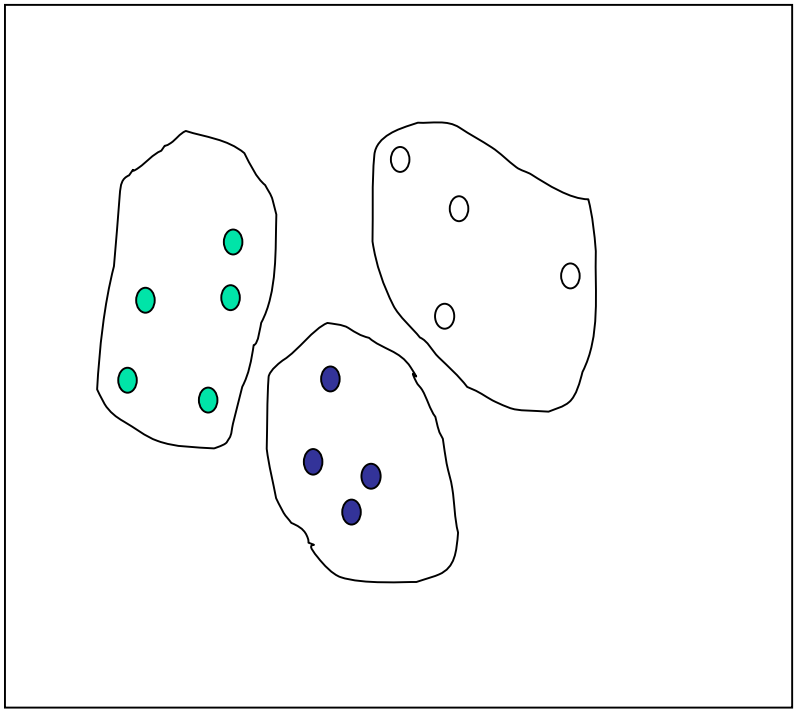

SRSWOR
(simple random sample without replacement)
SRSWR
Raw Data

Sampling

Raw Data



Cluster/Stratified Sample



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

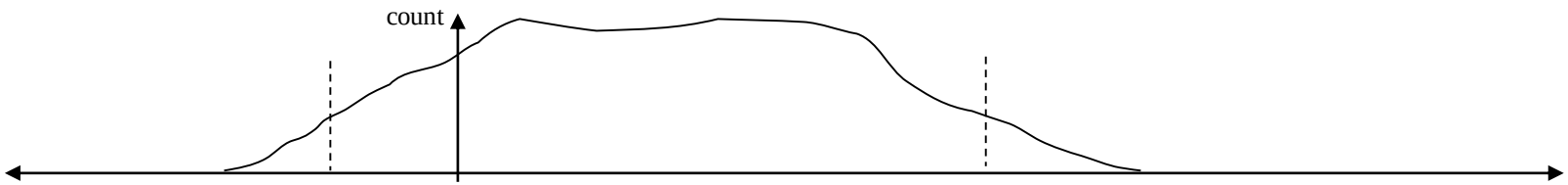
Entropy-Based Discretization

- 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

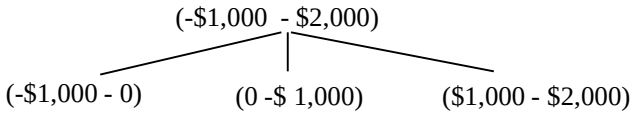


Step 1: -$351 -$159 profit $1,838 $4,700

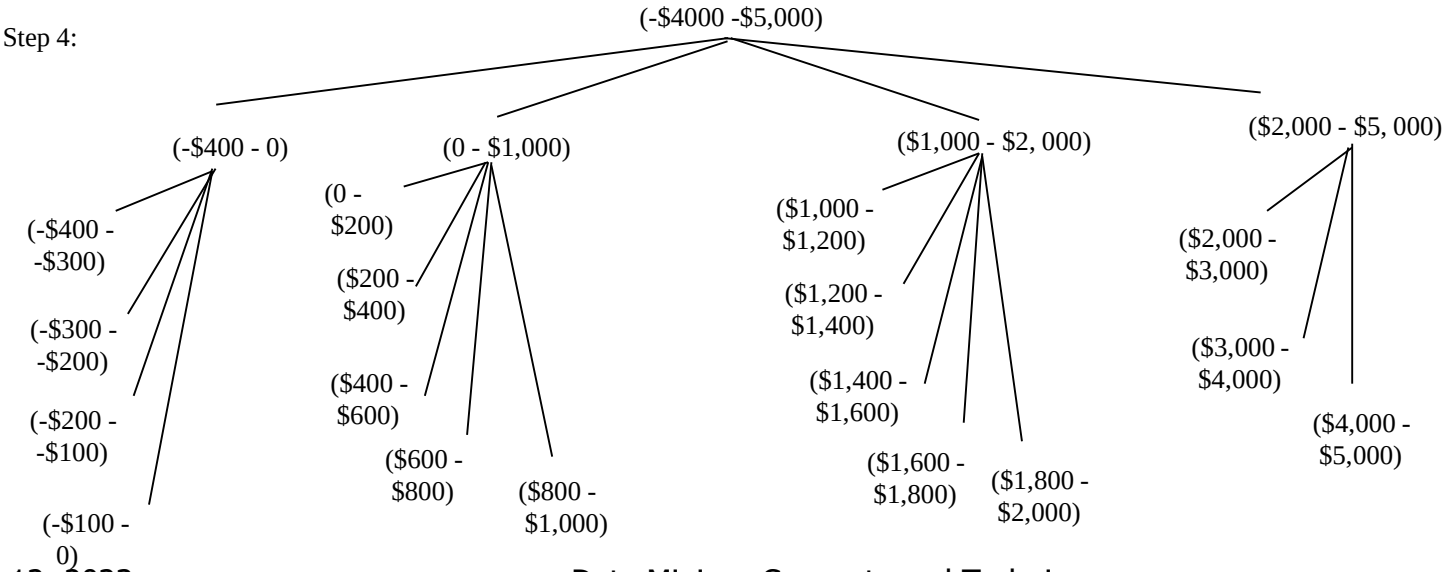
Min Low (i.e, 5%-tile) High(i.e, 95%-0 tile) Max

Step 2: msd=1,000 Low=-$1,000 High=$2,000

Step 3:



Step 4:

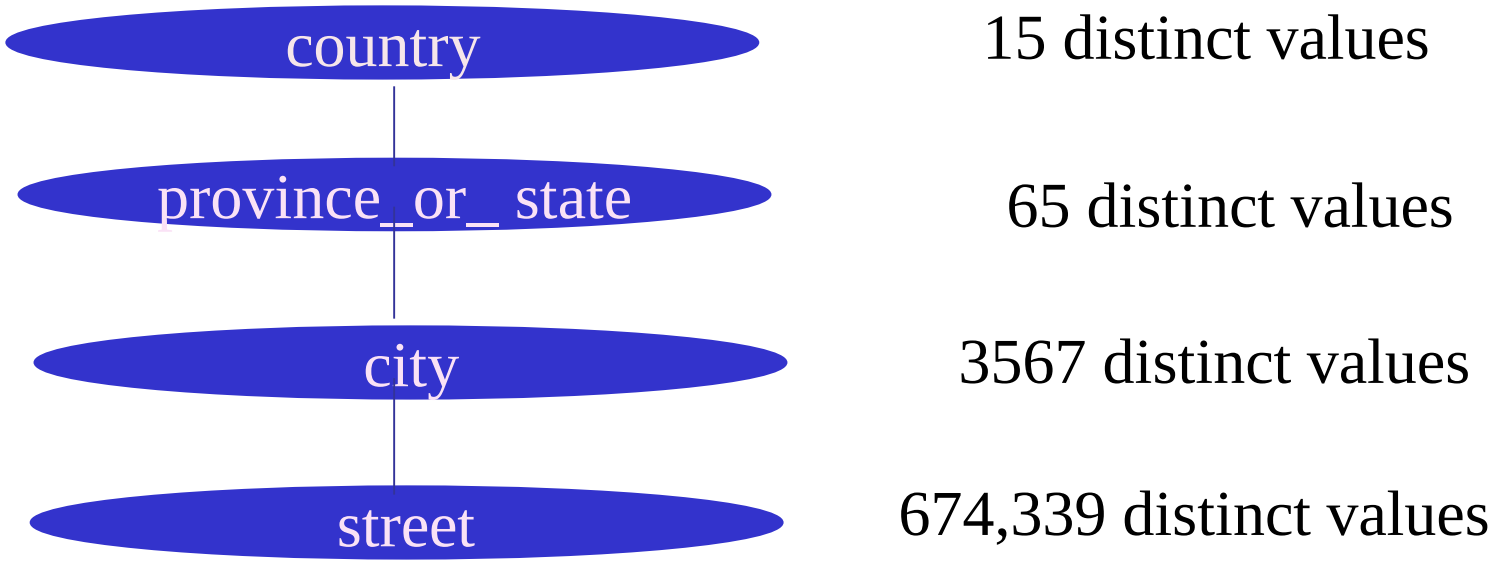


Concept Hierarchy Generation for Categorical Data

- 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

www.cs.uiuc.edu/~hanj

Thank you !!!