P-TRIAR: Personalization Based on TRIadic Association Rules

Selmane Sid Ali, Omar Boussaid, and Fadila Bentayeb

Laboratoire ERIC, Université Lyon 2, Bron, France

Abstract. This article describes a new personalization process on decisional queries through a new approach of triadic association rules mining. This process uses the query log files of users and models them in new way by taking into account their triadic aspect. To validate our approach, we developed a personalization software prototype *P-TRIAR* (Personalization based on TRIadic Association Rules) which extracts two types of rules from query log files. The first one will serve to query recommendation by taking into account the collaborative aspect of users during their decisional analysis. The second type of rules will enrich user queries. The approach is tested on a real data warehouse to show the compactness of triadic association rules and the refined personalization which we propose.

1 Introduction

OLAP [1] systems users formulate decisional queries to meet their needs of specific analysis for decision support. OLAP tools are known to be intuitive as their end users are not necessarily computer scientists. However, the large volume of data and the complexity of analytical queries which involve a lot of aggregations make this task of analysis more difficult to users. So it seems necessary to provide them solutions best suited to their way of thinking through methods of recommendation and enrichment of their analytical queries. These methods are called personalization. In this paper, we propose a new personalization process of analytical queries. We are particularly interested in collaborative recommendation and enrichment of decisional queries based on log files.

The personalization works which exploit query log files use in most cases frequent itemsets [11] and association rules [20]. However, the large number of frequent itemsets and association rules obtained makes the task of personalization more difficult. Contrary to these approaches, the work we propose is based on another type of more compact rules called triadic association rules. These rules convey a richer semantic than conventional rules as they are formed in addition to the premise and the conclusion of a condition which enrich the rule. Our personalization process consists of five steps:

1. Modeling data of OLAP servers query log files by a triadic context. This triadic context will consist of the set of users, the set of queries, the set of

[1] *On-line Analytical Processing abr. OLAP.*

Y. Manolopoulos et al. (Eds.): ADBIS 2014, LNCS 8716, pp. 234–247, 2014.

attributes (descriptors and measures) in the SELECT clause and a ternary relation between these three sets.

2. The mapping of a triadic (tridimensional) context into a dyadic (bidimensionnel) one will be done by flattening the set of users over the set of attributes.
3. The computation of dyadic association rules *(premise* $\rightarrow$ *conclusion*).
4. The generation of triadic association rules $(premise \rightarrow conclusion)_{(condition)})$ through a *factorization* of dyadic ones.
5. The exploitation of these triadic association rules for personalization. To validate our approach we developed a personalization software prototype *P-TRIAR* (Personalization based on TRIadic Association Rules) to extract two types of rules from query log files. The first one will serve query recommendation by taking into account the collaborative aspect of users during their analysis. This recommendation will be carried out by the user communities discovered across multiple links between them. The second type of rules aims to enrich user queries by recommending attributes to add to his query.

The rest of the paper is organized as follows. Section 2 presents the modeling of log data with Formal Concept Analysis (FCA) and Triadic Concept Analysis (TCA) while recalling their basic concepts. Section 3 describes the proposed approach and algorithms for producing triadic association rules. Then in Section 4 we detail *P-TRIAR* and our process of personalization. Section 5 sheds light on works in personalization and association rules mining from multidimensional data. Experiments are performed in Section 6 to illustrate the compactness of triadic rules compared to dyadic rules and their contribution to personalization. Finally, some perspectives for future work are presented in Section 7.

2 Modeling Data Log Based on Formal Concept Analysis

In this section, we develop the data modeling process based on FCA. These data are implicitly collected from query log files of *OLAP* servers. We are interested in this work, especially on three data contained in a *SQL server* query log files[2], namely *MSOLAP_User* which identifies users, *Dataset* which contains the query and their attributes and *StartTime* which indicates the date and launch time of the query. This last is used to determine the date from which the log file could be exploited. These three data are easily accessible in data warehouses query log unlike data on user profiles, which are often hidden because of their private aspect. Our following definitions of a triadic context and its equivalent are based on those introduced in [13].

Definition 2.1. (Triadic context) a *triadic context* is a quadruplet of the form $\mathbb{K} := (R, U, A, Y)$ where:

– R, U, A respectively define que**R**ies, **U**sers and **A**ttributes (descriptors and measures) of the query SELECT clause.

[2] `http://technet.microsoft.com/en-US/library/cc917676.aspx`

Table 1. (a) Triadic context $\mathbb{K} := (R, U, A, Y)$, formed from $R = \{R_1, R_2, R_3, R_4, R_5\}$ (queries), $U = \{U_1, U_2, U_3, U_4\}$ (users) and $A = \{a_1, a_2, a_3, a_4, a_5\}$ (attributes). (b) Equivalent dyadic context $\mathbb{K}^{(1)}$ obtained from $\mathbb{K}$.

$\mathbb{K}$	U_1	U_2	U_3	U_4
R_1	$a_1a_2a_4$	$a_1a_2a_4a_5$	a_1a_3	a_1a_5
R_2	$a_1a_4a_5$	$a_2a_3a_4$	$a_1a_2a_4a_5$	a_4a_5
R_3	$a_1a_2a_4$	a_4a_5	a_1a_2	a_1a_5
R_4	$a_1a_2a_4a_5$	a_2a_4	a_1a_2	a_4a_5
R_5	$a_1a_4a_5$	$a_1a_4a_5$	$a_1a_2a_4a_5$	a_1a_5

(a)

$\mathbb{K}^{(1)}$	U_1					U_2					U_3					U_4				
	a_1	a_2	a_3	a_4	a_5	a_1	a_2	a_3	a_4	a_5	a_1	a_2	a_3	a_4	a_5	a_1	a_2	a_3	a_4	a_5
R_1	1	1		1		1	1		1	1	1		1			1				1
R_2	1			1	1		1	1	1		1	1		1	1				1	1
R_3	1	1		1					1	1	1	1				1				1
R_4	1	1		1	1		1		1		1	1							1	1
R_5	1			1	1	1			1	1	1	1		1	1	1				1

(b)

– $Y \subseteq R \times U \times A$ represents a ternary relation where each $y \subseteq Y$ represents a triple: $y = \{(r, u, a) | r \in R, u \in U, a \in A\}$. In other words, query $\boldsymbol{q}$ is launched by user $\boldsymbol{u}$ and which involves the attribute $\boldsymbol{a}$.

We illustrate through an example (Table 1.a), the transition from log data to a triadic context. Each user in U =(U_1, U_2,..., U_4) performs analysis by launching a sequence of queries denoted R =(R_1, R_2,..., R_5) where each query is composed of a set A =(a_1, a_2,...,a_5) of attributes from different facts and dimensions of the data warehouse.

For example, the value $a_1a_2a_4$ located at the intersection of the first column and the first row means that the user U_1 launched the query R_1 composed of the attributes a_1, a_2 and a_4.

Definition 2.2. (Dyadic context) In Formal Concept Analysis a *dyadic formal context* is a triplet
$\mathbb{K}^{(1)} := (G, M, I)$ where G is a set of objects, M a set of proprieties and I a binary relation between G and M. Our equivalent dyadic context is formed by flattening of the triadic context we defined (see Definition 2.1). The objects in G are the queries in R and proprieties in M are pairs (a_j, a_k) in the projection of set users into set of attributes $U \times A$. The table 1.b represents the dyadic context $\mathbb{K}^{(1)}$ obtained from the triadic context $\mathbb{K}$, thus:
$\mathbb{K}^{(1)} := (R, U \times A, Y^{(1)})$ with ($(a_i, (a_j, a_k)) \in Y^{(1)} \iff (a_i, a_j, a_k) \in Y$). The value 1 of the first row and the first column means that the user U_1 launches the query R_1 which implies the attribute a_1. In what follows, the pair $(a_j, a_k) \in U \times A$ will be noted in a simplified manner by a_j-a_k.

Definition 2.3: (Derivation) For $X \subseteq G$ and $Z \subseteq M$, two subsets $X' \subseteq M$ and $Z' \subseteq G$ are respectively defined as a set of proprieties common to the objects in X and a set of proprieties which share all attributes in Z. Formally, the derivation denoted $'$ is defined as follows:

$X' := \{a \in M \mid o\mathrm{I}a \ \forall o \in X\}$ and $Z' := \{o \in G \mid o\mathrm{I}a \ \forall a \in Z\}$.

This proposal defines a pair of correspondence $(',')$ between the set of parts of G and the set of parts of M representing a Galois correspondence. The closure operators in G and M are denoted by $''$. For example, the closure of $U_2 - a_4$ is given by:

$(U_2 - a_4)'' = ((U_2 - a_4)')' = \{R_1, R_2, R_3, R_4, R_5\}' = \{U_1 - a_1, U_1 - a_4, U_2 - a_4, U_3 - a_1, U_4 - a_5\}$.

Definition 2.4: (formal concept) A formal concept (cf) is a pair (X, Z) with $X \subseteq G$, $Z \subseteq M$, $X = Z'$ and $Z = X'$. The set X is called *extension* of cf while Z is its *intention*. A formal concept (dyadic) corresponds to a maximum rectangle in a dyadic context.

Example. As $\{R_1, R_2, R_3, R_4, R_5\}' = \{U_1 - a_1, U_1 - a_4, U_2 - a_4, U_3 - a_1, U_4 - a_5\}$ and $\{U_1 - a_1, U_1 - a_4, U_2 - a_4, U_3 - a_1, U_4 - a_5\}' = \{R_1, R_2, R_3, R_4, R_5\}$, then the pair($\{R_1, R_2, R_3, R_4, R_5\}$,$\{U_1 - a_1, U_1 - a_4, U_2 - a_4, U_3 - a_1, U_4 - a_5\}$) form a formal concept.

Definition 2.5: (Dyadic association rule) Let (G, M, I) a formal dyadic context. An association rule (R) [2] has the following format $R : B \rightarrow C$ (s, c) where $B, C \subseteq M$ with $B \cap C = \emptyset$. The support s of a rule R is calculated by the formula: $Supp(R) = \frac{|B' \cap C'|}{|G|}$. The confidence c is given by: $Conf(R) = \frac{|B' \cap C'|}{|B'|}$. We speak of implication when the confidence of the association rule is equal to 1.

In the following section, we will show how to exploit the dyadic association rules for generating triadic ones. The dyadic rules are produced from our context ($\mathbb{K}^{(1)}$) by applying *Pasquier* algorithms [16].

3 Triadic Association Rules Extraction

3.1 Definitions

It is apparent from the literature study so far that [4] is the first to study the implications extraction problem in triadic contexts. A *triadic implication* has the following form: $(A \rightarrow D)_C$. This implication is true if "*whenever A is true under all conditions in C, then D is also true under all conditions*". Afterwards, [7] have extended the work of *Biedermann* and defined three types of implications: attribute - condition implications, conditional attribute implications, attributional condition implications. [14] extended these definitions to association rules and proposed three types: *Attributes-Conditions Association Rules* (**A-CAR**s); *Conditional Attribute Association Rules* (**CAAR**s); *Attributional Condition Association Rules* (**ACAR**s). In what follows, we consider our example (Table 1) of a triadic context $\mathbb{K} := (R, U, A, Y)$ and its equivalent dyadic one $\mathbb{K}^{(1)} := (R, U \times A, Y^{(1)})$ to define the different types of association rules.

Definition 3.1.1: An *Attribute-Condition Association Rule* (**A-CAR**) is a dyadic association rule in the form $A \rightarrow D$ (s, c), where A and D are subsets of $U \times A$, s and c represent respectively the support and confidence. These dyadic association rules are extracted from the dyadic context $\mathbb{K}^{(1)}$.

Example: $U_2 - a_1 \rightarrow U_2 - a_5, U_2 - a_4, U_3 - a_1, U_1 - a_1, U_1 - a_4, U_4 - a_1, U_4 - a_5$ $(0.4, 1)$ is an *A-CAR* with support equal to 40% and a confidence equal to 100%.

Definition 3.1.2. A *Conditional Attribute Association Rules* according to *Biedermann* formalism (**BCAAR**) is a triadic association rule with the following notation: $(A \rightarrow D)_C\ (s, c)$, where A and D are subsets of U, and C a subset of A and means that A *implies* D *under all conditions in* C *with a support* s *and a confidence* c.

Example. the rule $(U_2 \rightarrow U_1)_{a_1 a_2}(0.2, 1)$ is a $BCAAR$ with a support 20% and a confidence 100%.

Definition 3.1.3. An *Attributional Condition Association Rules* according to *Biedermann* formalism (**BACAR**) is a triadic association rule the following notation $(A \rightarrow D)_C\ (s, c)$, where A and D are subsets of A, and C are subsets of U and means that A *implies* D *under all conditions in* C *with a support* s *and a confidence* c .

Example. the rule $(a_2 \rightarrow a_4)_{U_2 U_1}(0.4, 1)$ is a $BACAR$ with a support 40% and a confidence 100%.

These two types of triadic association rules (i.e., $BCAAR$ and $BACAR$) have the same notation but the sets of their premises, conclusions and conditions differ.

3.2 Proposed Approach

In what follows, we present our approach based on formal definitions and illustrative examples. Several approaches for researching and analysing triadic concepts have emerged in the literature [19], [15] and [5] for $n = 3$. [14], propose an effective approach based on the triadic context analysis for the extraction of triadic association rules. It consists of taking as input a formal triadic context which is flattened to produce a dyadic one. Then dyadic concepts and dyadic generators are extracted. After that, triadic concepts are then generated from dyadic concepts and triadic generators from dyadic ones. Once these two sets gathered, it is then possible to extract the triadic association rules. These operations give good results but can be avoided by our alternative which does not calculate these two sets. Our approach is based on the same theoretical basis that the one proposed by [14]. Nevertheless, our extraction process is different in terms of input data and our algorithms are applied rather on a set of dyadic association rules of type **RAA-C**. To extract these **RAA-C** we apply the algorithms of [16] on the dyadic context $\mathbb{K}^{(1)} := (R, U \times A, Y^{(1)})$ obtained from the projection of the set of properties U on the set of conditions A of the formal triadic context $\mathbb{K} := (R, U, A, Y)$. Then, from these latter and the definitions recalled in section 3.1.1, we apply the algorithms which we have proposed to search for triadic association rules in their various forms: *Biedermann* Conditional Attribute Association Rules ($BCAAR$) and Biedermann Attributional Condition Association Rules ($BACAR$).

3.3 Proposed Algorithms

The transition from dyadic association rules to the set of all triadic one, in their various forms, is performed using a main procedure called *TRIAR*. It permit to produce the triadic association rule through the two sub procedures $BCAAR$ and

BACAR. This choice of decomposition is motivated by the parallelization of these two procedures during implementation to have two types of personalization.

The main procedure *TRIAR* (Algorithm 1) consists of three parts. The first one (lines 4-8) corresponds to a sorting procedure which identifies whether a dyadic association rule is eligible to become a triadic one or not. This step is justified by the mapping of the definition of triadic generator in [14] to a triadic rule. So as we collect the distinct values of attributes on the set A_L (line 6) of the rule premise LHS, the distinct values of conditions within the set M_L (line 7) of the rule premise LHS; and we check if their product corresponds to the size of LHS (line 8). The other two parts (lines 9 and 10) correspond to the procedures *BCAAR* and *BACAR* (Algorithms 2 and 3) which allow us to produce the set of triadic association rules.

Algorithm 1. Computation of Triadic Association Rule

1: **Procedure** TRIAR(D)
2: **In:** $D = \{(LHS, RHS, s, c)\}$
3: **Out:** $\Sigma = \{(L, R, C, t, s, c)\}$
4: $\Sigma \leftarrow \emptyset$;
5: **for** $RL = (LHS, RHS, s, c)$ in D **do**
6: $\quad A_L \leftarrow$DISTINCT A(LHS)
7: $\quad M_L \leftarrow$DISTINCT M(LHS)
8: $\quad$ **if** $Size(A_L) \times Size(M_L) = Size(LHS)$ **then**
9: $\qquad \Sigma \leftarrow \Sigma \cup \{(\mathrm{BCAARS}(A_L, M_L, RHS), 1, s, c)\}$
10: $\qquad \Sigma \leftarrow \Sigma \cup \{(\mathrm{BACARS}(A_L, M_L, RHS), 2, s, c)\}$
11: **out** Σ

We have as input *TRIAR* a set (D) of dyadic association rules **(RAA-C)** where each rule has the following form (LHS, RHS, s, c) representing respectively (the premise, the conclusion, the support and the confidence of the rule).

Example. the rule $U_3 - a_4, U_4 - a_4 \rightarrow U_2 - a_3, U_2 - a_2, U_2 - a_4, U_3 - a_1, U_3 - a_5, U_3 - a_2, U_1 - a_1, U_1 - a_5, U_1 - a_4, U_4 - a_5$(sup = 0.20; conf = 1.00) is written as follows ($\{U_3 - a_4, U_4 - a_4\}$, $\{U_2 - a_3, U_2 - a_2, U_2 - a_4, U_3 - a_1, U_3 - a_5, U_3 - a_2, U_1 - a_1, U_1 - a_5, U_1 - a_4, U_4 - a_5\}$, 0.20,1).

The output of the procedure *TRIAR*, we have a set of triadic association rule (Σ), where each rule is presented in the following form (L, R, C, t, s, c), representing the premise, the conclusion, the condition, the type respectively (1: *BCAAR*; 2: *BACAR*), the support and the confidence.

Example. the *BCAAR* ($U_3U_4 \xrightarrow{a_4} U_2U_1$ (sup = 0.20; conf = 1.0)) is written as follows $(U_3U_4, U_2U_1, a_4, 1,\ 0.20, 1.0)$.

To expand *TRIAR* algorithm, we take as an example the dyadic rule $(\{U_3 - a_4, U_4 - a_4\}, \{U_2 - a_3, U_2 - a_2, U_2 - a_4, U_3 - a_1, U_3 - a_5, U_3 - a_2, U_1 - a_1, U_1 - a_5, U_1 - a_4, U_4 - a_5\}, 0.20, 1)$. Lines 5-7 of Algorithm 1, we create two sets A_L and M_L which respectively contain the distinct attributes and distinct conditions of the premise of the rule LHS $\{U_3 - a_4, U_4 - a_4\}$. Accordingly, $A_L = \{U_3, U_4\}$, $M_L = \{a_4\}$. This entails that the product $Size(A_L) \times Size(M_L) = 2$ (line 8) is equal to $Size(LHS)$, as the portion M_L will become a condition for the constructed rules. All the elements of A_L must verify this condition thus this rule is eligible to become a triadic association rule. Lines 9 and 10 of Algorithm 1 involve both procedures $BCAAR$ and $BACAR$ to produce both types of triadic rules.

The procedure $BCAARs$ (algorithm 2) takes as input three sets A_L, M_L and RHS. The set M_L represents the conditions which apply to all attributes in the set A_L and we want to find in RHS other attributes which are affected by the same conditions, from where the search of the conditions 5-7 lines. These attributes are isolated within line 9 (*group by* on attributes), to see whether their conditions meet the conditions of M_L (line 11), if they are identical to those of M_L we can build a rule.

Algorithm 2. Computation of $BCAAR$ (type =1)

1: **Procedure** BCAARS(A_L, M_L, RHS)
2: **In:** A_L, M_L, RHS
3: **Out:** *BCAAR* $= (A_L, A_R, M_L)$
4: $A_R \leftarrow \emptyset$; $Temp \leftarrow \emptyset$
5: **for** $e \in RHS$ **do**
6:    **if** MODUS$(e) \in M_L$ **then**
7:       $Temp \leftarrow Temp \cup \{e\}$
8: **if** $Temp \neq \emptyset$ **then**
9:    Creates containers $B = b_1, ..., b_n$ by grouping elements of $Temp$ having the same part of attributes in common
10: **for** $elem \in B$ **do**
11:    **if** $Size(elem) = Size(M_L)$ **then**
12:       $A_R \leftarrow A_R \cup \{Attr(elem)\}$
13: **if** $A_R \neq \emptyset$ **then**
14:    **out** (A_L, A_R, M_L)

The sequence of the algorithm is performed as follows: after the initialization of the parameters (lines 2-4), we take the conclusion of the rule RHS (line 5) which corresponds to $\{U_2 - a_3, U_2 - a_2, U_2 - a_4, U_3 - a_1, U_3 - a_5, U_3 - a_2, U_1 - a_1, U_1 - a_5, U_1 - a_4, U_4 - a_5\}$ in the example, and calculate the $Modus$ of each element which corresponds the condition. For the first component, $Modus(U_2 - a_3)$=$\{a_3\}$ The test shows that it is not included in the set $M_L = \{a_4\}$ the condition is not satisfied, the loop move to the next item. For the fourth element, $Modus(U_2 - a_4)$=$\{a_4\}$ it is included in M_L the condition is satisfied. The variable *Temp* gets this item $(U_2 - a_4)$, then in line 9, we group in a container denoted B the elements which have the same part attribute, in our example $(U_2 - a_4)$, $(U_1 - a_4)$ will be contained in (B). The algorithm 2 checks in line 10-12, for each element contained in (B), if the size of the element is equal to the size of M_L. In our example, these two entities are equal for the two elements because they have a size equal to 1. The rule formed of triplet (A_L, A_R, M_L) is then constituted $(\{U_3, U_4\}, \{U_2, U_1\}, a_4)$ to which it is added type, support and confidence. The result is: $BCAAR$ $(U_3U_4 \rightarrow U_2U_1)_{a_4}$, type = 1, Sup = 0.20 and Conf= 1.00. This is the exit point of the algorithm 2 and the rule is added to the set of $BCAAR$.

In the procedure $BACARs$ (algorithm 3), we input three sets A_L, M_L and RHS. The set A_L represents the attributes which apply to all conditions in the set M_L and we want to find in RHS other attributes which are affected by the same conditions, from where the search of the conditions 5-7 lines. These attributes will be isolated from line 9 (*group by* on conditions), to permit viewing if their attributes meet the attributes of M_L (line 11), if the attributes are identical to those of M_L we can build a rule. The others steps of the algorithm 3 are unrolled in the same way of algorithm 2.

Algorithm 3. Computation of $BACAR$ (type = 2)

1: **Procedure** BACARs(A_L, M_L, RHS)
2: **In:** A_L, M_L, RHS
3: **Out:** *BACAR* $= (M_L, M_R, A_L)$
4: $M_R \leftarrow \emptyset$; $Temp \leftarrow \emptyset$
5: **for** $e \in RHS$ **do**
6:     **if** ATTRIB(e) $\in A_L$ **then**
7:         $Temp \leftarrow Temp \cup \{e\}$
8: **if** $Temp \neq \emptyset$ **then**
9:     Creates containers $B = b_1, ..., b_n$ by grouping elements of $Temp$ having the same part of attributes in common
10: **for** $elem \in B$ **do**
11:     **if** $Size(elem) = Size(A_L)$ **then**
12:         $M_R \leftarrow M_R \cup \{Cond(elem)\}$
13: **if** $M_R \neq \emptyset$ **then**
14:     **out** (M_L, M_R, A_L)

3.4 Complexity Study

In what follows, we present the study of the complexity of our main algorithm *TRIAR*. It uses the procedures *BCAARs* and *BACARs*. It takes as input D a set of dyadic association rules. Latter is obtained from a dyadic formal context $\mathbb{K}^{(1)} := (R, U \times A, Y)$. The maximum size of dyadic association rule is given by $|U| * |A|$. The overall complexity of the algorithm is linear in $|D|$ and is performed in $O(|D| * 2(|U| + |A|))$. This complexity is obtained by studying the loop "for" (line 5), which iterates through one time all the rules in D, it is given by: line 5 is performed in $O(|U|)$ because at worst, we have rules in all context properties, line 8 is performed in $O(|A|)$ because in the worst case we can have a rule in all properties of the context. The test in line 8 is performed $O(|D|)$ because this is the set of rules which is driven to test their eligibility to become triadic rules; Instructions 9 and 10, respectively, call the procedure *BCAARs* and *BACARs*. Such appeals are made in the worst case $O(|D| * |U| + |A|)$, where all dyadic association rules are eligible to become triadic ones.

4 Architecture of *P-TRIAR*

P-TRIAR involves five steps (see Figure 1). In Section 2, we described the first three stages, namely: Modeling a triadic context data from query log of *OLAP* analysis

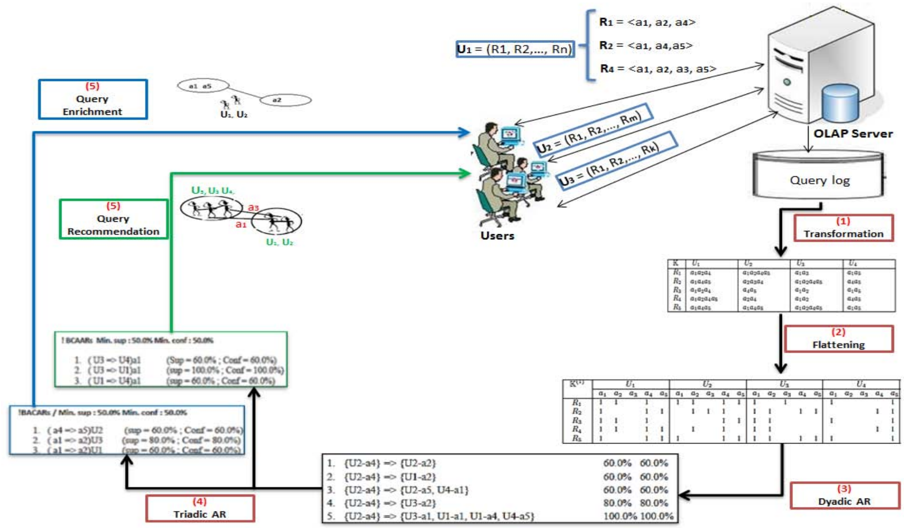

Fig. 1. Architecture of P-TRIAR

server; the transition of this triadic context to a dyadic one and finally the production of conventional dyadic association rules type *premise* $\rightarrow$ *conclusion*. Then, in section 3, we detailed the approach we propose to generate a set of triadic association rules type $(premise \rightarrow conclusion)_{(condition)}$ by *factorization* of dyadic rules. In what follows, we describe the fifth stage of *P-TRIAR* regard to the exploitation of triadic association rules (*BCAAR* and *BACAR*) obtained by our algorithms.

4.1 Query Recommendation by *BCAAR*

The *BCAAR* determine the associations which exist between users that have as a condition attributes. In other words, this type of rules allows us to discover the relationship between users through the attributes involved in their queries. For example, the *BCAAR* $(U_4 \rightarrow U_3U_1)_{a_1}$ (0.60,1) states that whenever a query is submitted by the user U_4 and contains the attribute a_1, the users U_3 and U_1 submit a query which contains the same attribute, with a support 60% and a confidence 100%. This rule highlights the similarity between user U_4 and users U_3 and U_1 but on condition to query the attribute a_1. Through this rule, we find the collaborative aspect because it allows forming a community link between three users. This community connection is conditioned by the involvement of the attribute a_1 in their queries and the degree of this link has a specific support and confidence.

The first personalization scenario, the user connects to *P-TRIAR* defines the initial parameters (the date from which he wants to explore the log, the threshold of support and confidence) and wants to know the links that he entertains with other users. *P-TRIAR* shows him the rules which satisfy these parameters. Assuming that U_4 choose the rule of our example, *P-TRIAR* will recommend a number of decisional queries that U_4 desires. These queries will be filtered and sorted: by frequencies, by users (U_3 and U_1) and by attributes (a_1). So as the user may choose the queries which are suitable for its analysis needs. If the user wants to directly access to queries, *P-TRIAR* recommend him a set of queries without having to choose among *BCAAR*, *P-TRIAR* detects which user is logged on and it offers a number of queries filtered by number of users and number of attributes, i.e., based on rules which have the largest number of users in the conclusion part rule and the largest number of attributes in the condition part.

4.2 Query Enrichment by *BACAR*

The *BACAR* determine the associations which exist between attributes which have users as a condition. This type of rules allows us to discover the relationships between attributes (descriptors and measures) involved in a query through users making it. For example the, *BACAR* $(a_2 \to a_4)_{U_2 U_1}(0.4, 1)$ is true when each time a request is submitted and which involves the attribute a_2, the attribute a_4 is also involved in the query on condition that users U_2 and U_1 formulate it.

The second scenario of personalization is based on *BACAR*. In this scenario, the user sets the same parameters of the first scenario and wishes to make a request for analysis taking inspiration the links which exist between the attributes of the warehouse. Assuming the user U_2 is connected and chooses the *BACAR* $(a_2 \to a_4)_{U_2 U_1}$ which means that each time a query is submitted and which contains the attribute a_2, the attribute a_4 is also involved in the query as long as users who formulates it are U_2 and/or U_1, with a support 40% and a confidence 100%. This rule highlights the similarity existing between the attributes a_2 and a_4 but under the condition that the users U_1 or U_2 formulate the query. *P-TRIAR* relies upon such rule to enrich the user U_2 query by recommendation of the attribute a_4 as element of its query.

5 Related Works

The personalization of queries has been the subject of several studies [3], [1], [12], [17]. It aims to help the user generally based on its behavior and its previous queries or those of other users. In the areas of databases and data warehouses the different personalization techniques have been classified into three categories [1], [17]: collaborative techniques [6]and [9] which exploits the similarity between users profiles and one for which the recommendation is determined; based on the content techniques [10] intended to recommend to a user attributes that frequently seeks; and finally hybrid techniques [18] which combine the two previous techniques. In literature, the recommender systems have as sources user data profiles, log files which are structured historization of queries for each user, or external sources such as ontologies, web pages, etc..

Several studies have exploited the idea of pattern extraction [11] and association rules [20], from log files, for the recommendation. However, their work was limited to a bi-dimensional framework. They represent the data log files across matching matrices (*users*× *query*) or (*attributes*× *query*) for association rules or patterns extraction. This modeling does not take into account the three-dimensionality of these data. In data warehouses, the association rules and the patterns they get are numerous and of dyadic type. This very large number of patterns and association rules makes more complicated the recommendation task and does not take into account at the same time the three sets of *users, attributes* and *queries.*

In addition, FCA [22], [8] and Galois lattices constitute a theoretical basis for solving many problems in the fields of artificial intelligence, software engineering and databases. The TCA was originally introduced by [23] and [13]. Their work focuses on the analysis of triadic contexts, concepts and lattices concepts called trilattices. They define the way, the theoretical basis for ATC. In this way, they defined the theoretical basis for the TCA. [4] provides a writing formalism of triadic implications and [21] defines polyadic concepts analysis and generalizes the work of [23] to polyadic formal contexts to produce polyadic formal concepts and n-lattice.

More recent work related to the TCA exist, [7] consider different types of triadic implications which he calls strong relying formalism stated by [4]. [15] propose an approach for mining rules applied to dynamic relational graphs which can be encoded in n-ary relationships ($n \geq 3$). The work of [14] offer not only an approach to triadic association rules production but also procedures to extract triadic concepts and generators from dyadic ones.

In [19], the authors deal with the calculation of generators and triadic association rules. However, the author provided a new definition of the latter which is different from that of [14] which, in turn, is based on the definition of *Biedermann.* In [15] and [5], the authors propose the generalization of the concept of association rules in a multidimensional context by working either on boolean matrices but on boolean tensors of arbitrary arity. They also provide measures of frequency and confidence to define the semantics of such rules.

Based on the literature review we conducted, our work is the first to model the log data through a triadic context. The proposed approach provides a personalization from triadic association rules. We show through our process, how to get triadic association rules from these triadic contexts, using only the dyadic association rules without calculating the triadic concepts and generators as proposed by the authors mentioned above.

6 Experiments

The tests we performed on the warehouse *PUBS* [3](Figure 2) focused on five users and 100 decisional queries, by user, composed of 34 distinct attributes that contains *PUBS.* It concerns the analysis of the turnover (CA) and quantity (Qty) of books sold. These measures are observed over the following dimensions: *Titles, Publishers, Stores, Times* and *Authors.*

Five users (U_1, U_2, U_3, U_4, U_5) logged on *PUBS* and submitted their different sequences of decisional queries denoted (R_1,..., R_n). Each query is formed in the SELECT clause of attributes (descriptors and measures) noted (a_1, ...,a_n).

Example. User U_4 launches a set of queries on the warehouse *PUBS*:

[3] Data warehouse constructed from the database *PUBS* provided by *Microsoft*: `http://technet.microsoft.com/fr-fr/library/ms143221(v=sql.105).aspx`

- R_1 = What is the turnover of the store *store_400* for the year *2013*. *PUBS* attributes involved in the SELECT clause of R_1 are (CA, *Stores.stor_id, Times.year*).
- R_2= Turnover realized on sales of books type *Computer Science* sold at stores in *Paris* during *2013*. (CA, *Titles.type, Stores.stor_id, Times.year*).
- R_3= the number of books written by *Parisian authors* and published by *Springer* in *2013*. (*Qty, Authors.city, Publishers.pub_name, Times.year*).

In this way, other users formulate other sequences of analytical queries which involve attributes already expressed in U_4 queries. Let us take for example, the user U_5 query:

- R_1 = What is the turnover (CA) of the store store_500 in Washington by month. R_1 (CA, *Stores.stor_id, Stores.city, Times.month*).

An example of triadic association rules extracted from of users U_4 and U_5 are:

- *BCAAR*: $(U_4 \longrightarrow U_5)_{(CA, Stores.stor\_id)}$ supp= 60%, conf= 80%.
- *BACAR*: $(CA \longrightarrow Stores.stor\_id)_{(U_4, U_5)}$ supp= 75%, conf= 100%.

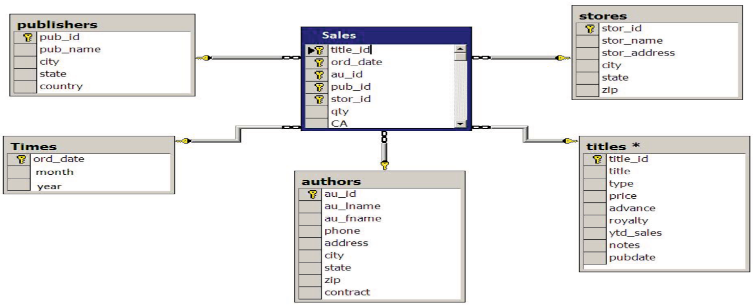

Fig. 2. PUBS data warehouse

In this paper, we propose a personalization based on these two types of rules. The user will interact with the interface *P-TRIAR* in two scenarios. According to the first scenario described in 4.1, the user U_4 wants to make new analysis on the data warehouse. He asks *P-TRIAR* and interrogates the log file from a specific date and requests all triadic association rules between him and other users with a condition on attributes, before setting a minimum threshold for the support and confidence. *P-TRIAR* will return him all the rules which satisfy these parameters. Then U_4 will choose according to the attributes he wants query the rules which suit him. Assuming he chooses the rule $(U_4 \longrightarrow U_5)_{(CA, Stores.stor\_id)}$, *P-TRIAR* will recommend him analysis queries the most frequent made by the user U_5 and having among their attributes CA and *Stores.stor_id*. Unlike the dyadic rule $(U_4 \longrightarrow U_5)$ which would recommend all queries made by U_5, we add a condition on query attributes, so the rule is enriched and the number of queries to recommend is reduced considerably. So U_4 could choose from these queries which suits his analysis or by modifying it in part.

According to the second scenario described in 4.2, U_4 wants to make a new query on the attributes of the warehouse by exploiting $BACAR$ (rules between query attributes which have as a condition users). U_4 sets the initial parameters such as date, minimum support and confidence. Then U_4 will choose attributes he wants to involve in its query and *P-TRIAR* will propose him triadic association rules associated with them. Assuming he chooses CA attribute *P-TRIAR* would recommend him the attribute *Stores.stor_id* based on the rule $(CA \longrightarrow Stores.stor\_id)_{(U_4,U_5)}$. Contrary to the rule $(CA \longrightarrow Stores.stor\_id)$ will be proposed to all users, this rule will only be recommended to U_4 and U_5.

We obtained with a threshold of support and minimum confidence 50%, a total of 123 $BCAAR$ and 95 $BACAR$ from 42,638 AR dyadic. This result shows the triadic association rules compactness compared to dyadic ones. Then for personalisation, we take the example of user U_3, we obtain 14 $BCAAR$ and 12 $BACAR$ which would recommend queries and enrich its own ones according to his choices of analysis.

7 Conclusion

In this article, we described a new personalization process, particularly collaborative recommendation and query enrichment, based on the query log files of users. We have, at first, modelled data from *log* files with formal concept analysis to build triadic contexts. Then, we proposed a new alternative which exploits ideas from the triadic concept analysis to generate triadic association rules from triadic contexts, and produce them by exploiting only dyadic association rules without having to manipulate concepts and triadic generators which are unnecessary in our process. Through the proposed approach, we have shown how to obtain triadic association rule *(BCAAR and BACAR)* less numerous and more compact than dyadic rules, while also conveying a richer semantics. We validated our personalization approach by developing *P-TRIAR* to extract these two types of rules from log files and personalize user queries according to each type.

This work opens up many opportunities for research. We plan in the short term to provide a system which collects user preferences through their choice of different personalization rules and queries recommended. Thus, they would be taken into account in their future choices. In the medium term, we plan to generalize the algorithms offered to polyadic association rules to deal with n-ary relationships to propose new methods for community detection in heterogeneous social networks.

Acknowledgement. We thank Rokia Missaoui for her collaboration in the work concerning the extraction of triadic associaton rules.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)

2. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: Proc. 20th Int. Conf. Very Large Data Bases, VLDB, vol. 1215, pp. 487–499 (1994)
3. Bellatreche, L., Giacometti, A., Marcel, P., Mouloudi, H., Laurent, D.: A personalization framework for olap queries. In: DOLAP, pp. 9–18 (2005)
4. Biedermann, K.: How triadic diagrams represent conceptual structures. In: ICCS, pp. 304–317 (1997)
5. Cerf, L., Besson, J., Nguyen, T.K.N., Boulicaut, J.-F.: Closed and Noise-Tolerant Patterrns in N-ary Relations. Data Mining and Knowledge Discovery 26(3), 574–619 (2013)
6. Chatzopoulou, G., Eirinaki, M., Polyzotis, N.: Query recommendations for interactive database exploration. In: Winslett, M. (ed.) SSDBM 2009. LNCS, vol. 5566, pp. 3–18. Springer, Heidelberg (2009)
7. Ganter, B., Obiedkov, S.A.: Implications in triadic formal contexts. In: ICCS, pp. 186–195 (2004)
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag New York, Inc. (1999), Franzke, C. (trans.)
9. Golfarelli, M., Rizzi, S., Biondi, P.: myolap: An approach to express and evaluate olap preferences. IEEE Trans. Knowl. Data Eng. 23(7), 1050–1064 (2011)
10. Khemiri, R., Bentayeb, F.: Interactive query recommendation assistant. In: 2012 23rd International Workshop on Database and Expert Systems Applications (DEXA), pp. 93–97. IEEE (2012)
11. Khemiri, R., Bentayeb, F.: Fimioqr: Frequent itemsets mining for interactive olap query recommendation. In: DBKDA 2013, pp. 9–14 (2013)
12. Koutrika, G., Ioannidis, Y.: Personalized queries under a generalized preference model. In: Proceedings of 21st International Conference on Data Engineering, ICDE 2005, pp. 841–852. IEEE (2005)
13. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: ICCS, pp. 32–43 (1995)
14. Missaoui, R., Kwuida, L.: Mining triadic association rules from ternary relations. In: Valtchev, P., Jäschke, R. (eds.) ICFCA 2011. LNCS, vol. 6628, pp. 204–218. Springer, Heidelberg (2011)
15. Nguyen, T.K.N.: Generalizing Association Rules in N-ary Relations: Application to Dynamic Graph Analysis. Phd thesis, INSA de Lyon (October 2012)
16. Pasquier, N.: Data Mining: algorithmes d'extraction et de réduction des règles d'association dans les bases de données. PhD thesis (January 2000)
17. Patrick, M., Elsa, N., et al.: A survey of query recommendation techniques for datawarehouse exploration. In: EDA 2011 (2011)
18. Stefanidis, K., Drosou, M., Pitoura, E.: You may also like results in relational databases. In: PersDB 2009, pp. 37–42 (2009)
19. Trabelsi, C., Jelassi, N., Yahia, S.B.: Bgrt: une nouvelle base générique de règles d'association triadiques. application à l'autocomplétion de requêtes dans les folksonomies. Document Numérique 15(1), 101–124 (2012)
20. Veloso, A., de Almeida, H.M., Gonçalves, M.A., Meira Jr., W.: Learning to rank at query-time using association rules. In: SIGIR, pp. 267–274 (2008)
21. Voutsadakis, G.: Polyadic concept analysis. Order 19(3), 295–304 (2002)
22. Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht-Boston (1982)
23. Wille, R.: The basic theorem of triadic concept analysis. Order 12(2), 149–158 (1995)