Visualizing Association Rules: Introduction to the R-extension Package arulesViz

Michael Hahsler
Southern Methodist University

Sudheer Chelluboina
Southern Methodist University

Abstract

Association rule mining is a popular data mining method available in R as the extension package **arules**. 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 data sets better accessible using techniques like selecting and zooming. In this paper we present the R-extension package **arulesViz** which implements several known and novel visualization techniques to explore association rules. With examples we show how these visualization techniques can be used to analyze a data set.

Keywords: data mining, association rules, visualization.

1. Introduction

Many organizations generate 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, Imielinski, and Swami (1993) and can formally be defined as:

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n binary attributes called *items*. Let $\mathcal{D} = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the *database*. Each transaction in $\mathcal{D}$ has an 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.

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

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

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 a more detailed treatment of association rules we refer the reader to the in introduction paper for package **arules** (Hahsler, Buchta, Grün, and Hornik 2010; Hahsler, Grün, and Hornik 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, increasing minimum support is used to keep the number of association rules found at a manageable size. However, this also removes potentially interesting rules with less support. Therefore, the need to deal with large sets of association rules is unavoidable when applying association rule mining in a real setting.

Visualization is successfully used to communicate both abstract and concrete ideas in many areas like education, engineering and science (Prangsmal, van Boxtel, Kanselaar, and Kirschner 2009). According to Chen, Unwin, and Hardle (2008), the application of visualization falls into two phases. First, the exploration phase where the analysts will use graphics that are mostly incompatible for presentation purposes but make it easy to find interesting and important features of the data. The amount of interaction needed during exploration is very high and includes filtering, zooming and rearranging data. After key findings are discovered in the data, these findings must be presented in a way suitable for presentation for a larger audience. In this second phase it is important that the analyst can manipulate the presentation to clearly highlight the findings.

Many researchers introduced visualization techniques like scatter plots, matrix visualizations, graphs, mosaic plots and parallel coordinates plots to analyze association rules (see Bruzzese and Davino (2008) for a recent overview paper). This paper discusses existing techniques and demonstrates how their implementation in **arulesViz** can be used via a simple unified interface. We extend most plots using techniques of color shading and reordering to improve their interpretability. Finally, this paper also introduces a completely new method called "grouped matrix-based visualization" which is based on a novel way of clustering rules. Clustering rules is usually based on distances defined on the items included in the rules or on shared transactions covered by the rules. However, here we cluster rules especially for visualization using similarities between sets of values of a selected interest measure.

The rest of the paper is organized as follows. In Section 2 we give a very short example of how to prepare data using package **arules** and then introduce the unified interface provided

by **arulesViz** for association rule visualization. In Sections 3 to 8 we describe the different visualization techniques and give examples. Most of the techniques are enhanced using color shading and reordering. Grouped matrix-based visualization in Section 5 is a novel visualization technique. In Section 9 compares the presented visualization techniques. Section 10 concludes the paper.

2. Data preparation and unified interface of arulesViz

To use **arulesViz** we fist have to load the package. The package automatically loads other needed packages like **arules** (Hahsler *et al.* 2010) for handling and mining association rules. For the examples in this paper we load the "Groceries" data set which is included in **arules**.

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

Groceries contains sales data from a local grocery store with 9835 transactions and 169 items (product groups). The summary shows some basic statistics of the data set. For example, that the data set is rather sparse with a density just above 2.6%, that "whole milk" is the most popular item and that the average transaction contains less than 5 items.

```
> summary(Groceries)

transactions as itemMatrix in sparse format with
 9835 rows (elements/itemsets/transactions) and
 169 columns (items) and a density of 0.02609146

most frequent items:
      whole milk other vegetables       rolls/buns             soda
            2513             1903             1809             1715
          yogurt          (Other)
            1372            34055

element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55
  16   17   18   19   20   21   22   23   24   26   27   28   29   32
  46   29   14   14    9   11    4    6    1    1    1    1    3    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   4.409   6.000  32.000

includes extended item information - examples:
       labels  level2           level1
1 frankfurter sausage meet and sausage
2     sausage sausage meet and sausage
3  liver loaf sausage meet and sausage
```

Next we mine association rules using the Apriori algorithm implemented in **arules**.

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

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen
        0.5    0.1    1 none FALSE            TRUE   0.001      1     10
 target   ext
  rules FALSE

algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)        (c) 1996-2004   Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
sorting and recoding items ... [157 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 done [0.01s].
writing ... [5668 rule(s)] done [0.00s].
creating S4 object  ... done [0.01s].

> rules

set of 5668 rules
```

The result is a set of 5668 association rules. The top three rules with respect to the lift measure, a popular measure of rule strength, 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
```

However, it is clear that going through all the 5668 rules manually is not a viable option. We therefore will introduce different visualization techniques implemented in **arulesViz**. All implemented visualization techniques share the following interface:

```
> plot(x, method = NULL, measure = "support", shading = "lift", interactive = FALSE, data
```

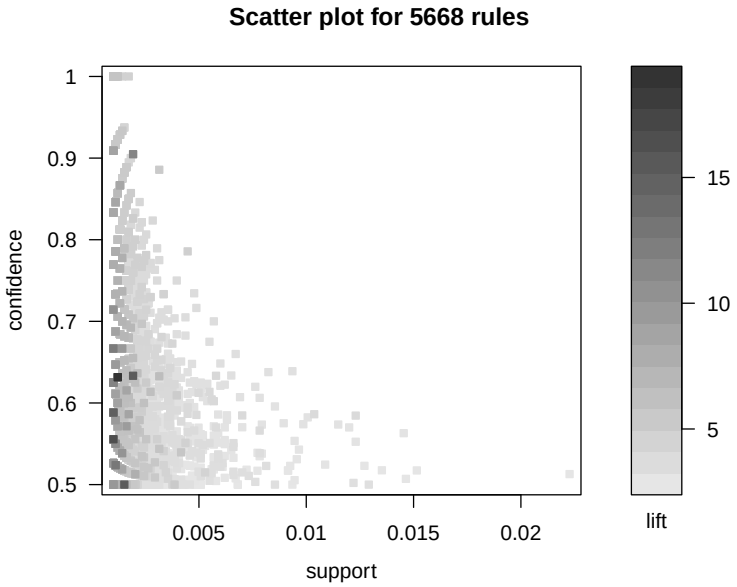

Figure 1: Default scatter plot.

where `x` is the set of rules to be visualized, `method` is the visualization method, `measure` and `shading` contain the interest measures used by the plot, `interactive` indicates if we want to interactively explore or just present the rules, `data` can contain the transaction data set used to mine the rules (only necessary for some methods) and `control` is a list with further control arguments to customize the plot.

In the following sections we will introduce the different visualization methods implemented in **arulesViz** and demonstrate how easy it is to use them.

3. Scatter plot

A straight-forward visualization of association rules is to use a scatter plot with two interest measures 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.*

The default method for `plot()` for association rules in **arulesViz** is a scatter plot using support and confidence on the axes. In addition a third measure (default: lift) is used as the color (gray level) of the points. A color key is provided to the right of the plot.

```
> plot(rules)
```

This plot for the rules mined in the previous section is shown in Figure 1. We can see that rules with high lift have typically a relatively low support. Bayardo, Jr. and Agrawal (1999) argue that the most interesting rules (sc-optimal rules) reside on the support/confidence border, which can be clearly seen in this plot. We will show later how the interactive features of this plot can be used to explore these rules.

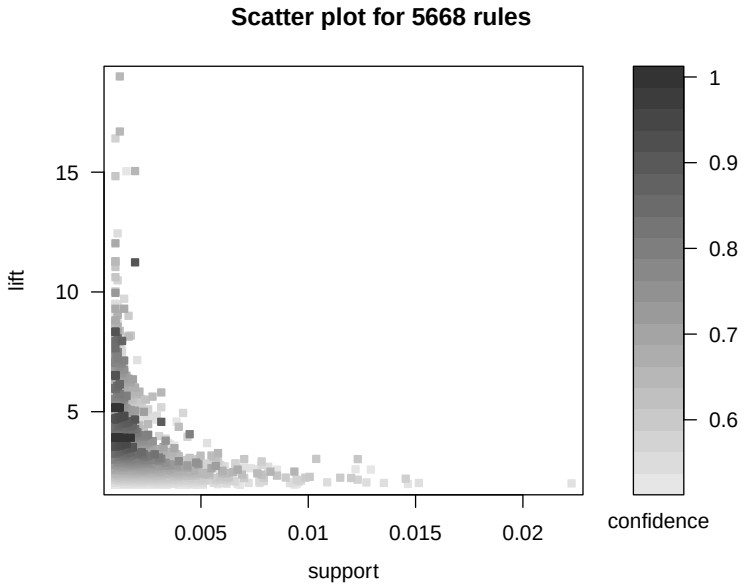

Figure 2: Scatter plot with lift on the y-axis.

Any measure stored in the quality slot of the set of rules can be used for the axes (vector of length 2 for parameter `measure`) or for color shading (`shading`). The following measures are available for our set of rules.

```
> head(quality(rules))
```

```
      support confidence     lift
1 0.001118454  0.7333333 2.870009
2 0.001220132  0.5217391 2.836542
3 0.001321810  0.5909091 2.312611
4 0.001321810  0.5652174 2.212062
5 0.001321810  0.5200000 2.035097
6 0.003660397  0.6428571 2.515917
```

These are the default measures generated by Apriori. To add other measures we refer the reader to the function `interestMeasure()` included in **arules**. For example we can customize the plot by switching lift and confidence:

```
> plot(rules, measure=c("support", "lift"), shading="confidence")
```

Figure 2 shows this plot with lift on the y-axis. Here it is easy to identify all rules with high lift.

Unwin, Hofmann, and Bernt (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. Two-key plots can be produced using the unified interface by:

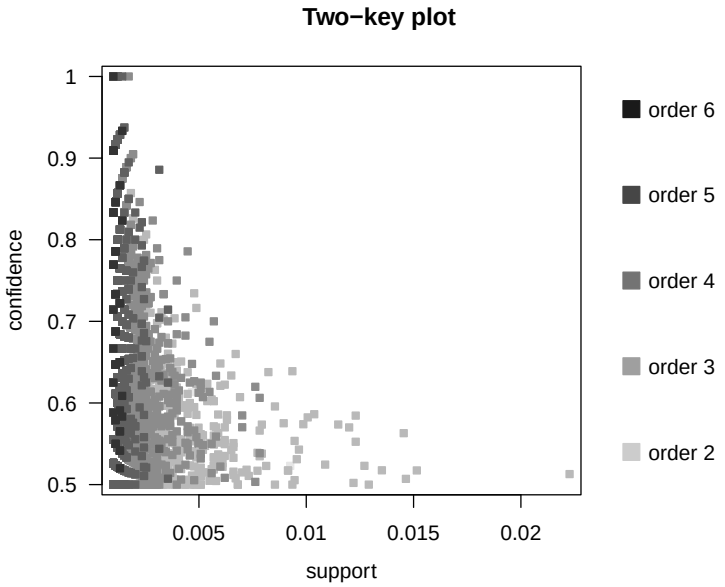

Figure 3: Two-key plot.

```
> plot(rules, shading="order", control=list(main = "Two-key plot"))
```

The resulting Two-key plot is shown in Figure 3. From the plot it is clear that order and support have a very strong inverse relationship, which is a known fact for association rules (Seno and Karypis 2005).

In addition to using order for shading, we also give the plot a different title (`main`). Other control options including `pch` (best with filled symbols: 20–25), `cex`, `xlim` and `ylim` are available and work in the usual way expected by R-users.

For exploration, the scatter plot method offers interactive features for selecting and zooming. Interaction is activated using `interactive=TRUE`.

```
> sel <- plot(rules, measure=c("support", "lift"), shading="confidence", interactive=TRUE)
```

Interactive features include:

- Inspecting individual rules by selecting them and clicking the inspect button.
- Inspecting sets of rules by selecting a rectangular region of the plot and clicking the inspect button.
- Zooming into a selected region (zoom in/zoom out buttons).
- Filtering rules using the measure used for shading by clicking the filter button and selecting a cut-off point in the color key. All rules with a measure lower than the cut-off point will be filtered.
- Returning the last selection for further analysis (end button).

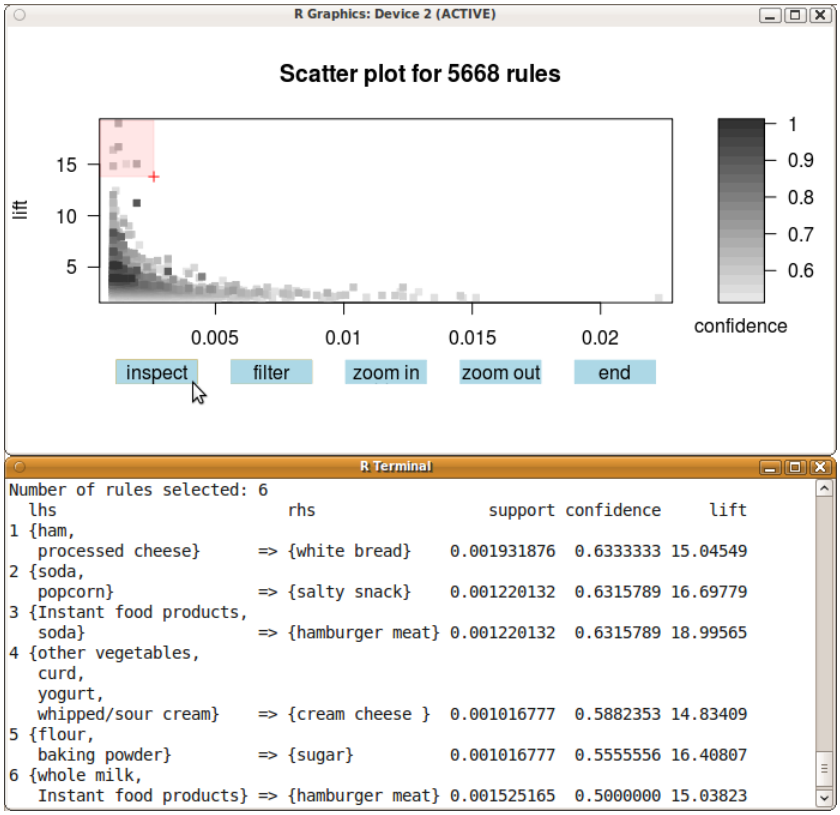

Figure 4: Interactive mode for scatter plot (inspecting rules with high lift).

The result of an example interaction is shown in Figure 4. Using a box selection the rules with the highest lift are selected. Using the inspect button, the rules are displayed in the terminal below the plotting device.

4. Matrix-based visualizations

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 a 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

$$\mathcal{R} = \{\langle a_1, c_1, m_1\rangle, \ldots \langle a_i, c_i, m_i\rangle, \ldots \langle a_n, c_n, m_n\rangle\}$$

where a_i is the antecedent, c_i is the consequent and m_i is the selected interest measure for the i-th rule for $i = 1, \ldots, n$. In $\mathcal{R}$ we identify the set of K unique antecedents and L unique consequent. We create a $L \times K$ matrix $\mathbf{M}$ with one column for each unique antecedent and one row for each unique consequent. Finally, we populate the matrix by setting $\mathbf{M}_{lk} = m_i$ for $i = 1, \ldots, n$ and l and k corresponding to the position of a_i and c_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, leong Ong, Ng, and Lim (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, Whitney, and Thomas 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 reduce the number of rules here by filtering out all rules with a low confidence score.

```
> subrules <- rules[quality(rules)$confidence > 0.8]
> subrules

set of 371 rules

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

The resulting plot is shown in Figure 5. Since there is not much space for long labels in the plot, we only show numbers as labels for rows and columns and the complete itemsets are printed to the terminal for look-up. We omit the complete output here, since this plot and the next few plots print several hundred labels to the screen. The output looks like:

```
Itemsets in Antecedent (lhs)
  [1] "{liquor,red/blush wine}"
```

```
  [2] "{curd,cereals}"
  [3] "{yogurt,cereals}"
  [4] "{butter,jam}"
  [5] "{soups,bottled beer}"
    (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}"
```

The visual impression can be improved by reordering rows and columns in the matrix such that rules with similar values of the interest measure are presented closer together. This removes some of the fragmentation in the matrix display and therefore makes it easier to see structure.

```
> plot(subrules, method="matrix", measure="lift", control=list(reorder=TRUE))
```

In the resulting plot in Figure 6 we see the emergence of two large blocks of rules with two different consequents and then smaller blocks for the rest. With the interactive mode (`interactive=TRUE`) colored squares can be selected and the rule it represents will be displayed on the screen. This can be used to explore different antecedents which have a similar impact on the same consequent in terms of the measure used in the plot.

Reordering is done using the **seriation** package (Hahsler, Hornik, and Buchta 2008) which provides a wide array of ordering methods for multivariate data. Typically, finding the optimal order subject to some defined objective function is a difficult combinatorial optimization problem which involves for n objects to check many of the $O(n!)$ possible permutations. However, for visualization purposes some suboptimal but fast heuristics are acceptable and it is often sufficient to try several methods to find the most useful representation. However, it is useful to understand the different methods and the objective function they try to optimize. Some available useful methods are:

- "PCA" – First principal component. Uses the first principal component for the data matrix and for the transposed matrix as order for rows and columns. This is a very fast approach which avoids the expensive distance matrix computation.
- "TSP" – Traveling salesperson problem solver. Computes distance matrices between rows and between columns and solves two separate TSPs. By default the nearest insertion heuristic is used. This method is reasonably fast for small rule sets, but the distance matrix computation and the associated memory requirements make it impractical for larger sets.
- "HC" – Hierarchical clustering. Computes distance matrices for rows and columns and then clusters twice. Distance matrix computation is again limiting the approach for smaller sets.
- "max", "avg" and "median" – Reorder rows/columns by their maximum, average or median value. This extremely simple and fast methods provides sometimes good visualizations.

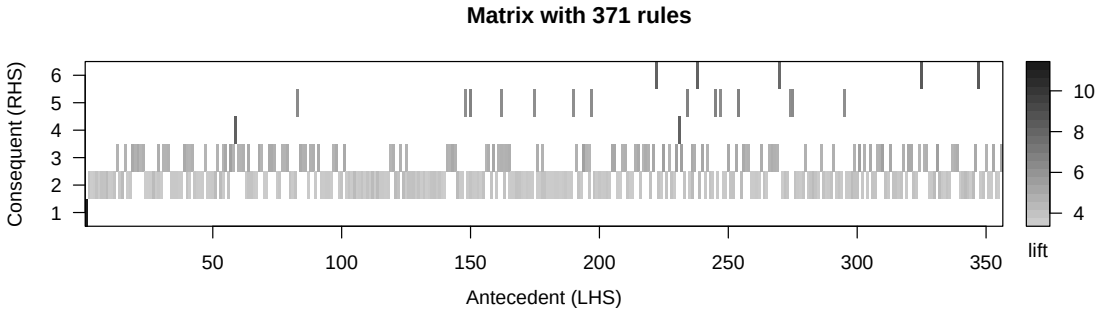

Figure 5: Matrix-based visualization with colored squares.

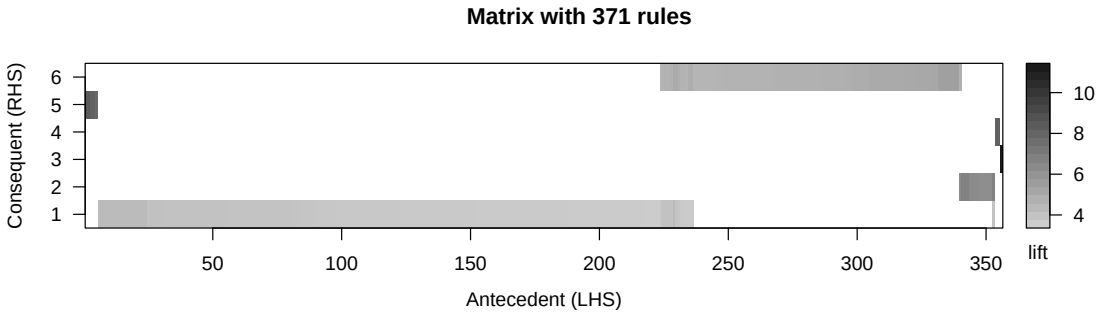

Figure 6: Matrix-based visualization with colored squares (reordered).

Other available methods include "BEA", "MDS", "OLO", "GW", "ARSA" and "BBURCG". We refer the interested reader to Hahsler *et al.* (2008) for detailed description of these methods. Different seriation methods can be selected via `reorderMethod` in the control list (default: TSP). For the methods which first compute dissimilarity matrices, the control list argument `reorderDist` can be used to specify the used measure (default: Euclidean distance).

Most seriation methods cannot handle missing values. However, the visualized matrix typically contains many missing values since minimum support discards of many rules during rule mining. We use a very crude imputation approach by replacing the missing values in the matrix (values for rules not contained in the rules set) by a fixed value (0 by default).

An alternative representation is to use 3D bars (method "matrix3D") instead of colored rectangles.

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

```
> plot(subrules, method="matrix3D", measure="lift", control=list(reorder=TRUE))
```

The 3D visualization is shown in Figures 7 and 8.

If we specify a vector with two measures, both measures are used simultaneously using color hue for one measure and luminance and chroma together for the other.

```
> plot(subrules, method="matrix", measure=c("lift", "confidence"))
```

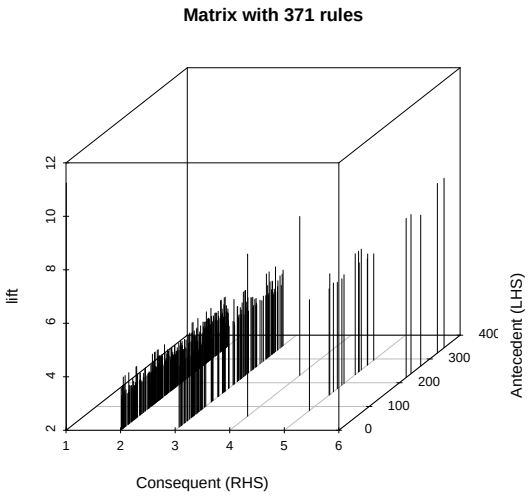

Figure 7: Matrix-based visualization with 3D bars.

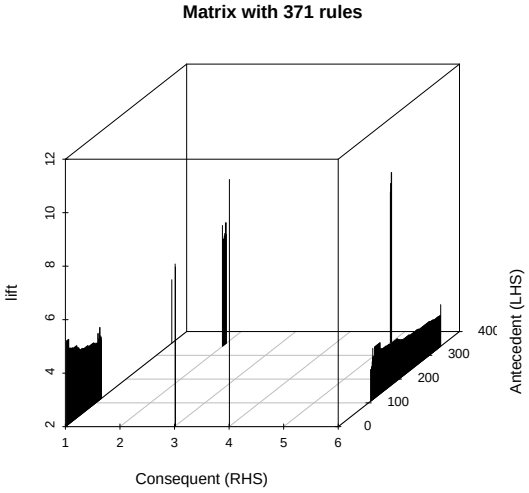

Figure 8: Matrix-based visualization with 3D bars (reordered).

```
> plot(subrules, method="matrix", measure=c("lift", "confidence"),
+         control=list(reorder=TRUE))
```

Figures 9 and 10 show plots with two measures of interest. The legend is here a color matrix. By matching a square with the closed color in the legend, we can determine both, support and confidence. High confidence/high support rules can be identified in the plot as hot/red (high confidence) and dark/intense (high support). Note that on a black and white printout, the different colors are not distinguishable.

5. 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 (Hahsler and Chelluboina 2011) that enhances matrix-based visualization using grouping of rules via clustering to handle a larger number of rules. Grouped rules are presented as an aggregate in the matrix and can be explored interactively by zooming into and out of groups.

A direct approach to cluster itemsets is to define a distance metric between two itemsets X_i and X_j. A good choice is the Jaccard distance defined as

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

The distance simply is the number of items that X_i and X_j have in common divided by the number of unique items in both sets. For a set of m rules we can calculate the $m(m-1)/2$ distances and use them as the input for clustering. However, using clustering on the itemsets

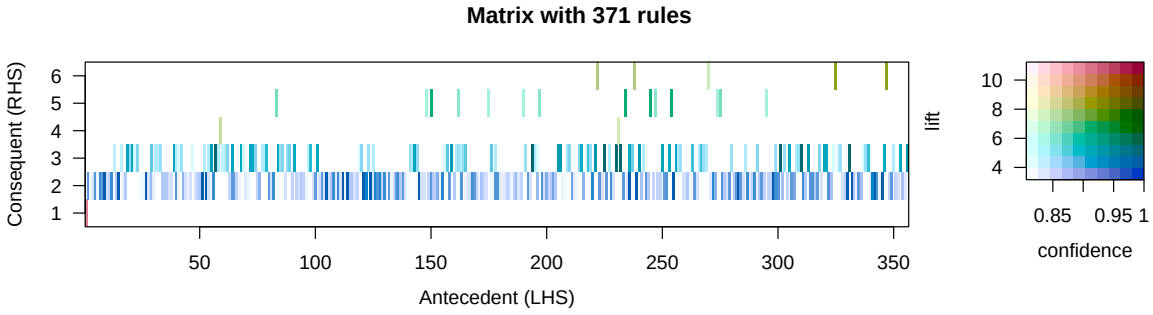

Figure 9: Matrix-based visualization of two measures with colored squares.

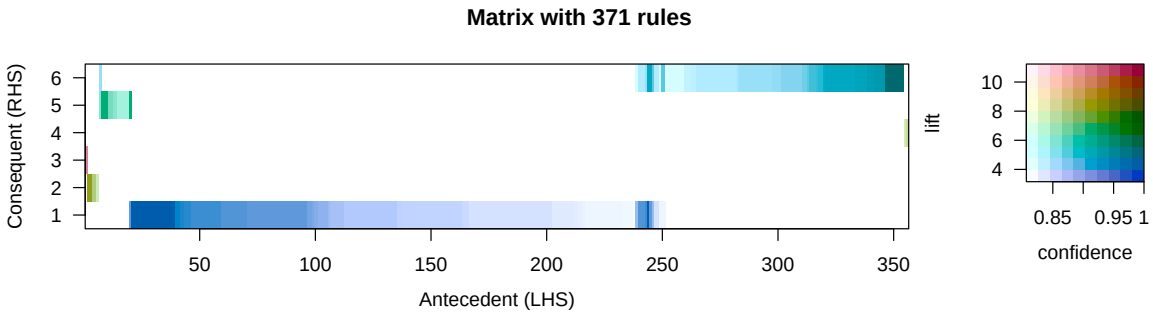

Figure 10: Matrix-based visualization of two measures with colored squares (reordered).

directly has several problems. First of all, data sets typically mined for association rules are high-dimensional, i.e., contain many different items. This high dimensionality is carried 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 since minimum support used in association rule mining leads in addition to relatively short rules resulting in extremely sparse data.

Several approaches to cluster association rules and itemsets to address the dimensionality and sparseness problem were proposed in the literature. Toivonen, Klemettinen, Ronkainen, Hatonen, and Mannila (1995), Gupta, Strehl, and Ghosh (1999) and Berrado and Runger (2007) propose clustering association rules by looking at the number of transactions which are covered by the rules. 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 of frequent itemsets, two subsets of a frequent itemset will cover many common transactions. This bias will lead to mostly 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 4 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 antecedents becomes the problem of grouping columns in $\mathbf{M}$. To group the column vectors fast and efficient into k groups we use k-means clustering. The default interest measure used is lift. The idea is that antecedents that are statistically dependent on the same consequents are similar and thus can 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 similar dependence to the same consequents.

The same grouping method can be used for consequents. However, since the mined rules are restricted to a single item in the consequent there is no problem with combinatorial explosion and such a grouping is typically not necessary.

To visualize the grouped matrix we use a balloon plot with antecedent groups as columns and consequents as rows (see Figure 11). The color of the balloons represent the aggregated interest measure in the group with a certain consequent and the size of the balloon shows the aggregated support. The default aggregation function is the median value in the group. The number of antecedents and the most important (frequent) items in the group are displayed as the labels for the columns. 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, placing the most interesting group in the top left corner.

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

```
> plot(rules, method="grouped")
```

The resulting visualization is shown in Figure 11. The group of 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."

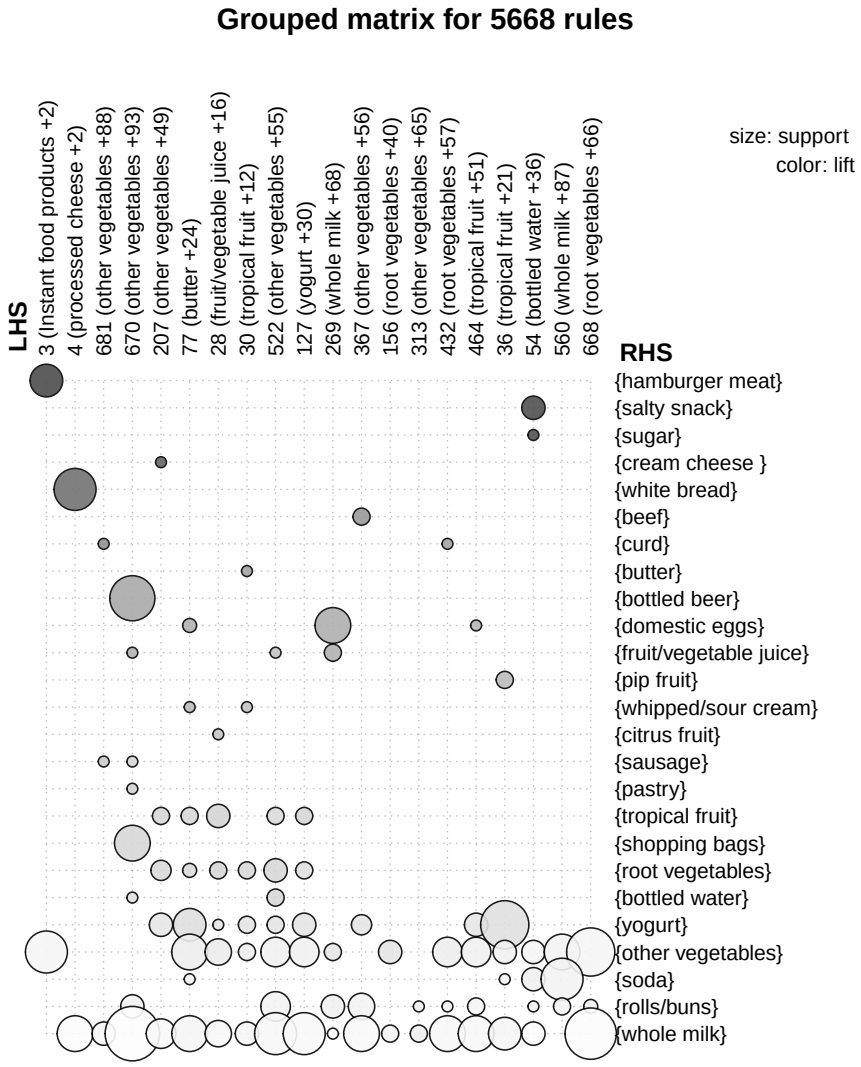

Figure 11: Grouped matrix-based visualization.

To increase the number of groups we can change k which defaults to 20.

```
> plot(rules, method="grouped", control=list(k=50))
```

The resulting, more detailed plot is shown in Figure 12.

An interactive version of the grouped matrix visualization is also available.

```
> sel <- plot(rules, method="grouped", interactive=TRUE)
```

Here it is possible to zoom into groups and to inspect the rules contained in a selected group.

6. Graph-based visualizations

Graph-based techniques (Klemettinen, Mannila, Ronkainen, Toivonen, and Verkamo 1994; Rainsford and Roddick 2000; Buono and Costabile 2005; Ertek and Demiriz 2006) visualize association rules using vertices and edges where vertices typically represent items or itemsets and edges indicate relationship 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. For the following plots we select the 10 rules with the highest lift.

```
> subrules2 <- head(sort(rules, by="lift"), 10)
```

arulesViz contains several graph-based visualizations rendered using either the *igraph* library via package **igraph** (Csardi and Nepusz 2006; **?**) or the interface to the *GraphViz* software in package **Rgraphviz** (Gentry, Long, Gentleman, Falcon, Hahne, Sarkar, and Hansen 2010). By default igraph is used. The following plot represents itemsets as vertices and rules as directed edges between itemsets (shown in Figure 13).

```
> plot(subrules2, method="graph")
```

Another variant uses items and rules as two types of vertices and edges indicate which items occur in which rule.

```
> plot(subrules2, method="graph",
+         control=list(type="items"))
```

Figure 14 shows the resulting graph. This representation focuses on how the rules are composed of individual items and shows which rules share items.

An interactive visualization is available in **arulesViz**, however, the built-in graph based visualizations are only useful for small set of rules. To explore large sets of rules with graphs, advanced interactive features like zooming, filtering, grouping and coloring nodes are needed. Such features are available in interactive visualization and exploration platforms for networks and graphs like *Gephi* (Bastian, Heymann, and Jacomy 2009). From **arulesViz** graphs for sets of association rules can be exported in the GraphML format or as a Graphviz dot-file to be explored in tools like Gephi. For example the 1000 rules with the highest lift are exported by:

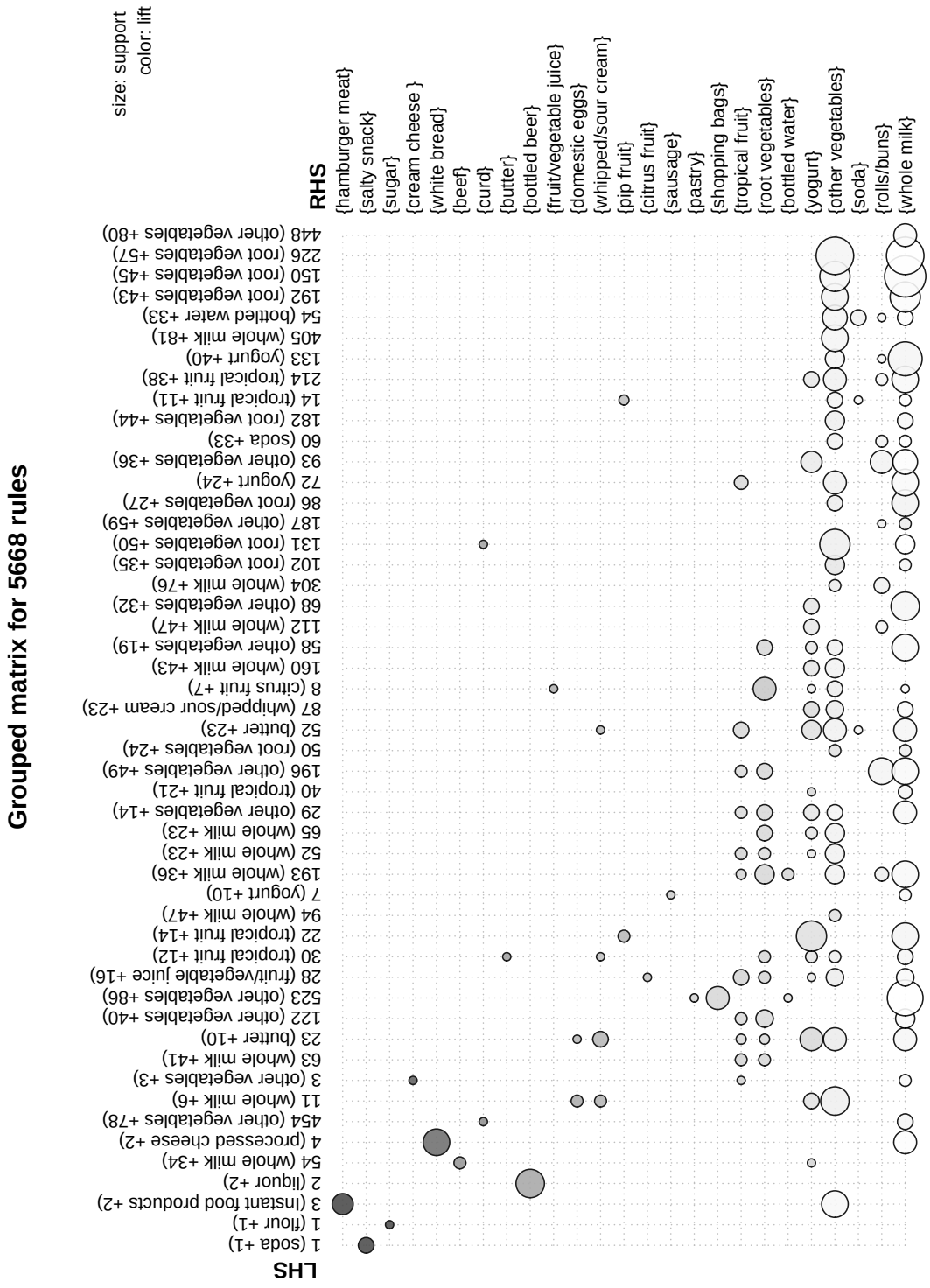

Figure 12: Grouped matrix with $k = 50$.

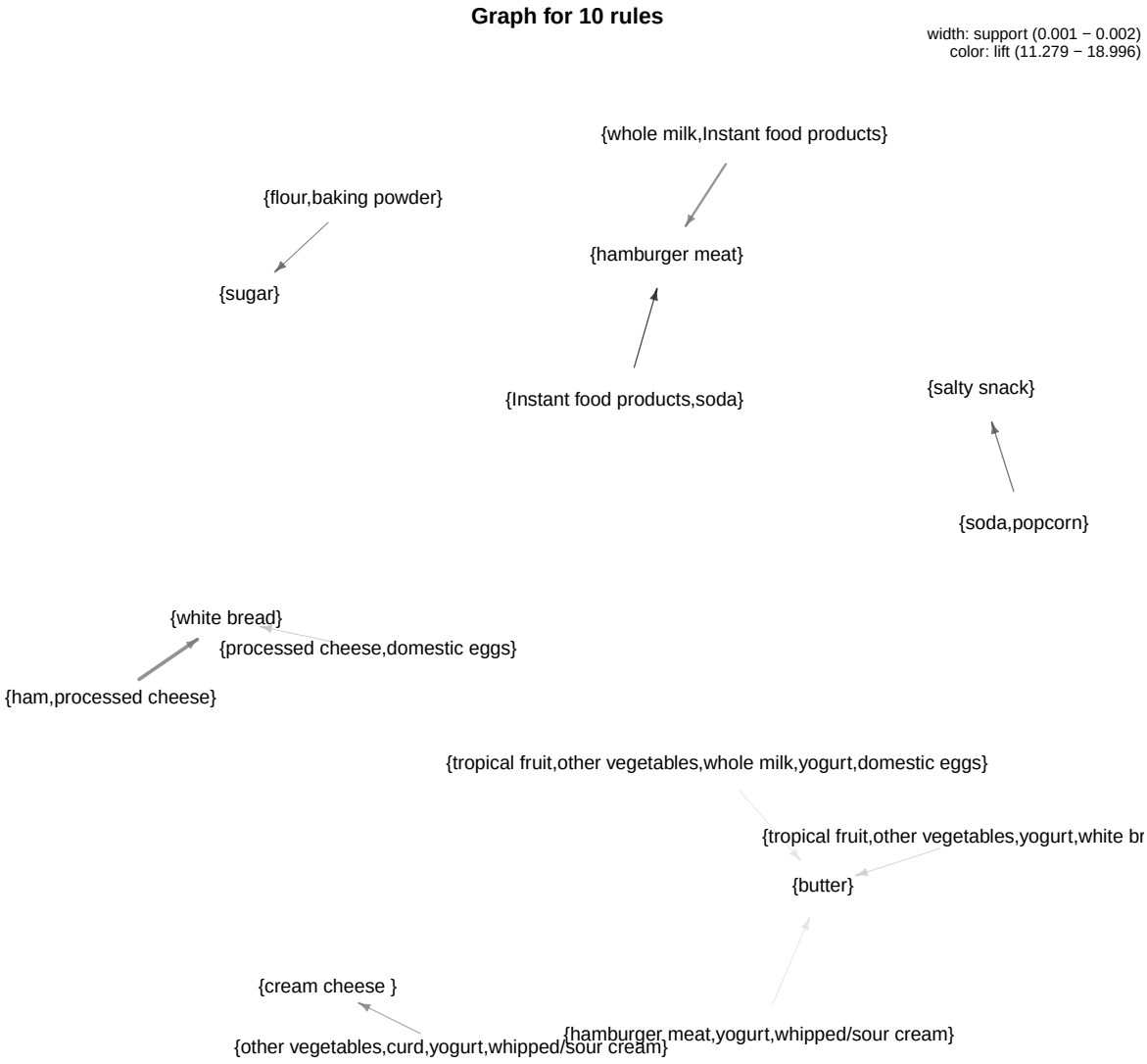

Figure 13: Graph-based visualization with itemsets as vertices.

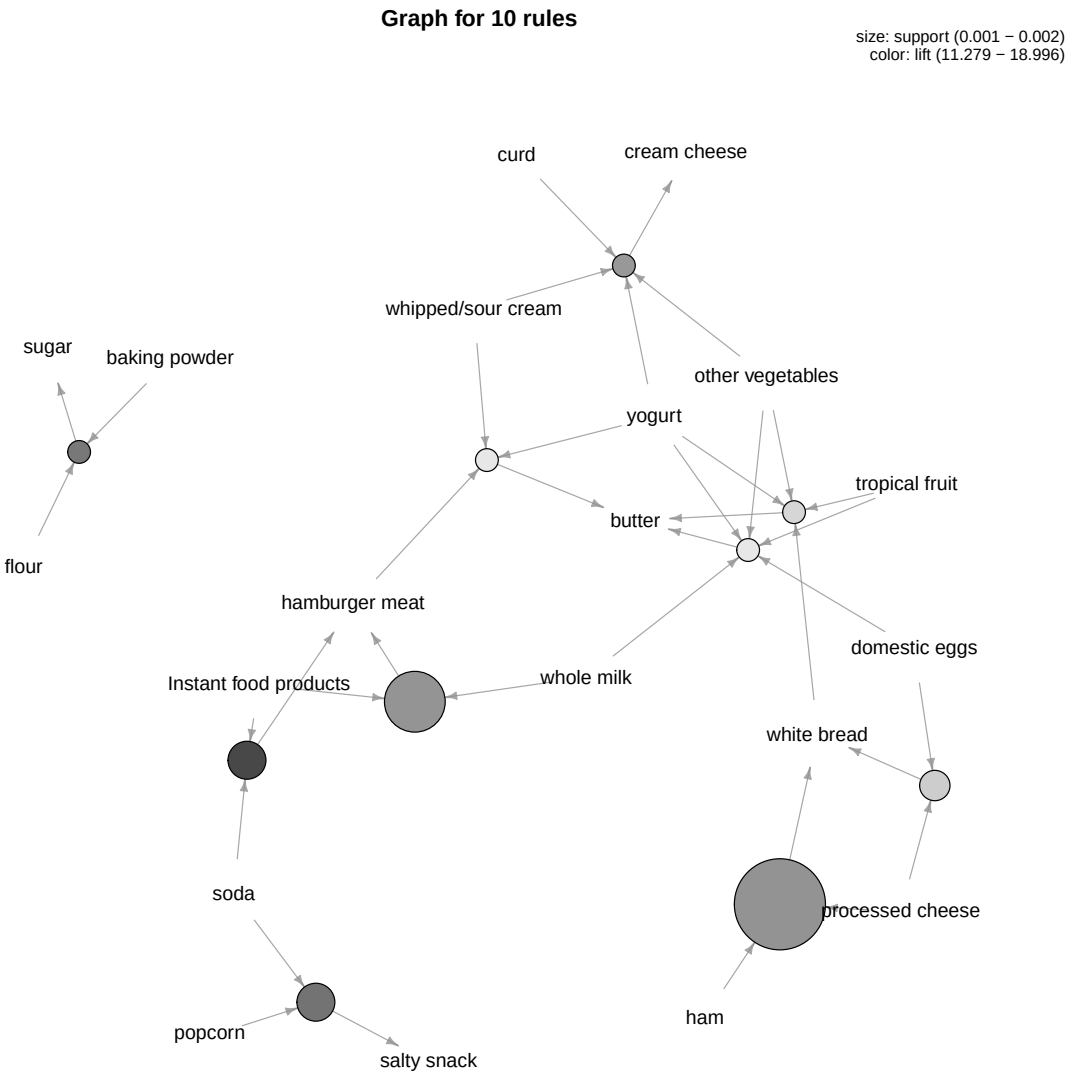

Figure 14: Graph-based visualization with items and rules as vertices.

```
> saveAsGraph(head(sort(rules, by="lift"),1000), file="rules.graphml")
> #saveGraphML(head(sort(rules, by="lift"),1000), file="rules.graphml")
```

Figure 15 shows a screenshot of exploring these rules interactively. Rules can be explored by zooming, filtering and coloring vertices and edges.

7. Parallel coordinates plot

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, An, and Cercone 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.

```
> plot(subrules2, method="paracoord")
```

Figure 16 shows a parallel coordinates plot for 10 rules. The width of the arrows represents support and the intensity of the color represent confidence. It is obvious that for larger rule sets visual analysis becomes difficult since with an increasing number of rules also the number of crossovers between the lines increases Yang (2003).

The number of crossovers can be significantly reduced by reordering the items on the y-axis. Reordering the items to minimize the number of crossovers is a combinatorial problem with $n!$ possible permutations. However, for visualization purposes a suboptimal but fast solution is typically acceptable. We applies a variation of the well known 2-opt heuristic Bentley (1990) for travelers salesman problem to the reordering problem. The objective function is to minimize the number of crossovers. The simple heuristic uses the following steps:

1. Choose randomly two items and exchange them if it improves the objective function.
2. Repeat step 1 till no improvement is found for a predefined number of tries.

Reordering is achieved with `reorder=TRUE`.

```
> plot(subrules2, method="paracoord", control=list(reorder=TRUE))
```

Figure 17 shows the parallel coordinates plot with reordered items to reduce crossovers.

8. Double Decker plots

A double decker plot is a variant of a mosaic plot. A mosaic plot displays a contingency table using tiles on a rectangle created by recursive vertical and horizontal splits. The size of each tile is proportional to the value in the contingency table. Double decker plots use only a single horizontal split.

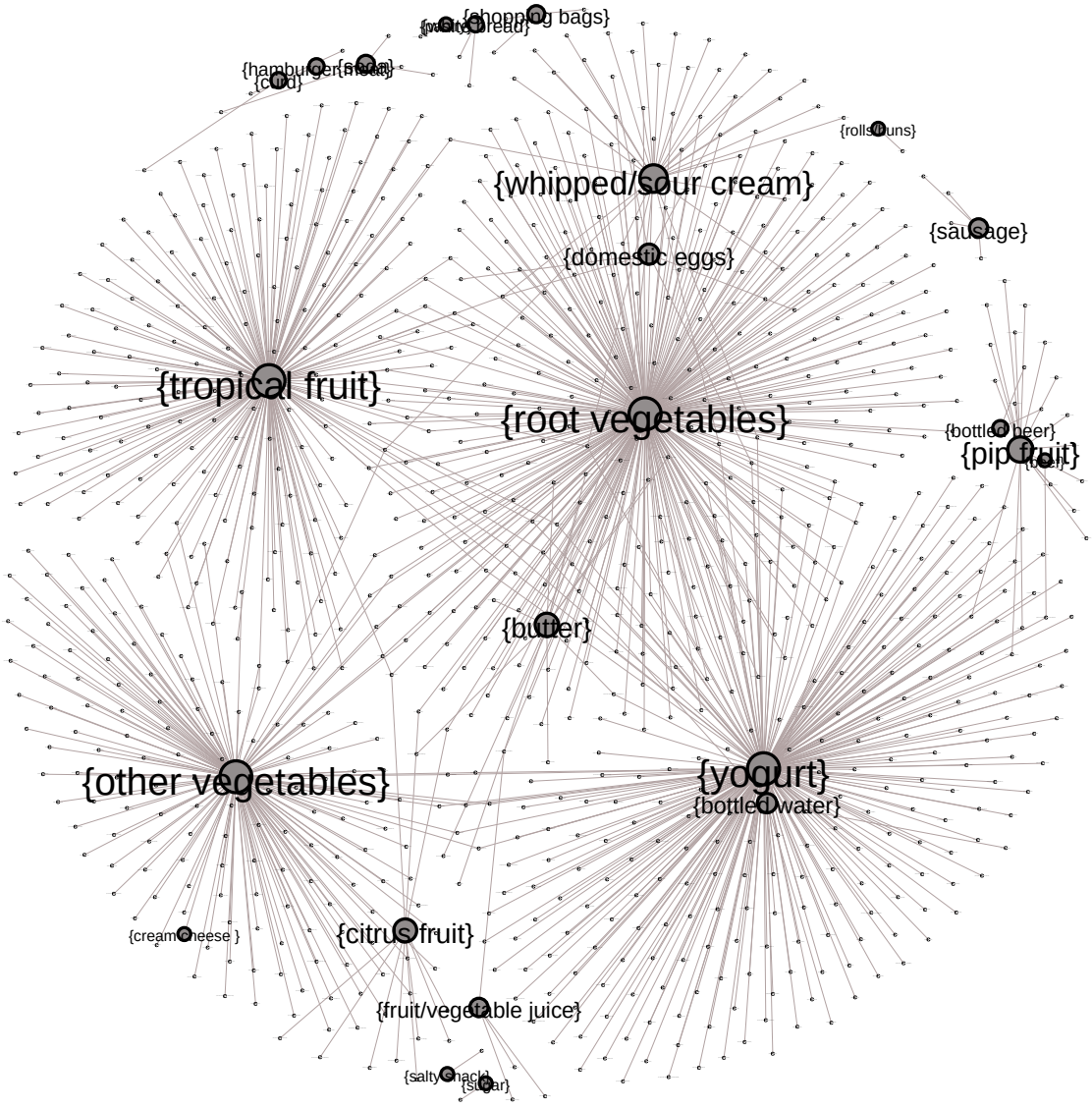

Figure 15: Visualization of 1000 rules with Gephi (Fruchterman Reingold layout, vertex and label size is proportional to the in-degree, i.e., the number of rules the consequent participates in).

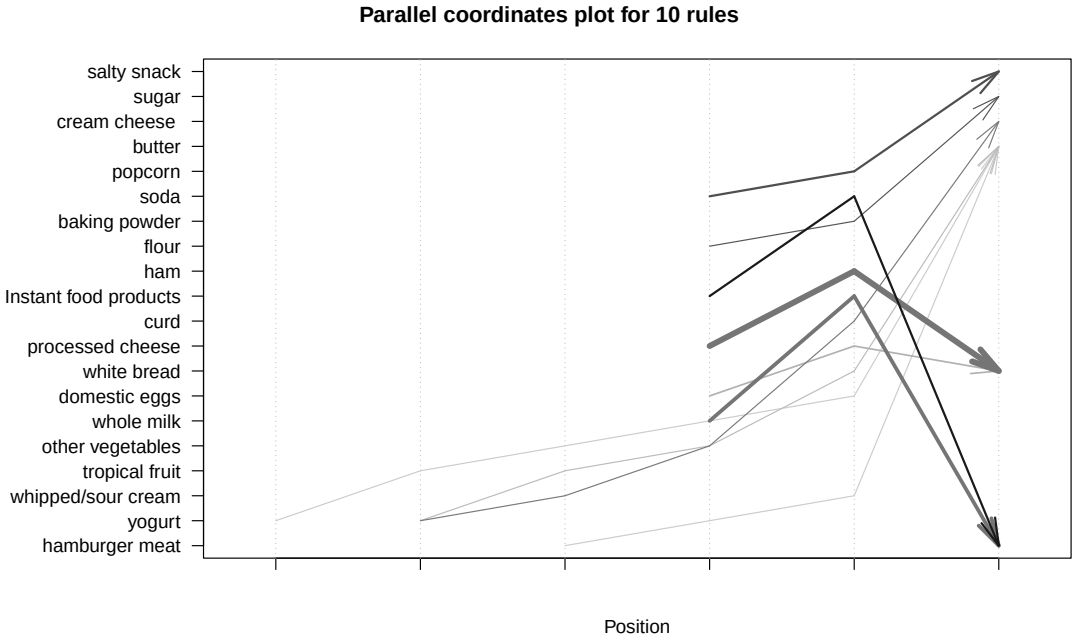

Figure 16: Parallel coordinate plot.

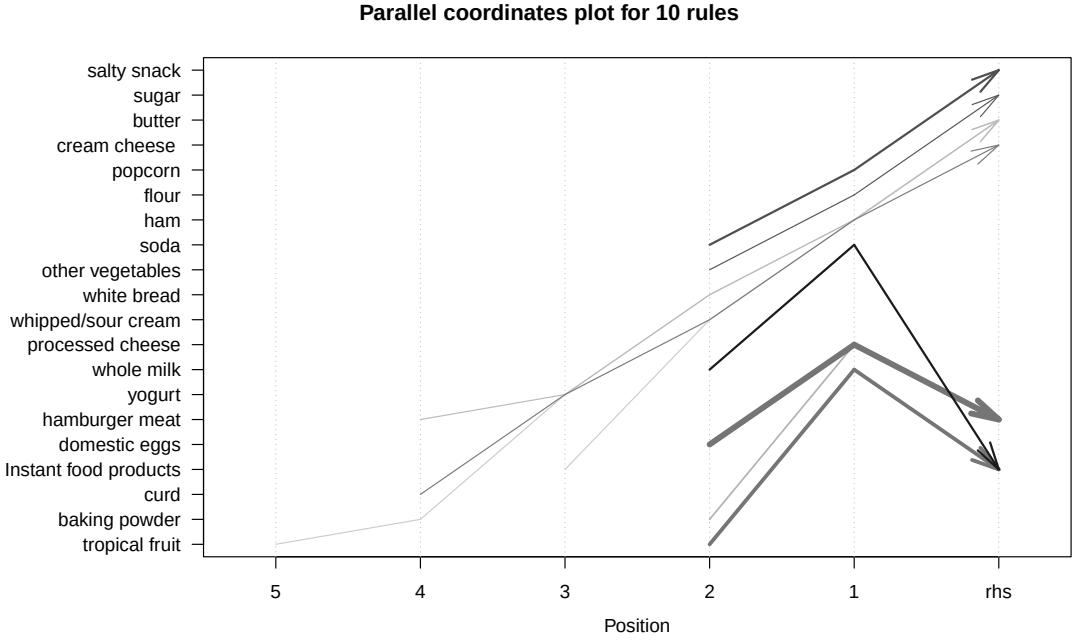

Figure 17: Parallel coordinate plot (reordered).

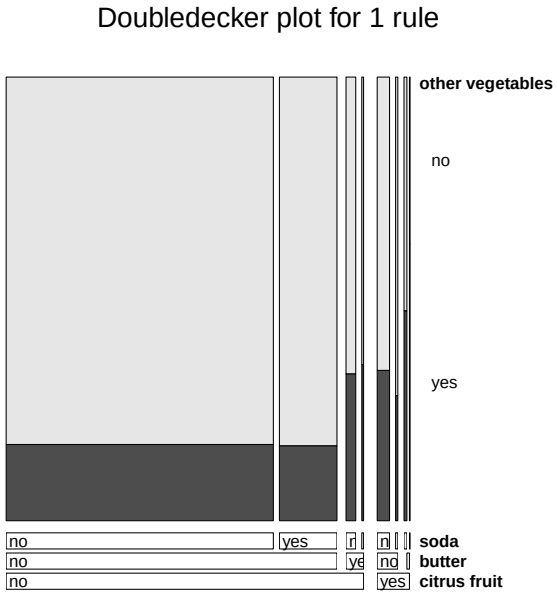

Figure 18: Double decker plot for a the rule citrus fruit,butter,soda => other vegetables.

Hofmann, Siebes, and Wilhelm (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 from the original data set. The items in the antecedent are used for the vertical splits and the consequent item is used for horizontal highlighting.

We randomly choose a single rule and visualize it with a double decker plot.

```
> oneRule <- sample(rules, 1)
> inspect(oneRule)

  lhs                rhs                    support confidence     lift
1 {citrus fruit,
   butter,
   soda}         => {other vegetables} 0.001016777      0.625 3.230097

> plot(oneRule, method="doubledecker", data = Groceries)
```

Figure 18 shows the resulting plot. The area of blocks gives the support and the height of the "yes" blocks is proportional to the confidence for the rules consisting of the antecedent items marked as "yes." Items that show a significant jump in confidence when changed from "no" to "yes" are interesting. This is captured by the interest measure *difference of confidence* defined by Hofmann and Wilhelm (2001).

9. Comparison of techniques

In this section, we compare the visualization techniques available in **arulesViz** based on the size of the rule set which can be analyzed, the number of interest measures which are shown

Technique	Method	Rule set	Measures	Interactive	Reordering	Ease of use
Scatterplot	`"scatterplot"`	large	3	✓		++
Two-Key plot	`"scatterplot"`	large	2 + order	✓		++
Matrix-based	`"matrix"`	medium	1		✓	0
Matrix-b. (2 measures)	`"matrix"`	medium	2		✓	– –
Matrix-b. (3D bar)	`"matrix3D"`	small	1		✓	+
Grouped matrix	`"grouped"`	large	2	✓	✓	0
Graph-based	`"graph"`	small	2			++
Graph-b. (external)	`"graph"`	large	2	✓	✓	+
Parallel coordinates	`"paracoord"`	small	1		✓	–
Double decker	`"doubledecker"`	single rule	(2)			–

Table 1: Comparison of visualization methods for association rules available in **arulesViz**.

simultaneously, if the technique offers interaction and reordering and how intuitive each visualization technique is. Note that most of these categories are only evaluated qualitatively here, and the results presented in Table 1 are only meant to guide the user towards the most suitable techniques for a given application.

Scatterplot (including two-key plots) and grouped matrix plot are capable to analyze large rule sets. These techniques are interactive to allow the analyst to zoom and select interesting rules. Matrix-based can accommodate rule sets of medium size. Reordering can be used to improve the presentation. To analyze small rule sets the matrix-based method with 3D bars, graph-based methods and parallel coordinates plots are suitable. Graphs for large rule sets can be analyzed using external tools like Gephi. Finally, double decker plots only visualize a single rule.

The techniques discussed in this paper can also be categorized based on the number of interest measures simultaneously visualized. Most methods can represent two measures and scatter plots are even able to visualize three measures for each rule in one plot.

Scatter plot and graph based techniques are the most intuitive while matrix-based visualization with two interest measures, parallel coordinates and double decker require time to learn how to interpret them correctly.

10. Conclusion

Association rule mining algorithms typically generate a large number of association rules which poses a major problem for understanding and analyzing rules. In this paper we presented several visualization techniques implemented in **arulesViz** which can be used to explore and present sets of association rules. In addition we present a new interactive visualization method called grouped matrix-based visualization which can used to effectively explore large rule sets.

Future development will focus on enhancing the visualizing techniques with advanced interactive features using for example **iplots** (Urbanek and Theus 2003) which supports in addition to selection and zooming also brushing and linking between different plots.

References

Agrawal R, Imielinski T, 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," pp. 207–216. ACM Press. URL http://doi.acm.org/10.1145/170035.170072.

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

Bayardo, Jr RJ, 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," pp. 145–154. ACM.

Bentley JL (1990). "Experiments on traveling salesman heuristics." In "SODA '90: Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms," pp. 91–99. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA. ISBN 0-89871-251-3.

Berrado A, Runger GC (2007). "Using metarules to organize and group discovered association rules." *Data Mining and Knowledge Discovery*, **14**(3), 409–431.

Brin S, Motwani R, Ullman JD, 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," pp. 255–264. Tucson, Arizona, USA.

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

Buono P, Costabile MF (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.

Chen CH, Unwin A, Hardle W (eds.) (2008). *Handbook of Data Visualization*. Springer Handbooks of Computational Statistics. Springer-Verlag.

Csardi G, Nepusz T (2006). "The igraph software package for complex network research." *InterJournal*, **Complex Systems**, 1695. URL http://igraph.sf.net.

Ertek G, Demiriz A (2006). "A Framework for Visualizing Association Mining Results." In "ISCIS," pp. 593–602. URL http://dx.doi.org/10.1007/11902140_63.

Gentry J, Long L, Gentleman R, Falcon S, Hahne F, Sarkar D, Hansen K (2010). *Rgraphviz: Provides plotting capabilities for R graph objects*. R package version 1.24.0.

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

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

Hahsler M, Chelluboina S (2011). "Visualizing Association Rules in Hierarchical Groups." In "42nd Symposium on the Interface: Statistical, Machine Learning, and Visualization Algorithms (Interface 2011)," The Interface Foundation of North America.

Hahsler M, Grün B, Hornik K (2005). "arules – A Computational Environment for Mining Association Rules and Frequent Item Sets." *Journal of Statistical Software*, **14**(15), 1–25.

Hahsler M, Hornik K, Buchta C (2008). "Getting Things in Order: An Introduction to the R Package seriation." *Journal of Statistical Software JSS*, **25**.

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

Hofmann H, Siebes A, Wilhelm AFX (2000). "Visualizing Association Rules with Interactive Mosaic Plots." In "KDD," pp. 227–235. URL http://doi.acm.org/10.1145/347090.347133.

Hofmann H, Wilhelm A (2001). "Visual comparison of association rules." *Computational Statistics*, **16**(3).

Klemettinen M, Mannila H, Ronkainen P, Toivonen H, Verkamo AI (1994). "Finding Interesting Rules from Large Sets of Discovered Association Rules." In "CIKM," pp. 401–407. URL http://doi.acm.org/10.1145/191246.191314.

Ong KH, leong Ong K, Ng WK, Lim EP (2002). "CrystalClear: Active Visualization of Association Rules." In "In ICDM'02 International Workshop on Active Mining AM2002," .

Prangsmal ME, van Boxtel CAM, Kanselaar G, Kirschner PA (2009). "Concrete and abstract visualizations in history learning tasks." *British Journal of Educational Psychology*, **79**, 371–387.

Rainsford CP, Roddick JF (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," pp. 91–96. Springer-Verlag.

Seno M, Karypis G (2005). "Finding Frequent Itemsets Using Length-Decreasing Support Constraint." *Data Mining and Knowledge Discovery*, **10**, 197–228.

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

Unwin A, Hofmann H, 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," pp. 472–483. Springer-Verlag.

Urbanek S, Theus M (2003). "iPlots: High Interaction Graphics for R." In "Proceedings of the 3rd International Workshop on Distributed Statistical Computing," .

Wong PC, Whitney P, Thomas J (1999). "Visualizing Association Rules for Text Mining." In "INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization," p. 120. IEEE Computer Society, Washington, DC, USA. ISBN 0-7695-0431-0.

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.

Affiliation:

Michael Hahsler
Computer Science and Engineering
Lyle School of Engineering
Southern Methodist University
P.O. Box 750122
Dallas, TX 75275-0122
E-mail: mhahsler@lyle.smu.edu
URL: http://lyle.smu.edu/~mhahsler