

A Formal Concept Analysis Approach to Discover Association Rules from Data

Mondher Maddouri

Department of Mathematics and Computer Sciences,
National Institute of Applied Sciences and Technology of Tunis - INSAT
University of Carthago,
Centre Urbain Nord, B.P. 676, 1080 TUNIS CEDEX, TUNISIE
`mondher.maddouri@fst.rnu.tn`

Abstract. The Discovery of association rules is a non-supervised task of data mining. Its mostly hard step is to look for the frequent itemsets embedded into large amounts of data. Based on the theory of Formal Concept Analysis, we suggest that the notion of *Formal Concept* generalizes the notion of itemset, since it takes into account the itemset (as the *intent*) and the support (as the cardinality of the *extent*). Accordingly, we propose a new approach to mine interesting item-sets as the optimal concepts covering a binary table (*concept coverage*). This approach uses a new quality-criteria of a rule: the *gain* that generalizes the support criteria.

1 Introduction

The volume of stored data increases rapidly. In the past, it doubles each three centuries. Nowadays, it doubles each 72 days. The analysis of huge volumes of data by computers leads to the emerging field of Data-Mining and Knowledge-Discovery. This includes techniques to create knowledge from data sets serving for classification, clustering, prediction, estimation and optimization tasks [1].

The notion of “Concept” as a couple of intent and extent (serving to represent nuggets of knowledge) was introduced and formalized many decades ago [2, 3, 4]. Recently, its use in Data-Mining and Knowledge-Discovery is in growth [5]. Our *ConceptMiner* project (a development environment in MS-Visual C++) aims to design new heuristic algorithms for Data-Mining based on Formal Concept Analysis.

In this paper, we study the clustering problem that we resolve by the induction of association rules. In fact, the experts (biologists) want to know which sub-sequences of proteins that appear together in Tall Like Receptor (TLR) sequences or even in Similar TLR sequences [8].

In section 2, we resume the basic concepts of association rules and we outline the well-known techniques. In section 3, we introduce the basic definitions of Formal Concept Analysis that we use later. In section 4, we give two heuristic algorithms to calculate the *optimal Item-sets* from rows of data. Illustrative examples are given throughout the paper. Finally, we present our concluding remarks and the future perspectives of this work.

2 Basics in Association Rules

Discovering association rules is a non-supervised data-mining task that aims to induce symbolic conditional rules from a data file [1]. The conditional rule has the following syntax: “IF conditions THEN results”. The combination of conditions leads to a descriptive rule like: “IF A AND NOT B THEN C AND D ”. It can lead, also, to a predictive rule including the time dimension like: “IF condition at time_1 THEN result at time_2”. This looks like: “IF $he_bays(TV, today)$ THEN $he_bays(recorder, in_one_year)$ ”.

The association rules are used in a wide range of applications, due to their comprehensive format. Mainly, it is used to analyze the supermarket tickets. This helps in optimal stock management, in merchandizing and marketing [1].

During the few last years, a wide variety of algorithms have been proposed. The most popular algorithm is certainly the iterative one: *Apriori* [9]. It consists in three steps. First aboard, it considers all the attributes/properties describing the set of data as the initial itemsets. Then, it combines the couples of item-sets in order to obtain larger item-sets. Finally, it searches for the rules that can be derived from each itemset.

MaxEclat and *MaxClique* [10] use graph theory to decompose the lattice into sublattices in order to mine maximal frequent item-sets. *MaxMiner* [11] is an exploring algorithm for the maximal frequent item-sets with a look-ahead pruning strategy. *PincerSearch* [12] follows both a bottom-up and a top-down search in the lattice at the same time while looking for the maximal frequent patterns. *GenMax* [13] uses a backtracking search to enumerate all maximal frequent item-sets and eliminates progressively non maximal ones. *Mafia* [14] works with a depth first search and uses pruning strategies to mine maximal frequent item-sets. *DepthProject* [15] follows dynamic re-ordering to reduce the search space. Salleb & al. [16] uses a boolean algebra approach that loads the data base in main memory as a binary decision diagram, and searches for maximal frequent item-sets by making iteratively the product of vectors representing the items. The reader can find further description on these algorithms in [13, 16].

Generally, the number of association rules (induced by these algorithms) grows in an exponential manner with the number of data-rows (or the attributes). This can reach hundreds’ of thousands using only some thousands of data rows [17]. So, it becomes very hard to read and interpret them. The induced rules can be classified in three categories:

- *Useful rules*: that a human expert can understand and use.
- *Trivial rules*: that represents evidence. They are valid but never used.
- *Weak rules*: those are not acceptable by the expert and not understandable.

To remedy to this problem, many methods were proposed [17]. The basic one consists of calculating a minimal set of rules that allows us to re-produce the others. This can be done while calculating the rules [18], or after that using the concept lattice [19, 20]. Another method searches only the rules where the condition or the result part matches a particular group of terms [21]. Agrawal et al. [9] maintains a partial order relation between the rules using two measures: the *support* and the *confidence*. Consider the rule: $X \rightarrow Y$, where X (conditions) and Y (results) are sets of atomic propositions:

- *Support* ($X \rightarrow Y$): the percentage of examples satisfying both X and Y .

- *Confidence* ($X \rightarrow Y$): the number of examples satisfying both X and Y , divided by the number of examples satisfying only the condition X .

The authors introduce two thresholds defined by the end-user:

- *MinSup*: the minimal value of the rule support that can be accepted.
- *MinConf*: the minimal value of the rule confidence that can be accepted.

The method discards all the item-sets with a support less than *MinSup*, and all the rules with a confidence less than *MinConf*.

To filter more efficiently the association rules, some authors [22] introduce the *conviction* measure and [17] propose five different measures like the *benefit* (interest) and the *satisfaction*.

In this paper, we introduce a new measure: the *gain*. It combines the support of the rule with the length of the itemset. This is done thanks to the theory of Formal Concept Analysis [3, 4], since we suggest that an item-set is completely represented by a formal concept as a couple of intent (the classic item-set) and extent (its support).

3. Basics in Formal Concept Analysis

Let O be a set of objects (or examples), P a set of properties (or attributes) and R a binary relation defined between O and P [3, 4].

Definition 1 [3]: A *formal context* (O, P, R) consists of two sets O and P and a relation R between O and P . The elements of O are called the *objects* and the elements of P are called the *properties* of the context. In order to express that an object o is in a relation R with a property p , we write oRp or $(o, p) \in R$ and read it as "the object o has the property p ".

Definition 2 [3]: For a set $A \subseteq O$ of objects and a set $B \subseteq P$ of properties, we define :

The set of properties common to the objects in A :

$$A \blacktriangleright = \{p \in P \mid oRp \text{ for all } o \in A\} \quad (1)$$

The set of objects which have all properties in B :

$$B \blacktriangleleft = \{o \in O \mid oRp \text{ for all } p \in B\} \quad (2)$$

The couple of operators $(\blacktriangleright, \blacktriangleleft)$ is a *Galois Connection*.

Definition 3 [3]: A *formal concept* of the context (O, P, R) is a pair (A, B) with

$$A \subseteq O, B \subseteq P, A \blacktriangleright = B \text{ and } B \blacktriangleleft = A. \quad (3)$$

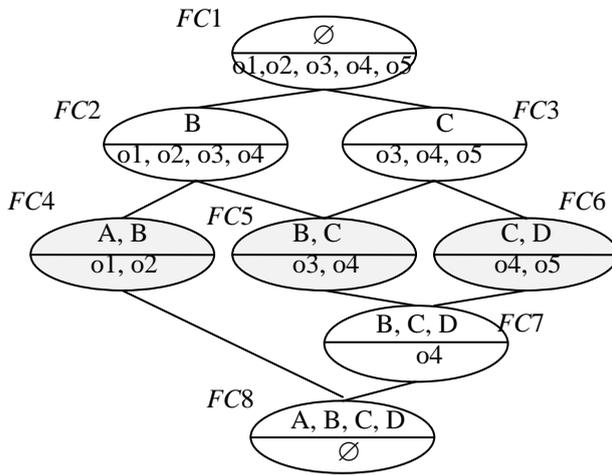
We call A the *extent* and B the *intent* of the concept (A, B) .

Definition 4 [3]: The set of all concepts of the context (O, P, R) is denoted by $\mathcal{F}(O, P, R)$. An ordering relation (\ll) is easily defined on this set of concepts by :

$$(A_1, B_1) \ll (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 \Leftrightarrow B_2 \subseteq B_1 \quad (4)$$

| P | A | B | C | D |
|-----------|----------|----------|----------|----------|
| o1 | 1 | 1 | 0 | 0 |
| o2 | 1 | 1 | 0 | 0 |
| o3 | 0 | 1 | 1 | 0 |
| o4 | 0 | 1 | 1 | 1 |
| o5 | 0 | 0 | 1 | 1 |

1.a Formal context (O, P, R)



1.b Concept Lattice of the context (O, P, R)

Fig. 1. An example of formal context (O, P, R) and its corresponding Concept Lattice.

Basic Theorem for Concept Lattices [3]:

$(\mathcal{B}(O, P, R), \ll)$ is a complete lattice. It is called the *concept lattice* (or *GALOIS lattice*) of (O, P, R), for which infimum and supremum can be described as follow:

$$\text{Sup}_{i \in I} (A_i, B_i) = ((\cup_{i \in I} A_i) \uparrow \downarrow, (\cap_{i \in I} B_i)) \quad (5)$$

$$\text{Inf}_{i \in I} (A_i, B_i) = (\cap_{i \in I} A_i, (\cup_{i \in I} B_i) \downarrow \uparrow) \quad (6)$$

Example : Figure 1 (graphic 1.a) is used to illustrate the notion of formal context (O, P, R). This context includes five objects {o1, o2, o3, o4, o5} described by four properties {A, B, C, D}. The concept lattice of this context is drawn in Figure 1 (graphic 1.b). It contains eight formal concepts.

Definition 5 [4]: Let (o, p) be a couple in the context (O, P, R) . The *pseudo-concept* PC containing the couple (o, p) is the union of all the formal concepts containing (o,p) .

Definition 6 [4]: A *coverage* of a context (O, P, R) is defined as a set of formal concepts $CV=\{RE_1, RE_2, \dots, RE_n\}$ in $\mathcal{F}(O, P, R)$, such that any couple (o, p) in the context (O, P, R) is included in at least one concept of CV .

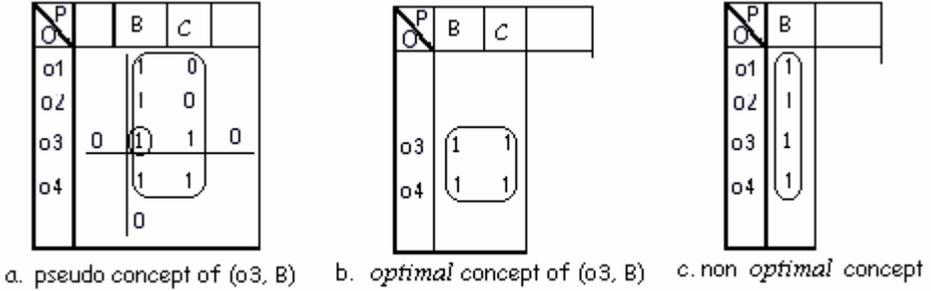


Fig. 2. An example of pseudo concept, optimal concept and non optimal concept containing the couple $(o3,B)$.

Example: To illustrate the notion of pseudo-concept and coverage, we consider the same formal context (O, P, R) from figure 1. Figure 2 (graphic 2.a) represents the pseudo-concept containing the couple $(o3, B)$. It is the union of the concepts $FC2$ and $FC5$ (see figure 1).

A coverage of the context (O, P, R) can be formed by the three concepts: $\{FC4, FC5, FC6\}$. $FC4$ is the concept containing the items $(\{o1, o2\}, \{A, B\})$. $FC5$ is the concept containing the items $(\{o3, o4\}, \{B, C\})$. $FC6$ is the concept containing the items $(\{o4, o5\}, \{C, D\})$. The lattice constitutes also a concept coverage.

4 Discovery of Optimal Item-sets

The most computationally expensive step to discover association rules is the calculation of the frequent item-sets [15]. Generally, this step consists of applying, iteratively, some heuristics to calculate candidate item-sets. At the iteration i , we combine the item-sets of the iteration $i-1$. After that, the support threshold (*minsupp*) is used to discard non-frequent candidates. The item-sets of iteration $i-1$, are also discarded. We keep the remaining item-sets of the latest iteration n (where n is the number of properties in the formal context).

In this step, we think that many approaches use, in an implicit manner, two basic characteristics of the association rules:

- *General rules:* we should ignore the item-sets with high support and low cardinality. General rules are weak, since they are valid for numerous examples (or observations).

- *Specific rules*: we should ignore the item-sets with high cardinality and low support. Specific rules are weak, since they are valid only for few examples (or observations).

The two characteristics are defined based on the cardinality of two special sets:

- *Support*: that is the cardinality of the set of examples verifying the rule. In Formal Concept Analysis, this represents the **extent** of a formal concept.
- *Cardinality of the Item-set*: that is the number of properties in the item-set. In Formal Concept Analysis, this represents the **intent** of a formal concept.

Consequently, we think that an association rule should not be represented only by the item-set (that is the intent). **An association rule is completely and explicitly represented by a formal concept (as a couple of intent and extent).**

Also, we think that the selection of “good” item-sets should not be based only on the support (that is the cardinality of the extent). **A powerful selection should be based on the cardinalities of both the extent and the intent of its formal concept.**

To formalize the new criterions, we give the following definitions.

Definition 7: Let $FC_i = (A_i, B_i)$ be a formal concept. We define:

- *Length* of a concept FC_i : the number of properties in the intent B_i of the concept.
- *Width* of a concept FC_i : the number of examples in the extent A_i of the concept.
- *Gain* of a concept: is a function of the width and the length, given by:

$$\text{Gain}(FC_i) = \text{width}(FC_i) * \text{length}(FC_i) - [\text{width}(FC_i) + \text{length}(FC_i)] \quad (7)$$

In a manner that the gain function cannot be maximised when one of the two parameters is low and the other is high. It becomes maximal only when the two parameters are high (a lot of examples and a lot of properties). Hence, we are sure that the rule is not a general one, neither a specific one. Note also that the *Width* indicates the support of the itemset (or the corresponding association rule).

Definition 8 [4]: A formal concept $FC_i = (A_i, B_i)$ containing a couple (o, p) is said to be *optimal* if it maximizes the gain function.

Definition 9 [4]: A coverage $CV = \{ FC_1, FC_2, \dots, FC_k \}$ of a context (O, P, R) is *optimal* if it is composed by optimal concepts.

Example: Figure 2 (graphic 2.b) represents the optimal concept $FC5$ containing the couple $(o3, B)$ and having the gain value 0. Figure 2 (graphic 2.c) represents the non optimal concept $FC2$ containing the couple $(o3, B)$ and having the gain value -1.

The optimal coverage of the context (O, P, R) is formed by three optimal concepts: $\{FC4, FC5, FC6\}$. $FC4$ is the concept containing the items $(\{o1, o2\}, \{A, B\})$. $FC5$ is the concept containing the items $(\{o3, o4\}, \{B, C\})$. $FC6$ is the concept containing the items $(\{o4, o5\}, \{C, D\})$.

4.1 Heuristic Searching for Optimal Concept

The pseudo-concept containing the couple (o, p) , introduced in definition 5, is the union of all the concepts containing (o, p) . The importance of the notion of pseudo-concept, indicated by *PCF*, is that it can be calculated in a simple manner as the restriction of the relation R by the set of objects/examples described by p , so $\{p\} \blacktriangleright$, and the set of properties describing the object o , so $\{o\} \blacktriangleleft$. Where $(\blacktriangleright, \blacktriangleleft)$ is the Galois connection of the context (O, P, R) .

Once we obtain the pseudo-concept *PCF*, two cases are considered:

- Case 1: *PCF* forms a formal concept. This means that no zero is found in the relation/matrix representing *PCF*. Then, *PCF* is the optimal concept that we are looking for. The algorithm stops.
- Case 2: *PCF* is not a formal concept. This means, simply, that we can find some zero entries in the relation/matrix representing *PCF*. In this case, we will look for more restraint pseudo-concepts within the pseudo-concept *PCF*. So, we consider the pseudo-concepts containing the couples like (X, p) or (o, Y) . These concepts contain, certainly, the couple (o, p) . The heuristic that we use here is that: ***the optimal concept is included in the optimal pseudo-concept***. So, we should calculate the Gain value of the different pseudo-concepts containing the couples like (X, p) or (o, Y) . Then we choose the pseudo-concept having the greatest value of the Gain function to be the new *PCF*.

This heuristic procedure (of case 2) is repeated until we meet the case 1: until *PCF* becomes a formal concept. To calculate the gain of a pseudo-concept, we introduce the general form of the previous function:

Definition 10: Let $PCF_i = (A_i, B_i, R_i)$ be a pseudo-concept, where R_i is the restriction of the binary relation R , to the subsets A_i and B_i . We define the:

- *Length* of a pseudo-concept PCF_i : the number of properties in B_i .
- *Width* of a pseudo-concept PCF_i : the number of examples in A_i .
- *Size* of a pseudo-concept PCF_i : the number of couples (of values equal to 1) in the pseudo-concept. When PCF_i is a formal concept, we have:

$$Size(PCF) = Length(PCF) \times Width(PCF) \quad (8)$$

- *Gain* of a pseudo-concept: is a function of the width, the length and the size given by (9):

$$Gain(PCF) = \left[\frac{Size(PCF)}{Length(PCF) \times Width(PCF)} \right] \times [Size(PCF) - (Length(PCF) + Width(PCF))]$$

4.2 Algorithm for Optimal Coverage

The problem of covering a binary relation by a set of optimal concepts is known as the problem of covering a binary matrix by a number of its complete sub-matrix. A

complete sub matrix is a matrix where all its entries are equal to '1'. This was found as an NP-hard problem. Here, we use a polynomial heuristic solution.

Let R be the binary relation to cover. The heuristic solution consists of dividing R into p packages (subsets): P_1, \dots, P_p . Each package represents one or more couples. The idea is to construct step by step the optimal coverage of R . In the first step, we cover the relation $R_1=P_1$ by CV_1 . In the k^{th} step, let $R_{k-1}=P_1 \cup \dots \cup P_{k-1}$ and let CV_{k-1} be its optimal coverage. The task is to construct the optimal coverage CV_k of $R_k=R_{k-1} \cup P_k$ using only the initial coverage CV_{k-1} and the package P_k . In the p^{th} step, we obtain a set of concepts covering the relation R .

Algorithm for building the Coverage

Begin

Let R be partitioned to p packages P_1, \dots, P_p .

Let $CV_0:=\emptyset$.

FOR $k=1$ **to** p **DO**

Sort the couples of P_k by the pertinence of their pseudo-concepts

While ($P_k \neq \emptyset$) **Do**

-Select a couple (a, b) in P_k by the sorted order of the gain function (9)

-Search PC : the pseudo-concept containing (a, b) within $R_k=CV_{k-1} \cup P_k$

-Search FC: the optimal concept containing (a, b) within PC

$CV_k := (CV_{k-1} - \{r \in CV_{k-1} / r \subseteq FC\}) \cup \{FC\}$: Delete all the redundant concepts from CV_k

$P_k := P_k - \{(X, Y) \in P_k / (X, Y) \in FC\}$

End While

End FOR

End.

| | P | A | B | C | D |
|----|---|---|---|---|---|
| O | | | | | |
| o1 | | 1 | 1 | 0 | 0 |
| o2 | | 1 | 1 | 0 | 0 |
| o3 | | 0 | 1 | 1 | 0 |
| o4 | | 0 | 1 | 1 | 1 |

a. optimal coverage of the context $\{(o1, o2, o3, o4), \{A, B, C, D\}, R4\}$

| | P | A | B | C | D |
|----|---|---|---|---|---|
| O | | | | | |
| o1 | | 1 | 1 | 0 | 0 |
| o2 | | 1 | 1 | 0 | 0 |
| o3 | | 0 | 1 | 1 | 0 |
| o4 | | 0 | 1 | 1 | 1 |
| o5 | | 0 | 0 | 1 | 1 |

b. optimal coverage of the context $\{(o1, o2, o3, o4, o5), \{A, B, C, D\}, R\}$

Fig. 3. Illustration of the algorithm for concept coverage.

Example: Let R be the relation of figure 1. R is partitioned into five packages: $P1=\{o1\} \times \{A, B\}$, $P2=\{o2\} \times \{A, B\}$, $P3=\{o3\} \times \{B, C\}$, $P4=\{o4\} \times \{B, C, D\}$ and $P5=\{o5\} \times \{C, D\}$. Initially, R is an empty relation and in each step we add a package. Figure 3 presents the incrementation step when adding $P5$.

In this step R4 encloses the four rows P1, ..., P4. The initial optimal coverage CV4 encloses the formal concepts $FC4 = (\{o1, o2\}, \{A, B\})$ and $FC5 = (\{o3, o4\}, \{B, C\})$ and $FC7 = (\{o4\}, \{B, C, D\})$ (graphic 3.a). The package P5 encloses only two couples. The pseudo concept containing the couple $(o5, C)$ represents the union of two formal concepts FC3 and FC6. So, the optimal concept containing $(o5, C)$ and $(o5, D)$ is the concept FC6. The concept FC7 is redundant. Thus, the final coverage of R contains the concepts FC4, FC5 and FC6.

In conclusion, from the data set of figure 1, we discover three Item-sets : $\{A, B\}$, $\{B, C\}$ and $\{C, D\}$. These Item-sets will be processed later by the Apriori algorithm, to calculate the corresponding association rules.

5 Experiments

The experiments are done with a PC PENTIUM II, 233 MHZ, 144 Mo of RAM and running under MS-Windows 98. The analyzed data sets are organized in two groups. The first group is made by five data sets representing biological sequences coding macromolecules (table 1). The first two data sets are proprietary sets for the TLR macromolecules. The three others are taken from the machine-learning repository at UCI University. Classic tables of data make the second group of data sets (table 2). The first two data sets are also taken from the machine-learning repository at UCI University. The other sets are taken from a proprietary database.

Table 1: Properties of the biological data sets.

| Biological Data Sets | Identifier | Rows | Columns |
|--------------------------|------------|------|---------|
| 2 Far a way TLR Families | DB1 | 763 | 16 |
| 6 Near TLR Families | DB2 | 619 | 14 |
| Promoter gene | DB3 | 106 | 11 |
| Junction gene 3' | DB4 | 1150 | 19 |
| Junction gene 5' | DB5 | 500 | 10 |

Table 2: Properties of the data sets having standard format.

| Standard Data Sets | Identifier | Rows | Columns |
|--------------------|------------|------|---------|
| Tic Tac Toe | DB6 | 958 | 10 |
| Mushroom | DB7 | 1191 | 23 |
| T20i4d1k | DB8 | 1000 | 9 |
| T10I4D | DB9 | 5000 | 9 |
| T20I6D | DB10 | 1000 | 9 |

To evaluate the performance of the proposed method (IAR: Incremental induction of Association Rules), we compare it to the existing approach Aprior [9, 23]. The Apriori method was programmed in C++ and integrated in our software package. We

use the two methods to analyze the ten data sets. We choose two sets of parameters. The first one with $MinSup=0.6$ and $MinConf=0.5$ is used for the biological data sets (table 1). The second one with $MinSup=0.35$ and $MinConf=0.75$ is used for the standard data sets (table 2).

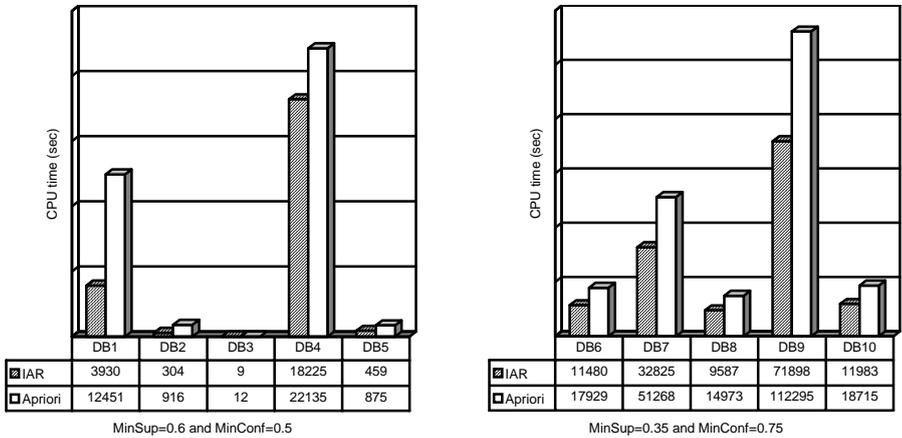


Fig. 4: Comparing the CPU time

5.1. Comparing the CPU time

Figure 4 presents the CPU time measured in seconds for the three methods. We remark that the three methods keep the same behavior overall the data sets. The Apriori method takes the greater time, since it is based on an exhaustive approach to test all the possible combinations. The proposed method IAR takes always a CPU time better than Apriori.

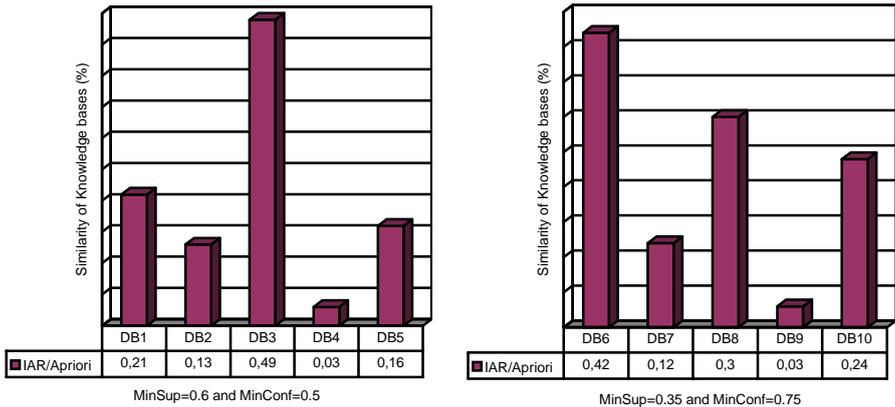


Fig. 5: Similarity between the knowledge bases of the two methods

5.2. Measuring the similarity of rules

Figure 5 presents the percentage of rules that are obtained by IAR in comparison with APRIORI. We remark that the percentage is always less than the half (0.5). We remark also that Apriori gives, always, a lot of rules compared to IAR.

6 Conclusion

In this paper, we presented a new Formal Concept Analysis approach to search for optimal item-sets. We gave two heuristic algorithms to build the coverage of formal concepts and to create optimal item-sets from data records. The association rules can be induced from the Item-sets in the same manner as *Apriori* [9].

We assume that a Formal Concept generalizes the notion of Item-set, since it is more powerful and more explicit. In fact, we note that an item-set represents only the intent of a Formal Concept. Also, the support of an item-set is the cardinality of the concept extent. Hence, the Formal Concept encloses more information than the item-set itself.

The first experimentation of the proposed method is done on a variety of data sets having different data formats (structured and unstructured). We compared the proposed method with a known one: Apriori [9]. We measured their CPU time, the number of association rules, and the percentage of similar rules. We remarked that Apriori gives a great number of rules and takes a great time. The proposed method IAR is faster than Apriori. It gives a low number of rules that have the advantage of being the common rules.

Future work will focus on the quality of the created association rules. In fact, we plan to study the significance of the discovered rules for human experts. For this reason we propose to present the rules concerning the TLR macromolecules to our biologists in order to get their points of view. Accordingly, we plan to study quality measures other than the support and the confidence [17, 22], in order to identify the measure that gives expert-acceptable rules.

References

1. P. Adriaans et D. Zantinge. *Data mining*. Addison Wesley Longman, England, 1996.
2. R. Wille. Restructuring Lattice Theory : an Approach Based on Hierarchies of Concepts. In *Proc. Symposium on Ordered Sets*, pp.445-470, Dordrecht-Boston: reidel, 1982.
3. Ganter B., Wille R., *Formal Concept Analysis: Mathematical Foundations*, Edition: Springer, Heidelberg-Berlin-New York, 1999
4. Jaoua A., Beaulieu J. M., Belkhitir N., Deshernais J., Reguig M., *Optimal rectangular decomposition of a finite binary relation*, International Conference on Discrete Mathematics (sixth conference, June 1992, Vancouver-Canada).
5. Wille R., Why can Concept Lattice Support Knowledge Discovery in DataBases, First Workshop on Formal Concept Analysis Advances for KDD, 2001
7. M. Maddouri. *Contribution to Concept Learning : an Incremental approach for Inducing Production Rules from examples*. PhD dissertation from the university of TUNIS II, May 2000.

8. M. Maddouri, M. Elloumi, A Data Mining Approach based on Machine Learning Techniques to Classify Biological Sequences, Knowledge Based Systems Journal, March 2002.
9. Agrawal R., Imielinski T, Swami A.: Mining association rules between sets of items in large databases. ACM SIGMOD Int. Conf. on Management of Data (1993) 207-213.
10. Zaki M. J.: Scalable algorithms for association mining. Int. J. IEEE Transactions on Knowledge and Data Engineering, Vol 12(3). (May/June 2000) 372-390.
11. Bayardo R.: Efficiently mining long patterns from databases. ACM SIGMOD Int. Conf. on Management of Data (1998) 85-93.
12. Lin D., Kedem Z.: Pincer search: a new algorithm for discovering the maximum frequent set. Int. Conf. on Extending Database Technology, (March 1998).
13. Gouda K., Zaki M. J.: Efficiently mining maximal frequent itemsets. IEEE Int. Conf. on Data Mining, (November 01).
14. Burdick D., Calimlim M., Gehrke J. E.: Mafia: a maximal frequent itemset algorithm for transactional databases. Int. Conf. on Data Engineering, (April 2001).
15. Agarwal R. C., Aggarwal C. C., Prasad V. V. V.: Depth first generation of long patterns. Int. Conf. on Knowledge Discovery and Data Mining, (August 2000) 108-118.
16. Salleb A., Maazouzi Z., Vrain C.: Mining maximal frequent item-sets by a Boolean based approach. European Conf. on Artificial Intelligence, Lyon France (July 2002) 285-289.
17. Cherfi H., Toussaint Y.: Text mining by combining association rules and statistics. Int. Workshop on Text Mining CIFT-02, Hammamet – Tunisia (October 21-23, 2002) 67-80.
18. Guigues J. L., Duquenne V.: Familles minimales d'implication informatives resultant d'un tableau de données binaires. J. Mathematiques, informatique et Sciences Humaines, Vol. 95 (1986) 5-18.
19. Stumme G., Taouil R., Bastide Y., Pasquier N., Lakhal L.: Intelligent structuring and reducing of association rules with formal concept analysis. Lecture Notes in Advances in Artificial Intelligence, KI-2000 Proceedings, Vol. 2174. Springer-Verlag (2001) 335-350.
20. Toussaint Y., Simon A.: Building and interpreting term dependencies using association rules extracted from Galois lattices. Int. Conf. on Content-Based Multimedia Information Access, Paris France (2000) 1686-1693.
21. Klemettinen M., Manilla H., Ronkainen P., Toivonen H, Verkamo A. I.: Finding interesting rules from large sets of discovered association rules. Int. Conf. on Knowledge Management, Gaithersburg USA (1994) 401-407.
22. Bayardo R., Agrawal R.: Mining the most interesting rules. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (1999) 145-154.
23. D. A. Zighed, J. P. Auray, G. Duru. *SIPINA: méthode et logiciel*. Alexandre Lacassagne, Lyon-France, 1992.
24. Dougherty J., Kohavi R., Sahami M.: Supervised and Unsupervised Discretization of Continuous Features. Int. Conf. on Machine Learning (1995).