Rule Schema Multi-Level for Local Patterns Analysis: Application in Production Field

Salim Khiat[1], Hafida Belbachir[2], and Sid Rahal[3]
[1]Computer Sciences Department, University of science and technology–Mohamed Boudiaf Oran, Algeria
[2]The Science and Technology University USTO, Algeria
[3]System and Data Laboratory (LSSD)

Abstract: *Recently, Multi-Database Mining (MDBM) for association rules has been recognized as an important and timely research area in the Knowledge Discovery Database (KDD) community. It consists of mining different databases in order to obtain frequent patterns which are forwarded to a centralized place for global pattern analysis. Various synthesizing models [8,9,13,14,15,16] have been proposed to build global patterns from the forwarded patterns. It is desired that the synthesized rules from such forwarded patterns must closely match with the mono-mining results, ie. the results that would be obtained if all the databases are put together and mining has been done. When the pattern is present in a site but fails to satisfy the minimum support threshold value, it is not allowed to take part in the pattern synthesizing process. Therefore this process can lose some interesting patterns which can help the decision maker to make the right decisions. To adress this problem, we propose to integrate the users knowledge in the local and global mining process. For that we describe the users beliefs and expectation by the rule schemas multi-level and integrate them in both the local association rules mining and in the synthesizing process. In this situation we get true global patterns of select items as there is no need to estimate them. Furthermore, a novel Condensed Patterns Tree (CP_TREE)structure is defined in order to store the candidates patterns for all organization levels which can improve the time processing and reduce the space requirement. In addition CP_TREE structure facilitate the exploration and the projection of the candidates patterns in differents levels. finally We conduct some experimentations in real world databases which are the production field and demonstrate the effectivlness of the CP_TREE structure on time processing and space requirement.*

Keywords: *Schema, association rules, exceptional rules, global rules, ontology.*

Received July 22, 2014; accepted August 12, 2015

1. Introduction

Database mining has emerged as a major application area for efficient discovery of the previously unknown and potentially useful patterns in large databases. Much of the data mining techniques developed in early 90s focused on the centralized database. Rapid strides made in the communication network technology and distributed, federated and homogeneous database systems have led to the development of several multi-database systems for real world applications. A multi-database environment consists of a group of databases or datasets distributed in a wide area network. However, many large organizations operate from multiple branches. Some of these ones collect data continuously. Thus, there are multi-branch organizations that process multiple databases. Global decisions made by such an organization might be more appropriate if they are based on the data distribution over the branches. For decision-making, large organizations need to mine their multiple databases distributed throughout their branches. Multi-Database Mining (MDBM) can be defined as the process of mining data from multiple databases, which may be heterogeneous, and finding novel and useful patterns of significance [7]. The local patterns analysis approach is probably the most used in the MDBM Process for association rule. It is performed in two steps: intra-site and inter-site processing. A traditional data mining is applied in the intra-site step in order to extract the local patterns. Afterwards, each branch forwards the discovered pattern base to the central office where they will be synthesized in the global ones and eventually makes decisions at central office. A pattern can be a frequent itemset or an association rule.

Furthermore, many works [8, 9, 13, 14, 15, 16] have been proposed to improve the global synthesizing process. They proceed by analyzing the local frequent patterns at different sites in order to discover other new and useful patterns. Indeed, to capture some global trends, these approaches are based on the linear equation which includes the sites weight and the patterns support. The notion of the data source weight has been largely studied in the literature [7, 11, 12]. On the other side, few studies address the problem of estimating the support of infrequent patterns in the synthesizing global pattern process. Effectively, when the pattern fails to acquire the minimum support threshold value in one site, it frequency vanishes and is not able to take part in the synthesizing process. In such circumstances, it doesn't

imply that the pattern is not present at all, because the pattern may have some significance in the site with a support value between 0 and minimum support. However to make the participation of those patterns and improve the synthesized results authors in [10] introduce a correction factor in the synthesized process in order to save some interesting patterns and obtain the results which tailed with the mono-mining results. A correction factor "h" is applied to the infrequent patterns in order to improve their support. The value of "h" is determined by lot of iteration until the results of multi-databases mining tailed with mono-database mining results. When the mean error is small between the two results authors chose the optimum value of "h".

They found that with h=0.5 the results converge to the mono-mining results. If we execute this algorithm in other databases we must recalculate the novel optimum value of "h". In addition, this algorithm needs to extract patterns in the mono-mining process in order to calculate the mean error between the synthesized value and the mono-mining result which consume the time computing. In the context of estimating support of itemsets in databases authors in [2] propose a method using Bonferroni-type inequalities which extend the inclusion-exclusion method. And authors in [5] use the maximum-entropy method to estimate the support of a general boolean expression. But these support estimation techniques are suitable for a single database only. Khiat andall [3] recently address this problem by using the probabilistic models for synthesizing the global pattern in MDBM process. They applied the maximum entropy model in inter-site step and use both the clique and bucket elimination in order to reduce the complexity of the maximum entropy method.

Experiments in [3] show that the results accuracy for the proposed synthesizing process is nearly than the mono-mining results. But the results still approximating and the fewer lost patterns can be very interesting for the users. To address this problem we propose in this paper to integrate the user belief and expectation on the *MDBM* process since we one does not need to estimate the patterns in multiple databases.

This paper is organized as follows. Section 2 describes the related work. Section 3 defines the proposed model. In section 4, several experiments have been conducted for evaluating the proposed approach. In the last section we conclude this paper.

2. Related Work

In this section we survey the integration of the user belief and expectation for association rules mining. We first describe approaches based on a single database research and second we expose others in MDBM literature.

Liu *et al*. [4] are the first authors that integrate the user beliefs in the mono-database mining process. They proposed a new framework to allow the user to explore the discovered rules in order to identify those interesting ones. This framework has two components, an interestingness analysis component, and a visualization component. The interestingness analysis component analyzes and organizes the discovered rules according to various interestingness criteria with respect to the user's existing knowledge. The visualization component enables the user to visually explore those potentially interesting rules. After this interestingness analysis component was developed by [1] where she proposed a new approach to prune and filter discovered rules. She addressed two main issues: The integration of user knowledge in the discovery process and the interactivity with the user. The first issue requires defining an adapted formalism to express user knowledge with accuracy and flexibility such as ontologies in the Semantic Web. Second, the interactivity with the user allows a more iterative mining process where the user can successively test different hypotheses or preferences and focus on interesting rules. For that she proposed a new rule-like formalism, called rule schema, which allows the user to define his expectations regarding the rules through ontology concepts. She applied the proposed framework successfully over the client database provided by Nantes Habitat.

In MDBM research, authors in [8] introduce the concept of select items in multi-database mining. First they propose a model of mining global patterns of select items from multiple databases and second present a measure of quantifying an overall association between two items in a database and finally they present an algorithm that is based on the proposed overall association between two items in a database for the purpose of grouping the frequent items in multiple databases. Each group contains a select item called the *nucleus* item and the group grows while being centered on the *nucleus* item.

In this paper, we report our recent work in addressing the association rule mining at multiple-levels of abstraction. We develop the rule schema proposed by [1] in order to represent user belief at different levels in the organization. We propose also a new-like formalism, called rule schema multi-level which allows the user of different levels in the organization to define their expectations regarding the rules through ontology concepts. In addition, we propose a new set of operators over each rule schema for interactive processing that these users can choose. Finally we propose a synthesizing process with the exact rule support which will solve the problem of the estimated rules support in MDBM process.

3. Proposed Local Mining Method

Rule Schemas Multi-Levels for Local Patterns Analysis (RSMLPA) is the proposed model for local patterns analysis. It proposes to select only the

association rules that are interesting for the user at different levels of abstraction in the organization. MDBM process works in local and global step.

In this context, MDBM process is executed in two steps: first, intra-site processing where association rules are generated by integrating the user knowledge in the rules mining process, next, inter-site processing selects only the interesting ones for each organization level. Figure 1 present the proposed RSMLPA algorithm.

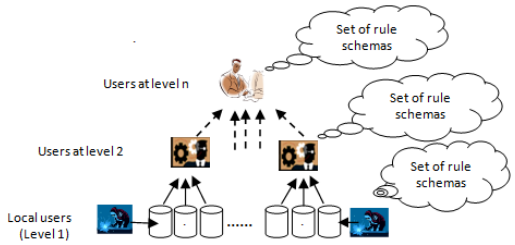


Figure 1. RSMLPA algorithm.

3.1. Intra-Site Processing

RSMLPA algorithm must take part the beliefs of different kinds of users in a typical organization. Figure 1 shows an organization with different branches and sites which represents a multi-level organization. Users at level 1 express their beliefs and expectations for extracting the local knowledge from the local sites.

Users at levels 2 express their beliefs and expectations in order to extract the global knowledge from the branches. And the last users at levels n express their beliefs and expectations in order to extract the global knowledge from all the organization. For an effective representation of the beliefs of the different users, we propose a model to represent user knowledge. This model must take part of the multi-level organization of the company. First, we propose a new rule-like formalism, called rule schema multi level which allows the different users to define their expectations regarding the rules through ontology concepts at different levels in the organization. Second users can choose among a set of operators for interactive processing the one to be applied over each Rule Schema (i.e., k-conforming, k-Objective...). The process of RSMLPA presented in Figure 2 aims to guide the user through the mining rules process phase. Several steps are suggested as follows:

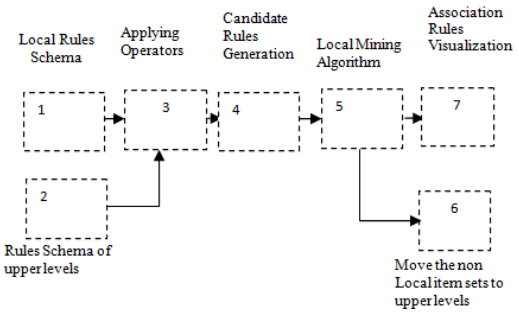


Figure 2. Intra-Site processing.

- *Steps* 1 and 2. Rule schema multi-level formalism (Local and Uppers levels)**:** To improve association rule selection, we propose a rule filtering model, called rule schema multi-level. In other words, a rule schema describes, in a rule-like formalism, the user expectations in terms of interesting rules at different levels on the organization. As a result, rule schemas act as a rule grouping, defining rule families. The base of rule Schema formalism is the user representation model introduced in [4] composed of: General Impression, Reasonably Precise Concepts and Precise knowledge. We propose to develop two of them: General Impression and Reasonably precise concepts. Thus, rule schemas bring the complexity of ontologies in rule mining combining not only item constraints, but also ontology concept constraints. Before formalizing the proposed rule schema multi-level in definition 2 let define the ontology confirmation concept.

Definition 1: Let us consider an ontology concept C associated in the database to: $F(C) = \{y_1,\ldots,y_n\}$, Where $\{y_1,\ldots,y_n\} \in I$ and an itemset $X = \{x_1,\ldots,x_m\}$ We say that the itemset X is conforming to the concept C if $conf(X, C) = TRUE$, where:

$$conf(X,C) = \begin{cases} TRUE & if \exists y_i, y_i \in X \\ FALSE & Otherwise \end{cases}$$

In other words, an itemset is conforming to an ontology concept if the latter is associated to at least one item of the itemset.

Definition 2: A rule schema multi-level is defined as: $<(X_1, X_2,\ldots. X_m) \rightarrow (Y_1, Y_2,\ldots. Y_k)(T)(N)>$, Where:

- X_i and Y_j are ontology concepts and the implication '$\rightarrow$' is optional.
- T={L,M,E,G} is the type of knowledge which can be Local(L), Majority (M), Exceptional (E) and Global rules (G).
- N is the level of the rule schema which indicates the level of users that formulate this one. The lower level (n=1) exposes the decision maker's belief in the lower organization level. The upper level n ($n \geq 0$) expresses the decision maker's belief in the head quarter of the organization.

If the implication '$\rightarrow$' is mentioned in the rule schema we say that the rule schema is an implication rule schema, it defines the reasonably precise concepts.

Meanwhile, If we do not keep the implication '$\rightarrow$' we define non implicative rules schemas generalizing general impressions.

For example, a rule schema $<(C_1, C_3 \rightarrow C_2)$ (M) (2)> Correspond to "all majority association rules whose condition verifies C_1 and C_3 and conclusion verifies C_2 at level 2".

- *Step* 3. Operations**:** From previous beliefs and knowledge, several operations can be designed that allow the user to explore the rule space. We

propose three intra-site operators: K-Conformation, K-Objective and K-Non Objective.

- K-Confirmation with K≥0, is the one of the primitive operations that may be performed on Rule Schemas. It finds all rules that comply with the support and confidence constraints and contain the items Condition in the antecedent, the items Conclusion in the consequent and any items with size=K in I-{Condition∪Conclusion} any of the two sides of the implication. Items in I-{Condition∪Conclusion} part may be split in any possible way between the antecedent and the consequent. More formally, researched rules are of the form: Condition→Conclusion∪({Condition∪Conclusion}), where |I-{Condition ∪Conclusion}|=K And Condition ∪ (I-{Condition ∪Conclusion})→ Conclusion, where |I-{Condition ∪Conclusion}|=K
- K-Objective with K≥0, allows the user to find the rules that have a more particular consequent that are defined in the rule schema. It finds all rules that comply with the support and confidence constraints and contain the items Condition in the antecedent, the items Conclusion in the consequent with any items Subset=I-{Condition ∪Conclusion} with size=K in the consequent side of the implication. More formally, researched rules are of the form: Condition∪ I-Subset→ Conclusion∪(I-{Condition∪Conclusion}), where subset=I-{Condition ∪Conclusion} and |subset|=K
- K-Non Objective with K≥0, allows the user to find the rules that have a more particular condition that are defined in the rule schema. It finds all rules that comply with the support and confidence constraints and contain the items Condition in the antecedent with any items Subset=I-{Condition ∪Conclusion} with size=K, the items Conclusion in the consequent. More formally, researched rules are of the form: Condition∪ Subset → Conclusion ∪ (I-Subset), where subset=I-{Condition ∪Conclusion} and |subset|=K
- *Step* 4. Candidates Rules Generation**:** Candidate's rules are all possible rules that are conforming to the specified schemas and operations. After generation, a pass through the database is performed in which the support and the confidence of candidate rules are computed. In order to be present in the output, rules must comply with the support and confidence requirements specified and the others rules which do not satisfy the support and confidence are transferred into the uppers level.

The generation of the rules candidates is performed by combining all the possible combination of the items that satisfy the operation.

Rule schema+operation→Rules candidates→Itemsets candidates

Example 1: Let the rule schema RS1 (<A→B> <L> <1>) and the set of items in databases is I={A,B,C} and the 1-confirmation operator is applied over this rule schema.

Candidates Rules are: CR_{11}: A→B; CR_{12}: A, C→B; CR_{13}: A→B, C, Candidates Item sets are: A, AB for the first candidate rule; AC,ACB for the second; A, ABC for the last. We can remark that item set A, ABC are redundant, they must stored in one place.

And let the rule schema RS_2 (<A→C><G><2>) and the set of items in databases is I= {A,B,C} and the 1-confirmation operator is applied over the rule schema.

Rules candidates are: CR_{21}: A→C; CR_{22}: A, B→C; CR_{23}: A→B, C, Item sets candidates are: A, AC for the first candidate rule; AB, ABC for the second; A, ABC for the last. We can remark that item set A, ABC are also redundant and they must stored in one place.

We can remark that RS_1 and RS_2 have one shared rule candidate A →B, C which may be stored in one place in memory.

The generation of the rules candidates of all the users is time and space consuming. Different users can express the same or nearly the same rule schema which can generate some common rules candidates. In order to address this problem we will propose a condensed structure that stores the shared candidates' rules and itemsets in one place.

Using efficient data structures and implementation is very important in improving the performance of the mining algorithm. We propose a condensed structure called Condensed Patterns Tree (CP_TREE). This one is efficient due to the following reasons:

- One could reduce rules and itemsets candidates' space memory.
- Facilitate the rules and itemsets candidates' research.
- Facilitate transferring the rules candidates to upper levels; we only perform a projection in this structure over the table header.

CP_TREE structure is a rooted, labeled tree as presented in Figure 3. It is composed on two parts: the index and the tree.

- The index is the table header that contains the levels of the organization. For each level an index is associated for indicating the first node of the tree that contains the rule candidate.

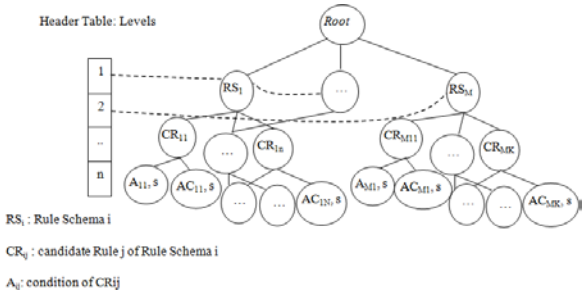


Figure 3. CP_TREE Structure.

The tree structure is composed of three kinds of nodes: first represent the rule schema; the second represent their candidates' rules and the last represent the condition and consequence of the candidate rule with their supports.

Example 2: The CP_TREE structure of the example 1 is presented in the Figure 4.

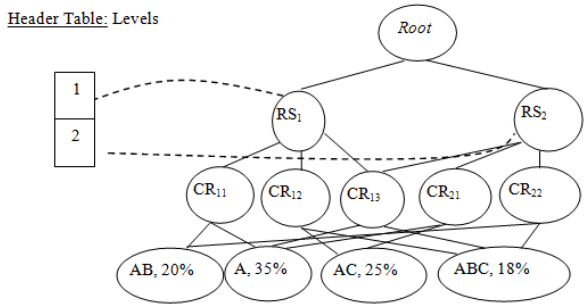


Figure 4. CP_TREE structure of the example 1.

- With the classical representation we need 18 spaces memory, but with the novel structure we need only 10 memory space. We can say that CP_TREE structure we save eight places instead with the classical representation. The CR_{23} is represented by the rule candidate CR_{13} and the itemset A is shared between CR_{11}, CR_{13}, CR_{21} and the item set AC is shared between CR_{12}, CR_{21} and the item set ABC is shared between CR_{12}, CR_{13}, CR_{22}.
- For the inter-site processing a simple projection is performed to the branch corresponding to the level 2. For that the remaining CP_TREE structure represented in the Figure 5 is transferred to the uppers level.

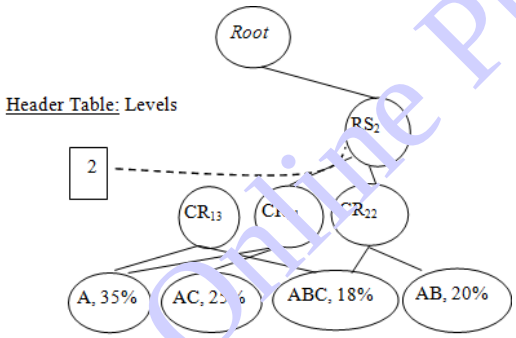


Figure 5. Projection of CP_TREE Structure of the example1.

- *Steps* 5 and 6. Local mining algorithm and move the non local item sets to upper levels**:** Association rule mining is widely used data mining approach for discovering patterns and relationships between variables from data. The Apriori [6] algorithm is one of the most commonly used methods for Association Rule. By an incremental approach, Apriori finds all frequent itemsets that have a support above a certain threshold. On the basis of the frequent itemsets, the algorithm builds all rules that have a confidence value above a given threshold. RSMLPA approach extract only interesting rules for that it integrates user knowledge and expectations into the rule mining process. In this approach, the search for interesting rules is done locally, in the neighborhood of rules and associations that the user believes to be true, specified by means of the rule schemas. Instead of generating all rules (by means of frequent itemsets), and filtering those that are conform to user knowledge, the new approach consists of first generating locally all candidate rules, based on the rule schemas of all the upper level and operators, and then checking their support and confidence against the transaction database. Rules that satisfy the support minimum will be presented to the users and the patterns that not satisfy the support minimum values are transferred to the upper levels for the synthesizing process. In this case we don't need to estimate them because we have the exact values of the support.
- *Step* 7. Visualization rules**:** The visualization phase is very important, proposing to the user the result of his research.

3.2. Inter-Site Step

The rule synthesizing process should generate meaningful rules which make sense with respect to the user's knowledge. It is proposed to get G, M and E set of synthesized rules, which are potentially useful for a multi-level organization in the decision-making process from the local rules. The synthesizing process is based on the user knowledge and expectations into the synthesizing process. Only interesting rules are synthesized into three groups: global, majority and exceptional rules. The construction of these groups is based on the rule schema and the operators as described in the Figure 6.

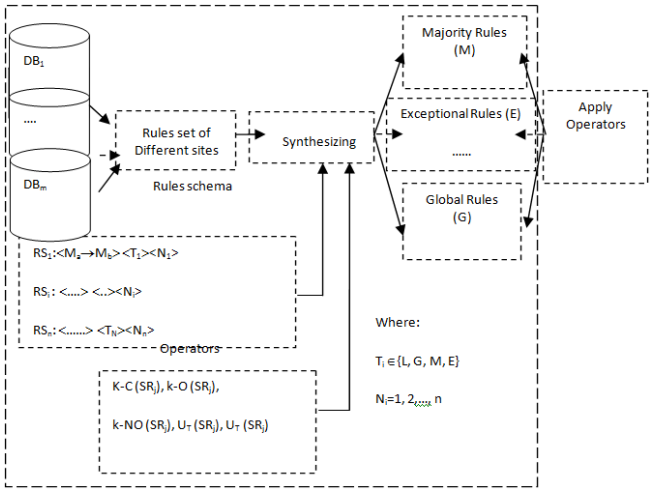


Figure 6. Synthesizing process.

Majority rules [16] can grasp the distribution of rules in local ones and reflect the "commonness" of branches in their voting. High-vote rules are useful for global applications of interstate companies.

Exceptional rules [15] can grasp the individuality of branches. It often present as more glamorous than high-vote rules in such areas as marketing, science discovery and information safety.

Global rules can grasp the globality of rules and reflect the distribution of the rules supports. It detects the global rules instead the mono-database mining. In other words, it reflects the global rules which are tailed with the mono-database mining. Our framework allows extracting the set of exact global rules.

Given n databases D_1, D_2,…,D_n, they represent the databases from n unites or plants of a large company. Let LP_1, LP_2,…, LP_n be the corresponding local patterns which are mined from every database; And minsupp be the user specified minimal support in all databases. For each pattern P, its support in D_i is denoted by $supp_i$ (P). We define the average vote of local patterns in the databases as follows.

$$AverVotes = \frac{\sum_{i=1}^{Num(Gp)} Num(P_i)}{Num(Gp)}$$

Where *Gp* means the Global Patterns, it is the set of all patterns from each database, that is $Gp=\{LP_1 \cup LP_2 \cup \ldots \cup LP_n\}$and *Num(Gp)* is the number of patterns in *Gp*. We regard the *AverVotes* as a boundary to identify exceptional patterns and high-voting patterns (Majority). If a pattern's vote is less than the *AverVotes*, then it will be considered as an exceptional pattern, otherwise as a high-voting pattern or majority patterns.

We say that pattern *P* is global if its global support is upper or equal to the *minsupp*:

$$supp_G(P) \geq minsupp$$

where:

$$supp_G(P) = \sum_{i=0}^{n} W(D_i) * supp_i(P)$$

$$W(Di) = \frac{W'i}{\sum_{i=1}^{n} W'i}$$

W'i is the transaction population of database *Di*.

In addition of the operators used in intra-site, type unexpectedness (U_T) operator is used for extracting the rule that contradict user knowledge type. Given a rule schema, an association rule is unexpected regarding the type if the type of the association rule is not conforming to the type of the rule schema, and if the antecedent and the consequent itemset of the association rule are conforming to each concept in the antecedent and the consequent of the rule schema.

Definition 3: Let us consider the following exceptional association rule A→B and a rule schema: RS (<M_A→M_B> <T> <N>), Where: MA= {C_1, …, C_k} AND MB={C'$_1$,…, C'$_{k'}$} AND T={L, G, M, E} AND N > 1.

We say that the exceptional association rule is selected by the type unexpectedness operator, in other words, that the association rule is conforming to the rule schema if:

$$\forall C_i \in M_A, conf(A, C_i) = TRUE$$
$$AND$$
$$\forall C'_{i'} \in M_B, conf(B, C'_{i'}) = TRUE$$
$$AND$$
$$T \neq E$$

Definition 4: Let the set of rules schema SR_1:<M_a→M_b> <T_1> <N_1>, ,… , SR_n:<$M_{a'}$→$M_{b'}$ <T_n><N_1> and a set of *K-Conforming, k-Objective and k-NOjective k-C(RS_1),…, k-O(RS_n),…, k-NO(RS_m)*, clusters generated by the synthesized process are T_1 ,…, T_n.

Definition 5: Let rules schema SR_1:<M_a→M_b> <T_1><N_1>… ,SR_n:<$M_{a'}$→$M_{b'}$> <T_n< N_1> and a set of *K-Conforming, k-Objective, k-NObjective* and type Unexpectedness, *k-C(RS_1),…, k-O(RS_n) ; U_t(RS_k),… k-NO(RS_m)*. All the three clusters are generated by the synthesized process.

4. Results and Discussion

An application was developed that implements the algorithm described above and allows the management of rule schemas. The application was tested on a real–life databases. The databases used were provided by Petroleum industry, in production fields. The organization of the company is multi-level, in our study we have limited only to three levels as shown in the Figure 7.

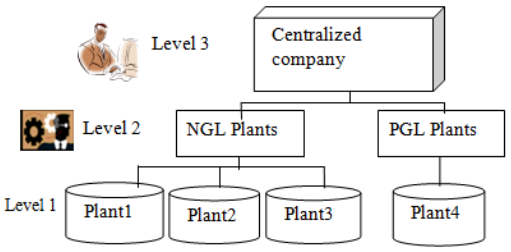


Figure 7. Structure of petroleum company.

The lower level consists of the operational databases represented by the plants represented by the local managers. The upper level is represented by the centralized head quarter manager Central Unit (CU). The middle level corresponds to the Propane Gas Liquefied (PGL) Plant and Natural Gas Liquefied (NGL) Plants represented by the manager by specialty PGL or NGL.

We explore the huge amount of production data in order to extract useful and knowledge to each data source that can be used for decision-making in order to optimize the production process. The database production is fueled by the daily data entered and validated by the relevant services in the plant unit. We are interested to the Lost Produce (LOP) data. The goal of this experiment is to optimize the efficiency of the petroleum installation and the equipments by reducing the LOP with analyzing the causes and the problems correspondent. The LOP is the difference of

the design of the plant and the difference of the real production and the production added. The design of the plant is the capacity of production for one year. In others way, LOP is the quantity that the plant can't produce because some triggers survey in a period. More formally, we can define the LOP in Equation 1 as follows:

$$LOP = capacity\ design - (Real\ production - product\ added) \quad (1)$$

Where:
Capacity design is the daily production of the plant *365 days.
Product added is the product quantity upper to the design.

Before the description the two studies that we have conducted, we define the databases and the ontology used and the rule schema structure. In the first study we demonstrate the efficiency of the CP_TREE for the local mining in time processing and space requirement. In the second study we demonstrate the efficiency of the transfer of the non frequent itemsets to the upper levels in the synthesizing process.

4.1. Databases Description

The four databases contain data between 10.000 and 30.000 records from 10 years. The structure of the database for each plant is: LOP (Unit, Day_hh_mm, train_Code, LOP_problem_code, class_code, LOP_Qty).
Where:
Unit: The plant;
Day_hh_mm: The date in hour and minutes;
Train_code: The code of the train;
LOP_problem_Code: The code of LOP problem;
Class_code: the classes code of the problem which can be (MC) Mechanical, (E) Electrical….
LOP_Qty: The quantity of LOP.

For building the transactional database, the domain expert must select the TID (Transactional identifier) field which can be the Train_code or Day_hh_mm and the attribut ITEMS which can be the LOP_Problem_code or Class_code. For this study we affect the TID by the Train_code and Day_hh_mm and the ITEMS by LOP_Problem_code attribute.

Table 1 shows an example of database of plant 1 (P_1). For example the first transaction describes that in 27/02/2000 and in the plant P_1 the quantity of LOP for the train T_5 is 529 because of the stop of the turbine (5002) which can be classed as a Process class problem (PR).

Table 1. Plant 1 database.

Unite	*Day_hh_mm*	*Train_code*	*LOP_problem_Code*	*Class_Code*	*LOP_Qty*
P_1	27/02/2000	T_5	5002	PR	529
P_1	28/02/2000	T_3	2010	SF	1585
P_1	28/02/2000	T_4	2011	SF	8870
…	…	…	…	…	…

In this study, the transactional database for the Plant1 is showed in the Table 2

Table 2. Plant 1 transactional database.

TID	ITEMS
27/02/2000	19001 19002
28/02/2000	19011 4007 5010
28/02/2000	10003 19011 19008
…	……..

For example in the Table 2, the first transaction describes that there are two problems occurred together which are the failure load for operating the train capacity and stop pumping.

4.2. Ontology Structure

Ontology is defined basically by two elements: a set of Concepts (C) hierarchized by the subsumption relation and a set of Relation (R) over concepts. We propose ontology composed of two mains parts, as shown in Figure 8. The first one is a database items organization with the root defined by the Attributes concept. The items are organized among the thematically structure of cause of LOP in the production databases. For instance, considering the Type concept: it regroups Technical, non Technical, ….Concepts.

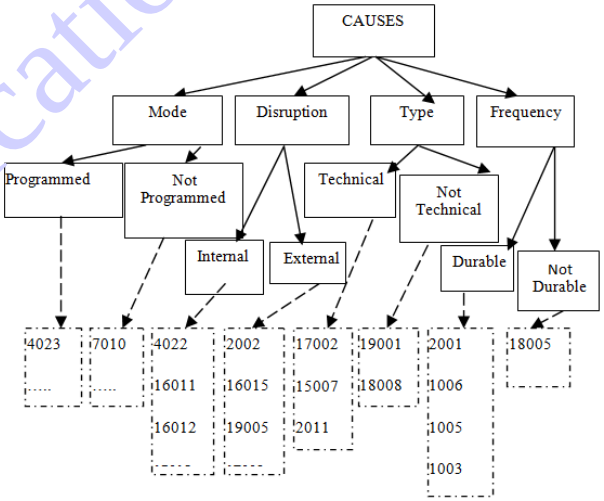


Figure 8. Ontology structure in OWL.

To describe the ontology we use the Web Semantic representation language, OWL-DL[1]. Based on description logics, OWL-DL language permits, along with the ontological structure, to create concepts using necessary and sufficient conditions over other concepts. Also, we use the software to edit the ontology.

4.3. Ontology Databases Mapping

Part of rule schema definition, ontology concepts are mapped to a/several items in the database. Thus, several ontology-database connection types can be conceived. Firstly, the simplest ontology-database

[1] http://www.w3.org/TR/owl-features

mapping is the direct one. It connects one leaf-concept of the *Attribute* hierarchy to a set of items.

Considering the concept C_1=technical of the ontology, it is associated to the attribute LOP_problem_code I_1=19001", I_2=18008, I_3=…. Furthermore, the concept C_1 is instantiated in the ontology by 2 instances describing the concept C_1 with 2 possible LOP_problem_code.

4.4. Rule Schemas

A rule schema allows user expectation representation and permit to the user to supervise association rule mining, meanwhile operators guide the intra-site and inter-site processing by filtering discovered rules.

The expert could use rule schemas for each level in order to compare the results and validate them. For that we choose three values of minimum support and minimum confidence in the following experiments.

- The first values is V_1 : $minsup_1$=1%, $minconf_1$=10%.
- The second is V_2 : $minsup_2$=5%, $minconf_2$=40%.
- And the third id V_3 : $minsup_3$=10%, $minconf_3$=60%.

Hence, the expert proposed a set of filtering rule schemas defined in Table 3. It presents the number of rules filtered by each rule schema and operator. We proceed with three examples or case studies, each one represent a set of rules schema and operators of different users at different levels. The first column represents the example number. The second one represents the rule schema number and the third represent the organization site and the fourth and the last one describe the rule schema and the operator.

Table 3. Operators and rule schemas for production database.

Example		Plant	Rule Schema	Operator
1	RS_1	Plant1 Plant2 Plant3 Plant4	<Défaut SONALGAZ> <L> <1>	1-C(RS_1)
	RS_2	NGL PGL	<Travaux sur filtre d'air d'admission turbine> <G> <2>	2-C(RS_2)
	RS_3	CU	<techniques > <M><3>	3-C(RS_3)
2	RS_4	Plant1 Plant2 Plant3 Plant4	<Fuite de vapeur→ durable> <L><1>	0-C(RS_4)
	RS_5	NGL	<Fuite de Gaz→Technique><E><2>	U_T(RS_5)
	RS_6	PGL	<défaillance du système ESD> <G><2>	1-C(RS_6)
	RS_7	CU	<Non Techniques→Techniques><E><3>	1-NO(RS_7)
3	RS_8	Plant1 Plant2	<Indisponibilité de chaudières→ Défaillance des réactances><G><1>	1-O(RS_{12})
	RS_9	Plant3 Plant4	< Savoir faire→ Problème d'instrumentation ><E> <2>	1-NO(RS_{13})
	RS_{10}	NGL	<Non Durable→ technique ><E><2>	U_T(RS_{14})
	RS_{11}	PGL	< Technique→Durable > <G><2>	1-CR(RS_{15})
	RS_{12}	CU	<Non Durable→ Durable ><G><3>	3-CR(RS_{16})

We give below interpretation of some rules schema and operator defined in Table 4.

- RS_3: This rule schema expresses that the headquarter company users are interested for the relation between the cause "*technique*" with others three (k=3) items but they are not sure over the side of the "*technique*" cause in the rule.
- RS_5: This rule schema expresses that decision makers of the NGL branch (level 2) are interested for the relation between the cause "*fuite de Gaz*" with the "*technique*" cause only for the majority and global rules. We can interpret this rule that a gas leak cause can induce a technical problem in the global and majority of the sites. So the decision maker can give importance to the "*fuite de Gaz*" cause like check the installation every days or week because this one can induce a serious technical problem which can take a lot of money to repair it.

4.5. Results

4.5.1. Study 1

In this study, we demonstrate the efficiency of the CP_TREE structure used in RSMLPA in time processing and the space requirement.

Table 4 report the time processing and the space requirement over the three values of minsup and minconf (V_1, V_2, V_3). The columns 3 and 4 define the time processing and the space requirement without using the CP_TREE structure. On the other side the columns 5 and 6 define the time processing and the space requirement with using the CP_TREE. Finally we calculate the ratio of the time processing and the space requirement in the two last columns. These two last columns present the gain of using the CP_TREE in time processing and space requirement.

Table 4. Time processing and space requirement with and without using the CP_TREE

Example		Without CP_TREE		With CP_TREE		Rate	
		Time (s)	Space (ko)	Time (s)	Space (ko)	Time %	Space %
1	V1	12	3	7	0,9	71,4	233,3
	V2	13	3	7	0,9	85,7	233,3
	V3	9	2,7	4	0,8	125	237,5
2	V1	10	1	5	0,9	100	11,1
	V2	10	1	5	0,9	100	11,1
	V3	8	0,9	3	0,8	166,7	12,5
3	V1	9	1	5	0,6	80	66,7
	V2	9	1	5	0,6	80	66,7
	V3	8	0,9	4	0,5	100	80

Figure 9 shows the evolution of the time processing over the three examples with varying the minsup and minconf values with and without using the CP_TREE structure. We can remark that for all values of minsup and minconf, the time processing using the CP_TREE structure has been improved.

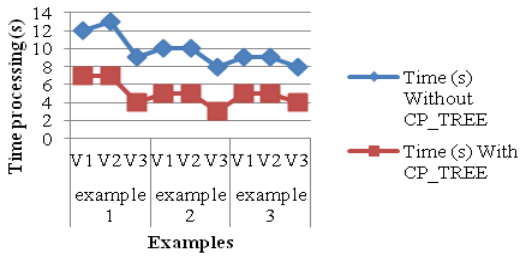


Figure 9. Time processing with and without using CP_TREE.

In example 2 with V_3, the time processing with using CP_TREE structure is improved about three times than without using the CP_TREE structure. We can also observe in example 1 with V_3 that the time processing with using CP_TREE structure is improved about two times than without using the CP_TREE structure.
We can interpret these results as follows:

- When the number of shared objects (Rules schemas, rules, item sets) is more important, which is the case in example 2 with V_3, the size of CP_TREE is small which facilitate the search space.
- When the number of shared objects (Rules schemas, rules, item sets) is less important, which is the case in example 1 with V_1, the size of CP_TREE is large which make the search space more important.
- The transfer of itemsets to the uppers level is performed with a simple projection on the CP_TREE structure over level. We don't need to search these itemsets which represent a gain in time processing.

Figure 10 shows that for all minsup and minconf values, we have a gain in the space requirement with using the CP_TREE structure. As shown in table 4 for the example 1, the space requirement with using CP_TREE structure can be improved about two times than without using the CP_TREE structure, we can interpret this result as follow:

- The condensed CP_TREE structure compresses the same objects (rules schemas, rules, item sets) in one node. Examples that have the same objects are regrouped in one object. So for an example with rules schemas with important shared objects (Example 1) it occupies less space than an example (Example 2) with rules schemas with less shared objects.

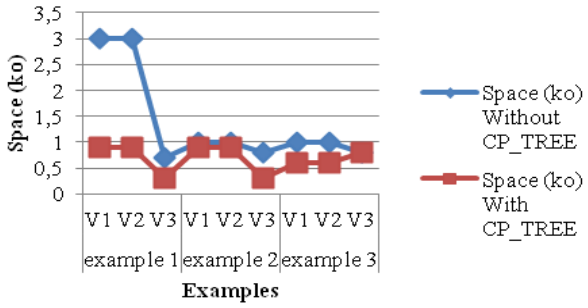


Figure 10. Space required with and without using CP_TREE.

4.5.2. Study 2

In this study, we demonstrate the efficiency of the transfer of the non frequent itemsets to the upper levels in the synthesizing process.

The first column of Table 5 expresses the number of the case study and second column contains the sites of levels 2 and 3. The third and fourth and fifth columns contain the number of rules extracted over the three values of the minsup and minconf. With using the traditional synthesizing local rules i.e., without transfer the none local frequent itemsets. On the other side the sixth and seventh and the eighth columns report the number of rules extracted over the three values of the minsup and minconf with transfer the none local frequent itemsets.

Table 5. The number of lost rules.

Examples		Without Transfer			With Transfer			Lost Rules		
		V_3	V_2	V_1	V_3	V_2	V_1	V_3	V_2	V_1
1	NGL	0	3	5	2	6	8	100%	50%	38%
	PGL	0	3	5	2	6	8	100%	50%	38%
	CU	0	7	11	4	10	15	100%	30%	27%
2	NGL	3	1	2	6	3	5	50%	67%	60%
	PGL	2	2	3	5	4	7	60%	50%	57%
	CU	3	4	7	7	7	14	57%	43%	50%
3	NGL	0	5	12	7	13	25	100%	54%	52%
	PGL	1	6	12	7	14	26	86%	57%	54%
	CU	0	9	18	3	20	37	100%	55%	51%

Figure 11 shows the number of lost rules with and without transfer the none frequent itemset. In about all examples with all values of minsup and minconf about 100% of rules are lost without using the transfer of none frequent itemset between the total number rules. These lost rules can help the decision-makers to make the right decision.

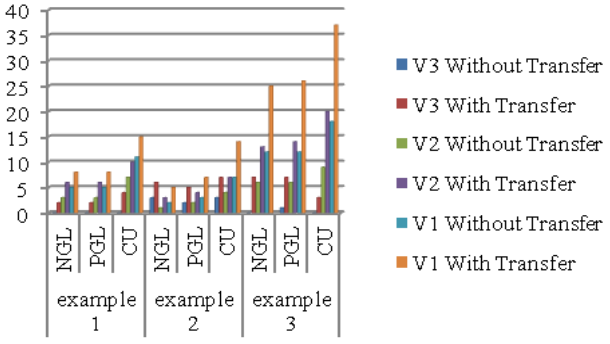


Figure 11. Number of rules with and without transfer.

5. Conclusions

This paper addresses the main issues: the integration of user knowledge in the multi-database mining process, without losing knowledge. In fact it discusses the problem of selecting interesting rules in the multiple large databases mining process of the interstate organization without lossless of patterns. The major contributions of our paper are stated below.

First, we propose a new formalism called Rule Schemas multi-level, extending the specification language proposed by [4] for user beliefs and expectations. Second, a set of operators, applicable over rule schemas, is proposed in order to guide the user throughout the mining process. Third we develop the local mining algorithm in order to reduce the space search and require only one scan to the database in order to extract only the interesting rules in the vicinity of what the user believes or expects. Finally for effective local mining we introduce the CP_TREE structure that contains only the candidate rules and itemsets which reduce significantly the time processing and the space requirement.

Therefore, the proposed model of mining global patterns of select items from multiple databases is efficient, since one does not need to estimate the patterns in multiple databases. The proposed algorithm was tested on a real-life example, which is the petroleum industry showing that the presented solution is valid and leads to good practical results.

Through the results of the experimentation we can say that RSMLPA algorithm extract exactly the same global rules as in the mono-database mining and we demonstrate the efficiency of the CP_TREE structure.

Future works will be directed towards experiments using others real-world and benchmarks databases for testing the efficiency of our approach.

References

[1] Andrei O., Claudia M and Fabrice G., "Local Mining of Association Rules with Rule Schemas". *IEEE* 2009.

[2] Jaroszewicz S., Simovici D., "Support approximations using Bonferroni-type inequalities". *In: Proceedings of Sixth European Conference on Principles of Data Mining and Knowledge Discovery*, Helsinki, Finland, pp. 212–223, 2002.

[3] Khiat S., Belbachir H and Rahal S.A., "Probabilistic Models for Local Patterns Analysis ". *In Journal of Information Systems Processing (JIPS)*. Vol. 10 N°.1 PP. 145-161 KIPS, 2014.

[4] Liu B., Wynne H., Ke W and Shu C., " Visually Aided Exploration of Interesting Association Rules". *Proceeding of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, Lecture Notes In computer science, Vol, 1574, Springer-Verlag, pages 26-28, 1999

[5] Pavlov D., Mannila H., Smyth P., "Probabilistics models for query approximation with large sparse binary data sets". *In: Proceedings of Sixteenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, pp. 465–472, 2000

[6] Rakesh A., Imielinski T., and Swami A., "Mining association rules between sets of large databases ". *ACM SIGMOD Record*, Vol.22, no.2, pp. 207-216, 1993.

[7] Ramkumar T., Srinivasan R., "Modified algorithms for synthesizing high-frequency rules from different data sources", *Knowl Inf Syst* 17:313–334, 2008 Springer

[8] Ramachandrarao P., Animesh A., Witold P., "Developing Multi-Database Mining applications " *Chapter 4, Advanced Information and Knowledge Processing*, Springer-Verlag London Limited, 2010.

[9] Senhadji S., Khiat S and Belbachir H.,"Association Rules Mining and Load Balancing Strategy in Grid Systems". *The International Arab Journal of Information Technology. Vol 11., N°4., July 2014.*

[10] Thirunavukkarasu R and Rengaramanujam S., " The Effect of Correction Factor in Synthesizing Global Rules in a Multi-databases Mining Scenario", *Journal of computer science*, no.6 (3)/2009, Suceava

[11] Unil Y., "Efficient mining of weighted interesting patterns with a strong weight and/or support affinity", *Information Sciences* 177 3477–3499 Elsevier 2007.

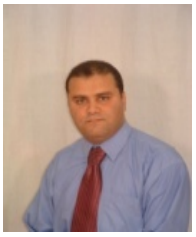
[12] Xindong W and Shichao Z., "Synthesizing High-Frequency Rules from Different Data Sources" *IEEE TransKnowledge Data Eng* 15(2):353–367 2003.

[13] Xindong W., Shichao Z., Chengqi Z., "Multi-Database Mining ", *IEEE Computational Intelligence* Bulletin Vol.2 No.1 2003.

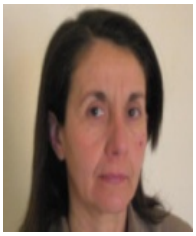
[14] Zhang C., Meiling L., Wenlong N., and Shichao Z., "Identifying Global Exceptional Patterns in Multi-database Mining", *IEEE Computational Intelligence Bulletin* February 2004 Vol.3 No.1.

[15] Zhang S., Chengqi Z., Jeffrey X., "An efficient strategy for mining exceptions in multi-databases ". Article in press, *An international journal information Science*, Elsevier, 2003

[16] Zhang S., Chengqi Z., Jeffrey X., "Identifying Interesting patterns in multi-databases ". Studies in Computational Intelligence (SCI) 4.91-112. Springer-Verlag Berlin Heidelberg 2005.



Salim Khiat He is Doctor in computer science since 2015 in University of science and technology–Mohamed Boudiaf Oran USTOMB Algeria. He teaches courses in undergraduate and graduate composition, at National School Polytechnic Oran Algeria.
He is memberships in Signal, System and Data Laboratory (LSSD). His current research interests include the databases, multi-database mining for software engineering, Ontology, grid and cloud computing.



Hafida Belbachir He Received PH.D degree in Computer Science from University of Oran, Algeria in 1990. Currently, she is a professor at the Science and Technology University USTO in Oran, where she heads the Database System Group in the LSSD Laboratory. Her research interests include Advanced Databases, DataMining and Data Grid.

Sid Rahal He is Doctor in computer science since 1989 in Pau University France.
He is memberships in professional activities are: - Member in LSSD (Laboratory Signal, System and Data) -Interest in Databases, Data Mining, Agent and expert systems.