Decision Analytics Seminar in R

– SEMINAR SUMMER SEMESTER 2014 –

Introduction to Association Rule Learning with the Statistical Software *R*

– SEMINAR PAPER –

Submitted by:
Nicolas Banholzer

Advisor:
Prof. Dr. Dirk Neumann

Contents

Abstract

Association rule learning is a machine learning technique that aims at finding associations between between attributes in large transaction data. After a brief explanation of the theory, this seminar paper will give an introduction to association rule mining with the statistical software *R*. Using the packages **arules** and **arulesViz**, the paper will go through a typical mining process and illustrate it at an appropriate dataset.

1 Introduction

The method of association rule learning gained popularity with the story of beer and diapers. It is said that a database query of a retail consulting group in 1992 showed that presumably male customers regularly purchased beer and diapers between five and seven pm (Power 2002).
Today, association rule learning is a highly valuable and widely used tool by retailers to turn their large amount of data into real profits. Every time a customer makes an transaction, his *transaction id* and his purchased *items* are stored in a *database* of the form $I \times T$, where $I = \{i_1, i_2, i_3, ..., i_n\}$ is the set of items and $T = \{t_1, t_2, t_3, ..., t_n\}$ is the set of transactions *(take Table 1 as an example)*. A predictive analyst then aims at finding a link between two items or a group of items, the so called *rule*. For example, the rule $beer \Rightarrow diapers$ means that customers that buy beer usually also buy diapers. The item or itemset to the left is thereby called the left-hand-side *(LHS)* or *antecedent* and the one to the right is called the right-hand-side *(RHS)* or *consequent* of the rule.
In a widely cited article, Agrawal and Swami (1993) first introduced association rule learning for market basket analysis. But since the conept can be used for any categorical dataset, association rule learning found its way into many other application areas such as Web usage mining or bioinformatics. The contribution of this work is to give an introduction to association rule mining with the statistical software *R*.

The remainder of this paper is organized as follows. Section 2 describes the theoretical framework of association rule learning. Using the *Income* dataset, Section 3 gives an introduction to the mining process in *R*, followed by an overview of visualization techniques in Section 4. Finally, Section 5 conludes with a short summary of the paper.

2 Theoretical Framework

This chapter explains the basic underlying theory of association rule learning. To quantify the quality of a rule, Section 2.1 introduces some important and frequently used interest measures, which will also be applied in Section 3 and Section 4. To make it more comprehensive, Table 1 serves as a simple example to explain the interest measures as well as the mining algorithm in Section 2.2.

t_i	Candy	Cheese	Fruits	Juices
t_1	1	1	0	0
t_2	0	1	1	1
t_3	0	0	1	1
t_4	0	0	1	1
t_5	0	1	1	1

Table 1: A small exemplary transaction dataset with four items and five transactions.

2.1 Interest Measures

To find interesting rules, we need some criteria according to which we sort out every rule that does not satisfy them. Agrawal and Swami (1993) introduced *support* and *confidence*, two interest measures, that mark the basis of most mining processes today. Over time, more measures have been developed to refine the mining process. In this subsection we refer to the measures *support, confidence* and *lift* and illustrate them at a small example. A short and comprehensive comparison of commonly used interest measures is provided by Hahsler (2015).

Support marks the starting part of most analyses. It gives the proportion of transactions that contain an item X. In the exemplary transaction dataset presented in Table 1 , the item *candy* is part in one out of five transactions. Hence, $supp(Candy) = \frac{1}{5} = 0.2$. Confidence is another frequently used interest measure, defined as

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}. \quad (1)$$

It relates the share of transactions that contain both X and Y to the share of transactions that do only contain item X. In the example, *juices* is part of every transaction that has also *fruits* in it. Thus, $conf(Fruits \Rightarrow Juices) = 1$. It means, that every time someone is buying *fruits*, he also buys *juices*. At this point, a retailer might for example come up with the idea to place the two items far away from each other in the supermarket, such that customers may grab some additional items on their way.

In a sufficient large dataset, some rules may simply occur by chance. For that reason, Brin et al. (1997) introduced lift, which is defined as

$$lift(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X) \cdot supp(Y)}. \quad (2)$$

It is an interesting measure because it compares the itemsets actual support ($\widehat{=}$ nominator) to what we would expect if the items were statistically independent ($\widehat{=}$ denominator). In the case of *fruits* and *juices*, we would expect a value greather than 1, since only then would it indicate a significant positive correlation. Indeed, $lift(Fruits \Rightarrow Juices) = \frac{5}{4} = 1.2$, which means that the two items appear 20% ($\widehat{=}$ one transaction) more often together than what we would expect under statistical indpendency.

2.2 The *Apriori* Mining Algorithm

Going through a large database and calculate the value of each interest measure for every rule is computationally prohibitive. In our small example, there are already 50 possible rules.[1] It would therefore be desirable to discard some rules without even calculating their interest measures. For this purpose, mining algorithms have been developed. Hipp et al. (2000) systematize and compare the common ones. Here, we will briefly look at *Apriori*, the most well-known of these.

In Figure 1 the rules derived from the earlier example in Table 1 are illustrated. Support and confidence thresholds are set to 0.3 and 0.7 respectively. For simplicity reasons, only one rule derived from the same itemset is considered. Note that, exchanging LHS and RHS items does not affect support values, but it certainly would affect confidence values.

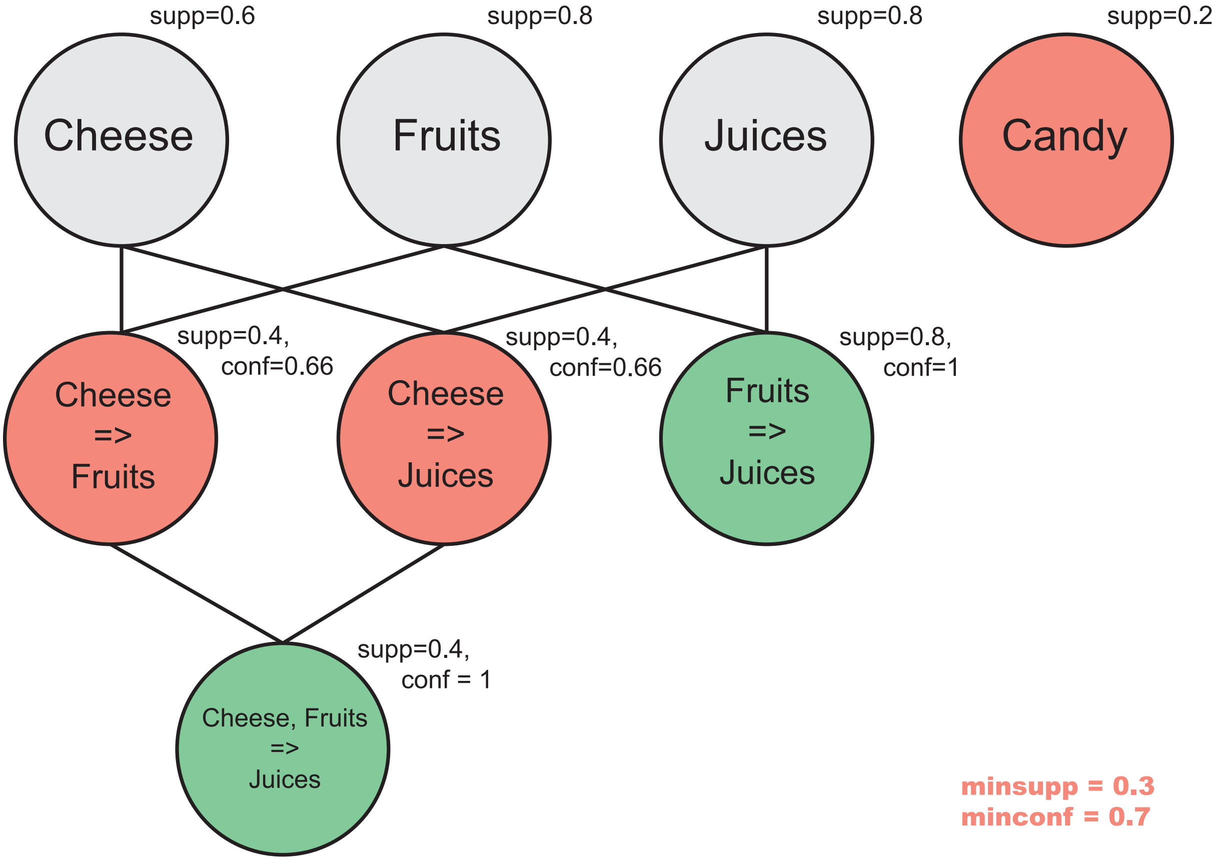

Figure 1: A simplistic illustration of association rule mining with *Apriori*.

Apriori follows a two-step-procedure. In the first step it mines all frequent itemsets – those itemsets that have a support value equal to or greater than *minsupp*. It thereby exploits the *downward-closed property* of support, which means that if an item/itemset is found to be infrequent, than its supersets are also infrequent and thus can be pruned. In the example, *candy* has a support below minsupp. Consequently, all itemsets and rules that consist of *candy* can be ignored. Subsequently, Apriori goes bottom-up, adding one item at a time, and checks if the supersets still satisfy the frequency constraint.

[1] For d items, all possible rules can be calculated with the formula $R = 3^d - 2^{d+1} + 1$ (Tan et al. 2006). The exponential growth results from the possibility to vary the size and composition of the itemset, as well as the item's place in the LHS or RHS of the rule

In the second step, Apriori verifies if all rules derived from frequent itemsets do also have minimal confidence. If not, they are pruned like the rules $Cheese \Rightarrow Fruits$ and $Cheese \Rightarrow Juices$. However, that these two rules do not have minimal confidence, does not imply that $\{Cheese, Fruits\} \Rightarrow \{Juices\}$ won't have it either. Confidence is not downward-closed as support is. That is why it could not be used in the first step.

Choosing the right interest measures and their minimal thresholds is critical to the mining process. If we set minimal support and confidence too low, we might get overwhelmed by the whole amount of rules. If we set minimal support too high, rare items are discriminated that may still contain interesting rules – the so-called *rare itemset problem* (Szathmary et al. 2007). Furthermore, statistical significance tests may be necesary to make sure that mined rules are not the result of coincidence.

3 Association Rule Mining in R

Starting from this section we will apply association rule learning to the statistical software *R*. Hahsler et al. (2009) and Hahsler and Chelluboina (2011) developed two very useful packages called **arules** and **arulesViz**, which facilitate mining and analyzing association rules from a large transaction database. In this section we will learn how to mine association rules using the package arules. Section 4 will then focus on the visualization of the mined rules using arulesViz.

We begin by loading the arules package. To illustrate the function calls, we will use the *Income* dataset, which is already included in arules.[2] `ItemLabels()` displays all items of the dataset. Table 2 in the Appendix shows them arranged by their related variables.

```
library(arules)
data(Income)
itemLabels(Income)
```

The Income dataset is already stored as transaction data. If we'd have a `data.frame` instead, the following example code would coerce the dataset to transaction (be aware that the `data.frame` has to be categorized first).

```
trans <- as(x, "transactions")
```

To get a first impression of the data, we can `inspect` one transaction, get the `summary` statistics and produce an `item frequency plot`.

```
inspect(Income[1])
itemFrequencyPlot(Income, topN=10)
summary(Income)
```

```
  items                         transactionID
1 {income=$40,000+,
   sex=male,
   marital status=married,
   age=35+,
```

[2] **Note**, that in the sense of an introduction, I will not state every possible argument of the function calls. For detailed information, I ask the interested reader, to refer to the arules and arulesViz package documentations that can be downloaded here: `http://lyle.smu.edu/IDA/arules`.

```
   education=college graduate,
   occupation=homemaker,
   years in bay area=10+,
   dual incomes=no,
   number in household=2+,
   number of children=1+,
   householder status=own,
   type of home=house,
   ethnic classification=white,
   language in home=english}                2
```

We see from the *R* output above that our first transaction is an over 35 years old male homemaker who has an income above $40000.
`Summary()` gives us basic information about the dataset. We have 6876 transactions, i.e. persons, (the rows of the `itemMatrix`) and 50 items (the columns of the `itemMatrix`). The Income dataset is prepared in such a way that there are no missings for any of the 14 variables with their 50 attributes/items. Hence, every transaction has the `length` of 14 items and therefore `density` equals $\frac{14\ variables}{50\ items} = 0.28$. Furthermore, `summary` displays the most frequent items and notes that extended item and transaction information are included in the dataset.
Figure 2 plots the ten most frequent items in the dataset. Instead of `topN` we could also have set a minimal `support`, which would have produced an item frequency plot of all items that have or exceed the minimal support.

```
transactions as itemMatrix in sparse format with
 6876 rows (elements/itemsets/transactions) and
 50 columns (items) and a density of 0.28

most frequent items:
     language in home=english education=no college graduate         number in household=1
                         6277                          4849                          4757
  ethnic classification=white         years in bay area=10+                       (Other)
                         4605                          4446                         71330

element (itemset/transaction) length distribution:
sizes
  14
6876

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     14      14      14      14      14      14

includes extended item information - examples:
             labels variables     levels
1 income=$0-$40,000    income $0-$40,000
2   income=$40,000+    income   $40,000+
3          sex=male       sex       male

includes extended transaction information - examples:
  transactionID
1             2
2             3
3             4
```

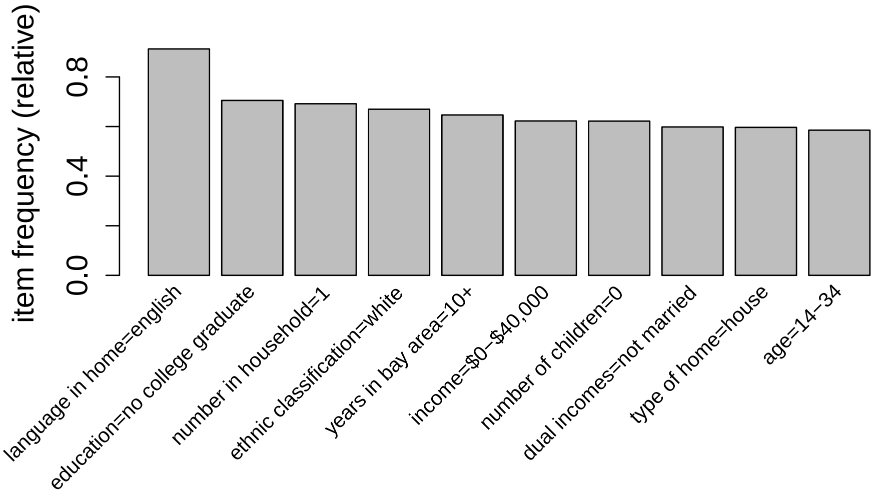

Figure 2: Item frequency plot of the ten items with the highest support in the *Income* Dataset.

Being familiar with the data we can start mining association rules. The mining algorithms Apriori and Eclat by Christian Borgelt (Borgelt and Kruse (2002) and Borgelt (2003)) are already implemented in arules. We call them with the function `apriori()` and `eclat()` respectively. Here, we'll choose `apriori` since the algorithm was explained at length in Section 2.2.

```
rules <- apriori(Income, parameter=list(support=0.1, confidence=0.3, minlen="3", ↩
    maxlen="6", target="rules"), appearance = list(rhs = c("income=$0-$40,000", ↩
    "income=$40,000+"), default="lhs"))
```

The list of `parameters` set the constraints for mining association rules or frequent itemsets. Apart from `support` and `confidence`, we can also set the minimum (`minlen`) or maximum (`maxlen`) length of the itemset (≙ the sum of the items in the LHS and RHS of the rule). Furthermore, we could first `target frequent`, `maximally frequent` or `closed frequent` itemsets before directly mining `rules`, using the function `ruleInduction`.[3]

```
closed_frequent_itemsets <- apriori(Income, parameter=list(support=0.1, ↩
    confidence=0.3, target="closed frequent"))
rules_closed <- ruleInduction(closed_frequent_itemsets, Income)
```

`Appearance` allows us to further specify the composition of the rule. Suppose we are interested in what determines income. In this case, we restrict the income items to the consequent and apply the `default="lhs"` option to keep all other items in the antecedent of the rule.
After Apriori has crawled through the data, we use `summary()` again to get information about the amount of mined rules, their length distribution as well as descriptive statistics concerning the interest measures support, confidence and lift.

[3] Frequent itemsets are all itemsets that have or exceed minimal support. A frequent itemset is closed frequent if there is no superset that has the same support and it is further maximally frequent if there is no frequent superset. For a more detailed and comprehensive description of maximally frequent and closed frequent itemsets, check out `http://www.hypertextbookshop.com/dataminingbook/working_version/contents/chapters/chapter002/section004/blue/page001.html` by Vemma (2009).

```
summary(rules)
```

```
set of 1761 rules

rule length distribution (lhs + rhs):sizes
  3   4   5   6
249 591 637 284

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.000   4.000   5.000   4.543   5.000   6.000

summary of quality measures:
    support         confidence          lift
 Min.   :0.1001   Min.   :0.3015   Min.   :0.5358
 1st Qu.:0.1115   1st Qu.:0.6485   1st Qu.:1.1739
 Median :0.1264   Median :0.7643   Median :1.2887
 Mean   :0.1414   Mean   :0.7195   Mean   :1.3177
 3rd Qu.:0.1550   3rd Qu.:0.8115   3rd Qu.:1.3784
 Max.   :0.4455   Max.   :0.9161   Max.   :2.3188

mining info:
   data ntransactions support confidence
 Income          6876     0.1        0.3
```

If we are satisfied with the set of mined rules, we can start analyzing them. For example by sorting the rules according to our preferred measure. It is recommended to try out various interest measures for comparison. Here, according to lift, if someone is married, householder and occupies a professional/mangerial position, he is very likely to have an income above $40000.

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

```
  lhs                                       rhs                  support  confidence     lift
1 {marital status=married,
   occupation=professional/managerial,
   householder status=own}              => {income=$40,000+} 0.1042757  0.8754579 2.318817
2 {marital status=married,
   occupation=professional/managerial,
   type of home=house}                  => {income=$40,000+} 0.1038394  0.8409894 2.227520
3 {marital status=married,
   dual incomes=yes,
   householder status=own,
   type of home=house,
   language in home=english}            => {income=$40,000+} 0.1051483  0.8339100 2.208769
```

In order to reduce the amount of rules to analyze, it might be helpful to create subset of rules. Below, `subrules1` gives us 297 rules that have a lift value greater than 1.5. `Subrules2` excludes all rules that contain the item `dual incomes="not married"` in their LHS, whereas `subrules3` excludes all rules that do not have that item in their LHS, resulting in a set of 1257 and 504 rules respectively.

```
subrules1 <- rules[quality(rules)$lift > 1.5]
subrules2 <- subset(rules, subset = !(lhs %in% "dual incomes=not married"))
subrules3 <- subset(rules, subset = lhs %in% "dual incomes=not married")
```

Moreover, we are able to add new interest measures to our analysis. Many of the commonly used interest measurs are already implemented in arules so that we just have to call them. Take for example `chiSquared` by Liu et al. (1999), which computes the chi-squared statistic to test whether the rule's LHS and RHS are statistically independent or not. The following code adds `chiSquared` to our set of interest measures. Just exchange the argument of `method` to add a different one.

```
quality(rules) <- cbind(quality(rules), chiSquared = interestMeasure(rules, method = ↩
    "chiSquared", significance=TRUE, transactions = Income))
```

`Significance=TRUE` is an additional argument that returns the P-Value instead of the chi-squared test statistic. Regarding the set of rules, it would then be reasonable to drop all rules where the LHS and RHS are not statistically dependent. Choosing a significance level of $\alpha = 0.05$, we create a subset that leaves us with a set of 1694 rules out of the original 1761.

```
subrules_significant <- rules[quality(rules)$chiSquared < 0.05]
subrules_significant
```

```
set of 1694 rules
```

4 Visualizing Association Rules

Running `apriori()` for the first time, usually yields a large set of rules. Besides sorting and filtering them, we can also make use of graphical visualizations provided by the **arulesViz** package (Hahsler and Chelluboina 2011). For the purpose of an introduction, we will only look at a few possible visualizations. We will start with a simple scatter plot in Section 4.1, followed by a colored matrix-based visualization of association rules in Section 4.2. The grouped matrix-based visualization presented in Section 4.3 groups similar rules together and represents them as ballons with varying size and color in a matrix. Finally, Section 4.4 looks at the graphed-based visualization, which displays rules using arrows and nodes that vary in size and color according to the interest measures we set.

4.1 Scatter Plot

```
plot(subrules_significant, method="scatter", measure=c("support", "confidence"), ↩
    shading="lift", interactive=TRUE)
```

All visualizations are called with the function `plot` and further specified with `method`. Our scatter plot consists of the rules we mined in Section 3 for the Income dataset. We select support and confidence as our interest `measures` on the horizontal and vertical axis, and choose lift for `shading` – we can exchange them as we like. The resulting plot is depicted in Figure 3.

Below the scatter plot are the features included in the `interactive` mode.The features are quite intuitive and let us explore the scatter plot in more detail by **inspecting** individual or sets of rules, by **zooming** into a selected region or by **filtering** all rules with a measure lower than a selected cut-off point in the shading bar (Hahsler and Chelluboina 2011).[4]

[4] An illustrative example of the interactive mode is given in the Appendix. Note, that the interactive mode is in some way available for all visualizations presented in this section.

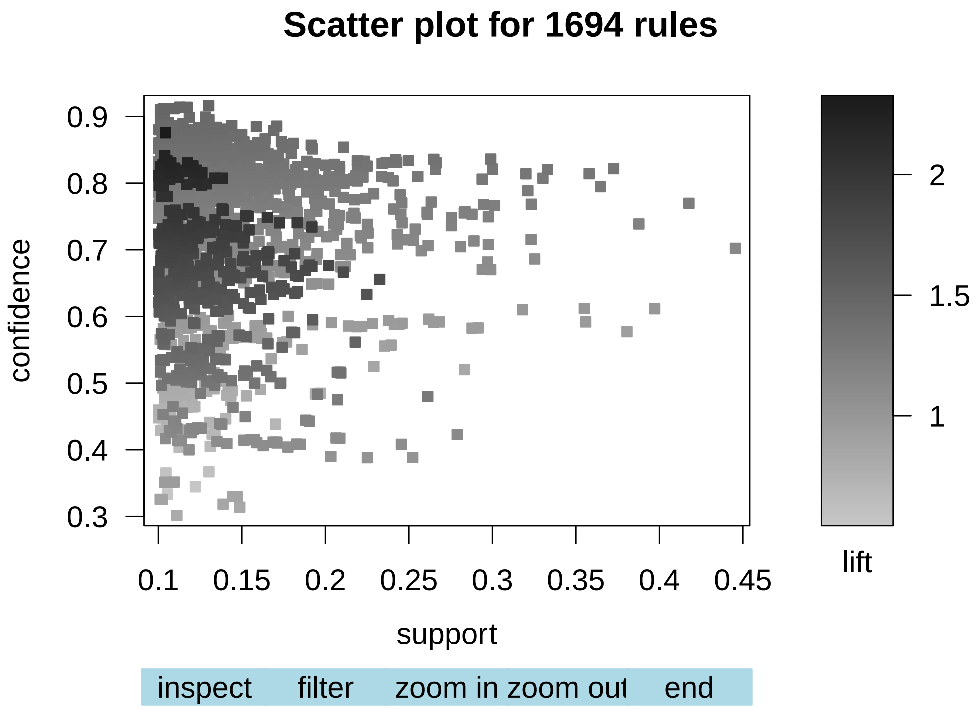

Figure 3: Scatter plot of the significant subset of rules mined in Section 3 according to the chi-squared test statistic.

An interesting fact that we can derive from the scatter plot is that support seems to be inversely correlated with confidence and lift. Apparently, the more we move on the horizontal axis to the right, the lighter the dots and the less cluttered the plot. Additionally, lift seems to be positively correlated with confidence, as dots tend to get darker when moving up the vertical axis.
Another correlation of that sort can be observed in Figure 4. Instead of shading the rules according to their lift measure, the Two-key plot shades them according to their length. As we increase support, the dots get lighter, suggesting that the support of a rule is a decreasing function of its length.

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

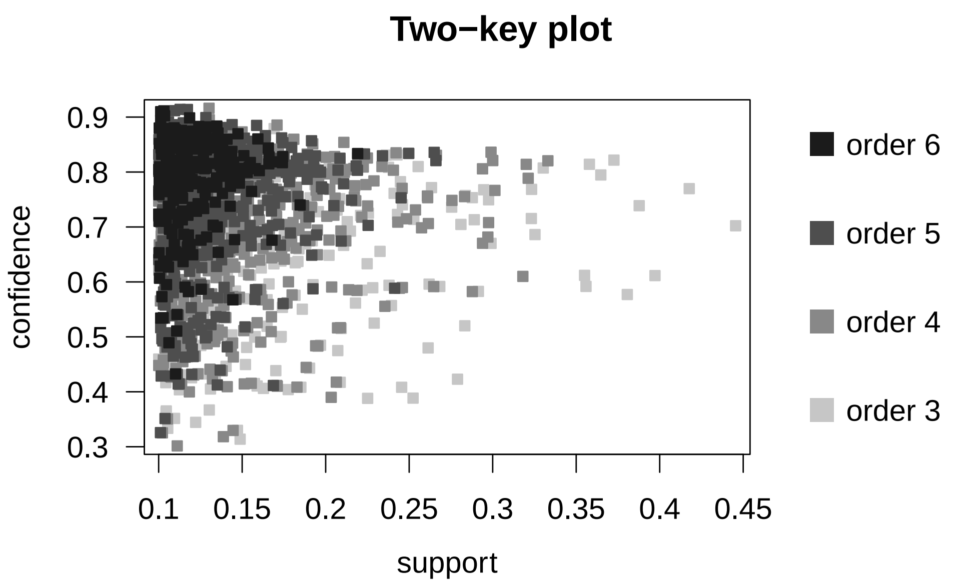

Figure 4: Two-key plot of the significant subset of rules mined in Section 3 according to the chi-squared test statistic.

4.2 Matrix-Based Visualization

```
plot(subrules_significant, method="matrix", measure=c("lift", "support"), ↩
    control=list(reorder=TRUE), interactive=TRUE)
```

By simply changing the method to `matrix` we get a matrix-based visualization.[5] Figure 5 shows the antecedent items on the horizontal and the consequent items – i.e. `income="$0-$40,000"` and `income="$40,000+"` – on the vertical axis. Each stripe represents one rule, which is colored according to the rule's values for the selected interest measures. Here, lift rises from blue to red and support from light to dark. The `control`-option `reorder=TRUE` orders the stripes to make it easier to see patterns.[6] If we want to inspect a rule, we take advantage of the interactive mode and just click on the respective stripe.

[5] Matrix-based visualizations are also available for just one measure or in 3D (Hahsler and Chelluboina 2011).

[6] Refer to the **seriation** package by Buchta et al. (2008) and Hahsler and Chelluboina (2011) for more information about how reordering is done.

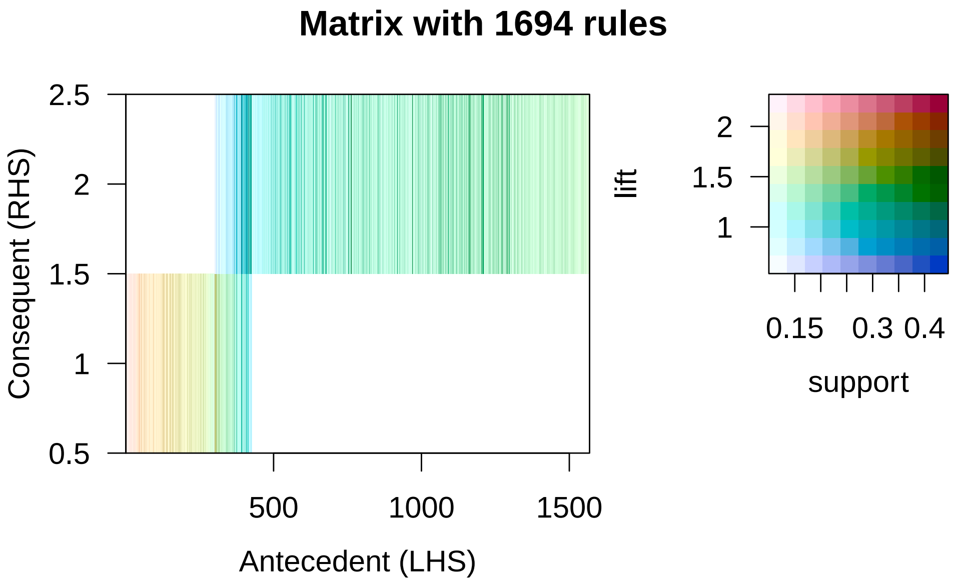

Figure 5: Matrix-based visualization of the significant subset of rules mined in Section 3 according to the chi-squared test statistic.

4.3 Grouped Matrix-Based Visualization

```
subrules_small <- head(sort(subrules_significant,by="confidence"), n=50)
library(colorspace)
plot(subrules_small, method="grouped", measure="support", shading="lift", ↩
    control=list(k=10, col=sequential_hcl(10, h=10, c.=c(200,10))), interactive=TRUE)
```

Method `grouped` produces a grouped matrix-based visualization. Using *k-means clustering* LHS items that are statistically dependent on the same RHS item are grouped together (Hahsler and Chelluboina 2011). In Figure 6 we have `k=10` LHS item groups, reported in columns, and two RHS items, reported in rows. The groups are lead by one item that is common to all rules, followed by the amount of all other items in the group. Behind the brackets, the amount of rules belonging to the group are stated.

At the intersection, the ballon represents the group's median value of the rules' interest measures. `Measure="support"` determines the ballon's size and `shading="lift"` its color.[7] Large and dark red ballons should indicate frequent and strong association rules. Again, we can employ the interactive features to inspect the ballons' underlying rules.

[7] I loaded the `colorspace` package to change the shading from black to red in `control=list(col=)`.

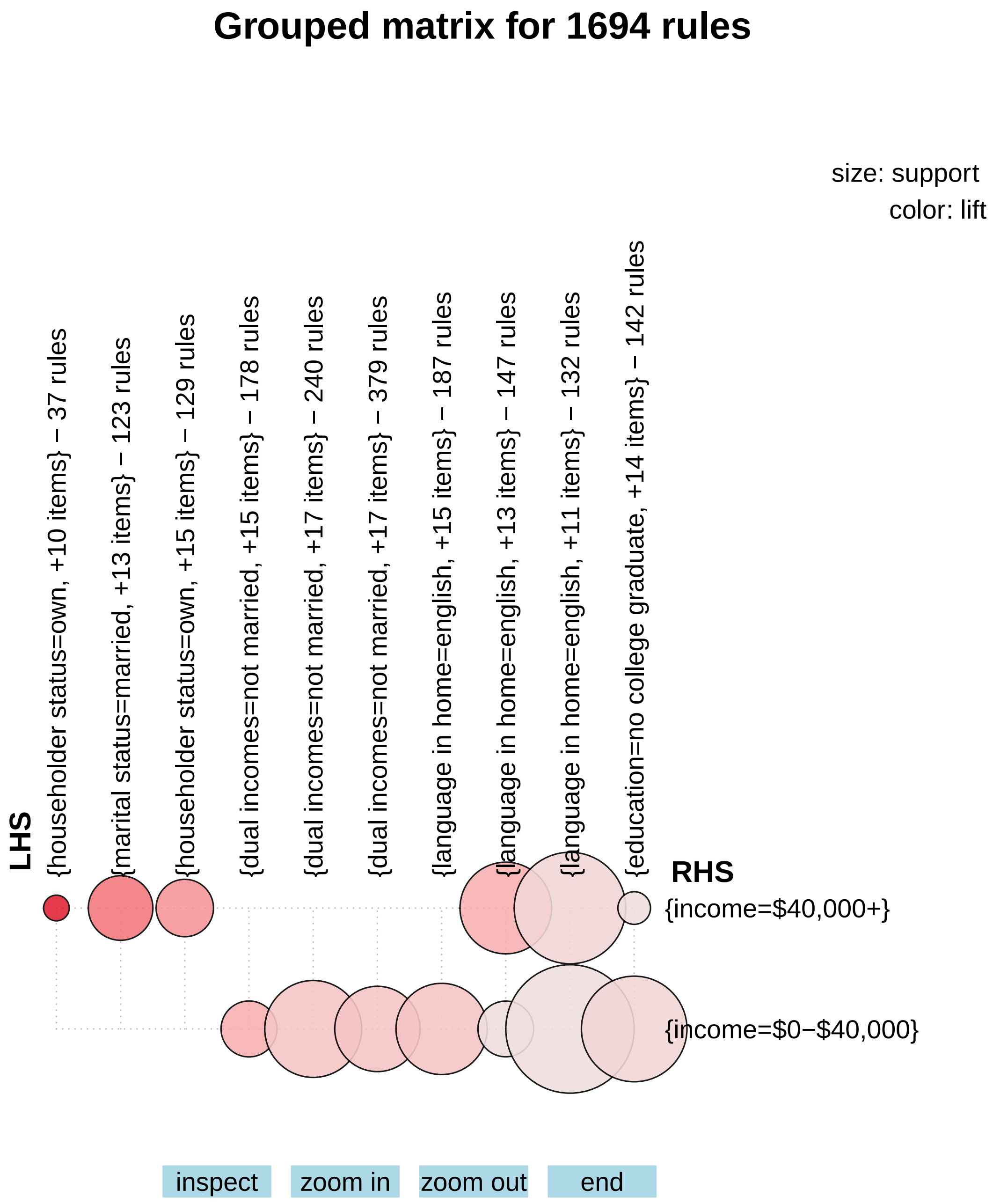

Figure 6: Grouped matrix-based visualization of the significant subset of rules mined in Section 3 according to the chi-squared test statistic.

4.4 Graphed-Based Visualization

```
subrules_smallest <- head(sort(subrules_significant, by="confidence"), n=2)
plot(subrules1_1, method="graph", measure="support", shading="lift")
```

Without additional interactive features to the built-in ones, the graphed plot is only feasible for a small set of rules (Hahsler and Chelluboina 2011). For that and illustrative reasons, we cut the statistically significant subset of rules down to only two rules with highest confidence. The resulting plot is shown in Figure 7. Each node stands for one rule, where the rule's LHS items point with an arrow to the node and from there with an arrow to the rule's RHS item. Just like in Figure 6, the node's size is determined by the rule's support and its color by the rule's lift measure. E.g., the small dark node to the left represents the rule {education=no college graduate, dual incomes=not married, type of home=apartment} ⇒ {income=\$0-\$40,000} that has a lower support but a higher lift measure compared to the other rule (node).

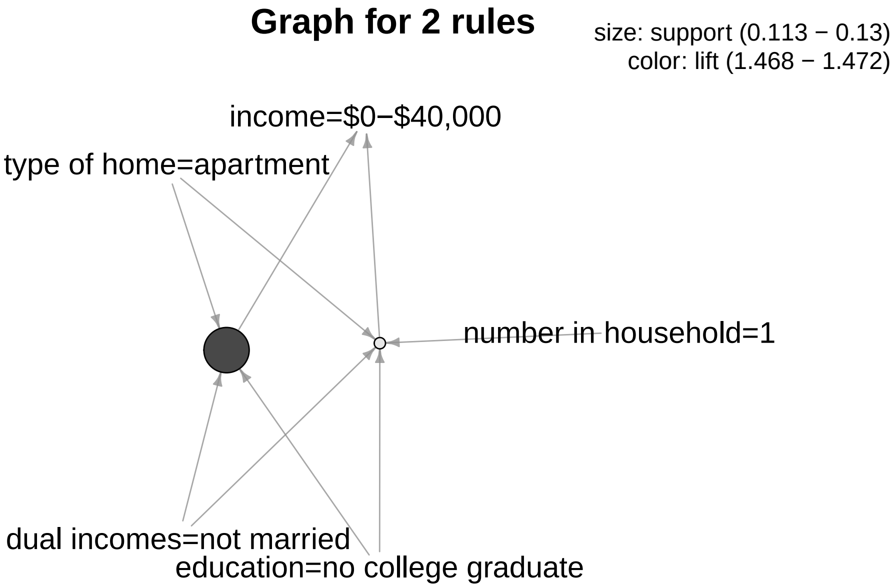

Figure 7: Grouped matrix-based visualization of the significant subset of rules mined in Section 3 according to the chi-squared test statistic.

Hahsler and Chelluboina (2011) compare every plot of the arulesViz package subject to the size of the rule set, to the amount of interest measures that can be applied, the ease of use and whether we can reorder the rules or explore them interactively. To summarize the plots presented here, rule sets vary from large (scatter plot) to small (graph). All plots allow for two or more interest measures. The ease of use is best for the scatter and graph-based plots. In some way, the interactive feature is always a possible option and reordering is avaibale for all but scatter plots.

5 Summary

Association rule learning is a commonly used method for discovering sales patterns in market basket analysis. But any categorized data can be coerced to transactions to mine association rules. In this seminar paper, I gave an introduction to association rule mining with the statistical software *R*. Using the packages **arules** and **arulesViz**, I conducted an exemplary mining process and presented some graphical visualizations of the mined rules.

To improve the mining process or to find more specific rules, advanced association rule learning will take advantage of other data mining techniques that built on the arules environment. For example, Hahsler and Hornik (2007) show how transaction data can be clustered before starting the mining process. More extensions are continuously added (Hahsler et al. 2011).

A Appendix

Item	item label
income	0-40,000; 40,000+
sex	female; male
marital status	married; cohabitation; divorced; widowed; single
age	14-34; 35+
education	college graduate; no college graduate
occupation	sales; professional/managerial; clerical/service; laborer; student; homemaker; military; retired; unemployed
years in bay area	1-9; 10+
dual incomes	not married; yes; no
number in household	1; 2+
number of children	0; 1+
householder status	rent; own; live with parents/family
type of home	house; condominium, apartment; mobile home; other
ethnic classification	asian; black; american indian; east indian; hispanic; pacific islander; white
language in home	english; spanish; other

Table 2: Items and item labels of the *Income* dataset.

A.1 The interactive feature

Following the scatter plot in Section 4.1, I tried out the different features of the interactive mode. First of all, I filtered all rules which have a lift value greater than 1.5 by first clicking somewhere near 1.5 in the shading bar and then on *Filter*. The resulting plot is depicted in Figure 8, from which I zoomed in the reddish-highlighted area (you create a rectangle by selecting its top-left and bottom-right corner with two clicks). Then, in Figure 9 I selected two rules with another rectangle and inspected them.

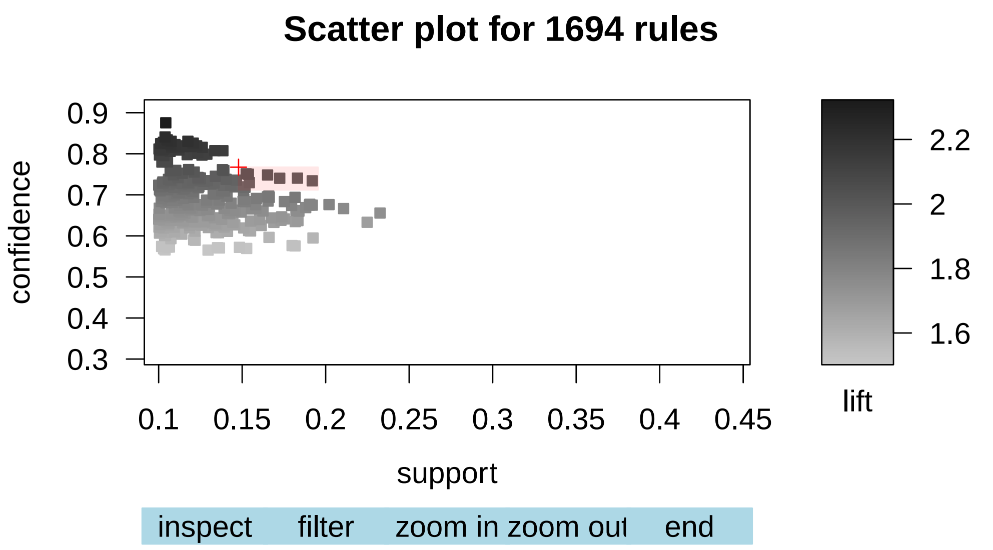

Figure 8: Scatter plot of the rules depicted in Figure 3 filtered by `lift>1.5`.

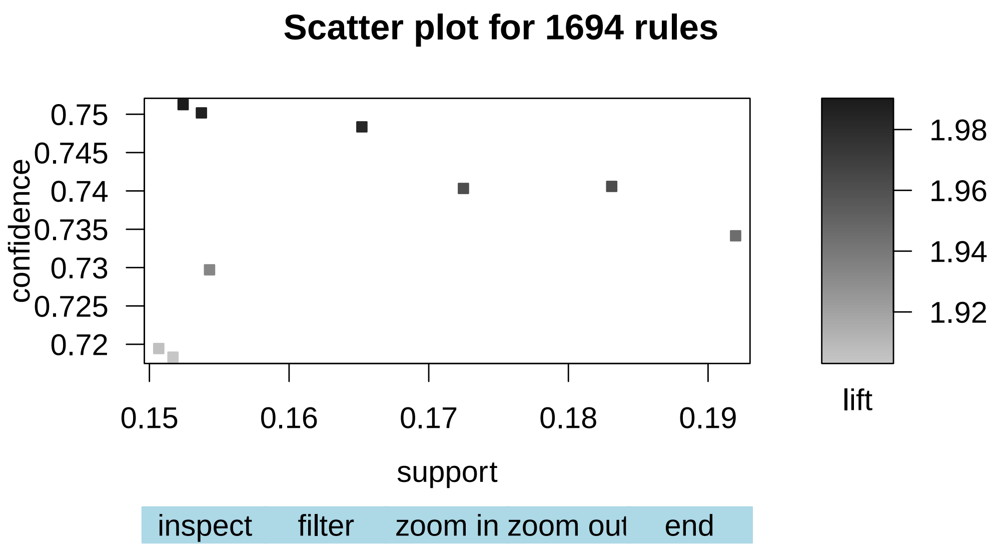

Figure 9: Zoomed-in scatter plot with highlighted area to inspect.

```
Number of rules selected: 2
  lhs                              rhs                   support  confidence     lift    chiSquared
1 {marital status=married,
   householder status=own,
   language in home=english} => {income=$40,000+} 0.1831006  0.7405882 1.961589 2.076852e-277
2 {marital status=married,
   householder status=own,
   type of home=house}       => {income=$40,000+} 0.1724840  0.7403246 1.960891 2.370116e-256
```

B References

Agrawal Rakesh.; Imielinski, T. and A. Swami (1993). *Mining Association Rules between Sets of Items in Large Databases*. In: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD*, pp. 207–216.

Borgelt, C. (2003). *Efficient Implementations of Apriori and Eclat*. In: *FIMI 03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations*.

Borgelt, C. and R. Kruse (2002). *Induction of Association Rules: Apriori Implementation*. In: *Compstat*. Springer, pp. 395–400.

Brin, S., R. Motwani, J. D. Ullman, and S. Tsur (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.

Buchta, C., K. Hornik, and M. Hahsler (2008). *Getting Things in Order: An Introduction to the R Package Seriation*. In: *Journal of Statistical Software*, Vol. 25, No. 3, pp. 1–34.

Hahsler, M. (2015). *A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules*. http://michael.hahsler.net/research/association_rules/measures.html. [Online; accessed 12-July-2015].

Hahsler, M. and S. Chelluboina (2011). *Visualizing Association Rules: Introduction to the R-extension Package ArulesViz*. In: *R project module*, pp. 223–238.

Hahsler, M. and K. Hornik (2007). *Building on the Arules Infrastructure for Analyzing Transaction Data with R*. In: *Advances in Data Analysis*. Springer, pp. 449–456.

Hahsler, M., B. Grün, K. Hornik, and C. Buchta (2009). *Introduction to Arules – A Computational Environment for Mining Association Rules and Frequent Item Sets*. In: *The Comprehensive R Archive Network*.

Hahsler, M., S. Chelluboina, K. Hornik, and C. Buchta (2011). *The Arules R-Package Ecosystem: Analyzing Interesting Patterns from Large Transaction Data Sets*. In: *The Journal of Machine Learning Research*, Vol. 12, pp. 2021–2025.

Hipp, J., U. Güntzer, and G. Nakhaeizadeh (2000). *Algorithms for Association Rule Mining – A General Survey and Comparison*. In: *ACM sigkdd explorations newsletter*, Vol. 2, No. 1, pp. 58–64.

Liu, B., W. Hsu, and Y. Ma (1999). *Pruning and Summarizing the Discovered Associations*. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 125–134.

Power, D. J. (2002). *What is the "True Story" about Using Data Mining to Identify a Relation between Sales of Beer and Diapers?* http://www.dssresources.com/newsletters/66.php. [Online; accessed 12-July-2015].

Szathmary, L., A. Napoli, and P. Valtchev (2007). *Towards Rare Itemset Mining*. In: *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on*. Vol. 1. IEEE, pp. 305–312.

Tan, P.-N., M. Steinbach, and V. Kumar (2006). *Introduction to Data Mining*. Pearson. Chap. 6, pp. 330–331.

Vemma, R. (2009). *Compact Representation of Frequent Itemset*. http://www.hypertextbookshop.com/dataminingbook/working_version/contents/chapters/chapter002/section004/blue/page001.html. [Online; accessed 13-July-2015].

C List of Figures

D List of Tables