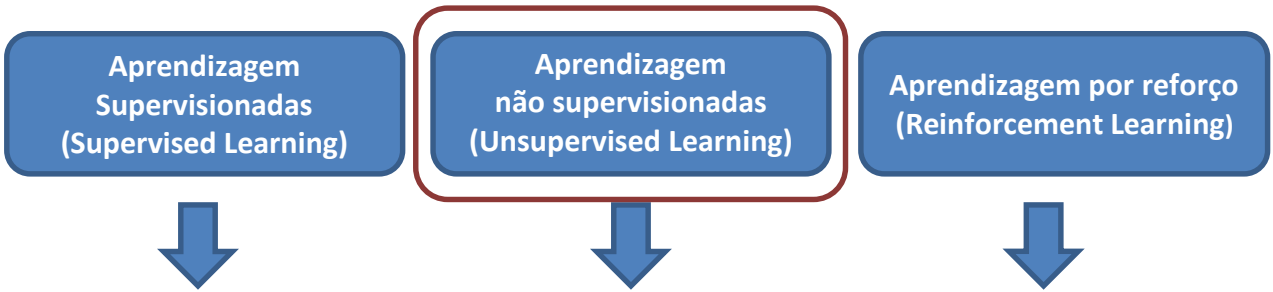
INSTITUTO FEDERAL
Santa Catarina
Câmpus São José

Aplicação de Técnicas de Aprendizagem de Máquina utilizando R

Mário de Noronha Neto e Richard Demo Souza

Alguns tipos de aprendizado de máquina

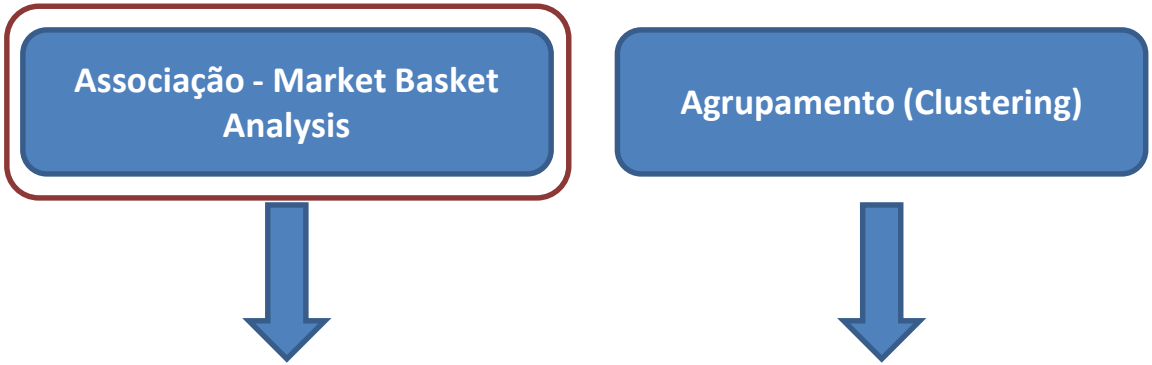


O processo de treinamento do modelo é realizado com um conjunto de dados em que as entradas e saídas são conhecidas

O processo de treinamento do modelo é realizado com um conjunto de dados em que apenas as entradas são fornecidas

O processo de treinamento é realizado a partir de recompensas nas interações com o ambiente

Técnicas de Aprendizagem não Supervisionada abordadas neste curso



Regras de Associação

As regras de associação buscam identificar padrões/regras de relacionamento existentes entre itens de um determinado conjunto de dados. Esta técnica não é utilizada para predições, mas sim para descobertas em base de dados em que apenas as entradas são fornecidas (não supervisionado).

$$\{\text{bread, peanut butter, jelly}\}$$

Conjunto de itens

Regra de associação

$$\{\text{peanut butter, jelly}\} \rightarrow \{\text{bread}\}$$

Se *peanut butter e jelly* são compradas juntas, é provável que *bread* seja comprado também

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Regras de Associação

Esta técnica pode ser aplicadas em tarefas como:

- **Market Basket Analysis**

- Encontrar padrões de compras ou reivindicações médicas que ocorrem em combinação com o uso fraudulento de cartões de crédito ou seguros.

- Identificar combinações de comportamentos que precedem cancelamentos de serviços de telefonia celular ou atualizações de pacotes de TV a cabo, por exemplo.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

O Algoritmo *Apriori* para regras de associação

- Em análises envolvendo regras de associação, o número de conjuntos de itens cresce exponencialmente com o número de características. Uma loja que vende 100 itens teria 2^100 = 1,27e+30 conjunto de itens para um algoritmo avaliar.

- Ao invés de avaliar cada um dos possíveis **conjuntos de itens**, uma regra prática seria não considerar combinações raras, com por exemplo {motor de carro, batom}.

- Ignorando conjunto de itens raros, é possível limitar o universo de possibilidades para tamanhos gerenciáveis.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

O Algoritmo *Apriori* para regras de associação

O nome *Apriori* vem do fato do algoritmo utilizar informações *a priori* do conjunto de dados disponíveis para diminuir o número de conjunto de itens a serem analisados. Todos os **subconjuntos** derivados dos **conjuntos de itens** que ocorrem com frequência devem ocorrer com frequência. Algumas características do algoritmo:

Strengths	Weaknesses
• Is capable of working with large amounts of transactional data • Results in rules that are easy to understand • Useful for "data mining" and discovering unexpected knowledge in databases	• Not very helpful for small datasets • Requires effort to separate the true insight from common sense • Easy to draw spurious conclusions from random patterns

Fonte: Brett Lantz – Machine Learning with R – Second Edition

O Algoritmo *Apriori*: Suporte e Confiança

Transaction number	Purchased items
1	*{flowers, get well card, soda}*
2	*{plush toy bear, flowers, balloons, candy bar}*
3	*{get well card, candy bar, flowers}*
4	*{plush toy bear, balloons, soda}*
5	*{flowers, get well card, soda}*

Suporte: medida da frequência com que um conjunto de itens ocorre

$$\text{support}(X) = \frac{\text{count}(X)}{N}$$

{get well card, flowers} = 3/6 = 0.6

{candy bar} = 2/5 = 0.4

Confiança: Proporção de transações em que a presença de **X** resulta na presença de **Y**

$$\text{confidence}(X \rightarrow Y) = \frac{\text{support}(X, Y)}{\text{support}(X)}$$

{flowers} -> *{get well card}* = 0.6/0.8 = *0.75*

{get well card} -> *{flowers}* = 0.6/0.6 = 1

Regra forte!

Fonte: Brett Lantz – Machine Learning with R – Second Edition

INSTITUTO FEDERAL
Santa Catarina
Câmpus São José

O Algoritmo *Apriori*: Construindo as regras

Princípio: Se *{A}* não é frequente, não deveremos considerar *{A,B}* ou qualquer outro conjunto de itens que contenha *{A}*. De forma geral, o processo de criação das regras ocorrem em duas fases:

1. Identificar todos os conjuntos de itens que atendam um limite mínimo de ***suporte***

2. Criar regras a partir destes conjuntos de itens usando aqueles que atendam um limite mínimo de ***confiança***

Fonte: Brett Lantz – Machine Learning with R – Second Edition

O Algoritmo *Apriori*: Exemplo

A primeira fase ocorre em múltiplas iterações. Cada iteração sucessiva envolve a avaliação do suporte para conjuntos de itens cada vez maiores. Para o exemplo a seguir, vamos considerar os itens {A}, {B}, {C} e {D}.

Primeira iteração: {A}, {B} e {C} ocorrem com frequência e {D} com pouca frequência. {D} é eliminado.

Segunda iteração: São considerados {A,B}, {A,C} e {B,C}. Destes novos conjuntos, {A,C} não atende o limite mínimo de suporte e é eliminado.

Terceira iteração: O princípio *Apriori* diz que {A,B,C} não pode ser frequente, já que o subconjunto {A,C} não é. Portanto, o algoritmo não gera um novo conjunto de itens e para nesta iteração.

Na segunda fase são geradas as regras com base nos conjuntos de itens que ocorreram com maior frequência. Por exemplo, {A} -> {B} e {B} -> {A}. Estas regras terão seu grau de confiança calculado e qualquer regra que não atenda o nível desejado será eliminada.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Exemplo: Identificando mantimentos comprados frequentemente

Passo 1: Coleta de dados

Dataset utilizado:

O *dataset* utilizado neste exemplo é composto por dados de compras coletados por um mês em um mercado. O *dataset* contém 9835 transações, uma média de 327 transações por dia, e foi adaptado do *dataset Groceries* do pacote *arule* do R. Para este exemplo, todas as marcas de pão, leite, detergente, etc, foram retiradas. Isto reduziu o número de itens para 169 tipos.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

INSTITUTO FEDERAL
Santa Catarina
Câmpus
São José

Passo 2: Explorando e preparando os dados

```
groceries <- read.csv("groceries.csv")
```

	V1	V2	V3	V4
1	citrus fruit	semi-finished bread	margarine	ready soups
2	tropical fruit	yogurt	coffee	
3	whole milk			
4	pip fruit	yogurt	cream cheese	meat spreads
5	other vegetables	whole milk	condensed milk	long life bakery product

Neste caso, a função *read.csv* importará o arquivo com quatro colunas em função da primeira compra ter apenas 4 itens, deixando o arquivo em um formato não apropriado para este exemplo. Como o número de itens varia em cada compra, uma solução é utilizar uma matriz esparsa para representar esses dados. A função *read.transactions()* do pacote *arules* funciona de forma similar ao comando *read.csv* com a diferença de que o resultado é apresentado em uma matriz esparsa. O comando *sep = ","* indica que os itens são separados por vírgula.

```
> groceries <- read.transactions("groceries.csv", sep = ",")
```

Fonte: Brett Lantz – Machine Learning with R – Second Edition

INSTITUTO FEDERAL
Santa Catarina
Câmpus São José

Passo 2: Explorando e preparando os dados

```
> summary(groceries)
transactions as itemMatrix in sparse format with
 9835 rows (elements/itemsets/transactions) and
 169 columns (items) and a density of 0.02609146

most frequent items:
      whole milk other vegetables       rolls/buns
            2513             1903             1809
            soda           yogurt          (Other)
            1715             1372            34055
```

O item *whole milk* aparece em 2513 compras (aproximadamente em 2513/9835 = 0.2555 = 25%)

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 2: Explorando e preparando os dados

```
element (itemset/transaction) length distribution:
sizes
   1    2    3    4    5    6    7    8    9   10   11   12
2159 1643 1299 1005  855  645  545  438  350  246  182  117
  13   14   15   16   17   18   19   20   21   22   23   24
  78   77   55   46   29   14   14    9   11    4    6    1
  26   27   28   29   32
   1    1    1    3    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   4.409   6.000  32.000
```

Observe que 32 que apenas uma compra/transação tem 32 itens e que 25% das compras possuem 2 ou menos itens.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

INSTITUTO FEDERAL
Santa Catarina
Câmpus
São José

Passo 2: Explorando e preparando os dados

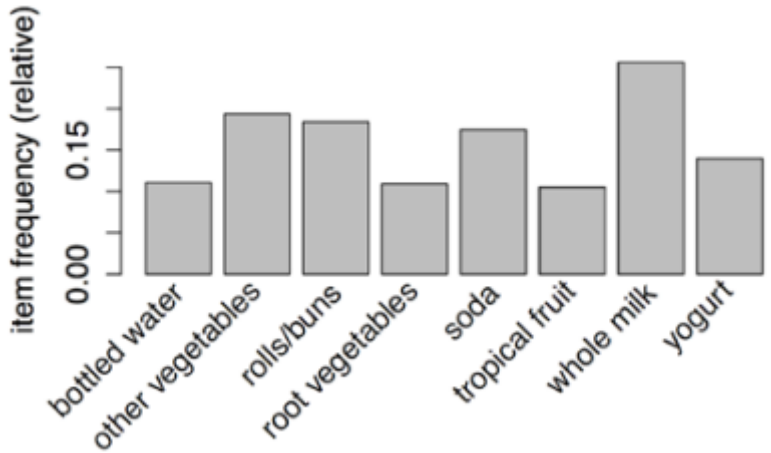
```
> inspect(groceries[1:5])
  items
1 {citrus fruit,
   margarine,
   ready soups,
   semi-finished bread}
2 {coffee,
   tropical fruit,
   yogurt}
3 {whole milk}
4 {cream cheese,
   meat spreads,
   pip fruit,
   yogurt}
5 {condensed milk,
   long life bakery product,
   other vegetables,
   whole milk}
```

```
> itemFrequency(groceries[, 1:3])
abrasive cleaner artif. sweetener   baby cosmetics
    0.0035587189     0.0032536858     0.0006100661
```

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 2: Explorando e preparando os dados

```
> itemFrequencyPlot(groceries, support = 0.1)
```

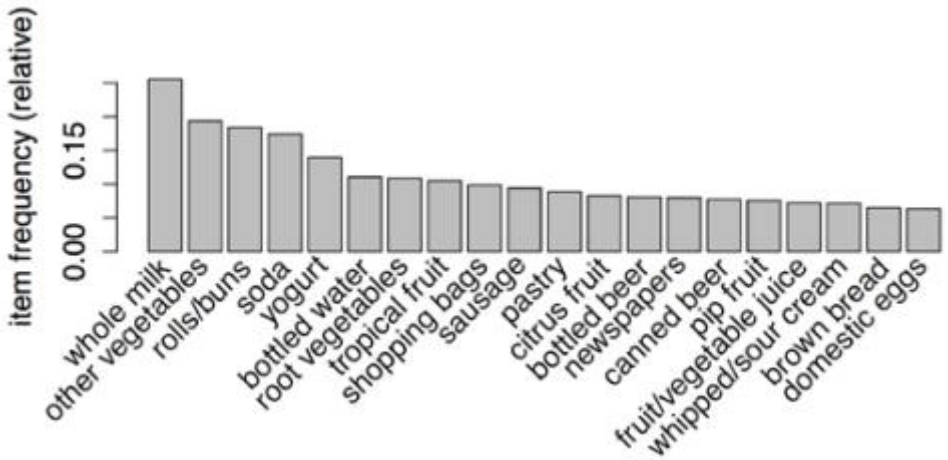


Apresenta um histograma com os itens que possuem um **suporte** de pelo menos 10%

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 2: Explorando e preparando os dados

```
> itemFrequencyPlot(groceries, topN = 20)
```



Apresenta um histograma com os 20 itens mais frequentes

Fonte: Brett Lantz – Machine Learning with R – Second Edition

INSTITUTO FEDERAL
Santa Catarina
Câmpus
São José

Passo 3: Treinando o modelo

Association rule syntax

using the `apriori()` function in the `arules` package

Finding association rules:

```
myrules <- apriori(data = mydata, parameter =
  list(support = 0.1, confidence = 0.8, minlen = 1))
```

- `data` is a sparse item matrix holding transactional data
- `support` specifies the minimum required rule support
- `confidence` specifies the minimum required rule confidence
- `minlen` specifies the minimum required rule items

The function will return a rules object storing all rules that meet the minimum criteria.

Examining association rules:

```
inspect(myrules)
```

- `myrules` is a set of association rules from the `apriori()` function

This will output the association rules to the screen. Vector operators can be used on myrules to choose a specific rule or rules to view.

Example:

```
groceryrules <- apriori(groceries, parameter =
 list(support = 0.01, confidence = 0.25, minlen = 2))
inspect(groceryrules[1:3])
```

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 3: Treinando o modelo

Com a configuração padrão, nenhuma regra é gerada.

```
> apriori(groceries)
set of 0 rules
```

Parâmetros sugeridos: Um item tem que ser comprado em média duas vezes por dia, as associações definidas pelas regras ocorrem pelo menos 25% das vezes e vamos considerar regras que contém menos de dois itens.

```
> groceryrules <- apriori(groceries, parameter = list(support =
                          0.006, confidence = 0.25, minlen = 2))

> groceryrules
set of 463 rules
```

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 4: Analisando o desempenho do modelo

```
> summary(groceryrules)
set of 463 rules

rule length distribution (lhs + rhs):sizes
  2   3   4
150 297  16

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   3.000   2.711   3.000   4.000
```

Left-hand side (LHS) e right-hand side (RHS)

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 4: Analisando o desempenho do modelo

```
summary of quality measures:
    support           confidence          lift             count
 Min.   :0.006101   Min.   :0.2500   Min.   :0.9932   Min.   : 60.0
 1st Qu.:0.007117   1st Qu.:0.2971   1st Qu.:1.6229   1st Qu.: 70.0
 Median :0.008744   Median :0.3554   Median :1.9332   Median : 86.0
 Mean   :0.011539   Mean   :0.3786   Mean   :2.0351   Mean   :113.5
 3rd Qu.:0.012303   3rd Qu.:0.4495   3rd Qu.:2.3565   3rd Qu.:121.0
 Max.   :0.074835   Max.   :0.6600   Max.   :3.9565   Max.   :736.0
```

$$\text{lift}(X \rightarrow Y) = \frac{\text{confidence}(X \rightarrow Y)}{\text{support}(Y)}$$

O valor do lift maior que 1 implica que X e Y são encontrados juntos com mais frequência do que um deles separadamente

```
mining info:
      data ntransactions support confidence
 groceries          9835   0.006       0.25
```

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Passo 4: Analisando o desempenho do modelo

```
> inspect(groceryrules[1:3])

  lhs                   rhs                  support     confidence lift
1 {potted plants}    => {whole milk}       0.006914082 0.4000000  1.565460
2 {pasta}            => {whole milk}       0.006100661 0.4054054  1.586614
3 {herbs}            => {root vegetables}  0.007015760 0.4312500  3.956477
```

Leitura da regra 1: se um cliente comprar vasos de plantas, ele está 1.56 vezes mais propenso a comprar leite integral do que um cliente qualquer. Essa regra cobre 0,7% das transações e ocorre em 40% das compras envolvendo vasos de plantas.

Uma abordagem comum é dividir as regras de decisão em três categorias: **Úteis**; **triviais** ou **inexplicáveis**.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

INSTITUTO FEDERAL
Santa Catarina
Câmpus São José

Passo 4: Analisando o desempenho do modelo

Ordenando as regras de associação

```
> inspect(sort(groceryrules, by = "lift")[1:5])
```

```
  lhs                     rhs                    support     confidence lift
1 {herbs}              => {root vegetables}    0.007015760 0.4312500  3.956477
2 {berries}            => {whipped/sour cream} 0.009049314 0.2721713  3.796886
3 {other vegetables,
   tropical fruit,
   whole milk}         => {root vegetables}    0.007015760 0.4107143  3.768074
4 {beef,
   other vegetables}   => {root vegetables}    0.007930859 0.4020619  3.688692
5 {other vegetables,
   tropical fruit}     => {pip fruit}          0.009456024 0.2634561  3.482649
```

Leitura da regra 1: Os clientes que compram ervas são quase quatro vezes mais propensos a comprar vegetais de raiz do que um cliente qualquer.

Fonte: Brett Lantz – Machine Learning with R – Second Edition

INSTITUTO FEDERAL
Santa Catarina
Câmpus São José

Passo 4: Analisando o desempenho do modelo

Analisando um subconjunto das regras

```
> berryrules <- subset(groceryrules, items %in% "berries")
> inspect(berryrules)
```

```
  lhs           rhs                      support     confidence     lift
1 {berries} => {whipped/sour cream} 0.009049314  0.2721713 3.796886
2 {berries} => {yogurt}             0.010574479  0.3180428 2.279848
3 {berries} => {other vegetables}   0.010269446  0.3088685 1.596280
4 {berries} => {whole milk}         0.011794611  0.3547401 1.388328
```

Fonte: Brett Lantz – Machine Learning with R – Second Edition

Salvando as regras em um arquivo ou *dataframe*

```
> write(groceryrules, file = "groceryrules.csv",
        sep = ",", quote = TRUE, row.names = FALSE)
```

```
> groceryrules_df <- as(groceryrules, "data.frame")
```

Fonte: Brett Lantz – Machine Learning with R – Second Edition