INF0613 – Aprendizado de Máquina Não Supervisionado

Trabalho 1 - Regras de Associação

Daniel Noriaki Kurosawa Eric Uyemura Suda Fernando Shigeru Wakabayashi

Neste primeiro trabalho vamos minerar Regras de Associação em uma base de dados que contém as vendas de uma padaria. A base de dados está disponível na página da disciplina no Moodle (arquivo `bakery.csv`).

Atividade 0 – Configurando o ambiente

Antes de começar a implementação do seu trabalho configure o *workspace* e importe todos os pacotes:

```
# Adicione os demais pacotes usados
# Bibliotecas usadas neste trabalho:
library(arules)

# Configurando ambiente de trabalho:
```

Atividade 1 – Análise Exploratória da Base de Dados (*3,0 pts*)

Dado um caminho para uma base de dados, leia as transações e faça uma análise Exploratória sobre elas. Use as funções `summary`, `inspect` e `itemFrequencyPlot`. Na função `inspect` limite sua análise às 10 primeiras transações e na função `itemFrequencyPlot` gere um gráfico com a frequência relativa dos 30 itens mais frequentes.

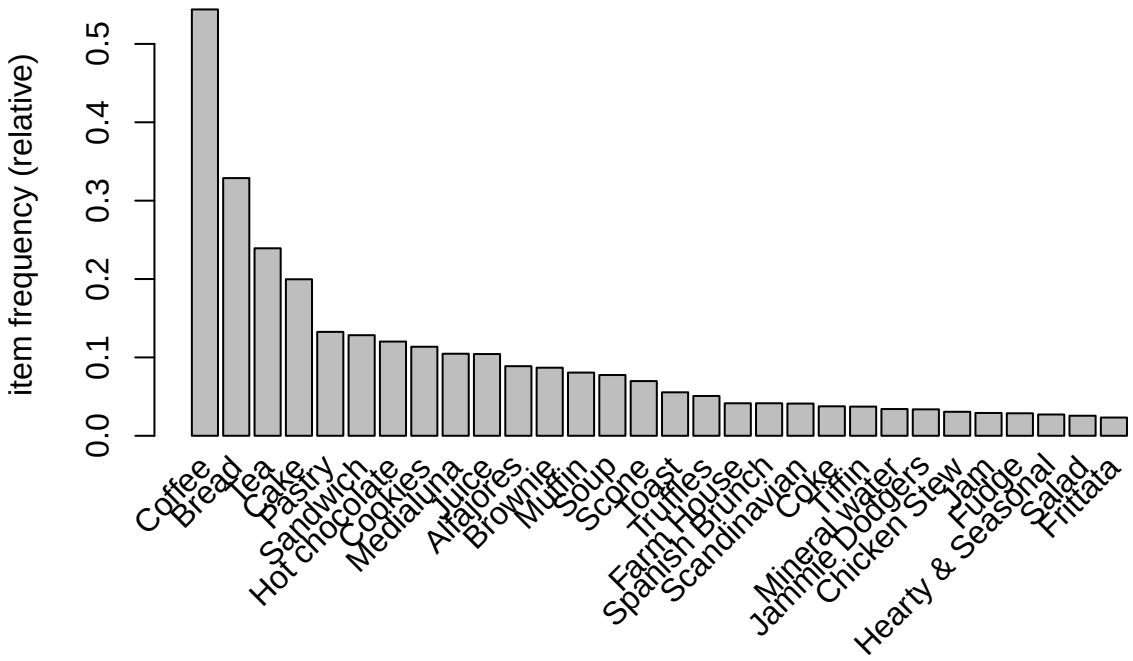
```
# Ler transações
transactions <- read.transactions("bakery.csv", format="basket", sep=",")

# Visualizando transações
inspect(transactions[1:10])
```

```
##      items
## [1]  {Coffee,Vegan mincepie}
## [2]  {Farm House,Muffin,Tea}
## [3]  {Bread,Ellas Kitchen Pouches,Jam,Juice,Muffin}
## [4]  {Bread,Juice,Salad,Sandwich}
## [5]  {Cake,Coffee,Sandwich,Smoothies,Soup}
## [6]  {Bread,Medialuna}
## [7]  {Chocolates,Coffee,Tea}
## [8]  {Alfajores,Brownie,Medialuna}
## [9]  {Alfajores,Coffee,Fudge}
## [10] {Bread,Pastry}
```

```r
# Sumário da base
summary(transactions)
```

```
## transactions as itemMatrix in sparse format with
##  2579 rows (elements/itemsets/transactions) and
##  91 columns (items) and a density of 0.0352
##
## most frequent items:
##  Coffee   Bread     Tea    Cake  Pastry (Other)
##    1403     848     617     515     342    4532
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10
##   20  664 1041  591  189   52   15    4    2    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     2.0     3.0     3.2     4.0    10.0
##
## includes extended item information - examples:
##                     labels
## 1 Afternoon with the baker
## 2               Alfajores
## 3          Argentina Night
```

```r
# Analisando a frequência dos itens
itemFrequencyPlot(transactions,topN = 30)
```



Análise

a) Descreva a base de dados discutindo os resultados das funções acima.

Resposta: Para este conjunto de dados há 2.579 transações sendo que existem 91 objetos únicos de compra registrados. Em média há 3,2 ítens sendo comprados por transação e uma distribuição da seguinte forma, primeiro quartil igual a 2 ítens, mediana igual a 3, terceiro quartil igual a 4 e máxmo de ítens igual a 10. Os ítens mais comprados individualmente fazem sentido já que as transações são de uma bakery, logo há no top 5 café, pão, chá, bolo e doces.

b) Ao gerarmos o gráfico de frequências, temos uma representação visual de uma informação já presente no resultado da função `summary`. Contudo, esse gráfico nos dá uma visão mais ampla da base. Assim podemos ver a frequência de outros itens em relação aos 10 mais frequentes. Quais informações podemos obter a partir desse gráfico (e da análise anterior) para nos ajudar na extração de regras de associação com o algoritmo `apriori`? Isto é, como a frequência dos itens pode afetar os parâmetros de configuração do algoritmo `apriori`?

Resposta: A frequencia dos ítens afeta o algoritmo pois quanto mais ítens distintos, mais combinações podem ser realizadas na hora de se gerar as regras, deve-se atentar aos parâmetros de suporte e confiança mínimos para se adequarem aos dados.

Atividade 2 – Minerando Regras (*3,5 pts*)

Use o algoritmo `apriori` para minerar regras na base de dados fornecida. Experimente com pelo menos *3 conjuntos* de valores diferentes de suporte e confiança para encontrar regras de associação. Imprima as cinco regras com o maior suporte de cada conjunto escolhido. Lembre-se de usar seu conhecimento sobre a base, obtido na questão anterior, para a escolha dos valores de suporte e confiança.

```
# Conjunto 1: suporte = 0.1   e confiança = 0.3
regras <- apriori(transactions, parameter=list(supp=0.1, conf=0.1))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5     0.1      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 257
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[91 item(s), 2579 transaction(s)] done [0.00s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [16 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
regras <- sort(regras, decreasing=TRUE, by="support")
inspect(regras[1:5])
```

```
##     lhs        rhs      support confidence coverage lift  count
## [1] {}      => {Coffee} 0.544   0.544      1.000    1.000 1403
## [2] {}      => {Bread}  0.329   0.329      1.000    1.000  848
## [3] {}      => {Tea}    0.239   0.239      1.000    1.000  617
## [4] {}      => {Cake}   0.200   0.200      1.000    1.000  515
## [5] {Bread} => {Coffee} 0.154   0.468      0.329    0.861  397
```

```
# Conjunto 2: suporte =  0.01  e confiança = 0.3
regras <- apriori(transactions, parameter=list(supp=0.01, conf=0.1))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##      10  rules TRUE
```

```
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 25
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[91 item(s), 2579 transaction(s)] done [0.00s].
## sorting and recoding items ... [39 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [204 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
regras <- sort(regras, decreasing=TRUE, by="support")
inspect(regras[1:5])
```

```
##     lhs        rhs      support confidence coverage lift  count
## [1] {}      => {Coffee} 0.544   0.544      1.000    1.000 1403
## [2] {}      => {Bread}  0.329   0.329      1.000    1.000  848
## [3] {}      => {Tea}    0.239   0.239      1.000    1.000  617
## [4] {}      => {Cake}   0.200   0.200      1.000    1.000  515
## [5] {Bread} => {Coffee} 0.154   0.468      0.329    0.861  397
```

```
# Conjunto 3: suporte =  0.000001  e confiança = 0.1
regras <- apriori(transactions, parameter=list(supp=0.000001, conf=0.9))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5   1e-06      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 0
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[91 item(s), 2579 transaction(s)] done [0.00s].
## sorting and recoding items ... [91 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.00s].
## writing ... [16650 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
regras <- sort(regras, decreasing=TRUE, by="support")
inspect(regras[1:5])
```

```
##     lhs                                       rhs      support confidence
## [1] {Extra Salami or Feta}                 => {Coffee} 0.00698 0.9
## [2] {Cake,Hearty & Seasonal}               => {Coffee} 0.00271 1.0
## [3] {Bread,Extra Salami or Feta}           => {Salad}  0.00233 1.0
## [4] {Farm House,Toast}                     => {Coffee} 0.00233 1.0
## [5] {Extra Salami or Feta,Spanish Brunch} => {Coffee} 0.00194 1.0
##     coverage lift  count
## [1] 0.00775   1.65 18
## [2] 0.00271   1.84  7
## [3] 0.00233  39.08  6
## [4] 0.00233   1.84  6
## [5] 0.00194   1.84  5
```

Análises

a) Quais as regras mais interessantes geradas a partir dessa base? Justifique.

Resposta: Ao testar os parametros, percebemos que devido a lista ser ordernada à partir dos maiores suportes, variar o parametro suporte mínimo não faz diferença, para uma mesma confiança. No terceiro conjunto, recuperamos as regras {Extra Salami or Feta} => {Coffee} com confiança 0.9 e Extra Salami or Feta,Spanish Brunch} => {Coffee} com confiança 1.0, indicando que 10 pontos percentuais a mais de pessoas preferem comprar Extra Salami or Feta juntamente com Spanish Brunch do que somente Extra Salami or Feta. Além disso, percebemos que {Bread,Extra Salami or Feta} => {Salad} tem um lift de 39.08 o que pode indicar uma causalidade forte, e confiança = 1.0, ou seja, todos os pedidos contendo o lado esquerdo, compraram também o lado direito.

Atividade 3 – Medidas de Interesse (*3,5 pts*)

Vimos na aula que, mesmo após as podas do algoritmo `apriori`, ainda temos algumas regras com características indesejáveis como redundâncias e dependência estatística negativa. Também vimos algumas medidas que nos ajudam a analisar melhor essas regras como o lift, a convicção e a razão de chances. Nesta questão, escolha um dos conjuntos de regras geradas na atividade anterior e o analise usando essas medidas. Compute as três medidas para o conjunto escolhido com a função `interestMeasure` e experimente ordenar as regras com cada uma das novas medidas.

```
# Usando Conjunto 02

# Compute as medidas de interesse
medidas <- interestMeasure(regras,
                           c("confidence", "conviction",
                             "coverage", "support",
                             "oddsRatio", "leverage"), transactions)


regras@quality$index <- as.numeric(row.names(regras@quality))
regras@quality<-cbind(regras@quality,medidas$oddsRatio,medidas$conviction)
# Apresente as regras ordenadas por lift
lift<-inspect(head(sort(regras, by="lift")))
```

```
##     lhs                       rhs              support confidence coverage lift count index medidas$
## [1] {Frittata,
```

```
##      Scandinavian,
##      Tea}                  => {Chicken sand}   0.000388          1 0.000388 2579     1   792
## [2] {Coffee,
##      Frittata,
##      Scandinavian,
##      Tea}                  => {Chicken sand}   0.000388          1 0.000388 2579     1  4836
## [3] {My-5 Fruit Shoot,
##      Tea}                  => {Hack the stack} 0.000388          1 0.000388 1290     1    38
## [4] {Extra Salami or Feta,
##      Farm House}           => {Bare Popcorn}   0.000388          1 0.000388 1290     1    51
## [5] {Art Tray,
##      My-5 Fruit Shoot,
##      Tea}                  => {Hack the stack} 0.000388          1 0.000388 1290     1   805
## [6] {Extra Salami or Feta,
##      Farm House,
##      Salad}                => {Bare Popcorn}   0.000388          1 0.000388 1290     1   809
```

```
#write.csv(lift,"./lift.csv")

# Apresente as regras ordenadas por convicção
conviction<-inspect(sort(head(regras, by="medidas$conviction")))
```

```
##     lhs                                    rhs      support confidence
## [1] {Extra Salami or Feta}              => {Coffee} 0.00698 0.9
## [2] {Cake,Hearty & Seasonal}            => {Coffee} 0.00271 1.0
## [3] {Bread,Extra Salami or Feta}        => {Salad}  0.00233 1.0
## [4] {Farm House,Toast}                  => {Coffee} 0.00233 1.0
## [5] {Extra Salami or Feta,Spanish Brunch} => {Coffee} 0.00194 1.0
## [6] {Bread,Sandwich,Spanish Brunch}     => {Coffee} 0.00194 1.0
##     coverage lift  count index medidas$oddsRatio medidas$conviction
## [1] 0.00775   1.65 18      18  7.63               4.56
## [2] 0.00271   1.84  7     732    NA                 NA
## [3] 0.00233  39.08  6     529    NA                 NA
## [4] 0.00233   1.84  6     781    NA                 NA
## [5] 0.00194   1.84  5     537    NA                 NA
## [6] 0.00194   1.84  5    4595    NA                 NA
```

```
#write.csv(conviction,"./conviction.csv")

# Apresente as regras ordenadas por razão de chances
odds_ratio<-inspect(sort(head(regras, by="medidas$oddsRatio")))
```

```
##     lhs                                    rhs      support confidence
## [1] {Extra Salami or Feta}              => {Coffee} 0.00698 0.9
## [2] {Cake,Hearty & Seasonal}            => {Coffee} 0.00271 1.0
## [3] {Bread,Extra Salami or Feta}        => {Salad}  0.00233 1.0
## [4] {Farm House,Toast}                  => {Coffee} 0.00233 1.0
## [5] {Extra Salami or Feta,Spanish Brunch} => {Coffee} 0.00194 1.0
## [6] {Bread,Sandwich,Spanish Brunch}     => {Coffee} 0.00194 1.0
##     coverage lift  count index medidas$oddsRatio medidas$conviction
## [1] 0.00775   1.65 18      18  7.63               4.56
## [2] 0.00271   1.84  7     732    NA                 NA
## [3] 0.00233  39.08  6     529    NA                 NA
```

```
## [4] 0.00233   1.84  6     781    NA                NA
## [5] 0.00194   1.84  5     537    NA                NA
## [6] 0.00194   1.84  5    4595    NA                NA
```

```
#write.csv(odds_ratio,"./odds_ratio.csv")
```

Análise

a) Quais as regras mais interessantes do conjunto? Justifique.

Resposta: Usando como base o Conjunto 02 (suporte = 0.01 e confiança = 0.3) ordenado por lift, percebemos que as regras {Coke} => {Sandwich} e {Coffee,Soup} => {Sandwich} possuem um lift relativamente alto, indicando uma possível causalidade e uma confiança pouco acima de 25%, ou seja, em um quarto das transações esta regra é satisfeita.