SQL-Like Language For Database Mining

Tadeusz Morzy, Maciej Zakrzewicz
Institute of Computing Science
Poznań University of Technology, Poland
morzy@put.poznan.pl
mzakrz@cs.put.poznan.pl

Presentation Outline

- Data Mining: Basic Problems, Knowledge, Process
- Overview of MineSQL Language
- MineSQL By Examples
- Concluding Remarks

Data Mining As a Research Area

- discovery of useful information (knowledge) from large datasets (relational databases)
- also referred to as: Database Mining, Knowledge Discovery in Databases (KDD)
- practical applications:
  - Database Marketing
  - Fraud Detection
  - Credit Scoring
  - Client Profiling and Segmentation

Knowledge Representation - Rules

- discovered knowledge is usually represented by means of rules

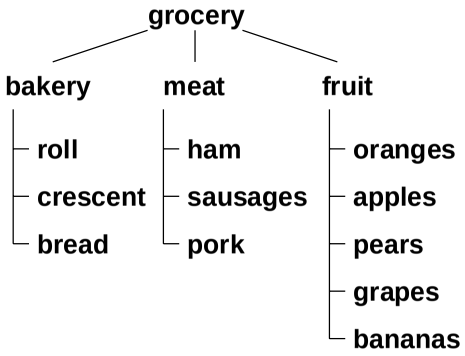
```
product='bread' & product='milk' -> product='butter'
```

- left hand side of a rule is called body, right hand side is called head
- each rule has two associated measures of statistical significance and strength: support and confidence
- data mining distinguishes association rules, classification rules, characteristic rules, discriminant rules, generalized rules, multiple-level rules etc.
- database records can satisfy or violate a given rule

Background Knowledge - Taxonomies

- domain experts can provide additional, background knowledge about the subject of exploration
- background knowledge employed by a rule discovery algorithm helps in finding stronger and simpler rules
- background knowledge is often represented by means of conceptual hierarchies (taxonomies) and user-defined attributes (virtual attributes)
- rules discovered with help of conceptual hierarchies are called generalized or multiple-level rules

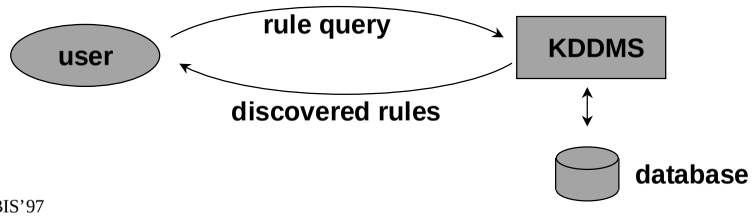
Taxonomy Example




```
product='bakery' & product='fruit' -> product='meat'
product='pork' -> product='bakery'
```


Interactive Knowledge Discovery Process

- user specifies rule query - request to discover rules that satisfy given constraints
- KDD Management System (KDDMS) uses rule generation algorithms to discover rules satisfying the user constraints
- discovered rules are returned to the user as a result of his rule query
- receiving the result, the user may decide to modify his constraints (query) and to discover rules again



Overview of MineSQL Language

- uniform API (Application Programming Interface) for building business applications dealing with knowledge discovery
- declarative, SQL-like language
- allows rule discovery, storage and manipulation
- support for conceptual hierarchies and generalized and multiple-level rules
- new datatype for storing rules in relational databases

RULE Datatype

- rules can be integrated into existing databases of structured data
- structured data and rules can be searched in a single query
- each RULE consists of a set of body elements, a set of head elements, support value and confidence value
- rule functions provide access to rule elements (body, head, support, confidence) as well as convert values between RULE and other datatypes
- two rule operators (SATISFIED BY, VIOLATED BY) allow joining rules with records or groups of records stored in database tables

Examples of Rule Functions

- support(*r*) - returns the support value of the rule *r*,
- confidence(*r*) - returns the confidence value of the rule *r*,
- body(*r*) - returns the rule *r* body,
- head(*r*) - returns the rule *r* head,
- bodylen(*r*) - returns the number of rule *r* body elements,
- headlen(*r*) - returns the number of rule *r* head elements,
- rulelen(*r*) - returns the number of all rule *r* body and head elements,
- to_char(*r*, *fmt*) - converts a value *r* of RULE datatype to a value of CHAR datatype, using the format string *fmt*,
- to_rule(*char*, *fmt*) - converts a value *char* of CHAR datatype to a value of RULE datatype, using the format string *fmt*,

Discovery Statement - MINE

```
MINE rule_expr [[AS] alias][,...]
[FOR {data_expr [USING tax_name][AS alias][,...]|*}]
[TO  {data_expr [USING tax_name][AS alias][,...]|*}]
FROM table [,table]...
[WHERE {data_condition|rule_condition}
 [{AND|OR} {data_condition|rule_condition}]...]
[GROUP BY data_expr [,data_expr]...
 [HAVING data_condition]]
[ORDER BY rule_expr [{ASC|DESC}][,...]]
```


Example Table For Exploration

trans_id	customer	product	date	day	hour	quantity
1	100	roll	30-12-96	Monday	9:20	6
1	100	crescent	30-12-96	Monday	9:20	3
1	100	pork	30-12-96	Monday	9:20	5
2	100	roll	30-12-96	Monday	11:10	3
2	100	crescent	30-12-96	Monday	11:10	2
2	100	sausage	30-12-96	Monday	11:10	10
2	100	apple	30-12-96	Monday	11:10	20
2	100	pork	30-12-96	Monday	11:10	10
3	101	bread	30-12-96	Monday	12:40	1
3	101	ham	30-12-96	Monday	12:40	1
3	101	oranges	30-12-96	Monday	12:40	10
4	102	apple	02-01-97	Thursday	16:00	5
4	102	pears	02-01-97	Thursday	16:00	5
4	102	pork	02-01-97	Thursday	16:00	5
4	102	crescent	02-01-97	Thursday	16:00	2
5	103	bananas	03-01-97	Friday	9:50	3
5	103	oranges	03-01-97	Friday	9:50	7

Mining of Simple Association Rules

```
MINE rule, support(rule) AS s., confidence(rule) AS c.
FOR product
FROM shoppings
WHERE 'product=''crescent''' IN body(rule)
AND bodylen(rule) = 2
AND support(rule) > 0.2
GROUP BY trans_id
```

```
rule                                                s.  c.
--------------------------------------------------- --- ---
product='roll' & product='crescent'->product='pork'  0.4 1.0
product='crescent' & product='pork'->product='roll'  0.4 0.7
```


Generalized And Multiple-Level Rules (1/2)

```
CREATE TAXONOMY supermarket_taxonomy
(NODE 'grocery',
 NODE 'bakery' REFERENCES 'grocery',
 LEAF 'roll' REFERENCES 'bakery',
 LEAF 'crescent' REFERENCES 'bakery',
 LEAF 'bread' REFERENCES 'bakery')
```

```
MINE rule, suport(rule) AS s., confidence(rule) AS c.
FOR product USING supermarket_taxonomy
FROM shoppings
WHERE confidence(rule) > 0.8
  AND support(rule) > 0.5
GROUP BY trans_id
ORDER BY support(rule) DESC
```


Generalized And Multiple-Level Rules (2/2)

```
rule                                                          s.  c.
------------------------------------------------------------- --- ---
product='bakery' -> product='meat'                            0.8 1.0
product='meat' -> product='bakery'                            0.8 1.0
product='roll' -> product='pork'                              0.6 1.0
product='pork' -> product='roll'                              0.6 1.0
product='bakery' -> product='fruit'                           0.6 1.0
product='bakery' & product='fruit' -> product='meat'          0.6 1.0
product='pork' -> product='bakery'                            0.6 1.0
product='crescent' -> product='meat'                          0.6 1.0
```


Storing Rules In Database Tables (1/2)

```
CREATE TABLE my_rules
(r RULE,
 description CHAR(20))
```

```
INSERT INTO my_rules (r, description)
MINE rule, 'first example'
FOR product, customer
TO time(hour) AS time
FROM shoppings
WHERE head(rule) = 'time=''morning'''
  AND support(rule) > 0.1
```

Storing Rules In Database Tables (2/2)

```
SELECT s.trans_id, s.customer, s.product
FROM shoppings s, my_rules m
WHERE m.r VIOLATED BY s.*
```

s.trans_id	s.customer	s.product
4	102	pork
4	102	crescent

Rule Discovery Issues

- need for fast and effective algorithms for on-demand rule discovery
- rule query processing in multi-user environment
- physical storage of rules and taxonomies, and its independence from logical representation

Concluding Remarks

- Data mining process is interactive and iterative
- Rule queries are processed by a KDD Management System
- MineSQL is a query language that allows expressing the common data mining problems
- Discovered rules can be stored for future use
- Rule query processing poses many new interesting research problems