

```

1 # Association Rules Mining mit apriori Algorithmus in Cran-R
2 #
3 # Agrawal, Mannila, Srikant, Toivonen, Verkamo 1996
4 #
5 # Idee:
6 # * Wenn eine Menge häufig ist, so auch all ihre Teilmengen. (Anti-Monotonie)
7 # * Wenn eine Menge selten ist, so auch all ihre Obermengen. (Monotonie)
8
9 download.file("http://www.rdatamining.com/data/titanic.raw.rdata", "titanic.raw.rdata") # download Titanic Daten
10 load("titanic.raw.rdata")
11
12 # Anzeige was im Workspace enthalten ist
13 ls()
14
15 # Inspektion des Datensatzes
16 str(titanic.raw)
17
18 # -> 4 Kategorische Variablen 2201 Beobachtungen
19
20 require(arules)
21
22 rules <- apriori(titanic.raw)
23 inspect(rules)
24
25 # support - Wahrscheinlichkeit, dass diese Itemmenge in einer Transaktion vorkommt.
26 # confidence - Konfidenz einer Regel entspricht der Wahrscheinlichkeit der Konklusion unter Bedingung der Prämisse:
27 # lift - Der Lift gibt an, wie hoch der Konfidenzwert für die Regel den Erwartungswert übertrifft, er zeigt also die generelle Bedeutung einer Regel
28
29 # siehe auch: http://www.borgelt.net/doc/apriori/apriori.html
30
31 # Wir setzen rhs=c("Survived=No", "Survived=Yes") in appearance damit nur "Survived=No" and "Survived=Yes" in der rhs auftauchen
32
33 rules <- apriori(titanic.raw,
34 parameter = list(minlen=2, supp=0.005, conf=0.8),
35 appearance = list(rhs=c("Survived=No", "Survived=Yes"),
36 default="lhs"),
37 control = list(verbose=F))
38
39 rules.sorted <- sort(rules, by="lift")
40
41
42 # Finden redundanter Regeln
43
44 subset.matrix <- is.subset(rules.sorted, rules.sorted)
45 subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
46 redundant <- colSums(subset.matrix, na.rm=T) >= 1
47 which(redundant)
48 rules.pruned <- rules.sorted[!redundant]
49
50 inspect(rules.pruned)
51
52 # Visualisierung der Regeln
53 require(arulesViz)
54
55 plot(rules)
56
57 plot(rules, method="graph", control=list(type="items"))
58
59 plot(rules, method="paracoord", control=list(reorder=TRUE))

```