Bridging the Gap between Data Diversity and Data Dependencies

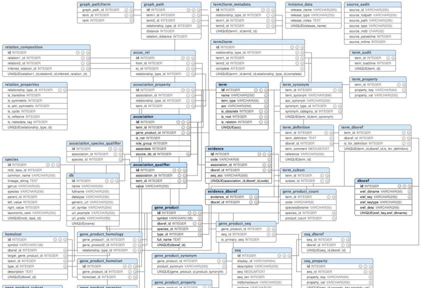
Jean-Marc Petit

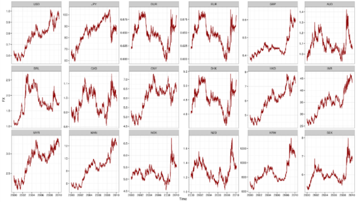
INSA Lyon, Université de Lyon
LIRIS CNRS (UMR 5205)

24th International Symposium on Methodologies for Intelligent Systems (ISMIS 2018)
Limassol, Cyprus

Data diversity

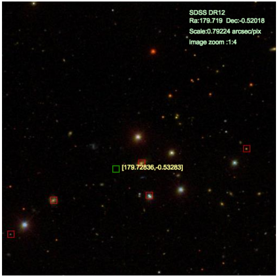
Data Diversity: not only a gender question !





Example from the astrophysics domain

The Sloan Digital Sky Survey (SDSS): Mapping the Universe !



Class	$\frac{u}{err_u}$	$\frac{g}{err_g}$	$\frac{r}{err_r}$	$\frac{i}{err_i}$	$\frac{z}{err_z}$
STAR	16.56	14.62	13.94	13.79	13.48
	0.01	0.00	0.01	0.01	0.00
Galaxie	19.79	17.77	16.59	16.07	15.63
	0.06	0.01	0.00	0.00	0.01
STAR	15.64	14.04	14.57	12.83	13.12
	0.01	0.00	0.01	0.00	0.01
Galaxie	21.61	20.81	19.87	19.30	19.03
	0.15	0.04	0.02	0.02	0.05
STAR	20.09	17.28	15.79	14.31	13.49
	0.04	0.00	0.00	0.00	0.00

5 magnitudes (u, g, r, i, and z) catalog database
$\Rightarrow$ **Require to deal with numerical interval data as first class citizen**

See `http://www.sdss.org/dr12/` for details

Data and metadata from SDSS

SDSS J115851.16-002903.1

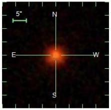
Look up common name

Type	run	rerun	camcol	field	obj	SDSS Object ID
STAR	756	301	2	427	250	1237648720693756154

RA, Dec		Galactic Coordinates (l, b)	
Decimal	Sexagesimal	l	b
179.713207494, -0.484210574	11:58:51.16, -00:29:03.15	276.198683367	59.630008868

Imaging

Flags	DEBLENDED_AT_EDGE STATIONARY BINNED1 INTERP CHILD



Magnitudes				
u	g	r	i	z
20.09	17.28	15.79	14.31	13.49

Magnitude uncertainties				
err_u	err_g	err_r	err_i	err_z
0.04	0.00	0.00	0.00	0.00

Image MJD	mode	Other observations	parentID	nChild	extinction_r	PetroRad_r (arcsec)
51259	PRIMARY	2	1237648720693756153	0	0.07	1.22 ± 0.008

Mjd-Date	photoZ (KD-tree method)	Galaxy Zoo 1 morphology
03/22/1999	-	-

SDSS J115852.32-003148.7

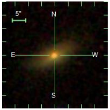
Look up common name

Type	run	rerun	camcol	field	obj	SDSS Object ID
GALAXY	756	301	2	427	259	1237648720693756163

RA, Dec		Galactic Coordinates (l, b)	
Decimal	Sexagesimal	l	b
179.718015380, -0.530204150	11:58:52.32, -00:31:48.73	276.243764984	59.589779466

Imaging

Flags	DEBLENDED_AT_EDGE STATIONARY BINNED1 INTERP COSMIC_RAY NOPETRO CHILD



Magnitudes				
u	g	r	i	z
19.79	17.77	16.59	16.07	15.63

Magnitude uncertainties				
err_u	err_g	err_r	err_i	err_z
0.06	0.01	0.00	0.00	0.01

Image MJD	mode	Other observations	parentID	nChild	extinction_r	PetroRad_r (arcsec)
51259	PRIMARY	5	1237648720693756162	0	0.06	6.00 ± 0.610

Mjd-Date	photoZ (KD-tree method)	Galaxy Zoo 1 morphology
03/22/1999	0.130 ± 0.0127	Uncertain

SDSS J115843.49-003233.8

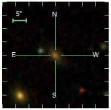
Look up common name

Type	run	rerun	camcol	field	obj	SDSS Object ID
GALAXY	756	301	2	426	465	1237648720693690833

RA, Dec		Galactic Coordinates (l, b)	
Decimal	Sexagesimal	l	b
179.681216944, -0.542727601	11:58:43.49, -00:32:33.81	276.187065846	59.563578544

Imaging

Flags	DEBLENDED_AT_EDGE STATIONARY MOVED BINNED1 CHILD



Magnitudes				
u	g	r	i	z
22.40	20.10	18.97	18.46	18.01

Magnitude uncertainties				
err_u	err_g	err_r	err_i	err_z
0.49	0.03	0.02	0.02	0.04

Image MJD	mode	Other observations	parentID	nChild	extinction_r	PetroRad_r (arcsec)
51259	PRIMARY	4	1237648720693690832	0	0.06	3.60 ± 0.304

Data diversity

To cope with data diversity, key notions have be studied for years in computer science:

- data and metadata representation,
- data uncertainty,
- data inconsistency,
- data heterogeneity ...

Dealing with data diversity remains the hardest thing in practise
⇒ Require to understand *what's hidden behind the data*:

- Where do they come from ? How are they produced ?

⇒ Be as close as possible of the available data sources and experts to better match their intended meaning

Data diversity

To cope with data diversity, key notions have be studied for years in computer science:

- data and metadata representation,
- data uncertainty,
- data inconsistency,
- data heterogeneity . . .

Dealing with data diversity remains the hardest thing in practise
$\Rightarrow$ Require to understand *what's hidden behind the data*:

- Where do they come from ? How are they produced ?

$\Rightarrow$ Be as close as possible of the available data sources and experts to better match their intended meaning

Data dependencies

Classical example of data dependencies: functional dependencies

$$r \models X \rightarrow Y \text{ iff for all } t_1, t_2 \in r$$

$$\textbf{If} \text{ for all } A \in X, t_1[A] = t_2[A] \textbf{ then} \text{ for all } B \in Y, t_1[B] = t_2[B]$$

Turns out to be a very general notion, related to implications.

a	b	$a \rightarrow b$
0	0	1
0	1	1
1	0	0
1	1	1

Many connections with lattice theory, formal concept analysis (Galois connection) and logics (see for ex [11])

Crucial to understand relational database design

Beyond database design

New and timely applications require some forms of FD:

- Data quality: Analysing existing data to identify data quality problems [17, 9]
- Machine learning over relational databases: FD-aware optimization for in-database learning [19]
- Semantic query optimization: Query rewriting techniques based on data dependencies [12]

⇒ Many extensions of FD have been proposed to take into account some forms of data diversity (e.g. see [10, 18] for a survey)

- Matching Dependencies, Denial constraints . . . [17, 9, 15]
- Implications in Formal Concept Analysis (FCA) [7, 6]
- Association rules . . . in Data mining [5]

Data diversity and data dependencies

Questions and Contributions

How to take into account data diversity for data dependencies ?
Does there exist unifying frameworks ?

Two contributions:

- RQL: a query language to express implications over relational databases (ISMIS 2005 [3], demo ICDM 2014 [13], TCS 2017 [14])
- Structural properties on attribute domains (ongoing work)

Contents

Important known results for FD

Let F be a set of FD over a schema R

$CL(F) = \{X \subseteq R | X_F^+ = X\}$: a **closure system** of F

$IRR(F)$ the set of **irreducible elements** of $CL(F)$ by intersection

Reasoning on F is **equivalent** to reasoning on $CL(F)$, for instance:

$$X_F^+ = \{A \in R \mid F \models X \to A\} = \cap\{Y \in CL(F) \mid X \subseteq Y\}$$

Let r be a relation over R.
The **agree set** of r is $ag(r) = \{ag(t_1, t_2) \mid t_1, t_2 \in r\}$ where $ag(t_1, t_2) = \{A \in R \mid t_1[A] = t_2[A]\}$

r is an Armstrong relation for F iff $IRR(F) \subseteq ag(r) \subseteq CL(F)$ [8]

Example

	$Bar(B)$	$Beer(Be)$	$Price(P)$
t_1	Nota bene	Adelscott	2
t_2	Montagne	1664	1.5
t_3	Nota bene	1664	2
t_4	Ritz	Adelscott	5
t_5	Café Flore	Affligen	6

$F = \{B \rightarrow P, P \rightarrow B\}$
$CL(F) = \{\emptyset, Be, BP, BBeP\}$
$IRR(F) = \{Be, BP\}$

$ag(r) = \{\emptyset, Be, BP\}$, often represented as:

B	Be	P
0	0	0
0	1	0
1	0	1

Towards a rule query language

Focus on rules equivalent to implications (or FD)
$\Rightarrow$ Armstrong axioms (reflexivity, augmentation, transitivity) have to be sound and complete

Idea: Defining a rule query language (RQL) such that every RQL statement turns out to deliver **implications**

Require to identify syntactic constraints such that we remain within the reasoning of implications

Semantics of implications

Let b_0 be a binary relation (given by a $\{0,1\}$-relation)

$$b_0 \models X \rightarrow Y \Leftrightarrow \forall t \in b_0$$
$$(\forall A \in X \ t.A = 1) \Rightarrow (\forall A \in Y \ t.A = 1)$$

Let $d = \{r_0, r_1, ..., r_n\}$ be a relational database

$$r_0 \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in r_0$$
$$(\forall A \in X \ t_1.A = t_2.A) \Rightarrow (\forall A \in Y \ t_1.A = t_2.A)$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_{i_0} \bowtie \ldots \bowtie r_{i_p}))$$
$$(\forall A \in X \ t_1.A = t_2.A) \Rightarrow (\forall A \in Y \ t_1.A = t_2.A)$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1 \in \pi_X(\sigma_F(r_{i_0} \bowtie \ldots \bowtie r_{i_n})),$$
$$\forall t_2 \in \pi_X(\sigma_{F'}(r_{j_0} \bowtie \ldots \bowtie r_{i_n}))$$
$$\text{such that } (t_1.rank = t_2.rank + 1)$$
$$(\forall A \in X \ t_1.A = t_2.A) \Rightarrow (\forall A \in Y \ t_1.A = t_2.A)$$

Semantics of implications

Let b_0 be a binary relation (given by a $\{0,1\}$-relation)

$$b_0 \models X \rightarrow Y \Leftrightarrow \forall t \in b_0$$
$$(\forall A \in X \; t.A = 1) \Rightarrow (\forall A \in Y \; t.A = 1)$$

Let $d = \{r_0, r_1, ..., r_n\}$ be a relational database

$$r_0 \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in r_0$$
$$(\forall A \in X \; t_1.A = t_2.A) \Rightarrow (\forall A \in Y \; t_1.A = t_2.A)$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_{i_0} \bowtie \ldots \bowtie r_{i_p}))$$
$$(\forall A \in X \; t_1.A = t_2.A) \Rightarrow (\forall A \in Y \; t_1.A = t_2.A)$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1 \in \pi_X(\sigma_F(r_{i_0} \bowtie \ldots \bowtie r_{i_n})),$$
$$\forall t_2 \in \pi_X(\sigma_{F'}(r_{j_0} \bowtie \ldots \bowtie r_{i_n}))$$
$$\text{such that } (t_1.rank = t_2.rank + 1)$$
$$(\forall A \in X \; t_1.A = t_2.A) \Rightarrow (\forall A \in Y \; t_1.A = t_2.A)$$

Semantics of implications

Let b_0 be a binary relation (given by a $\{0,1\}$-relation)

$$b_0 \models X \rightarrow Y \Leftrightarrow \forall t \in b_0$$
$$(\forall A \in X \;\; t.A = 1) \Rightarrow (\forall A \in Y \;\; t.A = 1)$$

Let $d = \{r_0, r_1, ..., r_n\}$ be a relational database

$$r_0 \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in r_0$$
$$(\forall A \in X \;\; t_1.A = t_2.A) \Rightarrow (\forall A \in Y \;\; t_1.A = t_2.A)$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_{i_0} \bowtie \ldots \bowtie r_{i_p}))$$
$$(\forall A \in X \;\; t_1.A = t_2.A) \Rightarrow (\forall A \in Y \;\; t_1.A = t_2.A)$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1 \in \pi_X(\sigma_F(r_{i_0} \bowtie \ldots \bowtie r_{i_n})),$$
$$\forall t_2 \in \pi_X(\sigma_{F'}(r_{j_0} \bowtie \ldots \bowtie r_{i_n}))$$
$$\text{such that } (t_1.rank = t_2.rank + 1)$$
$$(\forall A \in X \;\; t_1.A = t_2.A) \Rightarrow (\forall A \in Y \;\; t_1.A = t_2.A)$$

Semantics of implications

Let b_0 be a binary relation (given by a $\{0,1\}$-relation)

$$b_0 \models X \rightarrow Y \Leftrightarrow \forall t \in b_0$$
$$(\forall A \in X \;\; t.A = 1) \Rightarrow (\forall A \in Y \;\; t.A = 1)$$

Let $d = \{r_0, r_1, ..., r_n\}$ be a relational database

$$r_0 \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in r_0$$
$$(\forall A \in X \;\; t_1.A = t_2.A) \Rightarrow (\forall A \in Y \;\; t_1.A = t_2.A)$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_{i_0} \bowtie \ldots \bowtie r_{i_p}))$$
$$(\forall A \in X \;\; t_1.A = t_2.A) \Rightarrow (\forall A \in Y \;\; t_1.A = t_2.A)$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1 \in \pi_X(\sigma_F(r_{i_0} \bowtie \ldots \bowtie r_{i_n})),$$
$$\forall t_2 \in \pi_X(\sigma_{F'}(r_{j_0} \bowtie \ldots \bowtie r_{i_n}))$$
$$\text{such that } (t_1.rank = t_2.rank + 1)$$
$$(\forall A \in X \;\; t_1.A = t_2.A) \Rightarrow (\forall A \in Y \;\; t_1.A = t_2.A)$$

Semantics of implications (cont'ed)

$$d \models X \to Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_0 \bowtie \ldots \bowtie r_n))$$
$$(\forall A \in X(2 * ABS(t_1.A - t_2.A)/(t_1.A + t_2.A) < 0.1))$$
$$\Rightarrow (\forall A \in Y(2 * ABS(t_1.A - t_2.A)/(t_1.A + t_2.A) < 0.1))$$

$$d \models X \to Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_0 \bowtie \ldots \bowtie r_n))$$
$$(\forall A \in X\ t_1.A \leq t_2.A) \Rightarrow (\forall A \in Y\ t_1.A \leq t_2.A)$$

$$r_0 \models X \to Y \Leftrightarrow \forall t_1, t_2, t_3 \in r_0$$
$$(\forall A \in X(t_1.A \leq t_2.A) \wedge (t_3.A \leq t_2.A))$$
$$\Rightarrow (\forall A \in Y(t_1.A \leq t_2.A) \wedge (t_3.A \leq t_2.A))$$

Semantics of implications (cont'ed)

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_0 \bowtie \ldots \bowtie r_n))$$
$$(\forall A \in X(2 * ABS(t_1.A - t_2.A)/(t_1.A + t_2.A) < 0.1))$$
$$\Rightarrow (\forall A \in Y(2 * ABS(t_1.A - t_2.A)/(t_1.A + t_2.A) < 0.1))$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_0 \bowtie \ldots \bowtie r_n))$$
$$(\forall A \in X \;\; t_1.A \leq t_2.A) \Rightarrow (\forall A \in Y \;\; t_1.A \leq t_2.A)$$

$$r_0 \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2, t_3 \in r_0$$
$$(\forall A \in X(t_1.A \leq t_2.A) \wedge (t_3.A \leq t_2.A))$$
$$\Rightarrow (\forall A \in Y(t_1.A \leq t_2.A) \wedge (t_3.A \leq t_2.A))$$

Semantics of implications (cont'ed)

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_0 \bowtie \ldots \bowtie r_n))$$
$$(\forall A \in X(2 * ABS(t_1.A - t_2.A)/(t_1.A + t_2.A) < 0.1))$$
$$\Rightarrow (\forall A \in Y(2 * ABS(t_1.A - t_2.A)/(t_1.A + t_2.A) < 0.1))$$

$$d \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2 \in \pi_X(\sigma_F(r_0 \bowtie \ldots \bowtie r_n))$$
$$(\forall A \in X \;\; t_1.A \leq t_2.A) \Rightarrow (\forall A \in Y \;\; t_1.A \leq t_2.A)$$

$$r_0 \models X \rightarrow Y \Leftrightarrow \forall t_1, t_2, t_3 \in r_0$$
$$(\forall A \in X(t_1.A \leq t_2.A) \wedge (t_3.A \leq t_2.A))$$
$$\Rightarrow (\forall A \in Y(t_1.A \leq t_2.A) \wedge (t_3.A \leq t_2.A))$$

Approach and contribution

Replaying part of the story underlying SQL and relational languages, especially through Tuple Relational Calculus (TRC)

What we did:

- Extend TRC to support rule expression (SafeRL logical language, see [14] for details)
- Propose a new syntactic practical language (RQL) from SafeRL

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

Main result.

THM

Let Q be a RQL query over a database d.

1. $ans(Q, d)$ defines a closure system $CL(Q)$ over $sch(Q)$
2. There exists a SQL query Q' over d such that Q' computes a base $B(Q)$ of CL(Q), i.e. $IRR(Q) \subseteq B(Q) \subseteq CL(Q)$

- $B(Q)$: *agree sets* for FD and *binary relation* for implications
- Proof of 1. similar to the proof given for Functional Dependencies by Mannila and Raiha 1994 [21], Demetrovics and Thi 1995 [16].
- Proof of 2. a bit more elaborated

RQL: a Practical Language

RQL has 5 clauses (with the "look and feel" of SQL):

FINDRULES
OVER $A_1, \ldots, A_n$
SCOPE $t_1(SQL_1), \ldots, t_n(SQL_n)$
WHERE $condition(t_1, \ldots, t_n)$
CONDITION ON A IS $\delta(A, t_1, \ldots, t_n)$

Examples

```
FINDRULES
OVER Empno,Lastname,Workdept,Job,Sex,Bonus
SCOPE t1,t2 Emp
CONDITION ON A IS t1.A = t2.A;
```

```
FINDRULES
OVER Empno, Lastname, Workdept, Job, Sex, Bonus, Mgrno
SCOPE t1 Emp
CONDITION ON A IS t1.A IS NULL
```

Examples

```
FINDRULES
OVER ...
SCOPE t1,t2,t3 sensors
WHERE t2.time = t1.time+interval 1 minute AND
t3.time = t2.time+interval 1 minute
CONDITION ON A IS t1.A < t2.A AND t2.A > t3.A;
```

RQL query processing.

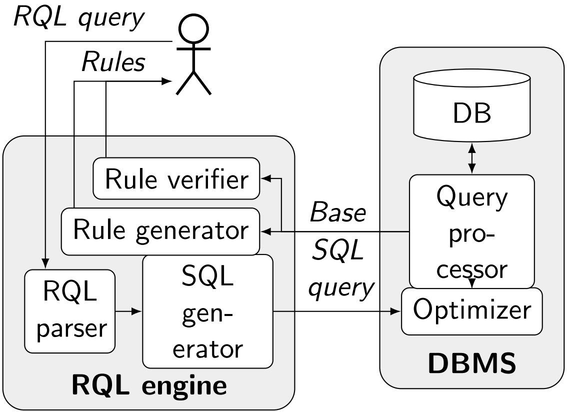


Figure: RQL queries processing overview

RQL Web Interface

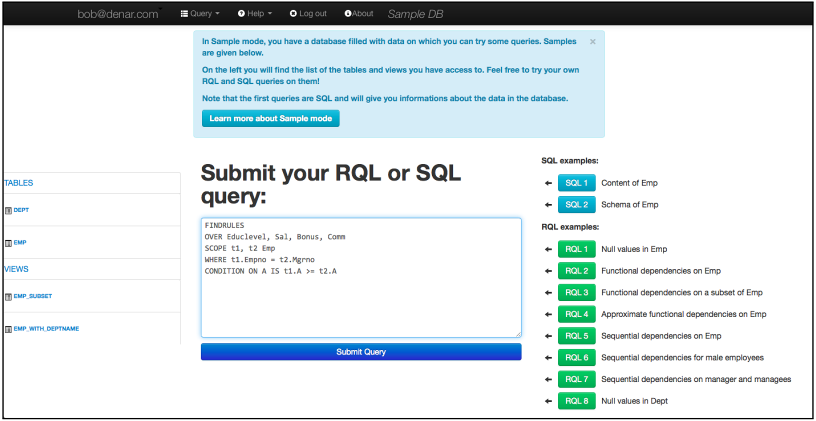


Figure: RQL Interface

RQL Web Interface

Rule verification:

The rule **Sal Educlevel ➜ Bonus** is **false**

Counter-example:

EMPNO	LASTNAME	WORKDEPT	JOB	EDUCLEVEL	SEX	SAL	BONUS	COMM	MGRNO
10	SPEN	C01	FINANCE	18	F	52750	500	4220	20
20	THOMP	null	MANAGER	18	M	41250	800	3300	null

Generated query:

```sql
SELECT t1.*, t2.*
FROM Emp t1, Emp t2
WHERE (t1.Sal >= t2.Sal AND t1.Educlevel >= t2.Educlevel)
AND CASE WHEN (t1.Bonus >= t2.Bonus) THEN 1 ELSE 0 END = 0
AND rownum <= 1
```

Figure: Counter-example with RQL

Summary

- RQL: a practical language to express different semantics for implication
- Discovery of implications seen as a *query processing* problem
- Side effect: data analysts may interact with their data through counter-examples
- Advantages
  - Easy to learn for SQL-aware data analysts (especially CS students !)
  - http://rql.insa-lyon.fr

Contents

Come back to functional dependencies

$$r \models X \to Y \text{ iff for all } t_1, t_2 \in r$$

$$\textbf{If} \text{ for all } A \in X, t_1[A] = t_2[A] \textbf{ then} \text{ for all } B \in Y, t_1[B] = t_2[B]$$

Let us focus on the equality $t_1[A] = t_2[A]$ **without** defining new predicates on $t_1[A]$ and $t_2[A]$ values

From equality to similarity

Two possibilities:

- Replace "$t_1[A] = t_2[A]$" by "$t_1[A]$ is similar to $t_2[A]$"
  $\Rightarrow$ Similarity seen as a *reflexive and symmetric binary relation*
- Replace "$t_1[A] = t_2[A]$" by "$t_1[A]$ and $t_2[A]$ are similar to some similarity value s"
  $\Rightarrow$ Similarity seen as an *idempotent and commutative map*

$\Rightarrow$ Focus on similarity map which appears to be less restrictive than similarity relation

Similarity relation

Let $\mathcal{D}_A$ be the domain of attribute A and $u, v \in \mathcal{D}_A$
Let S be a binary relation on $\mathcal{D}_A$

Similarity

S is a **similarity relation** if S is reflexive ($S(u, u) = 1$) and symmetric ($S(u, v) = S(v, u)$).

S subsumes the equality operator
Two meaningful values: true (1) and false (0)

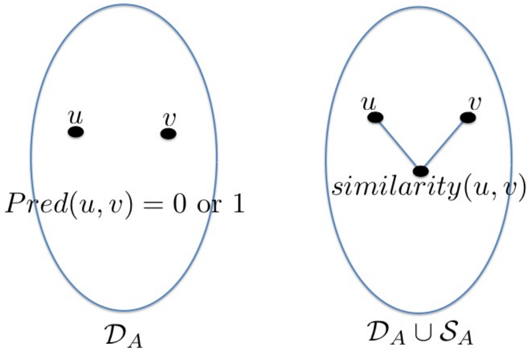
Assumptions on similarity map

Notations:

- A is an attribute, $\mathcal{D}_A$ its domain
- $\mathcal{S}_A$ new values denoting similarities for A (disjoint from $\mathcal{D}_A$)

Assumption:

- For any subset of $\mathcal{D}_A \cup \mathcal{S}_A$, there is a **unique** similarity value.



Similarity map: a semilattice version

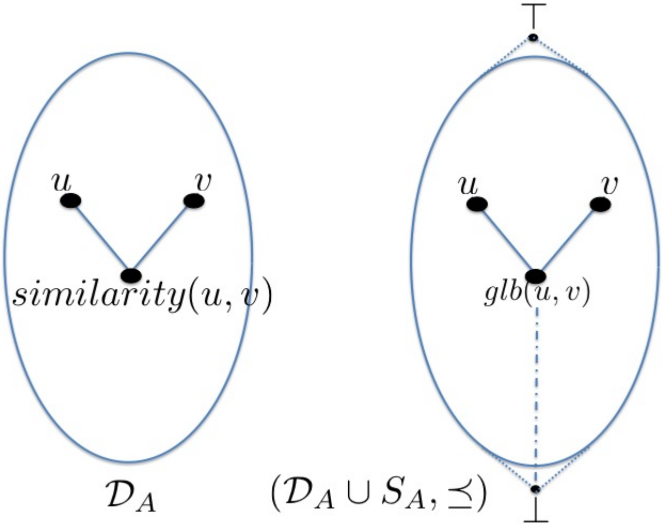
Let A be an attribute, $S = \mathcal{D}_A \cup \mathcal{S}_A$ and $m_A : S \times S \to S$ a **similarity map** that is:

- Idempotent ($m_A(a, a) = a$ for all $a \in S$),
- Commutative ($m_A(a, a') = m_A(a', a)$ for all $a, a' \in S$),
- Associative ($m_A(a, m_A(a', a'')) = m_A(m_A(a, a'), a'')$ for all $a, a', a'' \in S$).

m_A induces a **partial order** $\preceq$ on S:
for every $a, a' \in S$, $a \preceq a'$ whenever $m_A(a, a') = a$.

$(S, \preceq)$ is a **semilattice** where $glb(a, a') = m_A(a, a')$ for all $a, a' \in S$.

Illustration



Example with numerical interval values

Consider an attribute A whose domain is intervals of integer, i.e. $\mathcal{D}_A = \{[i,j] | i,j \in 1..n, i \leq j\}$

- What would be the similarity values $\mathcal{S}_A$?
  $\Rightarrow$ The set of closed sets of $\mathcal{D}_A$ by intersection
- Let $\{I_1, \ldots, I_m\} \subseteq \mathcal{D}_A \cup \mathcal{S}_A$. Similarity value of $\{I_1, \ldots, I_m\}$?
  $\Rightarrow$ its intersection $I = \bigcap\{I_1, \ldots, I_m\}$
  $\Rightarrow$ I is clearly unique

Two examples of similarity map

Equality can be defined as:

$$m_A(x, y) = \begin{cases} x & \text{if } x = y \\ \perp & \text{otherwise} \end{cases}$$

$\perp$ means "not similar" or 0 (false)

Similarity over intervals can be defined as:

$$m_A(I_1, I_2) = \begin{cases} I_1 \cap I_2 & \text{if } I_1 \cap I_2 \neq \emptyset \\ \perp & \text{otherwise} \end{cases}$$

Underlying assumption

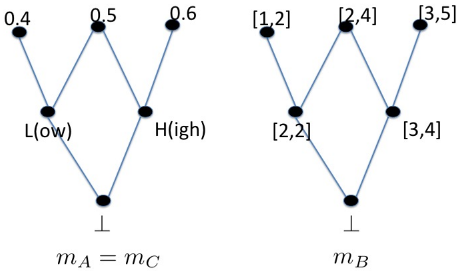
A dataset r has to be equipped with a semilattice structure for every attribute domain

$\Rightarrow$ Allow to be as close as possible of data values to quantify their similarities and differences
$\Rightarrow$ Require an important data pre-processing task, that could be partially automated using data mining techniques

A different approach to address data diversity

Running example

r	A	B	C
t_1	0.4	[1,2]	0.6
t_2	0.5	[2,4]	0.5
t_3	0.6	[3,5]	0.6
t_4	0.4	[2,2]	0.4
t_5	0.5	[3,5]	0.4



$\Rightarrow$ Semantics for m_A and m_C

- The values L and H qualify the different values
- $\perp$ otherwise, i.e. **not similar**.

Application to functional dependencies

$$\boxed{\begin{array}{l} r \models X \rightarrow Y \text{ iff for all } t_1, t_2 \in r \\ \text{ for all } A \in X, t_1[A] = t_2[A] \Rightarrow \text{ for all } B \in Y, t_1[B] = t_2[B] \end{array}}$$

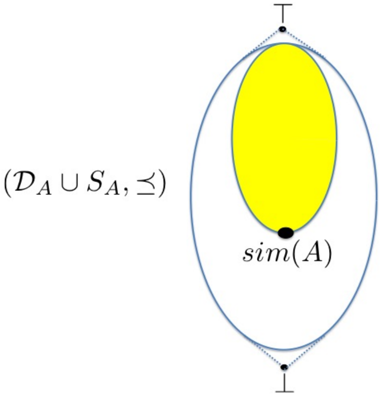
can be reformulated as follows:

$$\boxed{\begin{array}{l} \text{for all } A \in X, glb(t_1[A], t_2[A]) \neq \perp \Rightarrow \\ \qquad \text{for all } B \in Y, glb(t_1[B], t_2[B]) \neq \perp \end{array}}$$

$glb(t_1[A], t_2[A]) \neq \perp$ means **there exists a similarity between the values of A on t_1, t_2**

Minimal degree of similarities

Assume now an expert provides for each attribute A a minimal degree of similarity she expects.
Let $sim : sch(r) \to (\mathcal{D}_A \cup \mathcal{S}_A) \setminus \{\perp\}$ be such a map.



Examples

$$r \models X \to Y \text{ iff for all } t_1, t_2 \in r$$
$$\text{for all } A \in X, t_1[A] = t_2[A] \Rightarrow \text{ for all } B \in Y, t_1[B] = t_2[B]$$

becomes

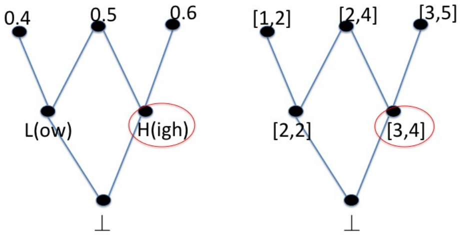
$$r \models_{sim} X \to Y \text{ iff for all } t_1, t_2 \in r$$
$$\text{for all } A \in X, sim(A) \preceq glb(t_1[A], t_2[A]) \Rightarrow$$
$$\text{for all } B \in Y, sim(B) \preceq glb(t_1[B], t_2[B])$$

$sim(A) \preceq glb(t_1[A], t_2[A])$ means **the similarity level between the values of A on t_1, t_2 is above the mimimum**

Example

Assume the expert tags those similarities: $sim(A) = sim(C) =$ H and $sim(B) =$ [3,4]

r	A	B	C
t_1	0.4	[1,2]	0.6
t_2	0.5	[2,4]	0.5
t_3	0.6	[3,5]	0.6
t_4	0.4	[2,2]	0.4
t_5	0.5	[3,5]	0.4



$m_A = m_C$ m_B

$r \models_{sim} A \rightarrow B$ (or $r \models_{sim} A, High \rightarrow B, [3,4]$)

$r \not\models_{sim} C \rightarrow B \Rightarrow$ for ex. see counter-example t_1, t_2

Many results follow ...

Many well-known results on FD can be re-defined in this new setting

Agree sets

Agree sets can be extended naturally: instead of getting a set of attributes (due to 0 and 1 interpretation values based on equality), we obtain a set of similarities

$$ag(r) = \{ag(t_1, t_2) \mid t_1, t_2 \in r\}$$
$$ag(t_1, t_2) = \{ag(t_1[A], t_2[A]) \mid A \in sch(r)\}$$
$$ag(t_1[A], t_2[A]) = glb(t_1[A], t_2[A])$$

Example: $ag(t_1, t_2) =< L, [2, 2], H >$

Example

r	A	B	C
t_1	0.4	[1,2]	0.6
t_2	0.5	[2,4]	0.5
t_3	0.6	[3,5]	0.6
t_4	0.4	[2,2]	0.4
t_5	0.5	[3,5]	0.4

ag(r)	A	B	C
$ag(t_1, t_2)$	L	[2,2]	H
$ag(t_1, t_3)$	$\perp$	$\perp$	0.6
$ag(t_1, t_4)$	0.4	[2,2]	$\perp$
$ag(t_1, t_5)$	L	$\perp$	$\perp$
$ag(t_2, t_3)$	H	[3,4]	H
$ag(t_2, t_4)$	L	[2,2]	L
$ag(t_2, t_5)$	0.5	[3,4]	L
$ag(t_3, t_4)$	$\perp$	$\perp$	$\perp$
$ag(t_3, t_5)$	H	[3,5]	$\perp$
$ag(t_4, t_5)$	L	$\perp$	0.4

From $ag(r)$, two interesting cases:

- replacing all values occurring in r by 1 and all other values by 0 $\Rightarrow$ classical FD with equality
- replacing $\perp$ by 0 (or false) and all other values by 1 (true) $\Rightarrow$ classical FD extended to similarities

Closures and agree sets

From the agree set of r, the family $\mathcal{F}_r$ of closed sets by the glb operation is:
$\mathcal{F}_r = \{glb_{\preceq_{sch(r)}}(T) | T \subseteq ag(r)\}$

Lemma

$(\mathcal{F}_r, \preceq_{sch(r)})$ *is a semilattice*

Let $\mathcal{M}(\mathcal{F}_r)$ be the meet irreducible elements of $\mathcal{F}_r$

Theorem

$\mathcal{M}(\mathcal{F}_r) \subseteq ag(r) \subseteq \mathcal{F}_r$

Similarity, attribute closure and implications

Let $\mathcal{F}$ be a family of closed sets, $X \subseteq sch(r)$ and $sim(X) = \{sim(A) | A \in X\}$

$$X^+_{sim(X)} = glb(\{Y \in \mathcal{F} \mid sim(X) \preceq_X Y\})$$

Theorem

$r \models_{sim} X \to Y$ *iff* $sim(Y) \preceq_Y X^+_{sim(X)}$

Example with $r \models_{sim} A \rightarrow B$ with $sim(A) = H$ and $sim(B) = [3, 4]$

r	A	B	C
t_1	0.4	[1,2]	0.4
t_2	0.5	[2,4]	0.5
t_3	0.6	[3,5]	0.6
t_4	0.4	[2,2]	0.4
t_5	0.5	[3,4]	0.4

ag(r)	A	B	C
$ag(t_1, t_2)$	L	[2,2]	H
$ag(t_1, t_3)$	$\perp$	$\perp$	0.6
$ag(t_1, t_4)$	0.4	[2,2]	$\perp$
$ag(t_1, t_5)$	L	$\perp$	L
$ag(t_2, t_3)$	**H**	[3,4]	H
$ag(t_2, t_4)$	L	[2,2]	L
$ag(t_2, t_5)$	**0.5**	[3,4]	L
$ag(t_3, t_4)$	$\perp$	$\perp$	$\perp$
$ag(t_3, t_5)$	**H**	[3,4]	$\perp$
$ag(t_4, t_5)$	L	$\perp$	0.4

$$A^{+}_{sim(A)} = glb_{\preceq_{ABC}}\{< H, [3,4], H >, < 0.5, [3,4], L >, < H, [3,4], \perp >\} = < H, [3,4], \perp >$$

$$sim(B) \preceq_B < H, [3,4], \perp >$$

$$\Rightarrow r \models_{sim} A, High \rightarrow B, [3,4]$$

Summary

- Using similarity maps on attribute domains allows to reconsider classical data dependencies
- Require to change our mind: most of the effort has to be done at the attribute domain level to define similarity map
- After this, the problem is embedded into a lattice structure allowing to revisit many known results

Contents

Conclusion

Two propositions to extend data dependencies

- First, through RQL, a query language devoted to implications (or FD)
- Second, through assumptions on attribute domains using semilattice structure induced by similarity maps

$\Rightarrow$ Both are elegant formalisms to extend functional dependencies by taking into account data diversity

Perspective

Theoretical question

$\Rightarrow$ Under which conditions the second approach leads to implications (Armstrong axioms) ?

Practical question

$\Rightarrow$ Given a dataset D equipped with semilattice structures, how to discover implications satisfied in D ?

Acknowledgments

Joint work with Brice Chardin, Emmanuel Coquery, Marie Pailloux on RQL (partly funded by ANR, DAG project)



and Lhouari Nourine on structural properties on attribute domains (partly funded by the CNRS Mastodon program on *Data Quality*)

For Further Reading I

S. Abiteboul, R. Hull, and V. Vianu,
Foundations of Databases. Addison-Wesley, 1995.

B. Ganter and R. Wille,
Formal Concept Analysis. Springer, 1999.

Marie Agier-Pailloux, Jean-Marc Petit, Einoshin Suzuki:
Towards Ad-Hoc Rule Semantics for Gene Expression Data.
ISMIS 2005: 494-503

M. Agier-Pailloux, J.-M. Petit, and E. Suzuki,
Unifying framework for rule semantics: Application to gene expression data,
Fundam. Inform., vol. 78, no. 4, pp. 543–559, 2007.

Rakesh Agrawal, Ramakrishnan Srikant:
Fast Algorithms for Mining Association Rules in Large Databases.
VLDB 1994: 487-499

Jaume Baixeries and Victor Codocedo and Mehdi Kaytoue and Amedeo Napoli,
Characterizing approximate-matching dependencies in formal concept analysis with pattern structures,
Discrete Applied Mathematics, 2018

For Further Reading II

F. Baklouti and G. Levy and R. Emilion
A fast algorithm for general Galois lattices building.
Elec. J. Symbolic data analysis, Vol. 2, N 1, 19-31, 2005

C. Beeri, M. Dowd, R. Fagin, and R. Statman.
On the structure of Armstrong relations for functional dependencies.
JACM, 31(1):30–46, 1984.

Bertossi, Leopoldo and Kolahi, Solmaz and Lakshmanan, Laks V. S.,
Data Cleaning and Query Answering with Matching Dependencies and Matching Functions,
ICDT 2011, pp. 268–279

Loredana Caruccio ; Vincenzo Deufemia ; Giuseppe Polese
Relaxed Functional Dependencies: A Survey of Approaches IEEE TKDE, vol 28, issue 1, 2016

N. Caspard and B. Monjardet,
The lattices of closure systems, closure operators, and implicational systems on a finite set: A survey,
Discrete Applied Mathematics, vol. 127, no. 2, pp. 241–269, 2003.

For Further Reading III

Upen S. Chakravarthy, John Grant and Jack Minker
Logic-based approach to semantic query optimization
ACM Transactions on Database Systems (TODS) TODS Homepage archive
Volume 15 Issue 2, June 1990, pages 162-207

Brice Chardin, Emmanuel Coquery, Marie Pailloux, Jean-Marc Petit:
RQL: A SQL-Like Query Language for Discovering Meaningful Rules.
IEEE ICDM demo 2014: 1203-1206

Brice Chardin and Emmanuel Coquery and Marie Pailloux and JM Petit,
RQL: A Query Language for Rule Discovery in Databases,
Theor. Comput. Sci., vol. 658, pp. 357–374, 2017,

Xu Chu, Ihab F. Ilyas, Paolo Papotti:
Discovering Denial Constraints.
PVLDB 6(13): 1498-1509 (2013)

J. Demetrovics and V. D. Thi
Some remarks on generating Armstrong and inferring functional dependencies relation,
Acta Cybernetica, vol. 12, no. 2, pp. 167–180, 1995.

For Further Reading IV

Wenfei Fan and Xibei Jia and Jianzhong Li and Shuai Ma,
Reasoning about Record Matching Rules,
PVLDB, vol 2, num 1, pp. 407–418, 2009

L. Jezková, Pablo Cordero, and Manuel Enciso.
Fuzzy functional dependencies: A comparative survey.
Fuzzy Sets and Systems, 317:88–120, 2017.

Abo Khamis, Mahmoud and Ngo, Hung Q. and Nguyen, XuanLong and Olteanu, Dan and Schleich, Maximilian:
In-Database Learning with Sparse Tensors,
PODS '18, pp. 325–340

S. Lopes, J.-M. Petit, and L. Lakhal,
Functional and approximate dependency mining: database and FCA points of view,
J. Exp. Theor. Artif. Intell., vol. 14, no. 2-3, pp. 93–114, 2002.

Heikki Mannila, Kari-Jouko Rih:
Algorithms for Inferring Functional Dependencies from Relations.
Data Knowl. Eng. 12(1): 83-99 (1994)

Thank you

Merci

σας ευχαριστώ