Mining Frequent Patterns in Data Using Apriori and Eclat

A Comparison of the Algorithm Performance and Association Rule Generation

Vlad Robu
NOVA Information Management School
NOVA University of Lisbon
Lisbon, Portugal

Vitor Duarte dos Santos
NOVA Information Management School
NOVA University of Lisbon
Lisbon, Portugal

Abstract—**This paper aims to compare Apriori and Eclat algorithms for association rules mining by applying them on a real-world dataset. In addition to considering performance efficiency of the algorithms, the research takes into consideration the distribution of the support, as well as the number of rules generated by Apriori and Eclat.**

Apriori; ECLAT; Association rules; Algorithm comparison

I. Introduction

In a world where the data is continuously generated by a multitude of sources and in extremely large quantities, it becomes crucial to transform it into valuable and actionable knowledge. The process of transforming raw data into knowledge becomes especially important for large companies that strive to understand as much as possible the behavior of their customers and build a strong competitive advantage by offering a better targeted and a more personalized experience. This is often transformed into a problem of finding the most frequent patterns in large datasets and create a generalizable and interpretable picture of reality. A branch of this problem area is represented by those problems that imply transactional datasets and require the identification of the most frequent item-sets, the so-called Market Basket Analysis type problems or in other words, Association Rule Learning problems. The latter type of problems aims to identify which items tend to occur together most frequently in the transactions of a dataset. It strives to find the most frequent relationship between the respective items. There are several algorithms designed to serve the purpose of the pattern mining nevertheless, this paper will focus only on two most known and used in the industry algorithms: Apriori and Elcat. In order to understand which one is better to be applied in a certain situation, many past researches focused on comparing Apriori and Eclat algorithm efficiency, interpreted as the time needed to mine the rules. The evaluation being mostly done by making the algorithms perform under predefined controlled circumstances and often applying them on simulated data [1][2][3][4]. Inspired by past researches, this paper aims to add new dimensions to the comparison by considering additional elements related to the association rule generation process. For the purposes of this research, Apriori and Eclat were applied on a real-world dataset, trying to simulate a real-world scenario and aiming to generate a more business driven comparison of results. The comparison therefore considers additional factors that might result important in a business environment in addition to the ones considered for academic research. Consequently, for the evaluation are considered elements such as the quantity of generated rules, interestingness of the rules, drawbacks of the algorithms, and finally, the time efficiency needed to generate the rules.

II. Frequent pattern mining algorithms

Association rules mining is commonly used for market basket analysis problems with the goal to identify combination of items more likely to appear together in the transaction of a dataset. It helps to identify interesting rules, frequent patterns, correlations or just casual data structures in transactional datasets [4]. An association rule can be seen as an expression of type X ⇒Y, where X and Y are sets of items. In a real-world scenario such a rule can be that, for example, customers that have purchased X also purchased Y. The interestingness of such a rule is represented by its *support* and *confidence.* Where *support* of a rule X ⇒ Y is the percentage of transaction that contain both X and Y. While the *confidence* of the rule represents the percentage of transactions of X that also contain Y [5].

A. *Apriori algorithm*

Apriori is an association rule mining algorithm proposed in 1994 by Srikant and Agrawal for Boolean association rules [6]. The Apriori algorithm employs level-wise search for frequent item-sets, using bottom-up research and moving upward level-wise in the lattice [7]. In order to identify the association rules, the algorithm runs through two phases: *Candidate Generation* and *Pruning*. In the *Candidate Generation* phase, the Apriori algorithm uses the prior knowledge of frequent itemset properties and k item-sets are used to explore k+1 item-sets. Intuitively, if an item-set X has minimum support, so do all subsets of X. After generating all the k+1 candidates, a new scan of the transaction is done and the support of these new candidates is determined. Afterwards, during the *pruning* step, the k-1 which are found not reaching the minimum support threshold are eliminated [8][9][10].

978-1-7281-5255-1 ©2019 IEEE

When applying the Apriori algorithm on data using the R opensource software one must be aware that it come with improvements such as a prefix tree and item sorting. The Apriori algorithm in R is part of the *arules* package [11] and requires the following code (1) and arguments to be applied (2)(3)(4)(5):

```
apriori (                  (1)
data,                      (2)
parameter = NULL,          (3)
appearance = NULL,         (4)
control = NULL             (5)
)
```

where *apriori* (1) recalls the algorithm to be applied on the data based on the defined arguments; *data* (2) is an object of class transactions or any structure which can be coerced into transactions (e.g. binary matrix or data frame); *parameter* (3) is an object of class *list* and defines the behaviour of the rules to be mined by allowing to set the minimum thresholds for support as well as the maximum number of items present in a rule and the maximal time for subset checking; *appearance* (4) is an object of class *list* and allows to restrict the item appearance; *control* (5) is an object of class *list* and controls the algorithmic performance of the mining algorithm (e.g. item sorting, report progress, etc.) [12].

B. ECLAT algorithm

ECLAT or Equivalence Class Transformation Algorithm, is a frequent pattern mining algorithm that mines efficiently frequent patterns by performing a bottom like depth first search or in other words, a bottom up Lattice traversal. In order to mine frequent patterns, it requires to be applied on a vertical database [13]. For this methodology, all the transactions that contain a certain itemset are grouped into the same record. After intersecting the frequent k-item sets, the frequent k+1 item sets are generated. This process occurs until no more frequent item sets can be found. What makes Eclat advantageous is that it does not need to scan the database multiple times in order to identify the k+1 item sets. Indeed, after scanning the database once, the k+1 item sets are discovered by intersecting the k-item sets with one another [14]. The support for each transaction is calculated and if it is equal or greater than the minimum support threshold set by the researcher, then it is considered for the analysis otherwise it gets discarded. One of the main advantages of Eclat is that it reduces the access time, while a disadvantage is represented by the fact that it does not consider rule *confidence* as an interestingness measure but considers only the *support* of the rules [15].

Eclat is one of the many frequent pattern algorithms that have been considered and implemented into R opensource software. Like Apriori, it is part of the *arules* package [11] and is defined by the following code line with the respective arguments:

```
eclat (                    (6)
data,                      (7)
parameter = NULL,          (8)
control = NULL             (9)
)
```

In the above lines of code *eclat* (6) [16][17] recalls the algorithm to be applied to the transactional dataset; *data* (7), as for *apriori* (1), is an object of class transactions or any structure which can be coerced into transactions; *parameter* (8) allows to define the minimum support threshold and the maximum length of the rules to be generated; *control* (9) being an object of class list allowing for the algorithmic controls. In comparison to *apriori,* because of the way it is structured, *eclat* does not consider confidence as a parameter [12].

III. METHODOLOGY

This research emerged from a real-world business scenario, where the researchers had to identify the most frequent patterns in a transactional dataset generated by a software logging the choices of the users, i.e. mining the most frequent patterns present in the users' choices. After cleansing the data, the transactional dataset was transformed into a binary transactional dataset, where the presence of a certain item was represented by a binary variable. Following, a dataset of 52.938 transactions and 26 binary variables was generated.

Apriori and Eclat were applied using the R opensource software. For this purpose, the "arules" and "arulesViz" software packages were used. After declaring the variables as *factors* (10) and the dataset type as transactional (11), Apriori and Eclat were applied by using the formulas (1) and (6), as showed in the example.

```
factor( )                  (10)
as(x, transactions)        (11)
apriori( )                 (1)
eclat( )                   (6)
```

In order to check the performance of the two algorithms and allow the identification of frequent patterns, the two algorithms were iterated with different minimum support thresholds. Therefore, the minimum support threshold used for this research was set at the following levels: 0.01%, 0.1%, 0.5%, 1%, 5%, 10%, 20%, 30%, 50%, defining in this way the minimum frequency of appearance of a item set in the dataset. Setting very low minimum thresholds, allowed both to stress the algorithms to mine frequent patterns in a larger universe of rules, as well as to find both frequent rules and rare rules that might be of interest for the research. The minimum confidence threshold for the Apriori algorithm was set at the software default level of 80%. On the other hand, confidence measure is not considered for Eclat.

In conclusion, the performance of the algorithms was tracked and compared considering the time needed to generate the rules, the number of generated rules at each iteration and the distribution of the interestingness of the rules based on the support.

IV. Comparative analysis

The following Fig. 1, Fig. 2, Fig. 3 and table 1, compare the performance of Apriori and Eclat under three aspects: time in seconds needed for the association rule generation, distribution of the support of the rules and the number of rules generated at different support thresholds.

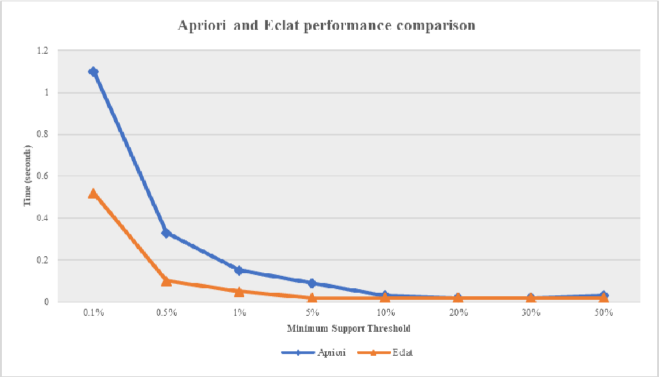

Figure 1. Apriori and Eclat time performance comparison represented in seconds

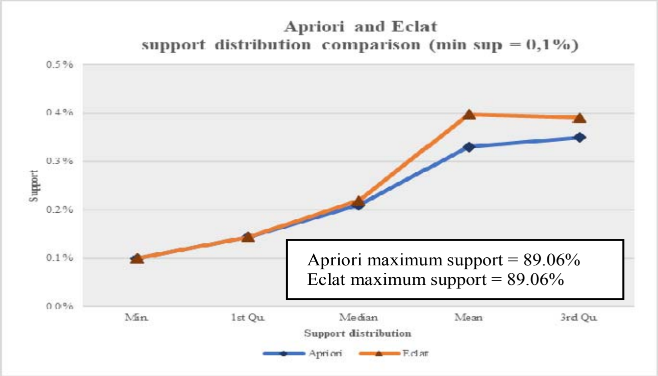

Figure 2. Apriori and Eclat rules' support distribution comparison

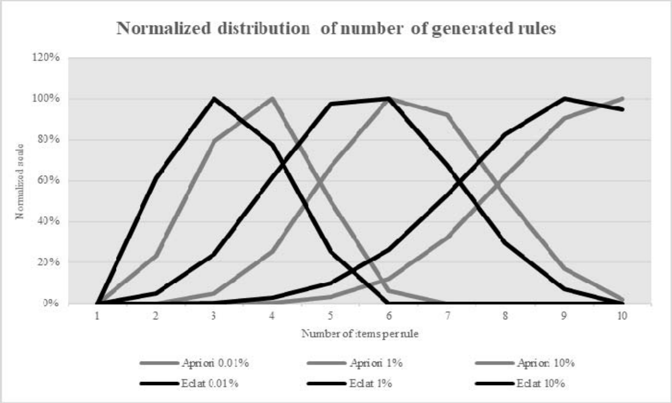

Figure 3. A comparison between the normalized distribution of the number of generated rules by Eclat and Apriori, per number of items in a rule.

TABLE I: Apriori and Eclat comparison on the number of generated rules

Comparison of number of rules generated by Apriori and Eclat at different support thresholds		
Min support threshold	**Apriori**	**Eclat**
0.01%	7,465,565	1,443,983
0.1%	686,979	154,242
0.5%	90,278	26,735
1%	26,345	9,341
5%	1,211	633
10%	312	187
20%	62	40
30%	34	21
50%	6	4

For the sake of visualization, in Fig.2, the maximum support reached by the rules generated by the two algorithms, is written on the graph instead of being visualized.

In line with previous researches [1][2], Apriori shows to perform slower than Eclat, resulting in more time required to generate the association rules. The performance of Apriori is slower and requires more computational power compared to the performance of Eclat especially when lowering the minimum support threshold at which the algorithms are applied. Considering that in a real-world scenario the researcher aims to find not only the most frequent but as well rare and interesting patterns, it might result crucial to operate with low support (and confidence) thresholds in addition to using larger datasets. In the latter situation, one may opt to avoid the use of Apriori in favor of faster association rule mining algorithms. A lower time required by Eclat can be explained partially by the fact that it tends to generate less rules than Apriori. For lower support thresholds, the gap between the output in terms of number of rules of the two algorithms under comparison increases significantly. Additionally, after comparing the distribution of the number of rules generated by the two algorithms across the number of items per association rule, it was possible to note how Eclat tends to generate association rules with less items compared to Apriori (Fig.3). For the sake of visualization only the distributions of the rules generated at a minimum support threshold of 0.01%, 1% and 10% were considered for comparison. At different minimum support thresholds, the distributions follow the same pattern. To make the data comparable, it was normalized through min-max feature scaling normalization (12)[18], where x_{min} is a lower data bound, x_{max} is an upper data bound, x is a data component and $\bar{x}$ is a data point in the range [0, 1].

$$\bar{x} = (x - x_{min}) / (x_{maz} - x_{min}) \qquad (12)$$

On the other hand, when comparing the interestingness of the generated association rules considering support as a measure, the algorithms reach the same maximum support value for the generated association rules and the rules with the highest support tend to be in the 4rh quartile of data for both Apriori and Eclat. Furthermore, Eclat tends to generate less rules than Apriori, which tend to have a slightly higher support measure (Fig. 2).

V. Conclusions

In order to mine the most frequent and potentially interesting patterns, association rule mining algorithms come to be perfectly suitable for the task. By comparing Apriori and Eclat in this paper, it was possible to highlight how it becomes more computationally expensive for Apriori compared to Eclat to mine frequent patterns at very low support threshold levels. It might result to be a critical difference and factor of choice when the research or the business goal is focused on identifying very rare and interesting patterns. Furthermore, it was noticed that at very low thresholds Apriori tends generate significantly more rules than Eclat, which in a real-world scenario can increase the complexity of identifying interesting and meaningful rules. Despite these elements might lead one to choose Eclat over Apriori, when using R opensource software Apriori shows the great advantage of allowing for calculation of additional interestingness measures [19] which might help to explore and objectively choose and navigate through the vast universe of generated association rules. Considering the limitations of the research described in this paper, future researches can have a more in-depth focus on the differences between the two algorithms based on the quality of the generated rules. Being aware of the tendency of Apriori to generate redundant rules [20], a new dimension of analysis can be added to the comparison by eliminating the redundancy within the rules and observe the impact on the output and algorithm performance. In conclusion, given the framework and used tools (R opensource software), there is no definitive verdict over which algorithm results to be better than the other. Given a specific real-world scenario and goal to be achieved, one will have to carefully evaluate the dimensions that differentiate Apriori from Eclat, and choose appropriately based on the advantages and disadvantages each method is characterized by.

References

[1] Chee, CH., Jaafar, J., Aziz, I.A. et al. Artif Intell Rev (2018). https://doi.org/10.1007/s10462-018-9629-z

[2] Gayathri, G. (2017). Performance comparison of Apriori, Eclat and FP-Growth algorithm for association rules learning. *International Journal of Computer Science and Mobile Computing*, 81-89.

[3] Vani, K. (2015). Comparative Analysis of Association Rule Mining Algorithms Based on Performance Survey. *International Journal of Computer Science and Information Technologies*, 3980-3985.

[4] Garg, K., & Kumar, D. (2013). Comparing the Performance of Frequent Pattern Mining Algorithms. *International Journal of Comoputer Applications*, 29-32.

[5] Sethi, A., & Mahajan, P. (2012). Association Rule Mining: A review. The International Journal Of Computer Science And Applications.

[6] R.Agrawal and R.Srikant, "*Fast Algorithms for Mining Association Rules*," In Proc. of VLDB '94, pp. 487-499, Santiago, Chile, Sept. 1994.

[7] Ghosh, S., Biswas, S., Sarkar, D., & Sarkar, P. (2012). 2012 Third International Conference on Emerging Appplications of Information Technology. International Conference on Emerging Applications of Information Technology, (pp. 202-205). Kolkata.

[8] Arora, J., Bhalla, N., & Rao, S. (2013). A review on association rule mining algorithms. *International Journal of Innovative Researc in Computer and Communication Engineering*, 1246-1251.

[9] Han, J., Kamber, M., & Pei, J. (2012). Apriori Algorithm: Finding Frequent Itemsets by Confined Candidate Generation. In J. Han, M. Kamber, & J. Pei, *Data Mining Concepts and Techniques - Third Edition* (pp. 243-246). Amsterdam: Elsevier.

[10] Christian Borgelt (2003) Efficient Implementations of Apriori and Eclat. Workshop of Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL, USA).

[11] Michael Hahsler, Christian Buchta, Bettina Gruen and Kurt Hornik (2019). arules: Mining Association Rules and Frequent Itemsets. R package version 1.6-3. https://CRAN.R-project.org/package=arules.

[12] R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[13] Kaur, M., Garg, U., & Kaur, S. (2015). Advanced Eclat Algorithm for Frequent Itemsets Generation. *International Journal of Applied Engineering Research*, 23263-23279.

[14] Chee, CH., Jaafar, J., Aziz, I.A. et al. Artif Intell Rev (2018). https://doi.org/10.1007/s10462-018-9629-z.

[15] Gayathri, G. (2017). Performance comparison of Apriori, Eclat and FP-Growth algorithm for association rules learning. *International Journal of Computer Science and Mobile Computing*, 81-89.

[16] M.J. Zaki and K. Gouda Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2003, Washington, DC), 326-335 ACM Press, New York, NY, USA 2003.

[17] Christian Borgelt Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2(6):437-456. J. Wiley & Sons, Chichester, United Kingdom 2012 doi:10.1002/widm.1074 wiley.com.

[18] S. Aksoy and R. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," Pattern Recognit. Lett., Special Issue on Image and Video Retrieval, 2000.

[19] Hahsler, Michael (2015). A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules, 2015, URL: http://michael.hahsler.net/research/association_rules/measures.html.

[20] Ashrafi M.Z., Taniar D., Smith K. (2004) A New Approach of Eliminating Redundant Association Rules. In: Galindo F., Takizawa M., Traunmüller R. (eds) Database and Expert Systems Applications. DEXA 2004. Lecture Notes in Computer Science, vol 3180. Springer, Berlin, Heidelberg.