A BOOLEAN MODELING FOR IMPROVING THE ALGORITHM APRIORI

Abdelhak Mansoul[1] and Baghdad Atmani[2]

[1]Computer Science Laboratory of Oran (LIO),
Department of Computer Sciences, University of Skikda, Algeria.
mansoul.abdelhak@yahoo.fr
[2]Computer Science Laboratory of Oran (LIO),
Department of Computer Science, University of Oran ES-Sénia, Algeria.
atmani.baghdad@gmail.com

ABSTRACT

Mining association rules is one of the most important data mining tasks. Its purpose is to generate intelligible relations between attributes in a database. However, its use in practice is difficult and still raises several challenges, in particular, the number of learned rules is often very large. Several techniques for reducing the number of rules have been proposed as measures of quality, syntactic filtering constraints, etc. However, these techniques do not limit the shortcomings of these methods. In this paper, we propose a new approach to mine association, assisted by a Boolean modeling of results in order to mitigate the shortcomings mentioned above and propose a cellular automaton based on a boolean process for mining, optimizing, managing and representing of the learned rules.

KEYWORDS

Cellular automaton, Data mining, Association Rules, Boolean modeling, Apriori-Cell

1. INTRODUCTION

Numerous studies on the association rules are made [2], [9], [11]. However, their uses in practice are difficult and still raises many challenges, especially the exorbitant number of rules learned, and the processing time. Recent studies have also proposed a series of solutions to improve the performance of the mining process [4], [5], [7], [15], [16], without eliminating the shortcomings of this method of search data. It became necessary to find adequate techniques and algorithmic solutions to minimize the cost for space and computing time. The Apriori algorithm introduced an approach called "test-and-generate" with pruning. However, this approach suffers from a number of candidates that generates, particularly for relatively small values of support. However, these approaches do not limit the shortcomings of these methods. Given this situation, it became necessary to invest in new methods to faces the following challenges:

- Find heuristics to prune the search space;
- Find technical or algorithmic solutions, specifically adequate data structures, to minimize the cost in space or in process time.

David C. Wyld et al. (Eds) : SAI, CDKP, ICAITA, NeCoM, SEAS, CMCA, ASUC, Signal - 2014
pp. 105–114, 2014. © CS & IT-CSCP 2014 DOI : 10.5121/csit.2014.41111

We will expose in our present article, the second part of our study (see 3.2 Step 4) and its experimentation that was performed with the basic Apriori in order to demonstrate relevance and efficiency of the approach that we have considered. Later (continuation of our study) we will present the first part (see 3.2, Steps 2 and 3), we adopt Apriori-Cell.

2. RELATED WORK

Recent studies have proposed a series of solutions to improve the performance of extraction process of frequent item sets, including cellular automata [10]. Solutions were oriented essentially on the Reduction of I / O and the minimization of the cost of the step of computing the support [9]. Other studies have been based on the discovery of "closed" item sets arising from the theory of formal concepts [11]. Others propose to generate a representative base or generic association rules [14] [16] and used techniques to reduce the number of rules with the use quality measurements [15], syntactic filtering by constraints [2].

3. THE PROPOSED APPROACH

We propose a new approach being located at the junction of two domains that are the Knowledge Discovery from Databases (KDD) on one hand and representation of knowledge from the other. Our approach proceeds in three steps:

1. Extraction of frequent patterns and generating association rules using the algorithm Apriori-Cell which operates on a cell basis;

2. Boolean modelling for association rules;

3. Rules management by the inference engine ICR of the cellular automaton.

A cellular automaton is a grid composed by cells which change states in discreet steps. After each step, the status of each cell is modified as states of its neighbors in the previous step [1].

Our approach is implemented by two modules:

- The module MAR (Mining Association Rules) ;

- The module ICR (Induction of Cellular Rules).

The dynamics of the cellular automaton. The inference engine of the cellular automaton simulates the basic operating principle of a classical inference engine using two finite layers of finite automata. The first layer CEL Fact/CEL Item for the basic facts/Items and the second layer, CEL Rule/CEL Transaction for the basic Rules/Transactions. Each cell at time t+1 depends only on the status of its neighbors and his own at time t.

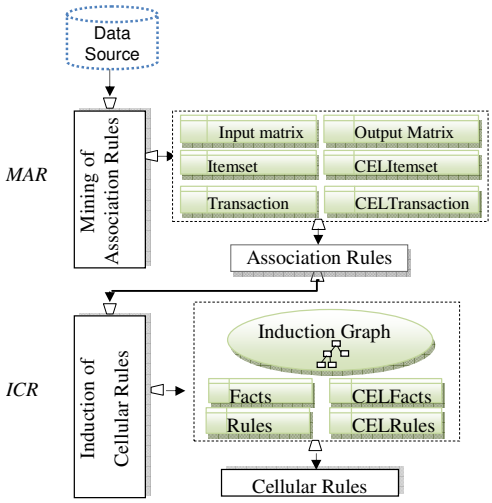


Figure 1. System Architecture

In each layer, the content of a cell determines if and how it participates in each inference step. At each step, a cell may be active (1) or passive (0) i.e whether or not participating in the inference. The principle adopted is simple:

- Any cell i of the first CEL Fact/CEL Item layer is considered an established fact if its value is 1, otherwise it is considered as fact to establish. It is presented in three states: input state (EF/EI), internal state (IF/II) and output state (SF/SI);

- Any cell j of the second layer CEL Rule/CEL Transaction layer is considered a Rule/Transaction candidate if its value is 1, otherwise it is considered as a Rule/ Transaction which shall not participate in the inference. It is presented in three states: input state (ER/ET), internal state (IR /IT) and output state (SR/ST).

- Incidence matrix RE and RS represents the input / output relation of Facts/Items and are used in forward chaining and backward chaining by reversing their order.

Thus, the cellular automaton will help optimize the representation of extracted knowledge (association rules) by the boolean principle and their management by using its inference engine through the basic functions δfact and δrule which provide the dynamics of cellular automaton (See 4.1).

3.1.The Proposed Algorithm

The process adopted by our system is a succession of four major steps:

Step 1: Selection and data preprocessing;
Step 2: Cellular representation of preprocessed data;
Step 3: Data mining by the cellular automaton using the algorithm *Apriori-Cell*;
Step 4: Post-processing of results.

Algorithm : Apriori-Cell

Input : Transactional-data-base (D), minimum-support *S*, confidence *C*
Output : lists-of-frequent-items (F_n), Association-rules-base (Br)
$F_n \leftarrow \{\}$
Begin
Input-matrix = data preprocessing for (D)
While we can make joints Do
  For each line of input-matrix Do
    calculate support for (item)
    If support(item) >= *S* Then $F_n \leftarrow F_n + \{item\}$
    EndIf
 EndFor
EndWhile
Br = Generate-rules (F_n)
End

3.2.Principle of the algorithm Apriori-Cell

This module applies the cellular principle on the basic Apriori algorithm for mining frequent itemsets. It simulates the basic operating of a join engine inspired by Apriori adapted to cellular automaton, on two finite layers of a finite automaton. the first layer *CELItems* for the items base, and the second layer *CELTransactions* for the transactions base.

The state of each cell at time t +1 depends only on the state of its neighbors and his own at time t. In each layer, the contents of a cell determines whether and how it participates in each inference step: a cell can be active (1) or passive (0), i.e whether or not participating in the inference.

The principle is simple, we suppose that there are *l* cells in the layer *CELItems*, and *r* cells in the *CELTransactions* layer. The states of the cells are: *EI, II, and SI*, respectively *ET, IT* and *ST* are the input, the internal state and the output of a cell of *CELItems*, and respectively a cell of *CELTransactions.* The internal state *II* of a cell of *CELItems* indicates the status of the item: in the case of an item, II = 1 corresponds to a state type *support_item* $\geq$ *minsupport_fixed*. For a cell from *CELTransactions*, the internal state *IT* will always be equal to 1 (the transactions are always established).

The join applied by Apriori-Cell. In the first step the join is made between items using the logical AND, line by line, i.e, it fixes line 1 for example, and it'll do its join to the rest of lines. Once completed, it will go to the second line without considering line 1, this time. And this process continues until the join between items become impossible. At the first iteration, the join is made unconditionally, but beyond 2 items, it applies the following rule: for the join of k-items we must have (k-1)-items in antecedent of the rule to be common.

Generation rules. The module **Generate-rules** is used for the generation and validation of association rules from the lists of frequent *n-itemset* extracted by Apriori-Cell. This module allows to minimize the number of reading of the database by using to calculate the confidence of each rule, the data cubes, which help in the positioning of the lists of *n-itemsets* extracted on three dimensions. A dimension for transactions, another one for the antecedent and the last for the consequent of each rule.

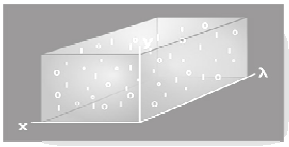


Figure 2. Representation of the data cube, with : x: axis of transactions (1,2) , y : axis of the list 1-frequent items (F1) (aac, acpP, aacA1) and λ : axis of the list l-frequent items (Fl)

Step 4 : post-processing of results

(a) Production of induction graph. An algorithm uses as input the association rules $\{R_i\}$, items of $Antecedent_i$ and $Consequent_i$, and it will give on output an induction graph, with a summit S_p and a node p on which we make a test with possible results binary or with multiple values.

(b) Generation of boolean rules from the induction graph. Induction graph is read to generate boolean rules in the following form:

$$Rb_k : \{ P_k \} \text{ Then } \{ C_k , S_p \}$$

(c) Representation of boolean rules. The generated rules (see step 4.b) are represented by cell layers where: $\{ Rb_k \}$ gives the set of rules {Rules} and $\{ P_k , C_k , S_p \}$ gives the set of facts {Facts}

(d) Integration. The cellular automaton integrates the generated rules in the knowledge base for use through different inference strategies.

4. ILLUSTRATIVE EXAMPLE OF THE REPRESENTATION OF RULES BY CELLULAR AUTOMATA

We suppose have obtained the following two rules of association with genes aceA-2, pstS-3, argC and phhB, using the Apriori-Cell algorithm:

R_1 : {aceA-2=1} , {pstS-3=1} , 45, 77

R_2 : {aceA-2=1, phhB=1} , { argC=1 } , 45, 70

and that these two rules have generated the following boolean rules from the induction graph :

Rb_1: Si { S_0 } Alors { pstS-3=1, S_1 }

Rb_2: Si { S_1 } Alors { argC=1, S_2 }

These rules will be represented (step 4, b) in layers CELFacts, CELRules, input matrix (R_E) and output matrix (R_S).

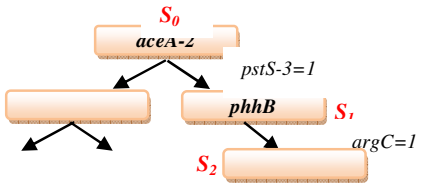
Step 3 : Data mining by the cellular automaton

From the sample test data base (see 5. Table1), we proceed to data mining by Apriori-Cell. We suppose have obtained two association rules with the following genes: aceA-2, pstS-3, argC and phhB.

Rule	*antecedent*	*consequent*	*Support %*	*Confidence %*
R_1	*aceA-2=1*	*pstS-3=1*	*45*	*77*
R_2	*aceA-2=1, phhB=1*	*argC=1*	*45*	*70*

Step 4 : post-processing of results

a) Production of the graph induction

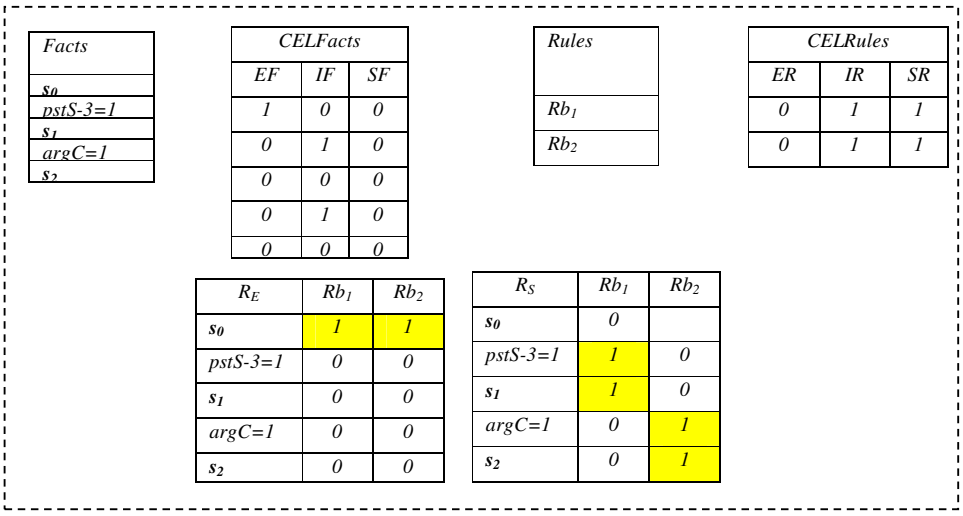


b) Generation of boolean rules from the graph induction

Rb_1: *If { S_0 } then { pstS-3=1, S_1 }* Rb_2: *If { S_1 } then { argC=1, S_2 }*

c) Representation of boolean rules

The boolean rules Rb_1 and Rb_2 produced are represented by the layers *CELFacts* (Facts + CELFacts) and *CELRules* (Rules + CELRules) and input matrix (R_E) and output matrix (R_S).



Facts
s_0
pstS-3=1
s_1
argC=1
s_2

CELFacts		
EF	*IF*	*SF*
1	0	0
0	1	0
0	0	0
0	1	0
0	0	0

Rules
Rb_1
Rb_2

CELRules		
ER	*IR*	*SR*
0	1	1
0	1	1

R_E	Rb_1	Rb_2
s_0	1	1
pstS-3=1	0	0
s_1	0	0
argC=1	0	0
s_2	0	0

R_S	Rb_1	Rb_2
s_0	0	
pstS-3=1	1	0
s_1	1	0
argC=1	0	1
s_2	0	1

Figure 3. Cell layers of the cellular automaton with input state (EF), internal state (IF) and output state (SF), input state (ER), internal state (IR) and output state (SR)

4.1. Simulation of The Cellular Inference Engine

The cellular automaton simulates the operating of an inference engine by using the transition functions previously mentioned, δ_{fact} and δ_{rule}.

We show that simulation starting from *CELFacts* and *CELRules* of the example shown previously (Figure 3) by considering that G_0 is the initial configuration of the cellular automaton, and $\Delta = \delta rule \; o \; \delta fact$ the global transition function: $\Delta(G_0) = G_1$.

The cellular automaton change state from G_0 to the state G_1 with

$\Delta(G_0) = G_1$ if $G_0 \xrightarrow{\delta_{fact}} G'_0$ and $G'_0 \xrightarrow{\delta_{rule}} G_1$

After application of the law of global transition Δ we obtain the configurations $\mathbf{G_1}$, $\mathbf{G_2}$ and finally $\mathbf{G_3}$.

1. G_0 is the initial configuration of the cellular automaton

G_0

Facts
s_0
pstS-3=1
s_1
argC=1
s_2

CELFacts		
EF	IF	SF
1	0	0
0	1	0
0	0	0
0	1	0
0	0	0

Rules
Rb_1
Rb_2

CELRules		
ER	IR	SR
0	1	1
0	1	1

2. Evaluation, selection and filtering (application of ***δfact***)

Facts
s_0
pstS-3=1
s_1
argC=1
s_2

CELFacts		
EF	IF	SF
1	0	1
0	1	0
0	0	0
0	1	0
0	0	0

Rules
Rb_1
Rb_2

CELRules		
ER	IR	SR
1	1	0
0	1	1

3. Execution (application of ***δrule***) $\Delta(\mathbf{G_0}) = \mathbf{G_1}$

G_1

Facts
s_0
pstS-3=1
s_1
argC=1
s_2

CELFacts		
EF	IF	SF
1	0	1
1	1	0
1	0	0
0	1	0
0	0	0

Rules
Rb_1
Rb_2

CELRules		
ER	IR	SR
1	1	0
0	1	1

4. Application of the global transition function : $\Delta = \delta rule \; o \; \delta fact \;\; \Delta(\mathbf{G_1}) = \mathbf{G_2}$

G_2

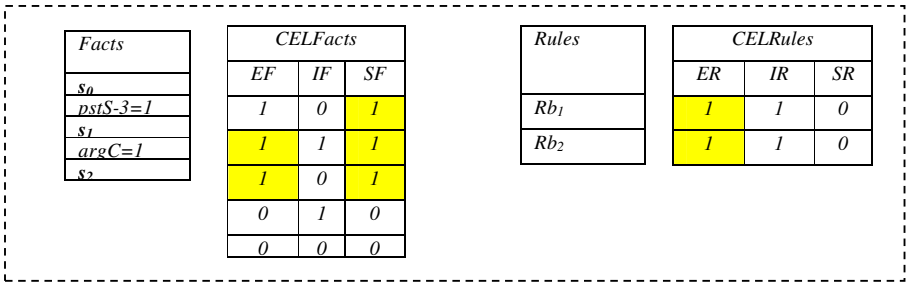
Facts
s_0
pstS-3=1
s_1
argC=1
s_2

CELFacts		
EF	IF	SF
1	0	1
1	1	1
1	0	1
0	1	0
0	0	0

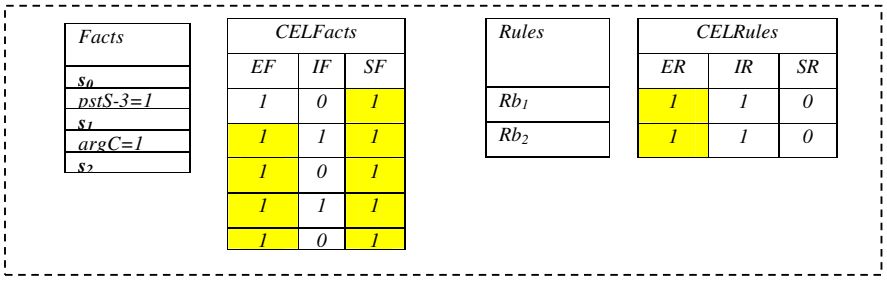
Rules
Rb1
Rb2

CELRules		
ER	IR	SR
1	1	0
0	1	1

5. Evaluation, selection and filtering (application of ***δfact***)



6. Application of the global transition function



Final configuration $\mathbf{G_3}$ obtained after four iterations.

5. EXPERIMENTATION

To examine the effectiveness in practice of our system, we have implemented the engine, and we conducted experimental tests on a machine (Intel Celeron 540 CPU frequency 186 GHz, 512 MB RAM) with a sample test data base (Table 1) representing the genomic sequences mycobacterium tuberculosis) with the first 12 genes of each strain, and the assumption that these genes are sufficiently distinctive and representative of each strain taken separately.

Table 1. Test Data Base [1] (12 genes of each strain)

Strain	**Genes**
Mt CDC155	*aac accD aceA-1 aceA-2 aceB aceE ackA acnA acp-1 acp-2 acpP acpS*
Mt F11	*aceE acpP acpS adk alaS alr argC argD argJ argS aroB aroE*
Mt H37Ra	*aac aao accA1 accA2 accA3 accD1 ccD2 accD3 accD4 accD5 accD6 aceAa*
Mt H37Rv	*35kd_a aac aao accA1 accA2 accA3 accD1 accD2 accD3 accD4 accD5 accD6*

5.2.Discuss of The Results

Processing time. We observe that the Apriori algorithm takes an important part in execution time of the system in whole, ie in its most important phases as the generation of association rules by Apriori and the generation of boolean rules.

Table 2. Evolution of execution time (Basic Apriori and global)

Confidence %	Support %	Number of Genes	Generated items	Number of rules	Execution of Apriori	Global Execution
10	30	12	37	69	0.00 s	0.00 s
50	50	12	37	125874	0.67 s	1.69 s
70	60	12	37	786756	3.56 s	6.17 s

Storage space. We find that cell representation is more interesting, and it will be much most prominently with a more substantial sample.

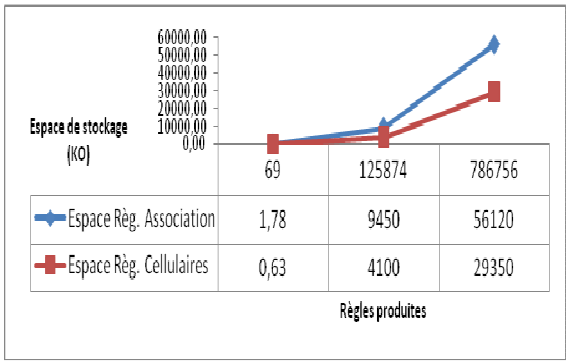


Figure 4 . Evolution of storage space

6. CONCLUSION

After describing the disadvantages of rule-based methods in data mining, we have proposed an extraction rules guided by a Boolean modeling based on the Boolean principle of cellular automata in order to have a base rules optimized and reduced processing time enough, and thus make a contribution to the construction of knowledge-based systems by adopting a new cellular technic Thus, the advantages of our method based on the cellular automaton can be summarized as follows:

- Simple and minimal preprocessing of association rules base, for its transformation into binary matrix according to the principle of cell layers,
- Ease of implementation functions δfact and δrule that are low complexity and well adapted to situations with many attributes of rules.

REFERENCES

[1] Atmani, B., Beldjilali, B. (2007) "Knowledge Discovery in Database : induction graph and cellular automaton", Computing and Informatics Journal, Vol. 26 N°2 171-197

[2] Besson, J., Robardet, c., Boulicaut, J.F. (2004) "Constraint-based mining of formal concepts in transactional data", Conference on Knowledge Discovery and Data Mining (PAKDD'04) volume 3056 of LNCS, Sydney- Australia, pp. 615–624

[3] Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L., Stumme, G. (2000) "Mining minimal non-redundant association rules using frequent closed itemsets", Proceedings of the Intl. Conference DOOD'2000, LNCS, Springer-verlag, July, pp. 972-986

[4] Hajek, P., Havel, I., Chytil, M. (1966) "The GUHA method of automatic hypotheses determination", Computing 1, pp. 293-308

[5] Boullé, M. (2007) "Recherche d'une représentation des données efficace pour la fouille des grandes bases de données", Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications Paris.
[6] Abdelouhab, F., Atmani, B. (2008) "Intégration automatique des données semi-structurées dans un entrepôt cellulaire", Troisième atelier sur les systèmes décisionnels, Mohammadia– Maroc, 10-11 octobre, pp. 109-120
[7] Agrawal, R., Imielinski, T., Swami, A. (1993) "Mining associations between sets of items in large databases", Proc. of the ACM SIGMOD Conf., Washington DC, USA
[8] Agrawal, R., Srikant, R. (1994) "Fast Algorithms for Mining Association Rules", Proceedings of the 20th International Conference on Very Large Data Bases, Santiago, Chile, pp 487-499
[9] Bykowski, A., Rigotti, C. (2001) "A condensed representation to find frequent patterns", Proceedings of the Twentieth ACM SIGACTSIGMOD SIGART Symposium on Principles of Database Systems, ACM Press, pp. 267-273
[10] Fawcett, T. (2008) " Data mining with cellular automata", ACM SIGKDD Explorations Newsletter,v.10 n.1, June 2008.
[11] Ganter, B., Wille, R. (2004) "Conceptual graphs and formal concept analysis", Lecture Notes in Computer Science Volume 1257, 1997, pp 290-303Springer-Verlag
[12] Mansoul, A., Atmani, B. (2009) "Fouille de données biologiques : vers une représentation booléenne des règles d'association", CEUR-WS:04-Dec-2009/Vol-547, Conférence Internationale sur l'Informatique et ses Applications CIIA'09, Saida – Algérie
[13] Mansoul, A., Atmani B. (2010) "Vers un automate cellulaire pour la fouille de données : Partie I : la représentation booléenne des règles d'association", ASD 5/6 Novembre 2010, Sfax, Tunisie.
[14] Pasquier, N., Bastide Y., N., Lakhal, L. (2000) "Mining minimal non-redundant association rules using frequent closed itemsets", Computational logic. International conference No1, London, royaume uni (24/07/2000), vol. 1861, pp. 972-986.
[15] Vaillant, B., Meyer, P., Prudhomme, E., Lallich, S., Lenca, P., Bigaret, S. (2005) "Mesurer l'intérêt des règles d'association", Atelier Qualité des Données et des Connaissances (DQK 05, Actes de EGC), pp. 69-78
[16] Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L. (2001) "Intelligent structuring and reducing of association rules with formal concept analysis", Proc. KI'2001. conference, LNAI 2174, Springer.

AUTHORS

Abdelhak Mansoul is an Assistant Professor at Skikda University and affiliated researcher in Oran Computer Lab of Oran University. His research interests are in Database Management System, Data Mining, decision support systems, and simulation.

Baghdad Atmani is a professor in computer science at the University of Oran (Algeria). His interest field is Data Mining and Machine Learning Tools. His research is based on Knowledge Representation, Knowledge-based Systems and CBR, Data and Information Integration and Modeling, Data Mining Algorithms, Expert Systems and Decision Support Systems.His research are guided and evaluated through various applications in the field of control systems, scheduling, production, maintenance, information retrieval, simulation, data integration and spatial data mining.