UNIT III

Data Mining primitives, languages & system architecture

- Data Mining primitives:
  - Task relevant data
  - kind of knowledge to be mined
  - Background knowledge
  - Interestingness measures
  - presentation & visualization of discovered pattern -
- Data Mining Query language
  - Designing Graphical User interfaces based on DMQL
  - Architecture of Data mining

What Defines a Data Mining Task ?

- Task-relevant data
  - Typically interested in only a subset of the entire database
  - Specify
    - the name of database/data warehouse (AllElectronics_db)
    - names of tables/data cubes containing relevant data (item, customer, purchases, items_sold)
    - conditions for selecting the relevant data (purchases made in Canada for relevant year)
    - relevant attributes or dimensions (name and price from item, income and age from customer)

What Defines a Data Mining Task ? (continued)

- Type of knowledge to be mined
  - Concept description, association, classification, prediction, clustering, and evolution analysis
    - Studying buying habits of customers, mine associations between customer profile and the items they like to buy
      - Use this info to recommend items to put on sale to increase revenue
    - Studying real estate transactions, mine clusters to determine house characteristics that make for fast sales
      - Use this info to make recommendations to house sellers who want/need to sell their house quickly
    - Study relationship between individual's sport statistics and salary
      - Use this info to help sports agents and sports team owners negotiate an individual's salary

What Defines a Data Mining Task ? (continued)

- Type of knowledge to be mined
  - Pattern templates that all discovered patterns must match
  - P(X:Customer, W) and Q(X, Y) => buys(X, Z)
    - X is key of customer relation
    - P & Q are predicate variables, instantiated to relevant attributes
    - W & Z are object variables that can take on the value of their respective predicates
  - Search for association rules is confined to those matching some set of rules, such as:
    - Age(X, “30..39”) & income (X, “40K..49K”) => buys (X, “VCR”) [2.2%, 60%]
    - Customers in their thirties, with an annual income of 40-49K, are likely (with 60% confidence) to purchase a VCR, and such cases represent about 2.2% of the total number of transactions

What Defines a Data Mining Task ?

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

Task-Relevant Data (Minable View)

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

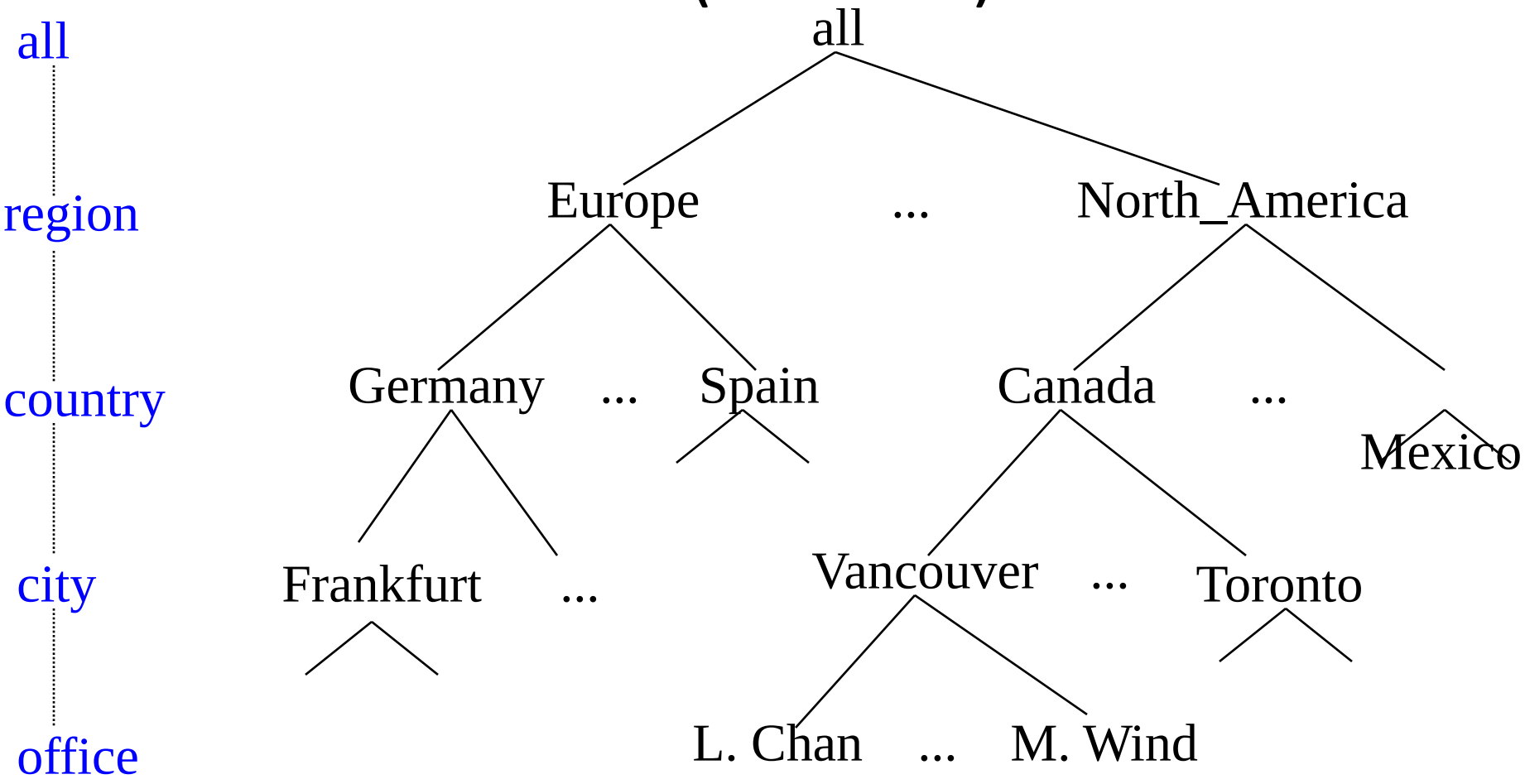
Types of knowledge to be mined

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

Background Knowledge: Concept Hierarchies

- Allow discovery of knowledge at multiple levels of abstraction
- Represented as a set of nodes organized in a tree
  - Each node represents a concept
  - Special node, all, reserved for root of tree
- Concept hierarchies allow raw data to be handled at a higher, more generalized level of abstraction
- Four major types of concept hierarchies, schema, set-grouping, operation derived, rule based

A Concept Hierarchy: Dimension (location)



Define a sequence of mappings from a set of low level concepts to higher-level, more general concepts

Background Knowledge: Concept Hierarchies

- Schema hierarchy – total or partial order among attributes in the database schema, formally expresses existing semantic relationships between attributes
  - Table address
    - create table address (street char (50), city char (30), province_or_state char (30), country char (40));
  - Concept hierarchy location
    - street < city < province_or_state < country
- Set-grouping hierarchy – organizes values for a given attribute or dimension into groups or constant range values
  - {young, middle_aged, senior} subset of all(age)
    - {20-39} = young
    - {40-59} = middle_aged
    - {60-89} = senior

Background Knowledge: Concept Hierarchies

- Operation-derived hierarchy – based on operations specified by users, experts, or the data mining system
  - email address or a URL contains hierarchy info relating departments, universities (or companies) and countries
  - E-mail address
    - dmbook@cs.sfu.ca
  - Partial concept hierarchy
    - login-name < department < university < country

Background Knowledge: Concept Hierarchies

- Rule-based hierarchy – either a whole concept hierarchy or a portion of it is defined by a set of rules and is evaluated dynamically based on the current data and rule definition
  - Following rules used to categorize items as low profit margin, medium profit margin and high profit margin
    - Low profit margin - < $50
    - Medium profit margin – between $50 & $250
    - High profit margin - > $250
  - Rule based concept hierarchy
    - low_profit_margin (X) <= price(X, P1) and cost (X, P2) and (P1 - P2) < $50
    - medium_profit_margin (X) <= price(X, P1) and cost (X, P2) and (P1 - P2) >= $50 and (P1 – P2) <= $250
    - high_profit_margin (X) <= price(X, P1) and cost (X, P2) and (P1 - P2) > $250

Measurements of Pattern Interestingness

- After specification of task relevant data and kind of knowledge to be mined, data mining process may still generate a large number of patterns
- Typically, only a small portion of these patterns will actually be of interest to a user
- The user needs to further confine the number of uninteresting patterns returned by the data mining process
  - Utilize interesting measures
- Four types: simplicity, certainty, utility,

Measurements of Pattern Interestingness (continued)

- Simplicity – A factor contributing to interestingness of pattern is overall simplicity for comprehension
  - Objective measures viewed as functions of the pattern structure or number of attributes or operators
  - More complex a rule, more difficult it is to interpret, thus less interesting
  - Example measures: rule length or number of leaves in a decision tree
- Certainty – Measure of certainty associated with pattern that assesses validity or trustworthiness
  - Confidence (A=>B) = # tuples containing both A & B/ #tuples containing A
  - Confidence of 85% for association rule buys (X, computer) => buys (X, software) means 85% of all customers who bought a computer bought software also

Measurements of Pattern Interestingness (continued)

- Utility – potential usefulness of a pattern is a factor determining its interestingness
  - Estimated by a utility function such as support – percentage of task relevant data tuples for which pattern is true
    - Support (A=>B) = # tuples containing both A & B/ total # of tuples
- Novelty – those patterns that contribute new information or increased performance to the pattern set
  - not previously known, surprising

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.

A Data Mining Query Language (DMQL)

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

Syntax for DMQL

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

Syntax for task-relevant data specification

- *use database* database_name, or *use data warehouse* data_warehouse_name
  - directs the data mining task to the database or data warehouse specified
- *from relation*(s)/cube(s) [*where* condition]
  - specify the database tables or data cubes involved and the conditions defining the data to be retrieved
- *in relevance to* att_or_dim_list
  - Lists attributes or dimensions for exploration

Syntax for task-relevant data specification

- *order by* order_list
  - Specifies the sorting order of the task relevant data
- *group by* grouping_list
  - Specifies criteria for grouping the data
- *having* condition
  - Specifies the condition by which groups of data are considered relevant

Top Level Syntax of DMQL

- *(DMQL) ::= (DMQL_Statement);{(DMQL_Statement)*
- *(DMQL_Statement) ::= (Data_Mining_Statement) | (Concept_Hierarchy_Definition_Statement) | (Visualization_and_Presentation)*

Top Level Syntax of DMQL (continued)

- *(Data_Mining_Statement) ::=* ***use database*** *(database_name)* | ***use data warehouse*** *(data_warehouse_name)* {***use hierarchy*** *(hierarchy_name)* ***for*** *(attribute_or_dimension)}* *(Mine_Knowledge_Specification)* ***in relevance to*** *(attribute_or_dimension_list)* ***from*** *(relation(s)/cube(s))* [***where*** *(condition)]* [***order by*** *(order_list)]* [***group by*** *(grouping_list)]* [***having*** *(condition)]* {***with*** *[(interest_measure_name)]* ***threshold*** *= (threshold_value)* [***for*** *(attribute(s))]}

Top Level Syntax of DMQL (continued)

- *(Mine_Knowledge_Specification) ::= (Mine_Char) | (Mine_Desc) | (Mine_Assoc) | (Mine_Class)*
- *(Mine_Char) ::=* ***mine characteristics*** *[****as*** *(pattern_name)]* ***analyze*** *(measure(s))*
- *(Mine_Desc) ::=* ***mine comparison*** *[****as*** *(pattern_name)]* ***for*** *(target_class)* ***where*** *(target_condition)* *{****versus*** *(contrast_class_i)* ***where*** *(contrast_condition_i)]* ***analyze*** *(measure(s))*
- *Mine_Assoc) ::=* ***mine association*** *[****as*** *(pattern_name)]* *[****matching*** *(metapattern)]*

Top Level Syntax of DMQL (continued)

- *(Mine_Class) ::=* ***mine classification*** *[****as*** *(pattern_name)]* ***analyze*** *(classifying_attribute_or_dimension)*
- *(Concept_Hierarchy_Definition_Statement) ::=* ***define hierarchy*** *(hierarchy_name) [****for*** *(attribute_or_dimension)] **on** (relation_or_cube_or_hierarchy) **as** (hierarchy_description) [****where*** *(condition)]*
- *(Visualization_and_Presentation) ::=* ***display as*** *(result_form) | {(Multilevel_Manipulation)}*

Top Level Syntax of DMQL (continued)

- *(Multilevel_Manipulation) ::=* ***roll***
  up on *(attribute_or_dimension)* | ***drill***
  down on *(attribute_or_dimension)* | ***add***
  (attribute_or_dimension) | ***drop***
  (attribute_or_dimension)

Specification of task-relevant data

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

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

□

Syntax for specifying the kind of knowledge to be mined

- Characterization

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

  – Specifies that characteristic descriptions are to be mined
  – ***Analyze*** specifies aggregate measures
  – Example: mine characteristics as customerPurchasing analyze count%

Syntax for specifying the kind of knowledge to be mined

- Discrimination

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

  - Specifies that discriminant descriptions are to be mined, compare a given target class of objects with one or more contrasting classes (thus referred to as comparison)
  - ***Analyze*** specifies aggregate measures
  - Example: mine comparison as purchaseGroups for bigSpenders where avg(I.price) >= $100 versus budgetSpenders where avg(I.price) < $100 analyze count

Syntax for specifying the kind of knowledge to be mined

- Association

  Mine_Knowledge_Specification ::=
  mine associations [as pattern_name]
  [matching (metapattern)]

  – Specifies the mining of patterns of association
  – Can provide templates (metapattern) with the matching clause
  – Example: mine associations as buyingHabits matching P(X: customer, W) and Q(X, Y) => buys (X,Z)

Syntax for specifying the kind of knowledge to be mined (cont.)

- Classification

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

  - Specifies that patterns for data classification are to be mined
  - Analyze clause specifies that classification is performed according to the values of (classifying_attribute_or_dimension)
  - For categorical attributes or dimensions, each value represents a class (such as low-risk, medium risk, high risk)

Syntax for concept hierarchy specification

- To specify what concept hierarchies to use

  use hierarchy **<hierarchy>** for **<attribute_or_dimension>**
- We use different syntax to define different type of hierarchies
  - schema hierarchies

    define hierarchy **time_hierarchy** on **date** as **[date,month quarter,year]**
  - set-grouping hierarchies

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

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

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

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

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

Syntax for concept hierarchy specification (Cont.)

- operation-derived hierarchies

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

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

- rule-based hierarchies

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

  level_1: low_profit_margin < level_0: all

  if (price - cost)< $50

  level_1: medium-profit_margin < level_0: all

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

  level_1: high_profit_margin < level_0: all

Syntax for interestingness measure specification

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

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

- **Example:**

  with support threshold **= 0.05**

  with confidence threshold **= 0.7**

Syntax for pattern presentation and visualization specification

- We have syntax which allows users to specify the display of discovered patterns in one or more forms

  display as **<result_form>**

  Result_form = Rules, tables, crosstabs, pie or bar charts, decision trees, cubes, curves, or surfaces

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

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

Putting it all together: the full specification of a DMQL query

use database **AllElectronics_db**

use hierarchy **location_hierarchy for B.address**

mine characteristics as **customerPurchasing**

analyze **count%**

in relevance to **C.age, I.type, I.place_made**

from **customer C, item I, purchases P, items_sold S, works_at W, branch**

where **I.item_ID = S.item_ID and S.trans_ID = P.trans_ID**

and P.cust_ID = C.cust_ID and P.method_paid = ``AmEx''

and P.empl_ID = W.empl_ID and W.branch_ID = B.branch_ID and B.address = ``Canada'' and I.price >= 100

with **noise** threshold = **0.05**

display as **table**

Other Data Mining Languages & Standardization Efforts

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

Designing Graphical User Interfaces based on a data mining query language

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

Data Mining System Architectures

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

Summary

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