

Data Mining:

Concepts and Techniques

— Chapter 5 —

Jiawei Han

Department of Computer Science

University of Illinois at Urbana-Champaign

www.cs.uiuc.edu/~hanj

©2006 Jiawei Han and Micheline Kamber, All rights reserved

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, surveillance ...
 - Science: Remote sensing, genomics, scientific simulation, ...
 - Society and everyone: social networks, news, digital cameras, ...
- **We are drowning in data, but starving for knowledge!**
- “Necessity is the mother of invention”—Data mining—Automated analysis of massive data sets: natural from the evolution of Database

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 amounts of data
 - Data mining: a misnomer?
- Alternative names
 - Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.
- Watch out: Is everything “data mining”?
 - Simple search and query processing
 - (Deductive) expert systems



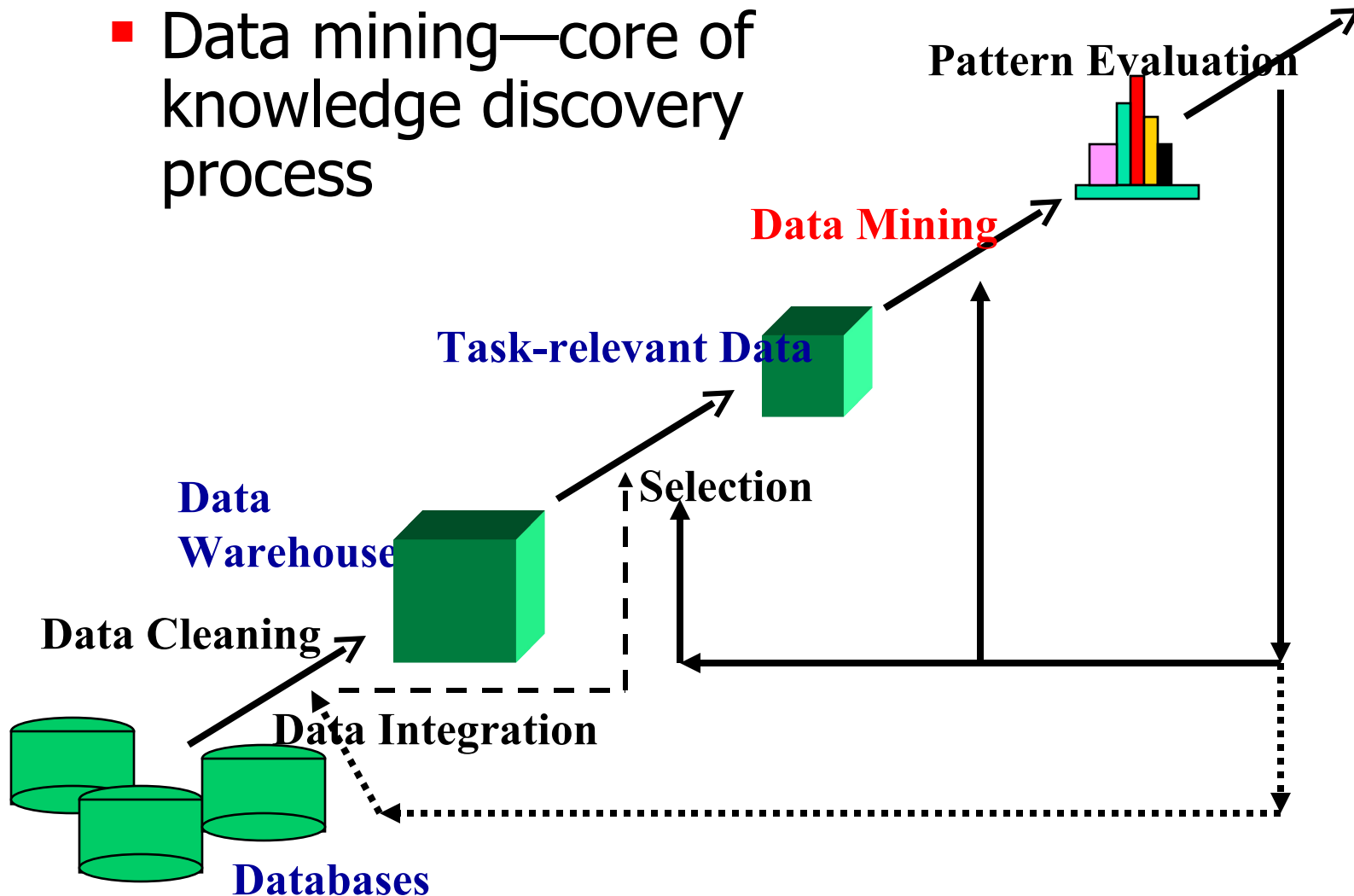
Why Data Mining?—Potential Applications

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

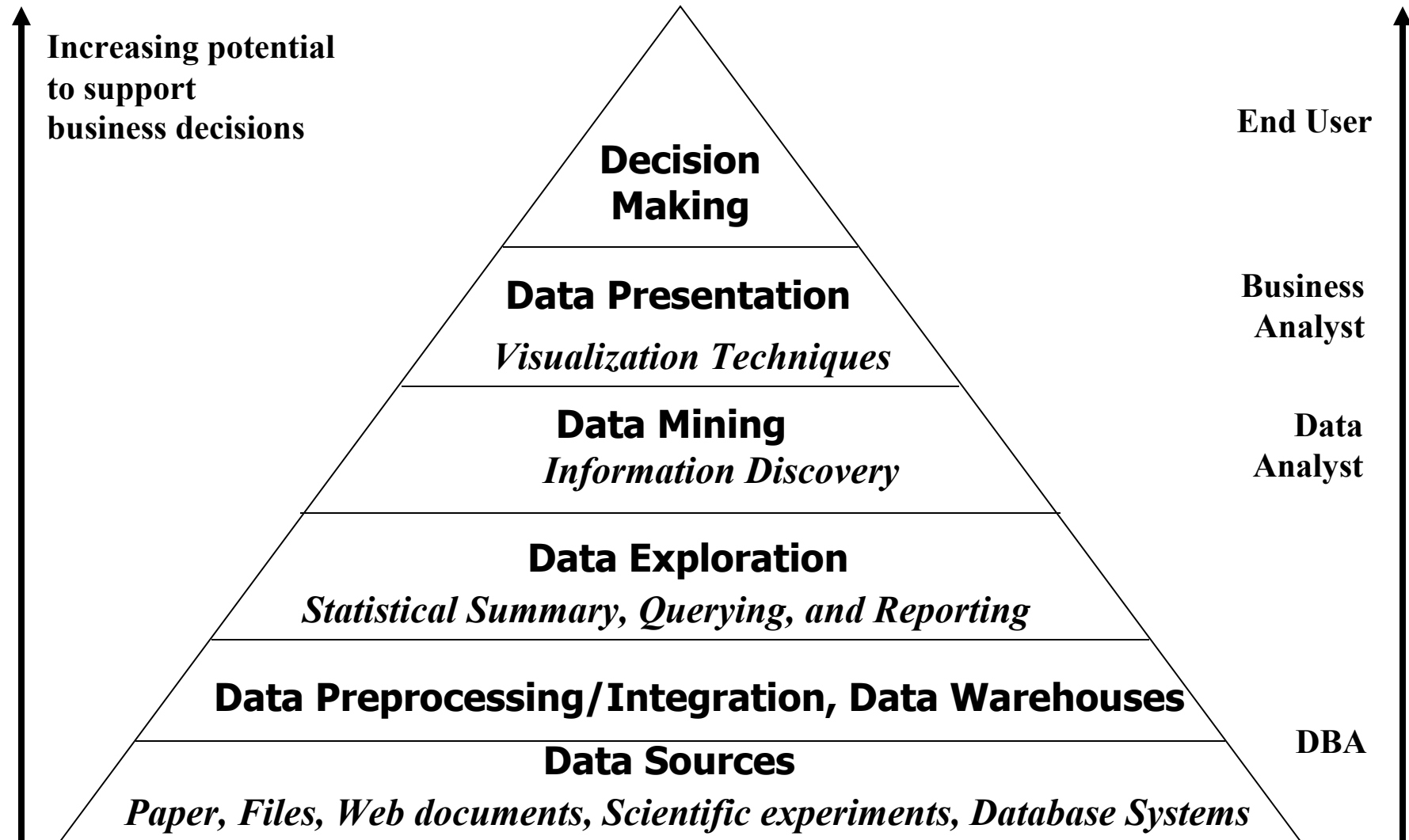
Knowledge Discovery (KDD) Process

Knowledge

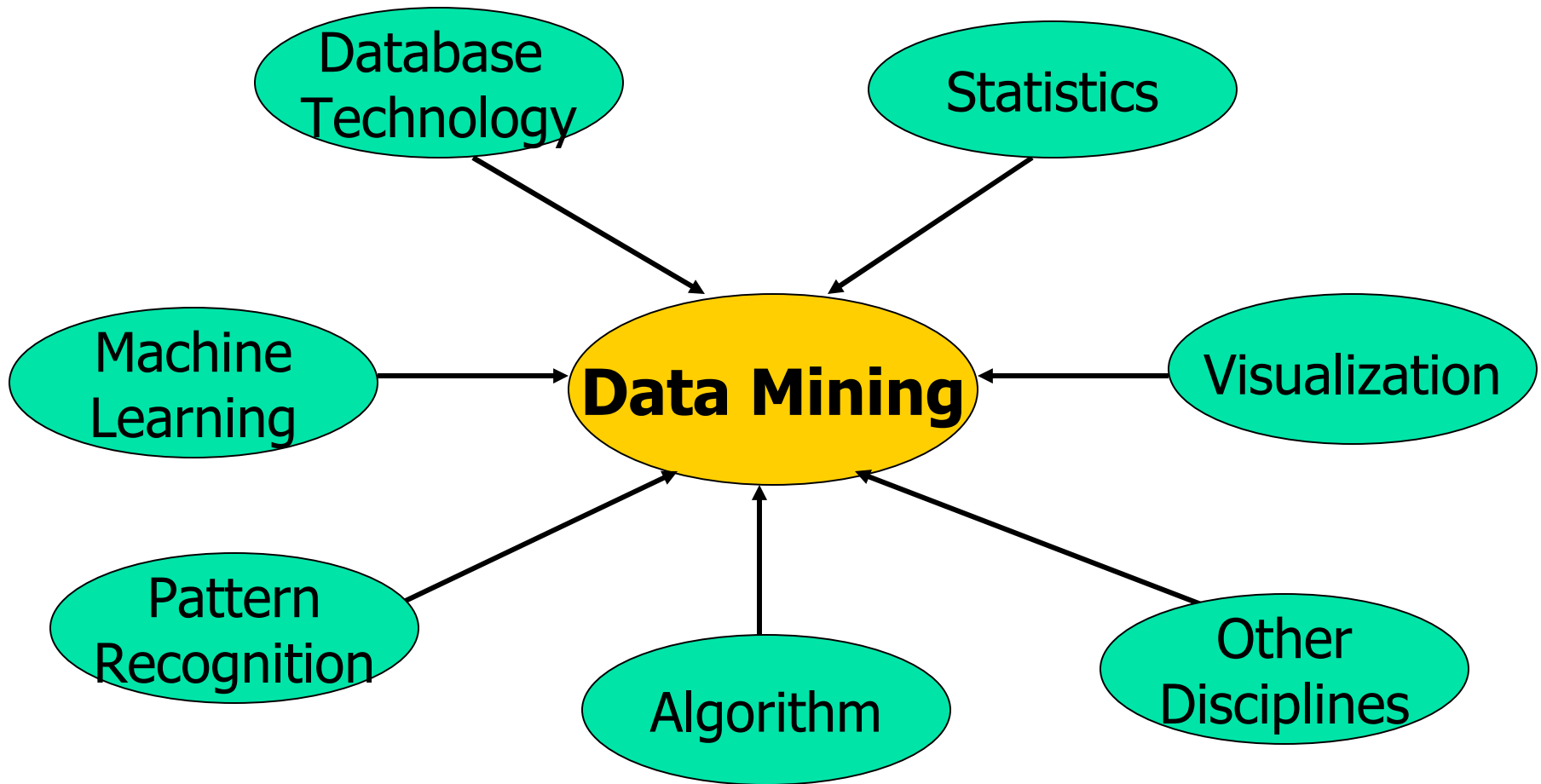
- Data mining—core of knowledge discovery process



Data Mining and Business Intelligence



Data Mining: Confluence of Multiple Disciplines



Why Not Traditional Data Analysis?

- Tremendous amount of data
 - Algorithms must be **highly scalable** to handle tera- and even peta-bytes of data
- High-dimensionality of data
 - Micro-array data may have tens of thousands of dimensions
 - Business data has typically 10-100 dimensions
- High complexity of data
 - Data streams and sensor data, surveillance data
 - Time-series data, temporal data, sequence data
 - Structure data, graphs, **social networks** and multi-linked data
 - Heterogeneous databases and legacy databases
 - Spatial-, spatiotemporal-, **multimedia**-, text- and web-data
 - Software programs, scientific simulations
- **New and sophisticated applications:** social networks, climate change, bioinformatics, etc.

Data Mining Characteristics

- **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 (1/2)

- **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 (2/2)

- 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
 - Periodicity analysis
 - Similarity-based analysis
- Other pattern-directed or statistical analyses

Are All the “Discovered” Patterns Interesting?

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

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?
- Heuristic vs. exhaustive search
- Association vs. classification vs. clustering

- **Search for only interesting patterns: An optimization problem**

- Can a data mining system find only the interesting patterns?
- Approaches
 - First generate all the patterns and then filter out the uninteresting ones
 - Generate only the interesting patterns—mining query optimization

Other Pattern Mining Issues

- Precise patterns vs. approximate patterns
 - Association and correlation mining: possible find sets of precise patterns
 - But approximate patterns can be more compact and sufficient
 - How to find high quality approximate patterns??
 - Gene sequence mining: approximate patterns are inherent
 - How to derive efficient approximate pattern mining algorithms??
- Constrained vs. non-constrained patterns
 - Why constraint-based mining?
 - What are the possible kinds of constraints? How to push constraints into the mining process?

Why Data Mining Query Language?

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

Primitives that Define a Data Mining Task

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

Primitive 1: 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

Primitive 2: Types of Knowledge to Be Mined

- Characterization
- Discrimination
- Association
- Classification/prediction
- Clustering
- Outlier analysis
- Etc.

Primitive 3: Background Knowledge

- A typical kind of background knowledge: Concept hierarchies
- Schema hierarchy
 - E.g., street < city < province_or_state < country
- Set-grouping hierarchy
 - E.g., {20-39} = young, {40-59} = middle_aged
- Operation-derived hierarchy
 - email address: hagonzal@cs.uiuc.edu
login-name < department < university < country
- Rule-based hierarchy
 - $\text{low_profit_margin}(X) \leq \text{price}(X, P_1) \text{ and } \text{cost}(X, P_2) \text{ and } (P_1 - P_2) < \50

Primitive 4: Pattern Interestingness Measure

- **Simplicity**

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

- **Certainty**

e.g., confidence, $P(A|B) = \#(A \text{ and } B) / \#(B)$, classification reliability or accuracy, certainty factor, rule strength, rule quality, discriminating weight, etc.

- **Utility**

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

- **Novelty**

not previously known, surprising (used to remove redundant uninteresting rules, e.g., student math $\rightarrow$ plays chess)

Primitive 5: Presentation of Discovered Patterns

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

DMQL—A Data Mining Query Language

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

An Example Query in DMQL

Example 1.11 Mining classification rules. Suppose, as a marketing manager of *AllElectronics*, you would like to classify customers based on their buying patterns. You are especially interested in those customers whose salary is no less than \$40,000, and who have bought more than \$1,000 worth of items, each of which is priced at no less than \$100. In particular, you are interested in the customer's age, income, the types of items purchased, the purchase location, and where the items were made. You would like to view the resulting classification in the form of rules. This data mining query is expressed in DMQL³ as follows, where each line of the query has been enumerated to aid in our discussion.

```
use database AllElectronics_db
use hierarchy location_hierarchy for T.branch, age_hierarchy for C.age
mine classification as promising_customers
in relevance to C.age, C.income, I.type, I.place_made, T.branch
from customer C, item I, transaction T
where I.item_ID = T.item_ID and C.cust_ID = T.cust_ID
      and C.income  $\geq$  40,000 and I.price  $\geq$  100
group by T.cust_ID
having sum(I.price)  $\geq$  1,000
display as rules
```

Other Data Mining Languages & Standardization Efforts

- Association rule language specifications
 - MSOL (Imielinski & Virmani'99)
 - MineRule (Meo Psaila and Ceri'96)
 - Query flocks based on Datalog syntax (Tsur et al'98)
- OLEDB for DM (Microsoft'2000) and recently DMX (Microsoft SQLServer 2005)
 - Based on OLE, OLE DB, OLE DB for OLAP, C#
 - Integrating DBMS, data warehouse and data mining
- DMML (Data Mining Mark-up Language) by DMG (www.dmg.org)
 - Providing a platform and process structure for effective data mining
 - Emphasizing on deploying data mining technology to solve business problems
- PMML (Predictive Model Markup Language) by DMG. (SPSS, SAS, Teradata, Weka, IBM (InofSphere, DB2), Microsoft SQL Server)
- DMX Data Mining extensions language for MS SQL Server 2008.

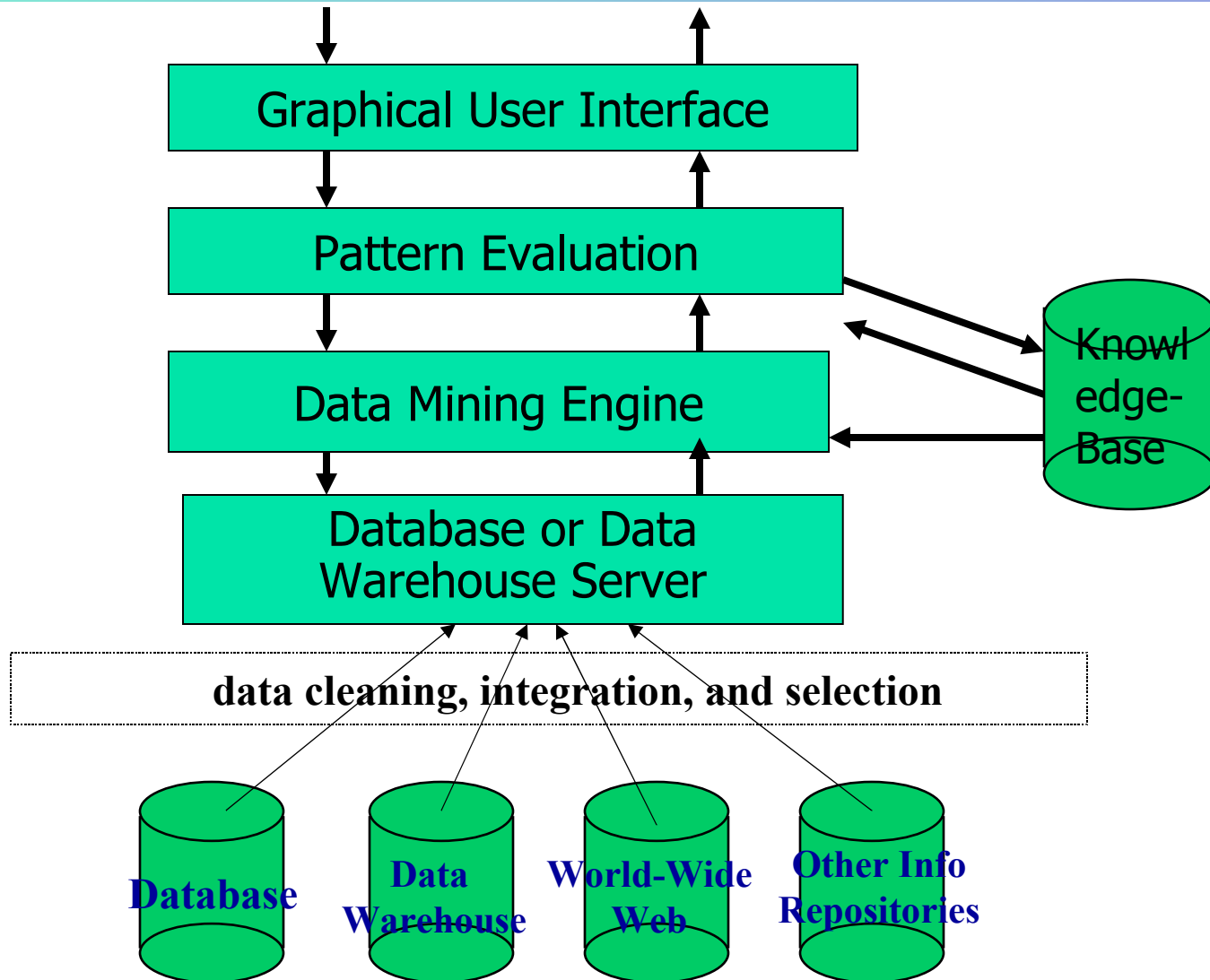
Integration of Data Mining and Data Warehousing

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

Coupling Data Mining with DB/DW Systems

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

Architecture: Typical Data Mining System



Major Issues in Data Mining (1/2)

- Mining methodology
 - Mining different kinds of knowledge from diverse data types, e.g., bio, stream, Web
 - Performance: efficiency, effectiveness, and scalability
 - Pattern evaluation: the interestingness problem
 - Incorporation of background knowledge
 - Handling noise and incomplete data
 - Parallel, distributed and incremental mining methods
 - Integration of the discovered knowledge with existing one: knowledge fusion

Major Issues in Data Mining (2/2)

- User interaction

- Data mining query languages and ad-hoc mining
- Expression and visualization of data mining results
- Interactive mining of knowledge at multiple levels of abstraction

- Applications and social impacts

- Domain-specific data mining & invisible data mining
- Protection of data security, integrity, and privacy

Summary

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

A Brief History of Data Mining Society

- 1989 IJCAI Workshop on Knowledge Discovery in Databases
 - Knowledge Discovery in Databases (G. Piatetsky-Shapiro and W. Frawley, 1991)
- 1991-1994 Workshops on Knowledge Discovery in Databases
 - Advances in Knowledge Discovery and Data Mining (U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, 1996)
- 1995-1998 International Conferences on Knowledge Discovery in Databases and Data Mining (KDD'95-98)
 - Journal of Data Mining and Knowledge Discovery (1997)
- ACM SIGKDD conferences since 1998 and SIGKDD Explorations
- More conferences on data mining
 - PAKDD (1997), PKDD (1997), SIAM-Data Mining (2001), (IEEE) ICDM (2001), etc.
- ACM Transactions on KDD starting in 2007

Conferences and Journals on Data Mining

- KDD Conferences
 - ACM SIGKDD Int. Conf. on Knowledge Discovery in Databases and Data Mining (**KDD**)
 - SIAM Data Mining Conf. (**SDM**)
 - (IEEE) Int. Conf. on Data Mining (**ICDM**)
 - Conf. on Principles and practices of Knowledge Discovery and Data Mining (**PKDD**)
 - Pacific-Asia Conf. on Knowledge Discovery and Data Mining (**PAKDD**)
- Other related conferences
 - ACM SIGMOD
 - VLDB
 - (IEEE) ICDE
 - WWW, SIGIR
 - ICML, CVPR, NIPS
- Journals
 - Data Mining and Knowledge Discovery (DAMI or DMKD)
 - IEEE Trans. On Knowledge and Data Eng. (TKDE)
 - KDD Explorations
 - ACM Trans. on KDD

Where to Find References? DBLP, CiteSeer, Google

- Data mining and KDD (SIGKDD: CDROM)
 - Conferences: ACM-SIGKDD, IEEE-ICDM, SIAM-DM, PKDD, PAKDD, etc.
 - Journal: Data Mining and Knowledge Discovery, KDD Explorations, ACM TKDD
- Database systems (SIGMOD: ACM SIGMOD Anthology—CD ROM)
 - Conferences: ACM-SIGMOD, ACM-PODS, VLDB, IEEE-ICDE, EDBT, ICDT, DASFAA
 - Journals: IEEE-TKDE, ACM-TODS/TOIS, JIIS, J. ACM, VLDB J., Info. Sys., etc.
- AI & Machine Learning
 - Conferences: Machine learning (ML), AAAI, IJCAI, COLT (Learning Theory), CVPR, NIPS, etc.
 - Journals: Machine Learning, Artificial Intelligence, Knowledge and Information Systems, IEEE-PAMI, etc.
- Web and IR
 - Conferences: SIGIR, WWW, CIKM, etc.
 - Journals: WWW: Internet and Web Information Systems,
- Statistics
 - Conferences: Joint Stat. Meeting, etc.
 - Journals: Annals of statistics, etc.
- Visualization
 - Conference proceedings: CHI, ACM-SIGGraph, etc.
 - Journals: IEEE Trans. visualization and computer graphics, etc.

Recommended Reference Books

- S. Chakrabarti. Mining the Web: Statistical Analysis of Hypertext and Semi-Structured Data. Morgan Kaufmann, 2002
- R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, 2ed., Wiley-Interscience, 2000
- T. Dasu and T. Johnson. Exploratory Data Mining and Data Cleaning. John Wiley & Sons, 2003
- U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996
- U. Fayyad, G. Grinstein, and A. Wierse, Information Visualization in Data Mining and Knowledge Discovery, Morgan Kaufmann, 2001
- **J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2nd ed., 2006**
- D. J. Hand, H. Mannila, and P. Smyth, Principles of Data Mining, MIT Press, 2001
- **T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer-Verlag, 2001**
- T. M. Mitchell, Machine Learning, McGraw Hill, 1997
- G. Piatetsky-Shapiro and W. J. Frawley. Knowledge Discovery in Databases. AAAI/MIT Press, 1991
- **P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Wiley, 2005**
- S. M. Weiss and N. Indurkha, Predictive Data Mining, Morgan Kaufmann, 1998
- **I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 2nd ed. 2005**

Chapter 5: Mining Frequent Patterns, Association and Correlations

- Basic concepts and a road map 
- Efficient and scalable frequent itemset mining methods
- Mining various kinds of association rules
- From association mining to correlation analysis
- Constraint-based association mining
- Summary

What Is Frequent Pattern Analysis?

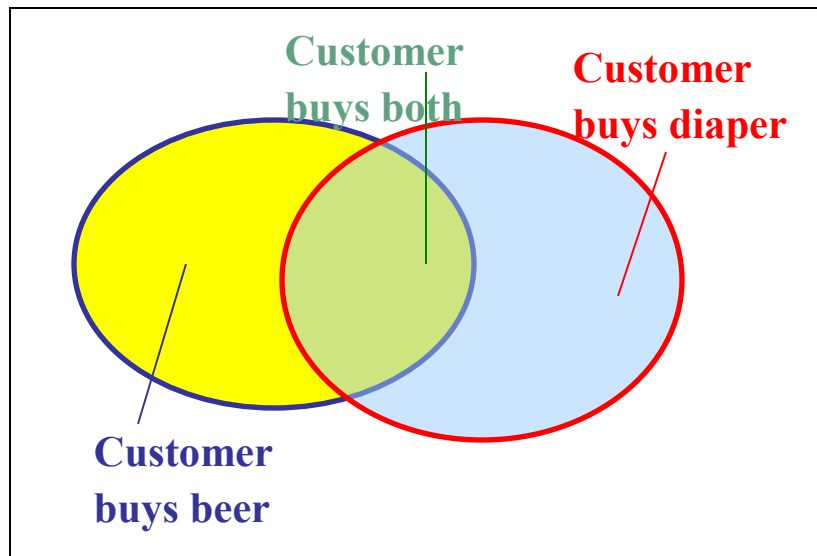
- **Frequent pattern**: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- First proposed by Agrawal, Imielinski, and Swami [AIS93] in the context of **frequent itemsets** and **association rule mining**
- Motivation: Finding inherent regularities in 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
 - Data warehousing: iceberg cube and cube-gradient
 - Semantic data compression: fascicles
 - 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 $X = \{x_1, \dots, x_k\}$
- 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:

$A \rightarrow D$ (60%, 100%)

$D \rightarrow A$ (60%, 75%)

Closed Patterns and Max-Patterns

- A long pattern contains a combinatorial number of sub-patterns, e.g., $\{a_1, \dots, a_{100}\}$ contains $\binom{1}{100} + \binom{2}{100} + \dots + \binom{100}{100} = 2^{100} - 1 \approx 1.27 \times 10^{30}$ sub-patterns!
- Solution: Mine *closed patterns* and *max-patterns* instead
- An itemset X is **closed** if X is *frequent* and there exists *no super-pattern* $Y \supset X$, with the same support as X (proposed by Pasquier, et al. @ ICDT'99)
Note: if $Y \supset X$ then $\text{support}(Y) \leq \text{Support}(X)$
- An itemset X is a **max-pattern** if X is frequent and there exists no frequent super-pattern $Y \supset X$ (proposed by Bayardo @ SIGMOD'98)
Note: X is frequent $\Rightarrow P(X \text{ in } T) \geq \text{sup}_{\min}$.
Furthermore: if $Y \supset X$ then $P(X \text{ in } T) \geq P(Y \text{ in } T)$ i.e., Y not necessary frequent.
- Closed pattern is a lossless compression of freq. patterns
 - Reducing the # of patterns and rules

Closed Patterns and Max-Patterns

- Exercise. $DB = \{ \langle a_1, \dots, a_{100} \rangle, \langle a_1, \dots, a_{50} \rangle \}$
 - $Min_sup = 1$.
- What is the set of **closed itemset**? (no super-pattern with same support)
 - $\langle a_1, \dots, a_{100} \rangle: 1$
 - $\langle a_1, \dots, a_{50} \rangle: 2$
- What is the set of **max-pattern**? (no frequent super-pattern)
 - $\langle a_1, \dots, a_{100} \rangle: 1$
- What is the set of **all patterns**?
 - !!

Chapter 5: Mining Frequent Patterns, Association and Correlations

- Basic concepts and a road map
- Efficient and scalable frequent itemset mining methods 
- Mining various kinds of association rules
- From association mining to correlation analysis
- Constraint-based association mining
- Summary

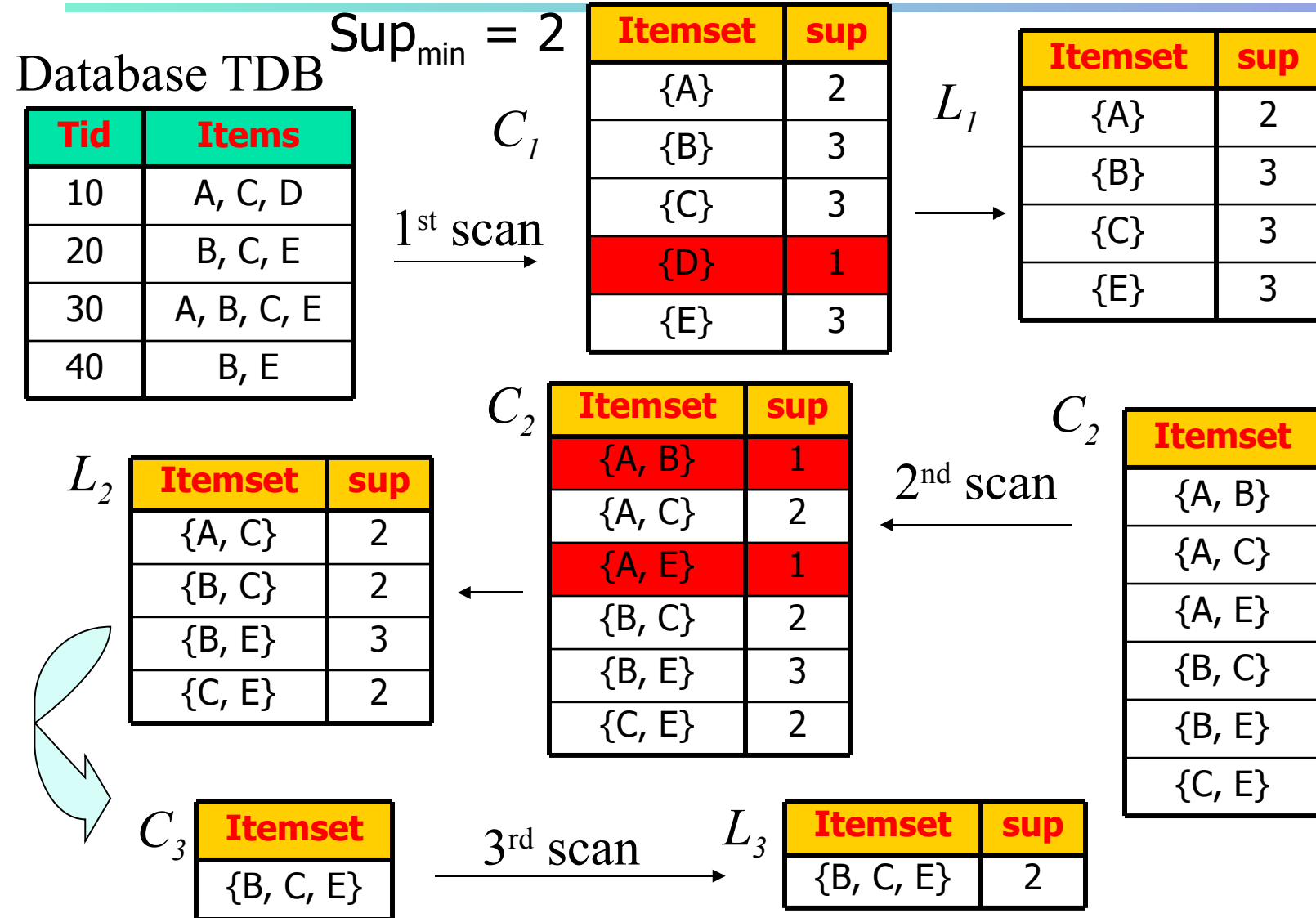
Scalable Methods for Mining Frequent Patterns

- The **downward closure** property of frequent patterns
 - Any subset of a frequent itemset must be frequent
 - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
 - i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}
- Scalable mining methods: Three major approaches
 - Apriori (Agrawal & Srikant@VLDB'94)
 - Freq. pattern growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
 - Vertical data format approach (Charm—Zaki & Hsiao @SDM'02)

Apriori: A Candidate Generation-and-Test Approach

- Apriori pruning principle: If there is **any** itemset which is infrequent, its superset should not be generated/tested! (Agrawal & Srikant @VLDB'94, Mannila, et al. @ KDD' 94)
- Method:
 - Initially, scan the DB once to get frequent 1-itemset
 - **Generate** all length $(k+1)$ **candidate** itemsets from length k **frequent** itemsets
 - **Test** the candidates against the DB
 - Terminate when no frequent or candidate set can be generated

The Apriori Algorithm—An Example



The Apriori Algorithm

- Pseudo-code:

C_k : Candidate itemset of size k

L_k : frequent itemset of size k

$L_1 = \{\text{frequent items}\}$

for ($k = 1$; $L_k \neq \emptyset$; $k++$) **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 and hence cannot be frequent
 - $C_4 = \{abcd\}$

How to Generate Candidates?

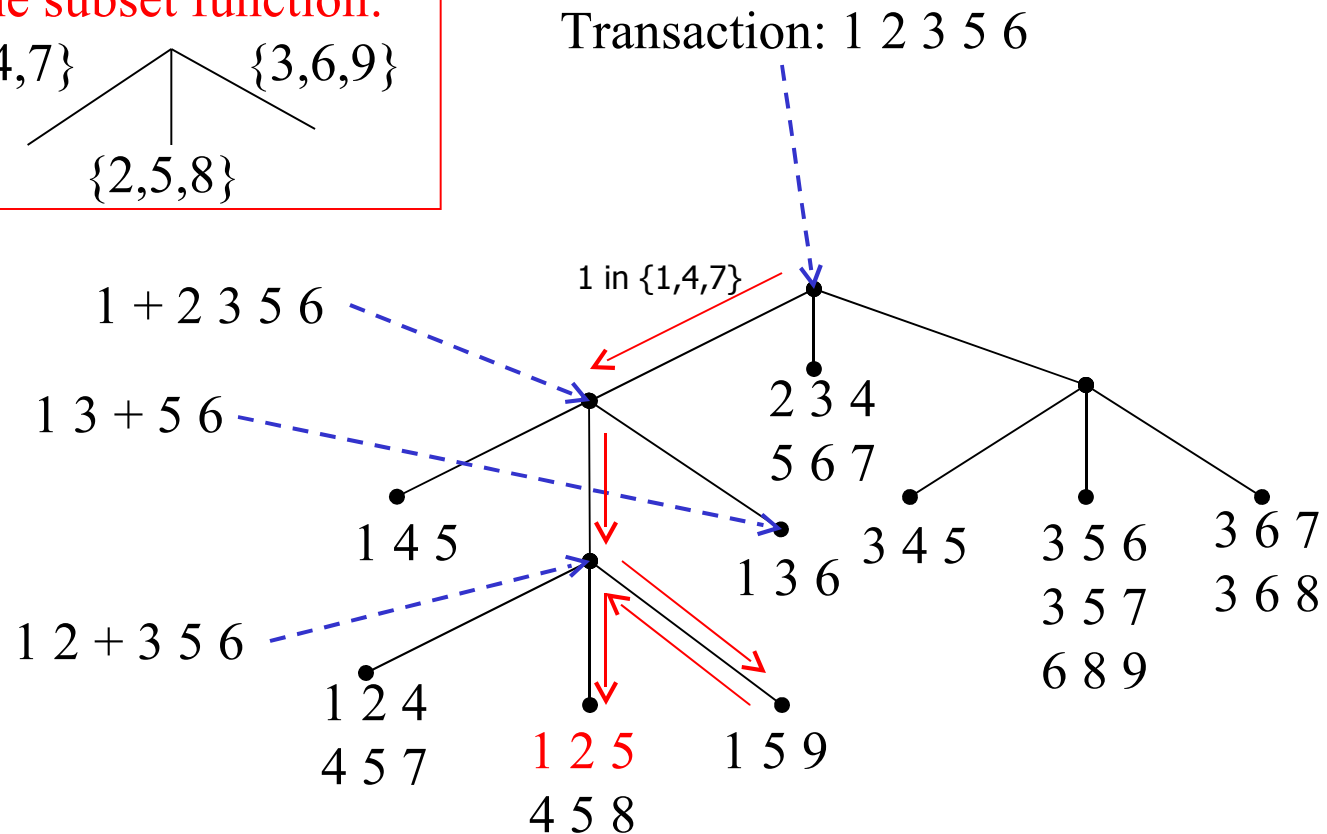
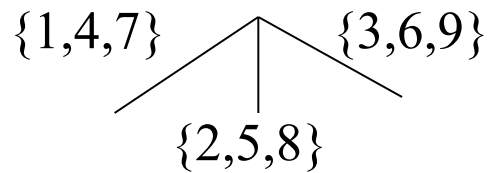
- Suppose the items in L_{k-1} are listed in an order
- Step 1: self-joining L_{k-1}
insert into C_k :
select **$p.item_1, p.item_2, \dots, p.item_{k-2}, p.item_{k-1}, q.item_{k-1}$**
from $L_{k-1} \text{ } p, L_{k-1} \text{ } q$
where **$p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}$** and
 $p.item_{k-1} < q.item_{k-1}$
- Step 2: Apriori pruning
for all ***itemsets* c in C_k** do
 for all ***(k-1)-subsets* s of c** do
 if (s is not in L_{k-1}) then delete c from C_k

How to Count Supports of Candidates?

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

Example: Counting Supports of Candidates

At each interior node
the subset function:



3-item set candidates

Efficient Implementation of Apriori in SQL

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

Challenges of Frequent Pattern Mining

- Challenges

- Multiple scans of transaction database
- Huge number of candidates
- Tedious workload of support counting for candidates

- Improving Apriori: general ideas

- Reduce passes of transaction database scans
- Shrink number of candidates
- Facilitate support counting of candidates

Partition: Scan Database Only Twice

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

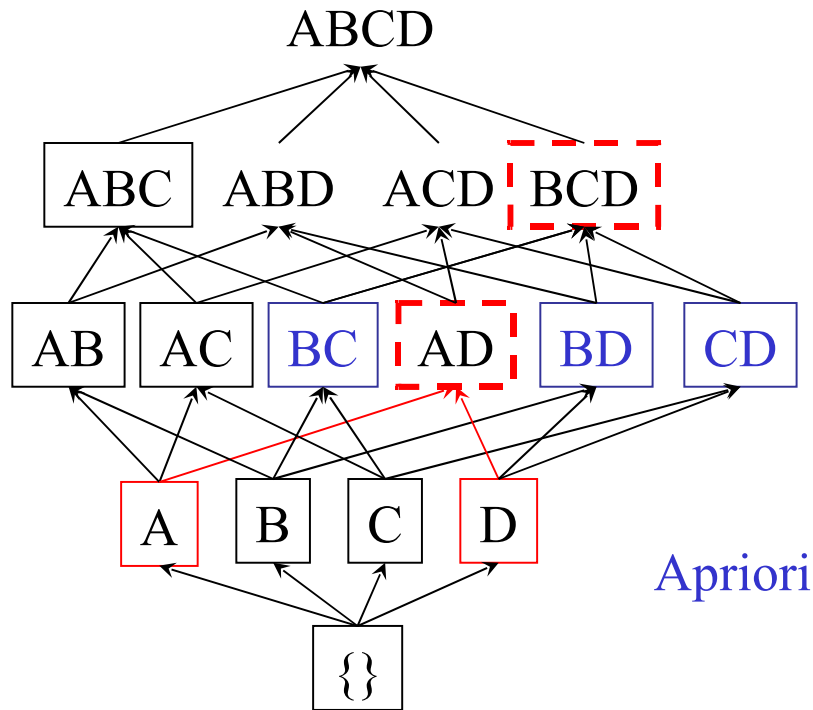
Direct Hashing and Pruning (DHP): Reduce the Number of Candidates

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

Sampling for Frequent Patterns

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

Dynamic Itemset Counting (DIC): Reduce Number of Scans

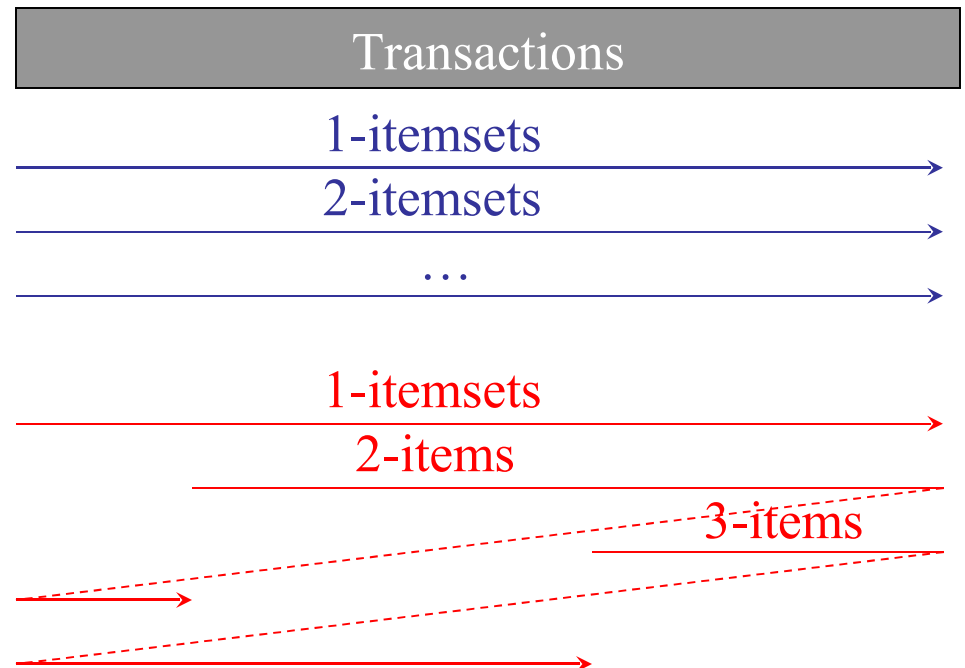


Itemset lattice

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

Apriori

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



DIC

Bottleneck of Frequent-pattern Mining

- Multiple database scans are **costly**
- Mining long patterns needs many passes of scanning and generates lots of candidates
 - To find frequent itemset $i_1 i_2 \dots i_{100}$
 - # of scans: **100**
 - # of Candidates: $\binom{1}{100} + \binom{2}{100} + \dots + \binom{100}{100} = 2^{100} - 1 \approx 1.27 \times 10^{30}$
 - Bottleneck: candidate-generation-and-test
- Can we avoid candidate generation?

Mining Frequent Patterns Without Candidate Generation

- Grow long patterns from short ones using local frequent items
 - “abc” is a frequent pattern
 - Get all transactions having “abc”: DB|abc
 - “d” is a local frequent item in DB|abc → abcd is a frequent pattern
- => Frequent Pattern Trees