UNIT-3 Data Mining Primitives, Languages, and System Architectures

Lecture Topic

****

Lecture-18 Data mining primitives: What defines a data mining task?

Lecture-19 A data mining query language

Lecture-20 Design graphical user interfaces based on a data mining query language

Lecture-21 Architecture of data mining systems

Lecture-18

Data mining primitives: What defines a data mining task?

Why Data Mining Primitives and Languages?

- Finding all the patterns autonomously in a database? — 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
- 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

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

- Schema hierarchy
  - street < city < province_or_state < country
- Set-grouping hierarchy
  - {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

Measurements of Pattern Interestingness

- Simplicity

  association rule length, decision tree size

- Certainty

  confidence, P(A|B) = n(A and B)/ n (B), classification reliability or accuracy, certainty factor, rule strength, rule quality, discriminating weight

- Utility

  potential usefulness, support (association), noise threshold (description)

- Novelty

  not previously known, surprising (used to remove redundant rules, Canada vs. Vancouver rule implication support ratio

Visualization of Discovered Patterns

- Different backgrounds/usages may require different forms of representation
  - rules, tables, cross tabs, pie/bar chart
- 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

Lecture-19

A data mining query language

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
    - 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

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

    — a DMQL query

Syntax for task-relevant data specification

- use database database_name, or use data warehouse data_warehouse_name
- from relation(s)/cube(s) [where condition]
- in relevance to att_or_dim_list
- order by order_list
- group by grouping_list
- having condition

Syntax for specifying the kind of knowledge to be mined

- Characterization

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

- 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)

- Association

  Mine_Knowledge_Specification ::=
  mine associations [as pattern_name]

Syntax for specifying the kind of knowledge to be mined

- Classification

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

- Prediction

  Mine_Knowledge_Specification ::=
  mine prediction [as pattern_name]
  analyze prediction_attribute_or_dimension
  {set {attribute_or_dimension_i= value_i}}

Syntax for concept hierarchy specification

- To specify what concept hierarchies to use

  use hierarchy **<hierarchy>** for **<attribute_or_dimension>**

- 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

– operation-derived hierarchies

define hierarchy age_hierarchy for age on customer as

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

Syntax for concept hierarchy specification

– 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

if (price - cost) > $250

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

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

    display as <result_form>

- 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

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

Lecture-20

Design graphical user interfaces based on a data mining query language

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
  - Interactive multilevel mining
  - Other miscellaneous information

Lecture-21

Architecture of data mining systems

Data Mining System Architectures

- Coupling data mining system with DB/DW system
  - No coupling—flat file processing,
  - 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- sorting, indexing, aggregation, histogram analysis, multiway join, precomputation of some stat functions

Data Mining System Architectures

- 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