Comparative of Data Base Evolution in Rule Association Algorithms in Incremental and Conventional Way

Euclides Peres Farias Jr., Júlio Cesar Nievola

Abstract— **Many results in the literature indicate that the incremental approach to association mining leads to gain regarding the time needed to obtain the rules, but there is no evaluation about their quality, compared to non-incremental algorithms. This paper presents the comparison of usage of two typical algorithms representing each approach: APriori and ZigZag. Execution time clearly shows the advantage of incremental approaches, but when someone needs accurate results concerning the association rules obtained, the matter should be taken with more caution, because the rules obtained are not necessarily in a relation one-to-one, according to the results obtained.**

I. Introduction

ATTRIBUTES that change over time in the association task have prompted the development of incremental techniques, in order to evaluate whether the extracted patterns are static or dynamic. These techniques are also useful, when the data used are so huge that traditional algorithms are not able to take it as a whole. So, when faced with the latter, one looks not only for results in an acceptable time, but also with a quality that looks similar as using the conventional algorithms.

Many results in the literature indicate that the incremental approach to association mining leads to gain regarding the time needed to obtain the rules [1]. Here the main question is to verify if extracted rules are of acceptable quality. In order to proceed, two known algorithms representing both approaches were chosen to be compared. APriori is, probably, the best known algorithm of association rule extraction [2]. It works in a batch mode, this means, it takes the whole database and extracts the association rules at once. For the sake of comparison, algorithm ZigZag was chosen as the incremental approach [3], since it is a modification of APriori, using only the maximal frequent itemsets to build the association rules [7].

This paper is organized in the following way. Section 2 provides the foundation for both algorithms used: APriori and ZigZag, introducing general concepts, presenting common notation and discussing the main characteristics. This is followed, in Section 3, by the description of the dataset used in the experiments. Section 4 deals with the procedures taken to evaluate the quality of the association rules generated by both algorithms, while Section 5 presents the results and compares the rules. Finally, Section 6 completes the paper with the differences in terms of rules, common ground and future works.

II. Association Rules

There are two ways to look at attribute associations: a)on the attribute-level, i.e., looking for statistical dependencies between the attributes (e.g., using graphical models); and b)on the value-level, where association rules are the premier tool for this class of problems.

Association mining usually takes two steps: identification of frequent sets within the dataset and the derivation of the association rules from the frequent sets (or itemsets). Major differences among existing algorithms lies in the first phase of the process, as its level of complexity is greater than the latter. Some algorithms (like APriori) focused on the discovery of all possible association rules, while others (including ZigZag) try to improve processing time or facilitate user interpretation by reducing result set size [4].

Given thresholds minsup and minconf for the support and confidence, these algorithms compute associations rules whose support and confidence exceed these thresholds. Itemsets with support at least minsup are called frequent. The support of an item A is the percentage of transactions in a dataset that contains A, being thus a statistical significance. The support calculation can be resumed according to formula (1):

$$\sup\left(A \rightarrow B\right) = \frac{\sum\left\{t \in D \middle| \left(A \cup B\right) \subseteq t\right\}}{\sum\left\{t \in D\right\}} \tag{1}$$

Where t is a transaction and D is the set of transactions in the database. Confidence, also known as strength, is the probability of a given itemset occurrence (consequent) after the occurrence of another itemset (antecedent), according to formula (2):

$$conf\left(A \rightarrow B\right) = \frac{\sum\left\{t \in D \middle| \left(A \cup B\right) \subseteq t\right\}}{\sum\left\{t \in D \middle| A \subseteq t\right\}} = \frac{\sup\left(A \rightarrow B\right)}{\sup\left(A\right)} \tag{2}$$

III. Algorithm APriori

The use of support for pruning candidate itemsets is guided by the APriori principle: “If an itemset is frequent, then all of its subsets must also be frequent”. This property, also known as anti-monotone property allows the trimming of the exponential search space based on the support measure. Any measure that possesses this property can be incorporated directly into the mining algorithm to effectively prune the exponential search space of candidate itemsets.

The APriori algorithm generates candidate itemsets by

978-1-4244-1821-3/08/$25.00©2008 IEEE

performing the following two operations: a)Candidate generation, which generates new candidate k¬-itemsets based on the frequent (k – 1)-itemsets found in the previous iteration; and b)Candidate pruning, which eliminates some of the candidate k-itemsets using the support-based pruning strategy [8].

The computational complexity of the APriori algorithm can be affected by the following factors: support threshold, number of items (dimensionality), number of transactions, average transaction width, generation of frequent 1-itemsets, candidate generation, and support counting..

IV. Algorithm ZigZag

Some algorithms obtain a reduced result set that can provide useful, although incomplete, information about dataset inferences. In this way, the analysis is reduced by discovering incomplete information about the complete set of valid itemsets. One such approach is by using maximal frequent set algorithms, that identify only those valid itemsets for which no valid supersets exist.

The ZigZag algorithm main idea is to maintain only the maximal frequent itemsets, also known as positive border, to build in an incremental way a frequent itemset grid. It uses the knowledge discovered in prior cycles to reduce the frequent itemset updating cost. The maximal frequent itemsets are updated through a process of backtrack search, previously developed for the non-incremental algorithm called GENMAX [5].

ZigZag is also appropriate to find reliable association rules by means of the stability property. The idea is that if there are few changes in the data related to an itemset, the rules from its subsets will probably stay very similar to the actual ones. In this way, the algorithm becomes more efficient, by not having to fully update some maximal frequent itemsets [6].

V. Database and Hardware

In order to verify whether the rules generated by the incremental algorithm ZigZag have the same quality of the rules generated in the extensive way (by the APriori algorithm), a database was built. Its size should be such that the rule extraction time and the amount of generated rules allowed the evaluation adequately. The data was obtained from a database belonging to a Health Care provider.

The data contained information regarding medical examination supplied by the enterprise, with ten chosen attributes: class of client, situation, specialty code, solicitor code, solicitor specialty code, executor code, service code, user code, user's code, user's age, marital status, and sex.

The data base utilized to the algorithm regard to the same kind of information, however in two distinct periods, the Data Base I is referred to a specific period of time of six months, from January to June, 2005. Its size was 38Mb and it contains 482,286 records. The Data Base II refers to a period of 12 months, from January to December of 2005. With the size of 385Mb, containing 5,075,715 registers.

The hardware used to process the algorithms was an AMD Opteron™ 248 server, bi-processor 2GHz with 4GB RAM, Linux OS – Ubuntu 6.10 – 64bits, Kernel – linux Hercules 2.6.17-11-generic#2SMP, UTC 2007 x86_64 GNU/Linux and MySQL 5.0.

VI. Experimental Setup

APriori algorithm takes the whole data and produces a set of association rules. Due to its nature, the resulting set is complete, in the meaning that all possible association rules are obtained, that comply with the assumptions made, based on the support and confidence levels.

ZigZag, on its own, produces a set of association rules regarding the support and confidence rules, but based on the maximal frequent itemsets. The suggested setup for it involves using half of the dataset to generate the initial set of rules, and then one fifth or one tenth of the remaining records in an incremental fashion. Each time a new set of records is supplied, it outputs a set of association rules built on the basis of the previously generated set.

Then, to compare the quality of the incremental process, both algorithms were subjected to the same inputs (set of records) and the outputs were compared. The main comparison was to compare whether they generated the same set of rules, and, if not, the percentage of agreement. In the latter case, it was of interest also, the position of the similar rules in both sets. In order to determine if there is a trend, various levels of support and confidence measure were used. Table I summarizes their usage.

For the experiments, figure 1 shows the experimental setup. The APriori algorithm used the whole dataset each time it was executed, that means, for each combination of support and confidence factors. For the sake of brevity, the results showed in the next section concerns only the values obtained in the last run, with 100% of the dataset.

The ZigZag algorithm adjusted the initial set of rules with 50% of the total records available (the same as the APriori algorithm) and then was adding and/or deleting some rules each time a new set (with the next 5% of records of the whole dataset) was given as input.

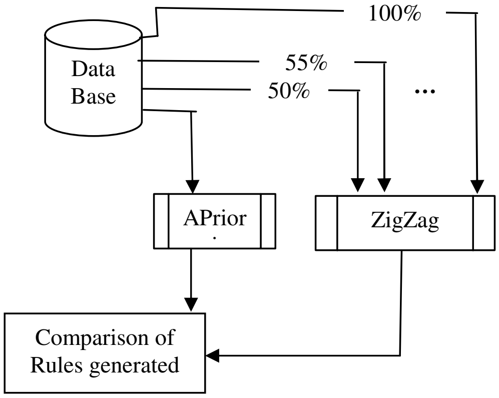

Fig. 1. Experimental Setup.

TABLE I
SUPPORT AND CONFIDENCE THRESHOLDS

Experiment	Support Level (%)	Confidence level (%)
1	10	10
2	10	30

3	10	45
4	10	60
5	30	10
6	30	30
7	30	45
8	30	60
9	45	10
10	45	30
11	45	45
12	45	60
13	60	10
14	60	30
15	60	45
16	60	60

VII. Results and Discussion

In this section some results of the experiments will be presented and discussed. As indicated before, the results presented correspond to measures taken at the end of the whole process. It means, the APriori algorithm has run on the whole dataset, and the ZigZag algorithm has taken the 50% of the initial records, created an initial set of association rules, and then used each next 5% of the data to update the set of rules, building new rules, deleting old rules and/or updating rules already existing.

According to this principle, Table 2 presents the results for each pair (Support threshold, Confidence threshold) the time taken to obtain the resulting set of association rules and the number of association rules generated by each algorithm applied in both data bases. It is important to remember that the time indicated for the APriori algorithm is that spent to find the rules, and was obtained in one cycle. For ZigZag the time indicated corresponds to the sum of all activities in the processing: to obtain the initial set of association rules and to generate the updating with all ten subsequent records (each with 5% of the total).

It's easy to see that for each level of support the time taken is almost independent of the confidence level, irrespective of the algorithm used. It was expected, since most of the time is used generating the candidate itemsets. The great deviation occurs for ZigZag, as it is more effective in the generation and thus the proportion of time in the two phases is more pronounced.

TABLE II
NUMBER OF RULES AND TIME FOR EACH ALGORITHM

Experiment	APriori Algorithm		ZigZag Algorithm	
	Data Base I		Data Base II	
	Rules	Time(s)	Rules	Time(s)
1	2,656	10.40	1,810	120.40
2	1,859	10.38	1,256	120.40
3	1,329	10.38	863	120.39
4	877	10.38	624	120.39
5	396	7.84	174	69.29
6	396	7.84	174	69.29
7	310	7.84	144	69.29
8	219	7.84	107	69.29
9	98	5.89	46	63.35
10	98	5.89	46	63.35
11	98	5.89	46	63.35
12	74	5.89	39	63.35
13	50	5.70	22	51.52
14	50	5.70	22	51.52
15	50	5.70	22	51.52
16	50	5.70	22	51.52

The analysis of the number of rules generated by each algorithm shows that it is equal for the lower level of support (10%) and for the upper levels (45% and 60%). The only differences correspond to extraction of rules with support 30%, this is applied to the first data base called "Data Base I". To the second Data Base, called "Data Base II", there was an analyses incidence to the first base, however in a proportion minor of extracted rules. Besides the quantity of register had grown about 10 times the size of the initial base, in view of the first Base represents the period of 6 months and the second Base represents a period of 12 months. The total of extract rules for all support and trust value generate on total average 48.55% less then the first base.

Whether this trend is particular to the database used or is characteristic of the algorithms remains to be determined with experiments using other datasets.

Despite both algorithms have found the same number of rules for some range of support confidence, it doesn't mean they are the same. Each set of association rules generated by APriori algorithm was compared to the set of association rules generated by ZigZag algorithm, in order to verify their equality. The results of such comparison are presented in Table 3, showing both data bases. The number of common rules is proportional to the number of total rules of association extracted from the database, but in no case it reached 100%. It reaches a maximum in the mid-scale of the support threshold, following almost the same pattern as for the total number of rules. Different from Table 2, the proportion of common rules was the same for the 45% support level.

According to these results, the confidence level mid-scale is where the association rules generated by ZigZag correspond to most of the same rules as the APriori algorithm has found. All this reasoning is based on the idea that APriori, is a non-incremental algorithm and also that its strategy of generation of candidate itemsets is, albeit not exhaustive, complete. That means, all candidate itemsets worth (regarding the support threshold) of testing are generated and used in the second step of the process (association rules building).

TABLE III
PROPORTION OF COMMON RULES

Experiment	Data Base I			Data Base II		
	Common rules	APriori % Common	ZigZag % Common	Common rules	APriori % Common	ZigZag % Common
1	850	32.00	32.00	608	33.59	33.59
2	620	32.72	32.72	434	34.55	34.55
3	485	36.49	36.49	309	35.81	35.81
4	320	36.49	36.49	309	35.81	35.81
5	188	47.47	56.63	110	63.22	63.22
6	188	47.47	56.63	110	63.22	63.22
7	152	49.03	58.46	90	62.50	62.94
8	107	48.86	56.02	68	63.55	63.55
9	74	75.51	75.51	34	73.91	73.91
10	74	75.51	75.51	34	73.91	73.91
11	74	75.51	75.51	34	73.91	73.91
12	50	51.02	51.02	29	74.36	74.36
13	50	52.00	52.00	18	81.81	81.81
14	26	52.00	52.00	18	81.82	81.82
15	26	52.00	52.00	18	81.82	81.82
16	26	52.00	52.00	18	81.82	81.82

Figures 2 and 3 show how is the relationship between the number of rules generated by each algorithm (APriori and ZigZag) according to the support threshold. In both cases the pattern is the same, with little deviations. It induces to the reasoning that, despite they have different mechanisms to find the candidate itemsets and to build the association rules, the ensemble share some characteristics in a global level, ensuing almost identical behavior.

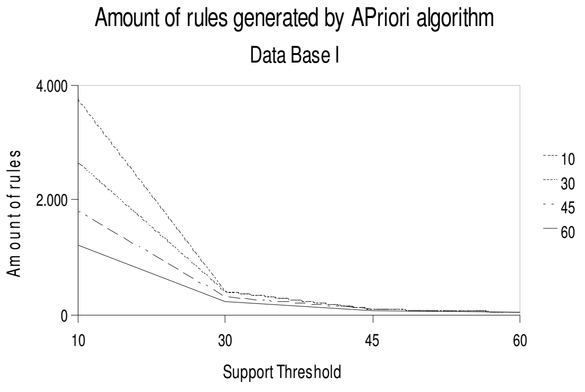

Fig. 2. Amount of rules (APriori algorithm – Data Base I).

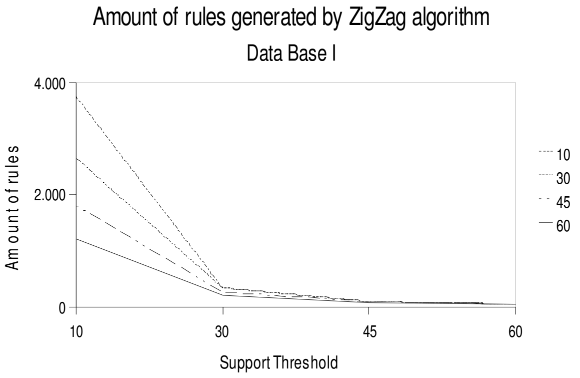

Fig. 3. Amount of rules (ZigZag algorithm – Data Base I).

Figures 4 and 5 show the relationship between the total number of rules (for each algorithm) and the rules that each one shares with the other. The same pattern can be seen again in both cases; one probable cause is a linear growth in common rules as the total amount of rules increase. It should be noted that it was achieved despite the fact that both attained the same amount of rules being generated. Another point that one should be careful about is that even in the case where both algorithms generated the same amount of rules, and within the subset of rules that are equal, not all of them presented the same (real) level of support and/or confidence.

Further study was pursued, regarding the position of the common rules. It was made without taking into consideration the support and confidence levels, only the ranking of the common rules in the set of the rules of both algorithms. In order to test it, the set of rules were divided in ten folders (ranked from 1 to 10), and it was verified whether each common rule was in the same folder in both sets. With this procedure all kinds of behavior were found, from total similarity, when for a specific support and confidence level all common rules were in the same position (ranking) of both algorithms, to some rules in the same position in both algorithms and other don't, to the extreme where the all the common rules were found in different percentiles.

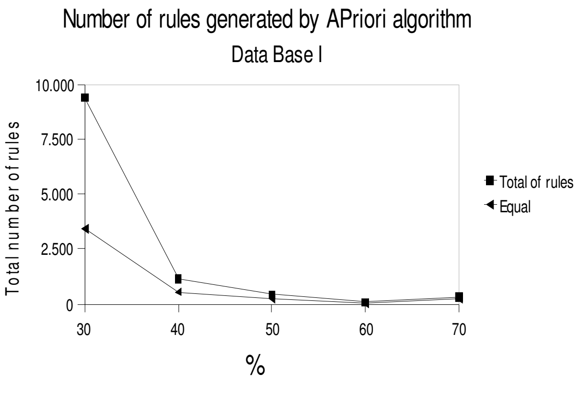

Fig. 4. Number of rules (APriori algorithm).

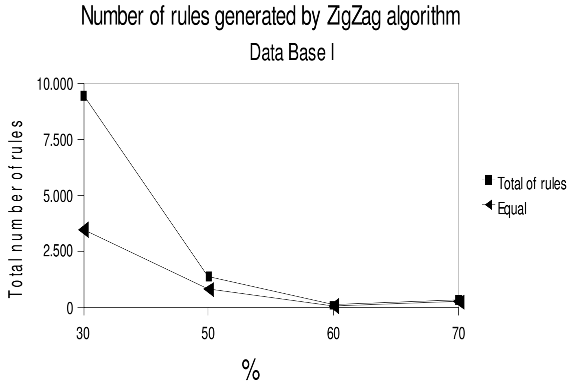

Fig. 5. Number of rules (ZigZag algorithm).

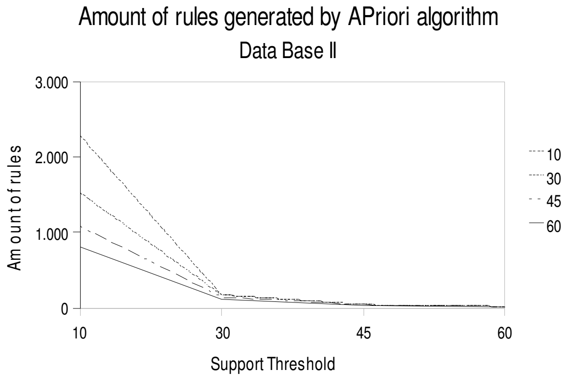

Fig. 6. Amount of rules (APriori algorithm – Data Base II).

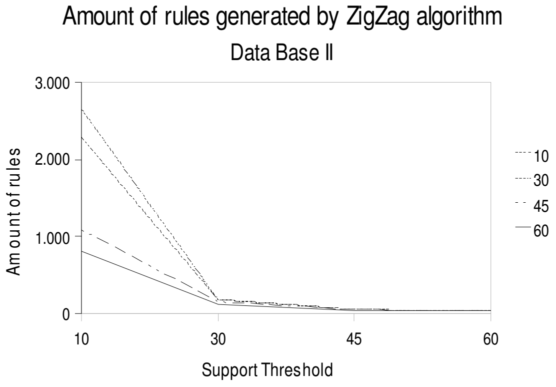

Fig. 7. Amount of rules (ZigZag algorithm – Data Base II).

In analyses way the graphics 2 and 3, the graphics 6 and 7 demonstrate the total number of generate rules trough the APriori and ZigZag algorithms, however applied to the Data Base II.

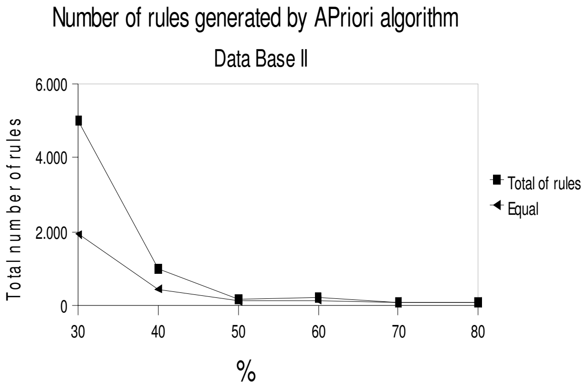

Fig. 8. Number of rules (APriori algorithm – Data Base II).

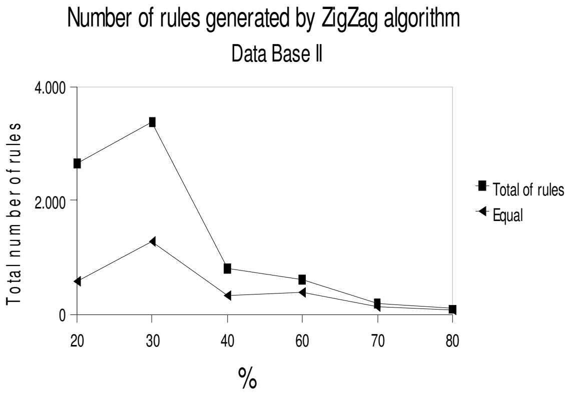

Fig. 9. Number of rules (ZigZag algorithm – Data Base II).

Following the example of pictures 4 and 5, the pictures 8 and 9 show the total quantity of generated rules by the APriori and ZigZag algorithms applied to Data Base II, as well as the quantity of generated rules equals to both algorithms.

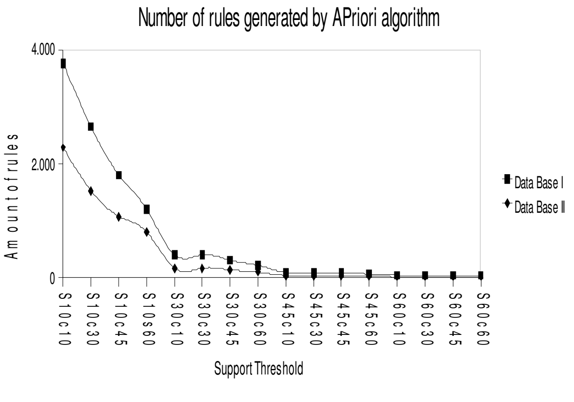

Fig. 10. Number of rules (APriori algorithm) in Data Bases I and II.

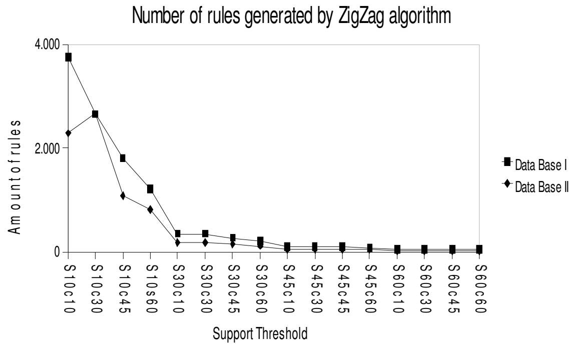

Fig. 11. Amount of rules (ZigZag algorithm) in Data Bases I and II.

Through pictures 10 and 11, we can observe the total number of extracted rules by the algorithms APriori and ZigZag, applied in both data bases (I and II). The objective of theses graphics is to demonstrate the evolution of the algorithms as many is applied an increment to the data base, from the initial base contain 482,286 registers and the final contain 5,075,715 register. However, even if the initial base suffer an increase of the data of about 10 times its initial size, the quantity of extracted rules, has suffered a meaningful 48.55% decrease in extracted rules for both algorithms, with the exception to all support value applied equal to 30, which demonstrate a average difference of 54.20% applied to both bases (I and II) to the APriori algorithm. The ZigZag algorithm applied in both bases (I and II), the difference was 51.50%.

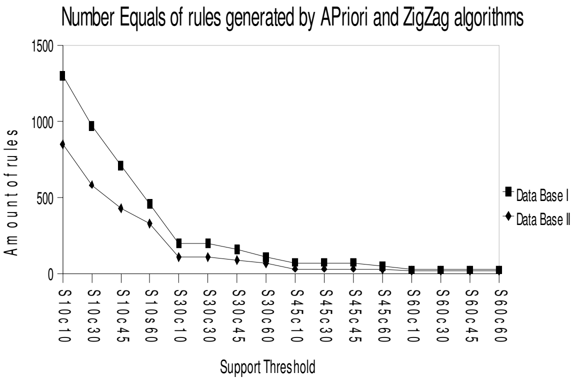

Fig. 12 Number of rules Equals (APriori and ZigZag algorithms) in Data Bases I and II.

Figure 12 presents the total number of rules generated by both algorithms (APriori and ZigZag) that are equal.

One can observe that the quantity of equal rules in both algorithms did not suffer changes within the data proportions, which garantees the quality of the ZigZag algorithm, considering APriori is the reference for Rules Association extraction.

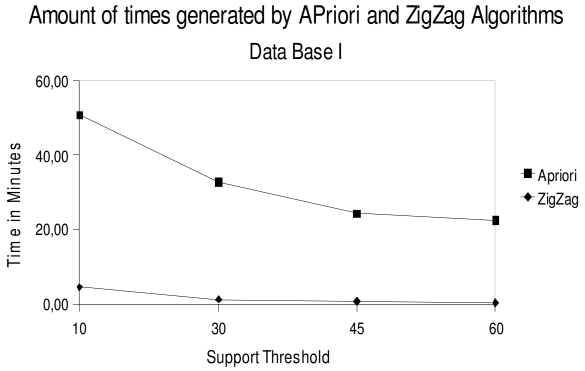

Fig. 13 Amount of Times spent by Apriori and ZigZag Algorithms in Data Base I.

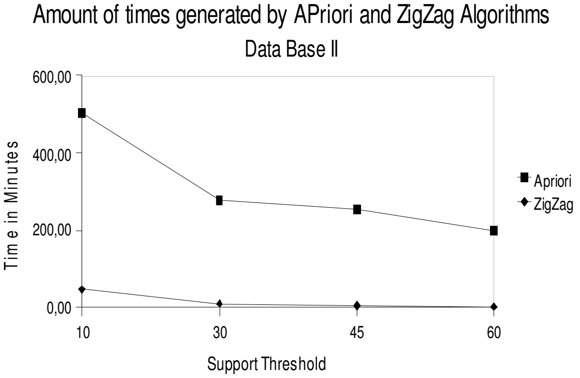

Fig. 14 Amount of Times spent by Apriori and ZigZag Algorithms in Data Base II.

Figures 13 and 14 has show the total time spent by both algorithms (APriori and ZigZag – Data Bases I and II) for the generation of the association rules.

VIII. CONCLUSION

Despite incremental algorithms produce results in times that are a fraction of the batch ones, their strategy uses mechanisms to simplify either the search for candidate itemsets or to generate the association rules. In both cases, it can create results that are not analogous to the complete (though not exhaustive) mechanisms of batch algorithms.

In this paper, a databases of data from a health care provider was used to compare to algorithms using both strategies: APriori and ZigZag. The results showed that the association rule set produced by both can present many differences, particularly in the mid-scale level of support (30-45%), where the rules shared by the algorithms is a fraction (35-70%) of the total amount of rules. Results also shed light that even in the case of common rules, their ranking in the sets generated by both algorithms can be quite different. As a final conclusion, experiments have shown that both algorithms applied on both databases confirm the general reability of ZigZag regarding the APriori reference.

REFERENCES

[1] Hidber, C., Online Association Rule Mining. *Technical Report TR-98-033, University of California at Berkeley*.

[2] Omiecinski, E.R., *Alternative Interest Measures for Mining Associations in Databases.* IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 1, January/February, 2003, pp. 57-69.

[3] Cheung, D.W., Lee, S.D., Kao, B., A General Incremental Technique for Maintaining Discovered Association Rules. Proceedings of the Fifth International Conference on Database Systems for Advanced Applications, 1997, Melbourne, Australia.

[4] Ceglar, A., Roddick, J.F., *Association Mining.* ACM Computing Surveys, Vol. 38, No. 2, Article 5, Publication date: July 2006.

[5] Veloso, A., Meira Jr., W., Carvalho, M., Pôssas, B., Parthasarathy, S., Zaki, M.J., *Mining Frequent Itemsets in Evolving Databases.* Proceedings of the Second SIAM International Conference on Data Mining, SIAM 2002.

[6] Veloso, A., Meira Jr., W., Carvalho, M., Parthasarathy, S., Zaki, M.J., *Parallel, Incremental and Interactive Mining for Frequent Itemsets in Evolving Databases*, Proceedings of the Sixth SIAM Workshop on High Performance Data Mining, 2003.

[7] Zheng, Q., Xu, K., Ma, S., *When to Update Sequential Patterns of Stream Data?*, Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2003.

[8] Tan, P.-N., Steinbach, M., Kumar, V., *Introduction to Data Mining*, Addison-Wesley,2006.