A New Approach to Improve Association Rules for Big Data in Cloud Environment

Djilali Dahmani, Sidi Ahmed Rahal, and Ghalem Belalem
Department of Computer Science, University of Science and Technology, Algeria

Abstract: *The technique of association rules is very useful in Data Mining, but it generates a huge number of rules. So, a manual post-processing is required to target only the interesting rules. Several researchers suggest integrating users' knowledge by using ontology and rule patterns, and then select automatically the interesting rules after generating all possible rules. However, nowadays the business data are extremely increasing, and many companies have already opted for Big Data systems deployed in cloud environments, then the process of generating association rules becomes very hard. To deal with this issue, we propose an approach using ontology with rule patterns to integrate users' knowledge early in the preprocessing step before searching or generating any rule. So, only the interesting rules which respect the rule patterns will be generated. This approach allows reducing execution time and minimizing the cost of the post-processing especially for Big Data. To confirm the performance results, experiments are carried out on Not Only Strutured Query Language (NoSQL) databases which are distributed in a cloud environment.*

Keywords: *Big data, association rules, rule patterns, ontology, cloud computing, NoSQL.*

Received April 5, 2016; accepted September 23, 2018

1. Introduction

Most of companies are interested in Data Mining. They use historical data to extract hidden knowledge, and then provide early feedback means to guess future data and predict actions in facts [22], e.g., a provider would have ideas about general tendencies of his clients, and use them to improve their future needs. The technique of Association Rules (AR) is a simple useful Data Mining method. However, its main problem is the large number of generated rules. To pick the interesting rules, a post-processing must be done manually by users. To avoid this manual work, many proposals are developed to find automatically pertinent rules after generating all rules. The results of these proposals are satisfactory.

However, the volume of data is steadily increasing, especially in industrial area. Several companies are equipped with digital acquisition systems that generate very large amounts of data. Hence, Big Data concepts and new systems, like Not Only Strutured Query Language (NoSQL) [13] and newSQL, become more popular. Also, many distributed environments have been developed to support storage and processing of Big Data, like Cloud Computing. This evolution has an enormous impact for AR technique. Looking for the frequent itemsets and generation of all AR become extremely hard regarding storage and processing sides. In this article, we propose an approach to deal with this issue, and integrate users' knowledge as rule patterns or ontology in early step.

The paper is organized in six sections. The first section gives an introduction. We start going over the general constraints of the AR technique in the second section. Related work is presented in the third section. In the fourth section, our approach is presented. In the fifth section, experimental illustration is given, and in the sixth section, their results are discussed to confirm the performance. Finally, a conclusion closes this article.

2. Background Information

Data mining groups a lot of techniques to extract knowledge. It is used in many areas, like business, statistics, research, biology, etc., [7]. Their methods can be classified by principle (supervised, unsupervised), by objective (descriptive, predictive) [7], etc., Many techniques are steadily developed [5], like association rules AR, clustering, decision trees, neurons network, etc. The choice of a fitting technique depends closely on requirements and nature data.

The technique of AR is very interesting. It allows detecting association or links between data (itemsets) in form of rules (X→Y) that can give interesting unknown results for users [17, 19]. An association rule X→Y means that the most transactions (records) which verify the premise X in a context (database), also verify the conclusion Y. Each rule is evaluated by two measures: support and confidence. A rule X→Y verifies a support S if at less S % of transactions verified X and Y, and it verifies a confidence C if at less C % of transactions that verify X verify also Y. For example, for the rule "80% of students who learn Linux, learn also Java, and 30 % of all students have learnt these two courses", we can say that this rule is verified with certitude more than 80% (confidence), and it is supported by at least 30% of students (support). A rule to be selected, must has its support and confidence greater or equal than a user defined threshold Sup_{min} and Conf_{min} respectively. The goal is to discover all rules that verify these two conditions. After selecting and preparing data, the process of rule extraction can be done in two steps [1]:

- First step: Looking for the frequent itemsets.
- Second step: Generating association rules list.

The second step is rather straightforward, and the first step dominates the processing time.

Many efficient algorithms are proposed to implement AR, the well-known is Apriori algorithm [1]. After, many scholars have improved this algorithm and have presented new variants algorithms like FP-Growth [6], Closet [16], etc. Nevertheless, their major inconvenient resides in the important huge number of generated rules. For example, in Aprioi algorithm, all potential itemsets (set of items or attributs of a transaction) in a context are checked. After building the trellis (graph of all combinations of itemset subsets), if we have m itemsets, then we will have 2^{m-1} scan iterations to do. These scans are necessary to calculate the support of each itemset and to mark it as frequent. So, the number of scans and generated rules are exponentially dependent on itemsets [25]. So, a post-processing must be done manually by users to target the interesting rules that can be considered as effective knowledge. This post-processing must to be adapted to both the user preference and data structure.

3. Related Work

In order to automatize the post-processing, many approaches were developed to find pertinent rules. Some approaches, like that of Silberschatz and Tuzhilin [19] proposed to decrease the number of rules generated by using interest objective or subjective measures.

- Objective measures that are relied just to data structure. Many works, like guided by Piatetsky-Shapiro *et al*. [17], Bayardo and Agrawal [2], Hilderman and Hamilton [8], Tan *et al*. [23], Guillet and Hamilton [4], have summarized these measures and compared them. But, these measures offered just a partial response to post-processing, since they are limited to just data evaluation.
- Subjective measures that integrate explicitly expert's or manager's knowledge. Approaches which integrate these measures are mainly distinctive with representation models of knowledge.

Some authors proposed using templates to describe interested and uninterested rules [10]. Others used two representation models for user's conviction: General Impressions (GI) and Reasonably Precise Knowledge (RPK). A version of RPK in fuzzy logic has been developed by Liu *et al*. [11] to select the classification rules based on syntactic comparison. Other more exact representations of user's knowledge by using rules have been developed by Padmanabhan and Tuzhuilin [15], and the rule interest was defined by logical contradiction. Agrawal and Srikant [1] proposed to represent user's knowledge by General Association Rules (GAR), and integrate knowledge by hierarchal taxonomy of attributes. The introduction of knowledge in attributes structure allowed decreasing number of rules. Later, Liu *et al*. [11] developed this taxonomy to become rule patterns which can represent vast user's knowledge in a particular domain. These rule patterns allowed defining a characteristic form of interested rules. Furthermore, some authors propose to integrate others measures like utility or incremental mining [9]. Marinica *et al*. [12] Laboratoire Informatique de Nantes Atlantique- Connaissance, Optimisation et Decision (LINA-COD team) proposed an interesting approach to introduce user's knowledge in the extraction of AR by using ontology associated with rule patterns. Since that, ontology has been used in AR process and then let ontology's benefits to be enjoyed. Ontology is a conceptual database that allows users to modelize knowledge, and provide a shared vocabulary. For a company, ontology represents its memory.

As result, most of these proposals integrate gradually and efficiently the users' knowledge, and contribute to automatize the post-processing. However, they continue steadily to deal with the huge number of rules since all possible rules are generated. Currently, with the data increasing cadence and Big Data emergence, the post-processing will become very hard and consumer in time and space. The number of rules cannot be supported and covered by the above proposals. In the next section, we present our approach to solve this issue in distributed Big Data context.

4. Our Proposal (ARBD approach)

Firstly, we are interested in LINA-COD's approach [12]. It is efficient since it integrates users' knowledge by using ontology and rule patterns. We give our suggestion to extend it in Big Data context. LINA-COD's approach is based on 3 elements (Figure 1):

- A database in which association rules are extracted.
- An ontology representing knowledge in database.
- A set of rules schemes, concerning the concepts of ontology to select interesting rules.

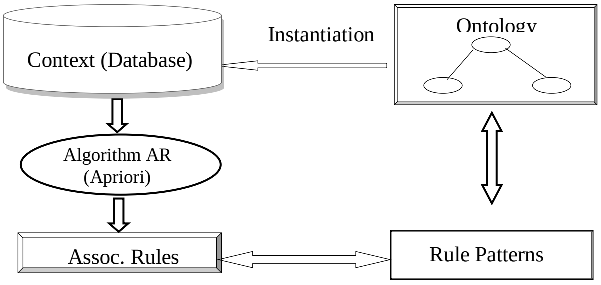

Figure 1. Representation of LINA-COD team‘s approach [12].

Formally, ontology is a set of concepts linked by relations of conceptualization [12], in which a parent concept is a generalization of child, and a child is a specialization of its parent. This hierarchy of concepts is an efficient tool to collect, and validate interesting new knowledge. Also, rule patterns allow carrying out a supervised selection on AR. They permit to express knowledge by using a model of investigated rules: X_1, X_2, X_3 ... → Y_1, Y_2, Y_3 ..., where X_i and Y_j are constraints on concepts (ontology) or on attributes (database). As a result, the rule patterns combined with ontology allow increasing ability to target interesting rules in the post-processing step.

However, in this approach, we note that the whole context (database) is taken into account in exploring AR, and afterward the rule patterns are used to filter just the interesting rules. Also, we note that all possible itemsets are considered in searching AR, regardless of what these itemsets will be or not considered later by rule patterns. In Big Data, it is very hard to consider the entire context and all possible itemsets. To deal with this issue, we propose to filter the context at the beginning by using only itemsets which are included and respect the rule patterns chosen by experts. This allows limiting the field of investigation. Only a part of context which contains itemsets appearing in the rule patterns can be considered in exploration step. In generating rules step, only rules respecting the rule patterns can be generated. In Figure 2, we give the schema for our approach. We name it acronym for Association Rules in Big Data (ARBD).

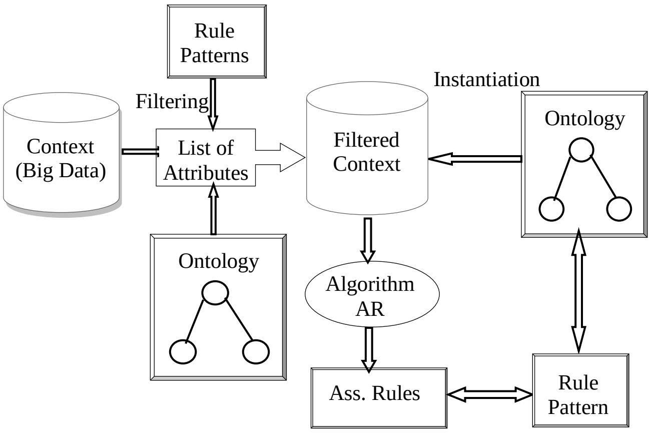

Figure 2. Representation of proposed ARBD approach.

We describe this proposal in the following phases.

- *Phase* 1: Filtering the Big Data Context. We use the rule patterns to filer the context to have just instances that contain the items included in the rules patterns. For instance, let consider *C1,C2→C3* as an interesting rule pattern for users, where *C1*, *C2* and *C3* are concepts in the ontology. This rule pattern means that we have to consider all rules which their promises verify *C1* and not *C2*, and their conclusions verify *C3*. Using the relation between concepts and attributes, a concept is joined with one or many attributes. In simple mono-relation concepts/attributes, this rule of concepts can be converted to a rule of attributes like *A1,A2→A3*. So, we can filter the context to consider just instances that contains *A1*, *A3* and not *A2*. We do the same action for all other rule patterns to exclude any itemset not included in any rule pattern. Finally, we get a new restricted context that will be used in the next step.
- *Phase* 2: Using the filtered database context. We use the restricted context as the same method described in LINA–COD‘s approach. The database consists of a set of *n* transactions described through *p* attributes. Let $I=\{I_1, I_2... I_p\}$ the set of attributes called features (items) and $T=\{t_1, t_2 ... t_n\}$ the set of *n* transactions. Each transaction $t_i=\{I_1, I_2, ..., I_m\}$ is a subset of the set of attributes *I*. Apriori algorithm (or any variant) allows extraction of rules in form *X→Y*, where X and Y are two disjoint sets of attributes. Ontology is defined by a set of concepts $C=\{C_1, C_2... C_c\}$ and a set relationships/properties $R=\{R_1, R_2... R_r\}$. The database is connected to ontology; each concept of ontology is instantiated in the database by a subset of attributes (records). A simple way is to associate a concept directly to one attribute. Finally, a rule pattern can express knowledge about the form of the rules sought. The semantic extension of "general impression" allows combining in rule patterns constraints on attributes and concepts [12]. Figure 3 recapitulates the steps of ARDB approach.

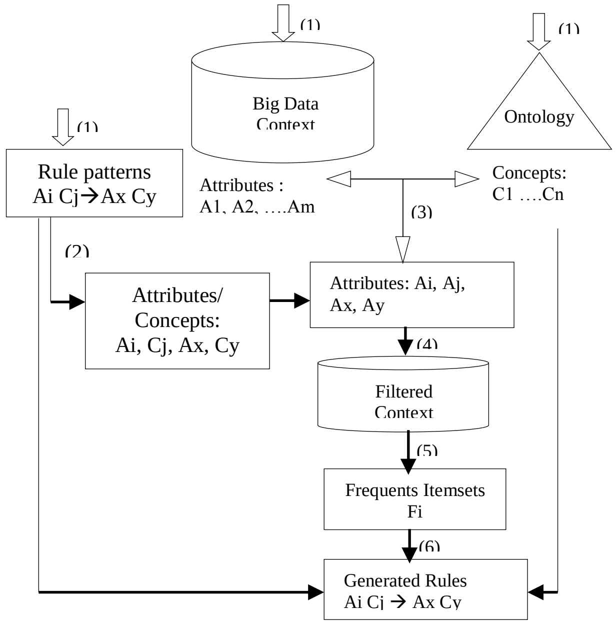

Figure 3. Plan of ARBD approach.

In order to bear out the efficiency of ARBD approach, we perform experiments in the next section.

5. Experimentation

We present data of oil and gas Company [21] and the distributed platform used. Then, we conduct three interesting experiments.

5.1. Data and Distributed Platform

Our company contains many activities [21]. We are interested in the downstream activity that is composed of a head office and six plants. Each plant has a database. The head office has large consolidated databases which were recently migrated from Oracle to NoSQL MongoDB [14]. This data migration has been done by using approach provided by Dahmani et al. [3]. MongoDB was chosen because of its document store nature, efficiency and current ranking [20].

We use industrial production data at the headquarters as shown in Figure 4. These data belong to a strategic business domain regulated by international standards (production, stocks, shipping, etc.,), and have a NoSQL nature (volume, velocity, variability) [13]. We use the shipping collection MongoDB that contains data about loading products into boats. Ontology is created to represent concepts related to this collection (units, products, clients, etc.,). This ontology is implemented by using Protégé [18] and OWL [24].

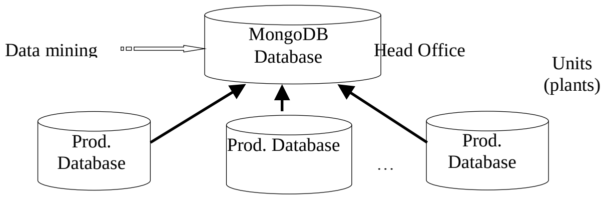

Figure 4. Production NoSQL databases for head office and units.

We use distributed MongoDB platform that does not require many resources than Cloud IaaS or Hadoop). Our platform contains 7 nodes or shards (Figure 5):

- *Mongo server* (*mongos*): manages data in 5 nodes.
- *Configuration server*: stores shards' configuration.
- 5 *Shards* servers, each one runs mongod service and hold a data partition distributed by Mongos.

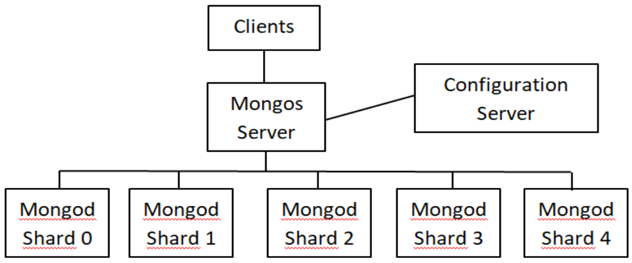

Figure 5. Distributed MongoDB used in experiments.

All nodes of are identical. We have configured this platform and installed a database on it according to the procedure and best practices of MongoDB [14].

5.2. Experiments

We conduct our experiments by taking some interesting rule patterns chosen by users, and we apply ARBD approach to restrict the context (only interesting itemsets), and then we carry out the Apriori algorithm. The frequent itemsets are discovered with a minimum support and confidence previously chosen by users. Note that since the same data and ontology are used in experiments, the same number of pertinent AR would be found at the end regardless of using or not ARBD.

In order to make things easy, we have developed a java application called Prontodam that implements Apriori. Prontodam (see Figure 6) allows users to connect to the database, set minimum support and confidence, choose ontology, and build rule pattern from concepts or attributes. It can be executed (1) with ARBD (2) or without it.

These two alternative executions allow comparing results and have an idea about gain.

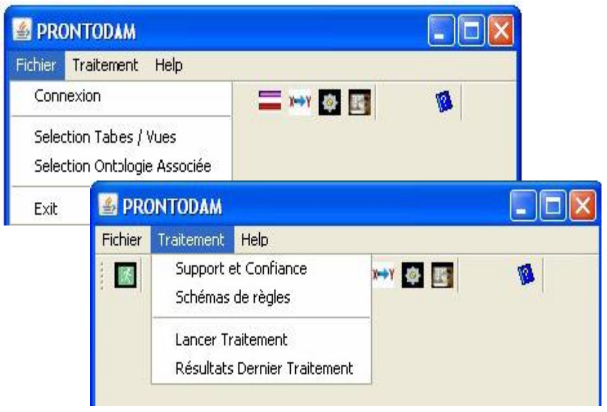

Figure 6. Some print screens of our application called ProntoDam.

- *First experiment*: Firstly, we start with the most interesting rule pattern. The users want to discover the client's interest in each product delivered by units. Note that the same product is different in each unit because its quality depends on specific parameters delivered by this unit. As example, the Specific Calorific Power (SCP) is a parameter for Liquefied Natural Gas (LNG) product. After each boatload, a quality certificate of product is delivered by the company and checked by its client. Users want to have an idea about product quality chosen by each client. So, the following rule pattern is given: "The client is interested in such product produced by such unit". This means that users are interested in AR with the form *"Client→Product, Unit"*. This rule pattern is very useful because it allows managers to know the clients' interests, and then they could recommend product quality to units. Note that *Client*, *Product*, *Unit* are concepts in ontology, and can be translated into their attributes *Code_Client*, *Code_Product*, *Code_Unit* respectively.

With *Prontodam*, we carry out the operation to generate the association rules by using alternatively the ARDB approach (first alternative) or the baseline LINA–COD's approach (second alternative). So, we repeat many times the execution on the rule pattern *"Client→Product, Unit"* by:

1. Varying minimum values for support and confidence.
2. Using or alternatively ARBD or the baseline approach.

In Table 1, we recapitulate the data context size, the numbers of both frequent itemsets and generated rules. As the context is filtered in ARBD, we give a very good reduction.

- *Second experiment.* Many clients rent boats from others companies to load and transport their products. They choose boats depending on many criteria. So, our users want to have an idea about their customers' choices. The following rule pattern is given for that: "The client is interested in such boat". This rule pattern means that users are interested in AR with the following form: "*Client → boat*". The ontology concepts Client and boat can be translated to their attributes Code_Client, Code_Boat respectively.

As the previous experiment, we do the same for the rule patter "Client→boat". We perform many times the execution on this rule pattern by:

1. Varying minimum support and confidence.
2. Using or not ARBD approach. The results of this experiment are shown in Table 2.

To reduce the display, we have given only important values (the number of generated rules and execution time).

Table 1. Results of the rule pattern "Client → Product, Unit".

Number of handled instances (data context size)		Minimum Support	Minimum confidence	Number of frequent itemsets		Number of generated rules (pertinent rules)			Execution time (hour, minute, second)	
Baseline approach (No filter)	ARBD approach (use filter)			Baseline approach	ARBD approach	Baseline approach		ARBD approach	Baseline approach	ARBD approach
						Before filter	After filter			
3371905	529824	10.25%	75%	525	109	1516	19	19	59 m 33s	5m 22s
		12. 50%	70%	695	98	1569	15	15	1h 1m 13s	4m 57s
		7.75%	80%	1071	159	3122	41	41	1h 23m 52s	10m 45s
		4.50%	85%	1663	194	5045	68	68	1h 48m 19s	21m 49s
		2.25%	65%	2156	321	9331	101	101	2h 3m 7s	39m 27s

Table 3. Results of the rule pattern "Client, Season → Product".

Number of handled instances (data context size)		Minimum Support	Minimum confidence	Number of generated rules (pertinent rules)			Execution time (hour, minute, second)	
Baseline approach (No filter)	ARBD approach (use filter)			Baseline approach		ARBD approach	Baseline approach	ARBD approach
				Before filter	After filter			
3371905	890509	10.25%	75%	820	11	11	52m 6s	4m 55s
		12. 50%	70%	1231	12	12	58m 46s	4m 31s
		7.75%	80%	2256	31	31	1h 27m 29s	8m 5s
		4.50%	85%	4584	45	45	1h 36m 15s	10m 40s
		2.25%	65%	6993	91	91	1h 49m 58s	26m 37s

Table 2. Results of the rule pattern "Client → boat".

Number of handled instances (data context size)			Minimu m Support	Minimum confidence	Number of generated rules (pertinent rules)			Execution time (hour, minute, second)
Baseline approach (No filter)	ARBD approach (use filter)			Baseline approach		ARBD approach	Baseline approach	ARBD approach
				Before filter	After filter			
3371905	1056658	10.25%	75%	1609	26	26	1h 2m 51s	6m 13s
		12. 50%	70%	1830	29	29	1h 10m 11s	6m 49s
		7.75%	80%	4312	48	48	1h 51m 2s	13m 15s
		4.50%	85%	7110	89	89	2h 13m 9s	27m 17s
		2.25%	65%	10556	208	208	2h 23m 17s	46m 36s

- *Third experiment*. Our users want to know which product is interesting for a client in a season. The AR is "*Client, Season→Product*". Their attributes are respectively Code_Client, Code_Season, Code_Product. We can calculate Code_Season from the attribute date (extract month and conclude its season). As the same of the previous experiments, we carry out the execution on this rule by varying the same values for minimum support and confidence, and using or not ARBD. Table.3 shows the results of this experiment. We can perform other experiments, but we will consider just these experiments to analyze and reveal the benefit of ARBD approach in the next section.

We can perform many other experiments on Shipping or other MongoDB collections, nevertheless we will consider just these experiments above to analyze and reveal the benefit of our proposed approach.

6. Results and Discussion

As result, we see that the same numbers of pertinent AR are always found with ARBD or with without (baseline LINA-COD approach). However with ARBD, the context is initially filtered, and the frequent patterns and the target rules are obtained with a very high-speed manner, and we give a high well reduction in all results. As we have explained before, the main reason behind this faster improvement is the filtering of the context. Only itemsets appearing in the rule patterns are considered in the exploration step, and only rules respecting these rule patterns are generated at the end.

The two graphs in Figures 7 and 8 show the great difference between the numbers of pertinent rules generated directly with ARBD and baseline approach for the previous experiments.

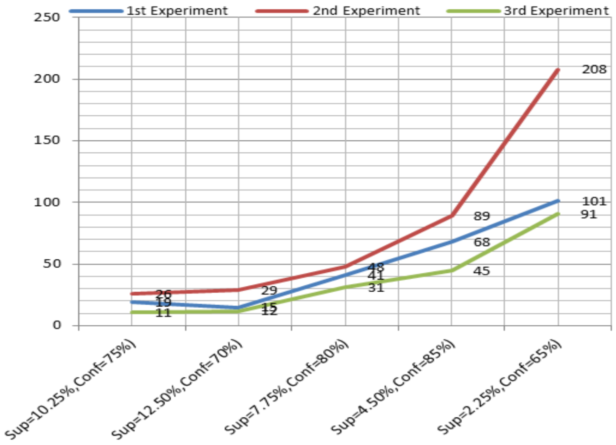

Figure 7. Number of generated rules with ARBD approach.

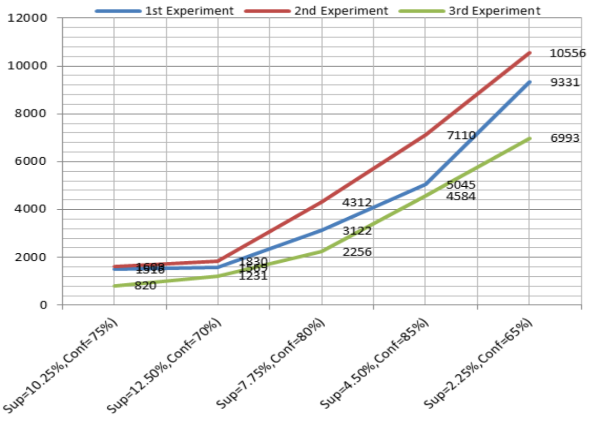

Figure 8. Number of generated rules with baseline approach.

In the same way, the graphs at Figures 9, 10 and 11 point up the profit given in execution time related to these experiments. We note the huge difference in execution time and number of generated association rules. As the context increases gradually, this execution time becomes more favourable for the suggested proposal. So, the ARBD approach proves its performance, and becomes very interesting in Big Data context.

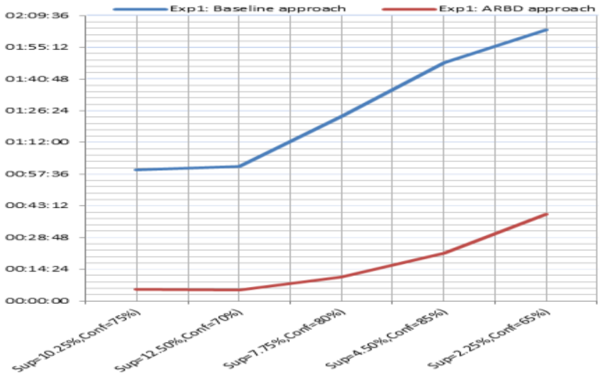

Figure 9. Comparing execution time for the first experiment.

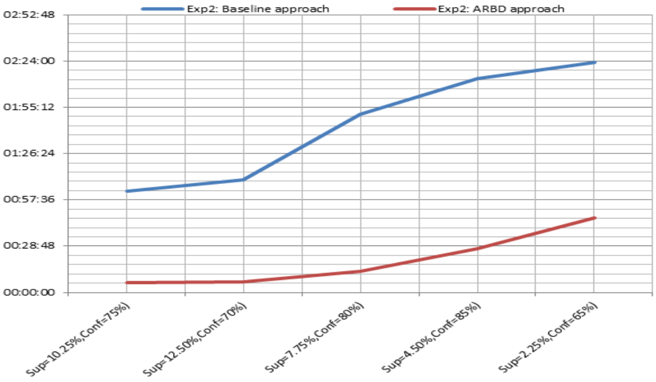

Figure 10. Comparing execution time for the second experiment.

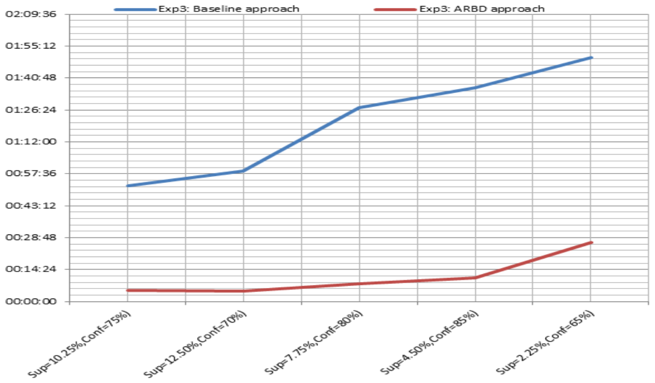

Figure 11. Comparing execution time for the third experiment.

7. Conclusions

In this paper, we have discussed the problem of the huge number of rules generated in the association rules technique. To avoid a hard manual post-processing, we have gone over some proposals. We have chosen that integrates users' knowledge by using ontology and rule patterns. The interesting rules are automatically selected after generating all association rules. However, with the emergence of Big Data typically deployed in distributed environments, these proposals are not efficient and need an adjustment.

To deal with this issue, we have presented a new approach called ARBD which uses rule patterns and ontology to filter data before any processing. Only the interesting rules which respect the rule patterns are generated at the end. To verify its efficiency, we have carried out experiments on distributed NoSQL MongoDB data system. After comparing results, we note that the adjustment given by ARBD approach allows reducing significantly the execution time, the number of frequent itemsets. So, the cost of the any post-processing is really minimized especially for Big Data context.

As perspective, we intend to use ARBD approach for other NoSQL systems in various areas like data analytics and semantic web. We plan to extend this work in a large-scale cloud with many nodes, like OpenStack or Eucalyptus to see the gain boundaries. Finally, we are planning to use this proposal and apply it to the interesting NewSQL systems which occupy an attractive place in our perspective.

References

[1] Agrawal R. and Srikant R., “Mining Generalized Association Rules,” *in Proceedings of 21st International Conference on Very Large Data Bases*, San Francisco, pp. 407-419, 1995.

[2] Bayardo J. and Agrawal R., “Mining the Most Interesting Rules,” *in Proceedings of 5th ACM SIGKDD, Conference on Knowledge Discovery and Data Mining*, California, pp. 145-154, 1999.

[3] Dahmani D., Rahal S., and Belalem G., “Improving the Performance of Data Mining by Using Big Data in Cloud Environment,” *Journal of Information and Knowledge Management*, vol. 15, no. 4, 2016.

[4] Guillet F. and Hamilton H., *Quality Measures in Data Mining*, Springer, 2007.

[5] Han J., Kamber M., and Pei J., *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, 2012.

[6] Han J., Pei J., Yin Y., and Mao R., “Mining Frequent Patterns without Candidate Generation: a Frequent-Pattern Tree Approach,” *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53-87, 2000.

[7] Hastie T., Tibshirani R., and Friedman J., *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer-Verlag, 2008.

[8] Hilderman R. and Hamilton H., “Evaluation of Interestingness Measures for Ranking Discovered Knowledge,” *in Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, London, pp. 247-259, 2001.

[9] JeyaKumar K., Dhanabalachandran M., and JeyaKumar K., “Effective and Efficient Utility Mining Technique for Incremental Dataset,” *The International Arab Journal of Information Technology*, vol. 15, no. 1, pp. 157-166, 2018.

[10] Klemettinen M., Mannila H., Ronkainen P., Toivonen H., and Verkamo I., “Finding Interesting Rules from Large Sets of Discovered Association Rules,” *in Proceedings of the 3rd International Conference on Information and Knowledge Management*, Maryland, pp. 401-407, 1994.

[11] Liu B., Hsu W., Wang K., and Chen S., “Visually Aided Exploration of Interesting Association Rules,” *in Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, Beijing, pp. 380-389, 1999.

[12] Marinica C., Guillet F., and Briand H., “Vers La Fouille De Règles D’association Guidée Par Des Ontologies Et Des Schémas De Règles,” *LINA-COD team, Conferance*, Nice, 2008.

[13] McCreary D. and Kelly A., *Making Sense of NoSQL*, Manning Publications, 2014.

[14] Mongo B., Documentation Project MongoDB, https://www.mongodb.com, Last Visited, 2018.

[15] Padmanabhan B. and Tuzhuilin A., “Unexpectedness as a Measure of Interestingness in Knowledge Discovery,” *Decision Support Systems*, vol. 27, no. 3, pp. 303-318, 1999.

[16] Pei J., Han J., and Mao R., “CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets,” *ACM SIGMOD DMKD*, Dallas, pp. 21-30, 2002.

[17] Piatetsky-Shapiro G. and Frawley W., *Knowledge Discovery in Databases*, AAAI Press, 1991.

[18] Protégé, modeling Ontology Tool, Stanford University, http://protege.stanford.edu, Last Visited, 2018.

[19] Silberschatz A. and Tuzhilin A., "What Makes Patterns Interesting in Knowledge Discovery Systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 970-974, 1996.

[20] Solid IT, Ranking database management systems. http://db-engines.com/en/ranking, Last Visited, 2018.

[21] Sonatrach, the Algerian Oil and Gas Company, the first company in Africa and Medit. https://www.sonatrach.com, Last Visited, 2018.

[22] Taha Ahmed S., Al-hamdani R., and Crook M., "Studying of Educational Data Mining Techniques," *International Journal of Advanced Research in Science, Engineering and Technology*, vol. 5, no. 5, pp. 5742-5750, 2018.

[23] Tan P., Kumar V., and Srivastava J., "Selecting the Right Objective Measure for Association Analysis," *Information Systems*, vol. 29, no. 4, pp. 293-313, 2004.

[24] W3C Web Ontology Language (OWL), http://www.w3.org, Last Visited, 2018.

[25] Zaki M. and Wagner M., *Data Mining and Analysis: Fundamental Concepts and Algorithms*, Cambridge University Press, 2014.

Djilali Dahmani Graduated from Department of computer science, Faculty of exact and applied sciences, University of Science and Technology MB, USTO, Algeria, where he received PhD degree in computer science in 2017. His current research interests are Big data, Cloud computing, Data mining, database models, data science, replication, consistency, fault tolerance, resource management, energy consumption, mobile environment, High Performance Computing.

Sidi Ahmed Rahal He is Doctor in computer science since 1989 in Pau University, France. Currently, he is a professor at Department of computer science, University of Science and Technology MB, USTO, Algeria. His current research interests are Data mining, Object-Oriented database, Data Mining, Agents Expert Systems, Big data, Cloud computing, database models. He is a member of SSD (Signal, System and Data) laboratory.

Ghalem Belalem Graduated from Department of computer science, Faculty of exact and applied sciences, University of Oran1 Ahmed Ben Bella, Algeria, where he received PhD degree in computer science in 2007. His current research interests are distributed system; grid computing, cloud computing, replication, consistency, fault tolerance, resource management, economic models, energy consumption, Big data, IoT, mobile environment, images processing, Supply chain optimization, Decision support systems, High Performance Computing.