



Rough Set Approaches to Scale Data Processing and Mining Operations

Dominik Šlězák
WIUI 2013
05.06 Olomouc



Rough Sets

- The theory of rough sets founded in early 80-ties by Prof. Pawlak provides the means for handling incompleteness and uncertainty in large data sets
- In the process of knowledge discovery, one can search for *decision reducts*, which are irreducible subsets of attributes determining decision values
- Dependencies in data can be expressed in terms of, e.g., *discernibility* or *rough set approximations*
- There are also rough-set-inspired computational models, such as *rough clustering*, *rough SQL* etc.



Different Approaches to Attribute Reduction

■ Reduction Constraints:

- Keep (almost) the same approximations of decision classes
- Discern between (almost) all pairs of objects with different decision values
- Keep at (almost) the same level a value of some quality function

■ Optimization Goals:

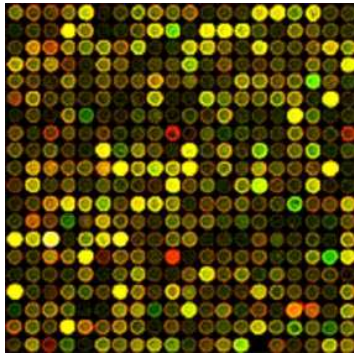
- Find minimal reduct(s)
- Find reducts, which induce minimum amount of rules
- Find ensembles of reducts, which work well together

■ Algorithms & Structures:

- Greedy methods, randomized methods, MapReduce methods, attribute clusters
- Discernibility matrices, data sorting, hashing, distributing, SQL-based scripts



The Case Study of Gene Expression Data



	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
Gene 1	-1.2	-2.1	-3	-1.5	1.8	2.9
Gene 2	2.7	0.2	-1.1	1.6	-2.2	-1.7
Gene 3	-2.5	1.5	-0.1	-1.1	-1	0.1
Gene 4	2.9	2.6	2.5	-2.3	-0.1	-2.3
Gene 5	0.1		2.6	2.2	2.7	-2.1
Gene 6	-2.9	-1.9	-2.4	-0.1	-1.9	2.9

- Thousands of genes-attributes to analyze
- Number of experiments-objects quite low
- Simple knowledge representation needed



Discernibility & Discretization

- Consider an arbitrary vector of cuts over the domains of attributes $b \in B$:

$$\text{cut}_B = \{(a, \text{cut}_a) : a \in B, \text{cut}_a \in (\underline{a}, \bar{a})\}$$
$$\underline{a} = \min_{u \in U} a(u) \quad \bar{a} = \max_{u \in U} a(u)$$

- We say that cut_B discerns objects $x, y \in U$, if there is at least one $b \in B$ such that:

$$\min(a(x), a(y)) < \text{cut}_a < \max(a(x), a(y))$$

- Define $\text{Ind}(d/\text{cut}_B) =$
 $|\{(x, y) : d(x) \neq d(y) \text{ \& \text{cut}_B \text{ doesn't discern } x, y}\}|$



Fuzzy Discernibility & Discretization

- Consider the following modification of $Ind(d/B)$ for numeric attributes B :

$$\sum_{x,y:d(x) \neq d(y)} \prod_{a \in B} \left(1 - \frac{|a(x) - a(y)|}{\bar{a} - \underline{a}} \right)$$

- The above measure equals to:

$$\prod_{a \in B} \frac{1}{\bar{a} - \underline{a}} \int_{I_B} Ind(d/cut_B) dcut_B$$

where $I_B = \times_{a \in B} [\underline{a}, \bar{a}]$

- An analogous relationship can be obtained also for a numeric decision attribute d



Lessons to be Learnt

- Fuzzy-rough attribute reduction criteria can be utilized to efficiently search for subsets of attributes, which would keep information about decision after their discretization.
- Crisp discernibility functions for discretized attributes can be utilized to speed up the process of fuzzy-rough attribute reduction (e.g. via Monte Carlo generation of cuts...)



	a	b	c	d
u1	3	7	3	0
u2	2	1	0	1
u3	4	0	6	1
u4	0	5	1	2

$$\text{POS}(a^*, b^*) = \text{POS}(a^*, b^*, c^*)$$

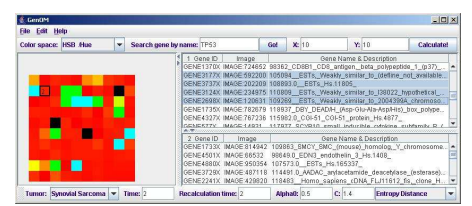
$$\text{POS}(a^*) \subset \text{POS}(a^*, b^*, c^*)$$

$$\text{POS}(b^*) \subset \text{POS}(a^*, b^*, c^*)$$

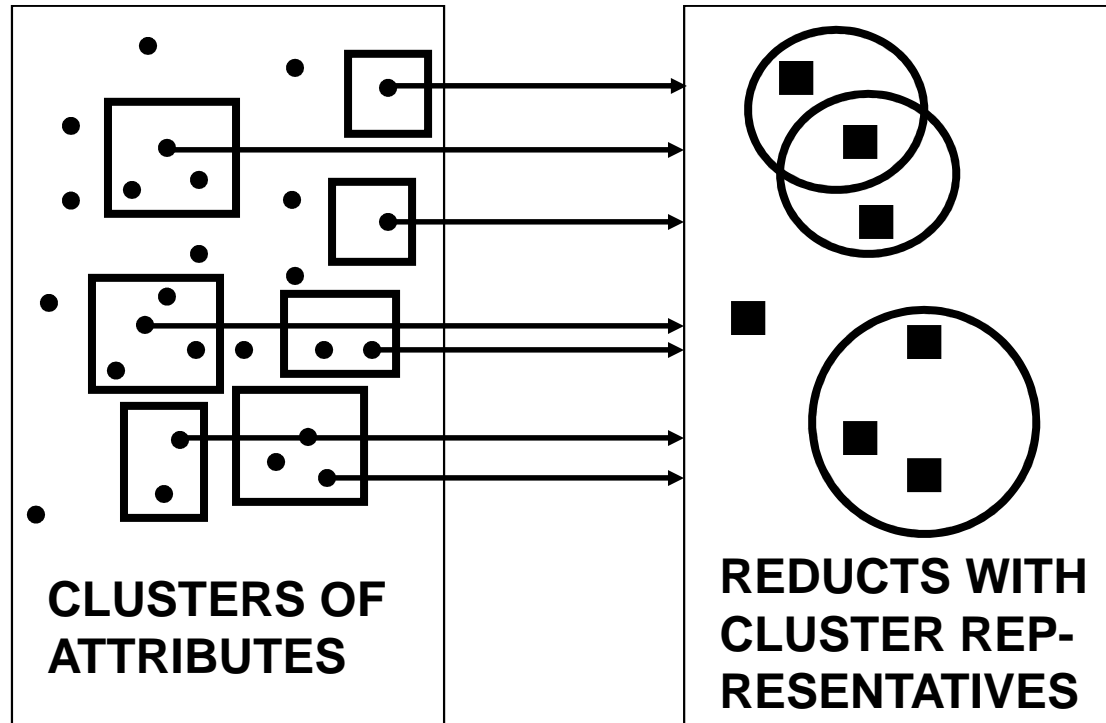
IF $a \geq 3$ AND $b \geq 7$ THEN $d = 0$
 IF $a \geq 3$ AND $b < 7$ THEN $d = 1$
 IF $a \geq 2$ AND $b < 1$ THEN $d = 1$
 IF $a < 2$ AND $b \geq 1$ THEN $d = 2$
 IF $a \geq 4$ AND $b \geq 0$ THEN $d = 1$
 IF $a \geq 0$ AND $b < 5$ THEN $d = 1$

	a^*	b^*	c^*	d^*
(u1,u1)	1+	1+	1+	0
(u1,u2)	1-	1-	1-	1
(u1,u3)	1+	1-	1+	1
(u1,u4)	1-	1-	1-	2
(u2,u1)	2+	2+	2+	0
(u2,u2)	2+	2+	2+	1
(u2,u3)	2+	2-	2+	1
(u2,u4)	2-	2+	2+	2
(u3,u1)	3-	3+	3-	0
(u3,u2)	3-	3+	3-	1
(u3,u3)	3+	3+	3+	1
(u3,u4)	3-	3+	3-	2
(u4,u1)	4+	4+	4+	0
(u4,u2)	4+	4-	4-	1
(u4,u3)	4+	4-	4+	1
(u4,u4)	4+	4+	4+	2

How about Attribute Granules?



Gruźdź, Ihnatowicz, Ślęzak: Interactive gene clustering – a case study of breast cancer microarray data. *Inf. Systems Frontiers* 8 (2006).



Abeel et al: Robust Biomarker Identification for Cancer Diagnosis with Ensemble Feature Selection Methods. *Bioinformatics* 26(3) (2010).

Andrzej Janusz, Dominik Ślęzak: Rough Set Methods for Attribute Clustering and Selection. *Applied Artificial Intelligence*. [Accepted.]

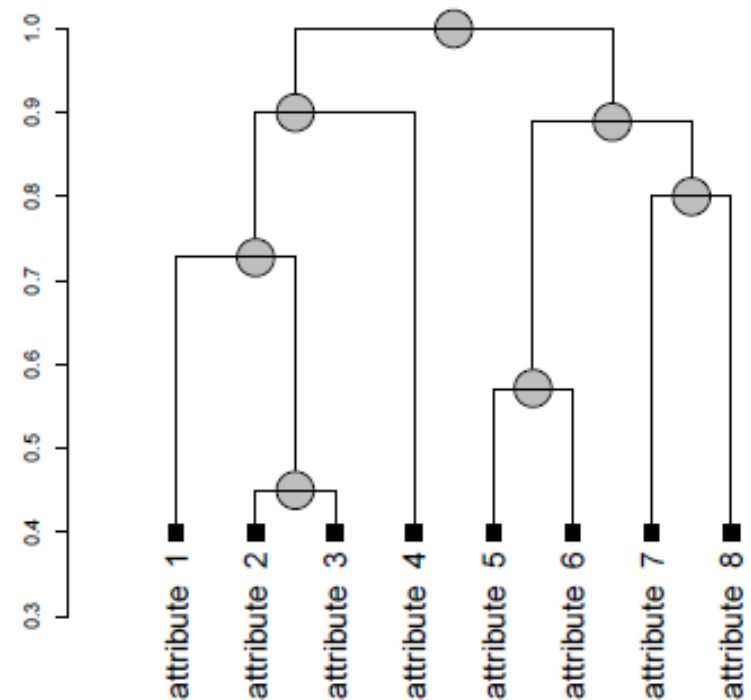
Clusters of Replaceable Attributes (1)

Exemplary decision system

$\mathbb{A} = (U, A \cup \{d\})$:

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	d
u_1	1	2	2	0	0	1	0	1	1
u_2	0	1	1	1	1	0	1	0	1
u_3	1	2	0	1	0	2	1	0	1
u_4	0	1	0	0	1	0	0	1	0
u_5	2	0	1	0	2	1	0	0	1
u_6	1	0	2	0	2	0	0	2	0
u_7	0	1	1	2	0	2	1	0	1
u_8	0	0	0	2	1	1	1	1	0
u_9	2	1	0	0	1	1	0	0	0

Hierarchical attribute clustering of \mathbb{A} :



Clusters of Replaceable Attributes (2)

Goal:

We would like to model interchangeability of attributes in reducts.

Why?

- to facilitate computation of reducts,
- to ensure diversity of obtained reducts,
- to explore complex dependencies between attributes in data.

How?

- a discernibility-based attribute dissimilarity measure:

$direct(a, b) =$

$$1 - \frac{|\{(u, u') : d(u) \neq d(u') \wedge a(u) \neq a(u') \wedge b(u) \neq b(u')\}|}{|\{(u, u') : d(u) \neq d(u') \wedge (a(u) \neq a(u') \vee b(u) \neq b(u'))\}|}$$

- different objects may have different weights (relative discernibility),
- regular clustering algorithms.

Computation of Reducts using Clusters

A cluster-based permutation generator:

Input: a clustering of attributes $CL_A = (C_1, \dots, C_k)$

Output: an ordered list $perm_A = [a_1, \dots, a_n]$

```
permA = []           (an empty list);
permk = [p1, ..., pk] (a permutation of numbers 1 to k);
i = 1;
while length(permA) ≠ n do
  if |Cpi| > 0 then
    pi = permk[i];
    randomly select an attribute a from Cpi;
    permA = [permA, a];
    Cpi = Cpi \ {a};
  end
  i = i + 1;
  if i > k then
    i = 1;
  end
end
return permA;
```

- It is designed to work with the permutation-based algorithm for computation of reducts.
- Similar procedures can be used to combine attribute clustering with other heuristics.

Rough Sets

- The theory of rough sets founded in early 80-ties by Prof. Pawlak provides the means for handling incompleteness and uncertainty in large data sets
- In the process of knowledge discovery, one can search for *decision reducts*, which are irreducible subsets of attributes determining decision values
- Dependencies in data can be expressed in terms of, e.g., *discernibility* or *rough set approximations*
- There are also rough-set-inspired computational models, such as *rough clustering*, *rough SQL* etc.



Rough Computing over Granulated Data

- Decompose the available data onto granules
 - Fast and dynamic decomposition methods are required
- Create statistical snapshots for each granule
 - Snapshots should be small but also informative enough
- Do approximate computations on snapshots
 - It requires redesigning standard computational methods
- When necessary, go down to data granules
 - Accessing granules should be minimized and optimized



Infobright.com & Infobright.org

Financial Services

Bluefin Group

Primatica Financial

Other Industries

Information Builders

Bell Helicopter

USDA

J. Craig Venter Institute

Dorel Juvenile Group

GeoPost UK

JCDecaux UK

Austin Energy

Fuseforward

Canadian Space Agency

Xerox

Telecommunications / Security

SonicWALL

Sonus Networks

JDSU

8x8, Inc.

IMImobile

Communications Service

Provider

Mavenir

Infobright technology enables companies to quickly provide access to critical information to their business users and customers, without the complexity and cost of traditional analytic solutions or data warehousing. Click on the links on the left to read about how our customers are using Infobright.

Online Analytics

YAHOO!

TradeDoublar

1024Degrés

effect^{ive}

sulake

Xtwenga

Smiley Media

bwin

eMAIL

bango

ONLINE
RETAILER

BUNCHBALL

LiveRail

adsafe

InMobi

Telecom / Security

JDSU

Polystar

MAVENIR
SYSTEMS

COMMUNICATIONS
SERVICE PROVIDER

IMImobile

8x8, Inc.

Download
Community Edition

Download
IEE Evaluation

Live Chat

Partner Portal

Customer Support Portal

Customer Stories



Canadian Space Agency

"This [Infobright] solution permits real time compression, compact storage and quick retrieval of relevant data segments using SQL query processing of measured data. Performance of this solution along with its...

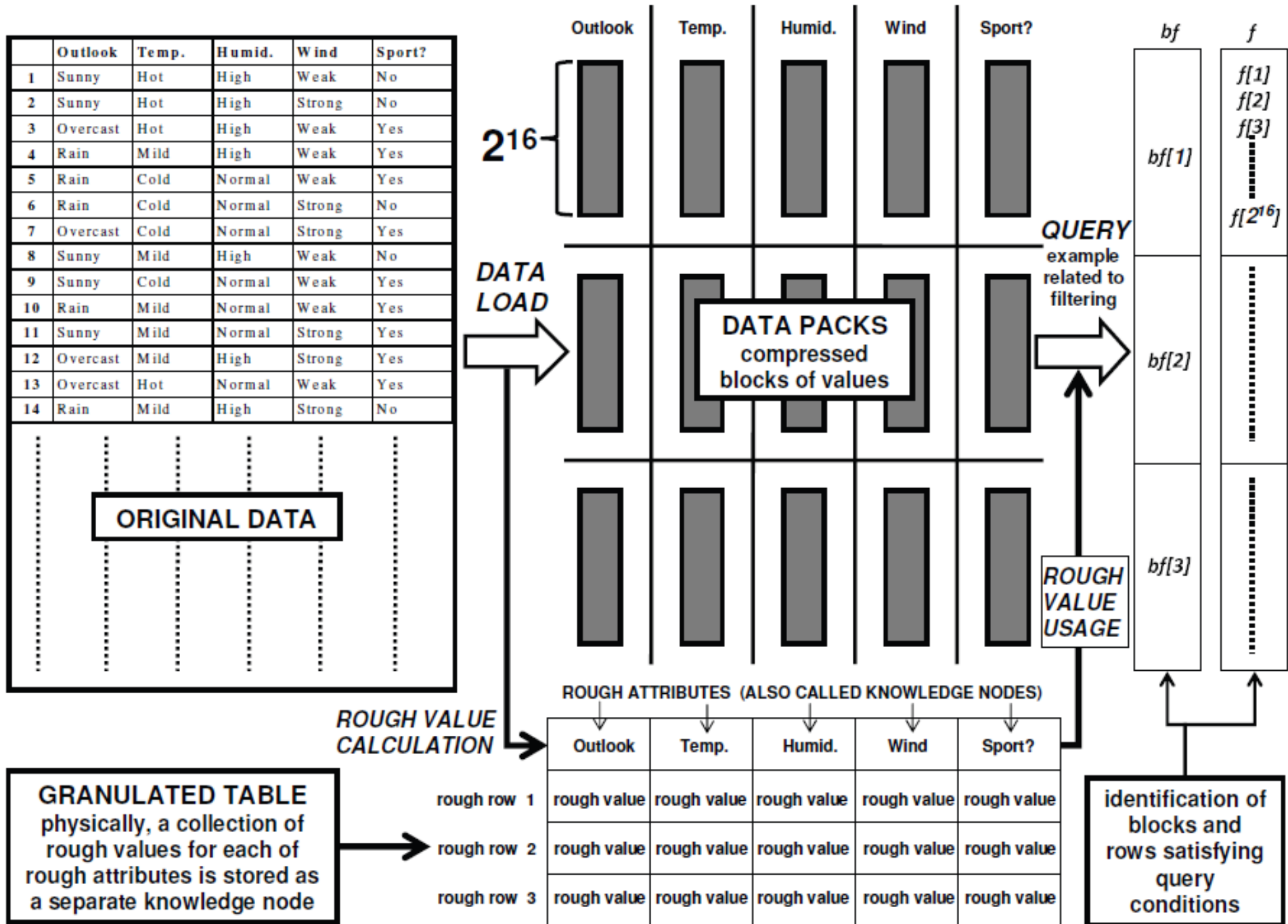
[read more >>](#)

A New Approach



The Analytic Data Warehouse

Traditional data warehouse products put a tremendous burden on IT in order to create and maintain an environment that will allow users to query against large volumes of data.



SELECT MAX(A) FROM T WHERE B > 15;

T (~350K rows)

B > 15

<u>Pack A1</u> Min = 3 Max = 25	<u>Pack B1</u> Min = 10 Max = 30		S
<u>Pack A2</u> Min = 1 Max = 15	<u>Pack B2</u> Min = 10 Max = 20		S
<u>Pack A3</u> Min = 18 Max = 22	<u>Pack B3</u> Min = 5 Max = 50		S
<u>Pack A4</u> Min = 2 Max = 10	<u>Pack B4</u> Min = 20 Max = 40		R
<u>Pack A5</u> Min = 7 Max = 26	<u>Pack B5</u> Min = 5 Max = 10		I
<u>Pack A6</u> Min = 1 Max = 8	<u>Pack B6</u> Min = 10 Max = 20		S

- **I**: Irrelevant Granules (Negative Region)
- **S**: Suspect Granules (Boundary Region)
- **R**: Relevant Granules (Positive Region)
- **E**: Exact Computation (necessary, if the final query result cannot be obtained only from the statistical snapshots)

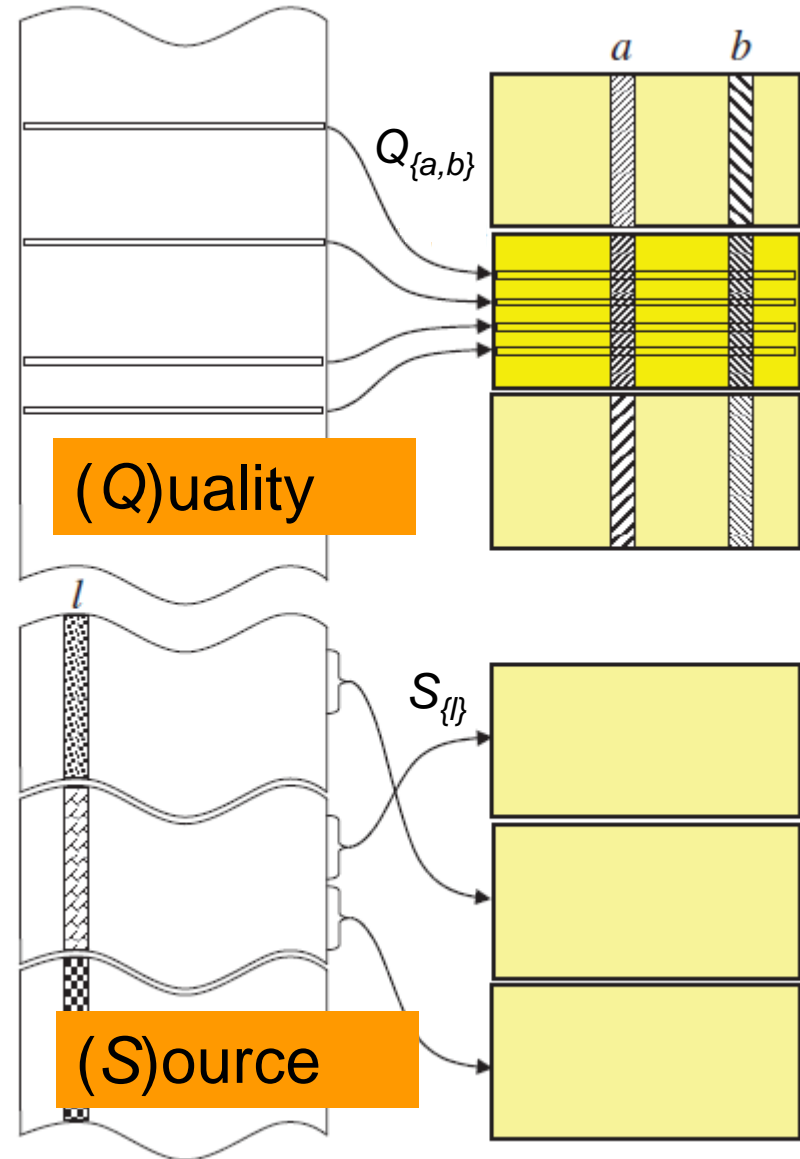
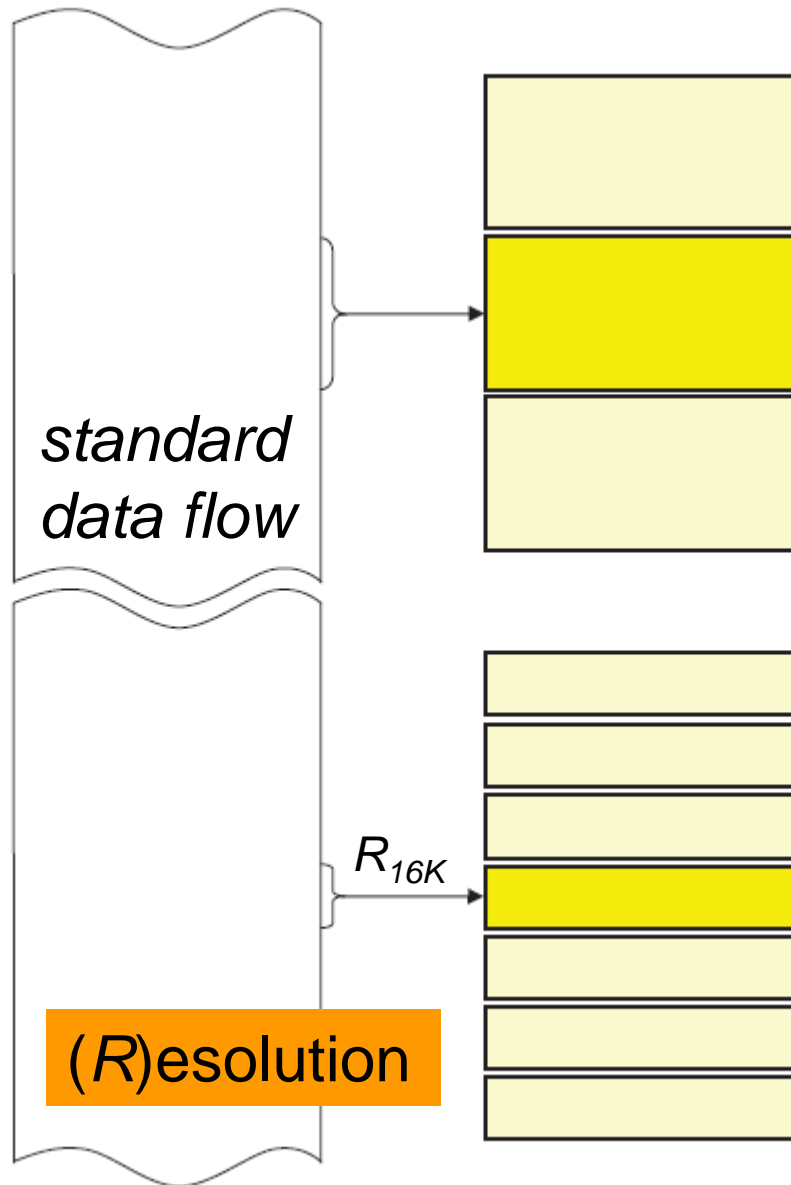
SELECT MAX(A) FROM T WHERE B > 15;

T (~350K rows)

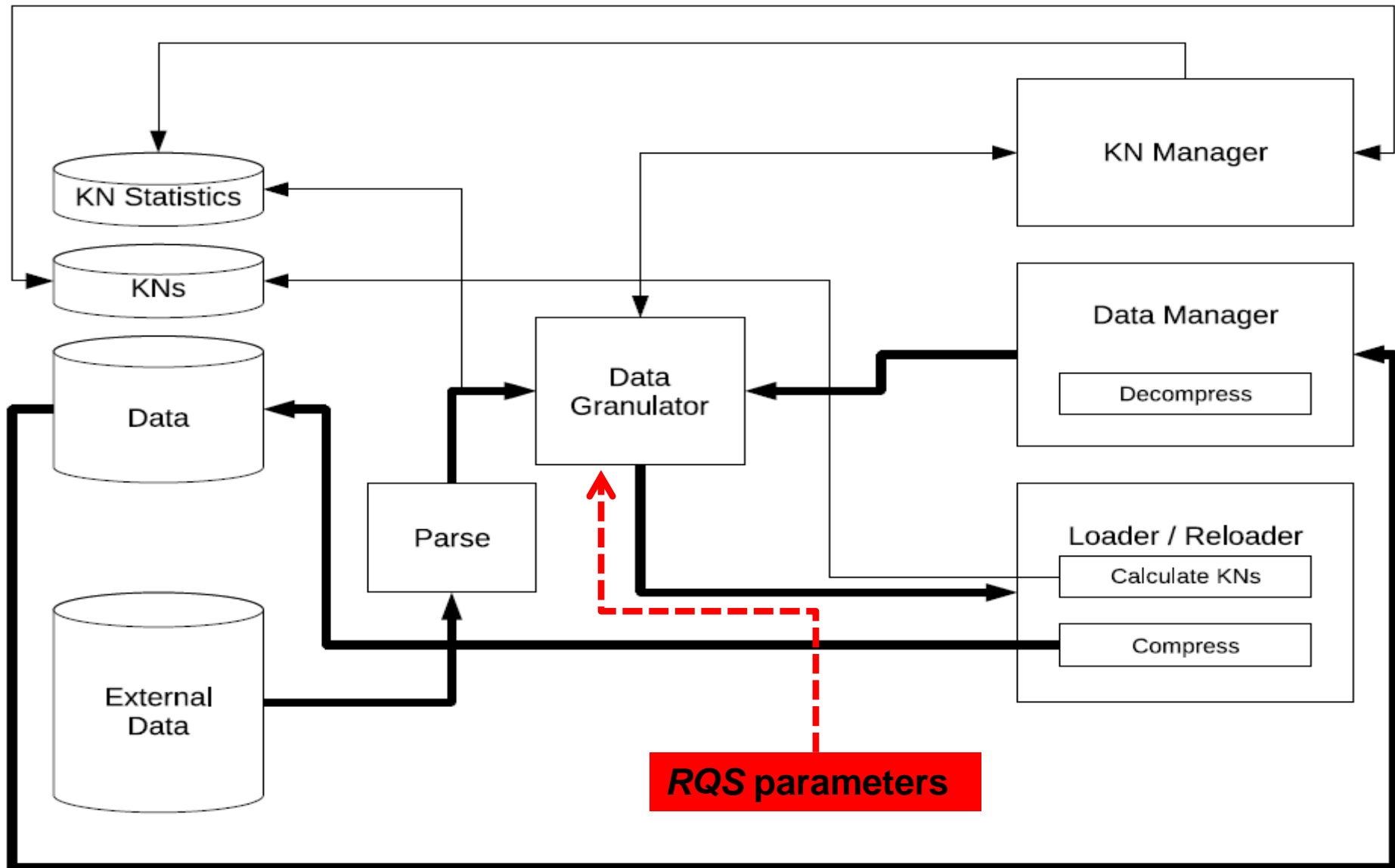
<u>Pack A1</u> Min = 3 Max = 25	<u>Pack B1</u> Min = 10 Max = 30
<u>Pack A2</u> Min = 1 Max = 15	<u>Pack B2</u> Min = 10 Max = 20
<u>Pack A3</u> Min = 18 Max = 22	<u>Pack B3</u> Min = 5 Max = 50
<u>Pack A4</u> Min = 2 Max = 10	<u>Pack B4</u> Min = 20 Max = 40
<u>Pack A5</u> Min = 7 Max = 26	<u>Pack B5</u> Min = 5 Max = 10
<u>Pack A6</u> Min = 1 Max = 8	<u>Pack B6</u> Min = 10 Max = 20

	B > 15	B > 15, A ≥ 18		B > 15, A ≥ X	
	S	S	S	E	E
	S	I	I	I	I
	S	S	S	I ↔ X ≥ 22	I ↔ X ≥ 22
	R	I	I	I	I
	I	I	I	I	I
	S	I	I	I	I

Data Granulation Parameters (Examples)



Architecture with Data Granulation



Rough Set Interpretation

- Packs of rows can be treated as *indiscernibility classes*
- Operators such as *RQS* can be treated as *conditional attributes*
- Snapshots can be compared to *generalized decision functions*
- The task is to adjust conditional attributes, so the resulting generalized decisions are good enough to approximate the concepts we are interested in

Classical Theory of Rough Sets

Let $A=(U, A \cup \{d\})$ be a decision table.

Indiscernibility class $[u]_B$ is defined as:

$$\{x \in U : \forall_{a \in B} a(u) = a(x)\}$$

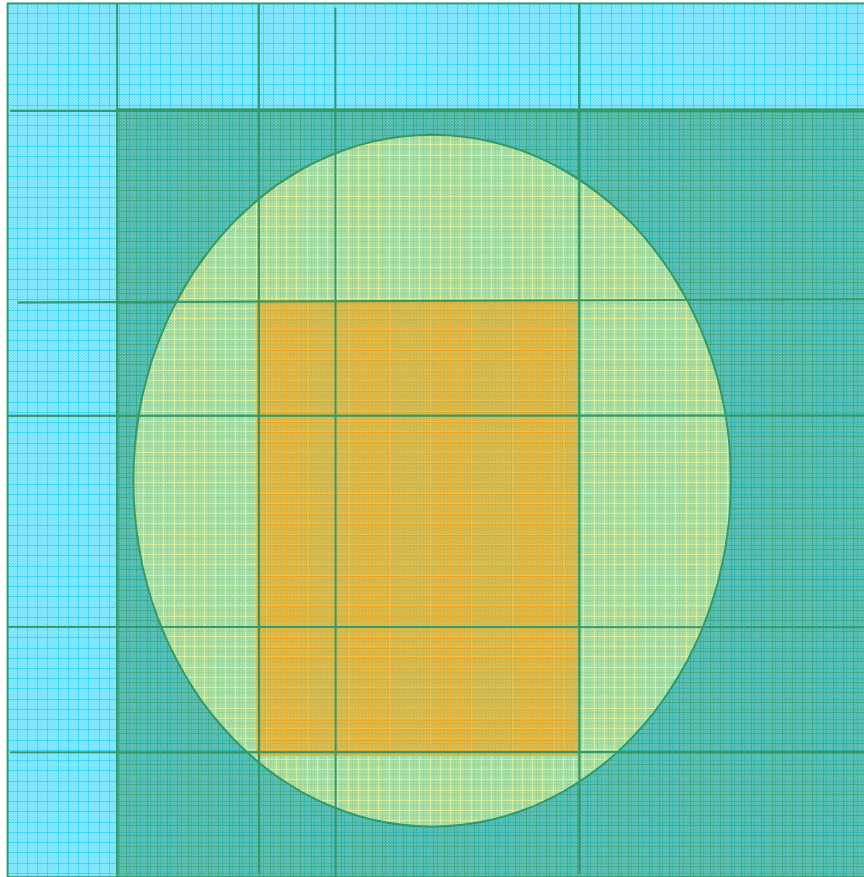
Generalized decision function is defined as $\partial([u]_B) = \{d(x) : x \in [u]_B\}$



$$POS_B(X) = \{ u \in U : \partial([u]_B) \subseteq X \}$$

$$NEG_B(X) = \{ u \in U : \partial([u]_B) \cap X = \emptyset \}$$

Rough Set Interpretation (continued)

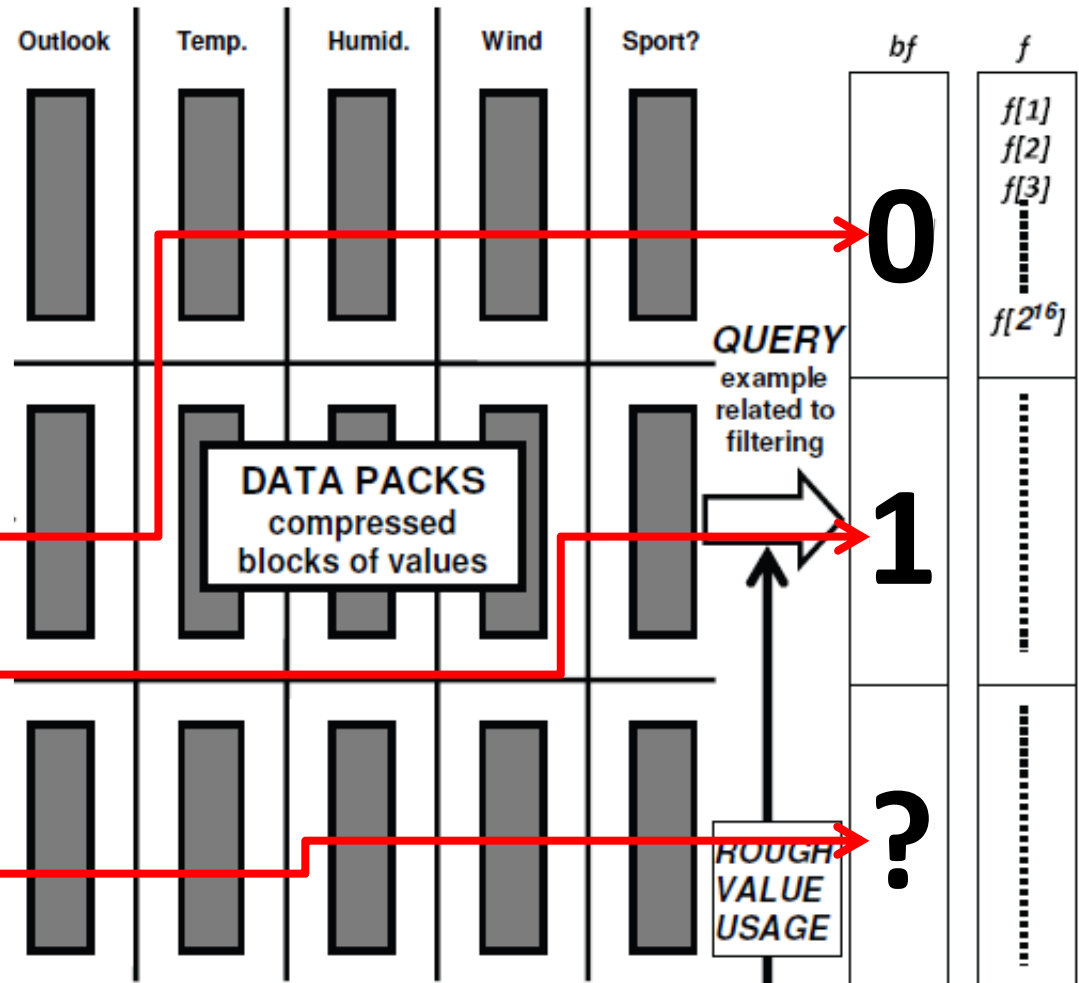


- Statistical snapshots of indiscernibility classes are used to approximate concepts related to the SQL statement execution
- Approximated concepts can be totally different in case of each statement
- Approximated concepts can dynamically evolve during the SQL execution



**SELECT COUNT(*)
FROM TABLE WHERE
Outlook = x;**

1. Take the rough attribute of Outlook
2. Filter out fully irrelevant data packs
3. Utilize rough values of fully relevant packs to compute the partial result
4. Decompress packs, which were not fully (ir)relevant
5. Compute the final result



ROUGH ATTRIBUTES (ALSO CALLED KNOWLEDGE NODES)

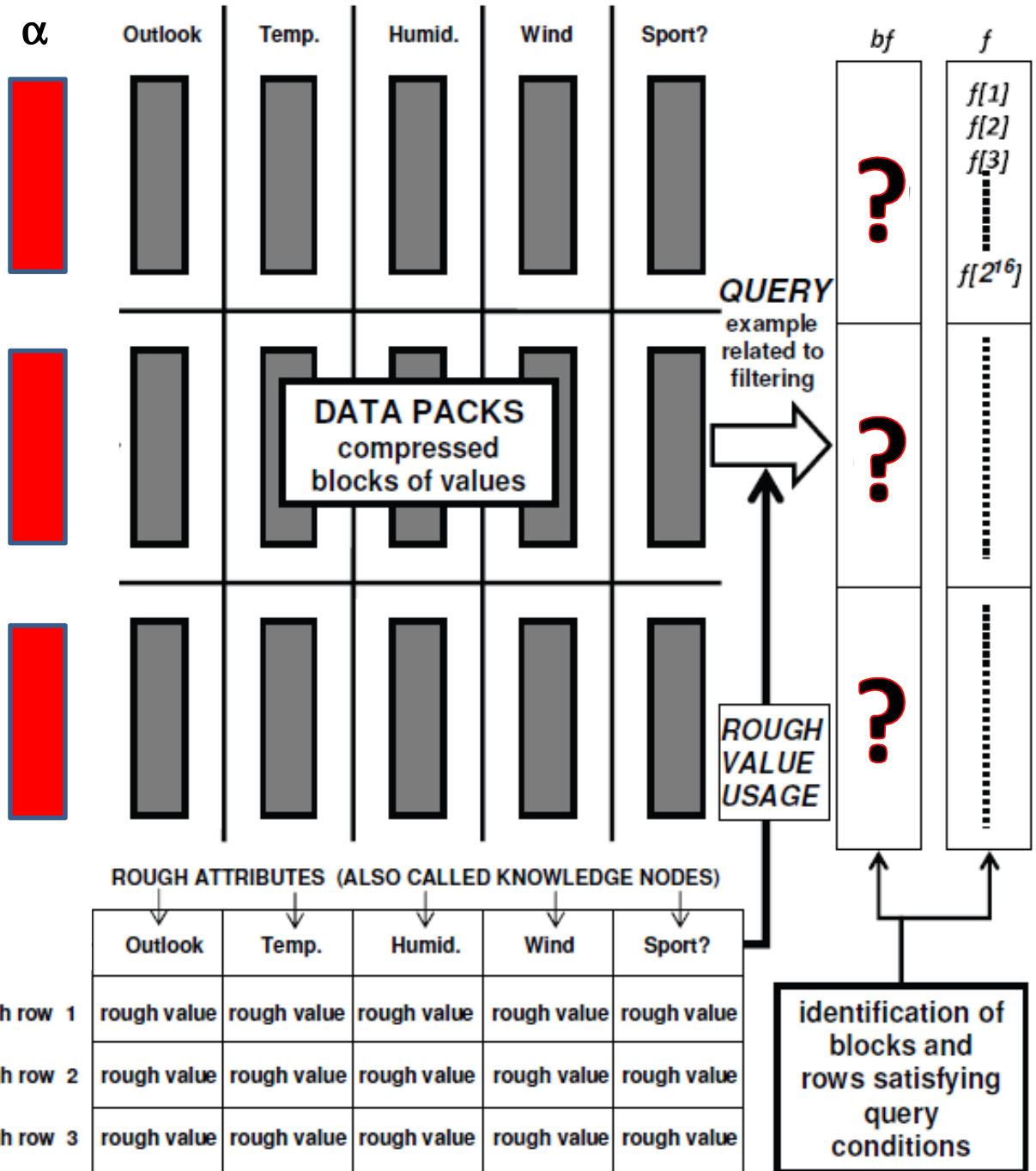
	Outlook	Temp.	Humid.	Wind	Sport?
rough row 1	rough value	rough value	rough value	rough value	rough value
rough row 2	rough value	rough value	rough value	rough value	rough value
rough row 3	rough value	rough value	rough value	rough value	rough value

GRANULATED TABLE
physically, a collection of rough values for each of rough attributes is stored as a separate knowledge node

identification of blocks and rows satisfying query conditions

**SELECT COUNT(*)
FROM TABLE WHERE
 $\alpha(\text{Outlook}, \text{Temp.}) = x;$**

1. Take the rough attributes of $\alpha(\text{Outlook}, \text{Temp.})$
2. Filter out fully irrelevant data packs
3. Utilize rough values of fully relevant packs to compute the partial result
4. Decompress packs, ~~which were not fully (ir)relevant~~
5. Compute the final result

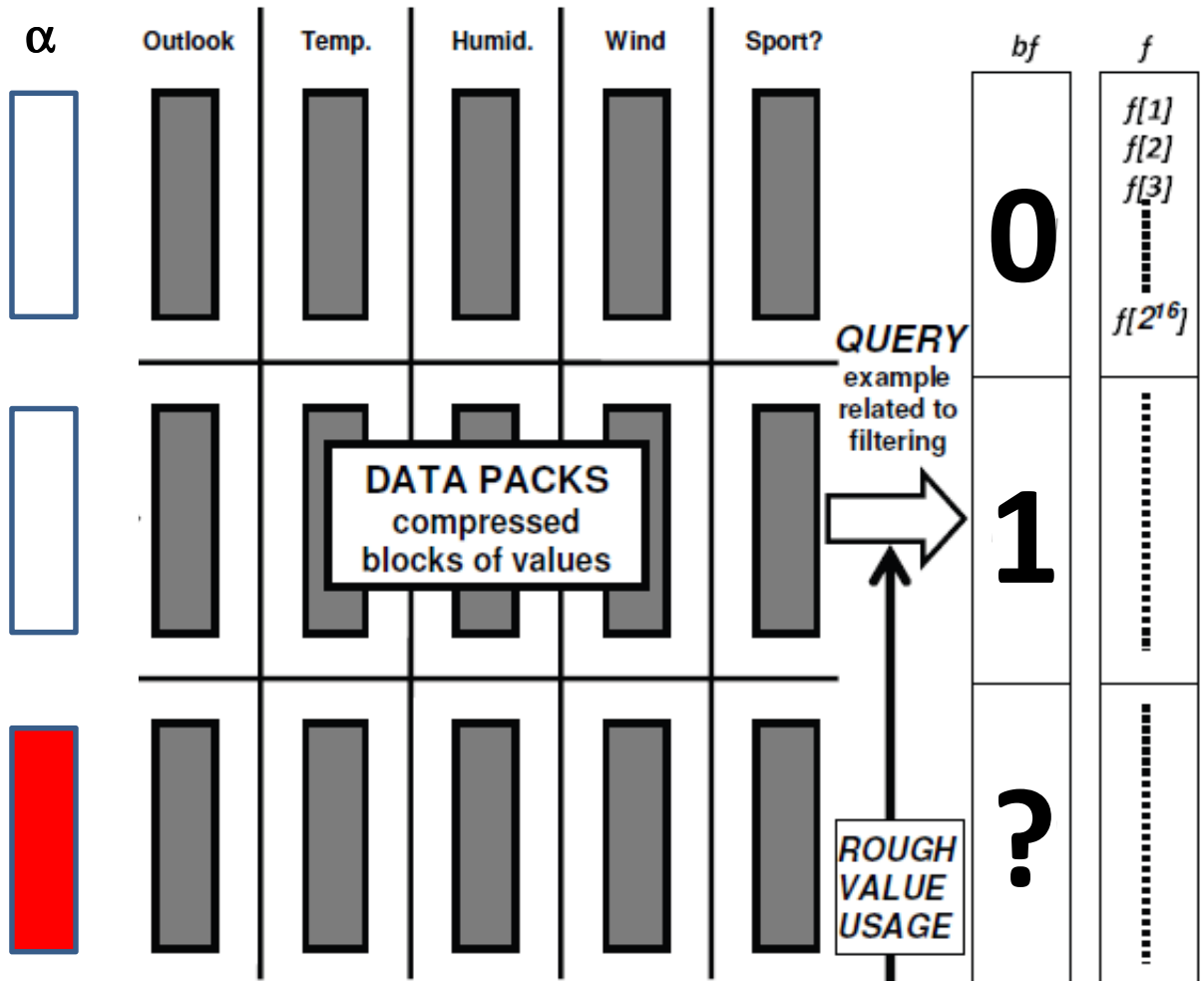


GRANULATED TABLE
physically, a collection of rough values for each of rough attributes is stored as a separate knowledge node

identification of blocks and rows satisfying query conditions

**SELECT COUNT(*)
FROM TABLE WHERE
 $\alpha(\text{Outlook}, \text{Temp.}) = x;$**

1. Compute rough attribute of $\alpha(\text{Outlook}, \text{Temp.})$
2. Filter out fully irrelevant virtual data packs of α
3. Utilize rough values of fully relevant virtual packs to compute the partial result
4. Create packs of α , which were not fully (ir)relevant
5. Compute the final result



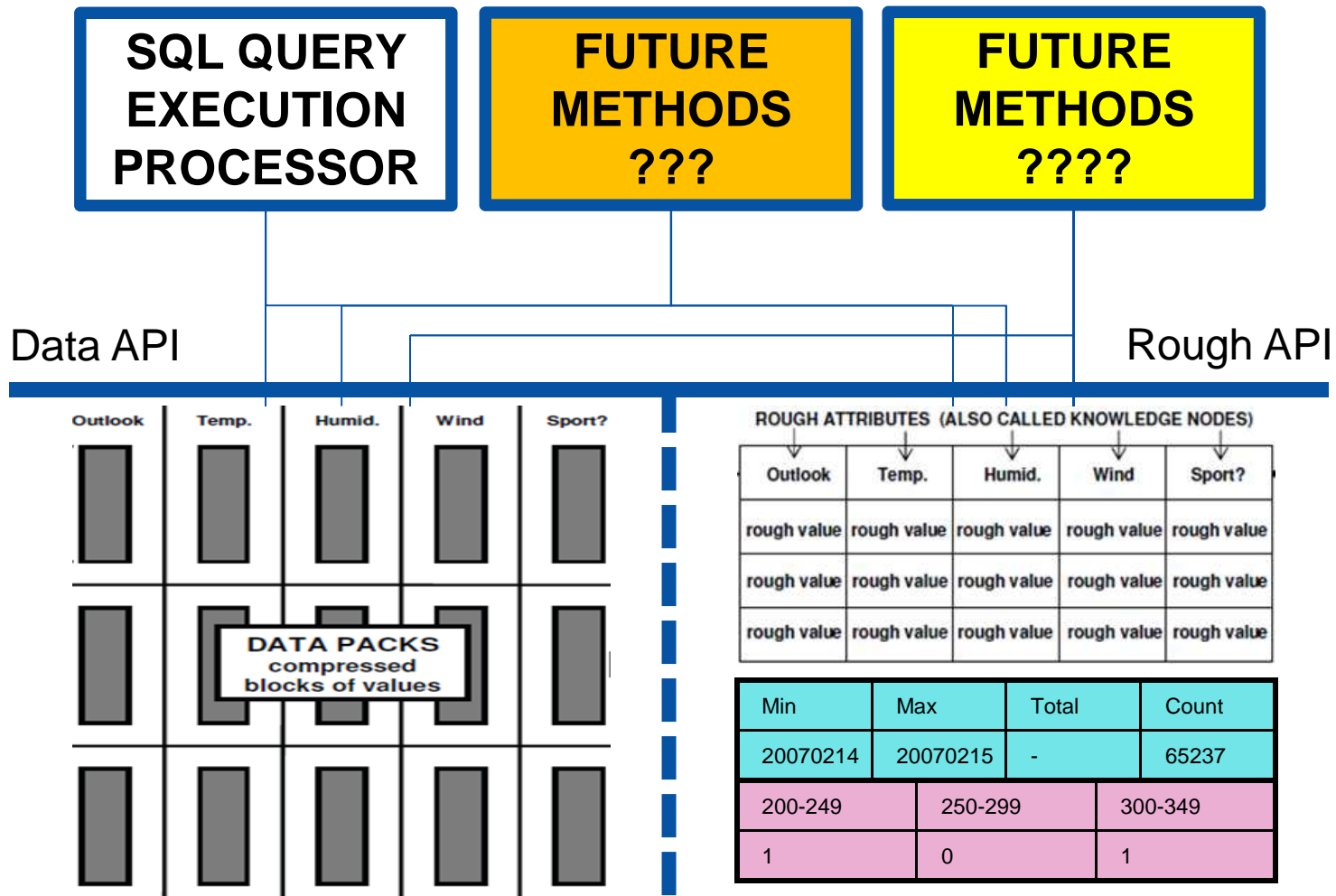
ROUGH ATTRIBUTES (ALSO CALLED KNOWLEDGE NODES)

α	Outlook	Temp.	Humid.	Wind	Sport?
rough value	rough value	rough value	rough value	rough value	rough value
rough value	rough value	rough value	rough value	rough value	rough value
rough value	rough value	rough value	rough value	rough value	rough value

GRANULATED TABLE
physically, a collection of rough values for each of rough attributes is stored as a separate knowledge node

identification of blocks and rows satisfying query conditions

Computing with Rough Approximations



Selected Papers about Infobright

- D. Ślęzak, P. Synak, J. Wróblewski, J. Borkowski, G. Toppin: Rough Optimizations of Complex Expressions in Infobright's RDBMS. RSCTC 2012: 94-99
- D. Ślęzak, P. Synak, G. Toppin, J. Wróblewski, J. Borkowski: Rough SQL – Semantics and Execution. IPMU 2012(2): 570-579
- D. Ślęzak, P. Synak, J. Borkowski, J. Wróblewski, G. Toppin: A Rough-Columnar RDBMS Engine – A Case Study of Correlated Subqueries. IEEE Data Eng. Bull. 35(1): 34-39 (2012)
- M. Kowalski, D. Ślęzak, G. Toppin, A. Wojna: Injecting Domain Knowledge into RDBMS – Compression of Alphanumeric Data Attributes. ISMIS 2011: 386-395
- D. Ślęzak, G. Toppin: Injecting domain knowledge into a granular database engine: a position paper. CIKM 2010: 1913-1916
- D. Ślęzak, P. Synak, J. Wróblewski, G. Toppin: Infobright Analytic Database Engine Using Rough Sets and Granular Computing. GrC 2010: 432-437
- D. Ślęzak, M. Kowalski: Towards Approximate SQL – Infobright's Approach. RSCTC 2010: 630-639
- D. Ślęzak, M. Kowalski: Intelligent Data Granulation on Load: Improving Infobright's Knowledge Grid. FGIT 2009: 12-25
- D. Ślęzak, V. Eastwood: Data warehouse technology by Infobright. SIGMOD Conference 2009: 841-846
- D. Ślęzak, J. Wróblewski, V. Eastwood, P. Synak: Brighthouse: An Analytic Data Warehouse for Ad-hoc Queries. PVLDB 1(2): 1337-1345 (2008)

Conclusions

- Rough set methods are very simple: they operate with three basic notions: rough set approximation, attribute reduction and (in)discernibility of objects
- Rough set methods are very powerful: one can modify the above three notions with only minor changes with respect to algorithmic framework
- Rough set principles can be utilized in many areas of mainstream research and applications, such as, e.g., database solutions and machine learning





INFOBRIGHT

THANK YOU VERY
MUCH ONE MORE
TIME!!!

slezak@mimuw.edu.pl
slezak@infobright.com
www.roughsets.org

