From SQL to RQL
or how to re-use SQL techniques for pattern mining

Jean-Marc Petit
INSA/Université de Lyon, CNRS

Joint work with: Brice Chardin, Emmanuel Coquery, Christine Froidevaux, Marie Pailloux (Agier), Jef Wijsen ... and Einoshin Suzuki!

Work partially funded by the French Research Agency (ANR), DAG project
RQL: A Query Language for discovering logical implications in DB
About logical implications

- Logical consequence (or entailment)
  - One of the most fundamental concepts in logic
  - Premises, Conclusion (If ... then ...): Reasoning using proofs and/or models
  - Examples
    - "If 2=3 then I am the queen of England"
    - right or wrong? Why?
    - "If 2=2 then I am the queen of England"
    - right or wrong? Why?

- Focus on a specific class of logical implications
  - Three properties to be verified
    - Reflexivity, augmentation, transitivity (Armstrong axioms)
About databases

- Relational databases systems **everywhere**!
- RDBMS market expected to **double** by 2016 [MarketResarch.com, Aug 2012]
- Query optimization: **Awesome**!

- **Simple goals**:
  - Query the data where they are
  - Use/extend DB languages for pattern mining problems (e.g. DMQL, MSQl, ...)

*April 15, 2014 Kyushu University*
Focus on pattern mining in DB
- patterns = logical implication, called rules hereafter
- DB = Relational DBs
- Quality measure: not considered

Pattern mining discovery seen as query processing
- Reusing optimization techniques from DB
RQL: contributions

« Original ideas

« Since then, what we have done ?
  SafeRL: a well-founded logical query language derived from Tuple Relational Calculus (TRC)
    not discussed in detail here
  RQL: Its practical counterpart derived from SQL
  A rewriting technique to use as much as possible the underlying DBMS
Getting started with RQL through examples

SafeRL and Query rewriting

RQL Web Application

Conclusions
Rules between attributes with NULL values in EMP?

FINDRULES
OVER workdept, mgrno
SCOPE t1 EMP
CONDITION ON $A IS t1.$A IS NULL

- Mgrno -> Workdept holds
- Workdept -> Mgrno does not
  => counter-example: Empno 30 (or 50)
Functional dependencies

Remember the definition:

$X \rightarrow Y$ holds in $r$ iff

$$\forall t_1, t_2 \in r, \text{ if } \forall A \in X, t_1[A] = t_2[A] \text{ then } \forall B \in Y, t_1[B] = t_2[B]$$

With RQL:

```
FINDRULES
OVER Lastname, Workdept, Job, Sex, Bonus
SCOPE t1, t2 Emp
CONDITION ON $A IS t1.$A = t2.$A
```
Variant of FDs: Conditional FDs

FINDRULES
OVER Lastname, Workdept, Job, Sex, Bonus
SCOPE t1, t2 (select * from Emp where educlevel > 16)
CONDITION ON $A IS t1.$A = t2.$A

sex -> bonus holds
i.e. « above a certain level of qualification, the gender determines the bonus »
Approximative FDs

FINDRULES
OVER Educlevel, Sal, Bonus, Comm
SCOPE t1, t2 EMP
CONDITION ON $A IS
\[2 \times \text{abs}(t1.$A - t2.$A)/(t1.$A + t2.$A) < 0.1\]

Sal -> Comm holds
i.e. « employees earning similar salaries receive similar commissions »
Sequential FDs

FINDRULES
OVER Educlevel, Sal, Bonus, Comm
SCOPE t1, t2 EMP
CONDITION ON $A t1.$A >= t2.$A

Sal -> Comm and Sal -> Comm hold
i.e. « higher salary is equivalent to higher commission »
Conditional sequential FDs

FINDRULES
OVER Educlevel, Sal, Bonus, Comm
SCOPE t1, t2 (select * from EMP where Sex = 'M')
CONDITION ON $A t1.$A >= t2.$A

EducLevel -> Bonus holds
i.e. « male employees with higher education levels receive higher bonus »
Another kind of « FD »

FINDRULES

OVER Educlevel, Sal, Bonus, Comm

SCOPE t1, t2 EMP

WHERE t1.empno = t2.mgrno

CONDITION ON $A t1.$A >= t2.$A

{} -> Bonus holds

i.e. « managers always earn a bonus greater than or equal to their employees»
Another example (1/2)

Example from gene expression data

Assume tuples are ordered

Idea: adapting FD for catching the evolution of attributes between two consecutive tuples

\[ X \Rightarrow Y \text{ is satisfied in } r \text{ iff } \forall t_i, t_{i+1} \in r, \]
\[ \text{if } \forall g \in X, t_{i+1}[g] - t_i[g] \geq \varepsilon_1 \text{ then } \forall g \in Y, t_{i+1}[g] - t_i[g] \geq \varepsilon_1 \]
\[ \varepsilon_1 = 1.0 \]
Rules over genes

\[ X \Rightarrow Y \text{ is satisfied in } r \text{ iff } \forall \ t_i, t_{i+1} \in r, \]
if \( \forall g \in X, t_{i+1}[g] - t_i[g] \geq \varepsilon_1 \) then \( \forall g \in Y, t_{i+1}[g] - t_i[g] \geq \varepsilon_1 \)
\( \varepsilon_1 = 1.0 \)

FINDRULES
OVER \ g1,g2,g3,g4,g5,g6,g7,g8
SCOPE \ t1, t2 \ GENES
WHERE \ t2.time= t1.time+1
CONDITION ON A IS \ t2.A - t1.A \geq 1.0
Another kind of rules (2/2)

If expression level of $g_2$ grows between $t_i$ and $t_{i+1}$, then expression level of $g_4$ also grows.

But $r \not\models g_4 \Rightarrow g_2$

t2, t3 is a counter-example
FINDRULES
OVER g1,g2,g3,g4,g5,g6,g7,g8
SCOPE t1, t2, t3 Genes
WHERE t2.time=t1.time+1 AND t3.time=t2.time+1
CONDITION ON $A IS t1.$A < t2.$A AND t3.$A< t2.$A

=> three tuples variables needed to express a local maximum
Synthesis of RQL

- RQL: « look & feel » of SQL
  - Very simple and easy to use by SQL analysts

- Powerful query language
  - Allow interactions with data analysts
  - Powerful tool, need some practice to get fluent with ...

- Can be used
  - To generate the rules (if schema permits)
  - To test whether or not a given rule holds
    - If yes, just say « Yes » 😊
    - Otherwise, find a counter-examples in the data and refine your query
Motivating examples

SafeRL and Query rewriting

RQL Web Application

Conclusions
SafeRL: a query language for rules

SafeRL: a well-founded logical query language

\[ Q = \{ X \rightarrow Y \mid \forall t_1 \ldots \forall t_n (\psi(t_1, \ldots, t_n) \land (\forall A \in X(\delta(A, t_1, \ldots, t_n)) \rightarrow \forall A \in Y (\delta(A, t_1, \ldots, t_n))))\} \]

- Syntax + semantics not detailed here: cf papers

- Every SafeRL query \( Q \) defines rules “equivalent to” FD or implications

- Result: There is a closure system \( C(Q) \) associated to \( Q \)
Contribution: Query rewriting

- A base $B$ of a closure system $C$ is such that
  - $\text{Irreducible}(C) \subseteq B \subseteq C$

- Main result (cf LML 2013):
  Let $Q$ be a SafeRL query over a DB $d$

  THM: There exists a SQL query $Q'$ over $d$ such that $Q'$ computes a base $B$ of $C(Q)$, the closure system associated to $Q$
Base of a query: the data-centric step

From the base $B$ of $Q$ in $d$, we can get:

- The closure of an attribute set
- The canonical cover of satisfied rules
- The cover of Gotlob&Libkin of approximate rules

and we can decide whether or not a given rule is satisfied

- If not, a counter example from $d$ can be provided

- Nothing new here, cf related works
Motivating examples
SafeRL and Query rewriting
RQL Web Application
Conclusions
Architecture

RQL query → Rules

Rule Generator

RQL Parser

SQL Generator

Base

SQL query → Optimizer

Query processor

DB

DBMS
RQL Web application

- Open to registered users (simple application form)
  - dedicated Oracle user, 200Ko quota

- Web Framework For Java
  - Play Framework (http://www.playframework.com/)
  - DBMS: Oracle v11 (+ MySQL)
  - + specific development in C++, Java, C (Uno’s code)

- Two modes: Sample (predefined schema) and SandBox (user schema)

- Try it out! http://rql.insa-lyon.fr
Submit your RQL or SQL query:

```
FINDRULES
OVER Edulevel, Sal, Bonus, Comm
SCAPE t1, t2 (SELECT ' FROM Emp WHERE Sex='M')
CONDITION ON A IS t1.A >= t2.A
```

SQL examples:
- SQL 1: Content of Dept
- SQL 2: Content of Emp
- SQL 3: List of tables
- SQL 4: Schema of Emp
- SQL 5: Definition of EMP_SUBSET

RQL examples:
- RQL 1: Functional dependencies on Emp
- RQL 2: Functional dependencies on a subset of Emp
Rule verification:
The rule Sal Educlevel → Bonus is false

Counter-example:

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>LASTNAME</th>
<th>WORKDEPT</th>
<th>JOB</th>
<th>EDUCLEVEL</th>
<th>SEX</th>
<th>SAL</th>
<th>BONUS</th>
<th>COMM</th>
<th>MGRNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>SPEN</td>
<td>C01</td>
<td>FINANCE</td>
<td>18</td>
<td>F</td>
<td>52750</td>
<td>500</td>
<td>4220</td>
<td>20</td>
</tr>
<tr>
<td>20</td>
<td>THOMP</td>
<td>null</td>
<td>MANAGER</td>
<td>18</td>
<td>M</td>
<td>41250</td>
<td>800</td>
<td>3300</td>
<td>null</td>
</tr>
</tbody>
</table>

Generated query:

1. SELECT t1.*, t2.*
2. FROM Emp t1, Emp t2
3. WHERE (t1.Sal >= t2.Sal AND t1.Educlevel >= t2.Educlevel)
4. AND CASE WHEN (t1.Bonus >= t2.Bonus) THEN 1 ELSE 0 END = 0
5. AND rownum <= 1
Motivating examples

RQL Web Application

SafeRL and Query rewriting

Conclusions

Outline
Conclusion

- From a logical query language for rules to the practical language RQL
  - Easy to use by SQL-aware analysts
  - No discretization
  - Promoting query processing techniques in pattern mining

- RQL: a practical Web application
  - For teaching
  - For research

- Future works: Data exploration with RQL through counter-examples
Query Rewriting for Rule Mining in Databases.  
