Assoziationsmining

Datalab Brown-Bag-Seminar

Thoralf Mildenberger

Institut für Datenanalyse und Prozessdesign
School of Engineering
Zürcher Hochschule für Angewandte Wissenschaften

09.07.2014

Zürcher Hochschule
für Angewandte Wissenschaften

School of
Engineering

IDP Institut für Datenanalyse
und Prozessdesign

Motivation

- Projekt mit Telekom-Anbieter, Ziel war Quantifizierung von Aufwand bei Supportanfragen
- Viele Daten vorhanden zu Kunden und Supporttickets, nur für Kunden mit Supportanfragen
- Interessant: Gruppen mit hohem Anteil an Supportvolumen oder -kosten
- Nicht möglich: Aussagen über Wahrscheinlichkeiten für Supportanfragen in bestimmten Kundengruppen, da keine Daten für Kunden ohne Anfrage vorliegen
- Kein Prognoseproblem!
- Keine eindeutige Response
- **Vorgehensweise:** Benutzung von Methoden des **Frequent Itemset Mining** bzw. **Mining von Assoziationsregeln**

Übersicht

- Frequent Itemset Mining
- Assoziationsregeln
- Erweiterung auf allgemeinere kategorielle Variablen
- Variation: Anwendung auf mit Kosten bewertertete Requests
- Bemerkungen und Fazit

Keine Originaldaten aus dem Projekt!

Frequent Itemsets

- Software: Algorithmen `apriori` und `eclat`, implementiert in `R`-Paket `arules` (Vignette lesenswert!)
- Ursprüngliche Anwendung der Methode: *Market Basket Analysis* (welche Produkte werden häufig zusammen gekauft)
- Variablen ursprünglich nur: Produkt A gekauft / nicht gekauft, Produkt B gekauft / nicht gekauft, etc.
- Nur Ja/Nein, keine Mengenangaben!

Frequent Itemsets

Typischer Datensatz sieht so aus:

Transaktion 1	Milch, Brot
Transaktion 2	Brot, Butter
Transaktion 3	Bier
Transaktion 4	Milch, Brot, Butter
Transaktion 5	Brot, Butter

Frequent Itemsets

Zur Analyse Darstellung als **Inzidenzmatrix** bzw. im sog. **horizontalen Layout**:

Transaktion	Milch	Brot	Butter	Bier
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	1	0

Frequent Itemsets

Alternative ("vertikale") Darstellung:

Item	Transaktionen
Milch	1,4
Brot	1,2,4,5
Butter	2,4
Bier	3

Frequent Itemsets

Qualität von Itemsets: Die Interessantheit wird gemessen durch den *Support* eines Itemsets;

$$\begin{aligned}&\text{supp}(\{\text{Item 1, Item 2,... }\})\\ =&\frac{\text{Anzahl Transaktionen mit Item 1 und Item 2,...}}{\text{Anzahl Transaktionen insgesamt}}\end{aligned}$$

Durch hinzufügen von Items wird der Support kleiner, da der Warenkorb immer genauer bestimmt wird.

Allgemein entsprechen $\cup$ und $\cap$ bei Itemsets $\cap$ und $\cup$ bei den zugehörigen Transaktionen!

Frequent Itemsets

- Von Interesse: Auffinden aller Itemsets mit Mindestsupport (z.B. 10%)
- Kombinatorisches Problem: Bei n verschiedenene Items gibt es $2^n - 1$ verschiedene nichtleere Kombinationen von Items
- Problem wird handhabbar durch: Obergrenze Anzahl Items pro Itemset (also z.B. nur Itemsets mit 1,2,...,7 Items)
- Wichtiger: Man weiss **apriori**, dass durch Hinzufügen eines Items der Support nur kleiner werden kann. Hat also ein Set von 2 Items einen Support kleiner dem Minimalsupport, so muss man Obermengen von Items nicht mehr betrachten
- `apriori`-Algorithmus fängt mit einelementigen Itemsets an und betrachtet dann grössere Sets, wobei in jedem Schritt Itemsets mit zu geringem Support eliminiert werden
- Es existieren effizientere Alternativen wie z.B. `eclat`, die auf algebraischen Überlegungen oder Bäumen etc. basieren

Beispiel aus `arules`:

```
> library(arules)
> data(Groceries)
>
> itemsets<-apriori(Groceries,parameter=list(supp=0.03,target="frequent itemsets"))

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen            target   ext
        0.8    0.1    1 none FALSE            TRUE    0.03      1     10 frequent itemsets 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 ... [44 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [63 set(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Frequent Itemsets

```
> inspect(itemsets)
   items                           support
1  {specialty chocolate}      0.03040163
2  {UHT-milk}                 0.03345196
3  {onions}                   0.03101169
4  {berries}                  0.03324860
5  {hamburger meat}           0.03324860
6  {hygiene articles}         0.03294357
7  {salty snack}              0.03782410
8  {sugar}                    0.03385867
9  {waffles}                  0.03843416
10 {long life bakery product} 0.03741739
11 {dessert}                  0.03711235
12 {canned beer}              0.07768175
13 {cream cheese }            0.03965430
14 {chicken}                  0.04290798
15 {white bread}              0.04209456
16 {chocolate}                0.04961871
17 {coffee}                   0.05805796
18 {frozen vegetables}        0.04809354
19 {beef}                     0.05246568
20 {curd}                     0.05327911
21 {napkins}                  0.05236401
22 {pork}                     0.05765125
23 {frankfurter}              0.05897306
24 {bottled beer}             0.08052872
25 {brown bread}              0.06487036
26 {margarine}                0.05856634
27 {butter}                   0.05541434
28 {newspapers}               0.07981698
29 {domestic eggs}            0.06344687
30 {fruit/vegetable juice}    0.07229283
```

Frequent Itemsets

	items	support
31	{whipped/sour cream}	0.07168277
32	{pip fruit}	0.07564820
33	{pastry}	0.08896797
34	{citrus fruit}	0.08276563
35	{shopping bags}	0.09852567
36	{sausage}	0.09395018
37	{bottled water}	0.11052364
38	{tropical fruit}	0.10493137
39	{root vegetables}	0.10899847
40	{soda}	0.17437722
41	{yogurt}	0.13950178
42	{rolls/buns}	0.18393493
43	{other vegetables}	0.19349263
44	{whole milk}	0.25551601
45	{whole milk, whipped/sour cream}	0.03223183
46	{pip fruit, whole milk}	0.03009659
47	{whole milk, pastry}	0.03324860
48	{citrus fruit, whole milk}	0.03050330
49	{sausage, rolls/buns}	0.03060498
50	{whole milk, bottled water}	0.03436706
51	{tropical fruit, other vegetables}	0.03589222
52	{tropical fruit, whole milk}	0.04229792

Frequent Itemsets

	items	support
53	{root vegetables, other vegetables}	0.04738180
54	{root vegetables, whole milk}	0.04890696
55	{rolls/buns, soda}	0.03833249
56	{other vegetables, soda}	0.03274021
57	{whole milk, soda}	0.04006101
58	{yogurt, rolls/buns}	0.03436706
59	{other vegetables, yogurt}	0.04341637
60	{whole milk, yogurt}	0.05602440
61	{other vegetables, rolls/buns}	0.04260295
62	{whole milk, rolls/buns}	0.05663447
63	{other vegetables, whole milk}	0.07483477

Assoziationsregeln

Idee: Betrachtung von Assoziationsregeln des Typs:

$$\{\text{Item 1, Item 2}\} \Longrightarrow \{\text{item 3}\},$$

also z.B. “wer Brot und Butter kauft, kauft auch Konfitüre”

Qualität von Assoziationsregeln: *Confidence*

$$\begin{aligned}&\text{conf}(\text{linke Seite} \Longrightarrow \text{rechte Seite})\\ =&\frac{\text{Anzahl Transaktionen mit Items der linken und rechten Seite}}{\text{Anzahl Transaktionen mit Items der linken Seite}}\end{aligned}$$

und *Lift*:

$$\text{lift}(\text{linke Seite} \Longrightarrow \text{rechte Seite}) = \frac{\text{conf}(\text{linke Seite} \Longrightarrow \text{rechte Seite})}{\text{supp}(\text{rechte Seite})}$$

Assoziationsregeln

```
> rules<-apriori(Groceries,parameter=list(supp=0.01,conf=0.5,target="rules"))

parameter specification:
 confidence minval smax arem  aval originalSupport support minlen maxlen target   ext
        0.5    0.1    1 none FALSE            TRUE    0.01      1     10  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 ... [88 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [15 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Assoziationsregeln

```
> inspect(rules)
   lhs                      rhs                    support confidence     lift
1  {curd,
    yogurt}              => {whole milk}       0.01006609  0.5823529 2.279125
2  {other vegetables,
    butter}              => {whole milk}       0.01148958  0.5736041 2.244885
3  {other vegetables,
    domestic eggs}       => {whole milk}       0.01230300  0.5525114 2.162336
4  {yogurt,
    whipped/sour cream}  => {whole milk}       0.01087951  0.5245098 2.052747
5  {other vegetables,
    whipped/sour cream}  => {whole milk}       0.01464159  0.5070423 1.984385
6  {pip fruit,
    other vegetables}    => {whole milk}       0.01352313  0.5175097 2.025351
7  {citrus fruit,
    root vegetables}     => {other vegetables} 0.01037112  0.5862069 3.029608
8  {tropical fruit,
    root vegetables}     => {other vegetables} 0.01230300  0.5845411 3.020999
9  {tropical fruit,
    root vegetables}     => {whole milk}       0.01199797  0.5700483 2.230969
10 {tropical fruit,
    yogurt}              => {whole milk}       0.01514997  0.5173611 2.024770
11 {root vegetables,
    yogurt}              => {other vegetables} 0.01291307  0.5000000 2.584078
12 {root vegetables,
    yogurt}              => {whole milk}       0.01453991  0.5629921 2.203354
13 {root vegetables,
    rolls/buns}          => {other vegetables} 0.01220132  0.5020921 2.594890
14 {root vegetables,
    rolls/buns}          => {whole milk}       0.01270971  0.5230126 2.046888
15 {other vegetables,
    yogurt}              => {whole milk}       0.02226741  0.5128806 2.007235
```

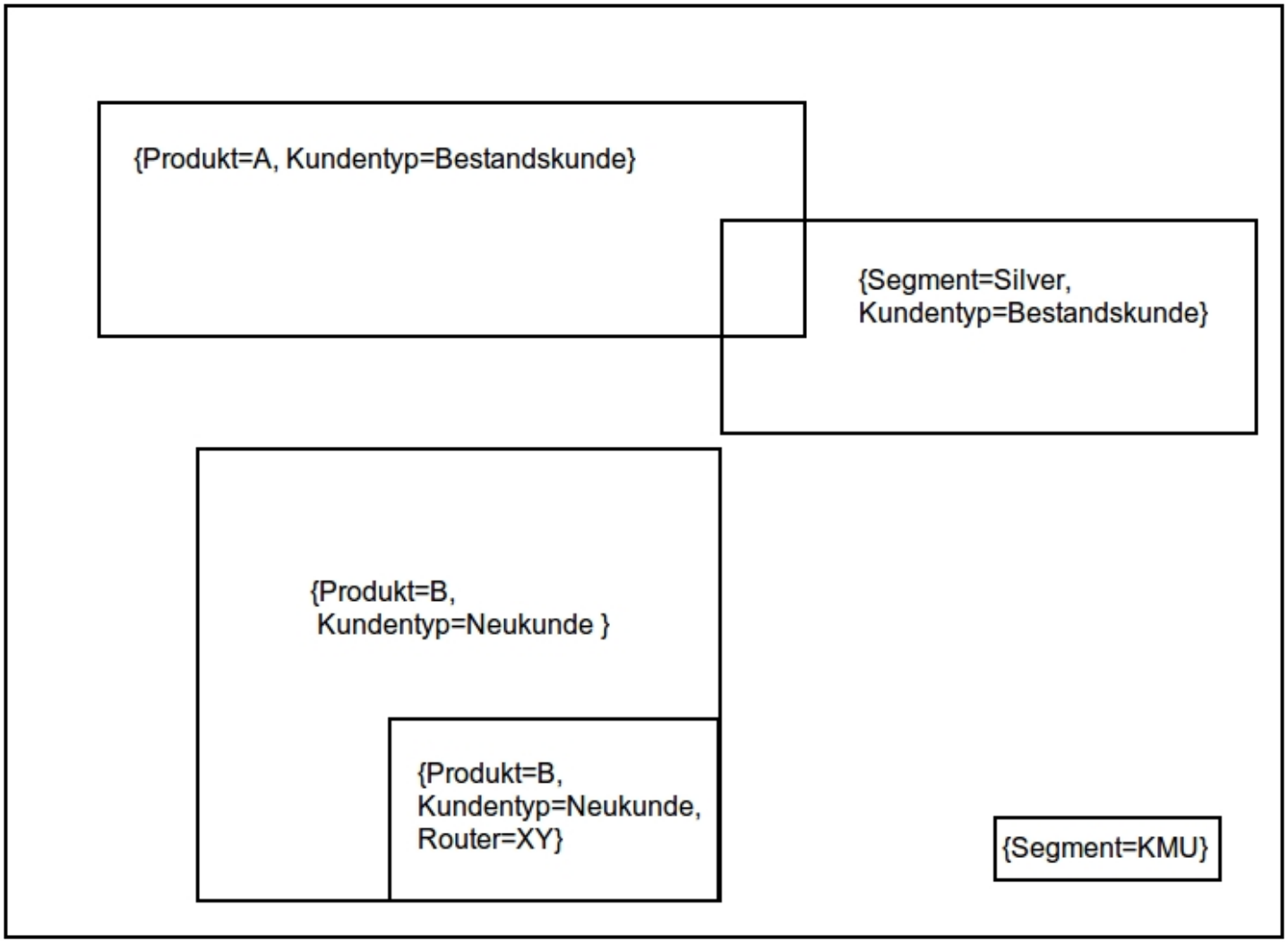
Erweiterung auf allgemeinere kategorielle Variablen

- Ursprüngliche Anwendung der Methode: *Market Basket Analysis* (welche Produkte werden häufig zusammen gekauft)
- Variablen ursprünglich nur: Produkt A gekauft / nicht gekauft, Produkt B gekauft / nicht gekauft, etc.
- Hier: erweitert auf kategorielle Variablen, wie z.B: `Segment`, codiert als `Segment=KMU`, ist dann jeweils "wahr" oder "falsch"
- Algorithmus sucht nun *"itemsets"*, also Kombinationen vom Eigenschaften, die häufig gemeinsam auftreten, zum Beispiel:
  - `{Produkt=A}`
  - `{Produkt=A, Kundentyp=Bestandskunde}`
  - `{Produkt=B, Tarif=Classic}`
- Es muss nicht vorher festgelegt werden, welche Variablen interessant sind, Gruppen können durch verschiedene Variablen charakterisert sein

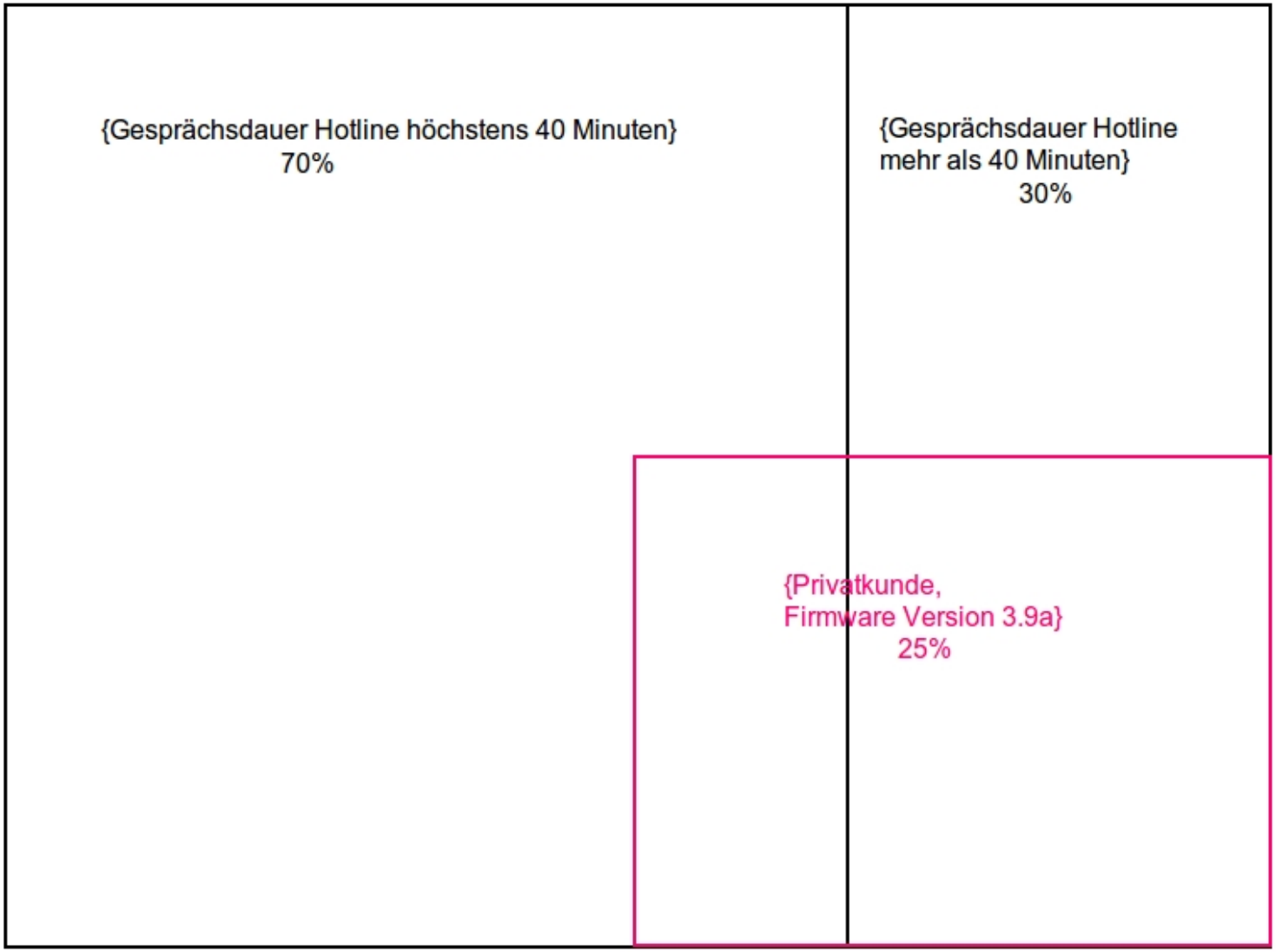
Erweiterung auf allgemeinere kategorielle Variablen

- Keine Zerlegung des gesamten Datensatzes wie bei der Cluster-Analyse, sondern suchen von möglichst grossen Teilgruppen, die sich überschneiden dürfen.
- Ein Request kann zu keiner, einer oder mehreren der gefundenen Gruppen gehören
- Nur möglich: Verknüpfung von Eigenschaften mit `UND`, nicht Verknüpfung mit `NICHT` und `ODER` (können notfalls aber mit Tricks codiert werden).

Erweiterung auf allgemeinere kategorielle Variablen



Erweiterung auf allgemeinere kategorielle Variablen



Erweiterung auf allgemeinere kategorielle Variablen

Beispiel:

$$\begin{aligned}
&\text{supp}(\{\text{Hotline} > 40\text{ Min.}\}) && = 0.40\\
&\text{supp}(\{\text{Privatkunde, Firmware V3.9a}\}) && = 0.25\\
&\text{supp}(\{\text{Privatkunde, Firmware V3.9a, Hotline} > 40\text{ Min.}\}) && = 0.15
\end{aligned}$$

Regel $\{\text{Priv., Fw. 3.9a}\} \Rightarrow \{\text{Hotl} > 40\}$ hat **Confidence:**

$$\begin{aligned}
&\text{conf}(\{\text{Priv., Fw. 3.9a}\} \Rightarrow \{\text{Hotl} > 40\})\\
=&\frac{\text{supp}(\{\text{Priv., Fw. V3.9a, Hotl} > 40\})}{\text{supp}(\{\text{Priv., Fw. V3.9a}\})}\\
=&\frac{0.15}{0.25} = 0.6
\end{aligned}$$

(bei 60% der Requests von Privatkunden mit Firmware V3.9a dauert das Hotline-Gespräch länger als 40 Minuten)

Erweiterung auf allgemeinere kategorielle Variablen

Regel $\{\text{Priv., Fw. 3.9a}\} \Rightarrow \{\text{Hotl} > 40\}$ hat **Lift:**

$$
\begin{aligned}
&\text{lift}(\{\text{Priv., Fw. 3.9a}\} \Rightarrow \{\text{Hotl} > 40\}) \\
&= \frac{\text{conf}(\{\text{Priv., Fw. 3.9a}\} \Rightarrow \{\text{Hotl} > 40\})}{\text{supp}(\{\text{Hotline} > 40 \text{ Min.}\})} \\
&= \frac{0.6}{0.3} \\
&= 2
\end{aligned}
$$

Der Anteil der Requests mit Hotlinegespräch länger als 40 Min. in der durch die Regel identifizierten Gruppe ist doppelt so gross wie über alle Requests gesehen.

Die Regel bringt also eine "Verbesserung" um einen Faktor 2.

Variation: Anwendung auf mit Kosten bewertertete Requests

Mit einigen Modifikationen einfach möglich: Analyse nicht nach Häufigkeit, sondern Kosten:

- Berechne für jeden Request (grob) Kosten
- Generiere Frequent Itemsets
- Berechne für jedes Itemset Kosten aller Requests in dieser Gruppe, Anteil an Gesamtkosten, Kosten pro Request und relative Kosten pro Request
- Filtriere Itemsets nach relevanten Kennzahlen, z.B. nur solche mit mindestens 10% Anteil an Gesamtkosten

Nachteil: Ursprüngliche Konstruktion der Itemsets nach Häufigkeit, nicht nach Kosten. Gewichte sollten eigentlich einfach in `apriori` zu implementieren sein!

Bemerkungen und Fazit

Unterschiede gegenüber klassischen Methoden:

- Keine Segmentierung der Daten, Beobachtung kann zu keiner, einer oder mehreren Gruppen gehören
- Fischen im Trüben: geeignet bei vielen Variablen ohne eindeutige Response
- Ziel nicht Vorhersage oder Modell für jeden möglichen Fall, sondern Identifikation relevanter Gruppen
- Algorithmisch definiert, Hauptziel: Kombinatorische Explosion in den Griff zu bekommen
- Unklar: Statistische Eigenschaften (massives multiples "Testen" bzw. überhaupt kein Testen; Lift und Confidence bei unterschiedlich grossen Gruppen kaum vergleichbar)
- Kann nur der Hypothesenbildung, nicht der Hypothesenüberprüfung dienen

Zum Weiterlesen:

Hastie, Tibshirani, Friedman (2009), *Elements of Statistical Learning*, 2nd ed., Springer [Kap. 14.2]

Hahsler, Grün, Hornik, Buchta (2014), *Introduction to arules - A computational environment for mining association rules and frequent item sets*, Vignette zu R-Paket `arules`

Han, Kamber, Pei (2012), *Data Mining*, 3rd ed, Morgan Kaufmann, Amsterdam [Kap. 6 u. 7]