

# Data Mining

## Regras de Associação

Prof. Dr. Joaquim Assunção

DEPARTAMENTO DE COMPUTAÇÃO APLICADA  
CENTRO DE TECNOLOGIA  
UFSM  
2019

[www.inf.ufsm.br/~joaquim](http://www.inf.ufsm.br/~joaquim)



# *Fair user agreement*

Este material foi criado para a disciplina de Mineração de Dados - Centro de Tecnologia da UFSM.

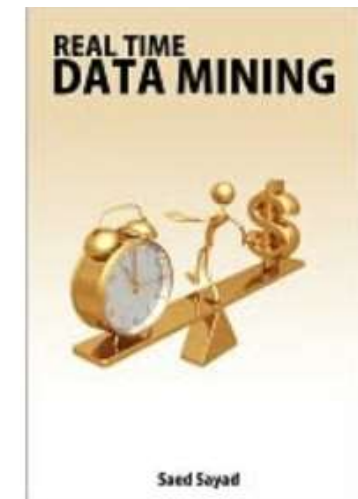
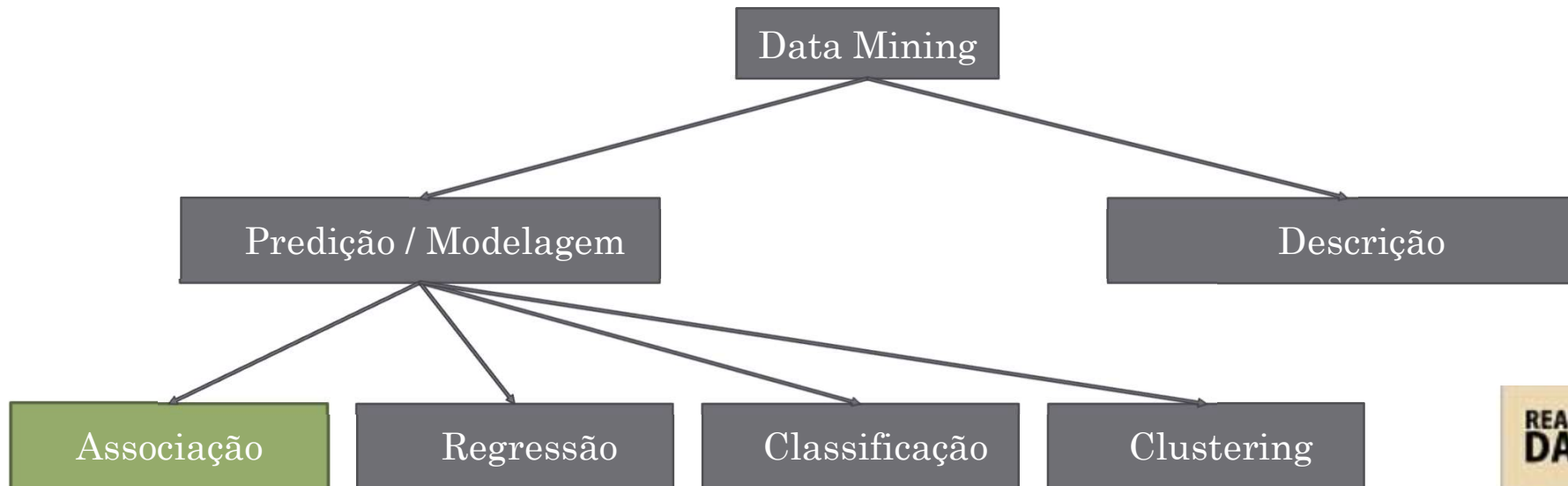
Você pode usar este material livremente\*; porém, caso seja usado em outra instituição, **me envie um e-mail** avisando o nome da instituição e a disciplina.

\*Caso você queira usar algo desse material em alguma publicação, envie-me um e-mail com antecedência.

Prof. Dr. Joaquim Assunção.

joaquim@inf.ufsm.br

# Mapa para Mineração de Dados\*



\*[http://www.saedsayad.com/data\\_mining\\_map.htm](http://www.saedsayad.com/data_mining_map.htm)

# Trabalho 1

Em dupla.

Ver a descrição na página da disciplina.



# Conviction

Similar a confiança, esta regra trata os conjuntos A e B como independentes, e no divisor usa a probabilidade de união do conjunto A com a negação do conjunto B.

Conviction é uma medida de implicação que resulta em 1 se os conjuntos não forem relacionados.  $\neg(A, \neg B)$

Formalmente:

$$\frac{\textit{suporte}(A)\textit{suporte}(\bar{B})}{\textit{suporte}(A \cup \bar{B})}$$

# Leverage (influência)

Proposta por Piatetsky-Shapiro em 1991, é uma medida da diferença entre a probabilidade de  $A \rightarrow B$  e a probabilidade esperada caso A e B fossem independentes.

$$\text{suporte}(A \Rightarrow B) - \text{suporte}(A)\text{suporte}(B)$$

# *Cross Support Ratio*

Definido em conjuntos de itens como a proporção do suporte do item menos frequente ao suporte do item mais frequente, ou seja,

$$\min(\text{Suporte}(a \in A)) / \max(\text{Suporte}(a \in A))$$

## *Added Value*

A confiança da regra menos o suporte da implicação.  
Um valor entre -5 e 1.

$$Conf(A \Rightarrow B) - Suporte(B).$$



# *Hands On!*

- Dadas as variáveis abaixo. Gere regras de associação que implicam em `goal==1`. Calcule a convicção e a influência para cada uma destas regras.

```
a <- c(1,1,0,0,1,1,0,1)
b <- c(0,1,0,1,1,0,0,0)
c <- c(0,1,1,0,1,1,1,0)
goal <- c(1,0,1,0,1,1,1,1)
```

# Geração de regras

Já parou para pensar na quantidade de regras possíveis por conjunto de itens???

Ex: Se tivermos A,B,C, poderíamos gerar:

$A \rightarrow B$        $B \rightarrow C$        $A,C \rightarrow B$

$B \rightarrow A$        $C \rightarrow B$        $A \rightarrow B,C$

$A \rightarrow C$        $A,B \rightarrow C$        $B \rightarrow A,C$

$C \rightarrow A$        $B,C \rightarrow A$        $C \rightarrow B,A$

# Geração de regras

Já parou para pensar na quantidade de regras possíveis por conjunto de itens???

Ex: Se tivermos A,B,C, poderíamos gerar:

$A \rightarrow B$	$B \rightarrow C$	$A, C \rightarrow B$	$\text{NULL}$
<del><math>B \rightarrow A</math></del>	<del><math>C \rightarrow B</math></del>	<del><math>A \rightarrow B, C</math></del>	<del><math>A, B, C \rightarrow \text{NULL}</math></del>
$A \rightarrow C$	$A, B \rightarrow C$	<del><math>B \rightarrow A, C</math></del>	$\text{NULL} \rightarrow A, B, C$
<del><math>C \rightarrow A</math></del>	$B, C \rightarrow A$	<del><math>C \rightarrow B, A</math></del>	

Podemos cortar as inversas, para N itens teremos  $2^N$  candidatos.

# Geração de regras

Já parou para pensar na quantidade de regras possíveis por conjunto de itens???

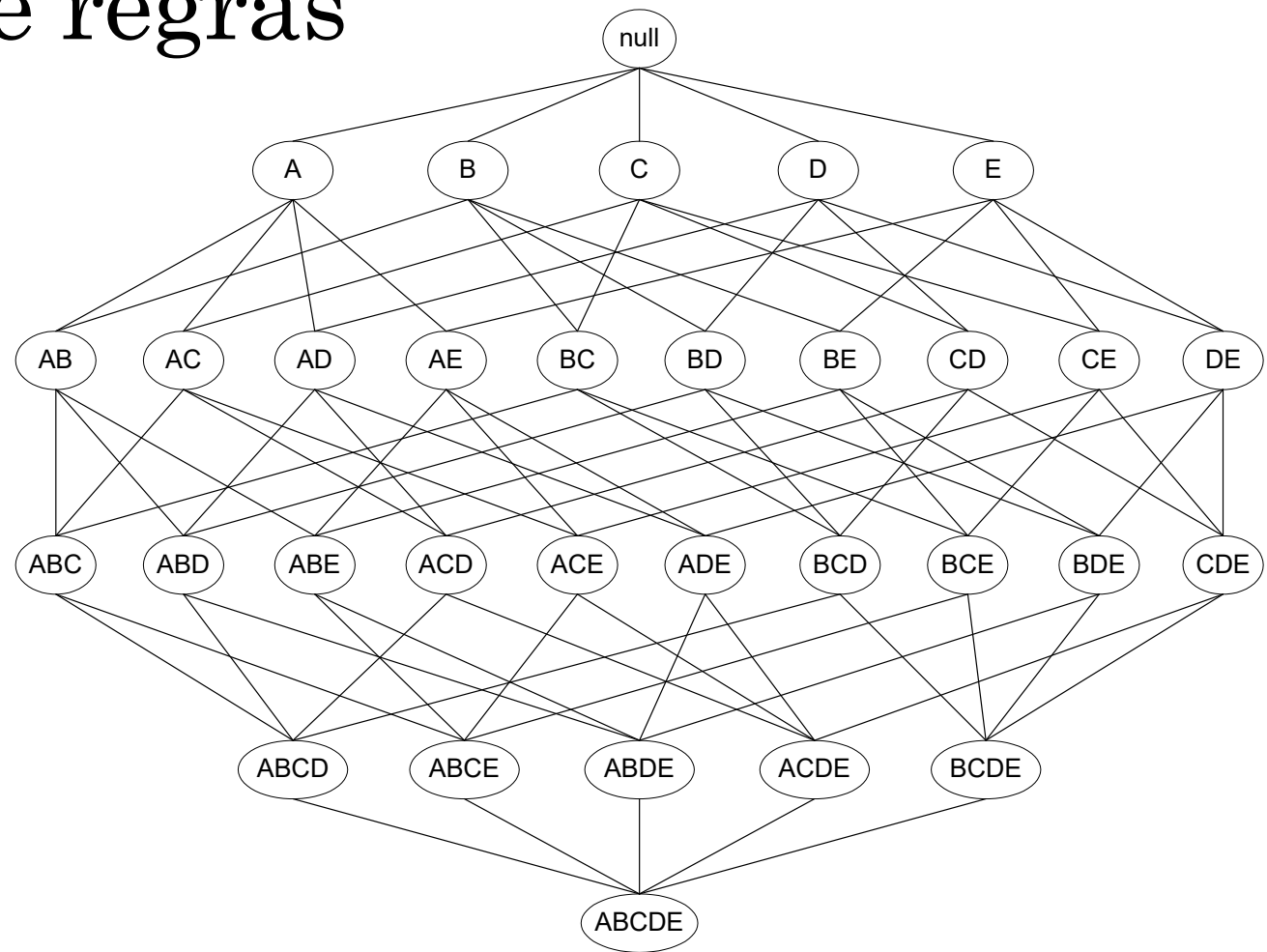
Ex: Se tivermos A,B,C, poderíamos gerar:

$A \rightarrow B$	$B \rightarrow C$	$A, C \rightarrow B$	$\text{NULL}$
<del><math>B \rightarrow A</math></del>	<del><math>C \rightarrow B</math></del>	<del><math>A \rightarrow B, C</math></del>	<del><math>A, B, C \rightarrow \text{NULL}</math></del>
$A \rightarrow C$	$A, B \rightarrow C$	<del><math>B \rightarrow A, C</math></del>	$\text{NULL} \rightarrow A, B, C$
<del><math>C \rightarrow A</math></del>	$B, C \rightarrow A$	<del><math>C \rightarrow B, A</math></del>	

Se considerarmos todos os itens como um conjunto possível, teremos  $2^N$  ou como formalmente atribuído a **D** o conjunto de itens  $2^D$

# Geração de regras

Agora vamos  
ver com 5 itens:

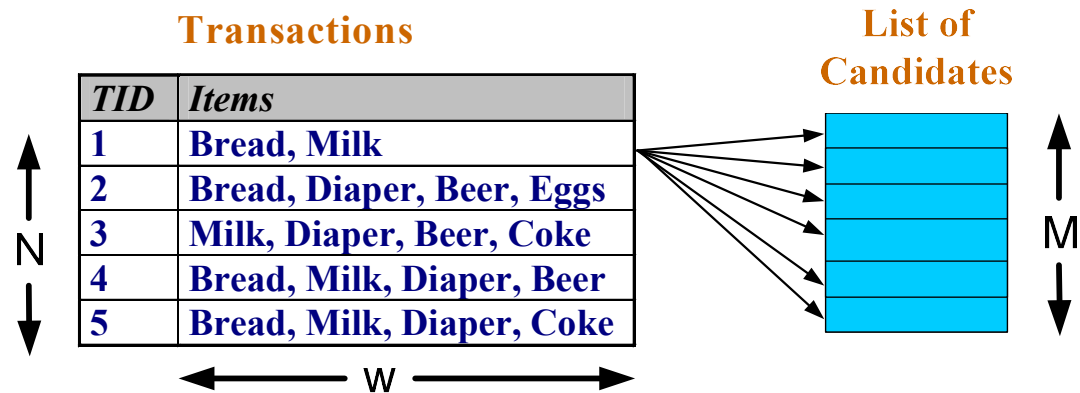


\* Exemplo do livro de Tan et. Al. (ver bibliografia da disciplina)

# Geração de regras

Na **abordagem por Força Bruta** cada item do conjunto é um candidato a frequente.

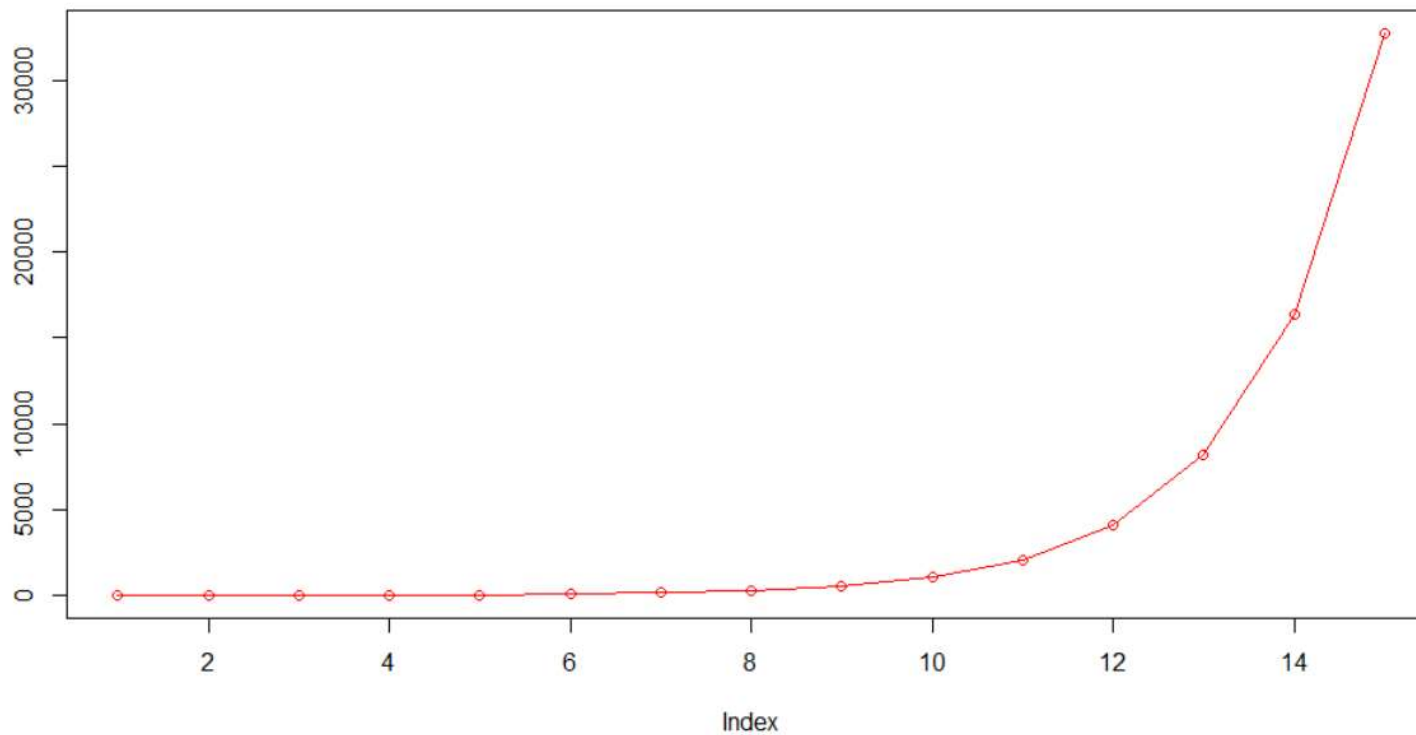
Primeiro se conta o suporte de cada item varrendo o conjunto.



\* Exemplo do livro de Tan et. Al. (ver bibliografia da disciplina)

# Geração de regras

Na abordagem por Força Bruta... 15 itens



# Geração de regras

O número total de possíveis regras é ainda maior!

$$R = \sum_{k=1}^{d-1} \left[ \binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right]$$
$$= 3^d - 2^{d+1} + 1$$

Para um conjunto de 3 itens temos 12 regras.

Para um conjunto de 8 itens é possível ter 6050 regras!



# Princípio APRIORI

O princípio APRIORI leva tem base na estratégia de reduzir o número de candidatos que se dá por  $2^D$  por meio de poda.

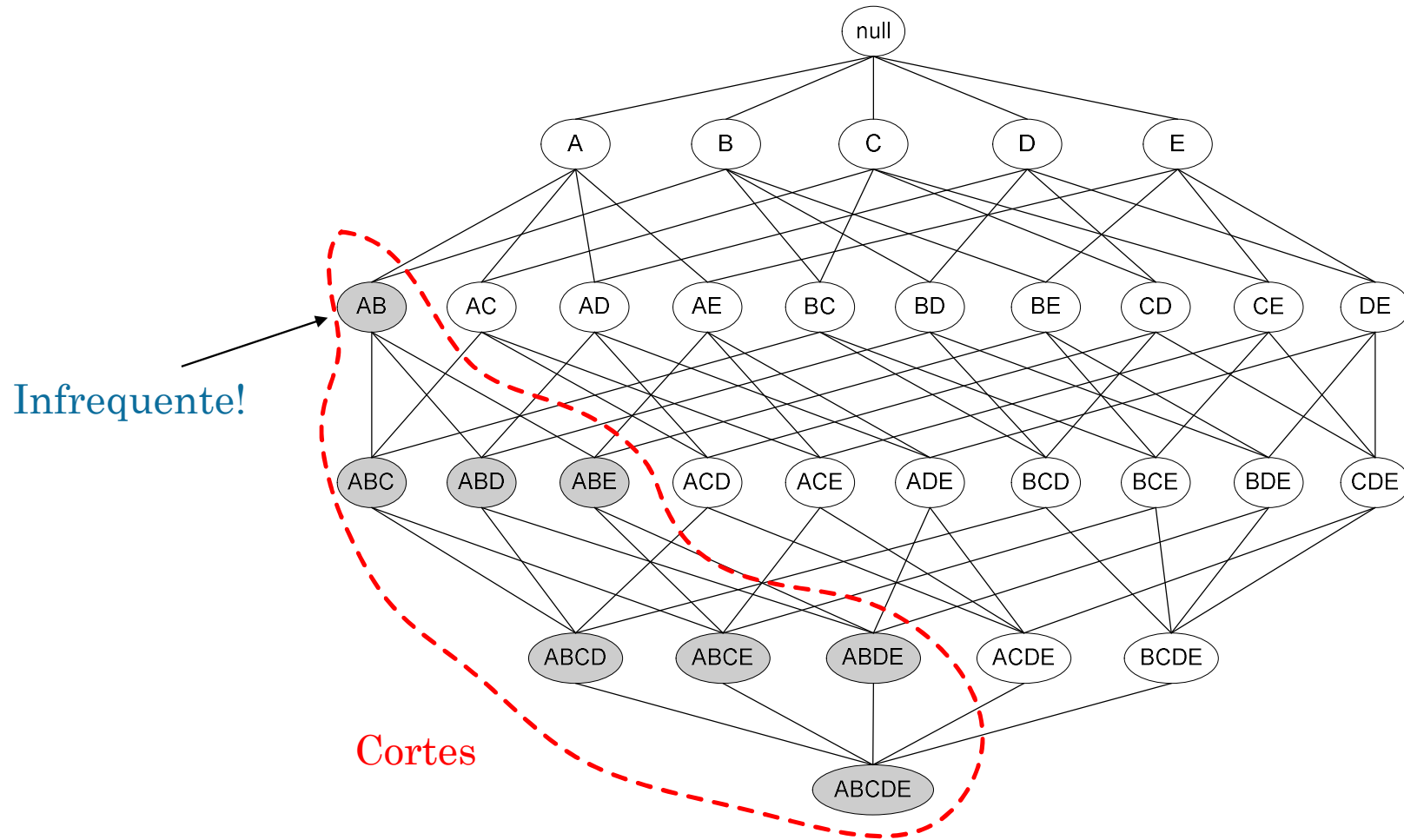
O princípio é:

*“Se um conjunto é infrequente, então todos seus subconjuntos também serão infrequentes”*

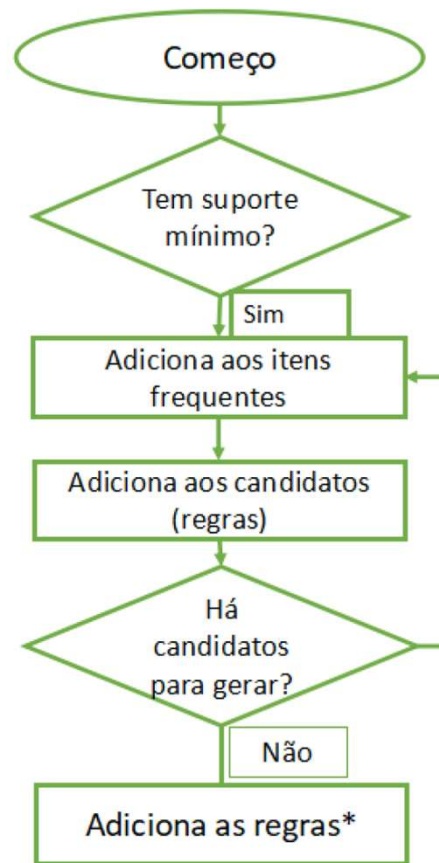
O suporte de um conjunto nunca é maior que o suporte de seus subconjuntos.

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

# Principio APRIORI



# Geração de regras – fluxo geral



TID	Itens
1	a, c, d
2	b, c, e
3	a, b, c, e
4	b, e

Item	Suporte
a	2
b	3
c	3
e	3

Candidato	Suporte
{a, b}	1
{a, c}	2
{a, e}	1
{b, c}	2
{b, e}	3
{c, e}	2

Candidato	Suporte
{a, c}	2
{b, c}	2
{b, e}	3
{c, e}	2

Regra Forte	Suporte
{b, c, e}	2

Geração de regras forte  
– exemplo  
(suporte  $\geq 2$ )

## *Hands On!*

- Dado o seguinte conjunto:

Item	Contagem
Coca-Cola	9
Cerveja	6
Suco	2
Néctar	5
Tônica	1
Água	4

Se o suporte mínimo for definido como 4.

1. Quais itens não seriam considerados?
2. Quantos conjuntos possíveis teríamos que verificar por força bruta?
3. Quantos conjuntos possíveis teríamos com a propriedade APRIORI?