Time-Saving Approach for Optimal Mining of Association Rules

Mouhir Mohammed
IIMCS Laboratory
Dept. of Mathematics and Computer Sciences
FSTS, University of Hassan 1st
Settat, Morocco

Balouki Youssef
IIMCS Laboratory
Dept. of Mathematics and Computer Sciences
FSTS, University of Hassan 1st
Settat, Morocco

Gadi Taoufiq
IIMCS Laboratory
Dept. of Mathematics and Computer Sciences
FSTS, University of Hassan 1st
Settat, Morocco

***Abstract*—Data mining is the process of analyzing data so as to get useful information to be exploited by users. Association rules is one of data mining techniques used to detect different correlations and to reveal relationships among data individual items in huge data bases. These rules usually take the following form: if X then Y as independent attributes. An association rule has become a popular technique used in several vital fields of activity such as insurance, medicine, banks, supermarkets… Association rules are generated in huge numbers by algorithms known as Association Rules Mining algorithms. The generation of huge quantities of Association Rules may be time-and-effort consuming this is the reason behind an urgent necessity of an efficient and scaling approach to mine only the relevant and significant association rules. This paper proposes an innovative approach which mines the optimal rules from a large set of Association Rules in a distributive processing way to improve its efficiency and to decrease the running time.**

Keywords*—MDP_{REF} *Algorithm; Association Rules mining; Data partitioning; Optimization (profitability, efficiency and Risks) ; Bagging

I. INTRODUCTION

Big data is an important research topic and it has attracted considerable attention. The huge numbers of data sets are unused and redundant in the databases of companies, universities, etc. Discovering the unused and redundant information stored in these data bases is grounded on the efficient KDD (Knowledge Discovery in Database) process. This latter does not only retrieve data or let researchers find new information from data [1] but also has the ability to reveal the patterns and relationships among large amounts of data in a single or several data sets. KDD process makes use of several techniques from statistics and artificial intelligence in a variety of activities. The main activities are as follows [2-11]: Association Rules; Clustering; Classification; Regression and Prediction. We are rather interested in the association rules mining, together with classification and clustering which are two of the major data mining applications where pattern mining is extensively used to transform raw data into pattern-based description that is accepted and processed by classification and clustering algorithms. In this context, patterns which occur in data are simply considered as features that characterize data. Patterns describing the data are also called explanatory variables. Whereas Association Rules Mining is one of the most common algorithm-based data mining techniques which can be defined as the extractor or generator of interesting relationships and correlations among items in large amounts of data. [10] Although Association Rules, Clustering and Classification are the techniques extensively used in this paper, Regression and Prediction are going to be taken into account in our future work to reinforce the reliability and to improve the quality of results. For reasons related to the obviation of a possible confusion or misunderstanding, we provide below the definitions of the activities meant by both concepts:

- **Regression** for a set of items is the analysis of the relationships of dependence between the values of the attributes. A model is automatically produced that can predict attribute values for new items.
- **Prediction** for a specific item and a corresponding model is the ability to predict the value of a specific attribute. For example, in a predictive model for treatment schema, prediction is used to determine the next procedure in the sequence of treatment.

Lately, many algorithms have been suggested in the literature for instance: Close, Close +, Charm, Sky Rules,… to help generate association rules, either by improving the process of "patterns'extraction" or by introducing other criteria and factors in order to determine which rule to keep and which one to discard [3]. However, these algorithms are mainly used to centralize computing systems and relatively evaluate small databases. Yet, the huge numbers of generated association rules and modern databases are growing dramatically in terms of size. Consequently, several parallel and distributed solutions have been proposed to tackle this issue. In addition to that, many distributed frameworks have been used to deal with the existing abundance of data. These distributed frameworks focus on the challenges of distributed system building and on simple programming models for data analysis. To solve these problems, we think that a data partitioning technique considering data characteristics should be applied. In this paper, we propose a scalable and distributive approach for large scale frequent association rules. The proposed approach offers the possibility to apply any of the known association rules mining algorithms in a distributive way. In addition, it allows many possibilities to apply any of the known clustering or classification algorithms as partitioning techniques for the association rules set.

Besides the introduction, the paper is made up of four sections, each of which deals with a particular aspect: section II deals with the necessary definitions, section III describes the proposed approach of large-scale association rules mining. Then, section IV is concerned with the experiments we have carried out to concretize the proposed approach. The last section concludes the paper and reveales our willingness to continue research for better results.

II. Background and Problem Definition

Definition 1 (Association Rules) An Implication expression having the form of $B \rightarrow H$ where: both B and H are sets of items, and are separate itemsets i.e. $B \cap H = \varnothing$. B is called a premise and H is called a conclusion.

Definition 2 (MDP_{REF} rules) MDP_{REF} is an algorithm which is short for the **M**ost **D**ominant and **Pref**erential rules. It is based on notions of dominance, preference and user profile.

Definition 3 (Loss Rate) Given S_1 and S_2 two set with $S_2 \subseteq S_1$ and $S_1 \neq \varnothing$, we define the loss rate S_2 as compared to S_1 by

$$\text{LossRate}(S_1, S_2) = \frac{|S_1 - S_2|}{|S_1|}$$

Definition 4 (Cost of a partitioning method) Let R = {$Runtime_1(PM),\ldots,Runtime_N(PM)$} be a set of runtime values. $Runtime_j(PM)$ represents the runtime of computing MDP_{REF} rules in the $partition_j$ ($Part_j$) of the database. The operator E denotes the average or expected value of R. Let μ be the mean value of R:

$$\mu = E[R].$$

The cost measure of a partitioning technique is:

$$\text{Cost}(PM) = \sqrt{E\left[(R - \mu)^2\right]}.$$

A large cost value indicates that the runtime values are far from the mean value and a small cost value indicates that the runtime values are near the mean value. The smaller the value of the cost is, the more efficient the partitioning is.

III. Big Data Analysis

A. Distributed machine learning and data mining techniques

Data mining and machine learning hold a vast scope in using the various aspects of Big Data technologies for scaling existing algorithms and solving some of the related challenges [4]. In the following, we present existing works on distributed machine learning and data mining techniques.

a) NIMBLE

NIMBLE [5] is a portable infrastructure that has been specifically designed to enable the implementation of parallel Machine Learning (ML) and Data Mining (DM) algorithms possible.

The NIMBLE approach allows composing parallel Machine Learning and Data Mining algorithms « ML-DM algorithms » using reusable (serial and parallel), building blocks that can be efficiently executed using MapReduce and other parallel programming models. The programming abstractions of NIMBLE have been designed so as to parallelize ML-DM computations and allow users to specify several tasks such as parallel data, parallel tasks and even pipelined computations.

The NIMBLE approach has been used to implement some popular data mining algorithms such as k-Means Clustering and Pattern Growth based Frequent Item set Mining, k-Nearest Neighbors, Random Decision Trees, and RBRP-based Outlier Detection algorithm.

As shown in Fig.1, NIMBLE is divided into four distinct layers:

1) The user API layer, which provides the programming interface to the users. Within this layer, users are able to design tasks and Directed Acyclic Graphs (DAGs) of tasks to indicate dependencies between tasks. A task processes one or more datasets in parallel and produces one or more datasets as output.

2) The architecture independent layer, which acts as the middleware between the user specified tasks/DAGs, and the underlying architecture dependent layer. This layer is responsible for the scheduling of tasks, and delivering the results to the users.

3) The architecture dependent layer, which consists of harnesses providing a means allowing NIMBLE to run portably on various and several platforms. Currently, NIMBLE only supports execution on the Hadoop framework.

4) The hardware layer, which consists of the used cluster.

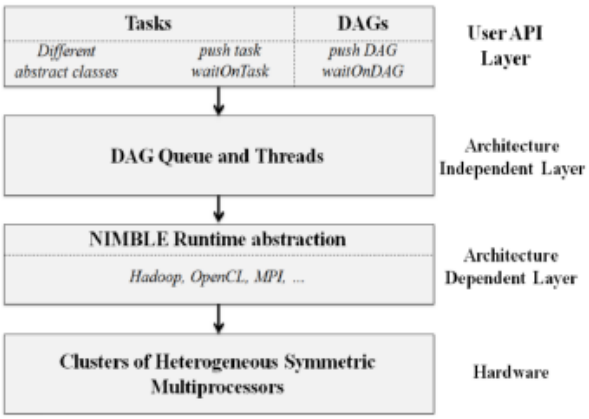

Fig. 1. An overview of software architecture of NIMBLE

b) SystemML

SystemML [6] is a system that enables the development of large scale Machine Learning algorithms. It first expresses a Machine Learning algorithm in a higher-level language called Declarative Machine learning Language (DML). Then, it executes the algorithm in a MapReduce environment.

On the one hand, DML is a system whose main goal is to simplify the usage or development of Machine Learning algorithm, it separates algorithms from data representation and execution plans.

On the other hand, DML language exposes arithmetical and linear algebra primitives on matrices that are natural to express a large class of Machine Learning algorithms.

There are different types of DML such as:

- DML Tasks : (for further clarification please refer to MLbase [18, 21], (fixed task) Columbus [25], DeepDive [20])
- DML Algorithms (fixed algorithm) : (for further clarification please refer to OptiML [23], SciDB [13-22] SystemML [12-16], SimSQL [14])
- Large-Scale ML Libraries (fixed plan) : (for further clarification please refer to MLlib [19], Mahout [24], MADlib [15-17], ORE, Rev R)

As shown in Fig.2, SystemML is classified into four distinct layers:

1) The Language component: It consists of user-defined algorithms written in DML.

2) The High-Level Operator Component (HOP): It analyzes all the operations within a statement block and chooses from multiple high-level execution plans. A plan is represented in a DAG of basic operations (called hops) over matrices and scalars.

3) The Low-Level Operator Component (LOP): It translates the high-level execution plans provided by the HOP component into low-level physical plans on MapReduce.

4) The runtime component: It executes the low-level plans obtained from the LOP component on Hadoop.

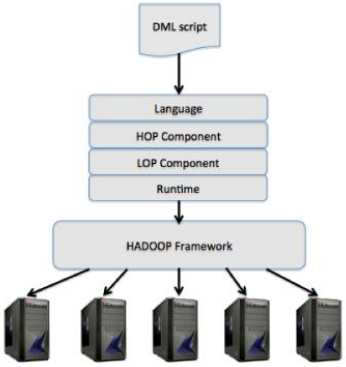

Fig. 2. An overview of software architecture of System ML

a) PARMA

PARMA [7] is a parallel randomized algorithm for mining Frequent Itemsets (FI's) and Association Rules (AR's). PARMA is built on top of MapReduce and the computations are performed twice following the two processing steps of MapReduce. As stressed in Fig.3, the Ellipses represent data, squares represent calculations on that data and arrows show the movement of data through the system.

PARMA creates multiple small random samples of the transactional dataset, at Phase 1 " Map1", and runs a mining algorithm on the samples independently and in parallel, at Phase 2 " Reduce1". The output results from each sample labeled "id", at Phase 3 "Map 2", are aggregated and filtered, at Phase 4 "Reduce 2", to provide a single collection as output which is a global set of Frequent Itemsets and Association Rules. The final result of PARMA is an approximation of the exact solution since it mines random subsets of the input dataset.

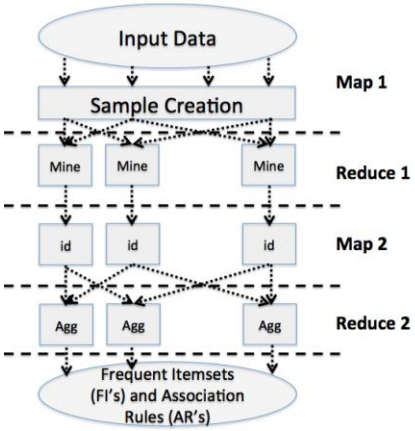

Fig. 3. An overview of the software architecture of PARMA

Table 1 presents the most popular data mining and machine learning techniques. For each technique, it lists the programming model, the implemented techniques and the programming language. We notice that the input and the output of the above presented approaches are user-defined.

TABLE I. OVERVIEW OF DATA MINING AND MACHINE LEARNING TECHNIQUES

Approach	Programming language	Programming model
NIMBLE	Java	MapReduce
System ML	Java and DML	MapReduce
PARMA	Java	MapReduce

IV. OVERVIEW OF THE PROPOSED APPROACH

1) The point of departure is the Association rules set (input) that is first distributed into J partitions (where $1 \le J \le k$) which are processed simultaneously by MDP_{REF} algorithm which is in itself distributed among the J partitions (see Fig. 4).

2) MDP_{REF} –J– is an MDP_{REF} algorithm that we execute in the assigned data partition – J^{th} partition – to generate the corresponding, locally most Dominant and Preferential association rules.

3) The Optimizing step uses the sets of locally Most Dominant and Preferential association rules as input and computes profitability, efficiency and risks of each one of the partitions. Then, it outputs the set of the globally optimal Most Dominant and Preferential association rules, i.e., Association Rules that are undominated, most preferable and efficient ones in the whole association rules set (AR-Set).

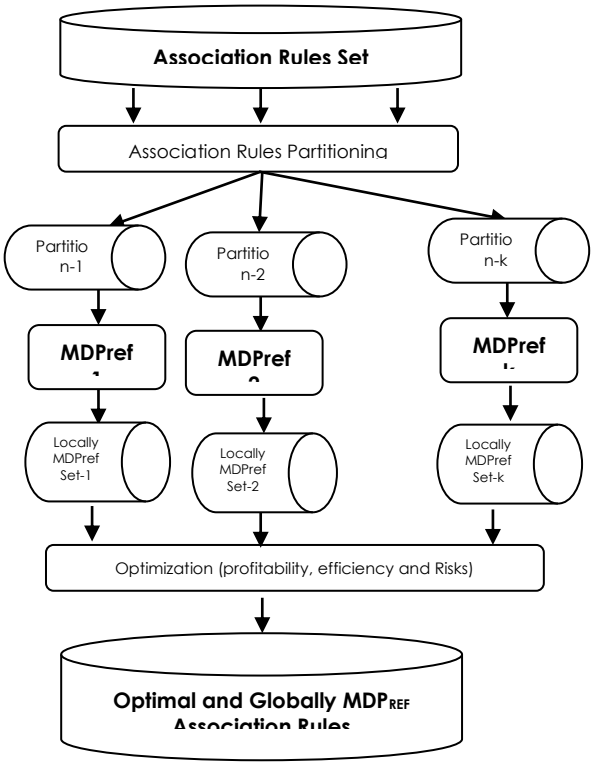

Fig. 4. An overview of the software architecture of our Approach

B. Data partitioning

In the data partitioning step, several techniques intervene to partition the dataset into a number of partitions with respect to particular criteria such as similarity or nearest neighbor criterion. These techniques may involve algorithms using different measures to partition data. It follows that the output (of a technique applied on the input) is a homogenous set, this homogeneity reflects the criterion which this technique uses. Therefore if the output is a group of similar association rules then the criterion used must be that of similarity. If the technique takes into consideration the distance between elements it generates a set of equidistant elements with regard to a determined central point. In our case the input database, a set of Association Rules AR-Set = {AR_1……AR_n}, is partitioned into a user-specified number "k" of partitions. The output is a set of partitions Part(AR-Set) = {$Part_1$(AR-Set)…..$Part_k$(AR-Set)}.

The proposed framework allows many partitioning techniques for the Association Rules Set, like k-Means, k-Medoids, Décision Tree in addition to other partitioning techniques or meta-algorithms like Bagging and Boosting whose objective is to improve predictions, classification and accuracy.

TABLE II. BAGGING AND BOOSTING FEATURES

	Bagging	**Boosting**
Partitioning of data into subsets	Random	Giving mis-classified sample higer preference
Goal to achieve	Minimize variance	Increase predictive force
Methods where this is used	Random subspace	Gradient descent
Function to combine single models	(Weighted) average	Weighted majority vote

Let AR-Set ={AR_1,…..AR_n} be a set of Association Rules. For $1 \le j \le k$, Let $Part_j(AR-Set) \subseteq AR-Set$ be a non-empty subset of AR-Set. We define a partitioning of the database over a k partitions by the following:

Part(AR-Set) = {$Part_1$(AR-Set),…..,$Part_k$(AR-Set)} such that :

- $\cup_{i=1}^{k}\{Part_i(AR-Set)\} = (AR-Set)$
- $\forall_{i \neq j} Part_i(AR-Set) \cap Part_j(AR-Set) = \varnothing$

C. Distributive Association Rules mining

The distributive ARM step mines a set of sub-sets of locally Most Dominant and Preferential association rules named MDP_{REF} Association Rules sub-sets. The input of this step is a partition of the AR-Set : Part(AR-Set) = {$Part_1$(AR-Set),…..,$Part_k$(AR-Set)}. The execution of Distributive Association Rules mining step is resumed by running the MDP_{REF} algorithm on each partion $Part_k$(AR-Set) in parallel.

In the last step, the Optimizing step, we run an algorithm permiting to determine the optimal set formed by the locally MDP_{REF} Association Rules with regard to the minimization of "Risks" and to the maximization of the " profitability – efficiency " of Association Rules.

D. MDP_{REF} Algorithm

MDP_{REF} stands for the **M**ost **D**ominant and **Pref**erential rules. It is based on dominance, preference and user profile. Besides being threshold-free, MDP_{REF} solves the subjectivity problem and keeps all measures so as no information would be lost. Its main goal is to successfully discover, filter and prune **AR** into subsets verifying a two-sided criterion. That is to say, each rule in a subset must meet two conditions; it has to be the most dominant as well as the most preferred by the user. To get at the above-mentioned objective, the algorithm takes into account the factor of time during the processing of the following tasks [8]:

- Creates a referential rule (r^T) which dominates all the rules (Having the maximum measurements);
- Computes the degree of similarity of all the rules one by one with the referential rule (r^T);
- Determines the dominant rule r* (which has a minimal "degree of similarity" with referential rule (r^T));
- Discards all the rules dominated by r*;
- If two rules are equivalent, we resort to the user's preferences to determine which one to keep;
- Keep both if the decision maker is indifferent in regard to the equivalent rules, otherwise we keep the one satisfying more preferences;
- Drop all rules where the user's preferences are already covered by those previously handled;

- Keep Rules covering the user's preference other than those already covered by those previously selected [9].

ALGORITHM: "MDP$_{REF}$" Algorithm

1.0 **Input :** Set of Rules+Set of Measures+Preference Set Ω (R, M, P_{ref})
2.0 **OutPut:** The Most Dominant and Preferential Rules **MDP$_{REF}$**
3.0 **Begin**----------
4.0 *MDPref* ⟵ Ø
5.0 *C* ⟵ R
6.0 **while** *C* ≠ Ø *do*
7.0 Create a referential rule r^T having a max of measure value
8.0 *r** ⟵ *r* ∈ *C having a* min (DegSim (r, r^T))
9.0 **For** (i=1 to k=| *C* |) **do**
10.0 *MDPref* ⟵ *MDPref U { r* }*
11.0 *C* ⟵ *C* \ *{ r* }*
12.0 **Foreach** r_i ∈ *C* **do**
13.0 **If** *r** > r_i
14.0 **then**
15.0 C ⟵ C \ { r_i}
16.0 **else**
17.0 **For** (j=1 to k') **do**
18.0 **If** $r_i[m_j] \geq r^*[m_j]$
19.0 **then**
20.0 ~~*MDPref*~~ *MDPref U { r_i}* ,
21.0 ⟵ *r** r_i
22.0 **else**
23.0 *(r_i is equivalent of r*)*
24.0 *S* ⟵ *set of equivalent rules*
25.0 *Z* ⟵ Ø
26.0 **while** *S* ≠Ø
27.0 $Z_{best}(r_i)$= *max* ◁ *S « r_i the most preferred rule in S »*
28.0 *Z= Z* ∪ $Z_{best}(r_i)$
29.0 P_{ref} ⟵ *{<t,u> ∈ P_{ref} / t* ◁ ≠ Z_{best} *u the preferences engendered by Z_{best}}*
30.0 *S* ⟵ *S* \ *{ $Z_{best}(r_i)$ }*
31.0 **end**
32.0 **end**
33.0 **end**
34.0 **end**
35.0 **end**
36.0 **end**
37.0 **end**
38.0 *MDPref* ⟵ *MDPref U { Z }*
39.0 **end**
40.0 **Return** *MDPref*
41.0 **end**

Concerning the quality of rules, it is assessed by the two measures of dominance and preference which are inherent in the algorithm. The first one is a statistical measure and the second one is subjective – related to users. Rules failing to meet these two measures are not mined.

V. EXPERIMENTS

This section presents an experimental study of the proposed approach on real datasets. First, it describes the datasets that have been used and the details of implementation. Then, it introduces a discussion of the results.

A. Experimental setup

The datasets used in the experimental study are presented in the table III, the proposed approach use six datasets, Diabete, Flare, Iris, Monks, Nursery, and Zoo taken from "UCI Machine Learning Repository". Each dataset deals with a particular domain such as human health, animals, and agriculture defined by an item count, transaction count and all counts of the association rules.

TABLE III. DESCRIPTIVE OF DATA SETS

DS	Dataset	#Item	# Transaction	#All Rules
DS1	Diabete	75	3196	62132
DS2	Flare	39	1389	57476
DS3	Iris	150	8124	440
DS4	Monks	57	415	181464
DS5	Nursery	32	12960	71302
DS6	Zoo	42	101	25062

Recalling that these experiments have been applied on a machine that has the following characteristics: 1.73 GHz and a memory capacity of 2GB.

The Fig.5 illustrates the effect of the proposed partitioning method on the rate of lost association rules. We can easily see in Fig.5 that the proposed partitioning method allows low values of loss rate especially with low values of tolerance rate.

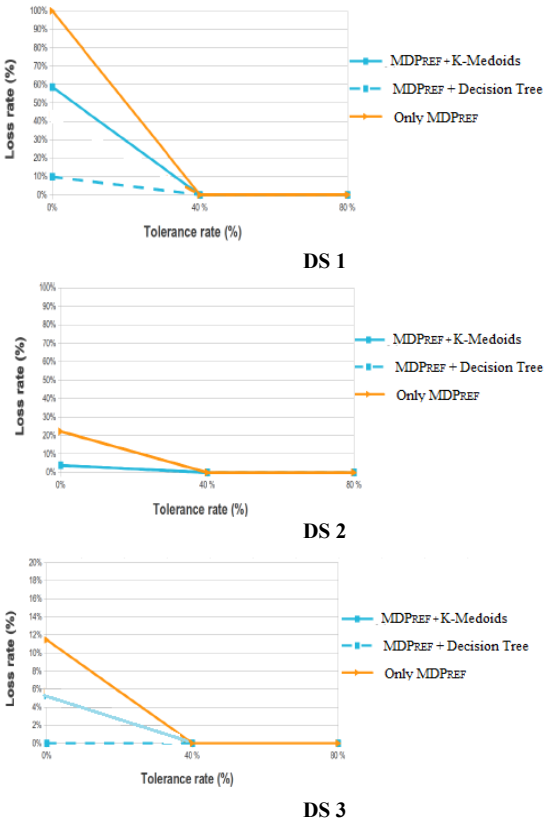

Fig. 5. Effect of partitioning method on the rate of lost Association Rules

In order to study the scalability of the proposed approach and to show the impact of the number of used machines on the large scale Association Rules mining runtime, the Fig.6 present the runtime of the proposed approach for each number of MDP$_{REF}$ (i) machines.

As illustrated in Fig.6, the proposed approach scales with the number of machines. In fact, the execution time of the proposed approach is proportional to the number of machines.

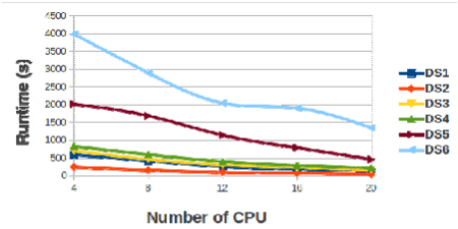

Fig. 6. Effect of the number of workers on the runtime. using K-Medoids as a partitioning method, MDP_{REF} as an association rules extractor

In order to evaluate the influence of some parameters on the performance of the proposed implementation, the block size is varied and computed the runtime of the distributive Association Rules mining process of the proposed approach. In this experiment, six datasets are used and the chunk size is varied from 10MB to 100MB.

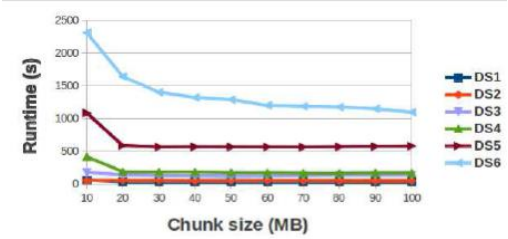

Fig. 7. Effect the varation the chunk size on the runtime. using K-Medoids as a partitioning method, MDP_{REF} as a association rules extractor

The experiment presented in Fig. 7 shows that as long as the data set is large the results are not notably affected no matter how big or small the chunk size values may be. Otherwise, the other values of chunk size do not notably affect the results.

VI. Conclusion

In this paper, we addressed the issue of the distributive Association Rules Mining process. We have described the proposed approach for large-scale association rules mining from large-scale association rules sets. The proposed approach relies on clustering / classification methods to build partitions of an association rules set in order to select the locally MDP_{REF} rules for each partition via applying any of the known "clustering / classification" algorithms . Then we apply an optimization algorithm [used in the last step of the distributive ARM process, see section V.B above] to extract a globally optimal MDP_{REF} rules. By running experiments on a variety of datasets, we have shown that the proposed method decreases significantly the runtime. Moreover, it may be functional in the case of large scale databases. This is a significant part of the future work to make sure that the performance and scalability of the proposed approach encompass also big databases.

References

[1] U.M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, 1996. Knowledge Discovery and Data Mining: Towards a Unifying Framework. In KDD 1996., 1996

[2] M. Goebel and L. Gruenwald "A survey of data mining and knowledge discovery software tools". SIGKDD Explor. Vol.1, Issue 1, pp.20–33, 1999. doi: 10.1145/846170.846172

[3] M. Mouhir, Y. Balouki, T. Gadi and M. El Far " A new way to select the valuable association rules", in KST 2015: Proceedings of the Knowledge and Smart Technology (KST), 2015 7th International Conference on, 2015 7th KST IEEE, Chonburi, Thailand, pp. 81-86.Junary 2015. doi: 10.1109/KST.2015.7051464

[4] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha. "Eficient machine learning for big data: A review. Big Data Research", Vol. 2, No.3,pp.87- 93, 2015.

[5] A. Ghoting, P. Kambadur, E. Pednault, and R. Kannan. "Nimble: a toolkit for the implementation of parallel data mining and machine learning algorithms on mapreduce". In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '11, pp. 334-342, New York, NY, USA, 2011. ACM.

[6] A. Ghoting, R. Krishnamurthy, E. Pednault, B. Reinwald, V. Sindhwani, S. Tatikonda, Y. Tian, and S. Vaithyanathan, "SystemML: Declarative machine learning on Mapreduce". In Proceedings of the 2011IEEE 27th International Conference on Data Engineering, ICDE '11, pp. 231-242, Washington,DC, USA, 2011. IEEE Computer Society.

[7] M. Riondato, J. A. DeBrabant, R. Fonseca, and E. Upfal. "Parma: a parallel randomized algorithm for approximate association rules mining in mapreduce" in Proceedings of the 21st ACM international conference on Information and knowledge management, CIKM '12, pp. 85-94, New York, NY, USA, 2012. ACM.

[8] M. Mouhir, A. Dahbi, Y. Balouki and T. Gadi. "$S_{EM}MDP_{REF}$: Algorithm to filter and sort rules using a semantically based ontology technique" in ACM-MEDES 2015: Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems - ACM-MEDES'15, Caraguatatuba, Sao Paulo, Brazil, pp. 29-34, October 2015. doi: 10.1145/2857218.2857223

[9] M. Mouhir, Y. Balouki and T. Gadi. "Selecting and Filtering Association Rules within a Semantic technique," International Review on Computers and Software (I.RE.CO.S.), Vol. 11, No6, June 2016, pp. 530-538. doi:http://dx.doi.org/10.15866/irecos.v11i6.9556

[10] M. Kamber, and J. Han, Data Mining: Concept and Techniques, 2nd ed., Vol. 6, The Morgan Kaufmann Series in Data Management Systems, Munich, Germany.2006

[11] S.A. Mahmoodi, K. Mirzaie and S.M. Mahmoudi. "A new algorithm to extract hidden rules of gastric cancer data based on ontology". SpringerPlus, Vol.5, No 312, 2016. doi:10.1186/s40064-016-1943-9.

[12] M. Boehm D. R. Burdick, A. V. Evfimievski, B. Reinwald, F. R. Reiss, P. Sen and Y. Tian. SystemML's Optimizer: Plan Generation for Large-Scale Machine Learning Programs. IEEE Data Eng. Bull., Vol 37, No 3, 2014.

[13] P. G. Brown. Overview of SciDB: Large Scale Array Storage, Processing and Analysis. In SIGMOD, 2010

[14] Z. Cai, Z. Vagena, L. L. Perez, S. Arumugam, P. J. Haas, and C. M. Jermaine. Simulation of Database-Valued Markov Chains Using SimSQL. In SIGMOD, 2013.

[15] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, and C. Welton. MAD Skills: New Analysis Practices for Big Data. PVLDB, Vol 2, No 2, 2009.

[16] A. Ghoting, R. Krishnamurthy, E. Pednault, B. Reinwald, V. Sindhwani, S. Tatikonda and S. Vaithyanathan. SystemML: Declarative Machine Learning on MapReduce. In ICDE, 2011

[17] J. M. Hellerstein, C. Ré, F. Schoppmann, D. Z. Wang, E. Fratkin, A. Gorajek and A. Kumar. The MADlib Analytics Library or MAD Skills, the SQL. Proceedings of the VLDB, Vol 5, No 12, 2012.

[18] T. Kraska, A. Talwalkar, J. C. Duchi, R. Griffith, M. J. Franklin, and M. I. Jordan. MLbase: A Distributed Machine-learning System. In CIDR, 2013

[19] X. Meng, J. Bradley, B. Yuvaz, E. Sparks, S. Venkataraman, D. Liu and D. Xin. MLlib: Machine Learning in Apache Spark. JMLR, Vol.17, No. 34, pp. 1-7, 2016.

[20] J. Shin, S. Wu, F. Wang, C. D. Sa, C. Zhang, and C. R´e. Incremental Knowledge Base Construction Using DeepDive. PVLDB, Vol 8, No11, 2015.

[21] E. R. Sparks, A. Talwalkar, D. Haas, M. J. Franklin, M. I. Jordan, and T. Kraska. Automating Model Search for Large Scale Machine Learning. In SOCC, 2015.

[22] M. Stonebraker, P. Brown, A. Poliakov, and S. Raman. The Architecture of SciDB. In SSDBM, 2011.

[23] A. K. Sujeeth, H. Lee, K. Brown, T. Rompf, H. Chafi, M. Wu and K. Olukotun. OptiML: An Implicitly Parallel Domain-Specific Language for Machine Learning. In ICML, 2011.

[24] The Apache Software Foundation. Mahout

[25] C. Zhang, A. Kumar, and C. Ré. Materialization Optimizations for Feature Selection Workloads. In SIGMOD, 2014.