INF0613 – Aprendizado de Máquina Não Supervisionado

Trabalho 1 - Regras de Associação

Evandro Santos Rocha Laíssa Pacheco de Oliveira Rafael Dantas de Moura

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:
# setwd("~/Documentos/mdc/aprendizado_nao_supervisionado/trabalho1")
```

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.

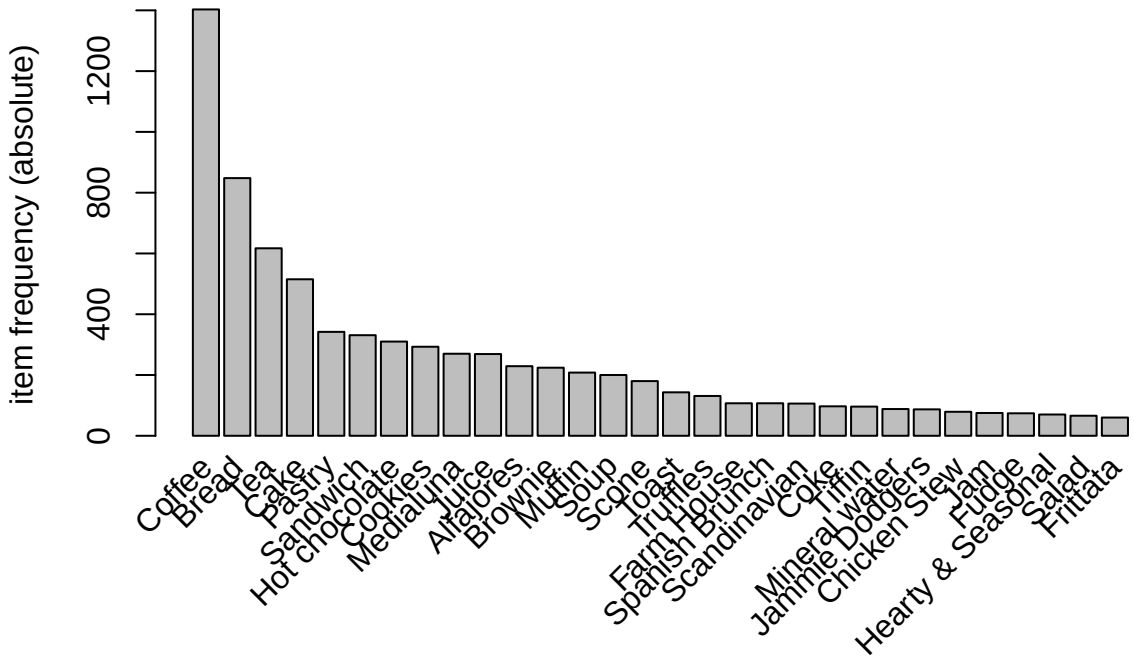
```
# Ler transações
transacoes <- read.transactions(file="bakery.csv", format="basket",
                                sep=",")

# Visualizando transações
inspect(transacoes[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(transacoes)
```

```
## 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(transacoes, topN = 30, type = "absolute")
```



Análise

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

Resposta: Pela função `summary`, acima, sabemos que a base de dados possui 2579 transações formadas por diferentes combinações de 91 itens. A mesma função mostra que os cinco itens mais frequentes nas transações são *coffe, bread, tea, cake, pastry.*

A função `inspect` retorna o valor das 10 primeiras transações, o que nos possibilita ter uma ideia da estrutura da base. Apesar de 10 transações representar um espaço amostral pequeno em relação ao número total de transações da base, neste já podemos notar que pelo menos 1 dos 5 itens mais frequentes aparece em 9 das 10 primeiras transações.

Outra ideia da estrutura da base pode ser vista na função `summary`, na parte referente à distribuição do tamanho das transações. Notamos que a maior parte das transações (1041 observações) possui 3 itens, seguido por transações de 2 itens (664 observações) e 4 itens (591 observações). A maior transação registrada possui 10 itens, e há apenas uma transação deste tamanho.

Por fim, pela função `itemFrequencyPlot` temos uma visualização dos 30 itens mais frequentes. Nota-se que a partir do 5º item há uma queda contínua da frequência, tornando nítido a importância dos 4 primeiros em relação aos demais. Se pensamos que para além dos 30 há ainda mais 61 itens, podemos ter uma ideia do quão pouco representativos os últimos itens serão, no total de transações.

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: Com uma base de dados composta por 91 itens, sabemos que o número total de conjuntos de itens possíveis é igual a 2^{91}, e o número total de regras de associação é de $3^{91} - 2^{91+1} + 1$. Dado o alto custo e complexidade de se analisar tantos conjuntos e regras, o algortimo `apriori` se torna essencial justamente por fazer uma selação prévia dos conjuntos mais frequentes, a partir de valores de “corte” (valor de suporte).

Se o valor de suporte for ajustado muito alto, pode-se perder conjuntos de itens envolvendo itens raros interessantes ou ainda selecionar um conjunto tão pequeno de itens que prejudicaria a criação de regras interessantes. Na prática, notamos que mesmo para os itens de maior frequência, apenas o *coffe* aparece em mais de 50% das transações e o quinto item mais frequente, *pastry*, está presente em apenas 13.26% das transações. Como sabemos que o valor suporte de um conjunto de itens nunca é maior do que o suporte de seus subconjuntos (propriedade anti-monotônica do suporte), caso adotemos um valor alto de suporte, nem os conjuntos formados por itens mais frequentes passarão na validação da regra.

Assim, por temos uma base com muitos itens e sendo a maior parte deles pouco frequentes, para poder incluir alguns deles nas análises, será necessário testar valores baixos para o parâmetro suporte. Além disso, ao optarmos por um valor de suporte menor, temos mais chances de encontrar regras interessantes, com maior valor de confiança, para os itens menos frequentes.

Por outro lado, se for atribuído um valor muito baixo para o parâmetro valor de suporte, isso resultaria em muitos conjuntos de itens frequentes, o que pode aumentar o número de candidatos e a dimensão máxima dos conjuntos de itens frequentes, o que poderia comprometer a confiabilidade da conclusão da regra.

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 = 0.15%   e confiança = 90%
regras_1 <- apriori(transacoes, parameter =
                    list(supp = 0.0015, conf = 0.9))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5  0.0015      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: 3
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[91 item(s), 2579 transaction(s)] done [0.00s].
```

```
## sorting and recoding items ... [67 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [20 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
regras_1 <- sort(regras_1, by="support", decreasing=TRUE)

inspect(regras_1[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
```

```
# Conjunto 2: suporte = 0.7%   e confiança = 65%
regras_2 <- apriori(transacoes, parameter =
                    list(supp = 0.007, conf = 0.65))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.65    0.1    1 none FALSE            TRUE       5   0.007      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: 18
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[91 item(s), 2579 transaction(s)] done [0.00s].
## sorting and recoding items ... [40 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [5 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
regras_2 <- sort(regras_2, by="support", decreasing=TRUE)

inspect(regras_2[1:5])
```

```
##     lhs                       rhs      support confidence coverage lift count
## [1] {Toast}                => {Coffee} 0.03994 0.720      0.0554   1.32 103
## [2] {Salad}                => {Coffee} 0.01745 0.682      0.0256   1.25  45
## [3] {Cake,Sandwich}        => {Coffee} 0.01435 0.685      0.0209   1.26  37
## [4] {Hot chocolate,Pastry} => {Coffee} 0.01086 0.667      0.0163   1.23  28
## [5] {Keeping It Local}     => {Coffee} 0.00969 0.781      0.0124   1.44  25
```

```
# Conjunto 3: suporte = 1%   e confiança = 60%
regras_3 <- apriori(transacoes, parameter =
                    list(supp = 0.01, conf = 0.6))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    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 ... [10 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
regras_3 <- sort(regras_3, by="support", decreasing=TRUE)

inspect(regras_3[1:5])
```

```
##     lhs                     rhs      support confidence coverage lift count
## [1] {Toast}              => {Coffee} 0.0399  0.720      0.0554   1.32 103
## [2] {Spanish Brunch}     => {Coffee} 0.0252  0.607      0.0415   1.12  65
## [3] {Salad}              => {Coffee} 0.0174  0.682      0.0256   1.25  45
## [4] {Cake,Hot chocolate} => {Coffee} 0.0167  0.606      0.0275   1.11  43
## [5] {Hearty & Seasonal}  => {Coffee} 0.0163  0.600      0.0271   1.10  42
```

```
# Conjunto 4: suporte = 0.5%   e confiança = 70%
regras_4 <- apriori(transacoes, parameter =
                    list(supp = 0.005, conf = 0.7))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
```

```
##         0.7    0.1    1 none FALSE            TRUE       5   0.005      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: 12
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[91 item(s), 2579 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 4 done [0.00s].
## writing ... [10 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
regras_4 <- sort(regras_4, by="support", decreasing=TRUE)

inspect(regras_4[1:5])
```

```
##     lhs                           rhs      support confidence coverage lift
## [1] {Toast}                    => {Coffee} 0.03994 0.720      0.05545   1.32
## [2] {Keeping It Local}         => {Coffee} 0.00969 0.781      0.01241   1.44
## [3] {Extra Salami or Feta}     => {Coffee} 0.00698 0.900      0.00775   1.65
## [4] {Extra Salami or Feta}     => {Salad}  0.00620 0.800      0.00775  31.26
## [5] {Extra Salami or Feta,Salad} => {Coffee} 0.00543 0.875      0.00620   1.61
##     count
## [1] 103
## [2]  25
## [3]  18
## [4]  16
## [5]  14
```

```
# Conjunto 5: suporte = 9%   e confiança = 30%
regras_5 <- apriori(transacoes, parameter =
                    list(supp = 0.09, conf = 0.3))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.3    0.1    1 none FALSE            TRUE       5    0.09      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: 232
##
```

```
## 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 ... [5 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
regras_5 <- sort(regras_5, by="support", decreasing=TRUE)

inspect(regras_5[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] {Bread} => {Coffee} 0.154   0.468      0.329    0.861  397
## [4] {Cake}  => {Coffee} 0.112   0.559      0.200    1.028  288
## [5] {Tea}   => {Coffee} 0.104   0.433      0.239    0.795  267
```

Análises

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

Resposta: Analisando apenas os valores de suporte e confiança, podemos dizer que as regras mais interessantes são, em ordem descrecente da frequência do item da esquerda:

{Bread} => {Coffee} : 46,81% das transações que contém *bread* também contém *coffee*

{Cake} => {Coffee} : 55,92 das transações que contém *cake* também contém *coffee*

{Tea} => {Coffee} : 43,27% das transações que contém *tea* também contém *coffee*

Todas as regras acima foram obtidas no conjunto 5, que possui alto valor de suporte.

Por outro lado, olhando apenas para o valor do lift, temos as seguintes regras como mais interessantes:

{Bread,Extra Salami or Feta} => {Salad}

{Extra Salami or Feta} => {Salad}

Esses resultados foram obtidos em conjuntos com valor de suporte menor.

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.

```
# Compute as medidas de interesse

medidas <- interestMeasure(regras_4, c("conviction", "oddsRatio", "lift"), transacoes)
```

```r
# Apresente as regras ordenadas por lift

   # solucao utilizando a regras_2
   #inspect(sort(regras_2, by="lift", decreasing=TRUE))

order_lift <- order(medidas$lift, decreasing = TRUE)
inspect(regras_4[order_lift])
```

```
##      lhs                                rhs      support confidence coverage
## [1]  {Extra Salami or Feta}          => {Salad}  0.00620 0.800      0.00775
## [2]  {Coffee,Extra Salami or Feta}   => {Salad}  0.00543 0.778      0.00698
## [3]  {Extra Salami or Feta}          => {Coffee} 0.00698 0.900      0.00775
## [4]  {Extra Salami or Feta,Salad}    => {Coffee} 0.00543 0.875      0.00620
## [5]  {Salad,Sandwich}                => {Coffee} 0.00543 0.824      0.00659
## [6]  {Keeping It Local}              => {Coffee} 0.00969 0.781      0.01241
## [7]  {Juice,Spanish Brunch}          => {Coffee} 0.00504 0.765      0.00659
## [8]  {Cookies,Scone}                 => {Coffee} 0.00504 0.765      0.00659
## [9]  {Juice,Pastry}                  => {Coffee} 0.00504 0.765      0.00659
## [10] {Toast}                         => {Coffee} 0.03994 0.720      0.05545
##      lift  count
## [1]  31.26  16
## [2]  30.39  14
## [3]   1.65  18
## [4]   1.61  14
## [5]   1.51  14
## [6]   1.44  25
## [7]   1.41  13
## [8]   1.41  13
## [9]   1.41  13
## [10]  1.32 103
```

```r
# Apresente as regras ordenadas por convicção

order_conviction <- order(medidas$conviction, decreasing = TRUE)
inspect(regras_4[order_conviction])
```

```
##      lhs                                rhs      support confidence coverage
## [1]  {Extra Salami or Feta}          => {Salad}  0.00620 0.800      0.00775
## [2]  {Extra Salami or Feta}          => {Coffee} 0.00698 0.900      0.00775
## [3]  {Coffee,Extra Salami or Feta}   => {Salad}  0.00543 0.778      0.00698
## [4]  {Extra Salami or Feta,Salad}    => {Coffee} 0.00543 0.875      0.00620
## [5]  {Salad,Sandwich}                => {Coffee} 0.00543 0.824      0.00659
## [6]  {Keeping It Local}              => {Coffee} 0.00969 0.781      0.01241
## [7]  {Juice,Spanish Brunch}          => {Coffee} 0.00504 0.765      0.00659
## [8]  {Cookies,Scone}                 => {Coffee} 0.00504 0.765      0.00659
## [9]  {Juice,Pastry}                  => {Coffee} 0.00504 0.765      0.00659
## [10] {Toast}                         => {Coffee} 0.03994 0.720      0.05545
##      lift  count
## [1]  31.26  16
## [2]   1.65  18
## [3]  30.39  14
```

```
## [4]   1.61  14
## [5]   1.51  14
## [6]   1.44  25
## [7]   1.41  13
## [8]   1.41  13
## [9]   1.41  13
## [10]  1.32 103
```

```
# Apresente as regras ordenadas por razão de chances
order_oddsRatio <- order(medidas$oddsRatio, decreasing = TRUE)
inspect(regras_4[order_oddsRatio])
```

```
##      lhs                                rhs      support confidence coverage
## [1]  {Extra Salami or Feta}          => {Salad}  0.00620 0.800      0.00775
## [2]  {Coffee,Extra Salami or Feta}   => {Salad}  0.00543 0.778      0.00698
## [3]  {Extra Salami or Feta}          => {Coffee} 0.00698 0.900      0.00775
## [4]  {Extra Salami or Feta,Salad}    => {Coffee} 0.00543 0.875      0.00620
## [5]  {Salad,Sandwich}                => {Coffee} 0.00543 0.824      0.00659
## [6]  {Keeping It Local}              => {Coffee} 0.00969 0.781      0.01241
## [7]  {Juice,Spanish Brunch}          => {Coffee} 0.00504 0.765      0.00659
## [8]  {Cookies,Scone}                 => {Coffee} 0.00504 0.765      0.00659
## [9]  {Juice,Pastry}                  => {Coffee} 0.00504 0.765      0.00659
## [10] {Toast}                         => {Coffee} 0.03994 0.720      0.05545
##      lift  count
## [1]  31.26  16
## [2]  30.39  14
## [3]   1.65  18
## [4]   1.61  14
## [5]   1.51  14
## [6]   1.44  25
## [7]   1.41  13
## [8]   1.41  13
## [9]   1.41  13
## [10]  1.32 103
```

Análise

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

Resposta:

Considerando apenas o conjunto 4, as regras mais interessantes são:

{Extra Salami or Feta} => {Salad}

{Coffee,Extra Salami or Feta} => {Salad}

Ambas regras possuem o maior valor para o lift, que mede o grau da independência entre os dois lados (antecedente e consequente) de uma regra de associação. Esse valor pode variar de 0 a infinito, e quanto mais próximo de 1, menos interessante a regra é, pois indica que os itens do lhs (left hand side) e o rhs (right hand side) são independentes. Se os dois valores são independentes, o aumento no consumo de um item não implicará, necessariamente, o consumo do outro.

Para a convicção, é esperado que regras mais interessantes apresentem valores entre 1 e 5, o que ocorre para as regras acima.

Para a razão de chances, temos que quanto maior o valor, maiores as chances do lado direito ocorrer na presença do dos itens do lado esquerdo. Ambas regras possuem os maiores valores para esta medida.

É útil ressaltar que apesar de serem as regras mais interessantes do ponto de vista da independência, essas regras ocorrem, respectivamente, em apenas 16 e 14 das 2579 transações. Sendo assim, apesar de serem as regras mais confiáveis, a baixa frequência desse conjunto poderia significar, do ponto de vista estratégico, pouco relevante para o tomador de decisões, dependendo das características do ramo de atividade.

Por fim, nota-se a diferença das melhores regras obtidas após considerar as medidas de interesse, se comparadas com as melhores regras obtidas apenas observando os valores de suporte e confiança. Essa diferença pode ser explicada pela grande quantidade de regras cujos itens são independentes, principalmente quando se privilegia altos valores de suporte, para um banco de dados cuja frequência da maior parte dos itens é baixa.