Homework 2 Question 3

Bejan Sadeghian

August 18, 2015

Task

Evaluate the grocery.txt dataset to find interesting item set discoveries.

My Method

I used the read.transactions function to import the data as a basket. I remove duplicates in each basket if there are any (just incase). I then create the rules using the apriori function.

My first set of rules uses the parameters support = 0.001, confidence=0.4, maxlen=10, and target='rules'. The reason why is because since I know there are ~10,000 transactions/baskets in the dataset, I want any itemset that appears at least 10 times to gather as much itemsets as possible and show any associated itemsets (RHS) that appear atleast 40% of the time. I set maxlen=10 to make large baskets which Ill change later.

- My result from this first set of parameters game me some interesting itemset relationships, things I would expect like…
  - {ham, processed cheese} => {white bread} (have to make sandwiches with bread!)
  - {baking powder, flour} => {sugar} (for bakers)
- What is a concern, is that we seem to have a lot of items that are confidence = 1.0 That would be great because it's saying that the RHS itemset is always bought with the LHS itemset, however lift is not very high indicating that the RHS itemset is purchased pretty frequently regardless
  - For example, with confidence = 1.0 and lift = 5.0, we are saying that 20% of the time our RHS itemset is purchased in the total transaction set.

Because of the high confidence I made some adjustments to my rule parameters, going to support = 0.001, confidence = 0.1, and maxlen = 2. I reduced the maxlen because I believe it was forcing some higher confidence numbers, and I adjusted my min confidence level accordingly.

- My result from this set of parameters gave me again some interesting itemsets, I can see that many meat purchases (beef, chicken, eggs) are purchased alongside a root vegetable and that yogurts are typically bought with a fruit or other dairy (which is to be expected).

Finally, I found a pretty neat library called 'arulesViz' and plotted the top 20 lift items for my second rule parameter set. It shows some of what I mentioned in my last bullet point, yogurt items tend to be purchased with other dairy products or fruits and root vegetables tend to be purchased with meat items like beef, chicken, and eggs.

```
library(arules)
```

```
## Loading required package: Matrix
##
## Attaching package: 'arules'
##
## The following objects are masked from 'package:base':
##
##     %in%, write
```

```
library(arulesViz)
```

```
## Loading required package: grid
##
## Attaching package: 'arulesViz'
##
## The following object is masked from 'package:base':
##
##     abbreviate
```

```
groceries = read.transactions('groceries.txt', format='basket', sep=',', rm.duplicates =
TRUE)

#Setup the rules
#I chose support = 0.001 because since we have around 10,000 transactions, if an item is
in approx. 10+ of the baskets I
#wanted to say that is an item worth looking at
#I chose maxlen = 10 to see the impact of a large itemset
#I chose confidence = 0.1 because I wanted the results that showed item sets were in atle
ast 10% of my LHS itemset
grocery.rules = apriori(groceries, parameter=list(support=0.001, confidence=0.4, maxlen=1
0, target='rules'))
```

```
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport support minlen maxlen
##         0.4    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 ... [8955 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(grocery.rules, by='lift'))) #I want to see what itemsets have a high fr
equency of being purchased with another itemset and primarily with that itemsets (in othe
r words not purchased without the first itemset)
```

```
##   lhs                        rhs                  support confidence     lift
## 1 {bottled beer,
##    liquor}                => {red/blush wine} 0.001931876  0.4130435 21.49356
## 2 {Instant food products,
##    soda}                  => {hamburger meat} 0.001220132  0.6315789 18.99565
## 3 {processed cheese,
##    white bread}           => {ham}            0.001931876  0.4634146 17.80345
## 4 {popcorn,
##    soda}                  => {salty snack}    0.001220132  0.6315789 16.69779
## 5 {baking powder,
##    flour}                 => {sugar}          0.001016777  0.5555556 16.40807
## 6 {ham,
##    processed cheese}      => {white bread}    0.001931876  0.6333333 15.04549
```

```
inspect(head(sort(subset(grocery.rules, subset=confidence == 1.0), by='lift'))) #Want to
see what RHS itemsets are purchased every time with a LHS set
```

```
##   lhs                        rhs                    support confidence     lift
## 1 {citrus fruit,
##    root vegetables,
##    soft cheese}          => {other vegetables} 0.001016777          1 5.168156
## 2 {brown bread,
##    pip fruit,
##    whipped/sour cream}   => {other vegetables} 0.001118454          1 5.168156
## 3 {grapes,
##    tropical fruit,
##    whole milk,
##    yogurt}               => {other vegetables} 0.001016777          1 5.168156
## 4 {ham,
##    pip fruit,
##    tropical fruit,
##    yogurt}               => {other vegetables} 0.001016777          1 5.168156
## 5 {ham,
##    pip fruit,
##    tropical fruit,
##    whole milk}           => {other vegetables} 0.001118454          1 5.168156
## 6 {butter,
##    fruit/vegetable juice,
##    tropical fruit,
##    whipped/sour cream}   => {other vegetables} 0.001016777          1 5.168156
```

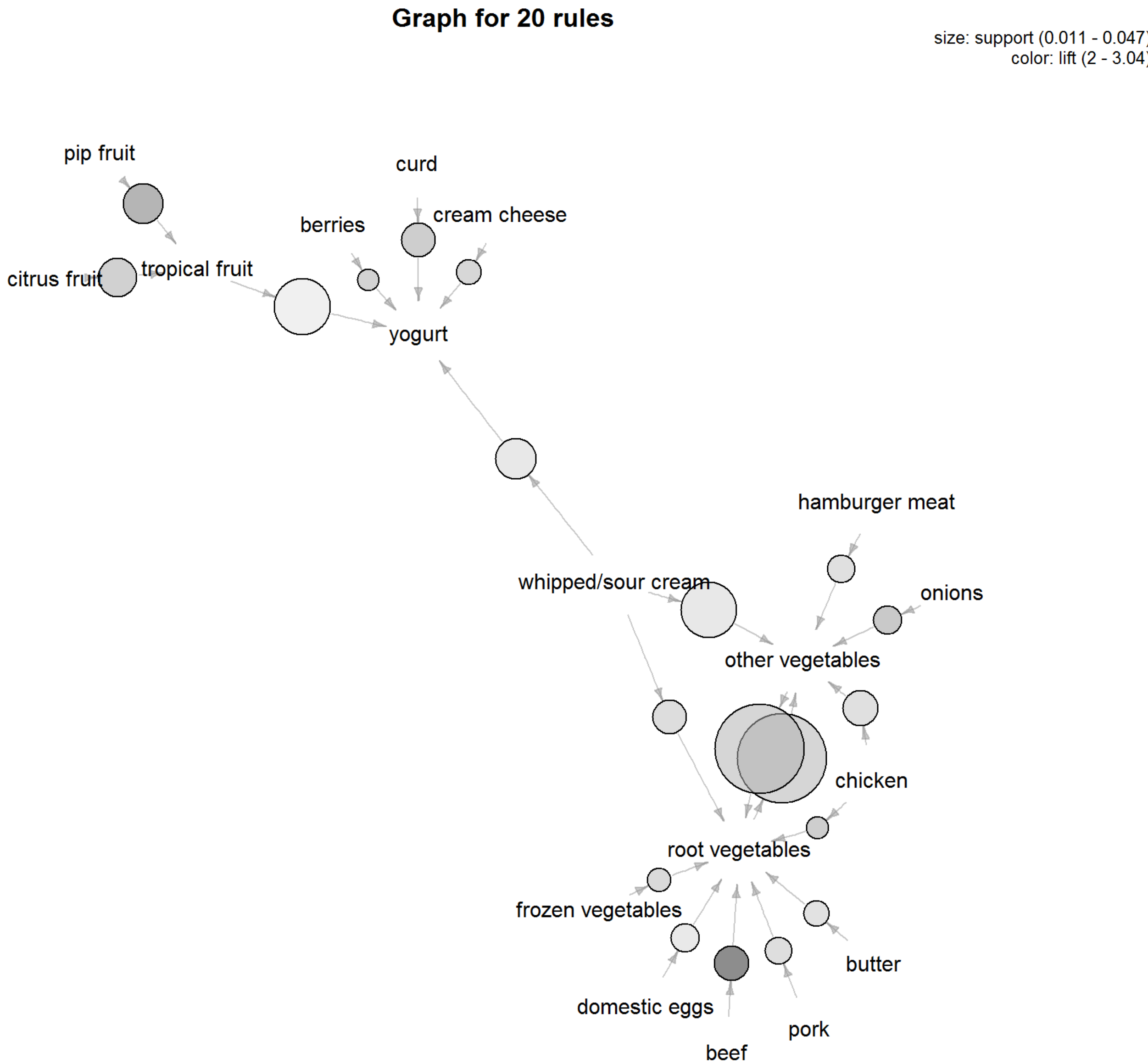
```
grocery.rules = apriori(groceries, parameter=list(support=0.001, confidence=0.1, maxle
n=2, target='rules')) # Because I was getting confidence of 1.0 I feel like that the maxl
en rule is letting things get too general. In other words, the max rule is just too large
and skewing my results
```

```
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport support minlen maxlen
##         0.1    0.1    1 none FALSE            TRUE   0.001      1      2
##  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 done [0.00s].
## writing ... [2129 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(subset(grocery.rules, subset=support > 0.01 & confidence > 0.2), by='li
ft'),25)) #want to see what 10% of our itemsets involve and what other itemsets have a hi
gh frequency in these
```

```
##    lhs                    rhs                  support confidence     lift
## 1  {beef}              => {root vegetables}  0.01738688  0.3313953 3.040367
## 2  {pip fruit}         => {tropical fruit}   0.02043721  0.2701613 2.574648
## 3  {onions}            => {other vegetables} 0.01423488  0.4590164 2.372268
## 4  {chicken}           => {root vegetables}  0.01087951  0.2535545 2.326221
## 5  {curd}              => {yogurt}           0.01728521  0.3244275 2.325615
## 6  {citrus fruit}      => {tropical fruit}   0.01992883  0.2407862 2.294702
## 7  {berries}           => {yogurt}           0.01057448  0.3180428 2.279848
## 8  {root vegetables}   => {other vegetables} 0.04738180  0.4347015 2.246605
## 9  {other vegetables}  => {root vegetables}  0.04738180  0.2448765 2.246605
## 10 {cream cheese }     => {yogurt}           0.01240468  0.3128205 2.242412
## 11 {frozen vegetables} => {root vegetables}  0.01159126  0.2410148 2.211176
## 12 {whipped/sour cream} => {root vegetables}  0.01708185  0.2382979 2.186250
## 13 {pork}              => {root vegetables}  0.01362481  0.2363316 2.168210
## 14 {chicken}           => {other vegetables} 0.01789527  0.4170616 2.155439
## 15 {hamburger meat}    => {other vegetables} 0.01382816  0.4159021 2.149447
## 16 {butter}            => {root vegetables}  0.01291307  0.2330275 2.137897
## 17 {whipped/sour cream} => {other vegetables} 0.02887646  0.4028369 2.081924
## 18 {whipped/sour cream} => {yogurt}           0.02074225  0.2893617 2.074251
## 19 {domestic eggs}     => {root vegetables}  0.01433655  0.2259615 2.073071
## 20 {tropical fruit}    => {yogurt}           0.02928317  0.2790698 2.000475
## 21 {yogurt}            => {tropical fruit}   0.02928317  0.2099125 2.000475
## 22 {citrus fruit}      => {root vegetables}  0.01769192  0.2137592 1.961121
## 23 {butter}            => {whole milk}       0.02755465  0.4972477 1.946053
## 24 {beef}              => {other vegetables} 0.01972547  0.3759690 1.943066
## 25 {pork}              => {other vegetables} 0.02165735  0.3756614 1.941476
```

```
plot(head(sort(subset(grocery.rules, subset=support > 0.01 & confidence > 0.2), by='lif
t'),20),method="graph",interactive=FALSE)
```


Graph for 20 rules
size: support (0.011 - 0.047)
color: lift (2 - 3.04)
pip fruit
curd
cream cheese
berries
citrus fruit
tropical fruit
yogurt
hamburger meat
whipped/sour cream
onions
other vegetables
chicken
root vegetables
frozen vegetables
butter
domestic eggs
pork
beef