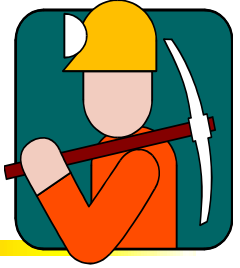
Why Data Mining?

- The Explosive Growth of Data: from terabytes to petabytes
  - Data collection and data availability
    - Automated data collection tools, database systems, Web, computerized society
  - Major sources of abundant data
    - Business: Web, e-commerce, transactions, stocks, …
    - Science: Remote sensing, bioinformatics, scientific simulation, …
    - Society and everyone: news, digital cameras, YouTube
- We are drowning in data, but starving for knowledge!
- "Necessity is the mother of invention"—Data mining—Automated analysis of massive data sets

Evolution of Database Technology

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

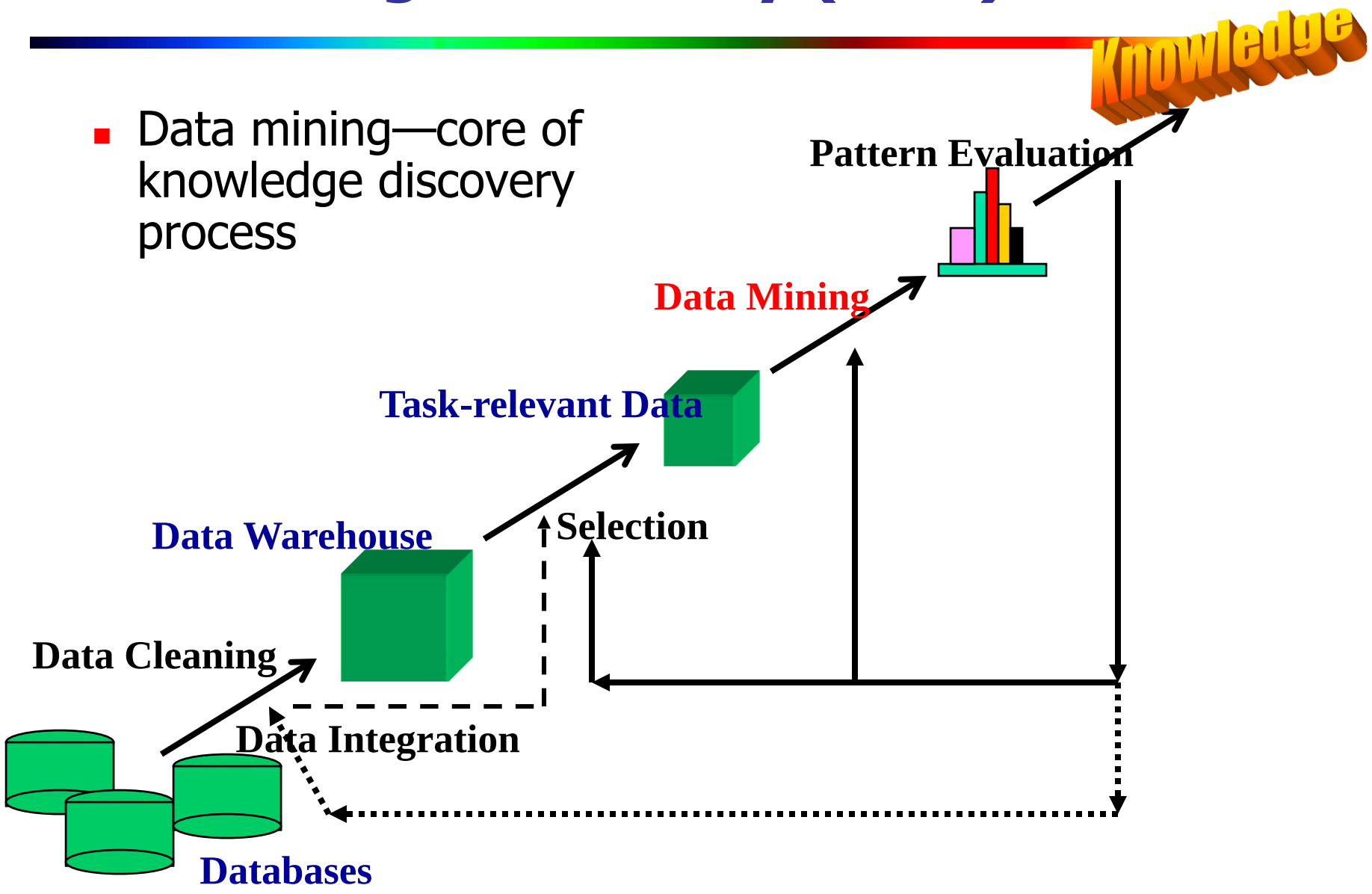
What Is Data Mining?



- Data mining (knowledge discovery from data)
  - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data
- Alternative names
  - Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.

Knowledge Discovery (KDD) Process

- Data mining—core of knowledge discovery process



KDD Process: Several Key Steps

- 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

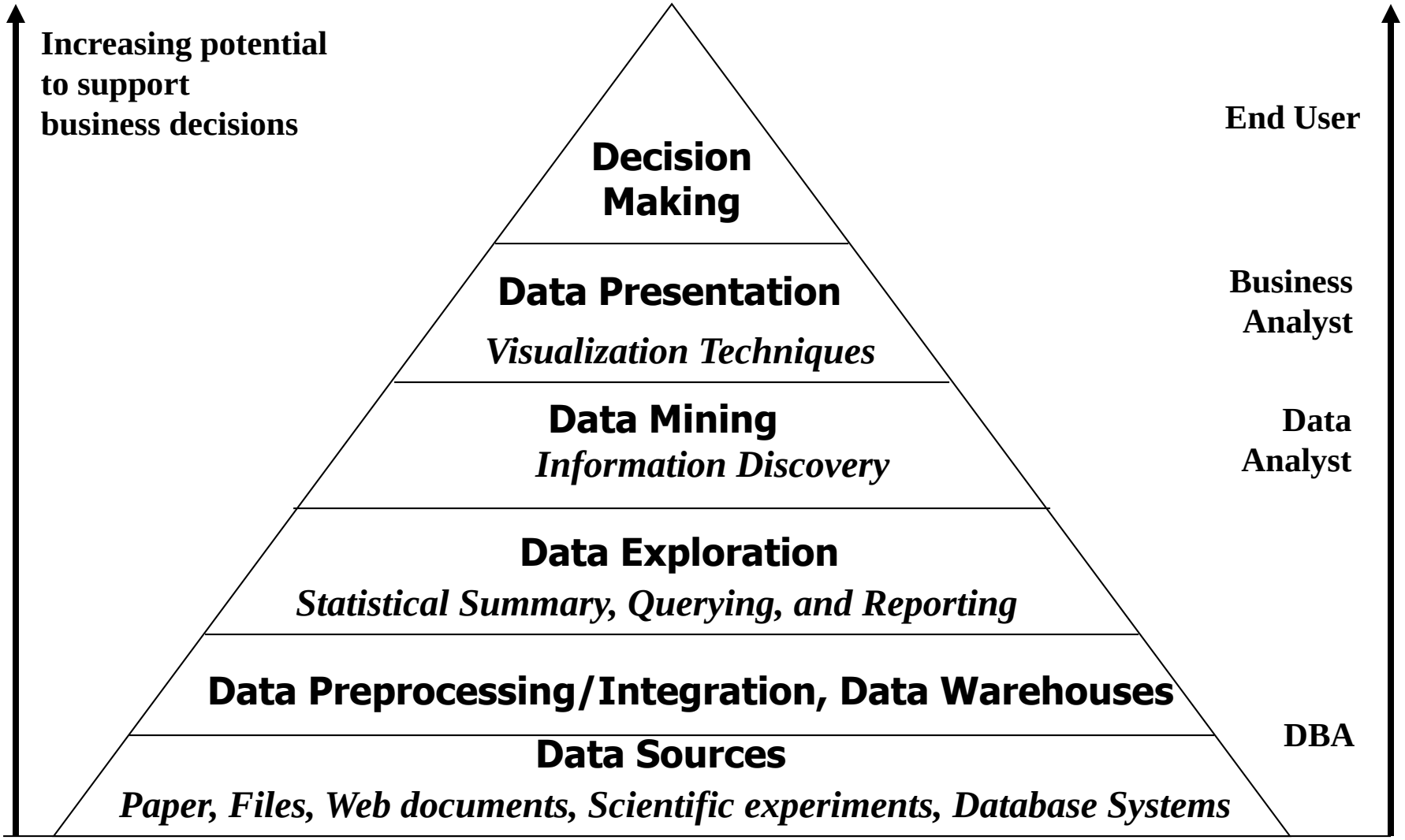
Are All the "Discovered" Patterns Interesting?

- 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.

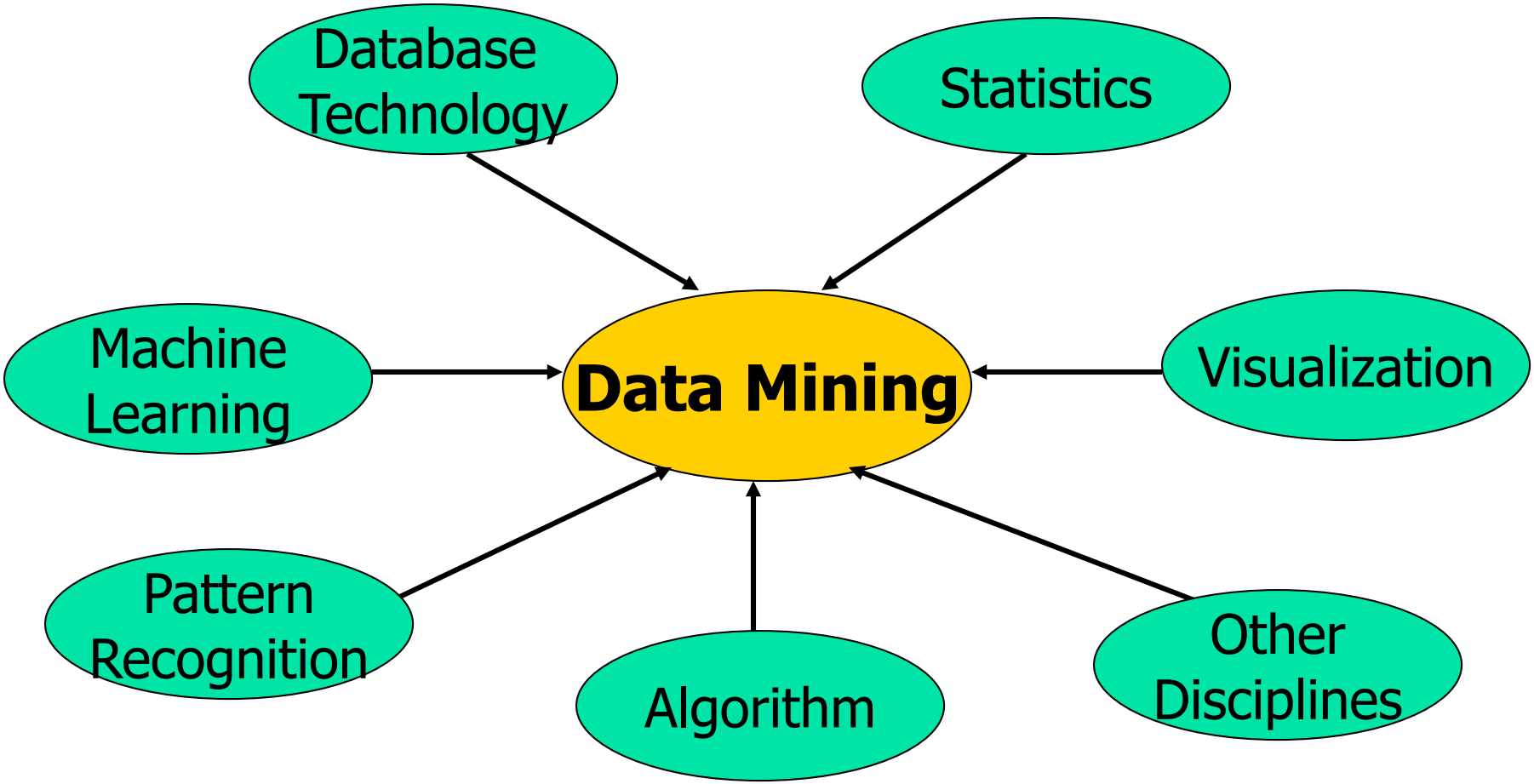
Find All and Only Interesting Patterns?

- Find all the interesting patterns: Completeness
  - Can a data mining system find all the interesting patterns? Do we need to find all of the interesting patterns?
  - 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

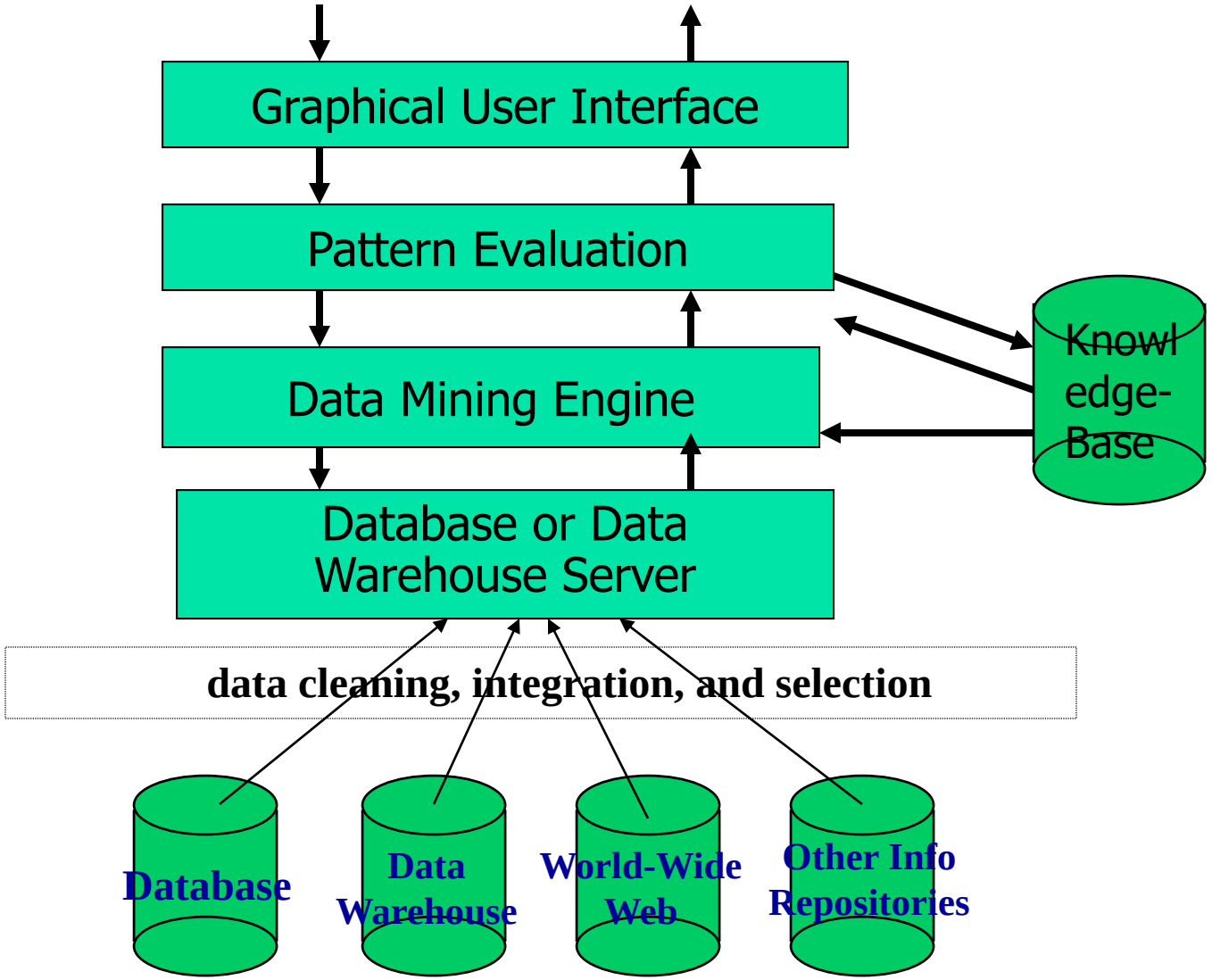
Data Mining and Business Intelligence



Data Mining: Confluence of Multiple Disciplines



Architecture: Typical Data Mining System



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, 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
  - Bioinformatics and bio-data analysis

Ex. 1: 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—Find associations/co-relations between product sales, & predict based on such association
- Customer profiling—What types of customers buy what products (clustering or classification)
- Customer requirement analysis
  - Identify the best products for different groups of customers
  - Predict what factors will attract new customers
- Provision of summary information
  - Multidimensional summary reports

Ex. 2: 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

Ex. 3: 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

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, text mining, Web mining, etc.

Data Mining: Classification Schemes

- General functionality
  - Descriptive data mining
  - Predictive data mining
- Different views lead to different classifications
  - Data view: Kinds of data to be mined
  - Knowledge view: Kinds of knowledge to be discovered
  - Method view: Kinds of techniques utilized
  - Application view: Kinds of applications adapted

Data Mining: On What Kinds of Data?

- Database-oriented data sets and applications
  - Relational database, data warehouse, transactional database
- Advanced data sets and advanced applications
  - Data streams and sensor data
  - Time-series data, temporal data, sequence data (incl. bio-sequences)
  - Structure data, graphs, social networks and multi-linked data
  - Object-relational databases
  - Heterogeneous databases and legacy databases
  - Spatial data and spatiotemporal data
  - Multimedia database
  - Text databases
  - The World-Wide Web

Data Mining Functionalities

- Multidimensional concept description: Characterization and discrimination
  - Generalize, summarize, and contrast data characteristics, e.g., dry vs. wet regions
- Frequent patterns, association, correlation vs. causality
  - Diaper → Beer [0.5%, 75%] (Correlation or causality?)
- 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)
  - 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: Data object that does not comply with the general behavior of the data
  - Noise or exception? Useful in fraud detection, rare events analysis
- Trend and evolution analysis
  - Trend and deviation: e.g., regression analysis
  - Sequential pattern mining: e.g., digital camera → large SD memory
  - Periodicity analysis
  - Similarity-based analysis
- Other pattern-directed or statistical analyses

Primitives that Define a Data Mining Task

- Task-relevant data
  - Database or data warehouse name
  - Database tables or data warehouse cubes
  - Condition for data selection
  - Relevant attributes or dimensions
  - Data grouping criteria
- Type of knowledge to be mined
  - Characterization, discrimination, association, classification, prediction, clustering, outlier analysis, other data mining tasks
- Background knowledge
- Pattern interestingness measurements
- Visualization/presentation of discovered patterns

Data Mining Tasks

- Data Mining makes use of various algorithms to perform a variety of tasks.
- These algorithms examine the sample data of a problem and determine a model that fits close to solving the problem.
- The models used to solve a problem are classified as predictive and descriptive.

Predictive Model

- A Predictive Model enables to predict the values of data by making use of known results from different set of sample data.
- The data mining tasks that forms the part of predictive model are:
- Classification
- Regression
- Time series Analysis

Descriptive Model

- A Descriptive Model enables to determine the patterns and relationships in a sample data.
- The data mining tasks that forms the part of Descriptive Model are:
- Clustering
- Summarization
- Association Rules
- Sequence Discovery

Classification

- Classification enables to classify the data in a large databank into pre defined set of classes.
- The classes are defined before studying or examining data in data bank.
- It not only enables to examine the existing sample data but also enable to predict future behavior of that sample data.
- Eg. Fraud detection in credit card related transactions.

Regression

- Regression enables to forecast future data values based on present and past data values.
- It examines the values of data and develops a mathematical formula.
- The result produced on using this formula enables to predict future behavior of existing data.
- Eg. Future behavior of Saving pattern

Time series Analysis

- It makes of current and past sample data to predict future values. The values used are evenly distributed as hourly,weekly,monthly,yearly,and so on.
- Eg. Examine trends in stock market for various companies for a specific period and accordingly make investments.

Clustering

- It enables to create new groups and classes based on the study of patterns and relationship between values of data in a data bank.
- It is similar to classification but does not require to predefine the groups of classes.
- It is known as unsupervised learning or segmentation.
- All those data items that resembles more closely with each other are clubbed together in a single group, known as clusters.

Summarization

- It enables to summarize a large chunk of data containing in a web page or document.
- The summarized data enables to get the gist of the entire web page or document.
- It is also known as characterization or generalization.
- It searches for specific characteristics and attributes of data in large amount of data and then summarizes the same.

Association Rules

- It enables to establish association and relationships between large unclassified data items based on certain attributes and characteristics.

Sequence Discovery

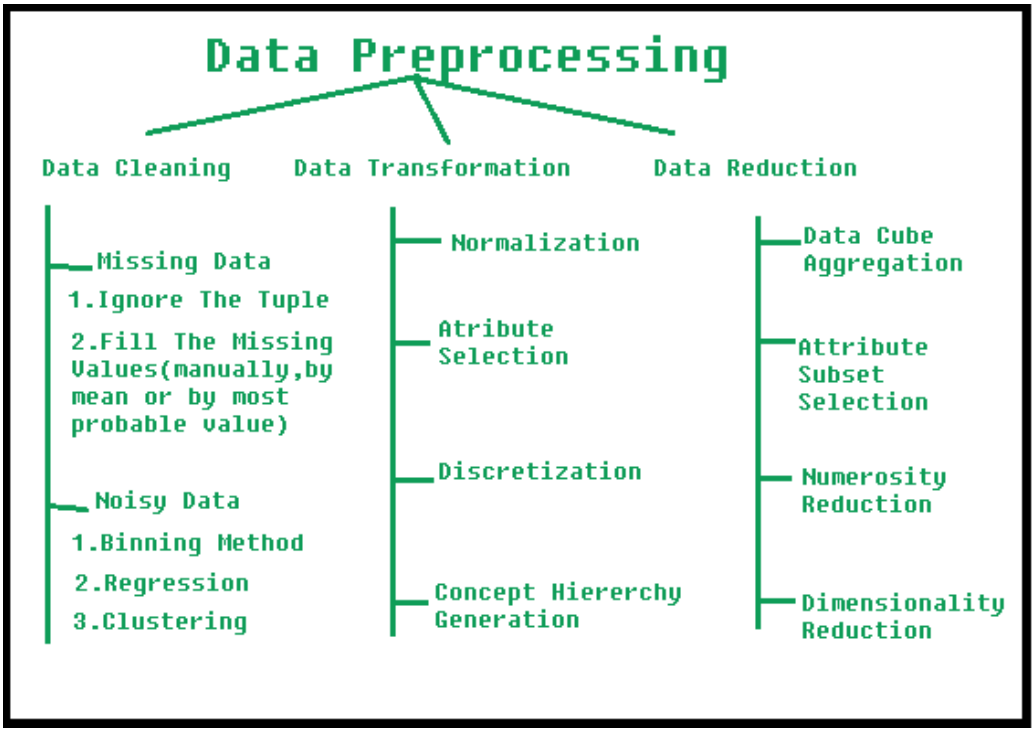
- It enables to determine the sequential patterns that exist in a large and unorganized data bank.

Data Preprocessing

Data Summarization: Summarization is a key data mining concept which involves techniques for finding a compact description of a dataset. Simple summarization methods such as tabulating the mean and standard deviations are often applied for exploratory data analysis, data visualization and automated report generation. Clustering [12, 21] is another data mining technique that is often used to summarize large datasets.

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing.

Data preprocessing is used database-driven applications such as customer relationship management and rule-based applications (like neural networks).



Steps Involved in Data Preprocessing:

1. Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

- **(a). Missing Data:**
  This situation arises when some data is missing in the data. It can be handled in various ways.
  Some of them are:
  1. **Ignore the tuples:**
     This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.
  2. **Fill the Missing values:**
     There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.
- **(b). Noisy Data:**
  Noisy data is a meaningless data that can't be interpreted by machines.It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :

  1. **Binning Method:**
     This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.
  2. **Regression:**
     Here data can be made smooth by fitting it to a regression function.The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).
  3. **Clustering:**
     This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

2. Data Transformation:

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. **Normalization:**
   It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)
2. **Attribute Selection:**
   In this strategy, new attributes are constructed from the given set of attributes to help the mining process.
3. **Discretization:**
   This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.
4. **Concept Hierarchy Generation:**
   Here attributes are converted from level to higher level in hierarchy. For Example-The attribute “city” can be converted to “country”.

3. Data Reduction:

Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we uses data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.
The various steps to data reduction are:

1. **Data Cube Aggregation:**
   Aggregation operation is applied to data for the construction of the data cube.
2. **Attribute Subset Selection:**
   The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value of the attribute.the attribute having p-value greater than significance level can be discarded.
3. **Numerosity Reduction:**
   This enable to store the model of data instead of whole data, for example: Regression Models.
4. **Dimensionality Reduction:**
   This reduce the size of data by encoding mechanisms.It can be lossy or lossless. If after reconstruction from compressed data, original data can be retrieved, such reduction are called lossless reduction else it is called lossy reduction. The two effective methods of dimensionality reduction are:Wavelet transforms and PCA (Principal Componenet Analysis).

Data Integration: is a data preprocessing technique that involves combining data from multiple heterogeneous data sources into a coherent data store and provide a

unified view of the data. These sources may include multiple data cubes, databases or flat files.
The data integration approach are formally defined as triple <G, S, M> where,
G stand for the global schema,
S stand for heterogenous source of schema,
M stand for mapping between the queries of source and global schema.

There are mainly 2 major approaches for data integration – one is “tight coupling approach” and another is “loose coupling approach”.

Tight Coupling:
Here, a data warehouse is treated as an information retrieval component.

- In this coupling, data is combined from different sources into a single physical location through the process of ETL – Extraction, Transformation and Loading.

Loose Coupling:

- Here, an interface is provided that takes the query from the user, transforms it in a way the source database can understand and then sends the query directly to the source databases to obtain the result.
- And the data only remains in the actual source databases.

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.

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—Nonvolatile

- 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*

- A data warehouse is a semantically consistent data store that serves as a physical implementation of a decision support data model and stores the information on which an enterprise needs to make strategic decisions.

Data Warehouse vs. Operational DBMS

- The major task of on-line operational database systems is to perform on-line transaction and query processing. These systems are called **on-line transaction processing (OLTP) systems.** They cover most of the day-to-day operations of an organization, such as purchasing, inventory, manufacturing, banking, payroll, registration, and accounting.
- Data warehouse systems, on the other hand, serve users or knowledge workers in the role of data analysis and decision making. Such systems can organize and present data in various formats in order to accommodate the diverse needs of the different users. These systems are known as **on-line analytical processing (OLAP) systems.**

Distinct features (OLTP vs. OLAP):

- **Users and system orientation:**
- An OLTP system is *customer-oriented* and is used for transaction and query processing by clerks, clients, and information technology professionals.
- An OLAP systems *market-oriented* and is used for data analysis by knowledge workers, including managers, executives, and analysts.
- **Data contents:**
- An OLTP system manages current data that, typically, are too detailed to be easily used for decision making.
- An OLAP system manages large amounts of historical data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity. These features make the data easier to use in informed decision making.

- **View:** An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historical data or data in different organizations.
- In contrast, an OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization. OLAP systems also deal with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, OLAP data are stored on multiple storage media.
- **Access patterns:** The access patterns of an OLTP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanisms.
- Accesses to OLAP systems are mostly read-only operations (because most data warehouses store historical rather than up-to-date information), although many could be complex queries.

OLTP vs. OLAP

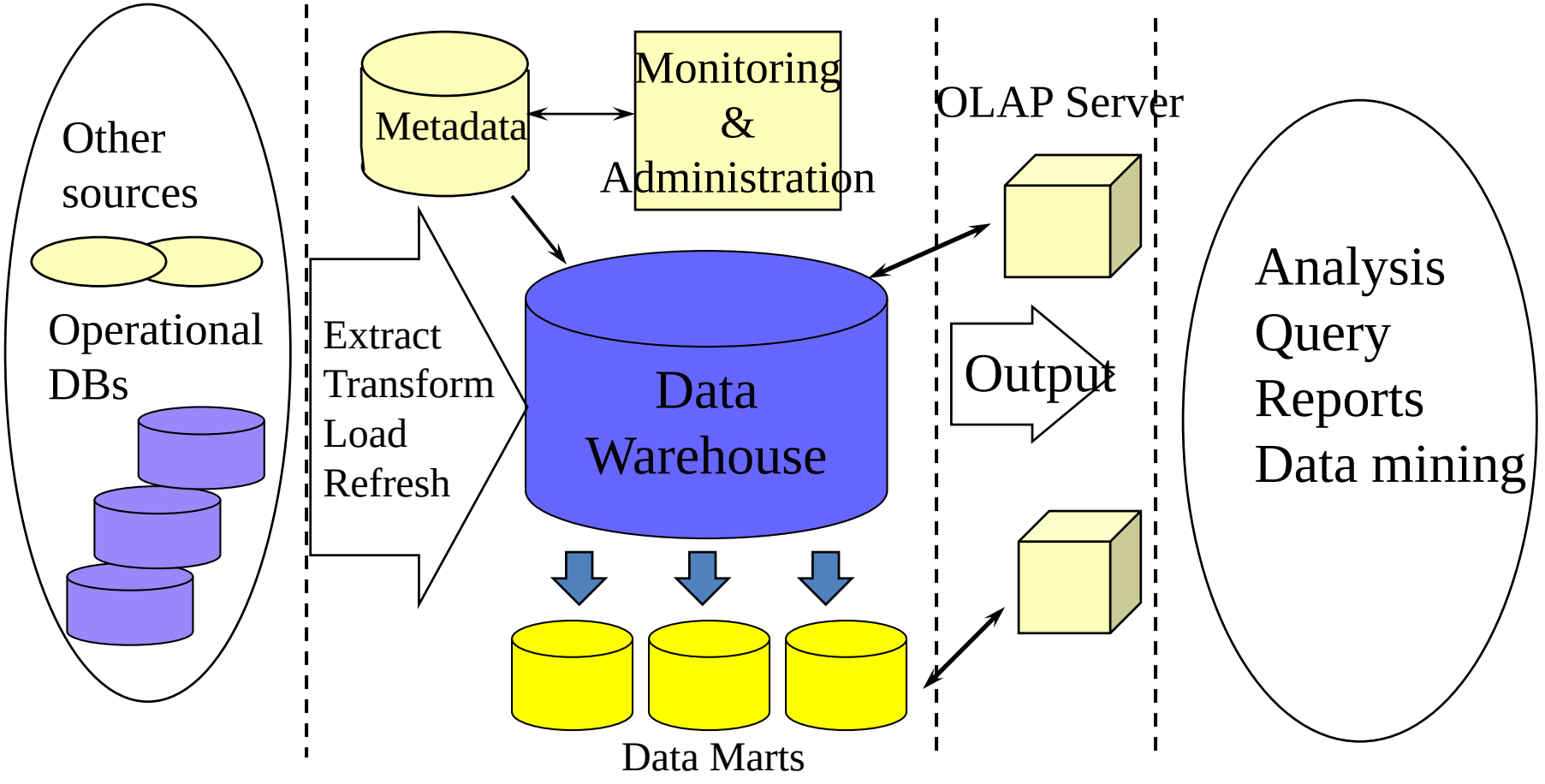
	OLTP	**OLAP**
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
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

Data warehouse architecture

Design of 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. This information matches the current and future business needs.
  - Data source view
- exposes the information being captured, stored, and managed by operational systems. This information may be documented at various levels of detail and accuracy, from individual data source tables to integrated data source tables.
  - Data warehouse view
- consists of fact tables and dimension tables. It represents the information that is stored inside the data warehouse, including precalculated totals and counts, as well as information regarding the source, date, and time of origin, added to provide historical context.
  - Business query view
    - sees the perspectives of data in the warehouse from the view point of end-user

Data Warehouse: A Multi-Tiered Architecture



- **1.** The **bottom tier** is a warehouse database server that is almost always a relational database system.

- Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources (such as customer profile information provided by external consultants).
- These tools and utilities perform data extraction, cleaning, and transformation (e.g., to merge similar data from different sources into a unified format), as well as load and refresh functions to update the data warehouse.
- This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

2. The **middle tier** is an OLAP server that is typically implemented using either
(1) a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or
(2) a multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations.

3. The **top tier** is a front-end client layer, which contains query and reporting tools,
analysis tools, and/or data mining tools (e.g., trend analysis, prediction, and so on).

Three Data Warehouse Models

- From the architecture point of view, there are three data warehouse models: the *enterprise warehouse*, the *data mart*, and the *virtual warehouse*.
- 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 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

Metadata Repository

- Meta data is the data defining warehouse objects. It stores:
- Description of the structure of the data 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

Types of OLAP Servers: ROLAP versus MOLAP versus HOLAP

- OLAP servers present business users with multidimensional data from data warehouses or data marts, without concerns regarding how or where the data are stored.
- The physical architecture and implementation of OLAP servers must consider data storage issues. Implementations of a warehouse server for OLAP processing include the following:

- **Relational OLAP (ROLAP) servers:** These are the intermediate servers that stand in between a relational back-end server and client front-end tools. They use a *relational or extended-relationalDBMS* to store and manage warehouse data, and OLAP middleware to support missing pieces. ROLAP technology tends to have greater scalability than MOLAP technology.

- **Multidimensional OLAP (MOLAP) servers:** These servers support multidimensional views of data through *array-based multidimensional storage engines*. They map multidimensional views directly to data cube array structures. The advantage of using a data cube is that it allows fast indexing to precomputed summarized data.
- **Hybrid OLAP (HOLAP) servers:** The hybrid OLAP approach combines ROLAP and MOLAP technology, benefiting from the greater scalability of ROLAP and the faster computation of MOLAP.
- **Specialized SQL servers:** To meet the growing demand of OLAP processing in relational databases, some database system vendors implement specialized SQL servers that provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

Data Warehouse Implementation

- Data warehouses contain huge volumes of data.
- OLAP servers demand that decision support queries be answered in the order of seconds. Therefore, it is crucial for data warehouse systems to support highly efficient cube computation techniques, access methods, and query processing techniques.

Data Warehouse Usage

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

What Is Frequent Pattern Analysis?

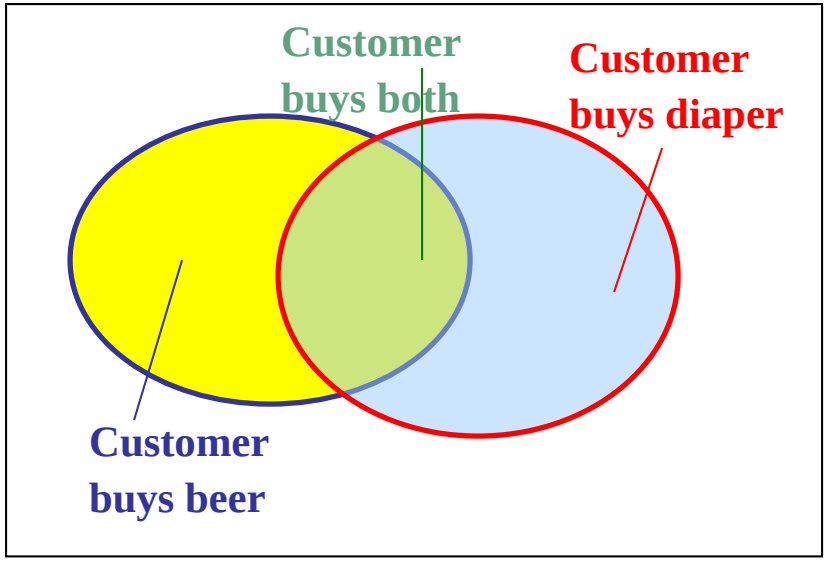
- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami in the context of frequent itemsets and association rule mining
- Motivation: Finding inherent regularities in data
  - What products 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?
- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.

Why Is Freq. Pattern Mining Important?

- Discloses an intrinsic and important property of data sets
- Forms the foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: associative classification
  - Cluster analysis: frequent pattern-based clustering
  - Broad applications

Basic Concepts: Frequent Patterns and Association Rules

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



- Itemset X = $\{x_1, \ldots, x_k\}$
- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, *s*, probability that a transaction contains $X \cup Y$
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*

Let sup_{min} *= 50%,* $conf_{min}$ *= 50%*
Freq. Pat.: {A:3, B:3, D:4, E:3, AD:3}
Association rules:

A → *D* (60%, 100%)
D → *A* (60%, 75%)

Association Rule mining is a two step process:

1. Find all frequent itemsets:

Each of these itemsets will occur atleast as frequently as a predetermined minimum support count

2. Generate strong association rules from frequent itemsets:

These rules must satisfy minimum support and minimum confidence.

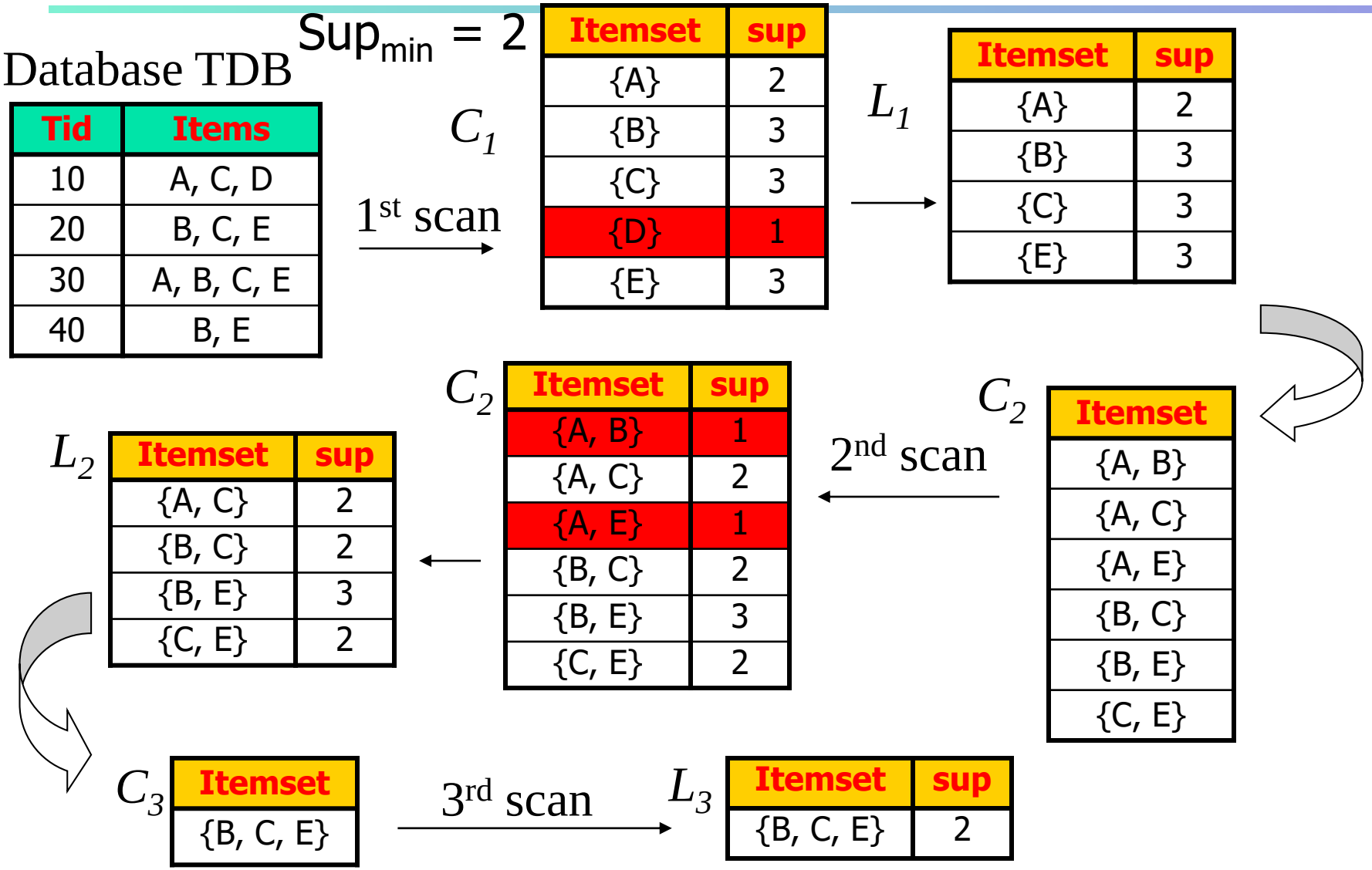
Apriori: Finding frequent itemsets mining using A Candidate Generation

- **Apriori Property:**

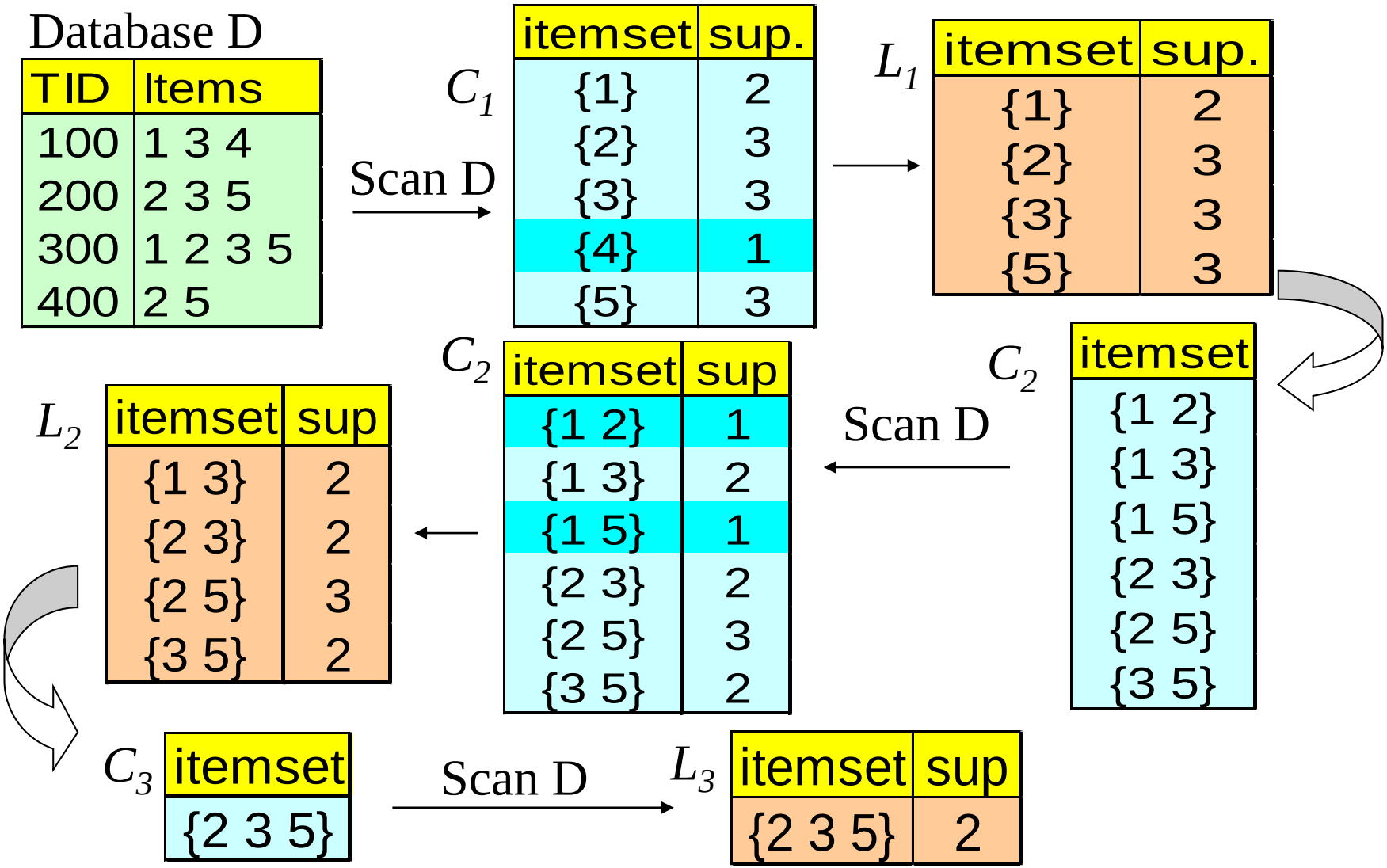
"All non-empty subset of frequent item set must be frequent"

- Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested!
- Method:
  - Initially, scan DB once to get frequent 1-itemset
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Test the candidates against DB
  - Terminate when no frequent or candidate set can be generated

The Apriori Algorithm—An Example



The Apriori Algorithm — Example



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*}

Generating Association Rules from Frequent Itemsets

- Once the frequent item sets from transactions in a database D have been found, it is straightforward to generate strong association rules from them (where strong association rules satisfy both minimum support and minimum confidence).
- This can be done using the following equation for confidence.

- Confidence (A => B) = P (B|A) = support_count (AUB) / support_count (A)

- The conditional probability is expressed in terms of item set support_count, where support_count (AUB) is the number of transactions containing the item set AUB, and support_count (A) is the number of transactions containing the item set A.
- Based on this equation, association rules can be generated as follows:

- For each frequent item set l, generate all non empty subsets of l.
- For every non empty subset s of l, output the rule "s => (l-s)" if

  support_count(l) / support_count(s) ≥ min_conf,

where min_conf is the minimum confidence threshold.

- Because the rules are generated from frequent item sets, each one automatically satisfies minimum_support.

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

Mining Frequent Patterns Without Candidate Generation

Construct FP-tree from a Transaction Database

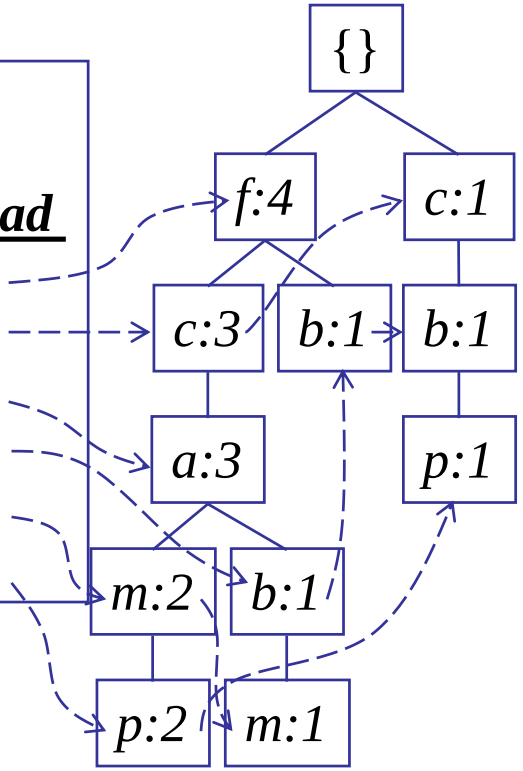
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

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, construct FP-tree

Header Table

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



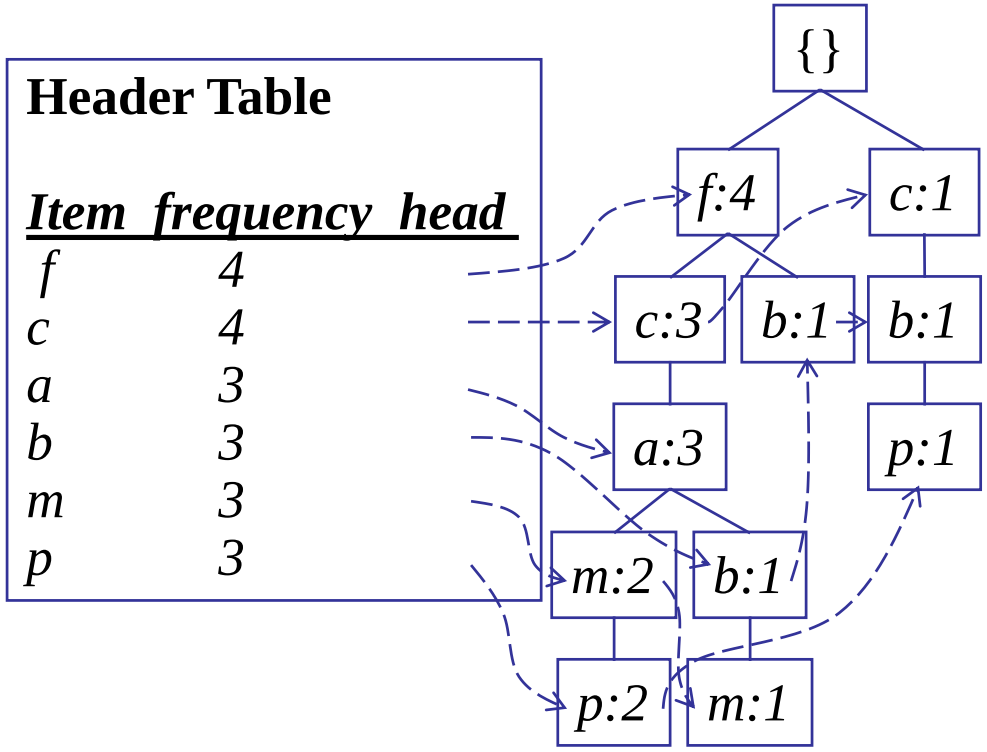
F-list=f-c-a-b-m-p

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

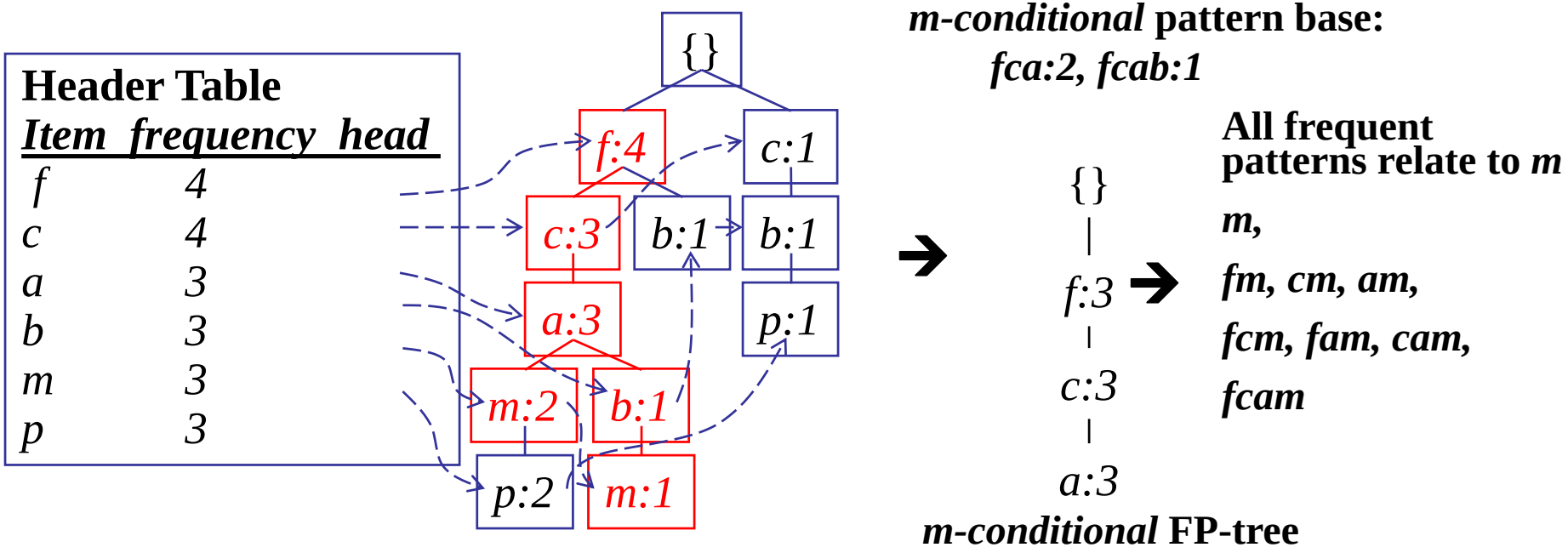


***Conditional* pattern bases**

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***

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



Mining Frequent Patterns With FP-trees

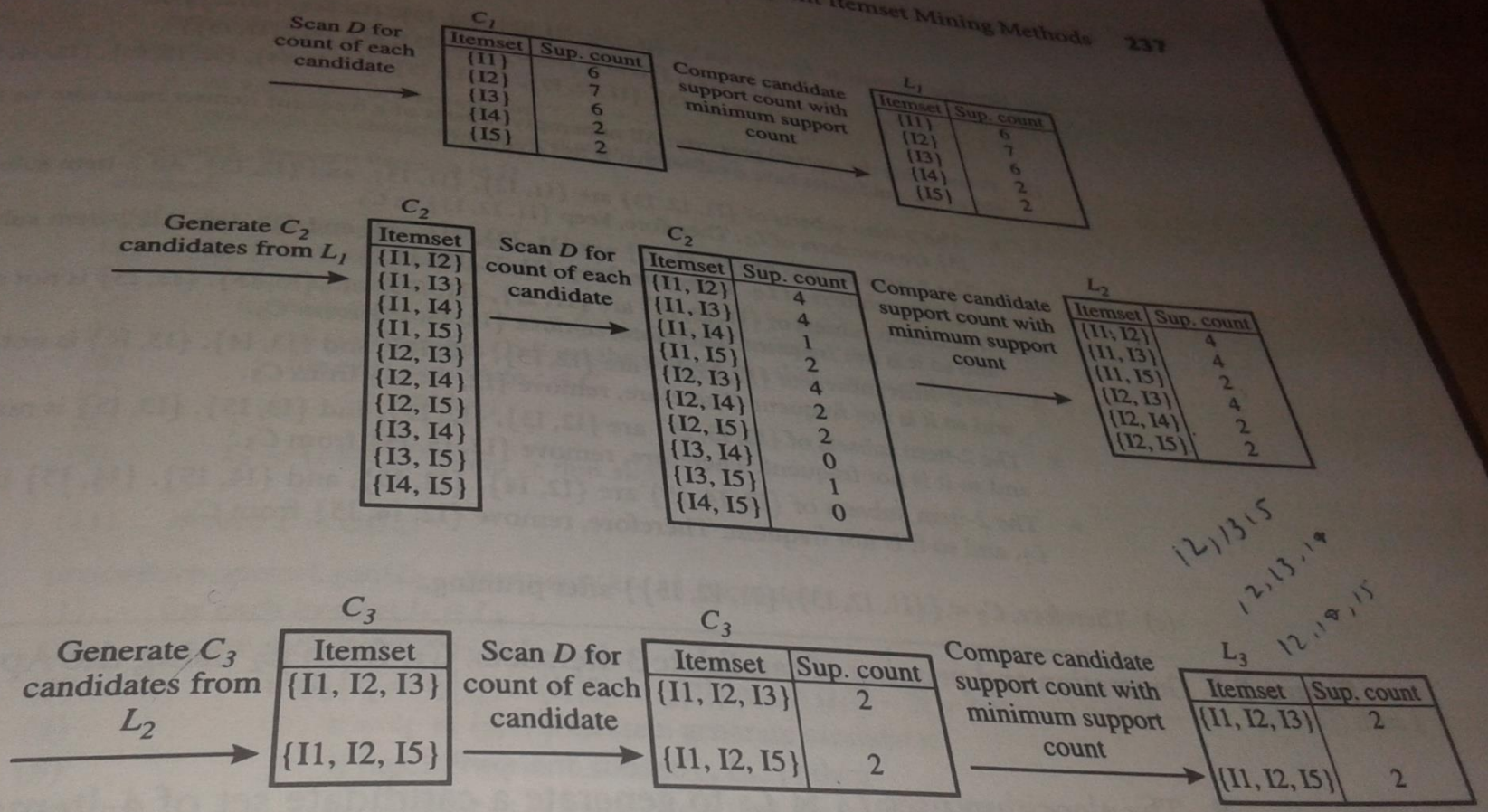
- Idea: Frequent pattern growth
  - Recursively grow frequent patterns by pattern and database partition
- Method
  - For each frequent item, construct its conditional pattern-base, and then its conditional FP-tree
  - Repeat the process on each newly created conditional FP-tree
  - Until the resulting 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

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

Table 5.1 Transactional data for an *AllElectronics* branch.

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



e 5.2 *Generation of candidate itemsets and frequent itemsets, where the minimum support count is 2.*

C3=L2*L2

- {I1,I2,I3}
- {I1,I2,I5}
- {I1,I3,I5}
- {I2,I3,I4}
- {I2,I3,I5}
- {I2,I4,I5}

- C4=L3*L3
- {I1,I2,I3,I5} is pruned because its subset{i2,i3,i5} is not frequent.
- So C4=0 and the algorithm terminates having found all the frequent itemsets.

Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

- D, a database of transactions;
- *min_sup*, the minimum support count threshold.

Output: L, frequent itemsets in D.

Method:

```
(1)  L1 = find_frequent_1-itemsets(D);
(2)  for (k = 2; Lk-1 ≠ ϕ; k++) {
(3)      Ck = apriori_gen(Lk-1);
(4)      for each transaction t ∈ D { // scan D for counts
(5)          Ct = subset(Ck, t); // get the subsets of t that are candidates
(6)          for each candidate c ∈ Ct
(7)              c.count++;
(8)      }
(9)      Lk = {c ∈ Ck | c.count ≥ min_sup}
(10) }
(11) return L = ∪k Lk;
```

```
procedure apriori_gen(L_{k-1}:frequent (k-1)-itemsets)
(1)   for each itemset l_1 ∈ L_{k-1}
(2)     for each itemset l_2 ∈ L_{k-1}
(3)       if (l_1[1] = l_2[1]) ∧ (l_1[2] = l_2[2]) ∧ ... ∧ (l_1[k-2] = l_2[k-2]) ∧ (l_1[k-1] < l_2[k-1]) then
(4)         c = l_1 ⋈ l_2; // join step: generate candidates
(5)         if has_infrequent_subset(c, L_{k-1}) then
(6)           delete c; // prune step: remove unfruitful candidate
(7)         else add c to C_k;
(8)       }
(9)   return C_k;

procedure has_infrequent_subset(c: candidate k-itemset;
          L_{k-1}: frequent (k-1)-itemsets); // use prior knowledge
(1)   for each (k-1)-subset s of c
(2)     if s ∉ L_{k-1} then
(3)       return TRUE;
(4)   return FALSE;
```

Classification vs. Prediction

- Classification
  - predicts categorical class labels
  - 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
  - Fraud detection

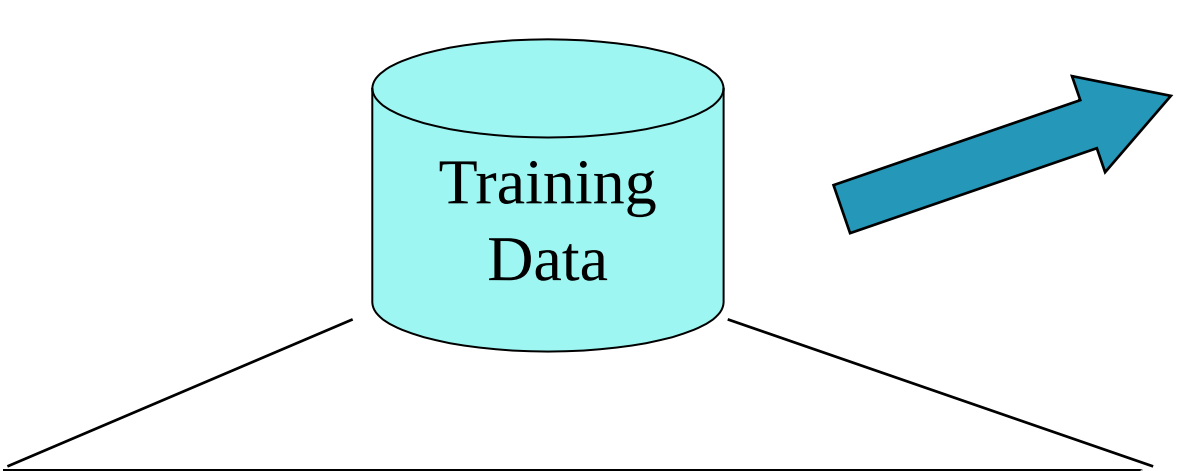
Supervised vs. Unsupervised Learning

- 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

Classification—A Two-Step Process

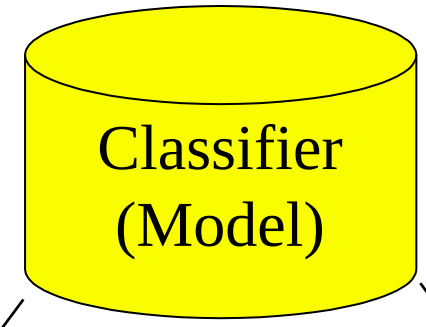
- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
  - The set of tuples used for model construction is training set
  - The model is represented as classification rules, decision 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
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

Process (1): Model Construction



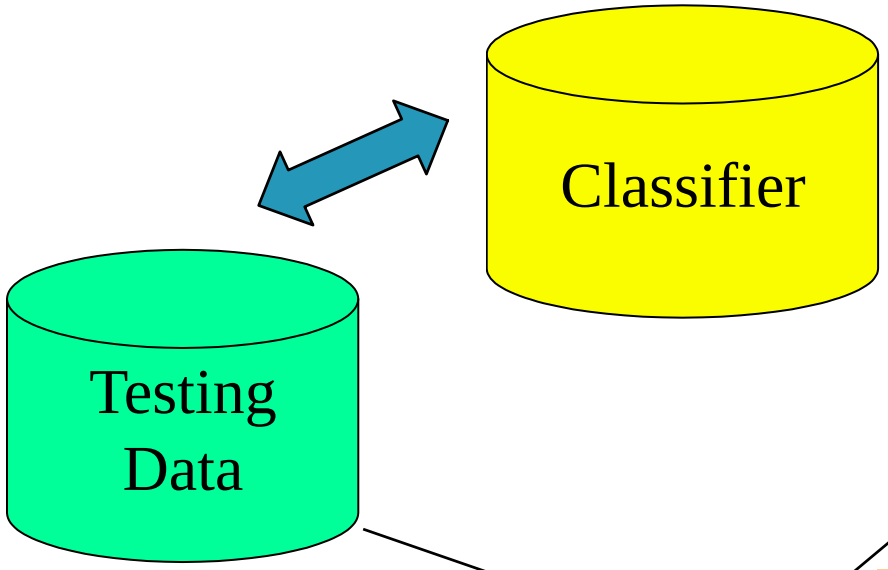
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

Classification Algorithms



IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Process (2): Using 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?

Yes

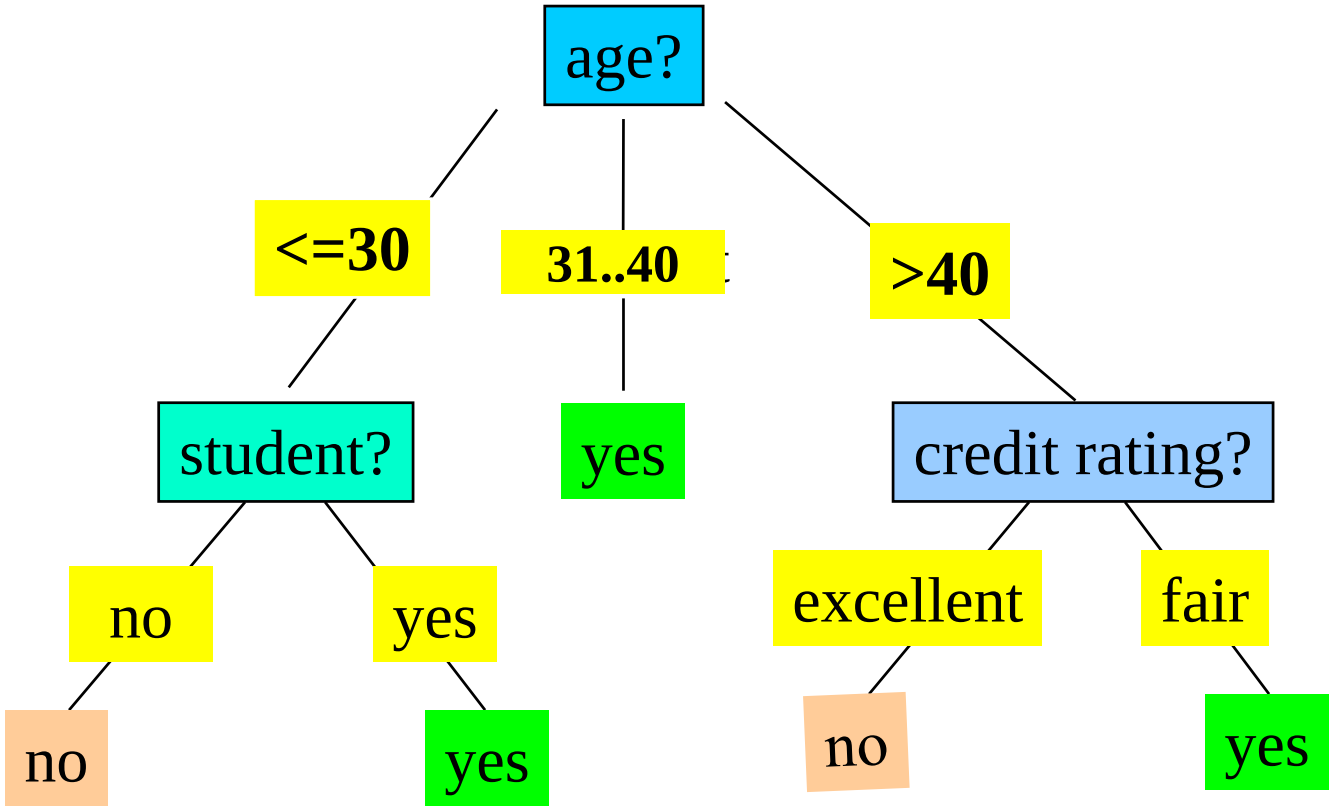
Decision Tree Induction: Training Dataset

Decision tree induction is the learning of decision tree from class-labeled training tuples.

This follows an example (Playing Tennis)

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

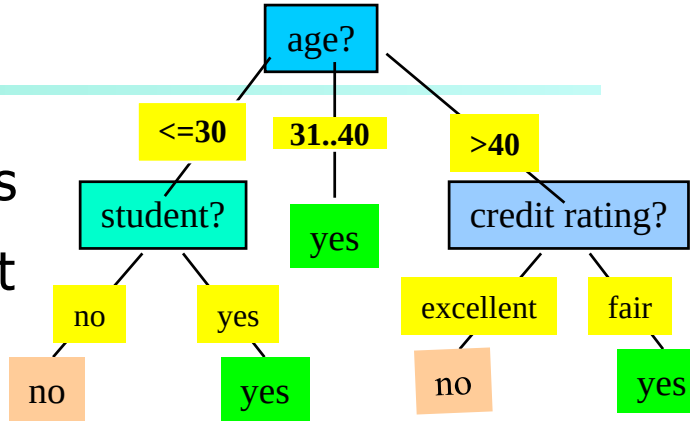
Output: A Decision Tree for "*buys_computer*"



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

Rule Extraction from a Decision Tree



- Rules are easier to understand than large trees
- 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 holds the class prediction
- Rules are mutually exclusive and exhaustive
- Example: Rule extraction from our *buys_computer* decision-tree

IF *age* = young AND *student* = *no* THEN *buys_computer* = *no*

IF *age* = young AND *student* = *yes* THEN *buys_computer* = *yes*

IF *age* = mid-age THEN *buys_computer* = *yes*

IF *age* = old AND *credit_rating* = *excellent* THEN *buys_computer* = *yes*

IF *age* = young AND *credit_rating* = *fair* THEN *buys_computer* = *no*

CSE5230 Tutorial: The ID3 Decision Tree Algorithm

MONASH UNIVERSITY

Faculty of Information Technology

CSE5230 Data Mining

Semester 2, 2004

The aim of this exercise is to learn how one famous algorithm for constructing decision trees, ID3, works. You will do this by building a decision tree by hand for a small dataset. At the end of this exercise you should understand how ID3 constructs a decision tree using the concept of *Information Gain*. You will be able to use the decision tree you create to make a decision about new data.

1 The weather data

The weather problem is a toy data set which we will use to understand how a decision tree is built. It comes from Quinlan (1986), a paper which discusses the ID3 algorithm introduced in Quinlan (1979). It is reproduced with slight modifications in Witten and Frank (1999), and concerns the conditions under which some hypothetical outdoor game may be played. The data is shown in Table 1.

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Table 1: The weather data (Witten and Frank; 1999, p. 9).

In this dataset, there are five categorical attributes *outlook*, *temperature*, *humidity*, *windy*, and *play*. We are interested in building a system which will enable us to decide whether or not to play the game on the basis of the weather conditions, *i.e.* we wish to predict the value of *play* using *outlook*, *temperature*,

humidity, and *windy*. We can think of the attribute we wish to predict, *i.e. play*, as the **output** attribute, and the other attributes as **input** attributes.

2 Building a decision tree using the ID3 algorithm

As we saw in lecture 6, a decision tree consists of nodes and arcs which connect nodes. To make a decision, one starts at the root node, and asks questions to determine which arc to follow, until one reaches a leaf node and the decision is made. This basic structure is shown in Figure 1.

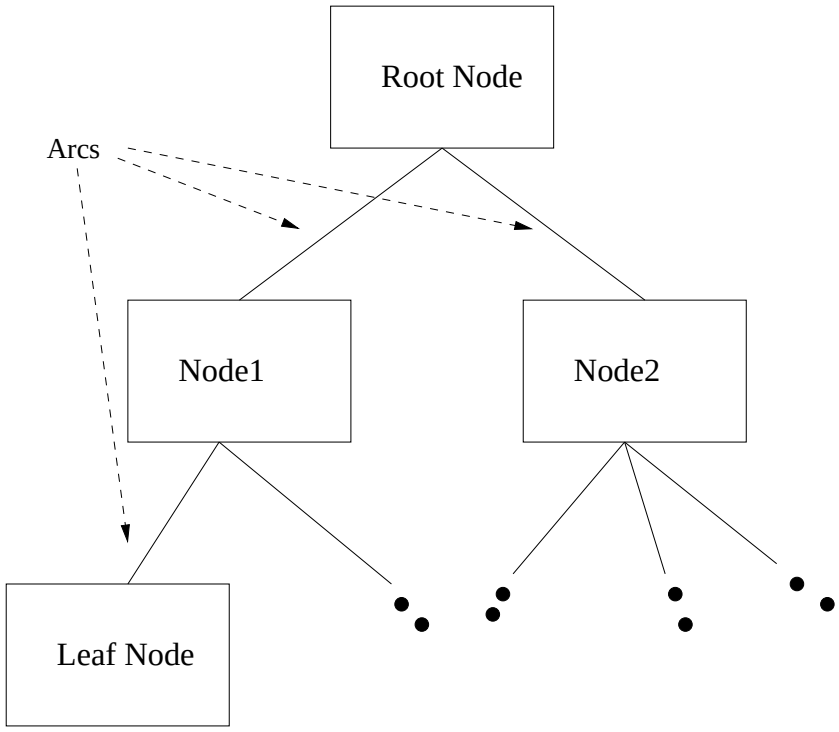


Figure 1: Basic decision tree structure

The main ideas behind the ID3 algorithm are:

- Each non-leaf node of a decision tree corresponds to an input attribute, and each arc to a possible value of that attribute. A leaf node corresponds to the expected value of the output attribute when the input attributes are described by the path from the root node to that leaf node.
- In a "good" decision tree, each non-leaf node should correspond to the input attribute which is the *most informative* about the output attribute amongst all the input attributes not yet considered in the path from the root node to that node. This is because we would like to predict the output attribute using the smallest possible number of questions on average.

- *Entropy* is used to determine how informative a particular input attribute is about the output attribute for a subset of the training data. Entropy is a measure of uncertainty in communication systems introduced by Shannon (1948). It is fundamental in modern information theory.

2.1 Entropy

In information theory, entropy is a measure of the uncertainty about a source of messages. The more uncertain a receiver is about a source of messages, the more information that receiver will need in order to know what message has been sent.
For example, if a message source always sends exactly the same message, the receiver does not need any information to know what message has been sent—it's always the same! The entropy of such a source is zero: there is no uncertainty at all. On the other hand, if a source can send n possible messages and each message occurs independently of the preceding message with equal probability, then the uncertainty of the receiver is maximised. The receiver will need to ask $\log_2 n$ yes/no questions to determine which message has been sent, *i.e.* the receiver needs $\log_2 n$ *bits* of information.

Question 1 Why does the receiver need $\log_2 n$ bits? Think about representing each message as a binary number.

The average number of bits required to identify each message is a measure of the receiver's uncertainty about the source, and is known as the entropy of the source.
Consider a source S which can produce n messages $\{m_1, m_2, \ldots, m_n\}$. All messages are produced independently of each other, and the probability of producing message m_i is p_i. For such a source S with a message probability distribution $P = (p_1, p_2, \ldots, p_n)$, the entropy $H(P)$ is

$$H(P) = -\sum_{i=1}^{n} p_i \log(p_i). \tag{1}$$

In the equation above, $\log(p_i)$ means $\log_2(p_i)$—from now on we will assume that all logarithms are to the base two.
If a set T of records from a database (*i.e.* the training set for building the decision tree) is partitioned into k classes $\{C_1, C_2, \ldots, C_k\}$ on the basis of the output attribute, then the average amount of information (measured in bits) needed to identify the class of a record is $H(P_T)$, where P_T is the probability distribution of the classes, estimated from the data as

$$P_T = \left(\frac{|C_1|}{|T|}, \frac{|C_2|}{|T|}, \ldots, \frac{|C_k|}{|T|} \right). \tag{2}$$

The notation $|C_i|$ means the number of elements in set C_i. For the weather data, where *play* is the output attribute, we have for the entire dataset T

$$P_T = \left(\frac{5}{14}, \frac{9}{14} \right),$$

where class C_1 corresponds to "no", and class C_2 to "yes". Using Equation 1, the entropy of set T is thus

$$\begin{aligned} H(T) = H(P_T) &= -\left(\frac{5}{14} \log(\frac{5}{14}) + \frac{9}{14} \log(\frac{9}{14}) \right) \\ &= 0.940. \end{aligned}$$

Note that here we identify the entropy of the set $H(T)$ with the entropy of the probability distribution of the members of the set, $H(P_T)$.

2.2 Information gain

Now consider what happens if we partition the set on the basis of an input attribute X into subsets $T_1, T_2, \ldots, T_n$. The information needed to identify the class of an element of T is the weighted average of the information needed to identify the class of an element of each subset:

$$H(X,T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} H(T_i). \tag{3}$$

In the context of building a decision tree, we are interested in how much information about the output attribute can be gained by knowing the value of an input attribute X. This is just the difference between the information needed to classify an element of T before knowing the value of X, $H(T)$, and the information needed after partitioning the dataset T on the basis of knowing the value of X, $H(X,T)$. We define the *information gain* due to attribute X for set T as[1]

$$\mathrm{Gain}(X,T) = H(T) - H(X,T). \tag{4}$$

In order to decide which attribute to split upon, the ID3 algorithm computes the information gain for each attribute, and selects the one with the highest gain.
As an example, let us consider the information gain for attribute *temperature* in the weather data. The attribute *temperature* can have three values, *cool*, *mild* and *hot*. Partitioning on the basis of temperature will thus divide T into three subsets. This reorganisation of the weather data is shown in Table 2. We have $|T_{cool}| = 4$, $|T_{mild}| = 6$ and $|T_{hot}| = 4$. We can calculate the information needed to classify an element of T after this partitioning using Equation 3:

$$\begin{aligned}
H(temperature,T) &= \frac{4}{14}H(T_{cool}) + \frac{6}{14}H(T_{mild}) + \frac{4}{14}H(T_{hot}) \\
&= \frac{4}{14}\left(-\left(\frac{1}{4}\log(\frac{1}{4}) + \frac{3}{4}\log(\frac{3}{4})\right)\right) + \frac{6}{14}\left(-\left(\frac{2}{6}\log(\frac{2}{6}) + \frac{4}{6}\log(\frac{4}{6})\right)\right) + \\
&\quad \frac{4}{14}\left(-\left(\frac{2}{4}\log(\frac{2}{4}) + \frac{2}{4}\log(\frac{2}{4})\right)\right) \\
&= \frac{4}{14} \times 0.811 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 1.00 \\
&= 0.911
\end{aligned}$$

The information gain for attribute *temperature* for set T is thus

$$\begin{aligned}
\mathrm{Gain}(temperature,T) &= 0.940 - 0.911 \\
&= 0.029 \text{ bits.}
\end{aligned}$$

[1] Note that in information theory $\mathrm{Gain}(X,T)$ is known as the *mutual information* between the input attribute X and the output attribute.

outlook	temperature	humidity	windy	play
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	cool	normal	false	yes
rainy	mild	high	false	yes
sunny	mild	high	false	no
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
rainy	mild	high	true	no
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
overcast	hot	normal	false	yes

Table 2: The weather data partitioned on the basis of attribute *temperature*.

Question 2 Calculate the information gain for the other input attributes, *outlook*, *humidity*, and *windy*. You may use whatever tools you like to do the calculation. Note the following useful identity:

$$\log_a(x) = \frac{\log_b(x)}{\log_b(a)}.$$

An *entropy calculator* has been created to help you with this, and subsequent, questions. You can find it at:
`http://www.csse.monash.edu.au/~davids/cgi-bin/cse5230/entropy_calc.cgi`.

Question 3 Given your answer to the question above, what attribute would ID3 select to split upon for set T, *i.e.* the complete weather data from Table 1?

2.3 The ID3 algorithm

The ID3 algorithm works by recursively applying the procedure above to each of the subsets produced until "pure" nodes are found—a pure node contains elements of only one class—or until there are no attributes left to consider. It can be stated in pseudocode, as is shown in Figure 2.

Question 4 Use the ID3 algorithm given in Figure 2 to construct a decision tree for the weather data by hand. The first call to `ID3()` uses the entire set of input attributes and the entire set of training data.

Question 5 Consider a new data element with input attributes $\{overcast, cool, high, true\}$. How is it classified by the tree you constructed above?

```
function ID3 (I, 0, T) {
/*  I is the set of input attributes
*   O is the output attribute
*   T is a set of training data
*
*   function ID3 returns a decision tree
*/
    if (T is empty) {
        return a single node with the value "Failure";
    }
    if (all records in T have the same value for O) {
        return a single node with that value;
    }
    if (I is empty) {
        return a single node with the value of the most frequent value of
        O in T;
        /* Note: some elements in this node will be incorrectly classified */
    }

    /* now handle the case where we can't return a single node */
    compute the information gain for each attribute in I relative to T;
    let X be the attribute with largest Gain(X, T) of the attributes in I;
    let {x_j| j=1,2, .., m} be the values of X;
    let {T_j| j=1,2, .., m} be the subsets of T when T is partitioned
        according the value of X;
        return a tree with the root node labelled X and
        arcs labelled x_1, x_2, .., x_m, where the arcs go to the
        trees ID3(I-{X}, O, T_1), ID3(I-{X}, O, T_2), .., ID3(I-{X}, O, T_m);
}
```

Figure 2: The ID3 algorithm

2.4 The problem of attributes with many values

The simple ID3 algorithm above can have difficulties when an input attribute has many possible values, because $\mathrm{Gain}(X, T)$ tends to favour attributes which have a large number of values. It is easy to understand why if we consider an extreme case.

Imagine that our dataset T contains an attribute that has a different value for every element of T. This could arise in practice if a unique record ID was retained when extracting T from a database—for example a patient ID number in a hospital database. Such an attribute would give the maximum possible information gain, since all the training data can be correctly classified by examining its value. It would result in a decision tree in which all nodes below the root node were leaf nodes. This tree, however, would be completely useless for classifying new data: there would be no arc corresponding to the value

of the ID attribute. Moreover, there is no reason to suspect that such an attribute would have any causal relationship with the output attribute.
The problem also arises when an attribute can take on many values, even if they are not unique to each element. Quinlan (1986) suggests a solution based on considering the amount of information required to determine the value of an attribute X for a set T. This is given by $H(P_{X,T})$, where $P_{X,T}$ is the probability distribution of the values of X:

$$P_{X,T} = \left(\frac{|T_1|}{|T|}, \frac{|T_2|}{|T|}, \ldots, \frac{|T_n|}{|T|} \right). \tag{5}$$

The quantity $H(P_{X,T})$ is known as the *split information* for attribute X and set T. We will call it $\mathrm{SplitInfo}(X,T) = H(P_{X,T})$. For the weather data, the split information for the attribute *temperature* is

$$\begin{aligned} \mathrm{SplitInfo}(temperature, T) &= -\left(\frac{4}{14}\log(\frac{4}{14}) + \frac{6}{14}\log(\frac{6}{14}) + \frac{4}{14}\log(\frac{4}{14}) \right) \\ &= 1.56 \text{ bits.} \end{aligned}$$

Quinlan (1986) suggests that rather than choosing the attribute with the biggest $\mathrm{Gain}(X,T)$, we select the one with the biggest $\mathrm{GainRatio}(X,T)$, where

$$\mathrm{GainRatio}(X,T) = \frac{\mathrm{Gain}(X,T)}{\mathrm{SplitInfo}(X,T)}. \tag{6}$$

Note that $\mathrm{GainRatio}(X,T)$ might not always be defined. Quinlan (1986) specifies a *gain ratio criterion*, which says that we should select the attribute X with the highest $\mathrm{GainRatio}(X,T)$ from amongst those attributes with average-or-better $\mathrm{Gain}(X,T)$.
If you have time, attempt the following questions.

Question 6 Under what circumstances is $\mathrm{SplitInfo}(X,T)$ equal to 0?

Question 7 What is $\mathrm{GainRatio}(temperature, T)$?

Question 8 What are the gain ratios for the other input attributes?

Question 9 Construct a decision tree for the weather data using the gain ratio criterion rather than the information gain.

References

Quinlan, J. R. (1979). Discovering rules by induction from large collections of examples, *in* D. Michie (ed.), *Expert Systems in the Micro Electronic Age*, Edinburgh University Press, pp. 168–201.

Quinlan, J. R. (1986). Induction of decision trees, *Machine Learning* **1**(1): 81–106.

Shannon, C. E. (1948). A mathematical theory of communication, *Bell System Technical Journal* **27**: 379–423 and 623–656.
`http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html`

Witten, I. H. and Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, San Francisco, CA, USA.
`http://www.cs.waikato.ac.nz/~ml/weka/book.html`

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
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters
- Unsupervised learning: no predefined classes
- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Examples of Clustering Applications

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- 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

Quality: 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

- Ability to deal with different types of attributes
- Ability to handle dynamic data
- 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
- Incorporation of user-specified constraints
- Interpretability and usability

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*:

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

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

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

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

Similarity and Dissimilarity Between Objects (Cont.)

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

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

  - Both eculidean and manhattan distance satisfy the following mathematic requirements of a distance function
  - $d(i,j) \geq 0$:Distance is a non negative number
    - $d(i,i) = 0$: The distance of an object to itself is 0
    - *$d(i,j) = d(j,i)$: Distance is symmetric function*
    - *$d(i,j) \leq d(i,k) + d(k,j)$: Going directly from objet I to objet j in sapce is no more than making a detour over any other object k(triangular inequality)*

Major Clustering Approaches (I)

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: k-means, k-medoids, CLARANS
- Hierarchical approach:
  - Create a hierarchical decomposition of the set of data (or objects) using some criterion
  - Typical methods: Diana, Agnes, BIRCH, ROCK, CAMELEON

Partitioning Algorithms: Basic Concept

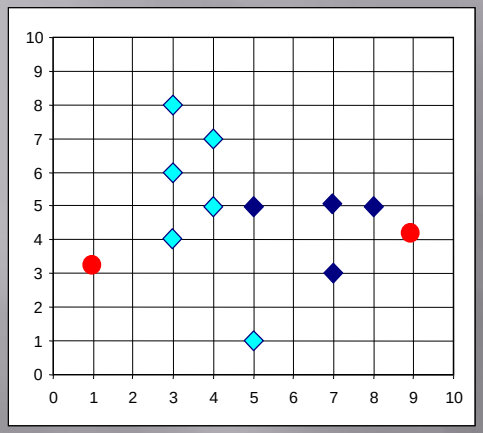
- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters.
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means*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

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

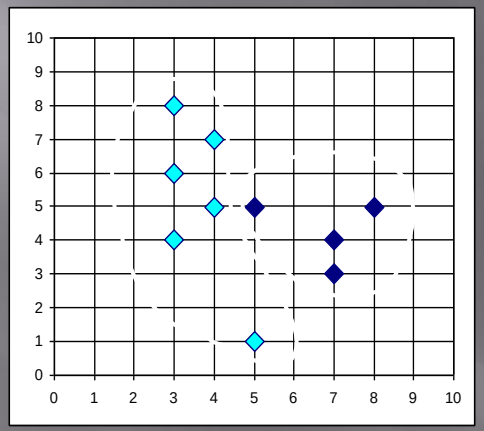
- Example



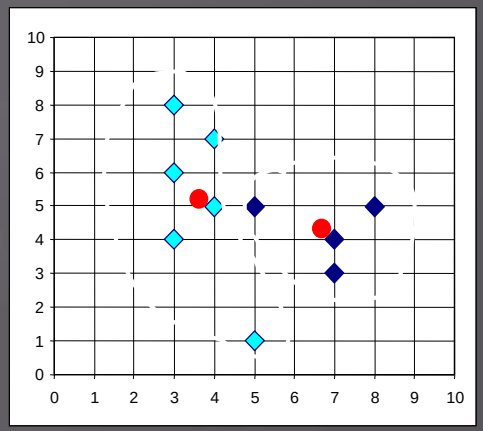
K=2

Arbitrarily choose K object as initial cluster center

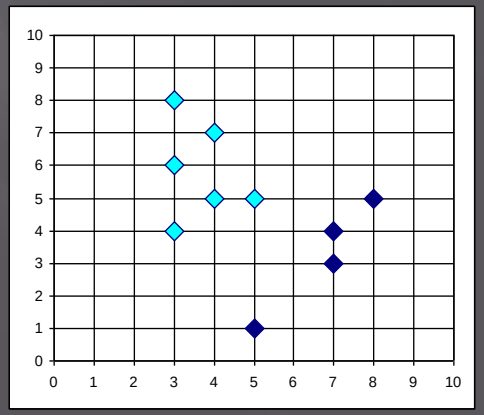
Assign each objects to most similar center



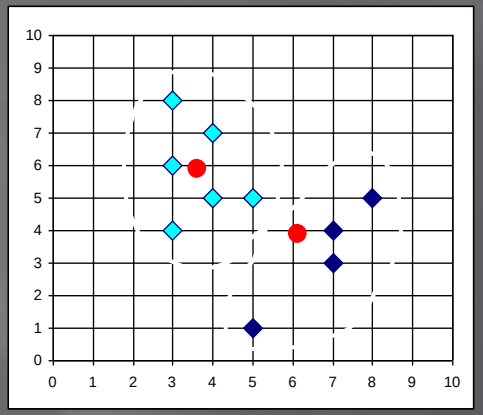
Update the cluster means



reassign



Update the cluster means



reassign

- Each object is distributed to a cluster center to which it is nearest.
- Next the cluster centers are updated. That is, mean value of each cluster is recalculated based on the current objects in the cluster. Using the new cluster centers, the objects are redistributed to the clusters based on which cluster center is the nearest.
- This process iterates. The process of iteratively reassigning objects to clusters to improve the partitioning is referred to as iterative relocation.
- When no redistribution of objects in any clusters occur, the process terminates.

Input:

k:number of clusters

D: a dataset containing n objects

Output: a set of k clusters

Method

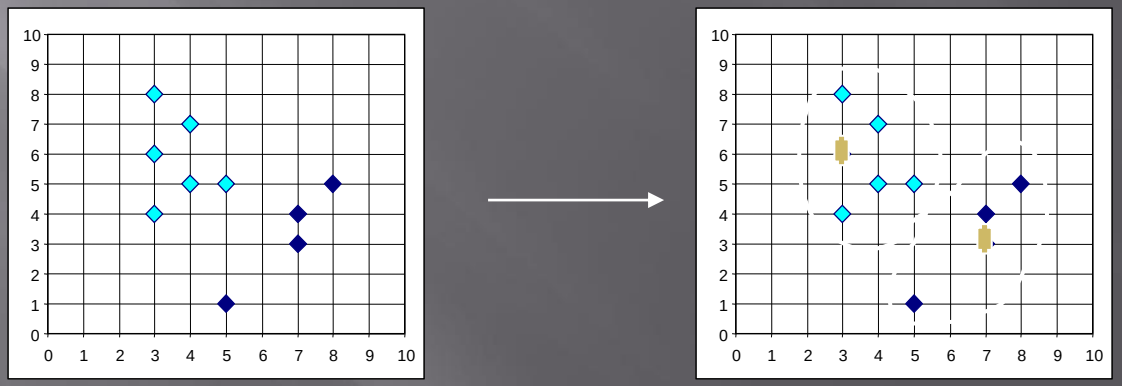
- 1.Arbitrarily choose k objects from D as the intial centers
- 2. reappear
- 3. (re) assign each object to the cluster to which the object is the most similar,based on the mean value of the objects in the cluster
- 4. Update the cluster means, i .e., calculate the mean value of the objects for each cluster
- 5. until no change

Comments on the *K-Means* Method

- 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*

What Is the Problem of the K-Means Method?

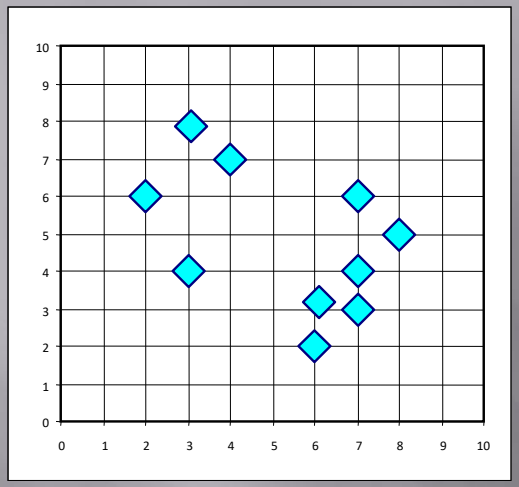
- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



The *K-Medoids* Clustering Method

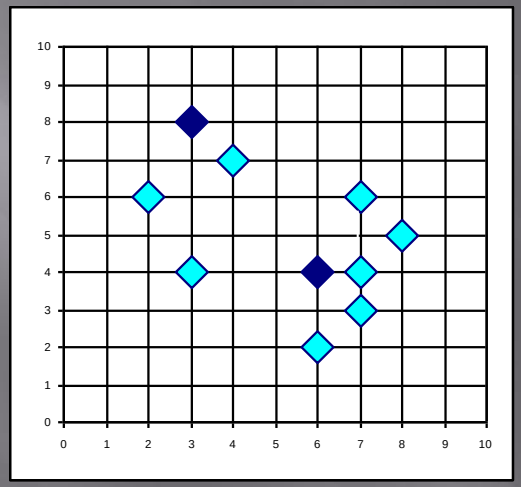
- Find *representative* objects, called medoids, in clusters
- *PAM* (Partitioning Around Medoids)
  - starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
  - *PAM* works effectively for small data sets, but does not scale well for large data sets.

A Typical K-Medoids Algorithm (PAM)



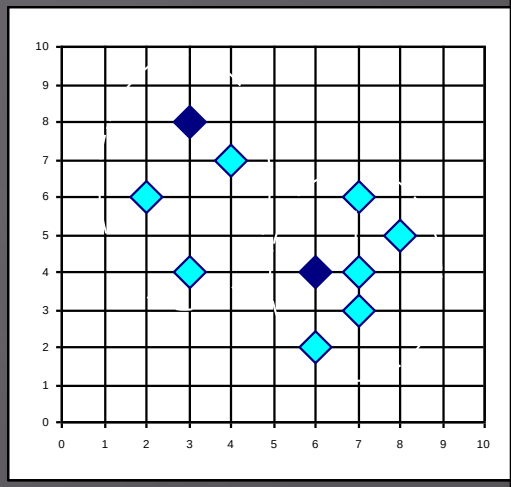
K=2

Arbitrary choose k object as initial medoids

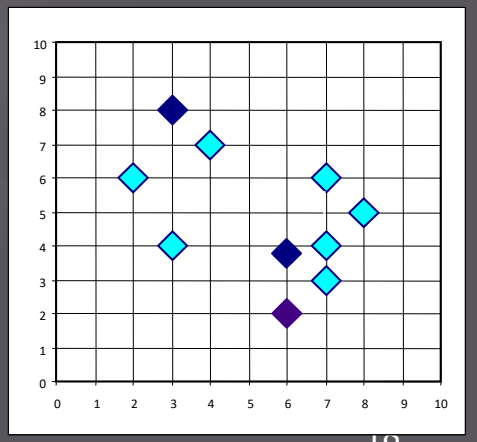


Assign each remaining object to nearest medoids

Total Cost = 20

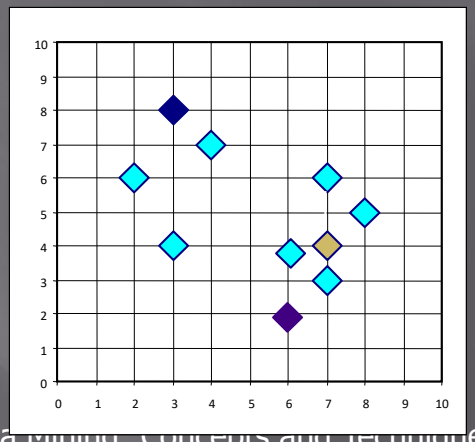


Randomly select a nonmedoid object, O_{ramdom}



Compute total cost of swapping

Total Cost



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

Algorithm: K-medoids. PAM, a k-medoid algorithm for partitioning based on medoid or central objects.

Input:

- k: the number of clusters,
- D: a data set containing n objects.

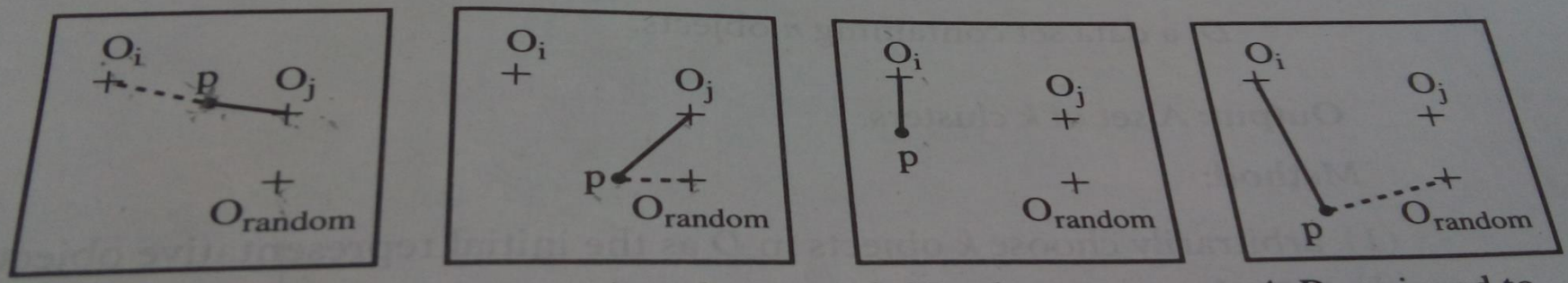
Output: A set of k clusters.

Method:

(1) arbitrarily choose k objects in D as the initial representative objects or seeds;
(2) **repeat**
(3) assign each remaining object to the cluster with the nearest representative object;
(4) randomly select a nonrepresentative object, o_{random};
(5) compute the total cost, S, of swapping representative object, o_j, with o_{random};
(6) if $S < 0$ **then** swap o_j with o_{random} to form the new set of k representative objects;
(7) **until** no change;

Four cases of cost function for k-medoid clustering

- **Case 1:** p currently belongs to representative object, o_j. If o_j is replaced by o_{random} as a representative object and p is closest to one of the other representative objects, o_i, $i \neq j$, then p is reassigned to o_i.
- **Case 2:** p currently belongs to representative object, o_j. If o_j is replaced by o_{random} as a representative object and p is closest to o_{random}, then p is reassigned to o_{random}.
- **Case 3:** p currently belongs to representative object, o_i, $i \neq j$. If o_j is replaced by o_{random} as a representative object and p is still closest to o_i, then the assignment does not change.



1. Reassigned to O_i
2. Reassigned to O_{random}
3. No change
4. Reassigned to O_{random}

- data object
+ cluster center
— before swapping
--- after swapping

Case 4: p currently belongs to representative object, o_i, $i \neq j$. If o_j is replaced by o_{random} as a representative object and p is closest to o_{random}, then p is reassigned to o_{random}.

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.

➔ Sampling based method,

CLARA(Clustering LARge Applications)

CLARA (Clustering Large Applications)

- 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.

Mining Complex Types of Data

- Mining Sequence Data
  - Mining Time Series
  - Mining Symbolic Sequences
  - Mining Biological Sequences
- Mining Graphs and Networks
- Mining Other Kinds of Data

Mining Sequence Data

- Similarity Search in Time Series Data
  - Subsequence match, dimensionality reduction, query-based similarity search, motif-based similarity search
- Regression and Trend Analysis in Time-Series Data
  - long term + cyclic + seasonal variation + random movements
- Sequential Pattern Mining in Symbolic Sequences
  - GSP, PrefixSpan, constraint-based sequential pattern mining
- Sequence Classification
  - Feature-based vs. sequence-distance-based vs. model-based

Mining Graphs and Networks

- Graph Pattern Mining
  - Frequent subgraph patterns, closed graph patterns, gSpan vs. CloseGraph
- Statistical Modeling of Networks
  - Small world phenomenon, power law (log-tail) distribution, densification
- Clustering and Classification of Graphs and Homogeneous Networks
  - Clustering: Fast Modularity vs. SCAN
  - Classification: model vs. pattern-based mining
- Clustering, Ranking and Classification of Heterogeneous Networks
  - RankClus, RankClass, and meta path-based, user-guided methodology
- Role Discovery and Link Prediction in Information Networks
  - PathPredict
- Similarity Search and OLAP in Information Networks: PathSim, GraphCube
- Evolution of Social and Information Networks: EvoNetClus

Mining Other Kinds of Data

- Mining Spatial Data
  - Spatial frequent/co-located patterns, spatial clustering and classification
- Mining Spatiotemporal and Moving Object Data
  - Spatiotemporal data mining, trajectory mining, periodica, swarm, …
- Mining Cyber-Physical System Data
  - Applications: healthcare, air-traffic control, flood simulation
- Mining Multimedia Data
  - Social media data, geo-tagged spatial clustering, periodicity analysis, …
- Mining Text Data
  - Topic modeling, i-topic model, integration with geo- and networked data
- Mining Web Data
  - Web content, web structure, and web usage mining
- Mining Data Streams
  - Dynamics, one-pass, patterns, clustering, classification, outlier detection

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, PC_games*} generalizes to {*sports, music, e_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 only those 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
  - 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

Ex.: Plan Mining by Divide and Conquer

- Plan: a 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 plan 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	…
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

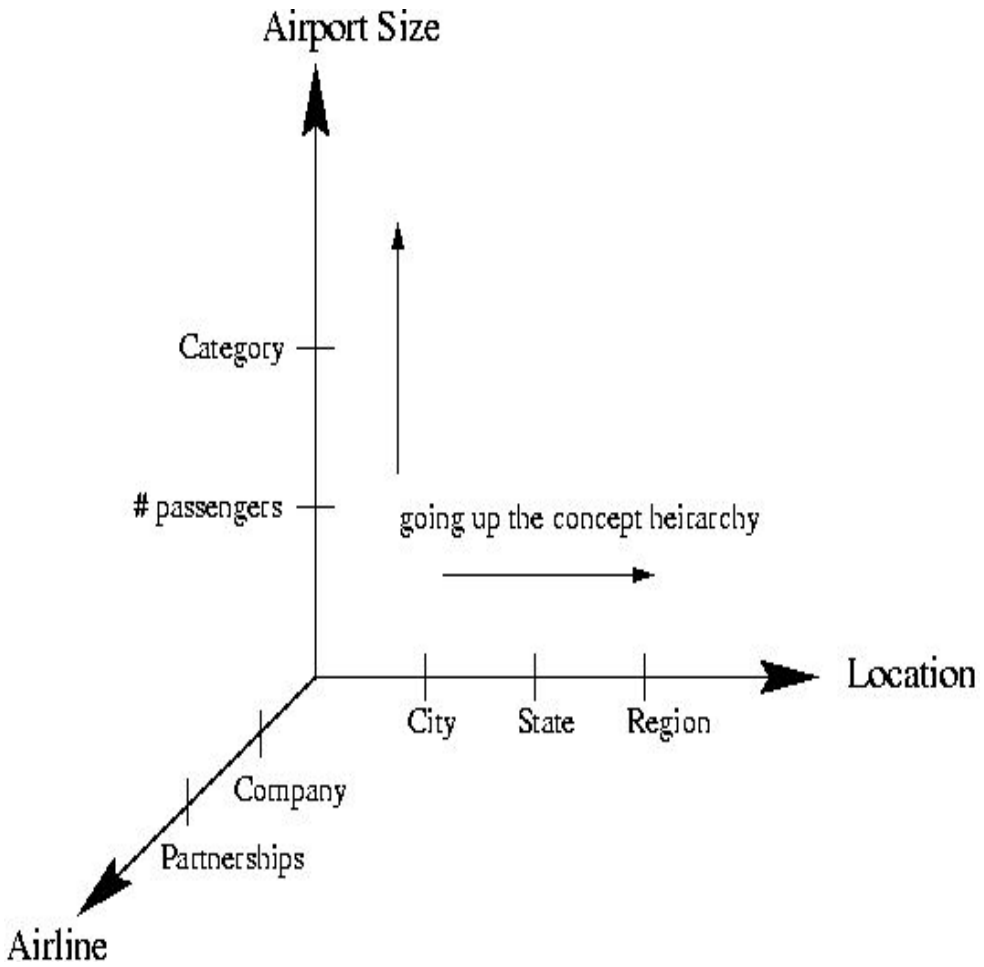
Airport info table

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-Dimensional 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+	…
.		.		.
.		.		.
.		.		.

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

What Is a Spatial Database System?

- Geometric, geographic or spatial data: space-related data
  - Example: Geographic space (2-D abstraction of earth surface), VLSI design, model of human brain, 3-D space representing the arrangement of chains of protein molecule.
- Spatial database system vs. image database systems.
  - Image database system: handling digital raster image (e.g., satellite sensing, computer tomography), may also contain techniques for object analysis and extraction from images and some spatial database functionality.
  - Spatial (geometric, geographic) database system: handling objects in space that have identity and well-defined extents, locations, and relationships.

GIS (Geographic Information System)

- GIS (Geographic Information System)
  - Analysis and visualization of geographic data
- Common analysis functions of GIS
  - Search (thematic search, search by region)
  - Location analysis (buffer, corridor, overlay)
  - Terrain analysis (slope/aspect, drainage network)
  - Flow analysis (connectivity, shortest path)
  - Distribution (nearest neighbor, proximity, change detection)
  - Spatial analysis/statistics (pattern, centrality, similarity, topology)
  - Measurements (distance, perimeter, shape, adjacency, direction)

Spatial DBMS (SDBMS)

- SDBMS is a software system that
  - supports spatial data models, spatial ADTs, and a query language supporting them
  - supports spatial indexing, spatial operations efficiently, and query optimization
  - can work with an underlying DBMS
- Examples
  - Oracle Spatial Data Catridge
  - ESRI Spatial Data Engine

Modeling Spatial Objects

- What needs to be represented?
- Two important alternative views
  - Single objects: distinct entities arranged in space each of which has its own geometric description
    - modeling cities, forests, rivers
  - Spatially related collection of objects: describe space itself (about every point in space)
    - modeling land use, partition of a country into districts

Modeling Single Objects: Point, Line and Region

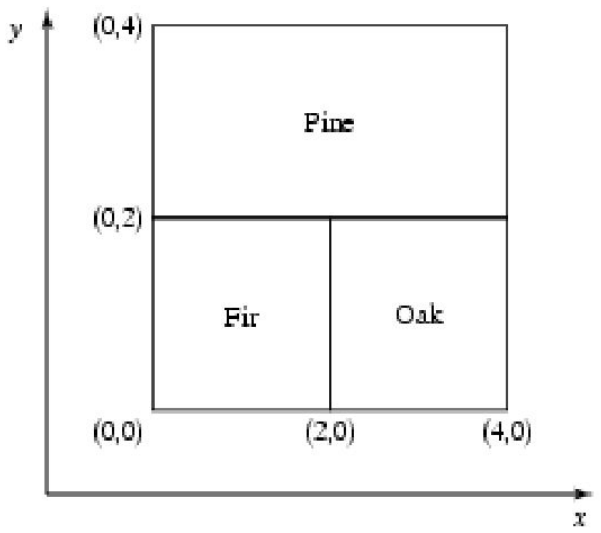
- Point: location only but not extent
- Line (or a curve usually represented by a polyline, a sequence of line segment):
  - moving through space, or connections in space (roads, rivers, cables, etc.)
- Region:
  - Something having extent in 2D-space (country, lake, park). It may have a hole or consist of several disjoint pieces.

Modeling Spatially Related Collection of Objects

- Modeling spatially related collection of objects: plane partitions and networks.
  - A partition: a set of region objects that are required to be disjoint (e.g., a thematic map). There exist often pairs of objects with a common boundary (adjacency relationship).
  - A network: a graph embedded into the plane, consisting of a set of point objects, forming its nodes, and a set of line objects describing the geometry of the edges, e.g., highways. rivers, power supply lines.
  - Other interested spatially related collection of objects: nested partitions, or a digital terrain (elevation) model.

Spatial Data Types and Models

- Field-based model: raster data
  - framework: partitioning of space
- Object-based model: vector model
  - point, line, polygon, Objects, Attributes



(a)

Object Viewpoint of Forest Stands

Area-ID	Dominant Tree Species	Area/Boundary
FS1	Pine	[(0,2),(4,2),(4,4),(0,4)]
FS2	Fir	[(0,0),(2,0),(2,2),(0,2)]
FS3	Oak	[(2,0),(4,0),(4,2),(2,2)]

(b)

Field Viewpoint of Forest Stands

$$f(x,y) = \begin{cases} \text{"Pine,"} & 2 \leq x \leq 4\,;\, 2 < y \leq 4 \\ \text{"Fir,"} & 0 \leq x \leq 2;\, 0 \leq y \leq 2 \\ \text{"Oak,"} & 2 < x \leq 4;\, 0 \leq y \leq 2 \end{cases}$$

(c)

Spatial Data Warehousing

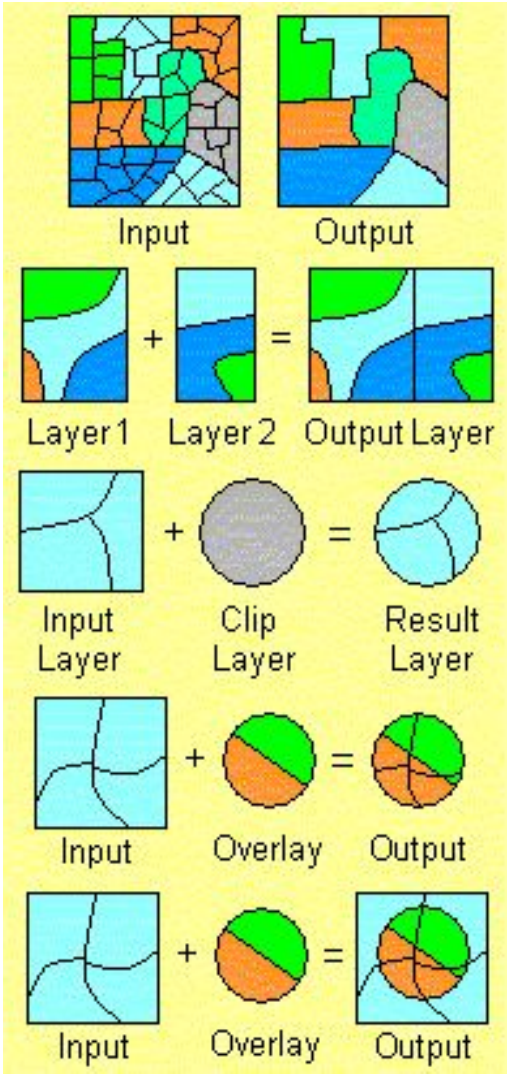
- Spatial data warehouse: Integrated, subject-oriented, time-variant, and nonvolatile spatial data repository
- Spatial data integration: a big issue
  - Structure-specific formats (raster- vs. vector-based, OO vs. relational models, different storage and indexing, etc.)
  - Vendor-specific formats (ESRI, 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

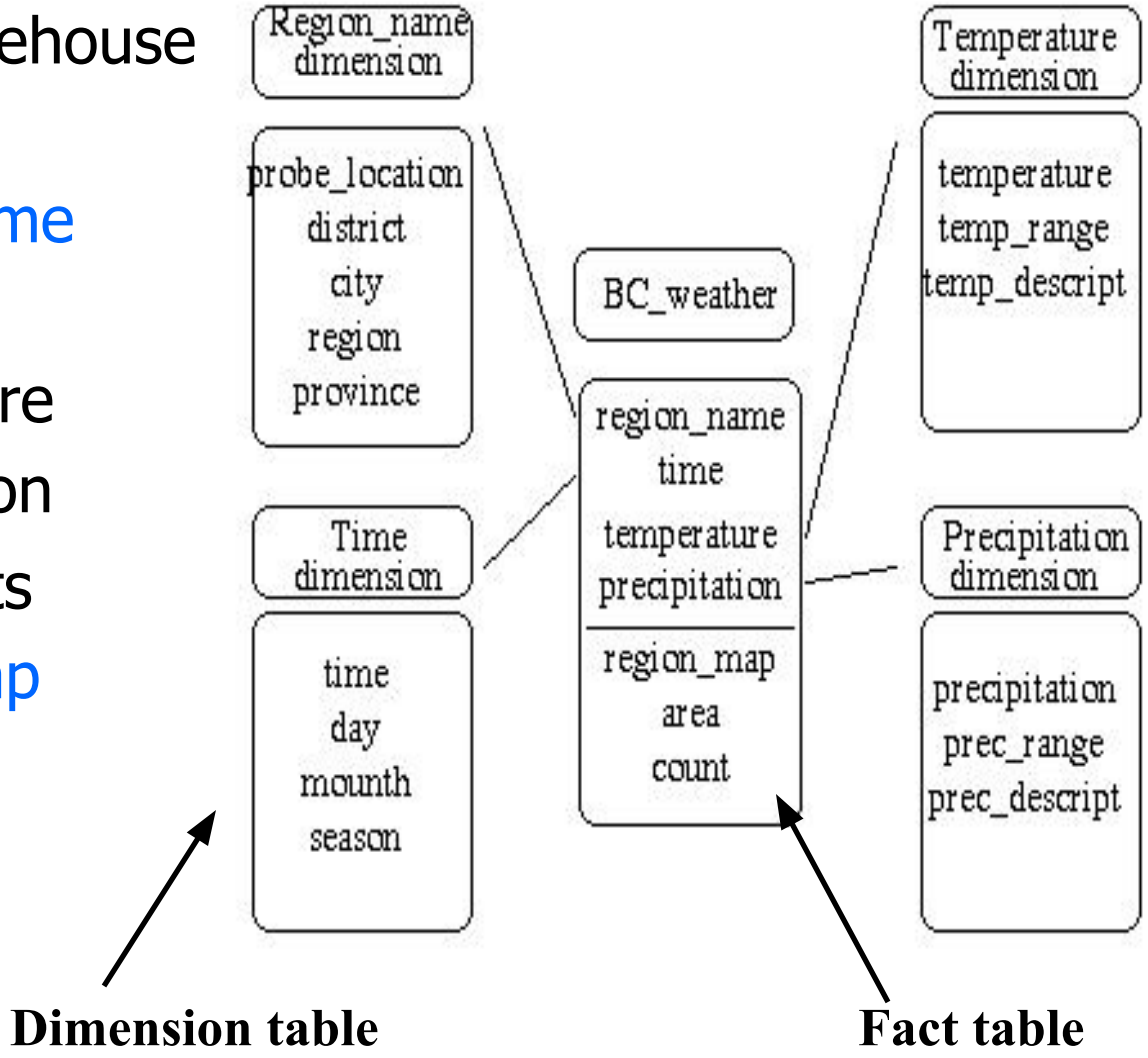


Example: British Columbia Weather Pattern Analysis

- 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

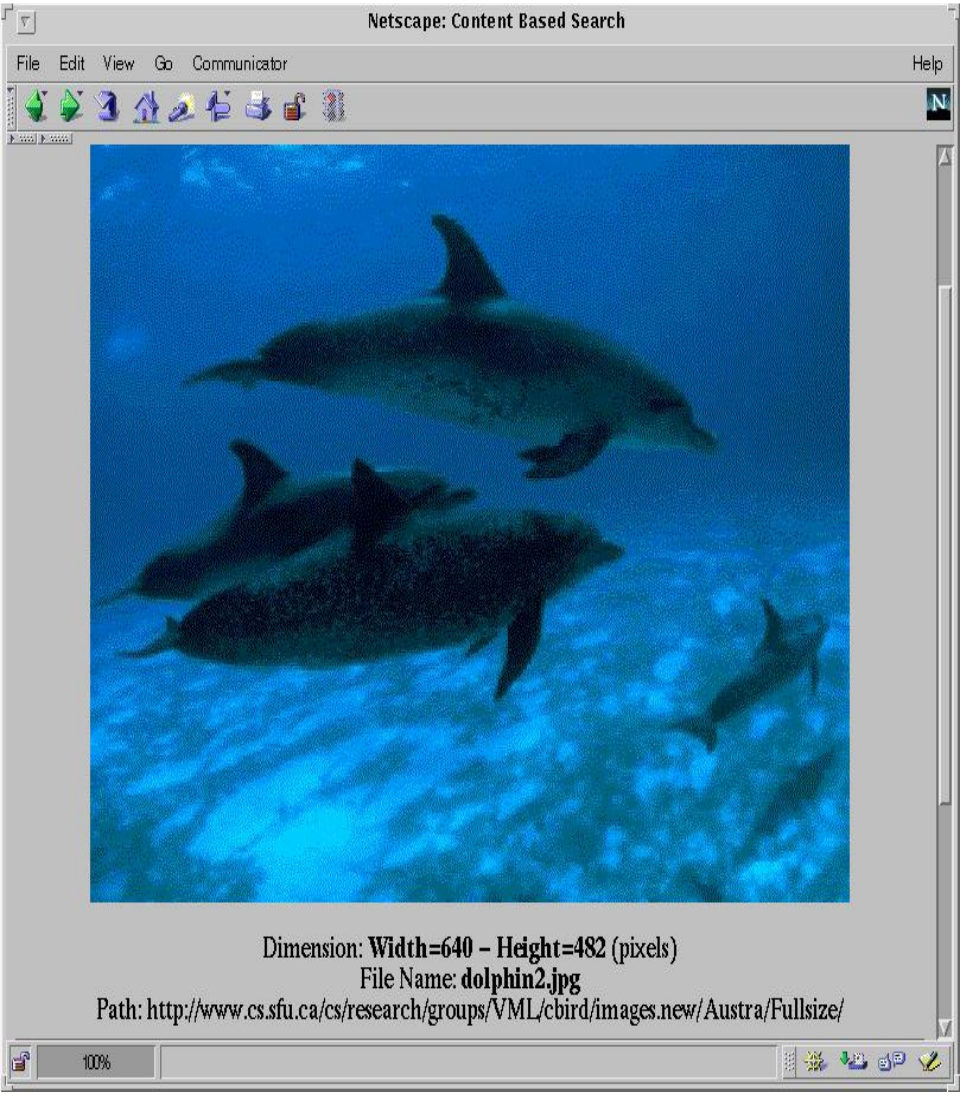
- Spatial data warehouse
  - Dimensions
    - region_name
    - time
    - temperature
    - precipitation
  - Measurements
    - region_map
    - area
    - count

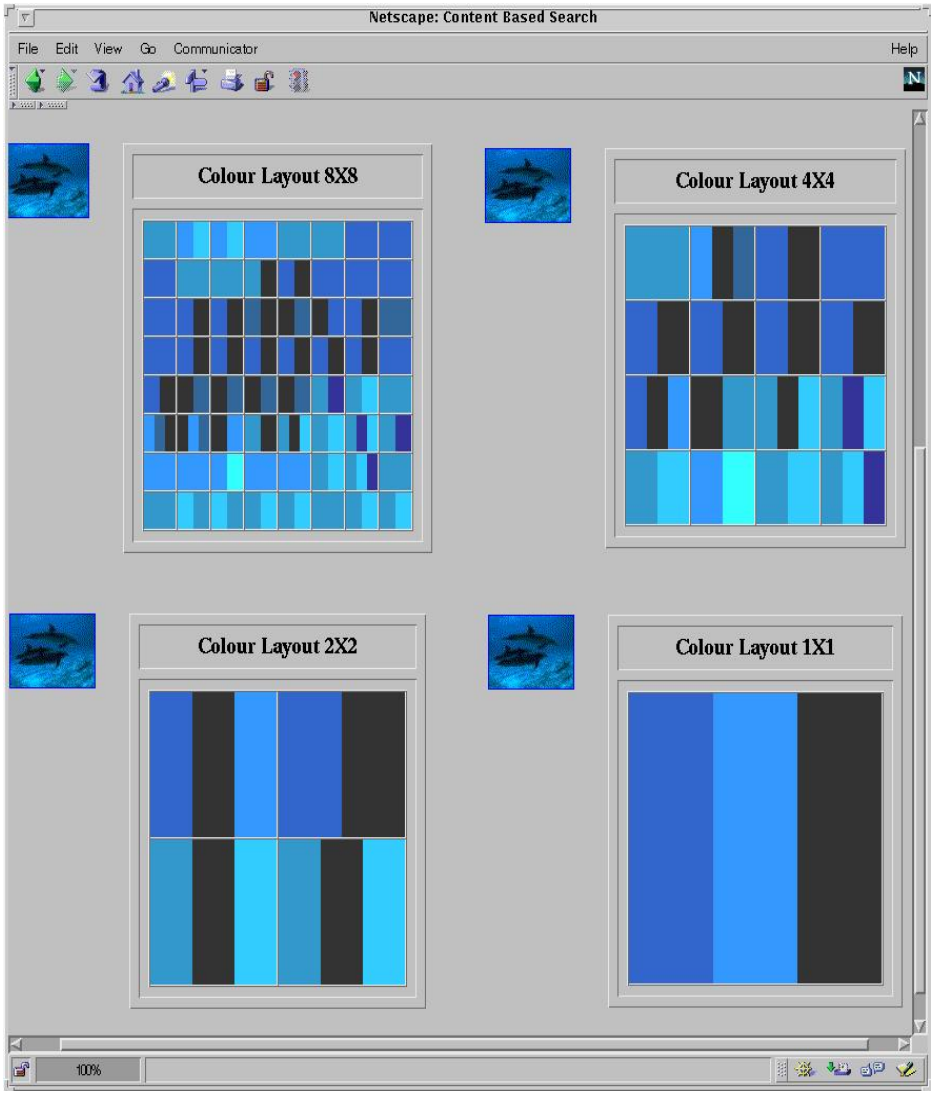


Multidimensional Analysis of Multimedia Data

- 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

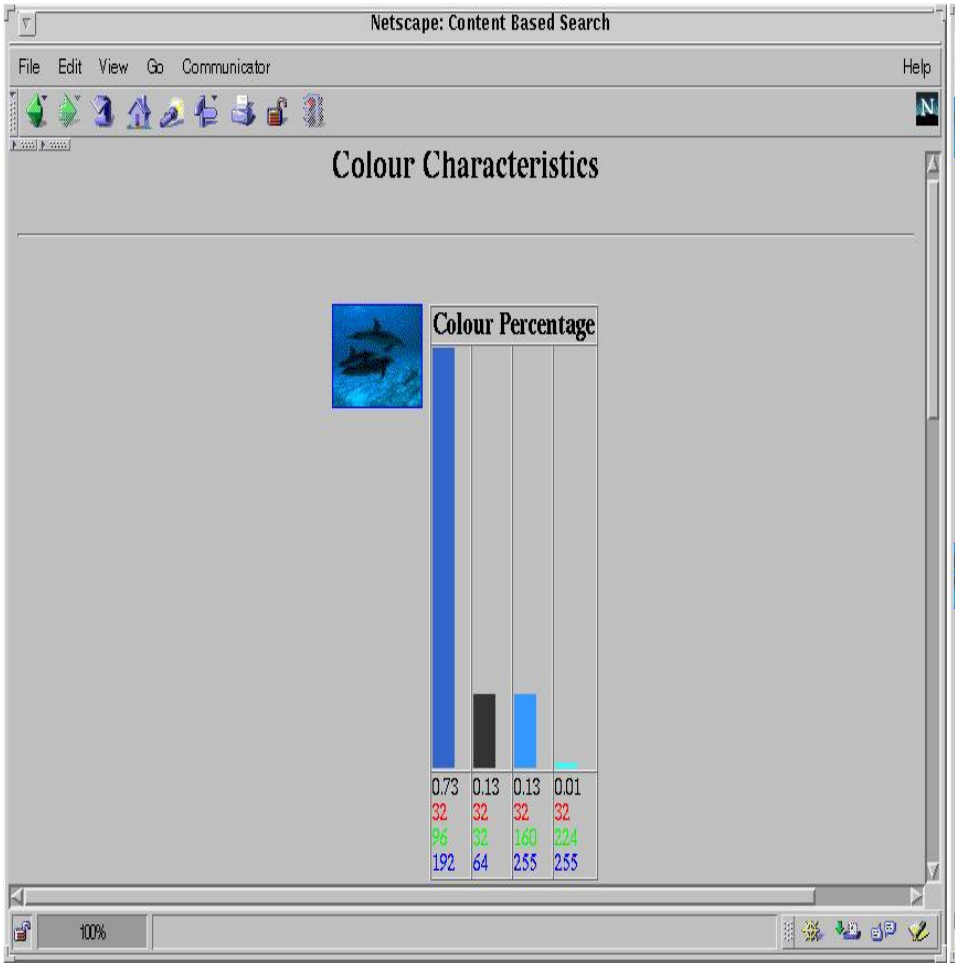
Multi-Dimensional Search in Multimedia Databases



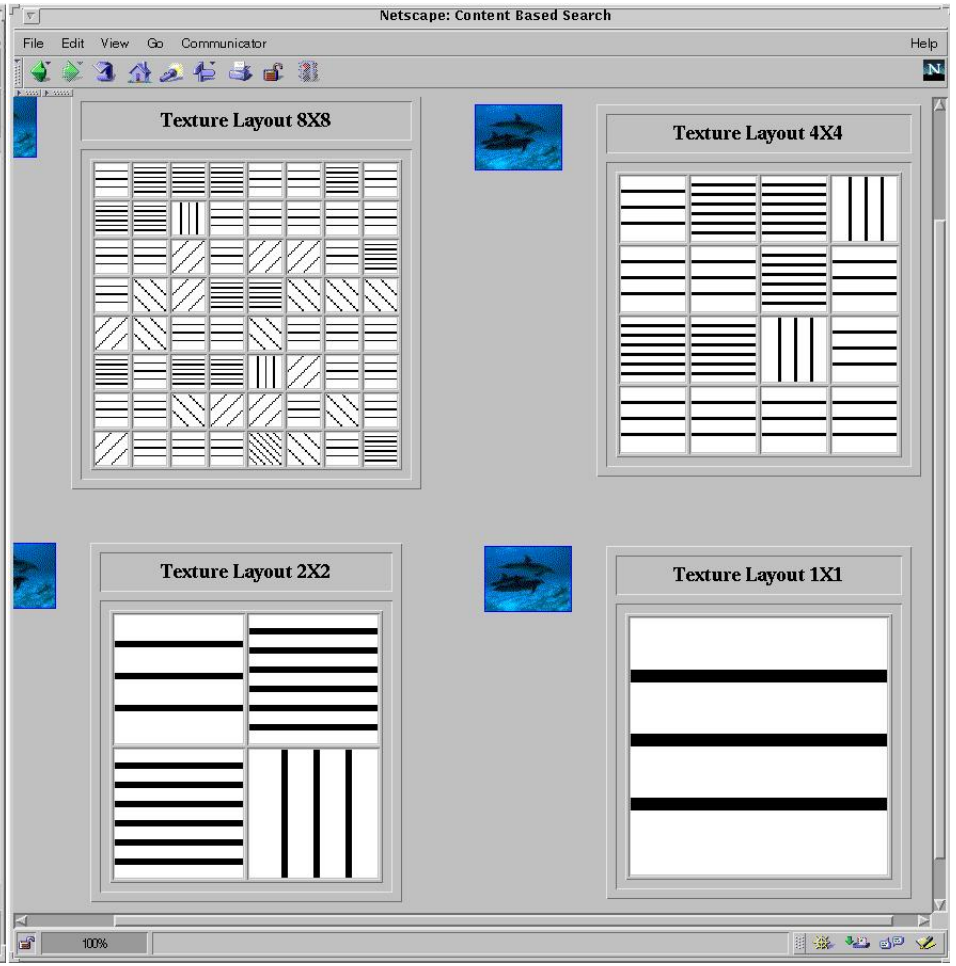


Multi-Dimensional Analysis in Multimedia Databases

Color histogram



Texture layout



Mining Multimedia Databases

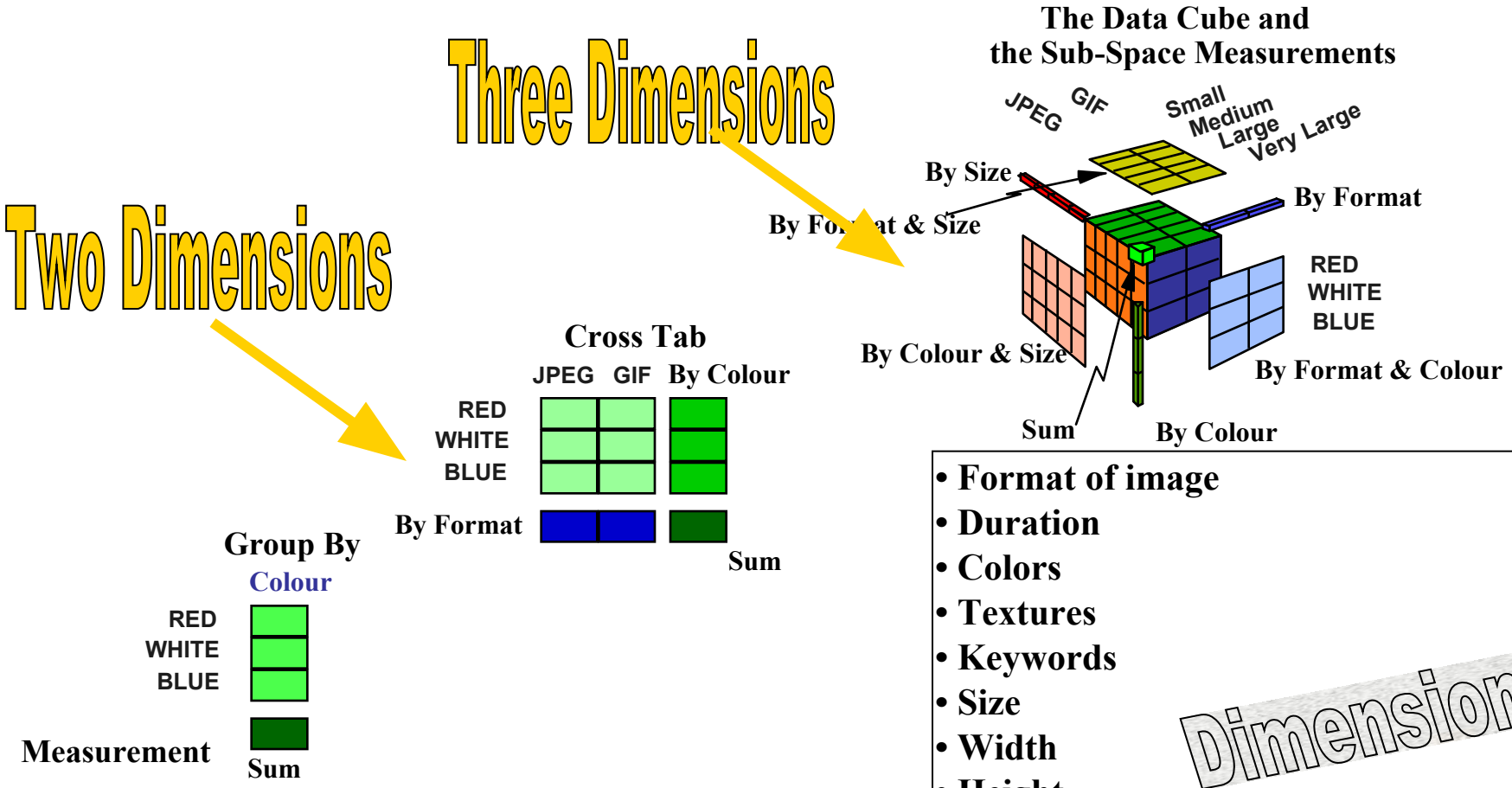
Refining or combining searches

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)

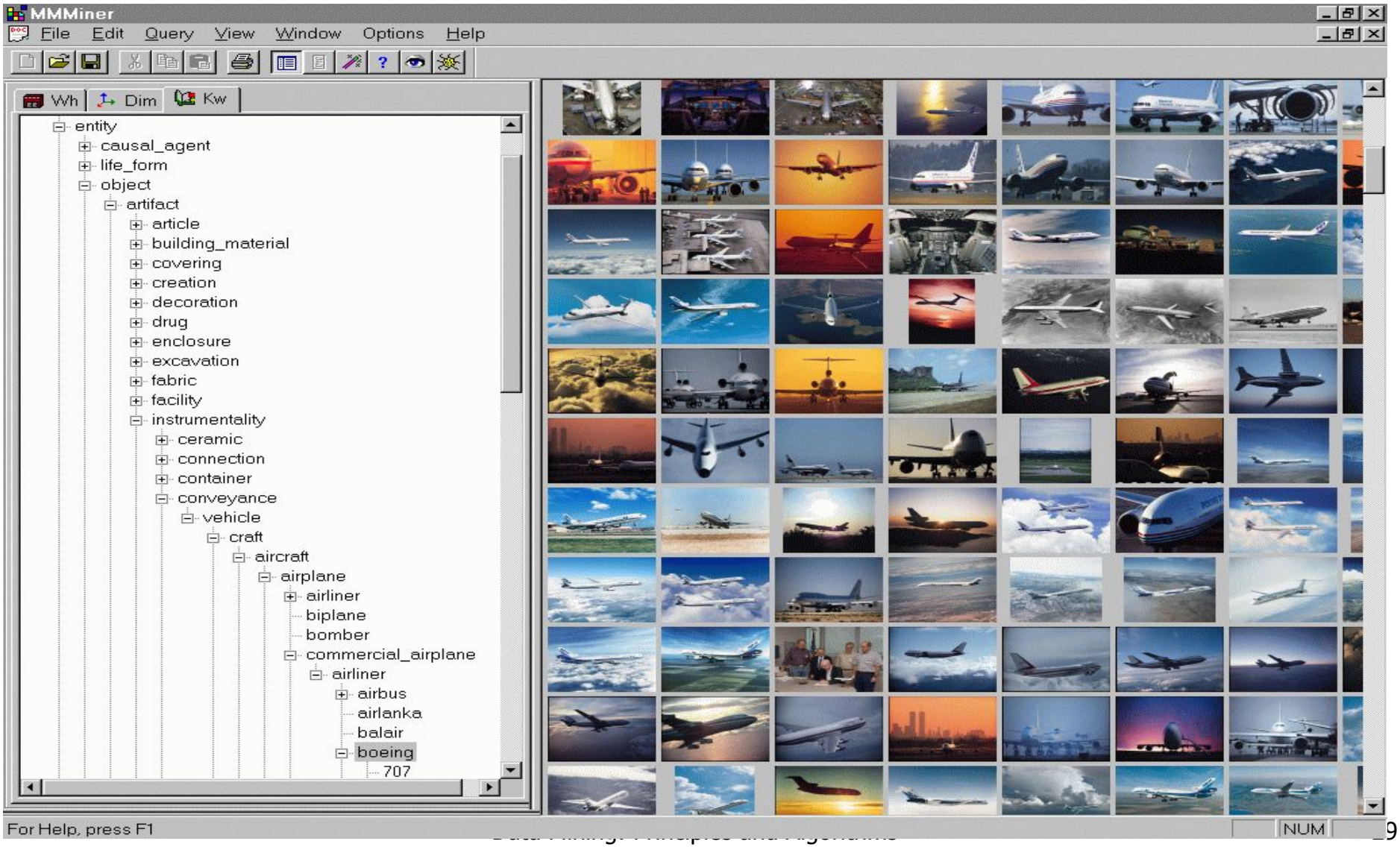
Mining Multimedia Databases



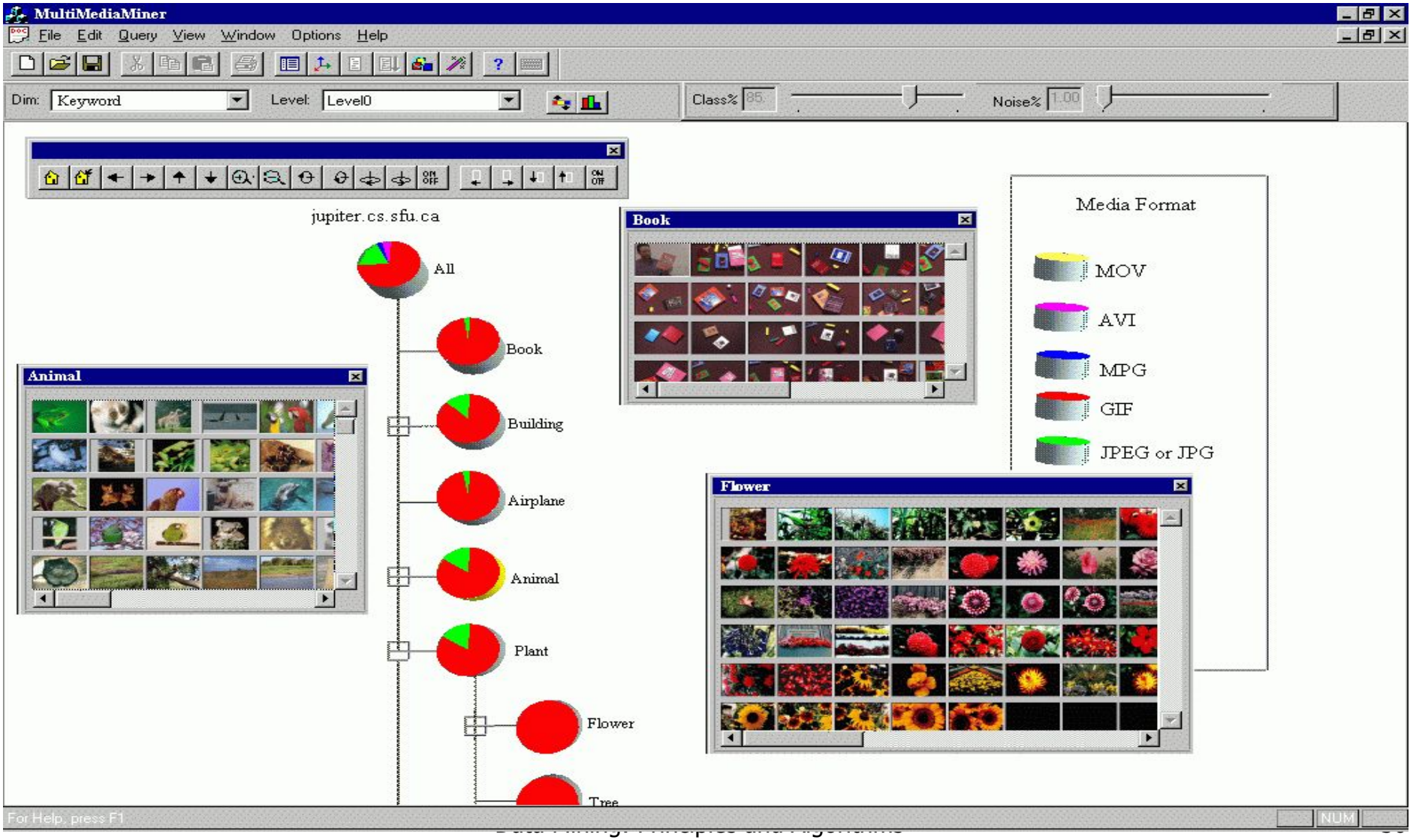
- Format of image
- Duration
- Colors
- Textures
- Keywords
- Size
- Width
- Height
- Internet domain of image
- Internet domain of parent pages
- Image popularity

Dimensions

Mining Multimedia Databases in MultiMediaMiner



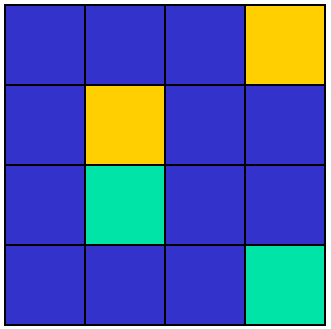
Classification in MultiMediaMiner



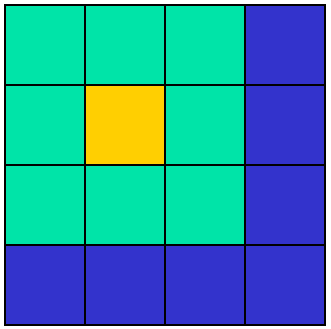
Mining Multimedia Databases

Spatial Relationships from Layout

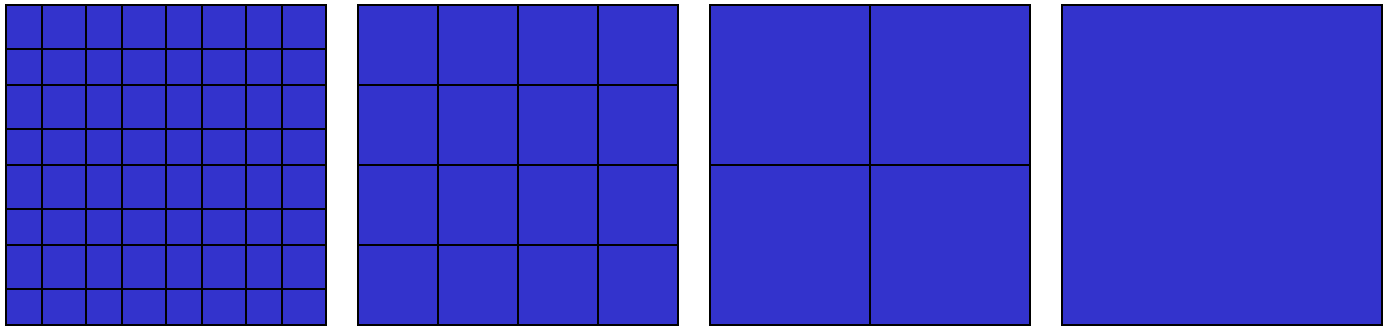
property P1 *on-top-of* property P2



property P1 *next-to* property P2

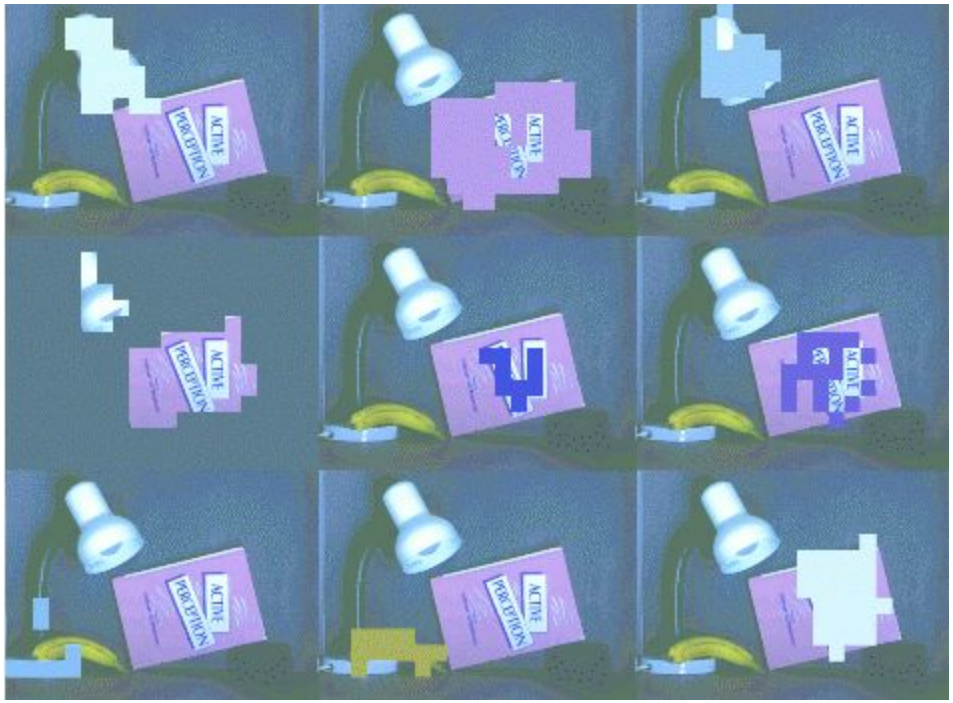


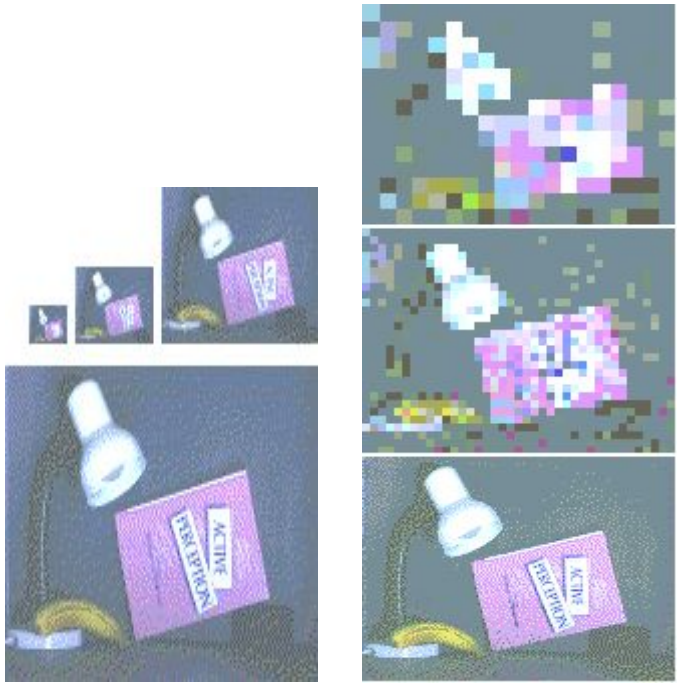
Different Resolution Hierarchy



Mining Multimedia Databases

From Coarse to Fine Resolution Mining





Data Mining Applications

- Data mining is an interdisciplinary field with wide and diverse applications
  - There exist nontrivial gaps between data mining principles and domain-specific applications
- Some application domains
  - Financial data analysis
  - Retail industry
  - Telecommunication industry
  - Biological data analysis

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

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 (2)

- Ex. 1. Design and construction of data warehouses based on the benefits of data mining
  - Multidimensional analysis of sales, customers, products, time, and region
- Ex. 2. Analysis of the effectiveness of sales campaigns
- Ex. 3. 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
- Ex. 4. 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

Biomedical Data Analysis

- DNA sequences
- 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

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

Privacy, Security and Social Impacts of Data Mining

- Many data mining applications do not touch personal data
  - E.g., meteorology, astronomy, geography, geology, biology, and other scientific and engineering data
- Many DM studies are on developing scalable algorithms to find general or statistically significant patterns, not touching individuals
- The real privacy concern: unconstrained access of individual records, especially privacy-sensitive information
- Method 1: Removing sensitive IDs associated with the data
- Method 2: Data security-enhancing methods
  - Multi-level security model: permit to access to only authorized level
  - Encryption: e.g., *blind signatures*, *biometric encryption*, and *anonymous databases* (personal information is encrypted and stored at different locations)
- Method 3: Privacy-preserving data mining methods

Trends in Data Mining

- 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

- 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

Data Mining Applications

- Data mining is an interdisciplinary field with wide and diverse applications
  - There exist nontrivial gaps between data mining principles and domain-specific applications
- Some application domains
  - Financial data analysis
  - Retail industry
  - Telecommunication industry
  - Biological data analysis

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

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 (2)

- Ex. 1. Design and construction of data warehouses based on the benefits of data mining
  - Multidimensional analysis of sales, customers, products, time, and region
- Ex. 2. Analysis of the effectiveness of sales campaigns
- Ex. 3. 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
- Ex. 4. 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

Biomedical Data Analysis

- DNA sequences
- 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

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

Trends in Data Mining

- 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

- 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