EbIDAM: Efficient Data Mining Java Library

George Gatuha, Tao JIANG

Abstract— **Pattern generation in transactional databases is an important Big Data mining problem. Developing efficient systems that are able to handle large volumes of data therefore becomes unique. Finding all frequent item sets and association rules in a given data set requires substantive computing resources; in this paper we implement EbIDAM, a Java data mining library for efficiently processing datasets. Three frequent item set data mining algorithms (Apriori, ECLAT, and FPGrowth) and two association rule algorithms (IGB and MMR) have been incorporated in EbIDAM. We compared task execution times of the algorithms within EbIDAM, and with other popular open source data mining applications, WEKA and Coron. Within EbIDAM, FPGrowth frequent item sets algorithm outperformed Apriori and ECLAT, while association rules mining IGB outperformed MMR. EbIDAM's FPGrowth outperformed WEKA's FPGrowth while ECLAT in EbIDAM outperformed Coron's ECLAT. Breast cancer and retail shop datasets were used for the exercise.**

Keywords: Big Data mining, association rule algorithms, open-source, frequent item set, transactional database.

I. INTRODUCTION

Data mining is the combination of machine learning techniques, statistical data analysis and artificial intelligence to uncover 'valid, novel, potentially useful, and ultimately understandable patterns in data' [1]. The technique has been successfully applied in banking, telecommunication, credit card fraud detection, market analysis and lately medicine [2] among many other areas. Application of Data mining in healthcare aims at describing disease symptoms as patterns which may appear in data and provides an additional source of knowledge for making decisions to healthcare professionals. These patterns may be used to support future decisions concerning disease diagnosis, treatment planning, and risk analysis, among many other uses [3].

The development of advanced ICT systems and techniques has led to the development of modern ways of disease diagnosis and treatment planning in healthcare [4]. The healthcare industry generates big amount of complex data concerning patient records, disease diagnosis, medical devices, hospitals resources, health insurance fraud detection among many other areas. These datasets provides a key resource for data mining to extract knowledge that enables support for policy and decision making [5]. The hidden information contained in these databases is enormous, and therefore focus is to sift through these records and extract hidden information [6].

Lack of adequate open source data mining projects specifically dealing with frequent pattern mining led us to develop EbIDAM. Frequent pattern mining aims at discovering patterns and associations in datasets. There exist several data mining applications [7] such as Knime, WEKA, and Mahout; however, they only cover a few frequent pattern mining algorithms such as FPGrowth, and Apriori. Other software's such as Illimine, Coron, LUCS KDD cover more pattern mining algorithms; however, they have restrictions on re-distribution and require an initial investment before usage.

This work has three main advances. Firstly, we implement EbIDAM, a java library containing five data mining algorithms namely: IGB, FPGrowth, MMR, ECLAT, and Apriori. These algorithms have not been implemented together in popular open source data mining software's. The uniqueness of EbIDAM is the fact that only one of its algorithms is implemented in Mahout, three in Knime and two in both WEKA and LUCS-KDD. It is also important to note that LUCS-KDD has been inactive and the reliability (bug free code) of Illumine is in question. Secondly we compare performance of the five algorithms with similar implementation in other popular data mining programs, WEKA and Coron on breast cancer and retail shop datasets. Thirdly we show that EbIDAM outperforms both WEKA and Coron in terms of utilization of computing resources.

EbIDAM can be used in a variety of domains such as sales forecasting, clinical text retrieval, web usage mining, restaurant recommendation, anomaly detection in medical treatment and forecasting crime incidents among many other areas. The program is provided as free software that respects user's freedom to copy, run, distribute, change and do some improvements on the software. Thus freeware is not a matter of the price tag but liberty. It is not free as in free lunch but free as in freedom of speech.

The paper is arranged as follows: Section 2 discusses previous data mining libraries that have been developed over the years. Section 3 discusses design and development of the application, various data mining algorithms contained in the application and the data sets used for evaluation. Experimental results and analysis of performance is discussed in section 4. Section 5 concludes the work and provides a glimpse of future research direction.

Manuscript received August 14, 2015; revised October 20, 2015. This paper is funded by the International Exchange Program of Harbin Engineering University for Innovation-Oriented Talents Cultivation.

George Gatuha & Tao Jiang are with the College of Information and Communication Engineering, Harbin Engineering University, China, Address: NO.145(150001), Nantong Street, Nangang District, Harbin, Heilongjiang Province, China; e-mail: ggrpb@yahoo.com, jiangtao@hrbeu.edu.cn.

II. Literature review

WEKA (Waikato Environment for Knowledge Analysis) is an open source data mining project for in-depth data analyses. It was developed using java programming language by the University of Waikato in New Zealand [8]. It covers a wide range of machine learning tasks, but it has limited algorithms for association rules, it only implements Apriori algorithm.

Knime was developed at the University of Konstanz, by a team of developers from a Silicon Valley software company specializing in pharmaceutical applications. They initially wanted a product for use in the pharmaceutical industry that was capable of processing huge amounts of diverse data. The first version of KNIME was released in 2006 [9].

Apache Mahout was developed by a team of volunteer developers and supports mainly three data mining areas, classification, clustering and recommendation. Recommendation mining addresses the user preferences from user's historical behaviors and predicts what users are interested in. Clustering endeavors to relate similar items into distinct groups. Classification is a supervised data mining technique that groups data into classes, it is best illustrated by the ID3 decision tree algorithm [10].

IlliMine was developed by the University of Illinois at Urbana-Champaign's Computer Science department. Most of the code and experimental data emanate from research papers from international conferences and journals. However, it has some restrictions and non-disclosure agreements on some software and therefore not very reliable as a platform [11].

Coron is a platform independent data mining project developed by Laszlo Szathmary, by then a PhD student it was first released on July 18, 2004. It is designed specifically for item sets extraction and association rule mining. It also supports data preparation and filtering; by 2005 the platform contained nine algorithms, namely Pascal, Pascal+, Apriori, Apriori-Close, ECLAT, Close, Titanic RMS Carpathia, and Zart [12].

III. Design and Development

EbIDAM is implemented in Java programming language and is a platform independent project. The application requires a minimum of Java version 7 installed in the client's system. Help documentation provides a detailed explanation of how to run the program. It also gives a detailed explanation on the expected input and output of the algorithms and their main characteristics. The application is centered on two important data mining techniques.

Frequent item sets mining aims at discovering item sets having support above minimum threshold. Support is a fraction of the transactions containing an item set and the total number of transactions. Item sets having support that is above minimum are referred to as large item sets and those that are below minimum are small item sets [13].

Association rule mining intends to identify strong relationships in databases. The algorithm was proposed in [14] for discovering regularities between products bought at a supermarket. An association rule can be determined by use of support and confidence. Support determines how often an item set appears in a relational database. Confidence is a measure of how often X item appears in a transaction that also contains Y item.

The five data mining algorithms evaluated were *IGB, FPGrowth, MMR, ECLAT,* and *Apriori*. We implemented the above algorithms in java programing language and tested the software thoroughly using frequently used data analysis datasets. The data used for evaluation is also described in detail later in this section.

Apriori item set algorithm for mining frequent item sets is the best illustration of Association rule mining. There are two main steps in Apriori. The candidates are generated by joining the frequent item sets level-wise. Infrequent item sets are discarded if their support is below the minimum support. Apriori algorithm is a relatively straightforward algorithm and is easy to implement. However, it is a very computationally expensive algorithm as it scans the database several times to generate large item sets and stores all the item sets in memory including infrequent item sets [15].

FPGrowth item set algorithm is an alternative to Apriori algorithm that takes a radically different approach to discovering frequent item sets. The database is scanned and generates a set of frequent item sets in a descending order, it scans the database again and compresses the database into an FP-tree which is then mined to obtain frequent item sets by recursively working on the results and pruning infrequent items. FPGrowth only scans the database twice and is therefore an efficient algorithm compared to Apriori algorithm [16].

ECLAT algorithm (Equivalence Class Transformation) algorithm is a depth-first search data mining algorithm that uses transaction id (tid) intersection and computes support of item sets and avoids generation of subsets not existing in the prefix tree [17].

IGB association rule algorithm (Informative Generative Base) is a sound and an informative, generic base association rule algorithm. The soundness property of the algorithm is because of the ability to assess syntactic derivation and the generic base ensures the generation of valid association rules. The support and confidence can be determined easily, hence the informative property [18].

MMR association rule algorithm (Minimum Condition Maximum Consequence Rules) is an algorithm that attempts to limit number of association rule to novel, interesting and original ones. It considers two approaches for determining the interestingness of association rules in a database. Association rules are considered to be interesting if they can be predicted as much as possible from minimal facts. The second approach consists of getting least association rules which are referred to as representative association rules and deduce other association rules without accessing the database [19].

Retail datasets represents retail transactions from a Belgian supermarket. The data was collected in 5 months or there about between December 1999 and November 2000. A total of 5,133 customers visited the retail shop datasets and 88,163 items were purchased in the entire period. The information contains five attributes namely, customer identification number, date of purchase, article number, article price, and items purchased [20].

Breast cancer datasets used in our evaluation is obtained from the UCI machine learning repository. The datasets were provided by Ljubljana University's Medical Center

Institute of Oncology, Yugoslavia. It comprises 286 instances and 9 attributes [21].

Each of the five algorithms is implemented in its own package and no other dependencies on external software libraries and therefore the Java code can be integrated into other Java based programs. Fig.1 is the GUI representation of EbIDAM.

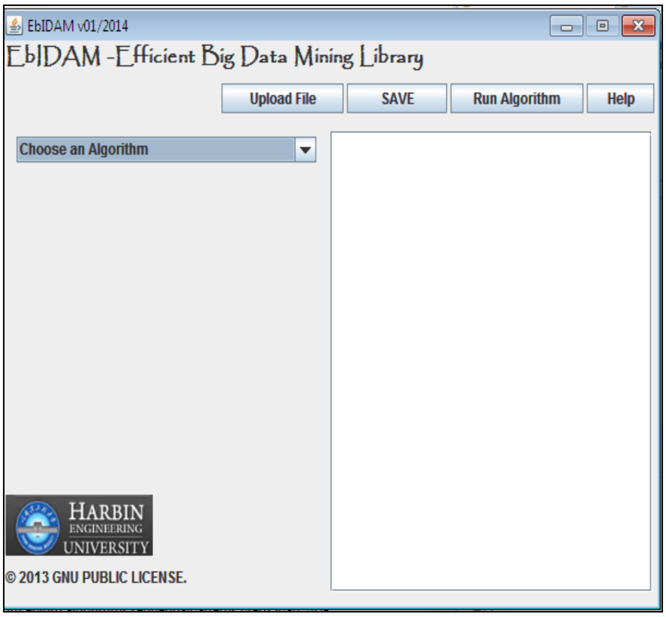
EbIDAM v01/2014
EbIDAM -Efficient Big Data Mining Library
Upload File
SAVE
Run Algorithm
Help
Choose an Algorithm
HARBIN ENGINEERING UNIVERSITY
© 2013 GNU PUBLIC LICENSE.

Fig. 1. EbIDAM GUI

IV. Results and Analysis

Toshiba L635 Laptop having 5 GB RAM and a core i3 1st generation processor, running Windows 7 ultimate operating system and Java version 8 was used for the experiment.

In the first experiment we compared Apriori, ECLAT and FPGrowth execution time in EbIDAM. Breast cancer datasets obtained from UCI machine learning repository were used for the exercise.

The execution time versus the minimum support was compared on all the algorithms. Minimum support was set at 0.3%, 0.4%, 0.5%, 0.6% and 0.7%. The results are outlined in Fig2.

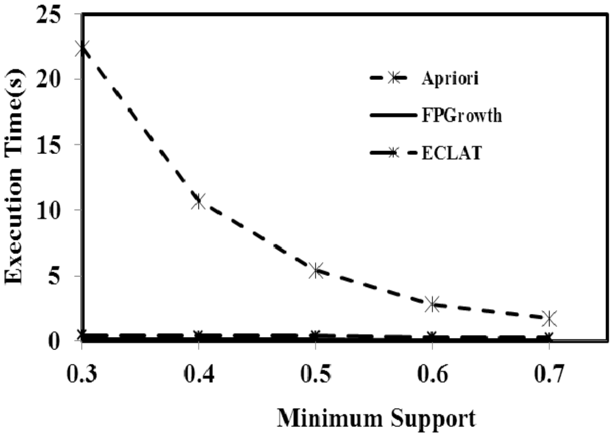

Fig. 2. Comparison of Apriori, ECLAT and FPGrowth algorithms execution time.

For clarity we inflated FPGrowth and ECLAT execution time above and generated the fig 3 below.

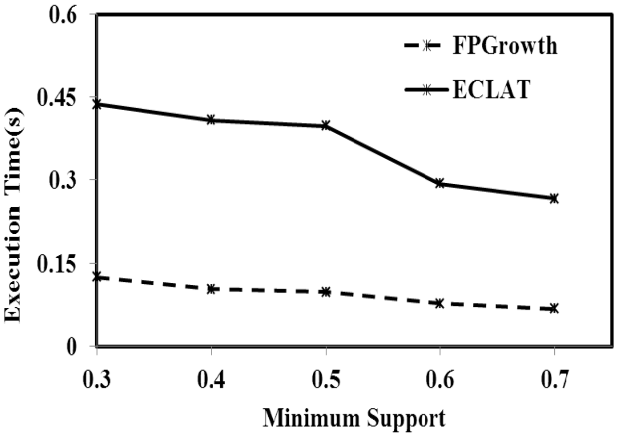

Fig. 3. Comparison of ECLAT and FPGrowth algorithms execution time.

From the experimental results we note that the execution time of Apriori is large compared to ECLAT and FPGrowth for small minimum support, however as the minimum support increases there is a sharp decrease in apriori execution time. This is due to the fact that Apriori algorithm scans the database several times as compared to the other algorithms. FPGrowth algorithm out performs the other algorithms by a big margin.

The second experiment dwelt on comparing performance of association rule mining algorithms MMR and IGB. Breast cancer dataset were used for the exercise. Fig 4 below represents the results. Two steps are needed for association rule mining. The first is discovering frequent item sets (fig4) and then we generate closed association rules (fig5) from the frequent item sets.

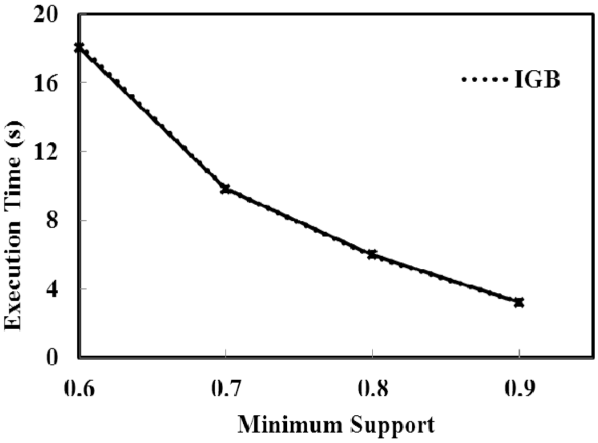

Fig. 4. IGB and MMR association rule algorithms execution time.

From the results the time to generate frequent item sets is almost similar between the two algorithms although MMR performs slightly better than IGB algorithm.

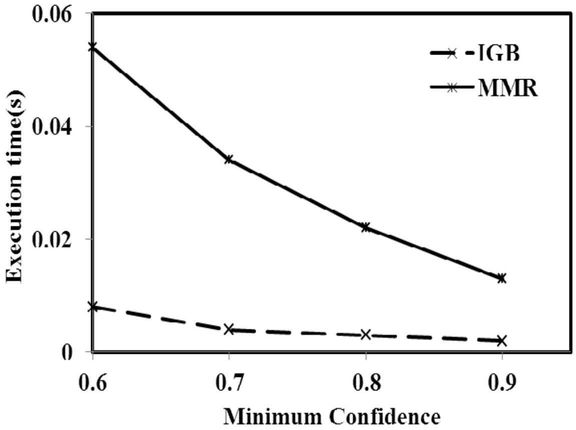

Fig. 5. IGB and MMR association rule generation execution time.

In the second step of association rule generation, IGB performs better than MMR.

In the third experiment we compared Apriori and FPGrowth item set algorithms execution time between WEKA and EbIDAM. Retail shop datasets were used for the exercise. The results are outlined in Fig6 and Fig7 for both Apriori and FPGrowth respectively.

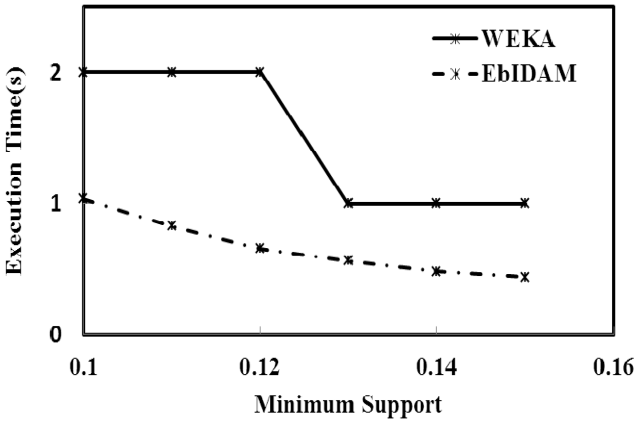

Fig. 6. Comparison of FPGrowth algorithm for both EbIDAM and WEKA

From the above results the FPGrowth algorithm implementation in EbIDAM outperformed WEKA's FPGrowth by a big margin.

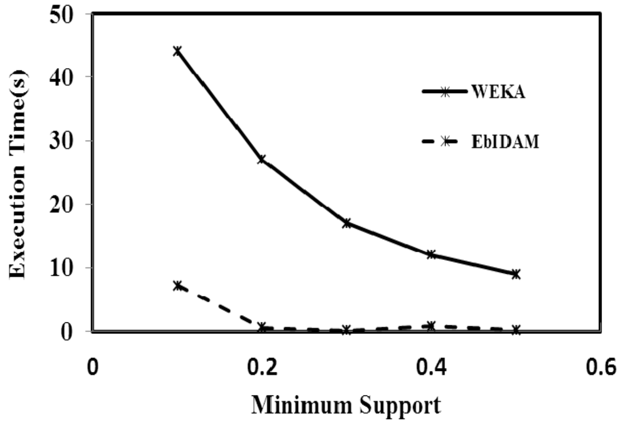

Fig. 7. Comparison of Apriori algorithm for both EbIDAM and WEKA

From the above results EbIDAM was found to be faster than WEKA in terms of execution time by a big margin.

In the fourth experiment ECLAT algorithm was used for comparison, it is implemented in Coron and therefore the performance was compared with that of EbIDAM, the results are as displayed in Fig8. The install package of Coron includes only batch files for launching Command Line Interface which is the primary interface of Coron. However they are written in bash (.sh), under Linux environment they pose no problem but in windows they require a Cygwin environment. Coron writes the results to disk while EbIDAM writes to a text file and therefore we had to modify Coron to output to a text file.

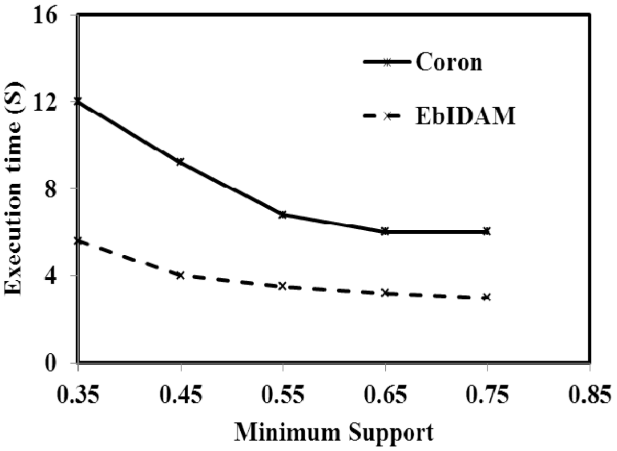

Fig. 8. Eclat algorithm for both EbIDAM and Coron.

From the above results EbIDAM was found to be faster than Coron by a big margin.

V. Conclusion

In this paper we have presented EbIDAM, data mining software that is highly scalable and specialized for frequent pattern mining. The application can be easily incorporated in other Java projects because the algorithms are extended from independent packages and can be modified to suit individual needs. To have a better understanding of the performance characteristics of EbIDAM's five algorithms, we compared the performance of its algorithms in-house and with other popular data mining software's. The first experiment involved the three frequent item set mining algorithms Apriori, ECLAT and FPGrowth, the results showed that FPGrowth is the most efficient in task execution time followed by ECLAT and finally Apriori was found to be the slowest. The second experiment involved comparison of IGB and MMR association rule algorithms implementation in EbIDAM, MMR outperformed IGB by a small margin in the first step of closed item set generation while in the second step of association rule generation IGB outperformed MMR algorithm by a big margin. The third experiment involved comparison with WEKA, EbIDAM was more efficient in terms of computational time in both Apriori and FPGrowth algorithms. The fourth experiment involved comparison of ECLAT algorithm's task execution time between Coron and EbIDAM the results shows that EbIDAM was more efficient by a big margin.

Future research direction would be to include other data mining techniques such as Classification and Clustering algorithms. We also intend to include more association mining algorithms that are new and more efficient.

ACKNOWLEDGMENTS

This paper is funded by the International Exchange Program of Harbin Engineering University for Innovation-Oriented Talents Cultivation.

REFERENCES

[1]. U. M. Fayyad, “Data mining and knowledge discovery: Making sense out of data”, *IEEE Expert*, vol. 11, no. 5, 20-25, 1996.

[2]. W. Duch, K. Grabczewski, R. Adamczak, K. Grudzinski, Z.S. Hippe, “Rules for melanoma skin cancer diagnosis”, *Komputerowe Systemy Rozpoznawania,* KOSYR, Wrocław, 59-68, 2001.

[3]. A. Tsymbal, N. Bolshakova, “Guest Editorial Introduction to the Special Section on Mining Biomedical Data”, *IEEE Transactions On Information Technology In Biomedicine,* vol. 10, no. 3, 425-428 ,2006.

[4]. G., Gatuha , T., Jiang. “ KenVACS: Improving vaccination of children through cellular network technology in developing countries”, *Interdisciplinary Journal of Information, Knowledge and Management* , vol:10, p 37-46, 2015.

[5]. Y. M. Chae, H. S. Kim, K. C. Tark, H. J. Park, S. H. Ho, “Analysis of healthcare quality indicator using data mining and decision support system”, *Expert Systems with Applications*, 167–172, 2003.

[6]. Cheng S. “Knowledge Discovery In Databases: An Information Retrieval Perspective”, *Malaysian Journal of Computer Science*, Vol. 13 No. 2, pp. 54-63, 2000.

[7]. J. Han, M. Kamber. *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann, 2006.

[8]. I. H. Witten, E. Frank, *Data Mining, Practical Machine Learning Tools and Techniques, 2nd Edition*. Morgan Kauffmann, San Francisco, 2005.

[9]. R. B. Michael, C. Nicolas, D. Fabian , R. Thomas, K. Tobias, M. Thorsten,….. and W. Bernd, “KNIME: The Konstanz Information Miner, Studies in Classification, Data Analysis, and Knowledge Organization”, *Springer*, 2007.

[10]. A.R. Curtis, K. Wonho, P. Yalagandula. "Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection", *INFOCOM, Proceedings IEEE*, On page(s): 1629 – 1637, 2011.

[11]. M. Gupta, J. Gao, Y. Sun, and J. Han. “Integrating Community Matching and Outlier Detection for Mining Evolutionary Community Outliers”, In *Proc. of the 18th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, 2012.

[12]. M. Kaytoue, F Marcuola, A. Napoli, L Szathmary and Jean Villerd.”The Coron System “ arXiv preprint arXiv:1111.5690, (2011)

[13]. R. Bayardo, B. Goethals, M. J. Zaki, editors, *Proceedings of the IEEE International Conference on Data Mining Workshop on Frequent Itemset Mining Implementations* (FIMI’04), Brighton, UK , 2004.

[14]. R. Agrawal, T. Imielin´ ski, and A. Swami “Mining association rules between sets of items in large databases”, *In Proceedings of the ACM SIGMOD International Conference on Management of Data. AC*M, 1993.

[15]. R. Agrawal, and R. Srikant. “Fast algorithms for mining association rules”, In *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487–499, 1994.

[16]. J. Han, J. Pei, and Y. Yin. “Mining frequent patterns without candidate generation.” In *Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM*, 2000.

[17]. M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. “New algorithms for fast discovery of association rules”, In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Min*ing, pp. 283–286 ,1997.

[18]. G. Gasmi , S. B. Yahia, E. M. Nguifo , Y.Slimani “A new generic base association rules”, *the ninth pacific- Asia Conference on Knowledge Discovery and Data mining (PAKDD-05), LNCS, Springer Verlag,* Hanoi, Vietnam ,2005.

[19]. M. Kryszkiewicz (1998): “Representative Association Rule and Minimum Condition Maximum Consequence Association Rules”, In *Proceedings of PKDD-98. Nantes, France. LNAI 1510. Springer-Verlag* 361-369, 1998.

[20]. T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets., “the use of association rules for product Assortment decisions: a case study”, In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining,* San Diego (USA), August 15-18, pp. 254-260. ISBN:1-58113-143-7,1999.

[21]. Asuncion, A & Newman, D.J. *UCI Machine Learning Repository Irvine, CA*: University of California, Department of Information and Computer Science, 2007.