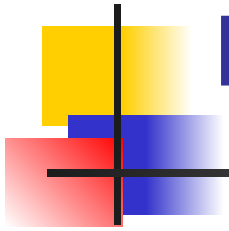
Data Mining Primitives

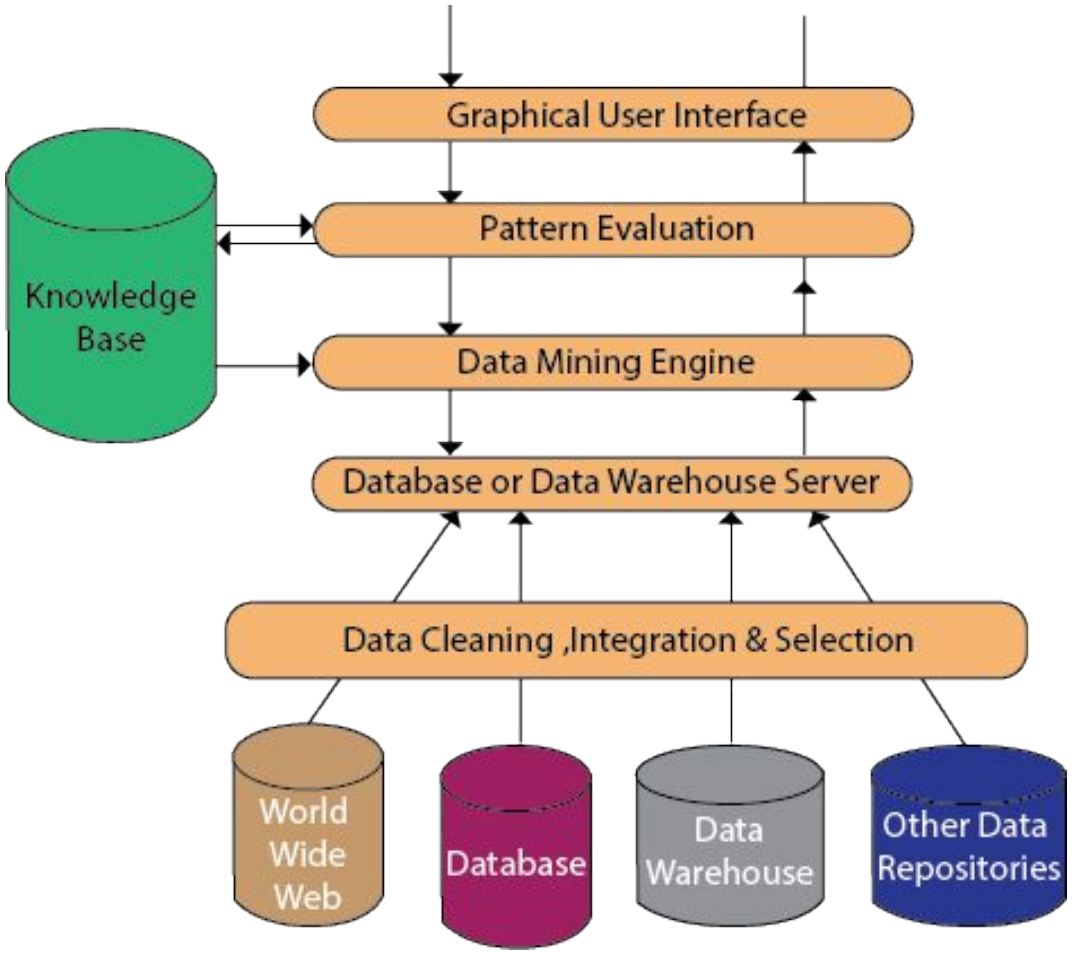


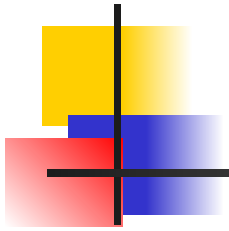
Data Mining

- Data Mining tools perform data analysis and may uncover important data patterns, contributing greatly to business strategies, knowledge bases, and scientific and medical research.

- Data mining refers to *extracting or "**mining**" knowledge from large amounts of data*. Also referred as Knowledge Discovery in Databases.

Data Mining Architecture



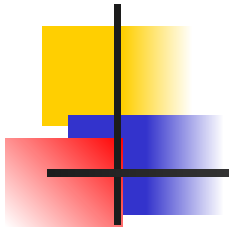


Data Mining Architecture

- Misconception: Data mining systems can autonomously dig out *all* of the valuable knowledge from a given large database, without human intervention.
- If there was no user intervention then the system would uncover a large set of patterns that may even surpass the size of the database. Hence, user interference is required.
- This user communication with the system is provided by using a set of *data mining primitives*.

Is patterns finding autonomous?

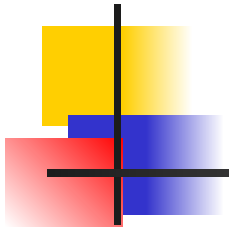
- Unrealistic because the patterns could be too many but uninteresting
- Data mining should be an interactive process
  - User directs what to be mined
- Users must be provided with a set of primitives to be used to communicate with the data mining system



Data Mining Primitives

- **Data mining primitives** define a data mining task, which can be specified in the form of a data mining query.

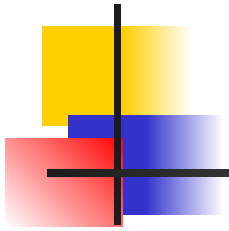
- Incorporating these primitives in a data mining query language
  - More flexible user interaction
  - Foundation for design of graphical user interface
  - Standardization of data mining industry and practice



Data Mining Primitives

Data mining primitives define a data mining task, which can be specified in the form of a data mining query.

- Task Relevant Data
- Kinds of knowledge to be mined
- Background knowledge
- Interestingness measure
- Presentation and visualization of discovered patterns

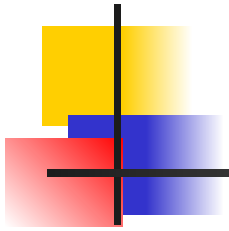


Data Mining Primitives

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

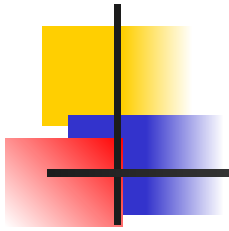
Task-Relevant Data (Minable View)

- Database/warehouse portion to be investigated
- Database or data warehouse name
- Database tables or data warehouse cubes
- Condition for data selection
- Attributes of interest (Relevant attributes or dimensions)
- Data grouping criteria



Example

- If a data mining task is to study **associations** between items frequently purchased at *AllElectronics* by customers in Canada, the task relevant data can be specified by providing the following information:
  - Name of the *database or data warehouse* to be used (e.g., *AllElectronics_db*)
  - Names of the *tables or data cubes* containing relevant data (e.g., *item, customer, purchases* and *items_sold*)
  - *Conditions* for selecting the relevant data (e.g., retrieve data pertaining to purchases made in Canada for the current year)
  - The *relevant attributes or dimensions* (e.g., *name* and *price* from the *item* table and income and age from the customer table)



Example

- In addition, the user may specify that the data retrieved be grouped by certain attributes, such as “group by date”.
- Moreover, in a data mining query, the conditions provided for data selection can be at a level that is conceptually higher than the data in the database or data warehouse.
  - For ex. a user may specify a selection on items at AllElectronics using concept type =”home entertainment” even though individual items in DB may not be stored according to type but at a lower conceptual level such as “TV”, “CD Player” or “VCR”.

Types of knowledge to be mined

- Characterization
- Discrimination
- Association
- Classification/prediction
- Clustering
- Outlier analysis
- Other data mining tasks
- Example: To study buying habits of customers in INDIA, choose to **mine association** between customer profiles and the items that these customers like to buy.

Types of knowledge to be mined

A user studying the buying habits of *allelectronics* customers may choose to mine *association rules* of the form:

P (X:customer,W) ^ Q (X,Y) => buys (X,Z)

Meta rules such as the following can be specified:

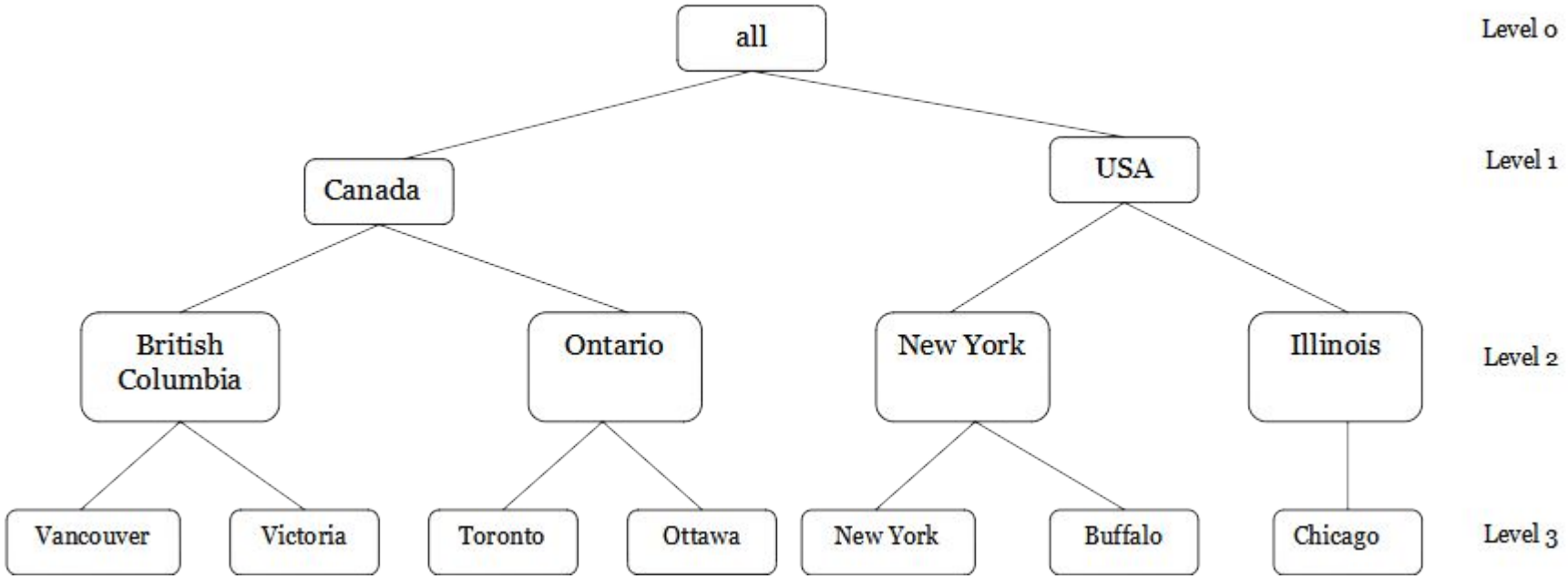
age (X, "30.....39") ^ income (X, "40k....49K") => buys (X, "VCR")
[2.2%, 60%]

occupation (X, "student ") ^ age (X, "20.....29")=> buys (X, "computer")
[1.4%, 70%]

Background Knowledge: Concept Hierarchies

- It is the information about the domain to be mined
- Concept hierarchy: is a powerful form of background knowledge.
- Four major types of concept hierarchies:
  schema hierarchies
  set-grouping hierarchies
  operation-derived hierarchies
  rule-based hierarchies

Example

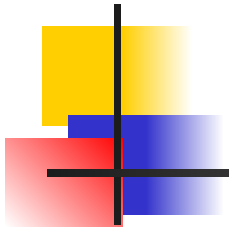

all
Level 0
Canada
USA
Level 1
British Columbia
Ontario
New York
Illinois
Level 2
Vancouver
Victoria
Toronto
Ottawa
New York
Buffalo
Chicago
Level 3

Background Knowledge: Concept Hierarchies

- Rolling Up - Generalization of data
  Allows to view data at more meaningful and explicit abstractions.
  Makes it easier to understand
  Compresses the data
  Would require fewer input/output operations
- Drilling Down - Specialization of data
  Concept values replaced by lower level concepts
- There may be more than concept hierarchy for a given attribute or dimension based on different user viewpoints
- Example:
  Regional sales manager may prefer the previous concept hierarchy but marketing manager might prefer to see location with respect to linguistic lines in order to facilitate the distribution of commercial ads.

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: login-name < department < university < country
- Rule-based hierarchy
  - low_profit_margin (X) <= price(X, P1) and cost (X, P2) and (P1 - P2) < $50



Schema Hierarchy

- Schema hierarchy is the total or partial order among attributes in the database schema.
- May formally express existing semantic relationships between attributes.
- Provides metadata information.
- Example: location hierarchy
  street < *city* < *province/state* < *country*

Set-grouping hierarchies

- Organizes values for a given attribute into groups or sets or range of values.
- Total or partial order can be defined among groups.
- Used to refine or enrich schema-defined hierarchies.
- Typically used for small sets of object relationships.
- Example: Set-grouping hierarchy for age

  {young, middle_aged, senior} all (age)

  {20....29} young

  {40....59} middle_aged

  {60....89} senior

Operation-derived hierarchies

- Operation-derived:
  based on operations specified
  operations may include
    - decoding of information-encoded strings
    - information extraction from complex data objects
    - data clustering

  Example: URL or email address
  xyz@cs.iitm.in gives login name < dept. < univ. < country

Rule-based hierarchies

- Rule-based:

  Occurs when either whole or portion of a concept hierarchy is defined as a set of rules and is evaluated dynamically based on current database data and rule definition

- Example: Following rules are used to categorize items as *low_profit*, *medium_profit* and *high_profit_margin*.

  $low\_profit\_margin(X) <= price(X,P1) \wedge cost(X,P2) \wedge ((P1-P2)<50)$

  $medium\_profit\_margin(X) <= price(X,P1) \wedge cost(X,P2) \wedge ((P1-P2)\geq 50) \wedge ((P1-P2)\leq 250)$

  $high\_profit\_margin(X) <= price(X,P1) \wedge cost(X,P2) \wedge ((P1-P2)>250)$

Interestingness measure

- To evaluate the discovered pattern
- Used to confine the number of uninteresting patterns returned by the process.
- Based on the structure of patterns and statistics underlying them.
- Associate a threshold which can be controlled by the user.
- Patterns not meeting the threshold are not presented to the user.

- Objective measures of **pattern interestingness**:
  - simplicity
  - certainty (confidence)
  - utility (support)
  - novelty

Interestingness measure

- Simplicity

  a patterns interestingness is based on its overall simplicity for human comprehension.

  Example: Rule length is a simplicity measure

- Certainty (confidence)

  Assesses the validity or trustworthiness of a pattern.

  confidence is a certainty measure

  confidence (A=>B) = # tuples containing both A and B / # tuples containing A

  A confidence of 85% for the rule buys(X, "computer")=>buys(X,"software") means that 85% of all customers who purchased a computer also bought software

Interestingness measure

- Utility (support)

  usefulness of a pattern

  support (A=>B) = # tuples containing both A and B
  total # of tuples

  A support of 30% for the previous rule means that 30% of all customers purchased both a computer and software.

- Association rules that satisfy both the minimum confidence and support threshold are referred to as **strong association rules.**

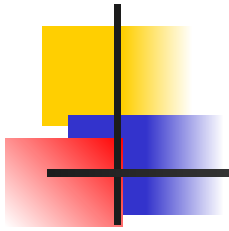
- Novelty

  Patterns contributing new information to the given pattern set are called novel patterns (example: Data exception).

  removing redundant patterns is a strategy for detecting novelty.

Strong association rules

- Association rules that satisfy both a user-specified ***minimum confidence threshold*** and user-specified ***minimum support threshold*** are referred to as ***strong association rules*** and are considered interesting.



Interestingness measure : Novelty

Suppose, the following association rules were mined from the AllElectronics database

location(X, “canada”) => buys (X, “SONY_TV”) [8%, 70%]

location(X, “Montreal”) => buys (X, “SONY_TV”) [2%, 71%]

The first rule is more general than the second rule and therefore we expect the first rule to occur more frequently than the second rule.

Measurements of Pattern Interestingness

- **Simplicity**
  - If the rule structure is complex – likely less interesting
  - (association) rule length
  - (decision) tree size

- **Certainty**
  - Measure for association "A=>B"
    - confidence = P(A|B) = n(A and B)/ n (B)
  - Classification
    - reliability or accuracy, certainty factor, rule strength, rule quality, discriminating weight, etc.

Measurements of Pattern Interestingness

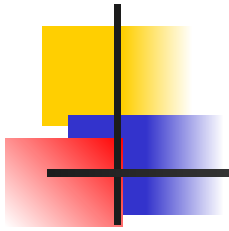
- **Utility**
  - potential usefulness,
  - support (association), noise threshold (description)

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

  Eg:

  Location (X, ”Canada”)=>buys (X,”Sony_TV”) [8%,70%]

  Location (X, ” Vancouver”)=>buys (X,”Sony_TV”) [2%,71%]



Data Visualization

For data mining to be effective, data mining systems should be able to display the discovered patterns in multiple forms, such as rules, tables, crosstabs (cross-tabulations), pie or bar charts, decision trees, cubes, or other visual representations.

User must be able to specify the forms of presentation to be used for displaying the discovered patterns.

Visualization of Discovered Patterns

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

Visualization

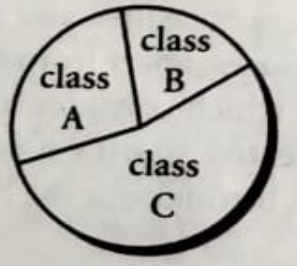
Rules

age(X, "young") and income(X, "high") => class(X, "A")
age(X, "young") and income(X, "low") => class(X, "B")
age(X, "old") => class(X, "C")

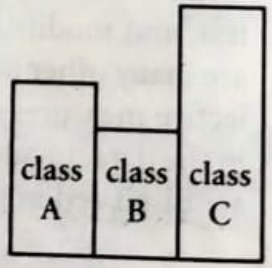
Table

age	income	class	count
young	high	A	1,402
young	low	B	1,038
old	high	C	786
old	low	C	1,374

Pie chart



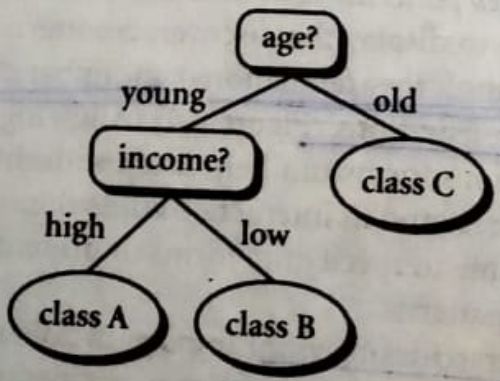
Bar chart



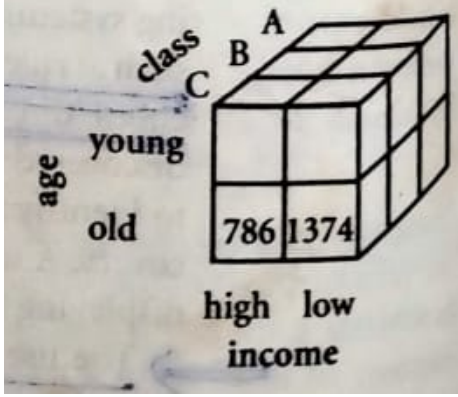
Crosstab

	income		class		
age	high	low	A	B	C
young	1,402	1,038	1,402	1,038	0
old	786	1,374	0	0	2,160
count	2,188	2,412	1,402	1,038	2,160

Decision tree



Data cube



Data Mining Task Primitives

Task relevant data

- Database Name
- Database tables
- Relevant attributes
- Data grouping criteria

Type of knowledge to be mined

- Classification
- Clustering
- Prediction
- Discrimination
- Correlation analysis

Background knowledge

- Concept Hierarchy
- User beliefs about relationships in data

Measures of patterns

- Simplicity
- Novelty
- Certainty
- Utility

Visualization of patterns

- Visualization of discovered patterns
- Cubes
- Charts
- Tables
- Graphs