Efficient Mining of Association Rules based on Clustering from Distributed Data

Marwa Bouraoui[1], Amel Grissa Touzi[2]
Signal, Image and Technology of Information Laboratory
National Engineering School of Tunis, Tunis El Manar University
Tunis, Tunisia

***Abstract*—Data analysis techniques need to be improved to allow the processing of data. One of the most commonly used techniques is the Association Rule Mining. These rules are used to detect facts that often occur together within a dataset. Unfortunately, existing methods generate a large number of association rules, without accentuation on the relevance and utility of these rules, and hence, complicating the results interpretation task. In this paper, we propose a new approach for mining association rules with an emphasis on easiness of assimilation and exploitation of the carried knowledge. Our approach addresses these shortcomings, while efficiently and intelligently minimizing the rules size. In fact, we propose to optimize the size of the extraction contexts taking advantages of the Clustering techniques. We then extract frequent itemsets and rules in the form of Meta-itemsets and Meta-rules, respectively. Experiments on benchmarking datasets show that our approach leads to a significant reduction of the number of generated rules thereby speeding up the execution time.**

Keywords—Distributed data; association rules mining; clustering; meta-items; meta-rules

I. Introduction

Association rules mining has become one of the core data mining tasks with many real world applications such as selective marketing, fraud detection in web, economic census, and several other applications. It aims to discover associations among transactions encoded in a database. An association rule is a probabilistic rule which implies certain association relationship among a set of objects in the form of "if-then" statement. Association rules mining was introduced by Agrawal et al. [1] and they have described the formal model of this problem as follows. Let I = {i1, i2... in} be a finite set of items. Let D = {T1, T2... Tn} be a finite set of transactions, each transaction Ti consists of a set of items where Ti ⊂ I. Let X be a subset of I. An association rule is a conditional implication of the form r: $X \rightarrow Y$ between two itemsets X, Y ⊂ I where $X \cap Y = \emptyset$. The basic algorithms for mining association rules are Apriori [2], FP-Growth [3].

Nowadays, most enterprises collect huge amounts of business data from daily transactions and store them in distributed datasets; especially, for security issues and communication overhead. Those distributed datasets are usually not allowed to be transmitted or joined together. Mining association rules from such data has attracted a lot of attention in recent data mining research. Several classes of parallel and distributed algorithms have been developed in this context [4-11]. Most of them compete in domains like the result accuracy, the execution time, memory consumption, communication cost and so forth. Yet, little attention has been paid to the readability of the outputted results. Indeed, as the dataset size grows, mining association rules activity tends, potentially, to generate a prohibitively large number of rules. Unfortunately, a likewise output adds only inconvenience to data exploitation task from mined rules rely heavily on human interpretation in order to infer their semantic meanings.

To overcome these shortcomings, the solution we consider is to combine clustering and association rules mining technologies, to efficiently mine rules from large distributed data. Indeed, clustering technology helps, inherently, reducing the data context size, bringing into play its segmentation property. We propose a new approach, Clustering based Distributed Association Rules Mining Algorithm (C-DARM), which continues to extract rules from business data, but avoid rendering irrelevant and extensive number of results. More specifically, our aim is refining the output for a better understanding, and an uncomplicated interpretation of the carried knowledge. To do this task efficiently, we propose to introduce a pre-processing step based on clustering to optimize the size of the remote extraction contexts. The main idea is to mine distributed frequent itemsets from a representative set consisting of a collection of classes, called Meta-Itemsets, we then mine rules in the form of Meta Association Rules. In global, our solution provides:

- An efficient distributed process for mining rules from initially distributed data.
- A reduced number of rules when dealing with large datasets while ensuring no loss of information.
- A global view of the rules which boosts the result assimilation and interpretation.
- A pre-processing strategy to exclude not interesting attributes according to type of data and user requirements.

The rest of this paper is organized as follows: Section 2 provides an overview of the common distributed algorithms for mining association rules. Section 3 details our new proposed approach. Performance analysis and experimental results are shown in Section 4. We finally conclude our paper and present our ideas for future work.

II. Literature Review

Apriori is one of the most versatile and effective algorithm for frequent itemsets mining. Various improvements of this algorithm have been proposed to solve several issues such as minimizing database scanning, reducing transaction base size, proposing effective pruning techniques and efficient data structures. We suggest AprioriTid [11], DHP [12], Partition [13], Sampling [14] and DIC [15] for a further detailed explanation of the cited points.

An alternative algorithmic scheme works by mining closed frequent itemsets in the first step. A set is closed if it has no superset with the same frequency. The notion of closed itemsets is strongly connected to the Formal Concept Analysis (FCA) [16] field. The FCA offers a condensed and concise representation allowing the deduction of association rules bearing information [17]. The state-of-the-art algorithm for mining closed frequent itemsets is Close [18]. Others algorithms have been introduced which offer various improvements of this one. We can cite A-Close [19], Close + [20], Charm [21], Titanic [22, 23], and Closet [24].

Modern organizations are geographically distributed and, typically, each site locally stores its amount of data. Challenges are raised due to this data diffusion over nodes, and therefore, researches have been initiated in parallel and distributed algorithms in general, and in mining association rules, in particular. The suggested solutions take advantage of the improvement made on processor speed and network technologies.

The first proposed algorithms are CD (Count Distribution) [25] and DD (Data Distribution) [26] which are, basically, an Apriori parallelization. CD and DD are based on data parallelism and task parallelism, respectively.

In order to reduce the CD communication overhead, the NPA algorithm [27] is suggested. Unlike CD, the computing phase of global supports takes place on a master site, and eventually avoids a redundant and overload calculation. NPA minimizes then the communication cost, reducing it from the order of O (| Ck | * n ^ 2), for the CD algorithm, to the order of O (| Ck | * n) for the NPA algorithm, where | Ck | is the candidates sum of size k, and n is the number of sites.

Based on CD, FDM (Fast Distributed Mining) algorithm was proposed in [28]. It reduces the size of generated candidates, introducing new pruning techniques namely the local pruning and the global pruning.

Another Apriori-based algorithm was presented, the Optimized Distributed Association Rules Mining (ODAM) [6] that derives from FDM and CD as well. It essentially removes infrequent 1-itemsets, merges identical transactions into a single one, and then inserts new transactions into memory. It outperforms CD and FDM because of its downsizing property. Furthermore, it reduces the total message exchange count and the communication cost by forwarding support counts of candidate itemsets to a single site, called “receiver”.

In [7] the Efficient Distributed Frequent Itemset Mining (EDFIM) algorithm is implemented which is an extension of the ODAM algorithm. It performs infrequent itemsets pruning and duplicate transactions merging operations, after every pass. It reduces, therefore, the size of transactions and subsequently the scan iterations count. EDFIM uses local and global pruning actions, and a merger site to reduce the communication overhead.

Here [5], an Apriori-TID based algorithm was proposed for mining distributed association rules, the Distributed Parallel Apriori algorithm (DPA). DPA uses diverse rule interesting measures such that Pearson coefficient, Chi square, etc. It provides also a technique for a faster rate for mining frequent itemsets by handling their sparse matrix.

In [4], Lin and Chung introduced the FLR-mining algorithm. It iteratively estimates the workload based on the number of header items. It is qualified to determinate the number of the carried computing nodes automatically and achieving better load balancing as compared with existing methods.

In [8], the Parallel FP-Growth is implemented. It examines the challenges of the parallelization process and a method to balance the execution efficiently on shared-nothing architecture. In the first place, it evaluates the horizontal subset of data. Then, it parallel constructs local FP-Tree. Finally, mining procedure took place on this FP-Tree.

In [29], the LMatrix algorithm is presented. It minimizes the number of database scans by generating a matrix that models local transactions, from which it calculate the local supports at each iteration.

The IDD (Intelligent Data Distribution) algorithm [30] is an enhanced version of the DD algorithm. To improve performance, local transactions are connected by ring structure rather than all-to-all broadcast, which decrease the communication cost.

Backed by CD and IDD algorithms, HD (Hybrid Distribution) [31] is suggested as a combination of the two. The processors are divided into groups and CD is applied considering one group as one processor.

In [32], ParEclat, ParMaxEclat, ParClique, ParMaxClique are proposed which are parallel versions of Eclat, MaxEclat, Clique and MaxClique algorithms, respectively. Each algorithm consists of three phases; preparation, asynchronous processing, and reduction. Data partition and computation are executed in the preparation phase. Next, each processor generates local frequent itemsets asynchronously. In the final phase, ultimate results are combined together.

These briefly introduced algorithms can be further classified by the following attributes:

- Base Algorithms (Basic sequential algorithm)
- Parallelism (Data, Task, Hybrid)
- Load Balancing (Static, Dynamic, Hybrid)
- Database Layout (Horizontal, Vertical)
- Database Partition (Partitioned, Replicated, Shared)
- Memory System (Distributed, Shared, Hierarchical)

TABLE I. CLASSIFICATION OF PARALLEL AND DISTRIBUTED ALGORITHMS OF MINING ASSOCIATION RULES

Algorithm	Base Algorithm	Parallelism	Load Balancing	Database Layout	Database Partition	Memory System
CD, FDM, NPA	Apriori	Data	Static	Horizontal	Partitioned	Distributed
DD, IDD	Apriori	Task	Static	Horizontal	Partitioned	Distributed
HD	Apriori	Hybrid	Hybrid	Horizontal	Partitioned	Distributed
CD, IDD	Apriori	Data	Static	Horizontal	Partitioned	Distributed
ODAM, EDFIM, LMatrix	Apriori	Data	Static	Horizontal	Partitioned	Distributed
PCCD	Apriori	Task	Static	Horizontal	Partitioned	Shared
Parallel FP-Growth	FP-Growth	Task	Static	Horizontal	Partitioned	Hierarchical
ParEclat, ParMaxEclat, ParClique, ParMaxClique	Eclat, Clique	Task	Static	Vertical	Replicated	Hierarchical

We can summarize the above characteristics in the form of a Table I.

Recently, new approaches involved clustering methods in the association rules mining field. The objective of clustering is to split the heterogeneous set of objects into a number of homogeneous subsets having a similar behavior, called clusters. The similarity of objects is generally measured in terms of geometric distance between objects. The distance function differs depending on the nature of the data.

As clustering breaks up a dataset based on item similarities, it guarantee better semantic results when dealing with association rules. In [9], authors proposed to mine rules based on clustering and soft sets. The idea is to apply the CFSFDP clustering technique to classify the transactions. From the resulting clusters, supports are obtained by considering logical formulas over the soft sets. However, this algorithm mines rules considering items belonging to the same cluster, and consequently misses cross clusters rules. In [10], clustering technique was combined with association rule mining to speed up the extraction of conceptual association rules. Conceptual association rules imply the relationships between concepts generated using Formal Concept Analysis. Unfortunately, the construction of concepts is a too heavy task and it can’t suit in memory when dealing with a large dataset.

Note that all these approaches that capitalize on clustering techniques to mine association rules 1) are applied in the sequential environments; 2) lead to an important loss of information; and 3) suffer from high computational cost when dealing with large datasets.

III. MOTIVATION

Nowadays, the volume of data generated in the web, transactional systems, and different other areas continue to increase explosively. Thence, a major problem of association rules discovery methods becomes the large number of results it tend to generate, even for a reasonable size of the extraction context. Above all, when the support threshold drops low, the number of resulted patterns goes up dramatically. Consequently:

- The number of generated rules is very large, so it is difficult to understand and interpret by the user.
- The management of structures for data modeling requires high execution time.
- Generated rules from data are usually redundant.
- The definition of data and data structures requires a large memory space because of the complex modeling algorithms such as trees or graphs.

It emerges from this ascertainment the importance of investigating efficient methods for distributed mining of association rules. In particular, it is crucial that outputted rules are understandable and useful to the user. When mining association rules from this sort of data, we may find thousands of rules. Moreover, the effectiveness degrades because it generates numerous redundant patterns. We may cite a trivial but illustrative example here, for a database having a transaction of length k, it will generate 2^k–1 frequent itemsets and even a large number of useless association rules.

In the other hand, clustering technology helps, inherently, reducing the data context size, bringing into play its segmentation property. This data input transformation discloses some interesting relationships between transactions and proposes a useful starting point for other purposes, such as pattern mining, in our case.

The solution we consider is to combine clustering and association rules mining technologies, to efficiently mine rules from large distributed databases. Our aim is to reduce the large number of association rules that are typically computed by existing algorithms, thereby rendering the emerging rules much easier to interpret and visualize.

Our approach takes place in three phases:

- *A data pre-processing phase based on clustering:* it consists on organizing data into groups (using a fuzzy clustering algorithm). Then, to generate, from these classes, a new, more condensed representation of the extraction context in the form of a Cluster-Fuzzy Formal Context. A cleaning step is thus necessary to optimize the size of the extraction contexts by filtering up the unnecessary data. The resultant classes (Meta-Itemset) are therefore used as starting points for the mining of distributed frequent itemsets.

To accomplish this phase, we propose two new concepts:

1) A new representation and definition of itemset: Meta-Itemset

2) A new representation and definition of the extraction context: Cluster-Fuzzy Formal Context

- *Distributed frequent itemsets mining phase:* it consists on mining frequent itemsets as a set of frequent Meta-Itemsets from the remote Cluster-Fuzzy Formal Contexts. This process is distributed and covers synchronization between local sites and a master site.
- *Generation of association rules phase:* it consists on mining association rules in the form of Meta-Association Rules by applying a rule generation algorithm on the obtained frequent Meta-Itemsets.

IV. New Approach

In this section, we detail our new approach for mining association rules from distributed data.

A. General Principle

Our contribution in mining association rules is to refine the results for a better understanding, and an uncomplicated interpretation of the carried knowledge, while efficiently and intelligently minimizing the rules size. To meet this challenge, we propose to combine clustering and association rules mining technologies. Thus, we introduce a pre-processing step based on clustering to optimize the size of the remote extraction contexts. The main idea is to mine distributed frequent itemsets from a representative set consisting of a collection of classes, called Meta-Itemsets. Hence, a clustering algorithm is applied to organize the data into classes. We generate, from these classes, a new representation more condensed of the extraction context in the form of a Cluster-Fuzzy Formal Context. A cleaning step is then carried out to remove the non-interesting Meta-Itemsets. From these new remote optimized contexts, we mine frequent Meta-Itemsets through a distributed process that deal with synchronization between remote sites and a receiver site. We finally generate a set of association rules in the form of Meta Association Rules. The number of these rules is much fewer than the number of rules generated by the classical association rules mining algorithms.

For a further explanation, we illustrate by the next figure (Fig. 1) the general process of our approach and with Fig. 2, the global one.

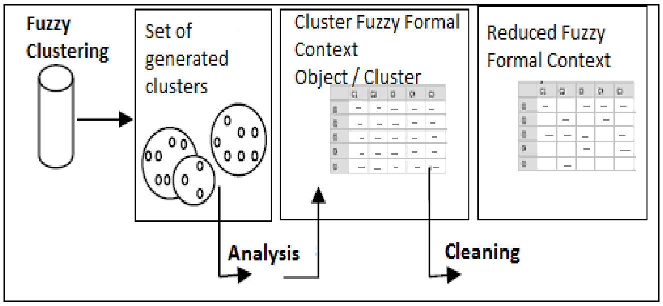

Fig. 1. Process in a Specific Site.

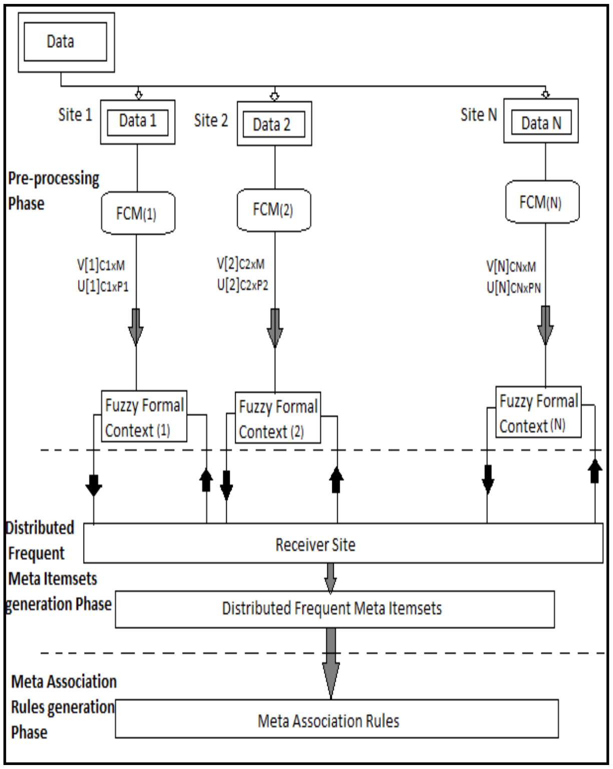

Fig. 2. Global Process.

We present in the following the general principle of our approach:

Input: N sites, D [n] (n = 1..N) a set of distributed data through N sites, s local minSupp, S global minSupp, C number of clusters, α accuracy
Step 1: Iterative Pre-processing Phase based on Clustering
For each site (i = 1, i ⩽ N, i ++)
Apply a fuzzy clustering algorithm to organize the data into different groups (or clusters). The output is a membership matrix mapping objects to clusters.
Construct the Cluster-Fuzzy Formal Context (Object / Cluster) from the obtained matrix.
Reduce the Cluster-Fuzzy Formal Context according to the accuracy parameter (a cleaning step).
End For
Step 2: Distributed Frequents Meta Itemsets Mining Phase
On the obtained Cluster-Fuzzy Formal Contexts, apply a distributed process to generate the distributed frequent Meta Itemsets.
Step 3: Meta Association Rules Generation Phase
Generate association rules in the form of rules between clusters, called Meta-Rules.
Output: A set of Meta-Association Rules

B. Process of Mining Association Rules

Our process takes place in three main phases, namely the iterative pre-processing phase based on clustering, mining distributed frequent Meta-Itemsets phase and the generation of the Meta Association Rules phase. In the following, we detail these phases.

1) *Iterative pre-processing phase based on clustering:* This phase depicts the clustering-based pre-processing process. Through this phase, we aim to optimize the size of itemsets as well as the extraction contexts, which are used later as starting points for mining distributed frequent itemsets. This process is iterative and runs through all sites.

To express our process, we assume that the data D is horizontally distributed through N sites. Each site D [n] (n = 1...N) is composed by Pn objects, where each one is described by a vector dn [j] (j = 1...Pn). On each site, we apply locally a fuzzy clustering algorithm such as the FCM algorithm [25]. We get Cn fuzzy clusters where Cn[i] (i = 1... Cn) is the ith cluster in the nth site. Each cluster is therefore characterized by its center Vi [n] ∈ RM, that we call a prototype. Each object dn [j] ∈ D [n] (n = 1...N) is characterized by a membership degree to each cluster. For each site D [n], U [n] = [μij [n]] Cn × Pn is the fuzzy partition, where Cn is the number of clusters, Pn is the number of objects belonging to this site, and μij is the membership degree of the ith object to the jth cluster. In order to enlighten these step proceedings, we define, next, two new concepts:

Definition: Meta-Itemset

A Meta-Item is a resulting class of the clustering process on the database. A Meta-Item is represented by its center. A Meta-Itemsets is a set of Meta-Items. A k-Meta-Itemset is a Meta-Itemset that is formed by k Meta-Items.

Definition: Cluster-Fuzzy Formal Context

In Cluster-Fuzzy Formal Context for the site D[n] (n=1..N), we link the objects to the clusters by the means of a relation that models the belonging relationship. The Cluster-Fuzzy Formal Context in the site D[n] (n=1..N) represents a triple (X[n], V[n], I[n]) where X[n] = dj = {dj : j = 1.. Pn) ∈ RM represents the set of objects, V[n]= Vi[n] = {cj : j = 1..Cn) ∈ RM represents the centers of the clusters, and I[n] represents the membership degree of X[n] to V[n].

The figure below (Fig. 3) is an illustration of the pre-processing phase:

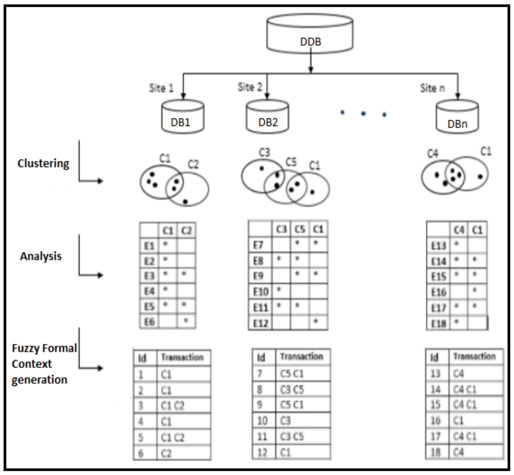

Fig. 3. Pre-Processing Phase.

Example:

Through this example, we explain the pre-processing step which takes place in a specific site. Let E1, E2, E3, E4, E5, E6, E7, E8 be a set of student markets in the following modules: Database (DB), Object Oriented Programming (OOP), Operating System (OS), Artificial Intelligence (AI) and Signal Processing (SP), as shown in the Table II.

We apply a fuzzy classification algorithm (FCM) on these data and obtain the membership matrix, mapping objects to clusters (Table III).

At this stage, we perform the polish step on the obtained matrix. We then apply a filtering step to exclude non-interesting entries to optimize the size of the final extraction context. We introduce a parameter of precision α, establishing an edge, below which, the belonging of an object to a class is considered insignificant. The choice of then α parameter depends on the nature of the data and the user requirements. In our example we fixed α = 0.3 (Table IV).

TABLE II. SAMPLE OF STUDENTS MARKETS

	BD	**OOP**	**OS**	**AI**	**SP**
E1	13	14	8	9	9
E2	15	12	1	11	6
E3	6	9	13	11	7
E4	10	6	17	14	13
E5	11	12	9	9	9
E6	16	5	13	14	13
E7	8	9	11	10	9

TABLE III. CLUSTERING RESULTS

	C1	**C2**	**C3**
E1	0.095	0.710	0.210
E2	0.049	0.800	0.061
E3	0.072	0.100	0.830
E4	0.836	0.072	0.110
E5	0.091	0.550	0.370
E6	0.820	0.110	0.083
E7	0.037	0.067	0.899

TABLE IV. NEW CONTEXT

	C1	**C2**	**C3**
E1	-	0.710	-
E2	-	0.800	-
E3	-	-	0.830
E4	0.836	-	-
E5	-	0.550	0.370
E6	0.820	-	-
E7	-	-	0.899

2) Distributed frequent meta-itemsets mining phase: This phase consists of mining distributed frequent Meta-Itemsets from the obtained Cluster-Fuzzy Formal Contexts. This process is a distributed that deal with synchronization between remotes sites and a receiver site. At each iteration, we calculate the local supports of candidates and, hence, select the locally frequent ones. Then, sites forward locally frequent Meta-Itemsets, along to the receiving site, for further processing. In the master site, the calculation of the Global Supports task is executed, and subsequently the deduction of globally frequent k-Meta-Itemsets. From these selected sets, we generate the candidate k+1-Meta-Itemsets, travelling back to local sites as an input for the next iteration. This process repeats until no other candidate can be generated and, subsequently, generate the ultimate global frequent Meta-Itemsets. Intervening a master site reduces the communication cost from the order of O (| Ck | * n ^ 2) to the order of O (| Ck | N) where | Ck | is the candidate size and n is the site number. For clarity, used notations are listed in Table V.

Definition: Locally and Globally frequent Meta-Itemset

A Meta-Itemset I is called locally frequent if s ⩾ minSupp. A Meta-Itemset I is called globally frequent if S ⩾ minSupp * T (T is the number of transactions).

Process is described below:

```
Input: N sites, N Cluster-Fuzzy Formal Contexts, s local minSupp, S global minSupp.
k =1
Repeat
For each site (i = 1, i ⩽N, i ++)
Local Sites:
Calculate the local supports Li(k) of the k-Meta-Items.
Remove the infrequent ones and then get the locally frequent k-Meta-Itemsets LLi(k).
Diffuse locally frequent Meta-Itemsets LLi(k) to the receiver site, along with supports of both frequent and infrequent k-Meta-Itemset.
End For
Receiver site:
Receive locally frequent k-Meta-Itemsets, calculate the supports sum of homologous, and thus deduce the global support G(k) of the k-Meta-Itemsets.
Remove the infrequent ones and then get the globally frequent k-Meta-Itemsets GLL(k).
Generate the candidate set GA(k+1) from GLL(k).
Disseminate the remaining candidates to local sites.
k +1
Until to find all frequent Meta-Itemsets.
Output: Set of frequent Meta-Itemsets
```

3) Meta association rules mining phase: This phase consists in applying an algorithm for generating association rules from the frequent Meta-Itemsets resulting from the previous steps. The output is a set of association rules in the form of Meta Association Rules.

TABLE V. NEW CONTEXT

Notation	Meaning
N	Number of sites
minSupp	Support threshold
s	Local minSupp
S	Global minSupp
Li(k)	Local supports of the k-Meta-Itemsets in site i
LLi(k)	Locally frequent k-Meta-Itemsets in site i
G(k)	Global supports of the k-Meta-Itemsets
GLL(k)	Globally frequent k-Meta-Itemsets
GA(k)	Candidate set generated from GLL(k-1)

Lemma

The Meta Association Rules obtained by running our distributed process on the distributed data obtained from Phase 1, are the same as if the process is applied centrally on each site. This validates the accuracy and compliance of the generated rules.

Theorem 1

Let Ci [1], Cj [2] be two clusters generated by a fuzzy clustering algorithm in the D [i] and D[j] sites, respectively. The meta-rule Ci [1] => Cj [2] with a coefficient (EF) is denoted by Ci[1] => Cnj[2] (CF) where EF = Sum ([µij [n]]Cn ×Pn) ; µij [n] ∈ [0, 1]; ∀ i ∈ {1..Pn}; ∀ j ∈ {1,2} ∀ n ∈ {1..N}.

The value EF is in the range] 0...1]. It is called the Exactitude Factor of this meta-rule. This value indicates the importance degree of this Meta-Rule. If the coefficient EF is equal to 1 then the rule is called exact rule. Then the following properties are equivalent:

Ci[1] ∊ Cj[2] (EF) ⇔

∀ Objets Ob1 ∊ Ci[1] => Ob1 ∊ Cj[2] (EF)

∀ Objets Ob1 ∊ Ci[1], Ob1 verifies the property p1 of Ci[1] and the property p2 of Cj[2]. (EF)

Theorem 2

Let Cn[1], Cn[2], Cn[3] be three clusters generated by a fuzzy clustering algorithm in the D [i] , D [j] and D[k] sites, respectively. The meta-rule Ci[1], Cj[2] Ck[3] with a coefficient (EF) is denoted by Ci[1], Cj[2] Ck[3] (EF) where EF = Sum ([µij [n]]Cn×Pn) ; µij [n] ∈ [0, 1]; ∀ i ∈ {1..Pn}; ∀ j ∈ {1,2,3} ∀ n ∈ {1..N}.

The value EF is in the range]0...1]. It is called the Exactitude Factor of this meta-rule. This value indicates the importance degree of this Meta-Rule. If the coefficient EF is equal to 1 then the rule is called exact rule. Then the following properties are equivalent:

Ci[1], Cj[2] ∊ Ck[3] (EF) ⇔

∀ Object Ob1 ∊ Ci[1] ∩ Cj[2]=> objects Ob1 ∊ Ck[3] (EF)

∀ Object Ob1 ∊ Ci[1] ∩ Cj[2] Ob1 verifies the property p1, p2 et p3. (EF)

V. Validation and Experimental Results

An in-depth performance study has been performed to compare our method to classical ones. First of all, note that our approach presents these two essential assets:

- The introduction of Meta Association Rules concept: This concept injects a layer of abstraction, which is very crucial and fundamental when we are dealing with a huge size of data. It allows having a more global view on the voluminous dataset. Besides, we define association rules between classes, thus, enabling automatic generation of association rules between data.
- The extensibility and versatility of the procedure: In our approach, the association rules mining step can be performed with any KDD algorithm. In the literature, studies have shown that one KDD algorithm could be more effective than another depending on the used data's domain. Thus, we have the luxury of choosing the most optimal method according to the domain of the used dataset. Even better, the clustering step in our process can be fulfilled with any fuzzy clustering algorithm, in order to classify the starting data.

In the following we discuss our experimental results:

Both the Java platform and the R platform are exploited in order to accomplish the implementation phase. The final program is, for the most part, in Java language, while some specific functionalities where dispatched for the R environment in sake of efficiency. In fact, R, by default, comes with a lot of commands carried out for data mining analysis. To interface between the two platforms, we integrated the rJava library which enables embedding basic R code snippets in Java code. Java is an object-oriented programming language that allows having a well-structured, modular and much more maintainable application. In addition, Java is designed to make easy distributed computing with intrinsically embedded network functionality. However it is not much efficient when it comes to statistical or mathematical modelling. In the other hand, R is a programming language that process and organize datasets in order to apply complex statistical tests. It propounds to organize and process large volumes of data quickly and flexibly. R is a programming language, but its use is strongly oriented towards the analysis of data and statistics. It provides a wide variety of modern and classical statistics (linear and nonlinear modeling, classical statistical tests, classification, clustering). Therefore, we adopted a combination of the two technologies.

In order to assess effectiveness of the proposed approach, we operate thorough tests on three real-life datasets. The first one is the Mushrooms dataset and it illustrates a set of dense data that depict fungi characteristics (surface, odor, color, edible or poisonous). The second one is the C20d10K dataset, which is a sample of the PUMS90KS file (Public Use Microdata Samples). It contains Census Kansas data carried out in 1990. The 10,000 lines of data (corresponding to the first 10,000 people listed) were selected to include only the first 20 attributes. The third one is the T10I4D100K dataset and it characterizes synthetic data from the marketing basket, generated artificially by the IBM generator.

The following table (Table VI) summarizes the properties of the datasets:

A basic and simple distribution method was to randomly split up horizontally each dataset into two sites. Next, we fixed the minSupport to 30% for Mushrooms, 20% for C20d10K and 0.02% for T10I4D100K. Therewith, The MinConf was varied between 80%, 40%, and 10% for each dataset. Next parameter to fix concerns the cleaning step; we settled the accuracy to 0.1 (α=0.1) for Mushrooms and T10I4D100K, and 0.03 for C20d10K. To tune the fuzzy clustering step, we fixed the fuzzy degree to 2 (m = 2), and we varied the cluster number between 5, 10, 15 and 20. Note that the mushrooms dataset is non-binary and non-digital, thus, we have run an extra pre-processing step to rewrite its data in digital format.

TABLE VI. The Characteristics of the Test Datasets

	Number of objects	Average size of objects	Number of items
Mushrooms	8 415	23	128
C20d10K	10 000	20	386
T10I4D100K	100 000	10	1 000

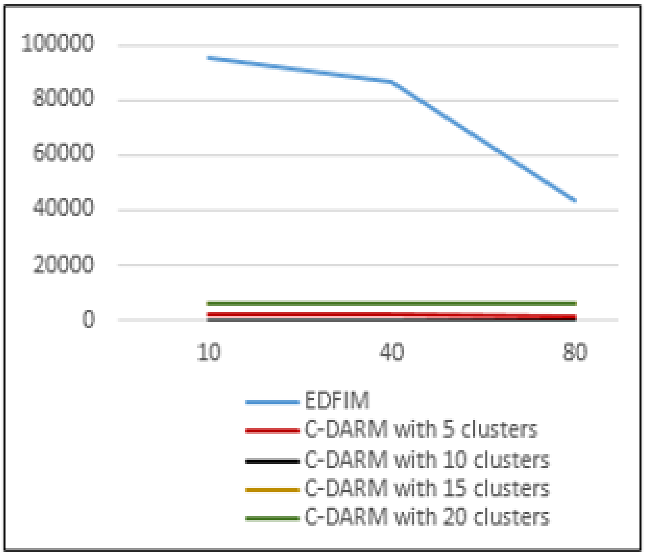

Fig. 4. Number of Generated Rules with Mushrooms.

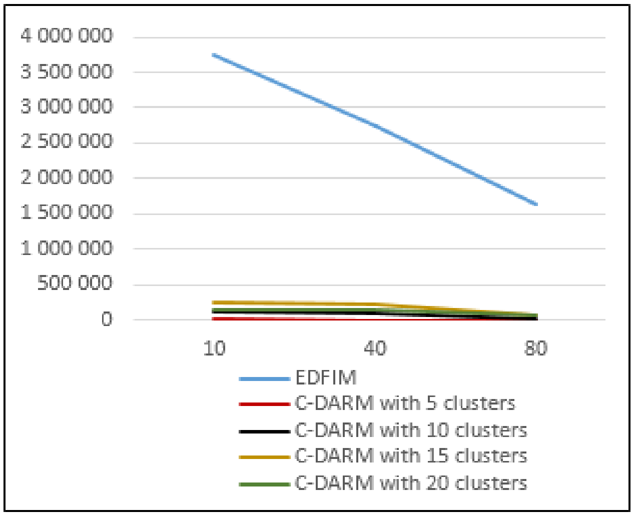

Fig. 5. Number of Generated Rules with C20d10K.

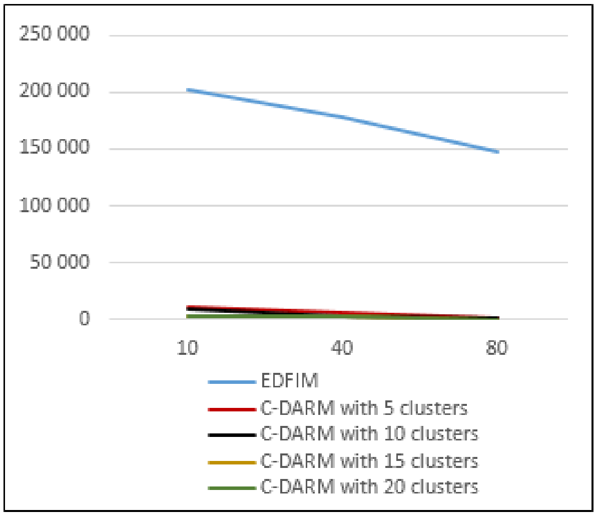

Fig. 6. Number of Generated Rules with T10I4D100K.

We summarize below (Fig. 4-6) our experimental results with emphasis on the number of generated rules, properties of each dataset. It confirms that the clustering phase led to a significant decrease of this number. This reduction is due to the fact that association rules are mined from classes, in contrast to the classical mining process which produces rules from raw data that is very large. Indeed, the number of clusters generated by clustering algorithm is always lower than the number of starting objects on which the clustering algorithm is applied.

Likewise, we notice a significant decrease on the execution time of our process, compared to traditional methods. This reduction could be discerned from the reduced number of generated rules. The main reasons for this are the size optimization of the the itemsets (Meta-Itemsets), as well as of the extraction contexts (Cluster-Fuzzy Formal Contexts). Moreover, such gain leads to an important decrease of the time necessary to scan the input at every pass. We summarize below (Fig. 7-9) our experimental results with emphasis on the execution time, properties of each dataset.

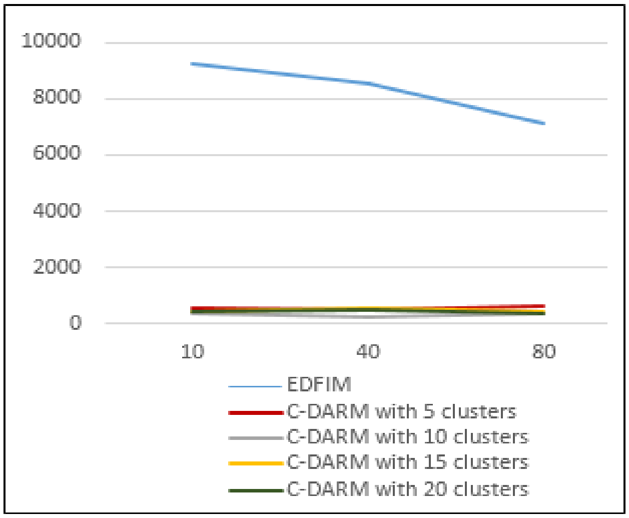

Fig. 7. Execution Time with Mushrooms.

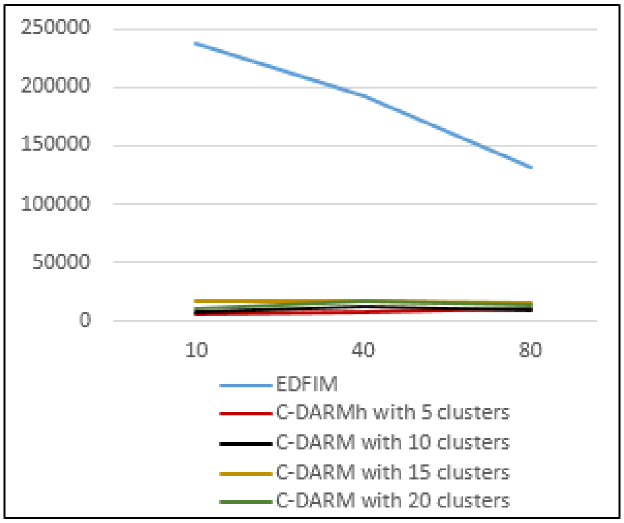

Fig. 8. Execution Time with C20d10K.

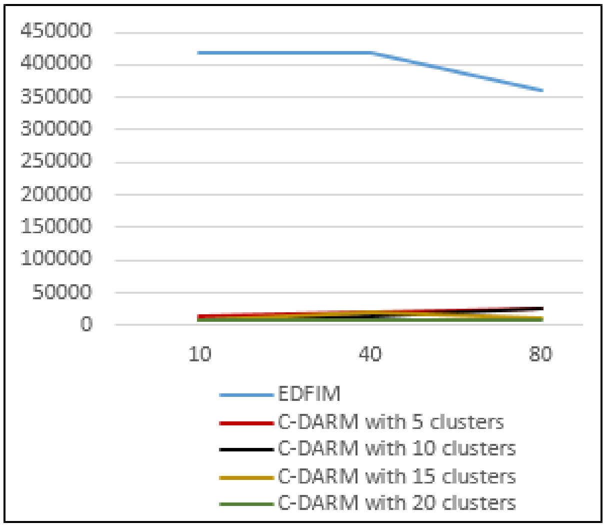

Fig. 9. Execution Time with T10I4D100K.

VI. Conclusion

In this paper, we have investigated embedding clustering technology in the association rules mining for the sake of readability and exploitation. The main idea is to mine distributed frequent itemsets from a representative set consisting of a collection of classes, called Meta-Itemsets. We generate from these classes a new representation more condensed and optimized of the extraction context in the form of a Cluster-Fuzzy Formal Context. From these new contexts, we mine distributed frequent Meta-Itemsets through a distributed process. We finally generate rules in the form of Meta-Rules. The number of these Meta-Rules is much fewer than the number of rules generated by the classical association rules mining algorithms. Indeed, the number of clusters generated by a clustering algorithm is always fewer than the number of starting objects on which the clustering algorithm is applied. As summary, our approach presents an essential asset which is the introduction of Meta Association Rules concept. It injects a layer of abstraction and more global view, which is very crucial and fundamental when we are dealing with a huge size of data.

As a perspective, we propose to intervene the Big Data technologies, like MapReduce, which prove their effectiveness for distributed data mining approaches.

References

[1] R. Agrawal, T. Imielinski, and R. Swami, “Database mining: A performance perspective”, IEEE Transaction on Knowledge and Data Engineering, 1993.

[2] R. Agrawal, R. Skirant, “Fast algorithms for mining association rules”, Proc. 20th International Conference on Very Large Databases, pp. 478–499, 1994.

[3] J. Han, J. Pei, and Y. Yin, “Mining Frequent Patterns without Candidate Generation”, Proc. ACM International Conference on Management of Data, vol. 29, no. 2, pp. 1-12, 2000.

[4] K. Lin, & S.-H. Chung, “A fast and resource efficient mining algorithm for discovering frequent patterns in distributed computing environments”, Future Generation Computer Systems, vol. 52, pp. 49-58, 2015.

[5] P. Asha, & S. Srinivasan, “Distributed Association Rule Mining with Load Balancing in Grid”, Journal of Computational and Theoretical Nanoscience, vol. 13(1), pp. 33-42, 2016.

[6] M. Z. Ashrafi, D. Taniar, K.Smith. “An Optimized Distributed Association Rule Mining Algorithm“, IEEE Computer Society, Distributed Systems, vol. 5, No. 3, pp. 1541-4922, 2004.

[7] A. Adelpoor, M. SanieeAbadeh, “An Efficient Frequent ItemSets Mining Algorithm for Distributed Databases”, International Journal of Computer Science and Electronics Engineering, Volume 1, Issue 1, ISSN 2320–4028, 2013.

[8] I. Pramudiono, and M. Kitsuregawa, “Parallel FP-growth on PC cluster”, Proc. Advances in Knowledge Discovery and Data Mining, Springer Berlin Heidelberg, pp. 467-473, 2003.

[9] Li, B., Pei, Z., & Qin, K, “Association Rules Mining Based on Clustering Analysis and Soft Sets”, In International Conference on Computer and Information Technology, pp. 675-680, 2015.

[10] T. T. Quan, L. N. Ngo, & S. C Hui, “An effective clustering-based approach for conceptual association rules mining”, Proc. International Conference on Computing and Communication Technologies, pp. 1-7, 2009.

[11] R. Agrawal and R. Srikant. “Fast algorithms for mining association rules in large databases.” In Proc. Of the 20th Int. Conf. On Very Data Bases (VLDB’94), pages 478-499, September 1994.

[12] J. S. Park, M.-S. Chen, and P. S. Yu. “An effective hash based algorithm for mining association rules.” In Proc. Of rhe 1995 ACM SIGMOD Int. Conf. (SIGMOD’95), pages 175-186, May 1995.

[13] A. Savasere, E. Omiecinski, and S. Navathe. “An effective algorithm for mining association rules in large databases.” In Proc. Of the 21st VLDB Int. Conf. (VLDB’95), pages 432-444, September 1995.

[14] H. Toivonen. "Sampling large databases for association rules." In Proc. Of the 22nd VLDB Int. Conf. (VLDB’96), pages 134-145, September 1996.

[15] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. "Dynamic itemset counting and implication rules for market basket data." In Proc. Of the 1997 ACM SIGMOD Int. Conf. (SIGMOD’97), pages 255-264, May 1997.

[16] R. Wille « Restructuring lattice theory: an approach based on hierarchies of concepts ». Ordered sets, pp. 445–470, 1982.

[17] L. Lakhal, G. Stumme «Efficient Mining of Association Rules Based on Formal Concept Analysis Formal Concept Analysis: Foundations and Applications, pp.180-195, 2005.

[18] N. Pasquier, « Extraction de Bases pour les Règles d'Association à partir des Itemsets Fermés Fréquents ». Inforsid'2000 Congress, 2000.

[19] N. Pasquier, « Data mining : algorithmes d'extraction et de réduction des règles d'association dans les bases de données ». Thèse de doctorat, Université de Clermont-Ferrand II, Janvier 2000.

[20] N.Pasquier, Bastide, Y., R.Taouil, & Lakhal, L. (1999). Discovering frequent itemsets for association rules. Proceeding of the 7th biennial international conference in database theory (‘ICDT 1999).

[21] M. J. Zaki, C. J. Hsiao, « CHARM: An Efficient Algorithm for Closed Itemset Mining », Proceedings of the 2nd SIAM International Conference on Data Mining, Arlington, pp. 34- 43, Avril 2002.

[22] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal, « Fast Computation of Concept Lattices Using Data Mining Techniques », M. Bouzeghoub, M. Klusch, W. Nutt, U. Sattler, Eds., Proceedings of 7th Intl. Workshop on Knowledge Representation Meets Databases (KRDB’00), Berlin, Germany, pp. 129-139, 2000.

[23] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal, « Computing Iceberg Concept Lattices with TITANIC », J. on Knowledge and Data Engineering (KDE), vol. 2, no 42, pp. 189-222, 2002.

[24] J. Pei, J. Han, R. Mao, S. Nishio, S. Tang, D. Yang, « CLOSET : An efficient algorithm for mining frequent closed itemsets », Proceedings of the ACM SIGMOD DMKD’00, Dallas,TX, pp. 21-30, 2002.

[25] Agrawal (R.), Shafer (J. C.),”Parallel Mining of Association Rules”. IEEE Transactions onKnowledge and Data Engineering, Vol. 8, No. 6, pp. 962-969, December 1996.

[26] M.J. Zaki. Parallel and Distributed Association Mining: A survey. In IEEE Concurrency, special issue on Parallel Mechanisms for Data Mining, Volume 7, N°4, pages 14-25, December 1999.

[27] Shintani (T.), Kitsuregawa (M.), “Hash Based Parallel Algorithms for Mining AssociationRules”. Proc. 4th Int. Conf. Parallel and Distributed Information Systems, IEEE Computer Soc.Press, Los Alamitos, Calif., 1996, pp. 19–30.

[28] Cheung (D. W.), Han (J.), Vincent (Ng.), Fu (A. W.), Fu (Y.). “A Fast Distributed Algorithm forMining Association Rules”. Proceedings of PDIS, 1996.

[29] J. ArokiaRenjit, K.L.Shunmuganathan, “Mining the data from distributed database using an improved mining algorithm”, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 3, March 2010, USA.

[30] M.J. Zaki. Parallel and Distributed Association Mining: A survey. In IEEE Concurrency, special issue on Parallel Mechanisms for Data Mining, Volume 7, N°4, pages 14-25, December 1999.

[31] Han E. H., Karypis G., and Kumar V., Scalable Parallel Data Mining for Association Rules, in Proceedings of ACM Conf. on Management of Data, ACM Press, New York, pp. 277-288, 1997.

[32] Zaki M. J., Parthasarathy S., Ogihara M., and Li W., Parallel algorithms for discovery of association rules, Data mining and knowledge discovery, vol. 1, no. 4, pp. 343-373, 1997c.