Transactions on Machine
Learning and Data Mining
Vol. 13, No. 2 (2020) 83 - 99

P-ISSN: 1865-6781, E-ISSN 2509-9337
ISBN 978-3-942952-78-1

A Visualization Framework for Post-Processing of Association Rule Mining

Manal A. Alyobi and Arwa A. Jamjoom

Faculty of Computing and Information Technology, King Abdulaziz University
Jeddah, Saudi Arabia
malyoubi0090@stu.kau.edu.sa
ajamjoom@kau.edu.sa

Abstract. Association rule mining is one of the most used techniques in data mining. It can be applied in different fields to uncover the hidden relationships in large data sets. Despite that fact, association rule mining algorithms often leave the analyst with the task of analyzing and understanding thousands of rules. Also, this large number of rules cannot be represented effectively in classical visualizations. Thus, the current interest in visualizing data mining is toward applying visual data exploration techniques. These techniques enable the user to interact with graphs. In this paper we introduce a new visualization technique for association mining results based on tree methods, specifically the collapsible tree. To do that, we converted a grouped matrix into a collapsible tree by utilizing R packages. We demonstrate our technique through a market basket analysis case study.

Keywords: Visual Data Mining, Association Rule Mining, Collapsible Tree, Market Basket Analysis.

1 Introduction

Data mining is the process of finding useful information from a large database. Data mining applies different analytical models (descriptive, predictive) to extract interesting patterns from a collection of data. These interesting patterns are non-trivial, unknown in advance, and considered implicit and actionable knowledge [1]. Additionally, data mining is considered the main step in knowledge discovery in the database [2].

Association rule mining (ARM) is a well-known data mining method. It is used to discover interesting rules from a large transaction database or data repositories [3]. It often generates a large number of rules, which have to be appropriately presented for easy understanding [4]. Then, the analyst must sift through these rules to look for non-trivial patterns. However, executing this task manually is strenuous, time-consuming, and sometimes does not succeed. Therefore, the relationship between data mining and visualization has a long history [5, 6]. The visualization supports the analytics phase in data mining through a representation of the rules in graphs and by applying different interactive techniques, like zooming and selecting [4].

Ronald et al [7] defined visualization as "The transformation of a problem into graphical form, to engage the visual intelligence of the human viewer." Many studies on human perception and intuition have found that graphical representation is a better illustration compared to presenting the same information in text or tables [8]. Also, these studies have confirmed that "Visual representation and interaction techniques take advantage of the human eye's broad bandwidth pathway into the mind to allow a user to see, explore and understand a large amount of information at once " [9].

Applying visualization techniques to represent Association rules (ARs) in graphs is a difficult task because ARs have various relational natures or many interrelated items, especially with a large number of rules. Even so, many classical visualization methods have been used to represent AR mining results, for instance, scatter plot, parallel coordinate, matrix, and graph-based visualization. Still, these methods cause screen clutter and are not suitable for displaying large numbers of rules [4].

Using large terms in the context of ARM does not mean big data. It is a term that researchers recently began to use to express the increase in the number of rules because of the increase in data transactions. The number of rules that was displayed by old visualization methods was in the range of tens, but now it may reach hundreds or thousands. However, it is possible to reduce the number of rules that are generated from ARM logarithms by setting constraints with high values (high support). However, this may miss opportunities to uncover important rules.

As result of that, in this paper we will tackle the issue of visualizing large numbers of rules by using a tree-based representation - a collapsible tree. Tree-based representation has proven an effective technique for hierarchical data structure [10]. Specifically, we aim to convert a grouped matrix [4], which organized rules hierarchically, after applying k-means clustering into a collapsible tree.

The study was conducted in one of the most recognized ARM applications—market basket analysis. Section 2 highlights related work. In section 3, the proposed framework is presented. Section 4 demonstrates the framework in a market basket analysis case study. We continue in section 5 with evaluation and present the result and analysis in section 6. We conclude the paper in section 7.

2 Related Work

Using visualization techniques to interpret ARs is not a self-evident or easy mission. This is due to ARs' multiple relational natures, especially if there is large number of rules [11]. Another factor that complicates the representation of ARs is the need to add interesting measures (support, confidence, and lift) in the visualization [11]. These measures can be titled as metadata.

Therefore, the basic idea in visualizing ARs is mapping the items and metadata. In mapping, the analyst can use one of three types: one-to-one, one-to-many/ many-to-one, and many-to-many. One-to-one means visualizing one item in both the antecedent and consequent side. One-to-many/many-to-one visualizes one item on one side and many items on the other side; it can be described as rule-to-item. Many-to-many mean showing many items on both sides [12]. To show many items on any side, clustering techniques are mostly used [4].

Visualization techniques can be classified into two categories, according to which direction it belongs. The first direction is called model visualization and the second direction is called visual data exploration [13]. More information about them is in the next two sub-sections.

2.1 Model Visualization Approach

In this approach, visualization techniques are used to transform the discovered knowledge into a graphical form [13]. The following are examples of these techniques:

Table- based view. Beginning in the 1970s, tables were used for data storing [14] and later as a way to represent data such as ARs [15]. A textual representation using the table-based view enables users to understand and analyze data more easily due to its simplicity. However, this will not be the case when there are more rows of ARs, which makes finding relations and interesting rules a difficult task [11].

Scatter plot. Scatter plot is an old visualization technique that has two dimensions to represent relationships [16]. In [17], the authors used a scatter plot to visualize 324 rules. The dots in a scatter plot represent the rules, with the support measure displayed in the horizontal axis and the lift measure displayed in the vertical axis. The confidence level is shown through the color shading of the dots. Scatter plot visualization is one of the few traditional visualization techniques that are capable of representing large numbers of rules. However, labeling is a significant limitation when using a scatter plot on a wide scale. The rule labels in a scatter plot are unclear as there is not enough space, so the user takes a long time to discover and understand relationships [3]. Even when there are rules in sparse areas in a scatter plot, they are mostly ignored and treated as noise data, which may affect the discovery of interesting relationships [16].

Parallel coordinates. Parallel coordinates are a visualization technique used for high dimensional data presented in a two-dimension space. Early visualization applications used parallel coordinates to visualize classification rules and ARs, such as [18, 19]. However, in recent ARM visualization works, there is no usage of this technique because more rules mean more crossovers between lines, which make the process of finding interesting rules very difficult [20].

Graph-based view. The graph-based view is a visualization technique that contains nodes and edges. Within the graph, the items are displayed by nodes, and edges display the relations between items. Arrows with different colors and widths are used to show metadata on the graph. In [21], the authors built a graph-based framework to visualize ARM results. The graph-based view is a very appropriate choice when the number of rules is small. However, when the number of rules increases the graph becomes more cluttered and more challenging to identify the name of items and finding interesting rules in.

Matrix-based view. The matrix-based view is a high-dimensional visualization technique. It is one of the most suitable techniques for ARs. Therefore, we find that most new works in visualizing ARs try to enhance the matrix-based view in many different ways to solve the issue of visualizing many rules. The authors in [22] introduced a novel technique to visualize ARs with multiple antecedents in the text mining domain (many-to-one). The main goal was to visualize a large number of ARs and metadata. It utilized a 3-D matrix-based visualization after grouping antecedents. According to [22], 2D matrix is effective with one-to-one relationship and the strength of this graph fall down, when visualizing many items in any sides. This is due to the inability to determine the items involved in the relationship.

2.2 Visual Data Exploration

In this approach, the visualization techniques enable the user to interact with the rules through interactive methods. Most techniques in this approach follow Shneiderman's visual information seeking mantra: overview first, zoom and filter, then details-on-demand [13]. Some examples of these techniques are mentioned below:

Visualization ARs using three graphs. The authors in [11] built a system that gives three different views of AR. The three views are the matrix view, graph view, and detail view. The three views connect via a filtering mechanism. The main reason behind having three views is to build an interactive system that enables the user to understand and interpret a large number of ARs. The proposed system is called scalable AR visualization (SARV). It applies highlighting and focusing techniques instead of using clustering techniques to represent the rules. Also, it applies one-to-one mapping (see Fig. 1).

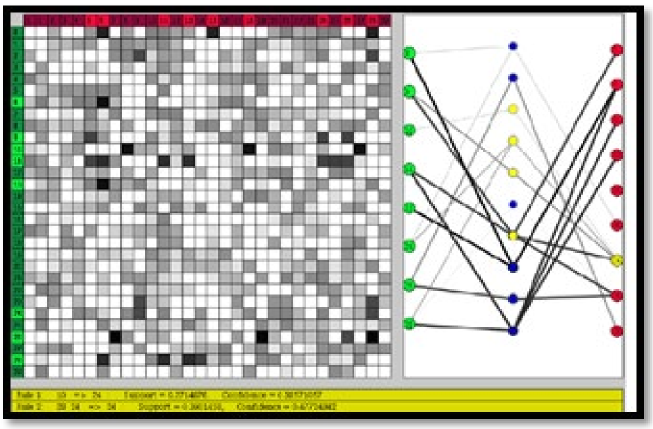

Fig. 1. SARV system

Visualization ARs in hierarchical groups. In [4], the authors proposed a new interactive visualization technique called the grouped-matrix representation. This new method applies rule-to-item mapping. It is an enhanced version of the traditional matrix-based visualization. It uses k-means clustering based on the interest measure lift. It enables users to interact with the graph through inspect-zoom in/out (see Fig. 2).

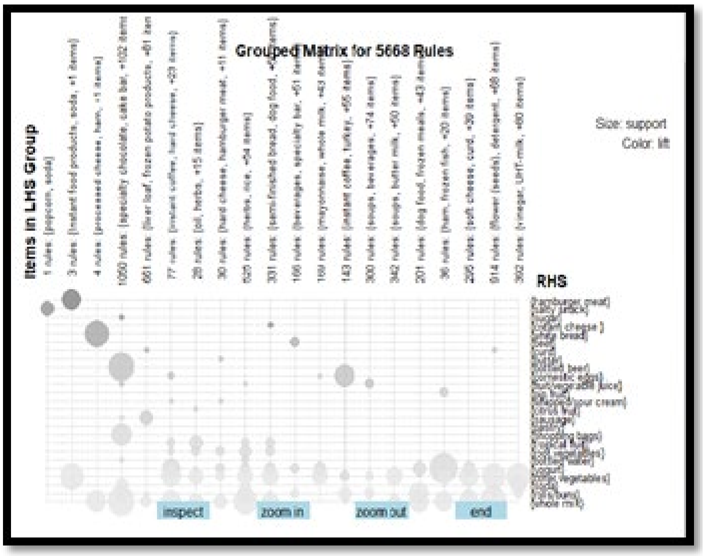

Fig. 2. Grouped matrix (balloon plot)

3-D matrix-based visualization system of ARs. The authors in [8] built a system called 3-D matrix-based visualization of ARs (3DMVS). It is one of the few techniques that used three dimensions. The main reason behind using 3D is to create a vivid and clear graph to investigate. It applies rule-to-item mapping. In addition to support and confidence measures, the system uses a new measurement called weigh and applied as a new merging method to decrease the number of rules. The interactive features in 3DMVS are sorting, filtering, and showing the details.

3 Proposed Framework

We propose a collapsible tree visualization framework to visualize ARs. The main idea in this framework is to exploit the clustering approach and the grouped matrix visualization [4] to construct the collapsible tree. The main steps of the proposed framework are shown in (Fig. 3).

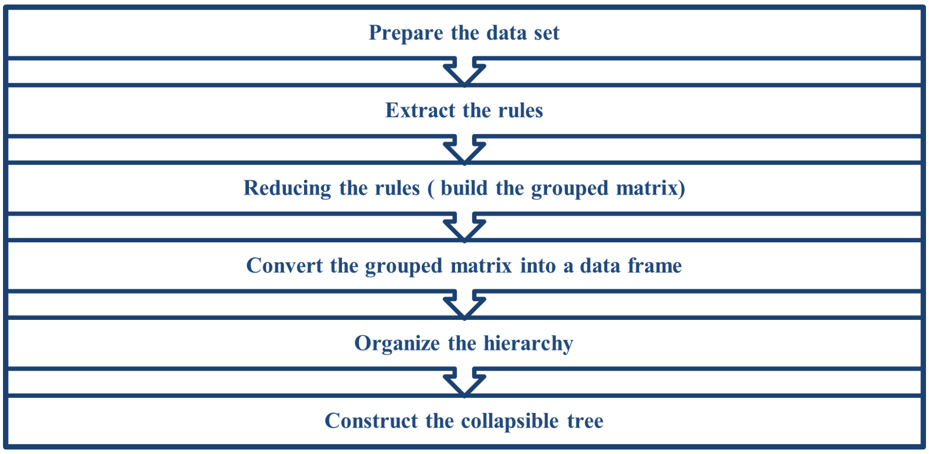

Fig. 3. The steps of the proposed framework

Prepare the data set. Data preparation is considered an essential step in data mining. Data preparation consists of organized steps aimed to ensure data quality. These steps are data collecting, data integration, data transformation, data cleaning, and data reduction. In this study, we used an available data set in one of the packages in R. As most of the transaction data, the data sets are sparse.

Extract The rules. Different algorithms can be used for ARM. However, we adopted the Apriori algorithm. The Apriori algorithm was first proposed by Agrawal and Srikant [3]. It is specifically designed for a transactional database. The basic theory of the Apriori algorithm is "if an itemset is frequent, then all of its subsets must also be frequent, and vice versa." By using the Apriori algorithm we can generate both frequent itemsets and ARs:

1. Generate frequent itemsets: The algorithm scans the database to count each item. If the item support value is greater than the predetermined minimum support, it is considered a frequent item. As a result of the first scan, a set of frequent 1-itemsets is identified. Then, the algorithm applies another scan to the database to determine the frequent 2-itemsets. This scanning process is repeated until the itemsets become null.
2. Generate rules: After identifying a list of the frequent itemsets, the algorithm identifies the rules in the form of antecedents => consequents. If the confidence value of frequent itemsets is greater than the predetermined minimum confidence, it is considered a candidate rule.

Reduce the number of rules (Build the Grouped Matrix). The number of rules generated from the Apriori algorithm is large. Therefore, different clustering approaches have been used to reduce the number of rules. This step is considered very essential when visualizing AR, to avoid a cluttering issue. In this paper, we adopt the clustering approach in [4]. Although it was about representing the rules on matrix-based visualization, we have benefited from their new way of clustering. This clustering is based on creating nested groups of antecedents via k-mean clustering and organizing the results hierarchically. In this clustering approach, the grouping is created based on the lift value. Simply put, rules group together if their antecedents have similar lift values with the same consequent. The strength of this method lies in its ability to group antecedents and their substitutes, as these rules have a strong association with the same consequent. We used the arulesViz package to get the grouped matrix. The columns in the grouped matrix represented the antecedent clusters while the rows represented the consequents. The lift value aggregated based on the median function and is represented by the size of the balloon.

Convert the Grouped matrix into a data frame. This is a general and major step and includes several actions to achieve.

Organize the hierarchy. After importing the clustering rules into a single data frame, are arranged in a hierarchical format. To do that, we used the data tree package. As data tree objects can easily be manipulated into collapsible tree objects. The hierarchy is organized downwards such that each cluster of antecedents is put under the consequent it corresponds to. Figure 4 shows the adopted format.

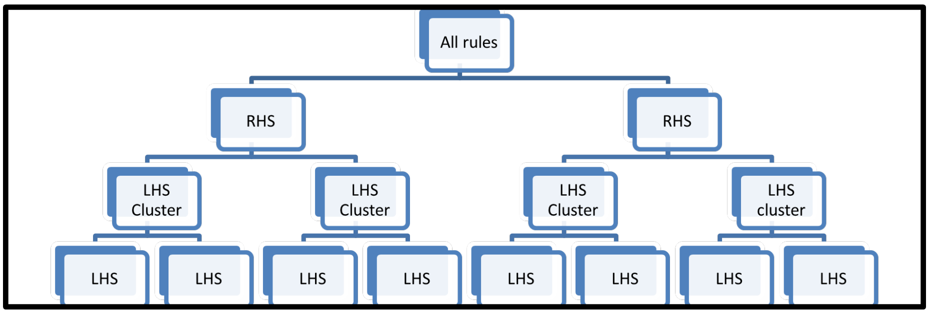

Fig. 4. Hierarchical format of ARs

Construct the Collapsible tree. The collapsible tree is constructing by using the framework implemented in [23]. The tree format is the Reingold-Tilford tree. The engine used is a java d3.js engine.

4 Case Study: Market Basket Analysis

To expound the proposed visualization framework, we will show the framework in the market basket analysis case study. We used the data set called "Groceries." It was provided by Michael Hahsler and his team for the arules package in R. It contains 9835 transactions and 169 different items [24].

The next step was to run the Apriori algorithm; we set a support threshold of 0.001 and a confidence threshold of 0.5. The number of rules generated from the algorithm was 5668. Within these rules, the number of unique antecedents was 4097 and the number of unique consequents was 25. Then we clustered the ARs by using the k-means algorithm, as discussed in the previous section. We then utilized the arulesViz package to generate the grouped matrix that was pictured in the balloon plot. Next, we converted the grouped matrix into a data frame. The basic steps of converting were as follows:

1. Create a data frame from the grouped-matrix: A data frame in R is a table or a 2-D array-like structure. In a data frame each column represents one variable, and each row represents the set of values of each instance. In this case study, we created the data frame from the grouped matrix by utilizing the function as.data.frame().
2. Re-shaped the data frame: Each cluster was represented by a unique column. To gather all the cluster columns and collapse them into a single column called "antecedents", we used the gather() function in the tidyr package, As a result, the new data frame had three columns: consequents, antecedents, and lifts.
3. Remove the missing values: We deleted the rows that had a lift value of NA because these rules did not pass minimum support or minimum confidence thresholds during extraction.
4. Create another data frame from the "rules": To get the values of support and confidence to be represented in the collapsible tree.
5. Merge the two data frames: We used a lift join to merge the two data frames.

Finally, we converted the data frame into a collapsible tree. Via the function htmlwidgets::saveWidget, the collapsible tree was saved as a stand-alone html file. Figure 5 depicts the collapsible tree, which has four layers of nodes. The first layer is the root node. Expanding/collapsing the root will make the second layer appear/hide. To expand the sub-trees, the user has to click on the collapsed node. The second layer presents the consequents. The number of nodes is based on the number of consequents. In this case, we have 25 consequents. The third layer shows the antecedents' clusters associated with each consequent. The label style of each node is as follows: the first three items are separated from each other with a comma and the number of items if there were more items in that cluster. As shown in figure 5 there are nine clusters of antecedents, associated with the consequent rolls\buns. The label of the node reveals the number of other items in antecedent. For example, cluster 5 has three other items besides frankfurter, beef, and yogurt. The fourth layer presents the antecedents that are in the antecedents' clusters (in the third layer) separately, where each antecedent is represented in its own node. All the nodes are the same size because all the nodes have similar support values. The color of each node indicates the aggregated lift value. We used a color-grading technique to differentiate between the nodes—the larger the lift the denser the color and vice versa.

Putting the cursor over any nodes will show detailed on-demand information. We used tooltip to represent the metadata, which are lift, support, confidence, number of rules, and numbers of items (see Fig. 6).

Fig. 5. The Collapsible tree

The final result takes the benefits of grouped matrix visualization and adds more benefits; among the benefits preserved is the clustering approach. According to [4], most of the clustering approaches were ineffective because high dimensionality in transaction datasets leads to highly dimensional rules and in turn distance metrics are rendered ineffective.

The ability to zoom in and inspect inner rules in the grouped matrix was replaced with the collapsibility of the tree. Opening the node lands the user into the next-level nodes. The user can get an overview of the most interesting groups (in terms of RHS) from the color of nodes which represent lift by default. A light blue color indicates that the node is among nodes with minimum lifts, a more dark color would indicate high values of lift.

In addition to color and collapsibility there is the tooltip which guides the user on the values of interesting ness at each level this is additional benefit. The user is able to browse several nodes at this level and get to know the node with interesting rules, for instance the node {whole milk} has a total of 555 rules which covers 74 items, the median lift of the rules on this node is 2.77 while median support is 0.0014, the median confidence is 0.07083.

A major limitation of grouped matrix visualization is that it doesn't have an html version, the interactive version relies only on grid graphic system, and this makes it impossible to import/save the interactive graph into html widget which can be opened outside r environment. With collapsible tree, the interactive graph can be exported to html widget which makes it easy to browse via web browsers this makes the approach useful when target users are not r programmers.

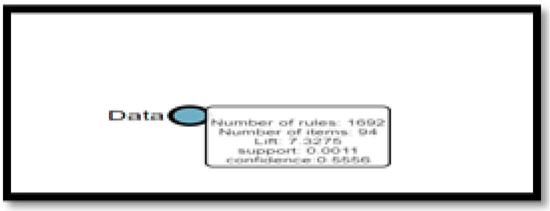

Fig. 6. A screenshot of showing the tooltip

5 Evaluation

Is the proposed framework based on the collapsible tree a good visualization for ARs? As we search for answers to these questions, we found many studies on evaluation methods and quality matrixes; to judge the effectiveness of the visualization. Some of these methods were taken from other fields, while the others specially developed for the visualization field. However, to determine the appropriate evaluation method for the visualization, some points need to take into consideration. For instance, the type of evaluation methodology needed to adopt (quantitative or qualitative), the evaluation objectives, and the extent of resource abundance [25].

The visualization evaluation is a complicated issue, as many challenges may reduce the importance of evaluation results or prevent get the optimal evaluation. Besides, many evaluation methods are based on cognitive reasoning research, perceptual psychology, and human-computer interaction research [25]. However, the evaluation is remaining essential.

According to [26], the basic evaluation criteria in the context of visual data mining include scalability, flexibility, ease of use, interaction techniques, process time, dimensionality, accuracy, and task analysis. With these points in mind, we assessed the quality of the collapsible tree regarding accuracy, scalability. The proposed evaluation framework is shown in figure 7. The next sub-sections provide more information about each of them.

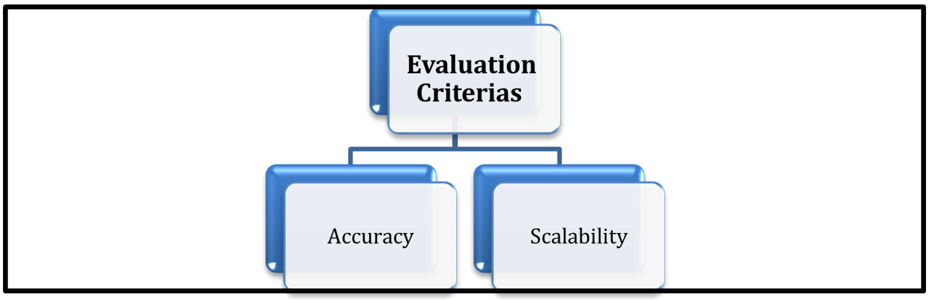

Fig. 7. The evaluation criteria`s framework

5.1 Accuracy

Was the visualization conversion from a grouped matrix to a collapsible tree done correctly? To be able to answer this question, we should measure the accuracy. Accuracy is one of the most important measures in evaluating the visualization. The accuracy of a measurement is how close it is to the true value.

The accuracy in the visualization can be evaluated for different considerations; this depends on the study focus. Some of these considerations are the human viewer, sampling, and algorithm. In this study, we focused on measuring the accuracy of the conversion process. To do that, we should compare the two visualizations regarding:

1. The number/names of consequents.
2. The number/names of clusters.
3. The number/names of rules.
4. The values of meta-data for each rule and each cluster.

The result of this comparison will show if the conversion process was done correctly or not. The true values are the values of the grouped matrix. If the values of both visualizations were the same, it means the conversion process was accurate. If they were different, then the conversion process was done incorrectly.

5.2 Scalability

The visualization scalability measures the visualization ability to display a large number of rules. The measurement will be calculated by comparing the number of rules that the grouped matrix was able to display, and the number of rules that the collapsible tree was able to view. If they were equal, this means that their ability is equal. If there was a difference, we will explain this difference and determine the reasons for its occurrence.

6 Result and Analysis

In this section, we elaborate and discuss the result and analysis of accuracy and scalability evaluations.

6.1 Result of Accuracy evaluation

After developing the collapsible tree from the grouped matrix, it is important to evaluate the converting process, is it correct or not? As mentioned before, we will evaluate the converting process conducted in the MBA case study which discussed previously at section 4. The converting process was evaluated from four aspects:

1. The number/names of consequents: The consequents in the grouped matrix are represented in rows; there are 25 unique consequent items as shown in figure 2. As for the collapsible tree, we can see there are also the same 25 consequents and represented at level two as shown in figure 5.
2. The number/names of clusters: In grouped matrix, the antecedent clusters are in columns. There 20 clusters of antecedents. Every cluster contains different number of antecedents. As the collapsible tree is a hierarchical representation, the antecedent clusters are put under the consequent which corresponds to (at level 3). Some of these consequents were associated to one antecedent cluster, and some associated to more clusters. However, there are no consequent associated with over 20 clusters. This is in line with the number of antecedents' groups in grouped matrix. The following Table 1 shows the consequent and its antecedent's clusters in the collapsible tree.
3. The number/names of rules: In this part, we are interested in the number of rules produced from the conversion process, not the number of rules represented in the collapsible tree (scalability issue). As shown in figure 2 the number of rules in the grouped matrix was 5668. On the other hand, the number of rules after the conversion is done has remained the same. Figure 8 shows a screenshot of 'R' after run the conversion functions. In the left side (plot section), we can see the grouped matrix that we used for converting. In the right side (editor section), we can see the data frame the output of the steps discussed previously. The first column in the table shows that all the 5668 rules were existing.
4. The values of meta-data: As each cluster contains many antecedents, the aggregation method, more specifically the median function, was used to represent the value of lift and support of each cluster. In the grouped matrix, the size of the balloon shows the aggregated lift, and the color of the balloon shows the aggregated support. If the user wants to explore rules he can pick any value (represented by the balloon) in the matrix then select zoom in feature. After zooming in, the sub-matrix will show up, contains in column side the subgroups of antecedents of the selected value, and in row side their associated consequents. The values of meta-date (support, confidence, and lift) are displayed in the console section after selected the inspect feature. While in the collapsible tree the values of meta-data are displayed in a tooltip, appear to the

user when putting the cursor at any node. Table 2 shows the values of the meta-data of randomly selected rules in both visualizations.

Table 1. The consequent and its antecedents' clusters in the collapsible tree.

Consequent	Number/antecedent groups
Whole milk	17
Rolls/buns	12
Other vegetable	14
Bottled beer	1
Root vegetable	6
Whipped / sour cream	2
Soda	4
Salty snack	1
Yogurt	10
Tropical fruit	5
Hamburger meat	1
Shopping bags	1
Bottled water	2
White bread	1
Sugar	1
Fruit/ vegetable juice	3
Pip fruit	1
Citrus fruit	1
Domestic eggs	3
Butter	1
Sausage	2
Curd	2
Cream cheese	1
Pastry	1
Beef	1

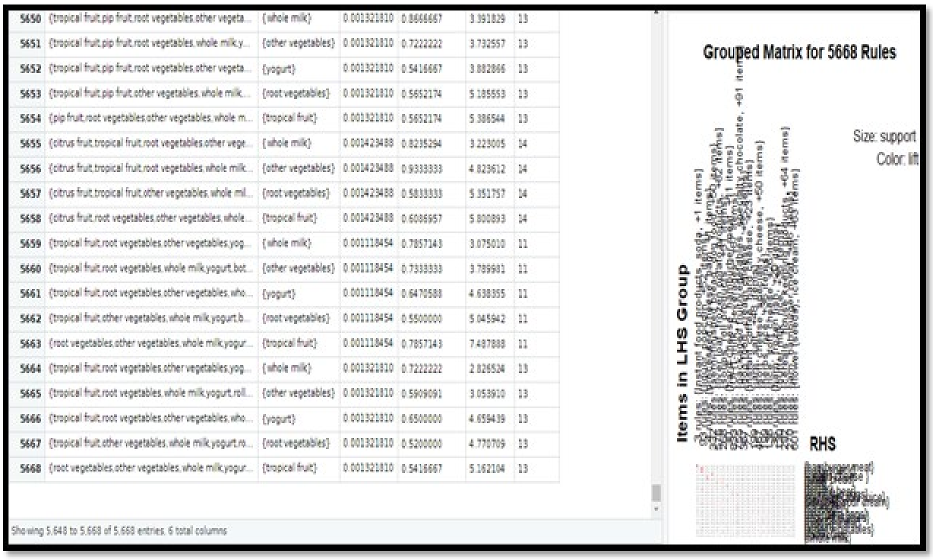

Fig. 8. Screenshot of R

Table 2. The values of the meta-dat

Rule	Grouped matrix			Collapsible tree		
	support	conf.	lift	support	conf.	lift
{Instant food products,soda} => {hamburger meat}	0.0012	0.631	18.995	0.0012	0.6316	18.995
{pork, whole milk, waffles} => {rolls/buns}	0.0010	0.588	3.1980	0.001	0.5882	3.1981
{sausage,root vegetables, whipped/sour cream}=> {yogurt}	0.0015	0.714	5.1202	0.0015	0.7143	5.1203
{waffles, candy} => {soda}	0.0015	0.517	2.9662	0.0015	0.5172	2.9662

We conclude from the previous table that the values of meta-data are the same in both visualizations. But in the collapsible tree, we used a decrease decimal method to show four digits after the decimal point, and that's the only difference between the two visualizations.

To sum up the result of accuracy evaluation, we can say that the conversion process was done accurately as all four aspects are the same in the two visualizations.

6.2 Result of Scalability evaluation

The grouped matrix and collapsible tree visualizations used the same clustering method to solve cluttering issues when displaying a large number of rules on screen. However, the scalability of both visualizations is not the same. The grouped matrix is capable of displaying 5668 rules without causing screen clutter. In contrast, the 5668 rules caused screen clutter by using a collapsible tree. That is due to the layout difference between matrix and tree in general. In the grouped matrix, the antecedents are split into a set of 20 clusters $S=\{S_1, S_2, S_{20}\}$. When the user wants to explore particular antecedent cluster S_i, he should select any value (balloon) then select zoom in feature. By doing this, another submatrix M_i will appear. The submatrix M_i has only the antecedents that in cluster S_i ,and in row side the consequents that associated with these antecedents. On the other side, the organization of the rules in the collapsible tree is different. Whereas, the clusters S_i are put under the consequent that associated with. So we find some consequents has associated with one cluster and other to more clusters which mean more rules to display, so caused screen cluttering. For instance, the consequent "whole milk" associate with 17 clusters of antecedents , and each cluster of them contains many antecedents (rules) shown in the final level. Of course "whole milk" is considered as a very frequent item in any grocery data set. Its presence in most of the rules is self-evident like many other items. However, to solve the cluttering issue we made a condition in the code. The condition states that" if the cluster contains more than 50 rules shows the first 50 rules based on lift value. If the number of rules less than 50, show all of them". By applying this condition, we managed to solve the cluttering issue, but the number of rules displayed by the collapsible tree was decreased from 5668 to 1628.

7 Conclusion

In this study, we tackled the issue of visualizing a large number of ARs. We proposed a new technique based on tree methods, specifically the collapsible tree. To reduce rules dimensionality, we used the clustering approach in the grouped matrix. With the already implemented the grouped matrix balloon plot, the clustering process and converting process into a collapsible tree has been simplified.

The accuracy evaluation showed that the converting process that we used was accurate. However, the scalability of the collapsible tree was less compared to the grouped matrix. Despite that, the collapsible tree is capable to visualize more ARs, than other visualizations [8, 11].

For future work, we plan to add features such as filtering to allow users to display the rules that meet certain conditions. In addition, we plan to apply the visualization framework to different marketing areas.

References

1. Olson, D. L., Lauhoff, G.: Descriptive data mining. In: Descriptive Data Mining. pp. 129-130, Springer, Singapore (2019)

2. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI magazine, 17(3), 37-37(1996)
3. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp. 207-216 (1993)
4. Hahsler, M., Karpienko, R.: Visualizing association rules in hierarchical groups. Journal of Business Economics, 87(3), 317-335 (2017).
5. Bayardo Jr, R. J., Agrawal, R.: Mining the most interesting rules. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 145-154 (1999).
6. Hofmann, H., Siebes, A. P., Wilhelm, A. F.: Visualizing association rules with interactive mosaic plots. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 227-235 (2000).
7. Rensink, R. A.: On the prospects for a science of visualization. In Handbook of human centric visualization , pp. 147-175. Springer-New York, NY (2014) .
8. Wang, B., Zhang, T., Chang, Z., Ristaniemi, T., Liu, G.: 3D matrix-based visualization system of association rules. In 2017 IEEE International Conference on Computer and Information Technology (CIT), pp. 357-362. IEEE (2017).
9. Khan, M., Khan, S. S.: Data and information visualization methods, and interactive mechanisms: A survey. International Journal of Computer Applications, 34(1), 1-14 (2011).
10. Itoh, T., Yamaguchi, Y., Ikehata, Y., Kajinaga, Y.: Hierarchical data visualization using a fast rectangle-packing algorithm. IEEE Transactions on Visualization and Computer Graphics, 10(3), 302-313 (2004).
11. Sekhavat, Y. A., Hoeber, O.:Visualizing association rules using linked matrix, graph, and detail views. (2013).
12. Liu, Y., Salvendy, G.: Design and evaluation of visualization support to facilitate decision trees classification. International journal of human-computer studies, 65(2), 95-110 (2007).
13. Simoff, S., Böhlen, M. H., Mazeika, A :Visual data mining: theory, techniques and tools for visual analytics. Science & Business Media Vol. 4404, Springer (2008).
14. Codd, E. F.: A relational model of data for large shared data banks. In Software pioneers, pp. 263-294. Springer- Berlin, Heidelberg (2002).
15. Romero, C., Luna, J. M., Romero, J. R., Ventura, S.: RM-Tool: A framework for discovering and evaluating association rules. Advances in Engineering Software, 42(8), 566-576(2011).
16. Mazeika, A., Böhlen, M. H., Mylov, P.: Using nested surfaces for visual detection of structures in databases. In Visual Data Mining , pp. 91-102 , Springer-Berlin, Heidelberg (2008).
17. Sitanggang, I. S.: Spatial multidimensional association rules mining in forest fire data. Journal of Data Analysis and Information Processing, 1(04), 90 (2013).
18. Kopanakis, I., Theodoulidis, B.: Visual data mining & modeling techniques. In Proceedings of the KDD-2001 workshop on visual data mining (2001).
19. Yang, L.: Visualizing frequent itemsets, association rules, and sequential patterns in parallel coordinates. In International Conference on Computational Science and Its Applications, pp. 21-30. Springer- Berlin, Heidelberg (2003).
20. Bruzzese, D., Buono, P.: Combining visual techniques for association rules exploration. In Proceedings of the working conference on Advanced visual interfaces, pp. 381-384 (2004)
21. Ertek, G., Demiriz, A.: A framework for visualizing association mining results. In International Symposium on Computer and Information Sciences , pp. 593-602. Springer- Berlin, Heidelberg (2006).
22. Wong, P. C., Whitney, P., Thomas, J.: Visualizing association rules for text mining. In Proceedings 1999 IEEE Symposium on Information Visualization, pp. 120-123. IEEE (1999).

23. Khan, A. (2018, May 5). Collapsible Tree: Geography Example. Github. https://adeelk93.github.io/collapsibleTree/
24. M. Hahsler, K. Hornik, and T. Reutterer, "Implications of Probabilistic Data Modeling for Mining Association Rules," From Data Inf. Anal. to Knowl. Eng., vol. 2, pp. 598–605, 2006.
25. Carpendale, S.: Evaluating information visualizations. In Information visualization , pp. 19-45 . Springer-Berlin, Heidelberg (2008).
26. Grinstein, G. G., Hoffman, P., Pickett, R. M., & Laskowski, S. J.: Benchmark development for the evaluation of visualization for data mining. Information visualization in data mining and knowledge discovery, 129-176 (2002).