

# Workshop on Inductive Databases and Constraint Based Mining

## **From KDD scenario description to data mining qualitative benchmarks**

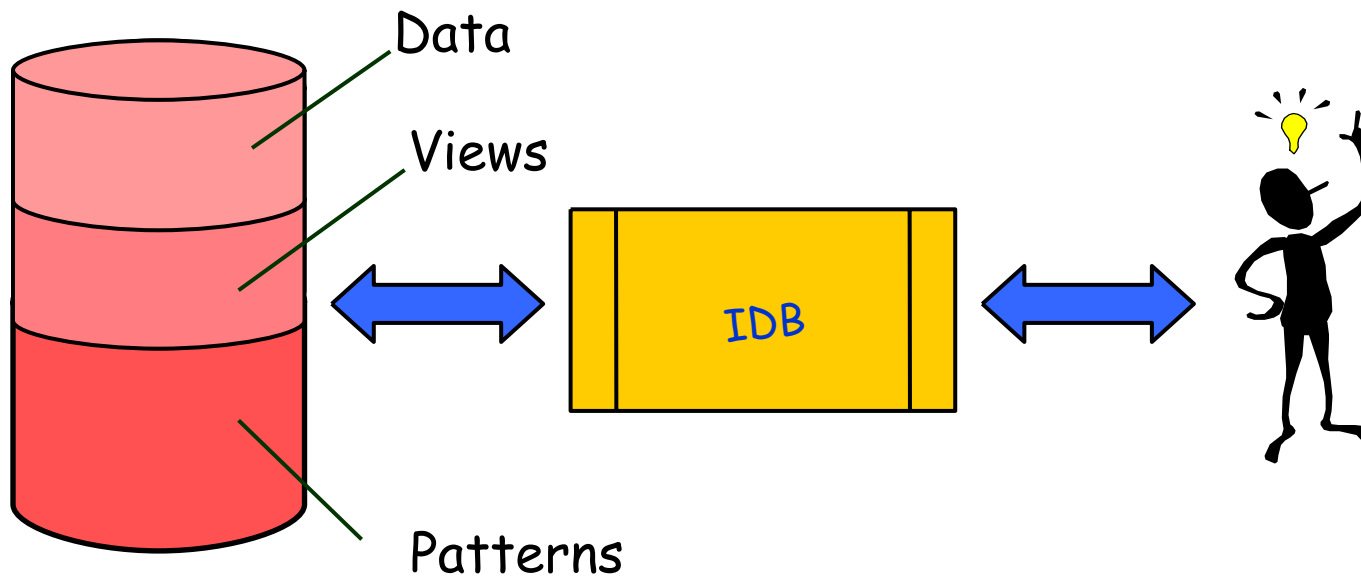
Cyrille Masson and Jean-François Boulicaut– LIRIS/INSA Lyon



Funded by the European cInQ project (IST-FET-2000-26469)

# Introduction

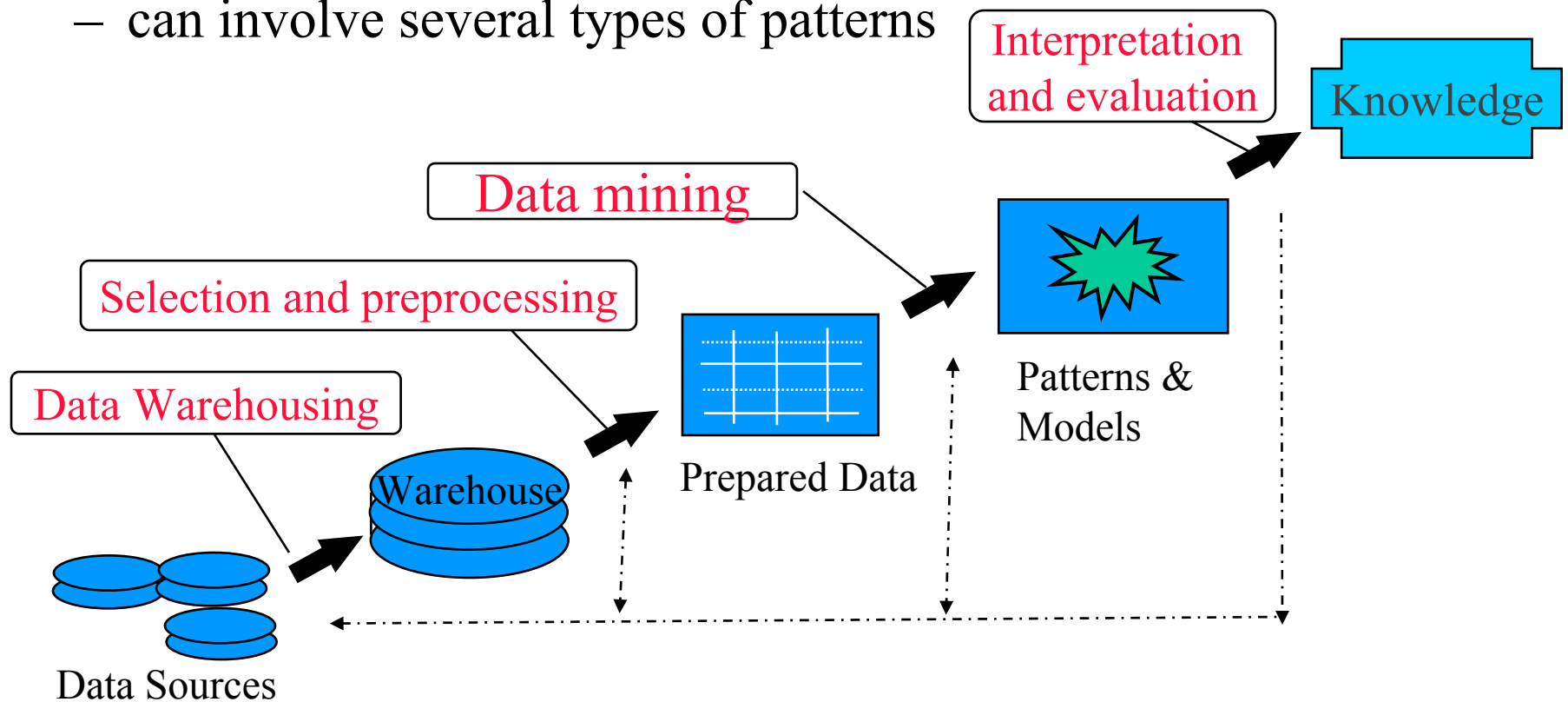
Main task of the last year of the cInQ project:  
Evaluation and assessment of the inductive database framework



Ongoing research on providing guidelines for the evaluation of IDBs and data mining tools

# KDD processes

- KDD processes are complex:
  - iterative and interactive
  - all steps are not easy to formalize
  - can involve several types of patterns



# Towards qualitative benchmarks

- Evaluation of Data Mining (sets of) tools
- Classical benchmarks
  - In the field of machine learning (e.g., UCI datasets)
    - Mostly designed for classification tasks
    - Time, memory, accuracy
    - What about unsupervised techniques (e.g., rule discovery)?
  - In the field of databases (e.g., TPC benchmarks)
    - One benchmark per business-usage type
    - OLTP vs OLAP, reporting, e-commerce, concurrent transactions

# Towards qualitative benchmarks for KDD

A framework for the evaluation of data mining solutions for the whole KDD process would be useful

Formal descriptions of KDD scenarios can be used to support the design of such qualitative benchmarks:

- designing prototypical scenarios from user-trace
- writing benchmarks from scenarios

The IDB framework can be used for such a purpose

# What is a KDD scenario ?

- A KDD scenario describes a sequences of tasks:
  - in an abstract way
  - that are taken from real practice
- It is an abstraction of what the user actually does or might do
- In the framework of Inductive Databases, it can be described as a sequence of queries

# Describing KDD scenario

- Standard queries on  $r$
- Inductive queries on pattern domains

- e.g., on itemsets

create pattern set  $P$  as  $C_{\text{MinFreq}(\gamma,r)}(X) \wedge C_{\text{Free}(r)}(X)$

- on sequential patterns

create pattern set  $P$  as  $C_{\text{MinFreq}(\gamma,r)}(s) \wedge C_{\text{Sim}(\text{ref})}(s)$

- Crossing-over manipulations

create pattern set  $P'$  as  $\alpha(r,P)$

# Example

	A	B	C	D	E	F
S1	0	1	1	0	1	1
S4	1	1	1	0	0	1

$$D_1 \leftarrow \text{Binarize}(D)$$

$$P_1 \leftarrow C_{\delta\text{-Free}(1,D_1)}(S) \wedge C_{\text{MinFreq}(D_1,0.4)}(S)$$

$$P_1 = \{A, C, F\}$$

$$P_2 \leftarrow \delta\text{-Rules}_{(1,D_1)}(P_1)$$

$$P_2 = \{F \rightarrow A, E\}$$

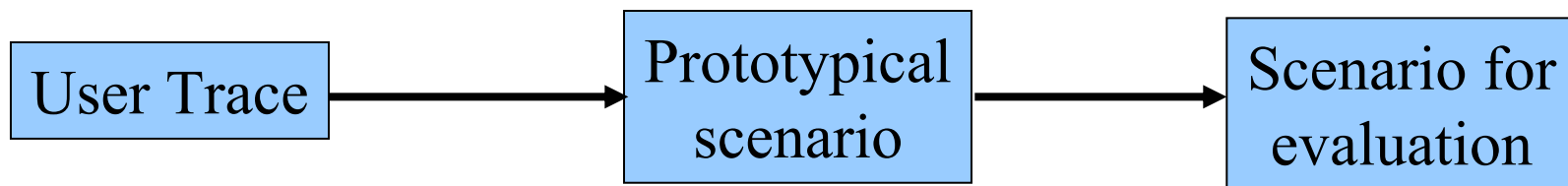
$$D_2 \leftarrow \neg\alpha(D_1, P_2)$$

$$P_3 \leftarrow C_{\text{close}(D_2)}(G) \wedge C_{\text{close}(D_2)}(T) \wedge T = h(G) \wedge G = g(T)$$

$$P_3 = \{(\{S1\}, \{D, C, E, F\}), (\{S4\}, \{A, B, C, F\}), (\{S1, S4\}, \{B, C, F\})\}$$

# Usage of scenarios

- Abstracting user traces to prototypical sequences of queries
- Instantiate prototypical scenarios to support both quantitative and qualitative evaluation



# User Trace

Situation	Cell	Time	Event	Gene <sub>1</sub>	Gene <sub>2</sub>	Gene <sub>3</sub>	...	Gene <sub>a</sub>
S3	C2	2	SB	0	0	1	...	0
S4	C2	5	SD	0	1	0	...	1
S5	C2	8	SB	1	0	0	...	0
S9	C4	7	SD	1	1	1	...	0
S10	C4	10	SC	0	0	1	...	1
...	...	...	...	...	...	...	...	...
Sp-3	C999	8	SF	1	0	1	...	0
Sp-2	C999	9	SB	0	1	1	...	1

Frequent closed itemsets extraction on gene properties

- First with threshold = 0.4
- Next with threshold = 0.3
- Finally with threshold = 0.15

# Prototypical scenario

- An abstraction of what the user does or might do
- Parameters like  $\gamma$  or  $\delta$  can be fixed or estimated  
Using data characterization (e.g., size, density)
- Using for a transfer of expertise (mining method)
  - binarization for encoding gene expression properties
  - computation of frequent closed sets  $X$
  - looking at the bi-sets  $\langle X, g(X,r) \rangle$  as a putative transcription module (See the talk by J. Besson)

# Prototypical scenario

- Looking for query optimization strategies
  - Scenario as formal objects:  
Reasoning on them is possible (relationship between queries, execution plans,...)
  - Study of primitive constraints:  
Formal properties, relaxations of the constraints  
Design of solvers for some interesting conjunction of constraints
  - Query compilation  
Generic processing of some queries

# Prototypical scenario

## Binarization

create data set  $C_3$  as  $\text{Binarize}(C_1)$

## Sequential pattern extraction

create pattern set  $P_1$  as  $C_{\text{MinFreq}(C_3, t_0)}(s) \wedge C_{\text{sim(ref)}}(s) \wedge C_{\text{MaxFreq}(C_1, t_1)}(s)$

## Crossing-over

create data set  $C_4$  as  $\alpha(C_3, P_1)$

## Frequent closed set extraction

create pattern set  $P_2$  as  $C_{\text{MinFreq}(C_4, t_2)}(g) \wedge C_{\text{Close}(C_4)}(g)$

# Scenario for the evaluation

- Sequence of queries instantiated from a prototypical scenario
- Study of optimization strategies
- Put in evidence algorithmic difficulties we want to test with different DM solutions

$$C_{\text{MinFreq}(C5,0.3)}(\mathbf{g}) \wedge C_{\text{Close}(C5)}(\mathbf{g})$$

$$C_{\text{MinFreq}(C5,0.1)}(\mathbf{g}) \wedge C_{\text{Close}(C5)}(\mathbf{g})$$

$$C_{\text{MinFreq}(C5,0.2)}(\mathbf{g})$$

# Study of optimization strategies

- Non anti-monotonic constraints
  - The case of regular expressions (SPIRIT, Garofalakis99) in sequential pattern mining
  - Different relaxations of the regular expression constraint
  - The selectivity of the constraint has an influence on the strategy for pushing constraint
  - Tradeoff between pruning based on frequency constraint and pruning based on regular expression constraint
  - A priori choice of the pruning technique
  - Work on adaptive strategies (Albert-Lorincz 03, Bonchi 03)

# Evaluation of optimization strategies

- Extraction of frequent sets that satisfy some syntactic constraint
  - Direct extraction with Apriori algorithm
  - Use of condensed representation and fast post-processing to regenerate all constrained frequent sets
$$C_{\text{close}(D)}(S) \wedge C_{\text{minfreq}(D,t)}(S)$$
$$C_{\text{free}(D)}(X) \wedge C_{\text{minfreq}(D,t)}(X) \wedge S=h(X,D)$$
  - If the syntactic constraint is monotonic, use of particular algorithms ( $C_m \wedge C_{am}$ )

# Sequence of queries

- How to reuse previous queries ?
- Caching techniques
  - Keeping previous results in a cache (e.g., Jeudy 02)
  - Build caches of frequent itemsets automatically to speed up some awaited queries on itemsets
- Equivalence of queries (e.g., Meo, SAC '03)
  - According to the attributes involved in a Mine RULE query, we can deduce relationships between result sets.

# Towards qualitative benchmarks



- Provide one instance of data and describe processes
- Using scenarios to evaluate DM tools
  - Use of data characterization
  - Choice of constraints
  - Comparison of the involved techniques (dedicated algorithms, scripts, ...)
  - Comparison of used resources (time, memory, ...)
  - Put in evidence required expertise of the user

**THE END**