A NEW EXTRACTION OPTIMIZATION APPROACH TO FREQUENT 2 ITEMSETS

Nombre Claude Issa[1], Brou Konan Marcellin[2], Kimou Kouadio Prosper[3]

[1]Polytechnic Doctoral School
[2]National Polytechnic Institute Houphouet Boigny – Yamoussoukro
[3]Research Laboratory of Computer Science and Technology

ABSTRACT

In this paper, we propose a new optimization approach to the APRIORI reference algorithm (AGR 94) for 2-itemsets (sets of cardinal 2). The approach used is based on two-item sets. We start by calculating the 1-itemets supports (cardinal 1 sets), then we prune the 1-itemsets not frequent and keep only those that are frequent (ie those with the item sets whose values are greater than or equal to a fixed minimum threshold). During the second iteration, we sort the frequent 1-itemsets in descending order of their respective supports and then we form the 2-itemsets. In this way the rules of association are discovered more quickly. Experimentally, the comparison of our algorithm OPTI2I with APRIORI, PASCAL, CLOSE and MAX-MINER, shows its efficiency on weakly correlated data. Our work has also led to a classical model of side-by-side classification of items that we have obtained by establishing a relationship between the different sets of 2-itemsets.

KEYWORDS

Optimization, Frequent Itemsets, Association Rules, Low-Correlation Data, Supports.

1. INTRODUCTION

In view of the growing number of large databases and the complexity of the algorithms implemented for its exploitation, the question of optimization is of increasing concern to researchers in the field of data mining.

The discovery of useful knowledge from large data is a process of data mining. One of the most used techniques for extracting this knowledge is the method of association rules. It makes it possible to look for the similarities between the data of a database.

Many works and applications have been proposed. From these works emerge two main lines of research:

-The first axis concerns the discovery of association rules that are relevant and useful to experts in a given field.

- The second axis is interested in the extraction optimization of these association rules.

To deal with this, various approaches have been proposed of which we mention some.
The Apriori algorithm (Agrawal and Srikant, 1994 [AS94]) based on the support-trust pair is the first efficient model that deals with the problem of extracting association rules. It is a level algorithm that relies on the anti-monotonic property of the support. The Apriori-TID algorithm proposed by Agrawal himself (Agrawal and Srikant [AS94]) seeks to keep the context in memory to reduce the storage space. The Partition algorithm (Savasere et al. [SON95]) partitions the entire

DOI:10.5121/ijcsa.2019.9201

database in sub bases empty intersection to hold in memory. The Eclat algorithm (Zaki et al., [ZPOL97]) dedicated to the search for frequent sets of patterns traverses the deep search space. The FP-growth algorithm (Han et al [HPYM00]) uses a particular data structure called FP-tree (Frequent-Pattern tree). The idea is to build a tree summarizing the database and browse it in depth to generate all the common patterns.

The minimal pattern study is also highly developed (Calders et al [CRB04], Li et al [LLW + 06], Liu et al [LLW08], Szathmary et al [SVNG09]). Those are width methods using the support-trust approach, so they have similar limits to those of Apriori. More recently, the DEFME algorithm (Soulet and Rioult [SR14]) offers a new method for efficiently extracting frequent itemsets. It is an in-depth algorithm that extends the concept of closure developed in (Han et al [HPYM00]).

In this paper, we focus primarily on the second axis of research by developing another approach to optimize the discovery of frequent k-itemsets (k = 2). The generation of association rules with a premise and a conclusion that is potentially useful to the end user results.

The association rules (of the form a→b) formed using two-item sets are interesting for our study because of their number considered high, especially in the case of large databases. Moreover, they also allow the classification of items of any database.

Our work will also rely on the couple of support-confidence measures for the discovery of frequent itemsets. But to increase the speed of extraction of these, we will use another strategy.

Our work will begin by presenting a state of the art in the field of the discovery of frequent itemsets while presenting its problematic context. This paper has on the one hand a main objective which consists in optimizing the discovery of frequent item sets, and on the other hand a specific objective which consists in classifying side by side the articles of a big commercial surface To pass of the main objective for the specific purpose, we use sets of maximal cardinal items equal to 2. Thus we propose a new approach that focuses only on one- and two-item sets to optimize the extraction of association rules. Then we came up with an interesting result that allows you to put together items from a large commercial area. This side-by-side arrangement of articles takes into account the relationship between the different sets of 2-itemsets. Finally we conclude with perspectives.

2. SECTION 1

2.1. STATE OF THE ART ON THE DISCOVERY OF FREQUENT ITEMSETS

Our work is based on the extraction of frequent itemsets (an itemset is a set of items of size k). Several works have been carried out in this field, thus opening two major lines of research. For two decades, research has focused on two main objectives:

- Extraction optimization of frequent itemsets
- Extraction of relevant association rules (knowledge) from frequent itemsets

The first axis of research uses the strategy according to which the set of frequent itemsets is traversed by iterative loop. At each step k, a set of candidates is created by joining the frequent (k-1) -itemsets. Then, the k-itemsets supports are calculated and evaluated with respect to a minimum threshold set by the user according to his domain, in order to discover frequent k-itemsets. The non-frequent itemsets are deleted. The algorithm that uses this strategy is the Apriori reference algorithm [AGR 94], proposed in parallel with the OCD algorithm [MAN 94]. Since then, interesting work has been done to improve one or more aspects of this algorithm [BRI

97b, GAR 98, PAR 95, SAV 95, TOI 96]. The common point of all these algorithms is to calculate the support of all itemsets without worrying about whether they are frequent or not.

Another Apriori optimization approach proposed in 2010 by Yves Bastide et al. [BAS 10] in his article "PASCAL: An optimization of extraction of frequent patterns. ". This method is intended to reduce the number of itemset support calculations when retrieving frequent itemsets. This method is based on the concept of key itemsets. A key reason is a minimal pattern of an equivalence class that groups all the itemsets contained in exactly the same objects in the database. All itemsets of an equivalence class have the same support, and the support of non-key itemsets of an equivalence class can therefore be determined using the support of the key itemsets of this class. With inference counting, only frequent (and some non-frequent) key itemets are computed from the database.

The second axis deals with the extraction of maximum frequent itemsets. This strategy makes it possible to determine sets of frequent itemsets (maxims) which none of its immediate supersets are frequent.

The algorithms that use this strategy run through the itemsets by combining the artificial intelligence technique of the first course from bottom to top and top to bottom. Frequent itemsets are extracted immediately after the maximum itemsets are known. Max-clique and Max-eclat algorithms [ZPOL97], Max-miner [Bay98], Pincer-search [LK98], Depth-Project [AAP00] allow to browse all the itemset of the lattice (2n itemsets possible), but their performance decreases as n increases due to the cost of the last scan. The most efficient algorithm based on this approach is the Max-Miner algorithm.

The third axis deals with the discovery of frequent closed itemsets. This strategy uses the closing of the Galois connection [GAN 99, DUQ 99]. A closed itemset is a set of items whose support is different from all the supports of its supersets. The algorithms that use this strategy perform a level scan, while discovering at each step all frequent itemsets and their support from frequent closed itemsets, without making access to the database. The most efficient algorithm based on this approach especially on strongly correlated data is the Close algorithm [PAS 99c],

The fourth axis is based on a hybrid method using the first two approaches. The algorithms adopting this strategy explore the space of deep search first such as the algorithms of the second method often called "Divide-and-generate", but without dividing the extraction context into sub-contexts. These algorithms generate a set of candidates as is the case in the first method often called "Generate-and-test". However, this set is always reduced to a single element. The latter consist in using a statistical metric in conjunction with other heuristics.

In addition there are other works based on the hybrid strategy. This work has made it possible to combine data mining techniques with one another or between data mining techniques and probabilistic methods such as Bayesian networks. Some interesting works have been done in this line of research:

- Jaroszewicz and Simovici (2004); Jaroszewicz and Scheffer (2005) describe the use of a Bayesian network to compute the interest of extracted attribute sets using an Apriori algorithm.

- The work of Nicolas Voisine et al. (2009) presented a new automatic algorithm for learning decision trees. They approach the problem according to a Bayesian approach by proposing, without any parameter, an analytical expression of the probability of a tree knowing the data. Finally, they transform the problem of tree construction into an optimization problem.

- The work of Eliane Tchiengue (2011) made it possible to propose in a project NICE Data mining of the information system of the French Credit Agricole, with the aim of bringing solutions for the implementation of a tool of analysis of Data mining data and modeling that optimizes customer knowledge.

- Clement Faure (2007, 2014) shows the interest of the implementation of Bayesian networks coupled with the extraction of so-called strong delta association rules, where the user (expert) is at the center of the discovery of the rules potentially useful association is facilitated by the exploitation of knowledge described by the expert and represented in the Bayesian network.

- The Microsoft Association Algorithm (2016) is an algorithm that is often used for recommendation engines. A referral engine recommends items to customers based on items they have already purchased, or in which they have expressed interest. The Microsoft Association algorithm is also useful for market basket analysis.

2.2. CONTRIBUTION

Our working hypothesis takes into account sets of one- and two-item itemsets. The contributions of our article are divided into two points. First, we show a new approach to optimization of 2-itemets extraction, then we try to solve a problem for decision-makers who often have a hard time choosing items to have together in a large area for example. The first point is to significantly improve the extraction time of frequent 2-itemsets.

3. SECTION 2

3.1 MATERIAL AND METHODOLOGY

3.1.1 Context

We have noticed that the populations of third world countries in general, and the Ivorian population in particular, according to their social category makes their purchases in supermarkets according to circumstantial events, periods and types of products. We can classify this clientele into two main categories:

- One who buys once a month, usually between the 28th of the current month and the 5th of the following month;
- The one who makes occasional purchases

With this in mind, we decided to tackle the problem more closely. For that, we had contacted a big store of the place, the CDCI, to have some receipts that we could gather over a period of three (03) consecutive months, going from January 2016 to July 2017.

By closely analyzing these documents and questioning the store managers, we were able to confirm the remark made above. Because specific articles were frequently bought together at the end of the month, from the 28th of the current month to the 5th of the following month. At this time, the turnover of the store climbed strongly. After the 5th and before the 28th of the current month, the store recorded a relative decline in the number of customers per day, and its turnover dropped significantly. In addition, the manager of the store said that larger purchases are recorded during the holiday season. It should also be noted that with the permission of the manager, we were able to visit the layout of products in the store. At the end of this visit, we drew up a map of the store's items. By reconciling the disposition of items with the cash receipts available to us; and another surprising finding has been made.

Indeed, on the cash receipts we found items that were bought most often together (finding made from cash receipts), but which were physically distant from each other. We therefore concluded that the management of Ivorian stores in general and in particular the CDCI of Yamoussoukro that we studied, is not dynamic. Thus, a number of questions that we will try to answer in the following our paper were released.

3.2. METHODOLOGICAL APPROACH

Our approach applies only to one- and two-item sets with two levels (i = 1 and i = 2). This choice is easily justified by:

- Since an association rule consists of at least two items of the form a→b (a is the premise and b the conclusion);

- The extraction time of frequent itemsets of cardinal k (k=1,n), depends on the sum of the elapsed times during the execution of each iteration (complexity). The time taken to execute each iteration depends on the number of itemsets of the same cardinality. Suppose a database D, containing k = 6 items. Let's check the number of itemsets per identical cardinal : SECTION 2

Cardinal n = 1, k = 6	$C_6^1 = \frac{6!}{1!5!} = 6$
Cardinal n = 2, k = 6	$C_6^2 = \frac{6!}{2!4!} = 15$
Cardinal n = 3, k = 6	$C_6^3 = \frac{6!}{3!3!} = 20$
Cardinal n = 4, k = 6	$C_6^4 = \frac{6!}{4!2!} = 15$
Cardinal n = 5, k = 6	$C_6^5 = \frac{6!}{5!1!} = 6$
Cardinal n = 6, k = 6	$C_6^6 = \frac{6!}{6!0!} = 1$

Table 1: The number of itemsets per cardinal n = 1 to 6

From this table, we find that for a small k, the number of sets of itemsets of cardinal 2 is among the highest. As k becomes large, the number of cardinal itemsets 2 becomes negligible. This should not change the general performance of itemet extractions from the i (i varying from 1 to n) levels of our algorithm, as we will demonstrate in Examples 1, 2 and 3 below. Since the factorial function is symmetric with respect to , we can say that optimizing cardinal 2 frequent itemsets is a major advance in the overall optimization of all cardinal items k ( ), mainly for a small k. The representative curve will have the following form: optimization of all cardinal items k (), mainly for a small k. The representative curve will have the following form:

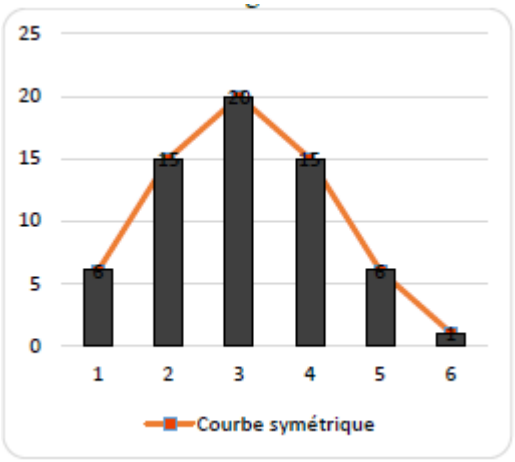


Figure 1-Graphical representation of

In addition, our job is to help the domain expert make good decisions. That is to say, that of arranging side by side the items that appear frequently together on the receipts. This decision therefore justifies our choice to limit the sets of itemset to 2 items maximum. Because by arranging two articles side by side, we can use the chaining technique whose links are chosen according to the measures of interest such as support and trust.

Example: Either a database D = {Bread, milk, oil}. If confidence (bread → milk) = 80%; confidence (bread →milk) = 75%; confidence (bread → oil) = 66%; confidence (bread →oil) = 60%; confidence (oil → milk) = 50%; trust (milk →oil) = 50%, then the arrangement of the three items will be as follows: bread - milk - oil

The problematic of our theme answers the following question: How to improve the times of extraction of the frequent 2- itemsets while discovering knowledge useful to the storage side by side of the articles of a big commercial surface? Several works have already been done in the field of frequent itemset discovery and useful association rules, as we mentioned earlier in the state of the art.

This part is interested in the approach to follow to find a solution to our problematic. Like ARIORI, our approach starts with calculating with access to the database, the supports of 1-itemsets (set containing an item). Then without a second access to the database the supports of the 2-itemsets are determined. Our approach is described in these steps:

1. Calculate the supports of 1-itemsets
2. Remove 1-itemsets not frequent
3. From the supports, sort the frequent 1- itemsets in descending order
4. Set i to 1 and vary

a. If (T(1, j)=0 and T(1, j+1)=0), then the rules T(1, j) →(1, j+1) and T(1, j+1) → (1,d+2) and T (1, j) → (1, d + 2)
b. As long as (T(1, j)=1 and T(1, j+1) = 0), then j←j + 1
c. If (T (1, j) = 1 and T (1, j + 1) = 1), then the rule T (1, j) →T(1, j + 1) gives 1 as partial support and k←k +1
d. If T (1, j) = 1 and T (1, j + 1) = 1 and T (1, j + 2) = 1, then

The rules T (1, j) → T (1, j + 1) gives as support 1 and T (1, j + 1) → T (1, j + 2) gives 1 as support, then T (1, j) → T (1, j + 2) gives 1 as partial support; k ← k + 1

5. Repeat the actions of step 4 for all i (i = 2, n)
6. Calculate the supports of 2-itemsets whose supports per transaction are worth 1
7. Remove the non-frequent 2-itemsets

By traversing all the columns in pairs, if the values of two attributes of the line i are different (one 0 and the other 1) or both equal to 0, then the rule resulting from these attributes is deleted (partial support = 0 - the definition of partial support is given in Definition 13 below). However, if the values are both equal to 1, then the rule is retained and its partial support is equal to 1.

By going through all the columns j, if the value of the attribute on the left is 1 and the value of the attribute on the right is 0, then the rule resulting from these attributes is deleted (support = 0). And then we skip the attribute whose value is equal to 0. We compare thereafter the value of the first attribute and that of the third. If the value of the third attribute is 1 and the value of the 4th attribute is also 1, then the rule resulting from these two attributes has a support equal to 1.

We go through all the lines while proceeding as before, then we calculate the partial supports (sum of the 1) rules which have values equal to 1 on the lines i. If the values of three attributes (consecutive or not) are equal to 1, then using the proposition of the transitivity (proposition 3 below), the support of the attributes 1 and 3 is equal to 1. From the sets of 2 itemsets, we generate all the rules of the form a → b.

Example 1: Let the database D.

Ti	Items
1	{a,c,e}
2	{b,d}
3	{a,b}
4	{a,d,e}
5	{a,c,d, e}

Table 2:The items of the transactions Ti

- Moving from the transactional database to a binary database

Ti	a	b	c	d	e
1	1	0	1	0	1
2	0	1	0	1	0
3	1	1	0	0	0
4	1	0	0	1	1
5	1	0	1	1	1

Table 3: Binary Database D '

- Calculate the supports of 1-itemsets

Support (a) = 4/5 Support (d) = 3/5
Support (b) = 2/5 Support (e) = 3/5
Support (c) = 2/5

Suppose that the minimum support chosen is min_sup = 3, hence the 1-itemset {b} and {c} are removed because they are not frequent.

- Let's sort in descending order of the supports of the 1-itemsets frequent, and we obtain the following new table :

Ti	a	d	e
1	1	0	1
2	0	1	0
3	1	0	0
4	1	1	1
5	1	1	1

Table 4 : Binary table

- Let's train the 2-itemsets while respecting the following conditions:

- If two neighboring attributes of the same line have the same values, then
  - If these values are equal to 0, then the rule resulting from these attributes is canceled (only on line i considered)
  - If these values are equal to 1, then the rule resulting from these attributes is retained
- If two neighboring attributes of the same line have opposite values, then
  - If the value of the second attribute is equal to 1, then the rule from these attributes is cancelled
  - If the value of the second attribute is equal to 0, then the rule resulting from these attributes is canceled and the second attribute is skipped, then the first attribute is compared with the attributes i ← 3 of the array

So for our example we will have :

i = 1 we get a→e {a, e}
i = 2 impossible
i = 3 impossible
i = 4 we get a → d; a→e; d → e {a, d}, {a, e}, {d, e}
i = 5 we get a → d; a→e; d → e {a, d}, {a, e}, {d, e}

- Calculate the supports of the 2-itemsets resulting from the frequent 1-itemsets

Ti	a□ e	a□ d	d □ e
1	1	0	0
2	0	0	0
3	0	0	0
4	1	1	1
5	1	1	1
SUPPORT	3/5	2/5	2/5

Table 5: Binary table of frequent 2-itemsets after transitivity

We see that with min_sup = 3, the 2-itemset {a, e} is retained. To validate the itemset {a, e} as a relevant association rule, its trust must be greater than or equal to a min_conf that we will set equal to 50%.

Confidence (a → e) = Support (a, e) / Support (a) = (3/5) / (4/5) = ¾ = 0.75. That's 75%. Hence, there is a relevant association rule.

3.3. WORKING TOOLS

Our work tools first go through a literature review that allowed us to highlight a state of the art on the discovery of frequent itemsets. We also used propositions based on mathematical notions to consolidate our work, techniques of artificial intelligence and data mining.

The data mining technique used here is the widespread method of association rules. The implementation phase of our work is based on the 2-itemets extraction optimization algorithm of the APRIORI reference algorithm that we call OPTI2I.

3.4. THEORETICAL PHASE

Definition 1. Database
Let O be a finite set of objects, P a finite set of elements or items, and R a binary relation between these two sets. We call database or formal context [GAN 99] the triplet D = (O, P, R). Database D represents our workspace

Definition 2: Transaction

Let T = {T1, T2, T3, ..., Tn}, Ti→ D. We call Ti a set of rows containing the occurrences of the database D. All Tis are called transactions. In the known example of the housewife's basket, the transactions are the cash receipts (that is to say the purchases made by the customers).

Definition 3: Item

We call item any variable Xi representing an occurrence of D

Definition 4: Itemset

We call itemset, the set of items. Example The singleton {X1} and the pair {X1, X2} are itemsets. A itemset of size k is noted k-itemset

Definition 5: Support

We call support the percentage of Ti where the association rule appears, ie

$$\text{Support}(Xi \rightarrow Xj) = \frac{\text{Freq}(Xi \cup Xj)}{\text{Card}(T)}$$

or Support(Xi→Xj) = Number of times

Definition 6: Confidence

We call trust the percentage of times the rule is checked, that is,

$$\text{Confidence}(Xi \rightarrow Xj) = \frac{\text{Freq}(Xi \cup Xj)}{\text{Freq}(Xi)}$$

where Xi and Xj appear togetheri n transactions T

Definition 7 : Superset

A superset is an itemset defined in relation to another itemset. Example {a, b, c} is a superset of {a, b}.

Definition 8: Items and Frequent

A frequent Itemset is an itemset whose support is support or equal to minsup (minimal support below which itemset is considered uncommon). If an itemset is not common, not all his supersets will be. If a superset is frequent then all its sub itemsets are also frequent (anti-monotonic property)

Definition 9: Itemset closed

A frequent itemset is said to be closed if none of its supersets has identical support. In other words, all his supersets have a strictly lower support.

Definition 10: Itemset free

An itemset is free if it is not included in the closure of one of its strict sets

Definition 11: Items and maximum

An itemset is said to be maximal if none of its supersets are frequent.

Definition 12: Generator Items

An itemset is called generator if all its sub itemsets have a strictly superior support.

Definition 13: Partial Support

We say that a support is partial when inside a transaction, two items have the value equal to 1. That is to say that the transaction contains two attributes of the binary base whose values are equal to 1 Example: if a = 1 and b = 1, then the partial support of the rule a→b is equal to 1.

Proposal 1

Let D be a database and Ti the transactions of D. If an attribute Xi appears only once or by no means in the transactions Ti, then the rules which contain it in conclusion will have very weak supports and confidences tending towards 0.

Proposal 2

A rule is considered uninteresting when one or more of the following three conditions exist:

Data	Type	I/O	Assignation
K	Integer	Input	Current iteration number
Ck	Matrix of integer	Output	Candidates of size k
Fk	Matrix of integer	Output	Frequent itemsets of size k
Minsup	Real	Input	Minimum threshold of support
Minconf	Real	Input	Minimum threshold of confidence
T	Vector of integer	Output	Set of transactions
Support	Real	Output	Support=Freq(k-itemsets)/number of transactions
Confidence	Real	Output	Confidence=Support(k itemsets)/support(premise)

Table 6. OPTI2I: Annotations

ALGORITHM 1 : OPTI2I

Input : D – Database
Minsyp, Minconf real
K Integer
Output :
Ck – candidate.itemsets, Fk – frequent itemsets
Support, Confidence real
T – transactions (t → T)
Begin
K←1
Ck→candiadates 1-itemsets
Fk←ϕ
For Each transaction t ϵ D do
If Support.t.items ≥ Minsup

1. Xk (k = 1, 2, …, i, i+1,…, j) D appears only once or never appears in Ti⟹ ∩Ti= Ø or equal to the singleton.
2. Support (Xi → Xj) <minSup and Conf (Xi →Xj) < minConf
3. Support (Xj → Xi) <minSup and Conf (Xj → Xi) < minConf

Proposal 3 (Transitivity)

Let two set I = {I_1, I_2, …, I_t}, T = {T_1, T_2, …,T_n} included in the database K such that I T. If the association rules: $I_t \rightarrow I_{t+1}$ and $I_{t+1} \rightarrow I_{t+2}$ have partial supports equal to 1, then by the transitive relation the association rule $I_t \rightarrow I_{t+2}$ will have a partial support equal to 1.

3.5. PRESENTATION OF THE OPTI2I ALGORITHM

The notations used are presented in table 2 and the pseudo code in the algorithms named respectively OPTI2I and OPTI2I_GEN.

Then Fk ← Ck U Fk
Else remove Ck.t.itemsets
End if
OPTI2I_**Gen**(Fk)
Endfor
Sort in descending order t. fréquent itemsets D

{ Frequent 1-itemsets sorted }
{ Let's train frequent 2-itemsets }
K ← 2
If (t.itemsets = 1) and ((t+1).itemsets = 1)
Then rule (t.itemsets) → ((t+1).itemsets)← 1
Else if (t.itemsets = 1) and ((t+1).itemsets = 0) **or** (t.itemsets = 0) and ((t+1).itemsets = 1)
then rule (t.itemsets) → ((t+1).itemsets) ←0
{Transitivity}
else if (t.itemsets = 1) **and** ((t+1).itemsets = 1) **and** (t+2).itemsets = 1)
then rule (t.itemsets) → ((t+2).itemsets) ← 1
Endif
Endif
Endif
If Support(t.itemsets) ≥ Minsup
Then Fk←Ck←Fk
Else remove Ck.t.itemsets
Endif
OPTI2I_Gen(Fk)
Endfor

return ∪ kFk
End

ALGORITHM 2 : OPTI2I_Gen

Input : t.itemsets
Output : t.candidat, Ck
Begin
For Each pair of itemsets
candidat←t.itemsets ∪ (t+1).itemsets
Ck←Ck ∪ t t.candidate
Endfor
r**eturn** Ck
End

3.6. COMPARISON OF APRIORI AND OPTI2I ALGORITHMS

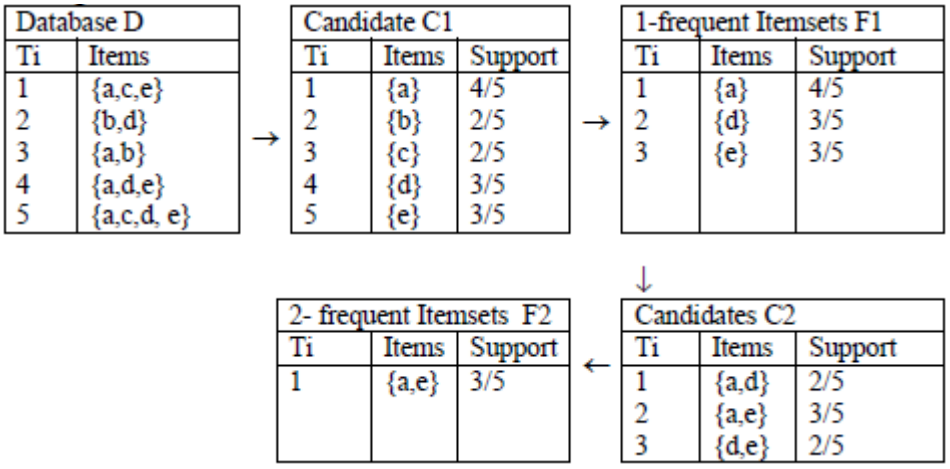
To theoretically compare our method with Apriori for k = 1 and k = 2, we introduce two examples. Set min_sup = 3.

Example 2

Let's go back to the database of example 1.
Let the database D

Tii	Items
1	{a,c,e}
2	{b,d}
3	{a,b}
4	{a,d,e}
5	{a,c,d, e}

- Algorithm APRIORI



Database D	
Ti	Items
1	{a,c,e}
2	{b,d}
3	{a,b}
4	{a,d,e}
5	{a,c,d, e}

Candidate C1		
Ti	Items	Support
1	{a}	4/5
2	{b}	2/5
3	{c}	2/5
4	{d}	3/5
5	{e}	3/5

1-frequent Itemsets F1		
Ti	Items	Support
1	{a}	4/5
2	{d}	3/5
3	{e}	3/5

Candidates C2		
Ti	Items	Support
1	{a,d}	2/5
2	{a,e}	3/5
3	{d,e}	2/5

2- frequent Itemsets F2		
Ti	Items	Support
1	{a,e}	3/5

Example 3

The data from 5 cash receipts we obtained from the supermarket CDCI Abidjan Yopougon (Ivory Coast):

Let D1 be the binary transaction database

$t_1 \rightarrow$	{ Jam of peach, Chocomax, Mayonnaise, Nuitella chocolate }
$t_2 \rightarrow$	{ Oil Aya Bottle, Toplait, Olgane Water, Tomato C. Alissa, Spaghetti }
$t_3 \rightarrow$	{ Mayonnaise, tomato C. alissa; Sunflower oil, Couscous }
$t_4 \rightarrow$	{Oil Aya bottle, Colgate, Spaghetti, Maggi Tablet, Tomato C. alissa }
$t_5 \rightarrow$	{ Spaghetti, Oil Aya bottle, Olgane water, Brown sugar, Toplait }

a→ Jam of peach	b→Chocomax	c →Mayonnaise	d→Nuitella chocolate
e→ Oil Aya Bottle	f→Toplait	g→ Tomato C.Alissa	h→Spaghetti
i→Sunflower oil	j→Couscous	k→ Colgate	l→Maggi Tablet
m→Olgane water	n→Brown sugar		

The following table represents the transaction is a list of items purchased by transaction database, where each one of the 5 customers in the supermarket:

Ti	a	b	c	d	e	f	g	h	i	j	k	l	m	n
t1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
t2	0	0	0	0	1	1	1	1	0	0	0	0	0	0
t3	0	0	1	0	0	0	1	0	1	1	0	0	0	0
t4	0	0	0	0	1	0	1	1	0	0	1	1	0	0
t5	0	0	0	0	1	1	0	1	0	0	0	0	1	1

Figure 2: list of products

Let the database D1

Ti	Items
1	{a,b,c,d}
2	{e,f,g,h}
3	{c,g,i,j}
4	{e,g,h,k,l}
5	{e,f,h,m,n}

Set min_sup = 2.

- Algorithm APRIORI

Database D

Ti	Items
1	{a,b,c,d}
2	{e,f,g,h}
3	{c,g,i,k}
4	{e,g,l,m}
5	{e,f,h,n,o}

→

Candidates C1

Ti	Items	Support
1	{a}	1/5
2	{b}	1/5
3	{c}	2/5
4	{d}	1/5
5	{e}	3/5
6	{f}	2/5
7	{g}	3/5
8	{h}	3/5
9	{i}	1/5
10	{j}	1/5
11	{k}	1/5
12	{l}	1/5
13	{m}	1/5
14	{n}	1/5

→

1-Frequent Itemsets F1

Ti	Items	Support
1	{c}	2/5
2	{e}	3/5
3	{f}	2/5
4	{g}	3/5
5	{h}	3/5

↓

Using the transitivity law without calculating the supports, we obtain the following 2-itemsets :

Candidates C2

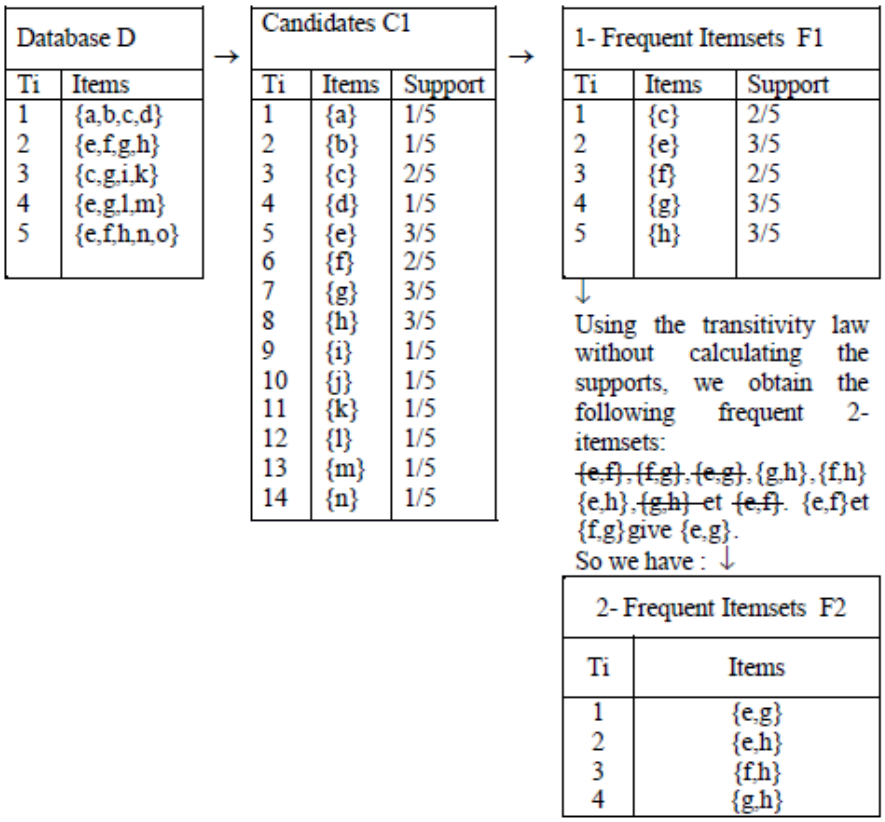
Ti	Items	Support
1	{c,e}	0
2	{c,f}	0
3	{c,g}	1/5
4	{c,h}	0
5	{e,f}	2/5
6	{e,g}	2/5
7	{e,h}	3/5
8	{f,g}	1/5
9	{f,h}	2/5
10	{g,h}	2/5

↓

2- Frequent Itemsets F2

Ti	Items	Support
1	{e,f}	2/5
2	{e,g}	2/5
3	{e,h}	3/5
4	{f,g}	1/5
5	{f,h}	2/5
6	{g,h}	2/5

- Algorithm OPTI2I



Database D

Ti	Items
1	{a,b,c,d}
2	{e,f,g,h}
3	{c,g,i,k}
4	{e,g,l,m}
5	{e,f,h,n,o}

→ Candidates C1

Ti	Items	Support
1	{a}	1/5
2	{b}	1/5
3	{c}	2/5
4	{d}	1/5
5	{e}	3/5
6	{f}	2/5
7	{g}	3/5
8	{h}	3/5
9	{i}	1/5
10	{j}	1/5
11	{k}	1/5
12	{l}	1/5
13	{m}	1/5
14	{n}	1/5

→ 1- Frequent Itemsets F1

Ti	Items	Support
1	{c}	2/5
2	{e}	3/5
3	{f}	2/5
4	{g}	3/5
5	{h}	3/5

↓

Using the transitivity law without calculating the supports, we obtain the following frequent 2-itemsets:
~~{e,f}~~,~~{f,g}~~,~~{e,g}~~,{g,h},{f,h} {e,h},~~{g,h}~~ et ~~{e,f}~~. {e,f}et {f,g}give {e,g}.
So we have : ↓

2- Frequent Itemsets F2

Ti	Items
1	{e,g}
2	{e,h}
3	{f,h}
4	{g,h}

Without accessing database D1 to calculate the supports of 2-itemsets, our algorithm uses the transitivity law of Proposition 3 to extract frequent 2-itemsets. In addition, our method also solves the problem of redundancy that APRIORI suffers. This allowed our algorithm to extract four (4) frequent 2-itemsets instead of six (6) as shown by the APRIORI method. Let's calculate the confidences of the 2-item association rules.

If we set min_conf = 50% we find that all the rules are valid. We show that the Apriori algorithm loses enough time to compute the association rules supports that will not actually be used by the end user. Compared to APRIORI our algorithm produces fewer two-item association rules because the redundant association rules produced in the APRIORI algorithm are automatically removed.

Note that during the second stage (k = 2) the search space is considerably reduced by the jumps made when there is a zero (0) between two item values.

For these association rules to be valid, they must meet the following three conditions:

1. Their supports ≥ minsup;
2. Their trusts ≥ minconf;
3. The rules are self-reciprocal, that is, trust (X→Y) = trust (Y→X)

Proposal 4

Let K be a database and two items X and Y of K. If trust (X→Y) equals confidence (Y→X) then the association rule X→Y is said to be self-reciprocal and support (X) equal support (Y).

Example: E→G is an auto-reciprocal association rule because Confidence (E→G) = confidence (G→E), but must also satisfy conditions 1 and 2 to be a valid rule.

3.7. SIDE-BY-SIDE CLASSIFICATION TECHNIQUE

After optimizing the extraction of frequent itemsets, we will proceed to their storage side by side. Thus items from association rules with the highest trusts will be ranked first and others will follow. If two rules of association have identical confidences, their premises will be placed one after the other and their consequences will follow immediately.
In the storage of items, we will retain only frequent itemsets.
Let's use the confidences found in Example

3. The association rules e→h and f→h have - If a_n-1 >= a_n

Item	a_1	a_2	a_3	a_4	a_5	a_6	a_7	…	a_n
Num q	0	1	2	3	4	5	6	…	q-n-1
Index k	1	2	3	4	5	6	7	…	q-n

q-k→q-k+1→ q-k+2→…→ q-k+m, k varies from 1 to n and m varies from 0 to n-1. The next Item is the index of the current Item incremented by one. From where The - If $a_{n-1} < a_n$

Item	a_n	a_{n-1}	a_{n-2}	a_{n-3}	a_{n-4}	a_{n-5}	a_{n-6}	…	a_1
Num q	q-n-1	q-n-2	q-n-3	q-n-4	q-n-5	q-n-6	q-n-7	…	0
Index k	q-n	q-n -1	q-n -2	q-n-3	q-n-4	q-n-5	q-n-6	…	1

q-k+m→……. → q-k+1→ q-k, k varies from 1 to n and m varies from 0 to n-1. The next Item is the index of the current Item incremented by one. From where The classification of items becomes:

$a_n \rightarrow a_{n-1} \rightarrow a_n\text{-}2 \rightarrow a_{n-3} \rightarrow a_n\text{-}4 \rightarrow a_{n-5} \rightarrow a_{n-6} \rightarrow \ldots \rightarrow a_1$

Example: Let 6 items a, b, c, d, e and d have their respective supports: supp (a) = 30%; supp (b) = 80%; supp (c) = 40%, supp (d) = 80%, supp (e) = 70% and supp (f) = 60% with a minimal support equal to 50%. The q identical confidences and they are the highest. Hence their items are classified as follows: e→f→h. Association rules e→g; g→h all have the same trusts. Since e and h are already classified, then it remains to classify g. Hence all items are classified as follows:

e → f → h→g.

- **Modeling of the technique**

The top support item is placed first and takes the number 0. The other items are numbered in descending order of their supports ranging from 1 to q-1. the Order Item k-1 is indexed k, where k is the position of the next item. Thus by a front chaining, we get all the items to be arranged side by side. q varies from 0 to n-1; k varies from 1 to n.

classification of items becomes:$a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow a_5 \rightarrow a_6 \rightarrow a_7 \rightarrow \rightarrow a_n$ numbers of the items (q varies from 0 to 5) are:

Items	a	b	c	d	e	f
Num q	0	1	2	3	4	5
Index k	1	2	3	4	5	6

The association rules discovered after the execution of the OPTI2i algorithm are as follows: b→d and e→f with a minimum confidence equal to 50%. The order of the indexes will be:

Items	b	d	e	f
Index k	4	5	6	X

The classification of articles becomes: b→d→e→f

4. SECTION 3

4.1 RESULTS

Our experiments are mainly focused on correlated data and weakly correlated data. Frequent itemsets generated are of sizes k = 1 and k = 2. We will limit ourselves to comparing OPTI2I with Apriori and Pascal, because the work of Yves Bastide published in the article "PASCAL: an algorithm for extracting frequent patterns", experiments have shown that the algorithm Pascal has optimized Apriori with times often better than Close algorithms (for frequently closed reasons), Max-miner (for maximal frequent reasons) and Apriori (for frequent reasons). Frequent itemsets and frequent motives mean the same thing.

To obtain the results of our work, we program with the PYTHON language, then with the generated data we use the Excel software to represent them graphically. The experiments were carried out on the following computer system:

- Core i3 2.4 GHZ processor
- RAM 4 GB
- 500 GB hard drive
- Windows 8.1 operating system
- Office 2013 Office Software

We used the following four datasets during these experiments:

- T20I6D100K and T25I20D100K, consisting of synthetic data constructed according to the properties of the sales data, which contains 100,000 objects with an average size of 20 items for an average size of the maximum potential frequent itemsets of 6 items.

- - C20D10K and C73D10K which are samples of the Public Use Micro data Samples file containing data from the 1990 Kansas Census. They consist of the 10,000 objects corresponding to the first 10,000 people, each object containing 20 attributes (20 items per object) and 386 items in total) for C20D10K and 73 attributes (73 items per object and 2,178 items in total) for C73D10K.

Case of weakly correlated data

- Dataset T20I6D100K

Support	Fréquents	OPTI2i	Pascal	Apriori
1	192	1,06	1,64	1,69
0,75	589	1,97	2,55	2,58
0,5	3 369	4,92	5,50	5,55
0,25	19 459	14,17	14,75	14,72

Table 6 : Response Time for T20I6D100K

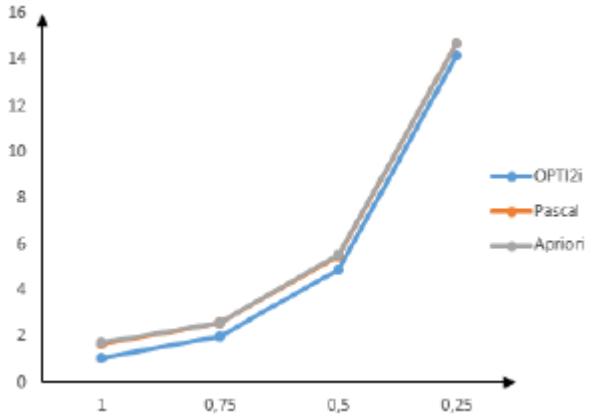


Figure 2 : Experimental results for T20I6D100K

– Dataset T25I20D100K

Support	Fréquents	OPTI2i	Pascal	Apriori
1	73	0,06	0,64	0,72
0,75	144	0,64	1,22	1,39
0,5	159 907	120,50	121,08	116,89

Table 7 : Response Time for T25I20D100K

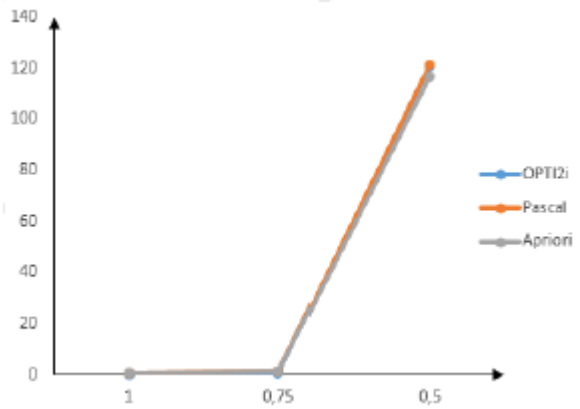


Figure 3 : Experimental results for T25I20D100K

Correlated data

- **Dataset C20D10K**

Support	Fréquents	OPTI2i	Pascal	Apriori
20	2 530	3,38	1,18	7,14
15	4 545	5,17	1,54	10,67
10	11 235	12,83	2,41	20,60
7,5	19 145	14,36	2,94	29,05
5	44 076	22,55	4,13	49,42
2,5	145 045	35,34	6,92	94,33

Tableau 8 : Response Time for C20D10K

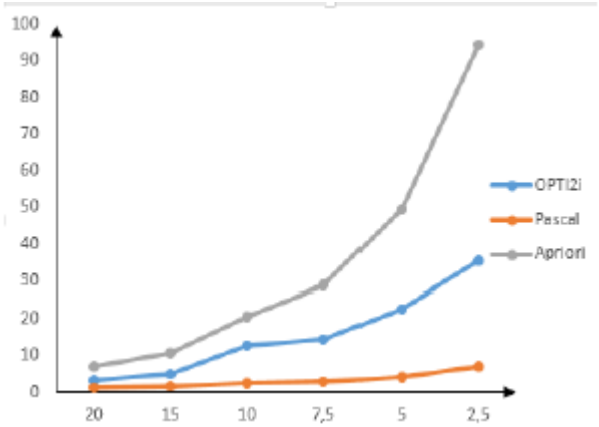


Figure 4 : Experimental results for C20D10K

- **Dataset C73D10K**

Support	Frequents	OPTI2i	Pascal	Apriori
80	13 645	54,61	22,19	457,66
75	29 409	126,52	49,10	956,70
70	71 511	193,73	98,31	ρ 183,14
60	544 443	724,93	496,51	13 650,50

Table 9 : Response Time for C73D10K

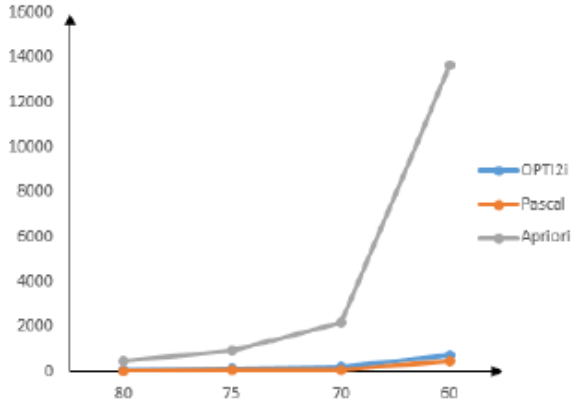


Figure 5 : Experimental results for C73D10K

The response times correspond to the frequent itemset extraction times for Apriori and the frequent itemset extraction times for the Pascal and OPTI2i algorithms. The extraction times of association rules with Apriori and bases for association rules with Pascal and A-Close for datasets T20I6D100K, T25I20D100K, C20D10K and C73D10K are shown in Figures 2, 3, 4 and 5. The number of generated itemsets is a function of the different ¼, ½, ¾ and 1 minpports for the T20I6D100K and T25I20D100K datasets. For the C20D10K data set, the number of generated itemsets is a function of the different minsupport ranging between 2.5 and 7.5 with a step of 2.5 and between 10 and 20 with a step of 5. For C73D10K, the number of itemsets generated is function Different sales data between 60 and 80 the sales data are sparse and weakly correlated, for these executions all frequent itemsets are frequent itemsets, which corresponds to the worst case for Apriori which performs more operations than Pascal and OPTI2i so to extract frequent itemsets.

Despite these differences, the execution times of the three algorithms, which vary from a few seconds to a few minutes, remain acceptable in all cases. For C20D10K and C73D10K the execution times of OPTI2i are much lower than those of Pascal and Apriori. These results are explained by the data characteristics of these games which are correlated and dense.

As a result, the number of frequent itemsets is important. The search space of the OPTI2i and Pascal algorithms is much smaller than the search space of the Apriori algorithm. In contrast to the response times obtained for T20I6D100K and T25I20D100K, the differences in response times between OPTI2i and Pascal are measured in minutes or tens of minutes for C20D10K, and in tens of minutes or hours for C73D10K. Moreover, Pascal and OPTI2i could not be executed for support thresholds lower than 70% on the C73D10K set and also the three Apriori, Pascal and OPTI2i algorithms could not be executed for support thresholds lower than 0.75% on the game T25I20D100K.

In Figure. 3, the response times of the three algorithms are almost identical. But the response time of Pascal is slightly higher than the other two algorithms for minimal supports less than 0.75%.
It is clear for the games T20I6D100K and T25I20D100K that the response times of our algorithm OPTI2i are better at the response times of Pascal and Apriori algorithms

On the correlated data OPTI2I gives higher response times to the Pascal algorithm, but lower than the Apriori algorithm. Let TOPTI2I be the response time of the OPTI2I algorithm; Tapriori - Response time of the Apriori and Tpascal algorithm - Response time of the Pascal algorithm.

TPASCAL ≤ TYELLI ≤ TAPRIORI for games C20D10K and C73D10K

5. SECTION 4

5.1. DISCUSSION

In our paper, we took the particular case of the marketing field of the "basket of the housewife". In addition, this could also apply to other areas such as medicine and the WEB, but in different contexts. We have located our context of work in the context of underdeveloped countries whose management of large sales areas is lagging behind that we should try to correct with the tools of data mining, including the technique of association rules that we have chosen. Because marketing tools show their limit when it comes to optimizing the management of large retail stores. Our approach can also be used to improve the management systems of Western supermarkets.

In our state of the art, we presented four categories of association rule algorithms based on frequent, maximal, closed and hybrid itemsets on strongly and weakly correlated data. Since the sets of frequent itemsets have weaker supports than the subsets of frequent itemsets, the probability of finding interesting association rules is higher with the reduced item sets. This reason guided our choice to work with itemsets of sizes k = 1 and k = 2. 1-itemsets cannot be rules, but 2-itemsets can produce very interesting and relevant association rules.

Figures 2, 3, 4 and 5 show the experimental results obtained by the application of our approach with the extraction algorithms like Apriori and Pascal. We observe in these figures that the extraction time of frequent 2-itemsets is better in our method than the Apriori and Pascal methods, only for scattered data (i.e uncorrelated).

The drop in time, although somewhat insignificant, is related to the fact that in a base where data is sparse, the binary zero (0) rate is high, which our algorithm automatically deletes. This in turn leads to the automatic pruning of the corresponding items. The law of transitivity evoked in our proposition 3 makes it possible to extract the items without access to the database. Through our results, our method has led to a time of extraction of frequent 2-itemsets much reduced in the case of a context of extraction of sparse data. If we apply the PASCAL algorithm to the iterations k + 2 (where k varies from 1 to n-2), the overall time of extraction of frequent k-itemsets will be better than with the other algorithms. For the weakly correlated data, the results show that, the more the supports tend towards 0%, the more the number of 2-itemsets is high, and the extraction time of these 2-itemsets is also relatively high. Despite the fact that our method does not give better results compared to PASCAL in the case of highly correlated data (Figure 4), it gives much more satisfactory results than the APRIORI reference algorithm. The conclusion made in FIG. 4 finally shows that OPTI2I is more efficient than APRIORI on all the types of data sets used, as is particularly demonstrated by the results of FIGS. 2, 3 and 4. The weakness of our algorithm compared to PASCAL, in the case of strongly correlated data is related to the rarity of 0 binaries, because of the density of the extraction contexts C20D10K and C73D10K.

6. CONCLUSION AND PERSPECTIVES

In this paper, we have proposed a new optimization approach for extracting frequent 2-itemsets. It allowed us to improve the time obtained in the previous work of discovery of even sets (2 items) on weakly correlated data. Our method is certainly interesting, because it is original, but it is limited in a very specific context for a need specifically related to putting side by side items of a large commercial area. A first perspective of the next work concerns the optimization of cardinal itemets extraction superior to 2 from our results and thus to try to show by our hypothesis that our method can improve the overall extraction time of frequent k-itemsets.

Another perspective might consider extending our experiments to dense data (strongly correlated data). A last perspective could concern the improvement of our algorithm in order to obtain a shorter time of extraction of 2-itemsets. In addition, regarding the classification of articles side by side, we plan to use another technique based on the method of triangular inequality.

REFERENCES

[1] R. Agrawal, R. Srikant, H, "Fast algorithms for mining association rules in large databases",Research Report RJ 9839, IBM Almaden Research Center, San Jose, California,June 1994.

[2] J. Azé. Extraction of knowledge from digital and textual data. Ph.D. thesis, Paris-Sud University, december 2003.

[3] C.-C. Yu and Y.-L. Chen: Mining sequential patterns from multidimensional sequence data. IEEE Transactions on Knowledge and Data Engineering, 17 (1): 136-140, 2005.

[4] Martine CADOT. Extract and validate complex relationships in human sciences: statistics, reasons and rules of association, PhD thesis of the University of Franche-Comté - Besançon December 12, 2006.

[5] G. Dong Z. Xing, J. Pei and P. Yu Yu: Mining Sequence Classifiers for Early Prediction. In Proceedings of the 2008 SIAM International Conference on Data Mining (SDM'08), Atlanta, GA, April 24-26, 2008.

[6] Ming-Chang Lee. Data mining - R - association rules and apriori algorithm 29 March 2009

[7] Yves Bastide et al. Pascal: an algorithm for extracting frequent motifs, Ph.D. article, IRISA-INRIA - 35042 Rennes Cedex, April 26, 2010.

[8] M Ramakrishna Murty, Murthy JVR, Prasada Reddy PVGD, Suresh Satapathy, "International Information Systems Design and Intelligent Application-2012, Springer-AISC" (indexed by SCOPUS) etc.), ISBN 978- 3-642-27443-5, Vol: 132, 2012, PP: 445-454. "

[9] Sadok Ben Yahia-Engelbert Mephu Nguifo. Association rule extraction approaches based on Galois matching. Lens Computer Research Center - IUT de Lens University Street SP 16, F-62307 Lens cedex

[10] Thierry Lecroq. Extraction of rules of association, University of Rouen France N. Pasquier Y. Bastide R. Taouil and L. Lakhal. Pruning closed items and lattices for association rules - 2016.

[11] Mostafa El Habib Daho and Al..Dynamic Pruning for Tree-based Ensembles. Pages 261. Year 2016.

[12] Professor Brou Konan Marcellin. Course material entitled "Chapter 2: Rules of Association" INPHB, 2015 - 2016.

[13] Abdel-Basset, M., El-Shahat, D. and Mirjalili, S. (2018). A hybrid whale optimization algorithm based on a local search strategy for the permutation flow store scheduling problem. Future Generation Computer Systems, 85, 129-145.

[14] Abdel-Basset, M., M., G., Abdel-Fatah, L., and Mirjalili, S. (2018). An improved meta-heuristic algorithm inspired by nature for 1-D bin packing problems. Personal and ubiquitous computing, 1-16.

[15] Abdel-Basset, M., M., G., A. Gamal and F. Smarandache (2018). A hybrid approach of neutrosophic sets and the DEMATEL method to develop supplier selection criteria. Automation of design for embedded systems, 1-22.

[16] M., G, M., Mohamed, M., and Smarandache, F. A new method to solve the problems of linear neutrino programming. Neural computation and applications, 1-11.

[17] M., M., G., Fakhry, A.E. and El-Henawy, I. (2018). 2 levels of grouping strategy to detect and locate copy transfer forgery in digital images. Multimedia Tools and Applications, 1-19.

[18] Abdel M., & Mohamed, M. (2018). Internet of Things (IoT) and its impact on the supply chain: a framework for creating intelligent, secure and efficient systems. Next-generation computer systems.

[19] Basset, M., G. Manogaran, M. Mohamed, and Rushdy, E. Internet of Things in an Intelligent Educational Environment: Supporting Framework in the Decision-Making Process. Competition and calculation: practice and experience, e4515.

[20] Abdel, M., M., G., Rashad, H., and Zaied, A.N.H. (2018). A comprehensive review of the quadratic assignment problem: variants, hybrids, and applications. Journal of Ambient Intelligence and Humanized Computing, 1-24.

[21] Ab, M., G, M., Mohamed, M. and Chilamkurti, N. (2018). Three-way decisions based on neutrosophic sets and the AHP-QFD framework for the supplier selection problem. Next-generation computer systems.