Visualizing Association Rules in Hierarchical Groups[*]

Michael Hahsler and Sudheer Chelluboina

May 19, 2011

Abstract

Association rule mining is one of the most popular data mining methods. However, mining association rules often results in a very large number of found rules, leaving the analyst with the task to go through all the rules and discover interesting ones. Sifting manually through large sets of rules is time consuming and strenuous. Visualization has a long history of making large amounts of data better accessible using techniques like selecting and zooming. However, most association rule visualization techniques are still falling short when it comes to a large number of rules. In this paper we present a new interactive visualization technique which lets the user navigate through a hierarchy of groups of association rules. We demonstrate how this new visualization techniques can be used to analyze a large sets of association rules with examples from our implementation in the R-package **arulesViz**.

1 Introduction

Many organizations are generating a large amount of transaction data on a daily basis. For example, a department store like "Macy's" stores customer shopping information at a large scale using check-out data. Association rule mining is one of the major techniques to detect and extract useful information from large scale transaction data. Mining association rules was fist introduced by Agrawal et al. (1993) and can formally be defined as:

Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of n binary attributes called *items*. Let $\mathcal{D} = \{t_1, t_2, \dots, t_m\}$ be a set of transactions called the *database*. Each transaction in $\mathcal{D}$ has a unique transaction ID and contains a subset of the items in I. A *rule* is defined as an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The sets of items (for short *itemsets*) X and Y are called *antecedent* (left-hand-side or LHS) and *consequent* (right-hand-side or RHS) of the rule. Often rules are restricted to only a single item in the consequent.

Association rules are rules which surpass a user-specified minimum support and minimum confidence threshold. The *support* $\text{supp}(X)$ of an itemset X is defined as the proportion of transactions in the data set which contain the itemset and the *confidence* of a rule is defined $\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y)/\text{supp}(X)$. Therefore, an association rule $X \Rightarrow Y$ will satisfy:

$$\text{supp}(X \cup Y) \geq \sigma$$

and

$$\text{conf}(X \Rightarrow Y) \geq \delta$$

where σ and δ are the minimum support and minimum confidence, respectively. Note that both minimum support and minimum confidence are related to statistical concepts. Finding itemsets which surpass a minimum support threshold can be seen as a simplification of the unsupervised statistical learning problem called "mode finding" or "bump hunting" (Hastie et al., 2001) where each item is seen as a variable and the goal is to find prototype values so that the probability density evaluated at these values is sufficiently large. Minimum confidence can be interpreted as a threshold on the estimated conditional

[*] 42nd Symposium on the Interface: Statistical, Machine Learning, and Visualization Algorithms (Interface 2011)

probability $P(Y|X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS (see e.g., Hipp et al., 2000).

Another popular measure for association rules used throughout this paper is *lift* (Brin et al., 1997). The lift of a rule is defined as

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)\text{supp}(Y)}$$

and can be interpreted as the deviation of the support of the whole rule from the support expected under independence given the supports of both sides of the rule. Greater lift values ($\gg 1$) indicate stronger associations. Measures like support, confidence and lift are generally called interest measures because they help with focusing on potentially more interesting rules.

For example, let us assume that we find the rule {milk, bread} ⇒ {butter} with support of 0.2, confidence of 0.9 and lift of 2. Now we know that 20% of all transactions contain all three items together, the estimated conditional probability of seeing butter in a transaction under the condition that the transaction also contains milk and bread is 0.9, and we saw the items together in transactions at double the rate we would expect under intependence between the itemsets {milk, bread} and {butter}. For a more detailed treatment of association rules we refer the reader to the introductory paper for the R-package **arules** (Hahsler et al., 2005) and the literature referred to there.

Association rules are typically generated in a two-step process. First, minimum support is used to generate the set of all *frequent itemsets* for the data set. Frequent itemsets are itemsets which satisfy the minimum support constraint. Then, in a second step, each frequent itemsets is used to generate all possible rules from it and all rules which do not satisfy the minimum confidence constraint are removed. Analyzing this process, it is easy to see that in the worst case we will generate $2^n - n - 1$ frequent itemsets with more than two items from a database with n distinct items. Since each frequent itemset will in the worst case generate at least two rules, we will end up with a set of rules in the order of $O(2^n)$. Typically, users increase the minimum support threshold σ to keep the number of association rules found at a manageable size. However, this has the disadvantage that it removes potentially interesting rules with lower support. Therefore, the need to deal with large sets of association rules is unavoidable when applying association rule mining in a real setting.

Many researchers introduced visualization techniques like scatter plots, matrix visualizations, graphs, mosaic plots and parallel coordinates plots to analyze association rules. However, most existing visualization techniques are not suitable for displaying large sets of rules. This paper introduces a completely new method called “grouped matrix-based visualization” which is based on a novel way of creating nested groups of rules (more specifically antecedent itemsets) via clustering. The nested groups form a hierarchy which can be interactively explored down to the individual rule.

The rest of the paper is organized as follows. In Section 2 we discuss existing visualization techniques for association rules. In Section 3 we introduce matrix-based visualization and illustrate with an example its shortcomings. The new grouped matrix-based visualization is introduced and illustrated in Section 4. Section 5 concludes the paper.

2 Related Work

In the last decade several visualization techniques have been developed. In the following we review the most important types. A more thorough overview is provided by Bruzzese and Davino (2008).

A straight-forward visualization of association rules is to use a scatter plot with two interest measures (typically support and confidence) on the axes. Such a presentation can be found already in an early paper by Bayardo, Jr. and Agrawal (1999) when they discuss *sc-optimal rules.* Unwin et al. (2001) introduced a special version of a scatter plot called *Two-key plot.* Here support and confidence are used for the x and y-axes and the color of the points is used to indicate “order,” i.e., the number of items contained in the rule. Scatter plots work well for very large sets of association rules and zooming into the plot can be easily implemented. The main drawback is there is not enough space in the plot the display the labels of the items in the rules. In an interactive version of this plot this information can be obtained by selecting a single or a small set of rules. However, this still makes exploring a large set of rules time-consuming.

Graph-based techniques (Klemettinen et al., 1994; Rainsford and Roddick, 2000; Buono and Costabile, 2005; Ertek and Demiriz, 2006) visualize association rules using vertices and directed edges where vertices typically represent items or itemsets and edges connect antecedents and consequents in rules. Interest measures are typically added to the plot as labels on the edges or by color or width of the arrows displaying the edges. Graph-based visualization offers a very clear representation of rules but they tend to easily become cluttered and thus are only viable for very small sets of rules. To explore large sets of rules with graphs, different layout mechanisms and advanced interactive features like zooming, filtering, grouping and coloring nodes are vertices. Such features are available in interactive visualization and exploration platforms for networks and graphs like *Gephi* (Bastian et al., 2009).

Parallel coordinates plots are designed to visualize multidimensional data where each dimension is displayed separately on the x-axis and the y-axis is shared. Each data point is represented by a line connecting the values for each dimension. Parallel coordinates plots were used previously to visualize discovered classification rules (Han et al., 2000) and association rules (Yang, 2003). Yang (2003) displays the items on the y-axis as nominal values and the x-axis represents the positions in a rule, i.e., first item, second item, etc. Instead of a simple line an arrow is used where the head points to the consequent item. Arrows only span enough positions on the x-axis to represent all the items in the rule, i.e., rules with less items are shorter arrows. Parallel coordinates plots are prone to clutter caused by crossing lines. Reordering the items on the x and y-axes can alleviate some of the problems but it still makes large sets of rules hard to analyze.

A double decker plot is a variant of a mosaic plot which displays a contingency table using tiles created by recursively vertical and horizontal splitting a rectangle. The size of each tile is proportional to the value in the contingency table. Double decker plots use only a single horizontal split. Hofmann et al. (2000) introduced double decker plots to visualize a single association rule. Here the displayed contingency table is computed for a rule by counting the occurrence frequency for each subset of items in the antecedent and consequent in the original data set. The items in the antecedent are used for the vertical splits and the consequent item is used for horizontal highlighting. Although this visualization is powerful for analyzing a single rule, it cannot be used for large sets of rules.

Another popular method is matrix-based visualization. We introduce it in more detail in the following section since it is related to the new technique presented in this paper.

3 Matrix-based Visualization

Matrix-based visualization techniques organize the antecedent and consequent itemsets on the x and y-axes, respectively. A selected interest measure is displayed at the intersection of the antecedent and consequent of a given rule. If no rule is available for an antecedent/consequent combination the intersection area is left blank.

Formally, the visualized matrix is constructed as follows. We start with the set of association rules

$$\mathcal{R} = \{\langle X_1, Y_1, \theta_1\rangle, \ldots, \langle X_i, Y_i, \theta_i\rangle, \ldots, \langle X_n, Y_n, \theta_n\rangle\}$$

where X_i is the antecedent, Y_i is the consequent and θ_i is the selected interest measure for the i-th rule, $i = 1, \ldots, n$. In $\mathcal{R}$ we identify the set of A unique antecedents and C unique consequent. We create a $A \times C$ matrix $\mathbf{M} = (m_{ac})$, $a = 1, \ldots, A$ and $c = 1, \ldots, C$, with one column for each unique antecedent and one row for each unique consequent. We populate the matrix by setting $m_{ac} = \theta_i$ where $i = 1, \ldots, n$ is the rule index, and a and c correspond to the position of X_i and Y_i in the matrix. Note that $\mathbf{M}$ will contain many empty cells since many potential association rules will not meet the required minimum thresholds on support and confidence.

Ong et al. (2002) presented a version of the matrix-based visualization technique where a 2-dimensional matrix is used and the interest measure is represented by color shading of squares at the intersection. An alternative visualization option is to use 3D bars at the intersection (Wong et al., 1999; Ong et al., 2002).

For this type of visualization the number of rows/columns depends on the number of unique itemsets in the consequent/antecedent in the set of rules. Since large sets of rules typically have a large number of different itemsets as antecedents (often not much smaller than the number of rules themselves), the size of the colored squares or the 3D bars gets very small and hard to see.

We illustrate matrix-based visualization using the R-package **arulesViz** (Hahsler and Chelluboina, 2011) an extension for package **arules** (Hahsler et al., 2010). For the examples in this paper we load the "Groceries" data set which is included in **arules**.

```
> library("arulesViz")
> data("Groceries")
> Groceries

transactions in sparse format with
 9835 transactions (rows) and
 169 items (columns)
```

Groceries contains sales data from a local grocery store with 9835 transactions and 169 items (product groups). The data sets most popular item is "whole milk" and the average transaction contains less than 5 items. Next we mine association rules using the Apriori algorithm implemented in **arules**. We use $\sigma = 0.001$ and $\delta = 0.5$.

```
> rules <- apriori(Groceries, parameter = list(support = 0.001,
+     confidence = 0.5), control = list(verbose = FALSE))
> rules

set of 5668 rules
```

The result is a set of 5668 association rules. The top three rules with respect to the lift measure are:

```
> inspect(head(sort(rules, by = "lift"), 3))

  lhs                          rhs                  support confidence     lift
1 {Instant food products,
   soda}                    => {hamburger meat} 0.001220132  0.6315789 18.99565
2 {soda,
   popcorn}                 => {salty snack}    0.001220132  0.6315789 16.69779
3 {flour,
   baking powder}           => {sugar}          0.001016777  0.5555556 16.40807
```

These rules represent easy to explain purchasing patterns. However, it is clear that going through all the 5668 rules manually is not a viable option. Therefore, we create a matrix-based visualization using shaded squares and 3D bars.

```
> plot(rules, method = "matrix", measure = "lift")

> plot(rules, method = "matrix3D", measure = "lift")
```

The resulting plots are shown in Figures 1 and 2. The rules contain 4097 unique antecedent and 25 unique consequent itemsets. Since there is not much space for long labels in the plot, we only show numbers as labels for rows and columns (x and y-axis) and the complete itemsets are printed to the terminal for look-up. We omit the complete output here, since this plot print several thousand labels in the following form to the screen.

```
Itemsets in Antecedent (lhs)
  [1] "{liquor,red/blush wine}"
  [2] "{curd,cereals}"
  [3] "{yogurt,cereals}"
  [4] "{butter,jam}"
  [5] "{soups,bottled beer}"
```

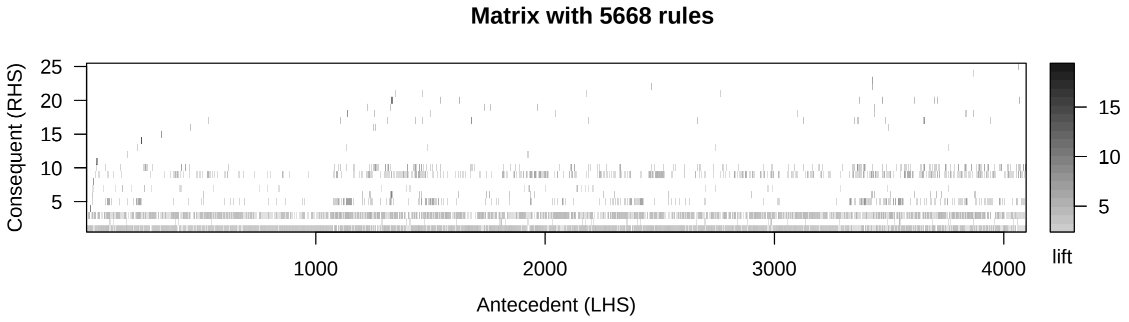

Figure 1: Matrix-based visualization with colored squares.

```
           (lines omitted)
  [343] "{tropical fruit,root vegetables,rolls/buns,bottled water}"
  [344] "{tropical fruit,root vegetables,yogurt,rolls/buns}"
Itemsets in Consequent (rhs)
  [1] "{bottled beer}"     "{whole milk}"       "{other vegetables}"
  [4] "{tropical fruit}"   "{yogurt}"           "{root vegetables}"
           (lines omitted)
```

Obviously matching the labels to the entries on the x and y-axis is cumbersome. In order to be able to print the complete labels on the axes we would have to reduce the number of rules significantly to typically less than 100 rules. Alternatively, rules in the plot can be interactively selected to reveal the rule's antecedent and consequent itemsets, but the plot is so crowded, that it is almost impossible to select a specific rule. In Hahsler and Chelluboina (2011) we experimented with several reordering strategies to improve the plots usefulness for large number of rules, but only with very limited success. This illustration clearly shows that the usefulness of simple matrix-based visualization is very limited when facing large rule sets.

4 Grouped matrix-based visualization

Matrix-based visualization is limited in the number of rules it can visualize effectively since large sets of rules typically also have large sets of unique antecedents/consequents. Here we introduce a new visualization techniques that enhances matrix-based visualization by grouping rules via clustering to handle large sets of rules. Groups of rules are presented by aggregating rows/columns of the matrix. The groups are nested and organized hierarchically allowing the user to explore them interactively by zooming into groups.

A direct approach to clustering itemsets (and rules) is to define a distance metric between two itemsets X_i and X_j. Distance between two sets can be measured for example by the Jaccard distance defined as

$$d_{\text{Jaccard}}(X_i, X_j) = 1 - \frac{|X_i \cap X_j|}{|X_i \cup X_j|}.$$

This distance is based on the number of items that X_i and X_j have in common divided by the number of unique items in both sets and was called for clustering association rules *conditional market-basket probability* by Gupta et al. (1999). For a set of m rules we can calculate the $m(m-1)/2$ distances between the sets of all items in each rule and use them as the input for clustering. However, using clustering on the itemsets creates several problems. First of all, data sets typically mined for association rules are high-dimensional, i.e., contain many different items. This high dimensionality carries over to the mined rules and leads to a situation referred is as the "course of dimensionality" where, due to the exponentially increasing volume, distance functions lose their usefulness. The situation is getting worse

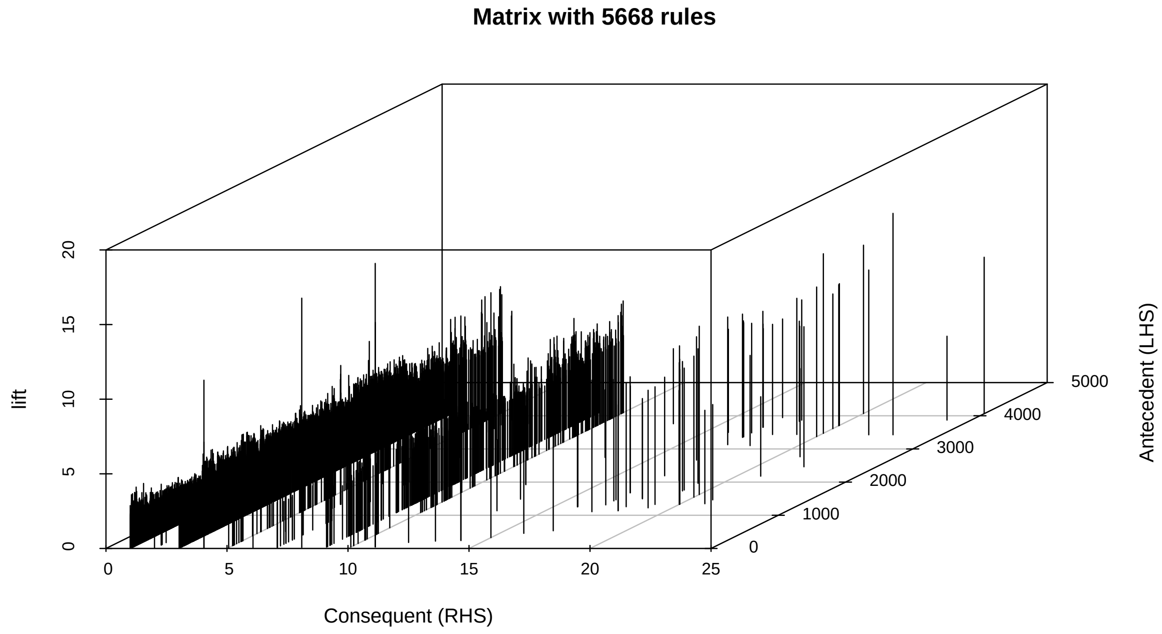

Figure 2: Matrix-based visualization with 3D bars.

since minimum support used in association rule mining leads in addition to relatively short rules resulting in extremely sparse data.

Several approaches for clustering association rules and itemsets to address the dimensionality and sparseness problem were proposed in the literature. Toivonen et al. (1995) and Berrado and Runger (2007) propose clustering association rules by looking at the number of transactions which are covered by the rules. A transaction is covered by a rule if it contains all the items in the rule's antecedent. Using common covered transactions avoids the problems of clustering sparse, high-dimensional binary vectors. However, it introduces a strong bias towards clustering rules which are generated from the same frequent itemset. By definition two subsets of a frequent itemset will cover many common transactions. This bias will lead to just rediscovering the already known frequent itemset structure from the set of association rules.

Here we pursue a completely different approach. We start with the matrix $\mathbf{M}$ defined in Section 3 which contains the values of a selected interest measure of the rules in set $\mathcal{R}$. The columns/rows are the unique antecedents/consequents in $\mathcal{R}$, respectively. Now grouping rules becomes the problem of grouping columns or rows in $\mathbf{M}$.

Since for most applications the consequents in mined rules are restricted to a single item there is no problem with combinatorial explosion and we can restrict our treatment to only grouping antecedents (i.e., columns in $\mathbf{M}$). However, note that the same grouping method can be used also for consequents.

We use the interest measure lift here, but other interest measures can be used as well. The idea for lift is that antecedents that are statistically dependent on the same consequents (i.e., have a high lift value) are similar and thus should be grouped together. Compared to other clustering approaches for itemsets, this method enables us to even group antecedents containing substitutes (e.g., butter and margarine) which are rarely purchased together since they will have a similar dependence relationship with the same consequents (e.g., bread). Clustering based on shared items or common covered transaction cannot uncover this type of relationship.

For grouping we propose to split the set of antecedents into a set of k groups $\mathcal{S} = \{S_1, S_2, \dots, S_k\}$

while minimizing the within-cluster sum of squares

$$\operatorname{argmin}_{\mathcal{S}} \sum_{i=1}^{k} \sum_{\mathbf{m}_j \in S_i} ||\mathbf{m}_j - \boldsymbol{\mu}_i||^2,$$

where $\mathbf{m}_j$, $j = 1, \ldots, A$, is a column vector representing all rules with the same antecedent and $\boldsymbol{\mu}_i$ is the center (mean) of the vectors in S_i. Minimizing the stated loss function is known as the k-means problem which is *NP-hard* (Aloise et al., 2009). However, several good and fast heuristics exist which do not require a precomputed distance matrix. We use the k-means algorithm by Hartigan and Wong (1979) and restart it 10 times with random initialized centers to find a suitable solution.

To visualize the grouped matrix we use a balloon plot with antecedent groups as columns and consequents as rows (see Figure 3). The color of each balloon represents the aggregated interest measure in the group and the size of the balloon shows the aggregated support. Aggregation in groups can be achieved by several aggregation functions (e.g., maximum, minimum, average, median). In the examples in this paper we use the median to represent the group since it is robust against outliers. The number of rules and the most important (frequent) items in the group are displayed as the labels for the columns followed by the number of other items in that antecedent group. Furthermore, the columns and rows in the plot are reordered such that the aggregated interest measure is decreasing from top down and from left to right, directing the user to the most interesting group in the top left corner.

To allow the user to explore the whole set of rules we can create a hierarchical structure of subgroups. This is simply achieved by creating for each group $S_i, i = 1, \ldots, k$, a submatrix $\mathbf{M}_i$ which only contains the columns corresponding to the elements in S_i. Now we can use the use same grouping process again on a submatrix selected by the user. This allows the user to recursively "drill down" into the rule set. An advantage of this process is that we only need to run the k-means algorithm on demand when the user wants to explore a group further.

A chellange with using the k-means algorithm is that $\mathbf{M}$ contains many missing values for rules which are not included in $\mathcal{R}$ since they do not pass the minimum support or minimum confidence threshold. Since most values will be missing, marginalization (i.e., remove antecedents/consequents with missing values) is not an option and we use imputation. Imputation strategies typically assume that the values are missing randomly which is not the case here. Values miss in our case systematically when rules do not meet the support and confidence thresholds and thus are deemed not interesting. This means that we would like to group antecedents when they have many missing values with the same set of consequents in common. To achieve this we replace all missing lift values with 1, a value indicating that antecedent and consequent of the rule are statistically independent. This ensures that matching missing values will contribute positively for grouping while it will help to separate them from existing rules with most likely larger lift values.

The matrix visualization with grouped antecedents for the set of 5668 rules mined earlier can be easily created with **arulesViz** by

```
> plot(rules, method = "grouped", control = list(k = 20))
```

The resulting visualization uses $k = 20$ groups and is shown in Figure 3. The group with the most interesting rules according to lift (the default measure) are shown in the top-left corner of the plot. There are 3 rules which contain "Instant food products" and up to 2 other items in the antecedent and the consequent is "hamburger meat."

In the interactive version an antecedent group can be inspected by selecting a column. The result for the left-most group is:

```
Selected rules:
  lhs                        rhs                    support confidence       lift
  1 {Instant food products,
     soda}                  => {hamburger meat}    0.001220132    0.631579 18.995654
  2 {whole milk,
     Instant food products} => {hamburger meat}    0.001525165    0.500000 15.038226
```

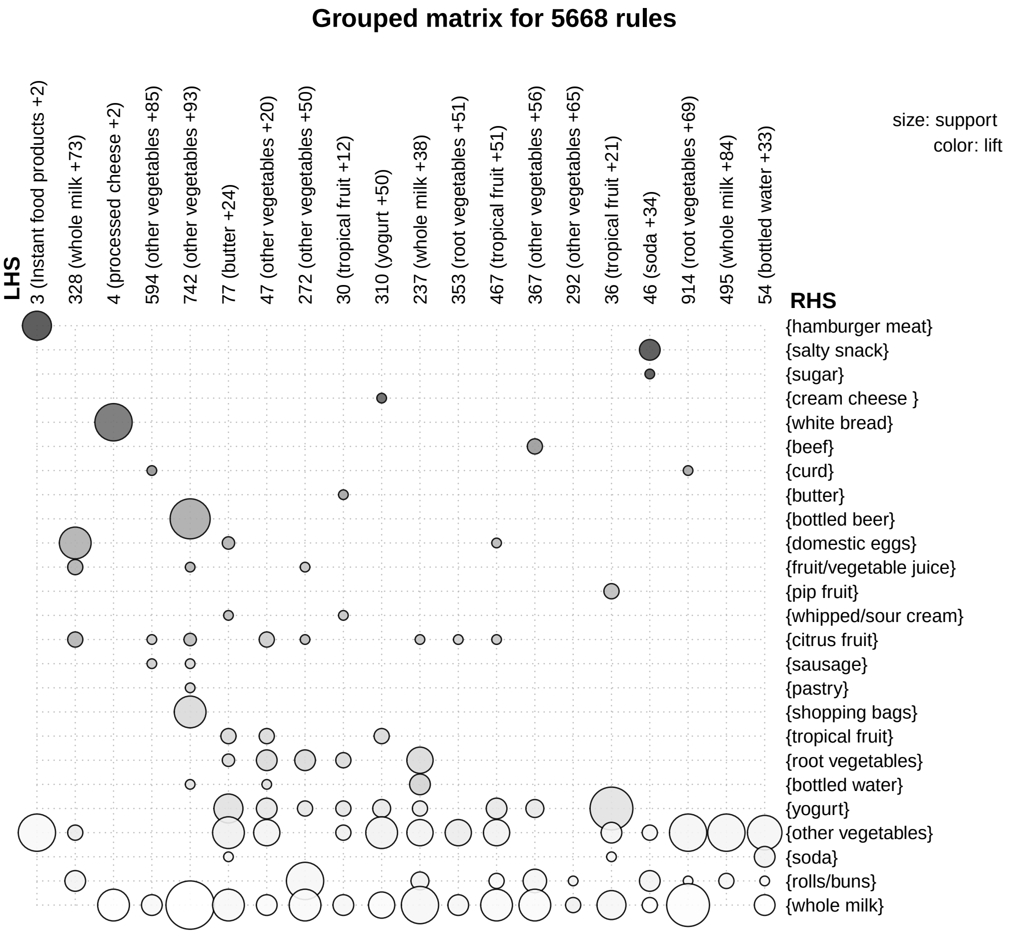

Figure 3: Grouped matrix-based visualization.

```
  3 {whole milk,
     Instant food products} => {other vegetables} 0.001525165   0.500000  2.584078
```

The first two rules with rather large lift values are represented in Figure 3 by the upper balloon. While the third weaker rules is the second balloon in the figure.

The grouped matrix visualization can be used interactively to zoom into groups and inspect rules. Figure 4 shows the interactive version zoomed into the 4th group in Figure 3. This group contains 594 rules and the most common item in the antecedents is *other vegetables*. In Figure 4 we see that all antecedents have *whole milk* as a possible consequent. This can be easily explained by the fact that milk is the most frequent item in the whole data set. Most groups are very similar and they are only split into several groups because we require the algorithm to split the antecedents into $k = 20$ groups. However, there are a few different antecedent groups which have *curd*, *citrus fruit* and *sausage* also as consequents. These groups contain more interesting rules since they have a higher lift (darker color of the balloon) and they are displayed therefore closer to the top-left corner. Using the inspect button we look at the 2 rules in the the first antecedent group.

```
Selected rules:
  lhs                       rhs               support confidence      lift
```

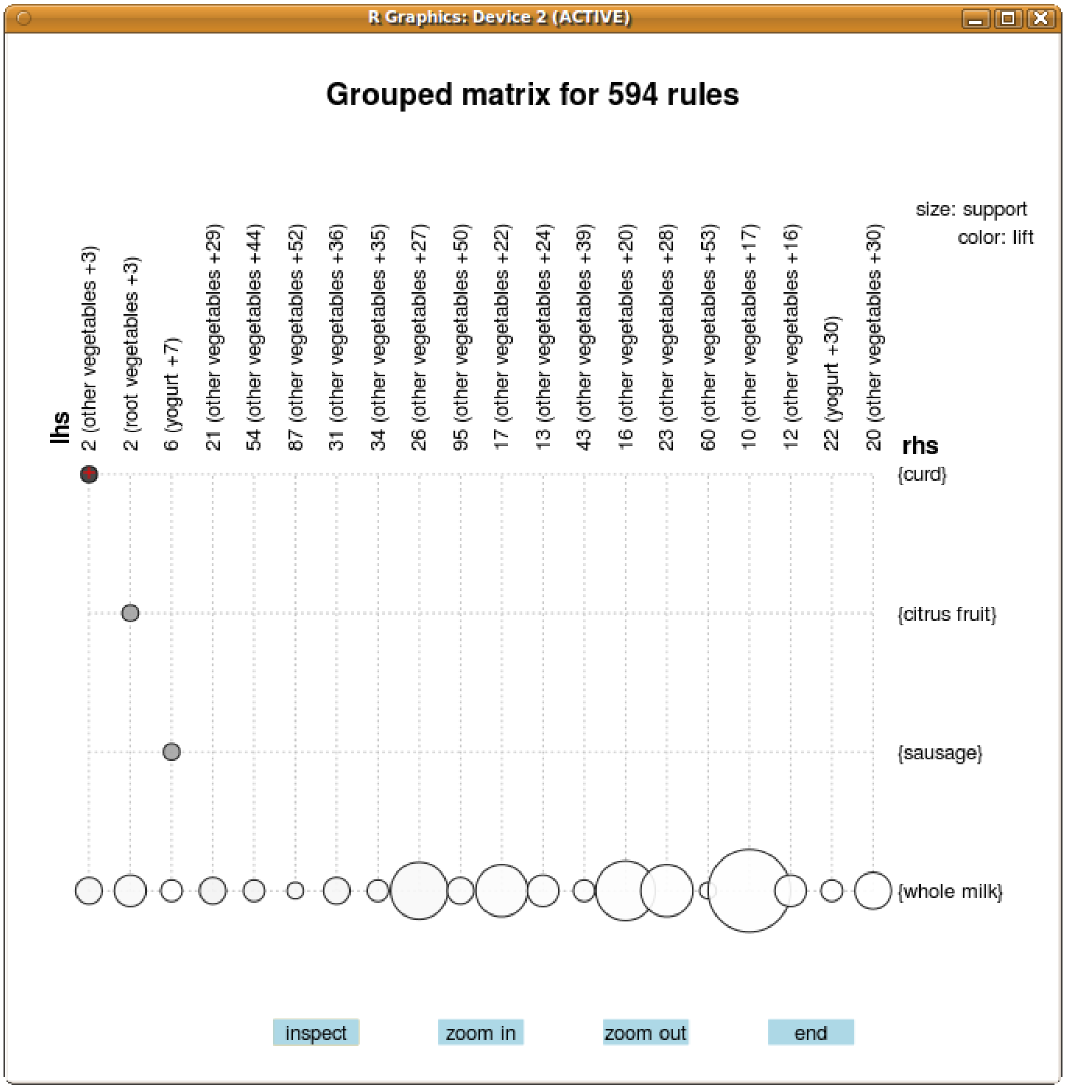
R Graphics: Device 2 (ACTIVE)
Grouped matrix for 594 rules
size: support
color: lift
lhs
2 (other vegetables +3)
2 (root vegetables +3)
6 (yogurt +7)
21 (other vegetables +29)
54 (other vegetables +44)
87 (other vegetables +52)
31 (other vegetables +36)
34 (other vegetables +35)
26 (other vegetables +27)
95 (other vegetables +50)
17 (other vegetables +22)
13 (other vegetables +24)
43 (other vegetables +39)
16 (other vegetables +20)
23 (other vegetables +28)
60 (other vegetables +53)
10 (other vegetables +17)
12 (other vegetables +16)
22 (yogurt +30)
20 (other vegetables +30)
rhs
{curd}
{citrus fruit}
{sausage}
{whole milk}
inspect
zoom in
zoom out
end

Figure 4: Interactive grouped matrix-based visualization (zoomed into the 4th group in Figure 3).

```
1 {other vegetables,
   yogurt,
   whipped/sour cream,
   cream cheese }      => {curd}       0.001016777  0.5882353 11.040638
2 {other vegetables,
   yogurt,
   whipped/sour cream,
   cream cheese }      => {whole milk} 0.001220132  0.7058824  2.762576
```

Although the first rule has a lower support and confidence, it is the more important rules with an extremely high lift. The whole rule set can be explored by zooming in and out of groups and by inspecting different subsets of rules.

5 Conclusion

In this paper we introduced a completely new visualization method for association rules. The method addresses the problem that sets of mined association rules are typically very large by grouping antecedents and allowing the user to interactively explore a hierarchy of nested groups. Coloring and the position of elements in the plot guide the user automatically to the most interesting groups/rules. Finally, the ability to interpret the grouped matrix-based visualization can be easily acquired since it is based on the easy to understand concepts of matrix-based visualization of association rules and grouping.

Grouped matrix-based visualization is unique in the way that most other visualization methods (see Bruzzese and Davino, 2008) are not able to efficiently deal with very large sets of association rules and that no existing method can handle complementary items.

Interesting areas for future research are to explore different other ways to group antecedents and to look at grouping antecedents and consequents simultaneously (i.e., by co-clustering/two-mode clustering).

References

Agrawal, R., Imielinski, T., and Swami, A. (1993), "Mining Association Rules between Sets of Items in Large Databases," in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, ACM Press, pp. 207–216.

Aloise, D., Deshpande, A., Hansen, P., and Popat, P. (2009), "NP-hardness of Euclidean sum-of-squares clustering," *Machine Learning*, 75, 245–248, 10.1007/s10994-009-5103-0.

Bastian, M., Heymann, S., and Jacomy, M. (2009), "Gephi: An Open Source Software for Exploring and Manipulating Networks," pp. 361–362.

Bayardo, Jr., R. J. and Agrawal, R. (1999), "Mining the most interesting rules," in *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 145–154.

Berrado, A. and Runger, G. C. (2007), "Using metarules to organize and group discovered association rules," *Data Mining and Knowledge Discovery*, 14, 409–431.

Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. (1997), "Dynamic Itemset Counting and Implication Rules for Market Basket Data," in *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, USA, pp. 255–264.

Bruzzese, D. and Davino, C. (2008), "Visual Mining of Association Rules," in *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*, Springer-Verlag, pp. 103–122.

Buono, P. and Costabile, M. F. (2005), "Visualizing Association Rules in a Framework for Visual Data Mining," in *From Integrated Publication and Information Systems to Virtual Information and Knowledge Environments*, pp. 221–231.

Ertek, G. and Demiriz, A. (2006), "A Framework for Visualizing Association Mining Results," in *ISCIS*, pp. 593–602.

Gupta, G., Strehl, A., and Ghosh, J. (1999), "Distance Based Clustering of Association Rules," in *Intelligent Engineering Systems Through Artificial Neural Networks (Proceedings of ANNIE 1999)*, ASME Press, pp. 759–764.

Hahsler, M., Buchta, C., Grün, B., and Hornik, K. (2010), *arules: Mining Association Rules and Frequent Itemsets*, R package version 1.0-3.

Hahsler, M. and Chelluboina, S. (2011), *arulesViz: arulesViz - Visualizing Association Rules*, R package version 0.1-1.

Hahsler, M., Grün, B., and Hornik, K. (2005), "arules – A Computational Environment for Mining Association Rules and Frequent Item Sets," *Journal of Statistical Software*, 14, 1–25.

Han, J., An, A., and Cercone, N. (2000), *CViz: An Interactive Visualization System for Rule Induction*, Springer Berlin / Heidelberg, pp. 214–226.

Hartigan, J. A. and Wong, M. A. (1979), "A K-means clustering algorithm," *Applied Statistics*, 28, 100–108.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning (Data Mining, Inference and Prediction)*, Springer Verlag.

Hipp, J., Güntzer, U., and Nakhaeizadeh, G. (2000), "Algorithms for Association Rule Mining – A General Survey and Comparison," *SIGKDD Explorations*, 2, 1–58.

Hofmann, H., Siebes, A., and Wilhelm, A. F. X. (2000), "Visualizing Association Rules with Interactive Mosaic Plots," in *KDD*, pp. 227–235.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., and Verkamo, A. I. (1994), "Finding Interesting Rules from Large Sets of Discovered Association Rules," in *CIKM*, pp. 401–407.

Ong, K.-H., leong Ong, K., Ng, W.-K., and Lim, E.-P. (2002), "CrystalClear: Active Visualization of Association Rules," in *In ICDM'02 International Workshop on Active Mining AM2002.*

Rainsford, C. P. and Roddick, J. F. (2000), "Visualisation of Temporal Interval Association Rules," in *IDEAL '00: Proceedings of the Second International Conference on Intelligent Data Engineering and Automated Learning, Data Mining, Financial Engineering, and Intelligent Agents*, Springer-Verlag, pp. 91–96.

Toivonen, H., Klemettinen, M., Ronkainen, P., Hatonen, K., and Mannila, H. (1995), "Pruning and Grouping Discovered Association Rules," in *Proceedings of KDD'95.*

Unwin, A., Hofmann, H., and Bernt, K. (2001), "The TwoKey Plot for Multiple Association Rules Control," in *PKDD '01: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, Springer-Verlag, pp. 472–483.

Wong, P. C., Whitney, P., and Thomas, J. (1999), "Visualizing Association Rules for Text Mining," in *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*, Washington, DC, USA: IEEE Computer Society, p. 120.

Yang, L. (2003), "Visualizing Frequent Itemsets, Association Rules, and Sequential Patterns in Parallel Coordinates," in *Computational Science and Its Applications – ICCSA 2003*, Lecture Notes in Computer Science, pp. 21–30.