Regras de Associação

Departamento de Engenharia Informática (DEI/ISEP)
Fátima Rodrigues
mfc@isep.ipp.pt

Regras de Associação

Têm por objetivo gerar todas as associações para as quais a presença de um ou vários itens específicos numa transação impliquem a presença de outros itens

Analisam-se as transações produzidas por um conjunto de indivíduos

→ o **que** fazem

Para extrair conhecimento sobre os indivíduos

→ **quem** são

→ **como** se comportam

→ o que permite **AGIR**

Regras de Associação

Transacção	Conhecimento	Acção
Compras no supermercado	Perfil dos clientes	Promoções Cupões de desconto Layouts de lojas
Movimentos com o cartão de crédito	Despesas associadas	Oferta de produtos financeiros
Revistas assinadas	Associação de preferências	*Mailings* direcionados
Pedidos de indemnização	Descrição de comportamento	Deteção de fraudes
Registos médicos	Relação tratamento-efeitos secundários	Estudar novos tratamentos

Regras de Classificação vs. Associação

Regras de Classificação

- Têm apenas um atributo objetivo
- Especificam a classe para todas as instancias
- Medidas avaliação: accuracy, precision, recall, …
- Aprendizagem **supervisionada**

Regras de Associação

- Vários atributos objetivo
- Aplicáveis em apenas alguns casos
- Medidas de Avaliação: suporte, confiança, interesse,…
- Aprendizagem **não supervisionada**

Regras de Associação

Um Modelo de Associação

Identifica grupos de itens que co-ocorrem

Tipos de Regras de Associação

Úteis

Quem compra X compra Y

Triviais

Todos os inquiridos em estado de gravidez são do sexo feminino

Inexplicáveis

Na abertura de uma determinada loja o artigo mais vendido foram os guardanapos

Regras de associação triviais são as que mais ocorrem

Conceitos Básicos

Conjunto de objetos $\{o_1, o_2, ..., o_n\}$ (cestos) $\rightarrow$ **Definição Original**
tipo especial de dados chamado
"basket data" (dados de cesto)

Cada objeto (transação, cesto) é um conjunto de itens (ou produtos)
$T = \{i_a, i_b, ..., i_t\}$

$T \subset I$, em que I é o conjunto de todos os possíveis itens $\{i_1, i_2, ..., i_n\}$

Uma Regra de Associação é da forma:

$$A \Rightarrow B, \quad \text{em que} \quad A \subset I, \; B \subset I \quad \text{e} \quad A \cap B = \{\,\}$$

As Regras de Associação são calculadas a partir dos dados e têm **natureza probabilística**

Regras de Associação devem ser interpretadas com cuidado
$A \Rightarrow B$ não implica necessariamente causalidade mas sim co-ocorrência
Se observarmos um conjunto de **itens A** devemos também observar um outro conjunto de **itens B**

Medidas de Avaliação

Medidas subjetivas [Silberschatz & Tuzhilin, KDD95]

Uma regra é interessante se

- ela é inesperada ; e/ou
- utilizável (*actionable*)

Medidas subjetivas variam de utilizador para utilizador

- ↪ daí que os algoritmos de Regras de Associação apliquem medidas objetivas baseadas em conceitos de Estatística

Medidas Objetivas

- Suporte
- Confiança
- Cobertura
- Interesse
- …

Suporte

Simbolicamente

$\{A_1, A_4\} => \{A_6\}$ Sup, Conf

Suporte da Regra: % de "cestos" em que a ocorrência $\{A_1, A_4\}$ e $\{A_6\}$ ocorre

$$Suporte(\{A_1, A_4\} => \{A_6\}) = Probabilidade(A_1 \cap A_4 \cap A_6)$$

Probabilidade de uma transação conter $\{A_1 \cap A_4 \cap A_6\}$

Suporte é a **Significância Estatística** da Regra de Associação, **mede a frequência dos itens** $\{A_1, A_4, A_6\}$ na base de dados

$$Suporte(\{A_1, A_4\} \Rightarrow \{A_6\}) = \frac{(A_1 \cap A_4 \cap A_6).cont}{n}$$

Suporte é usado para **eliminar regras que ocorrem pouco na BD**

Confiança

Simbolicamente

{A1, A4} => {A6} Sup, Conf

Confiança: % de casos em que a ocorrência de {A1, A4} corretamente prevê a ocorrência de {A6}

$$Confiança\ (\{A_1, A_4\} => \{A_6\}) = Probabilidade(\{A_6\} | \{A_1, A_4\})$$

Probabilidade condicional que uma transacção que contenha $\{A_1 \cap A_4\}$ também contém A_6 - indica, no conjunto de dados, o grau de correlação entre A_1, A_4 e A_6

$$Confiança = \frac{P(A_1 \cap A_4 \cap A_6)}{P(A_1 \cap A_4)}$$

A **confiança** da regra é uma medida do seu **poder de previsão**

Suporte e Confiança

Cliente (TID)	Itens (Item-set)
100	chá(c), manteiga(m), bolachas(b), leite(l)
200	chá(c), bolachas(b), açúcar(a)
300	manteiga(m), chá(c), bolachas(b), açúcar(a)
400	chá(c), açúcar(a)

Uma Regra de Associação é um relacionamento

{b, c} => {l}

$$\text{Suporte}(\{b, c\} \Rightarrow \{l\}) = \frac{\text{Nº de registos com b, c e l}}{\text{Nº Total de registos}} = \frac{1}{4}$$

$$\text{Confiança}(\{b, c\} \Rightarrow \{l\}) = \frac{\text{Nº de registos com b, c e l}}{\text{Nº de registos com c, b}} = \frac{1}{3}$$

{b, c} => {l} Suporte = 25% , Confiança = 33%

Especificação do Problema

Dados

Um conjunto de transações **D**

$$D = \{ T \mid T \text{ é um conjunto de itens} \}$$

Um suporte mínimo $\mathbf{Sup_{min}}$

Uma confiança mínima $\mathbf{Conf_{min}}$

Obter

TODAS as regras de associação

A ⇒ B (s:Sup, c:Conf) tais que $Conf \geq Conf_{Min}$ e $Sup \geq Sup_{Min}$

↪ Regras válidas ou **Regras Fortes**

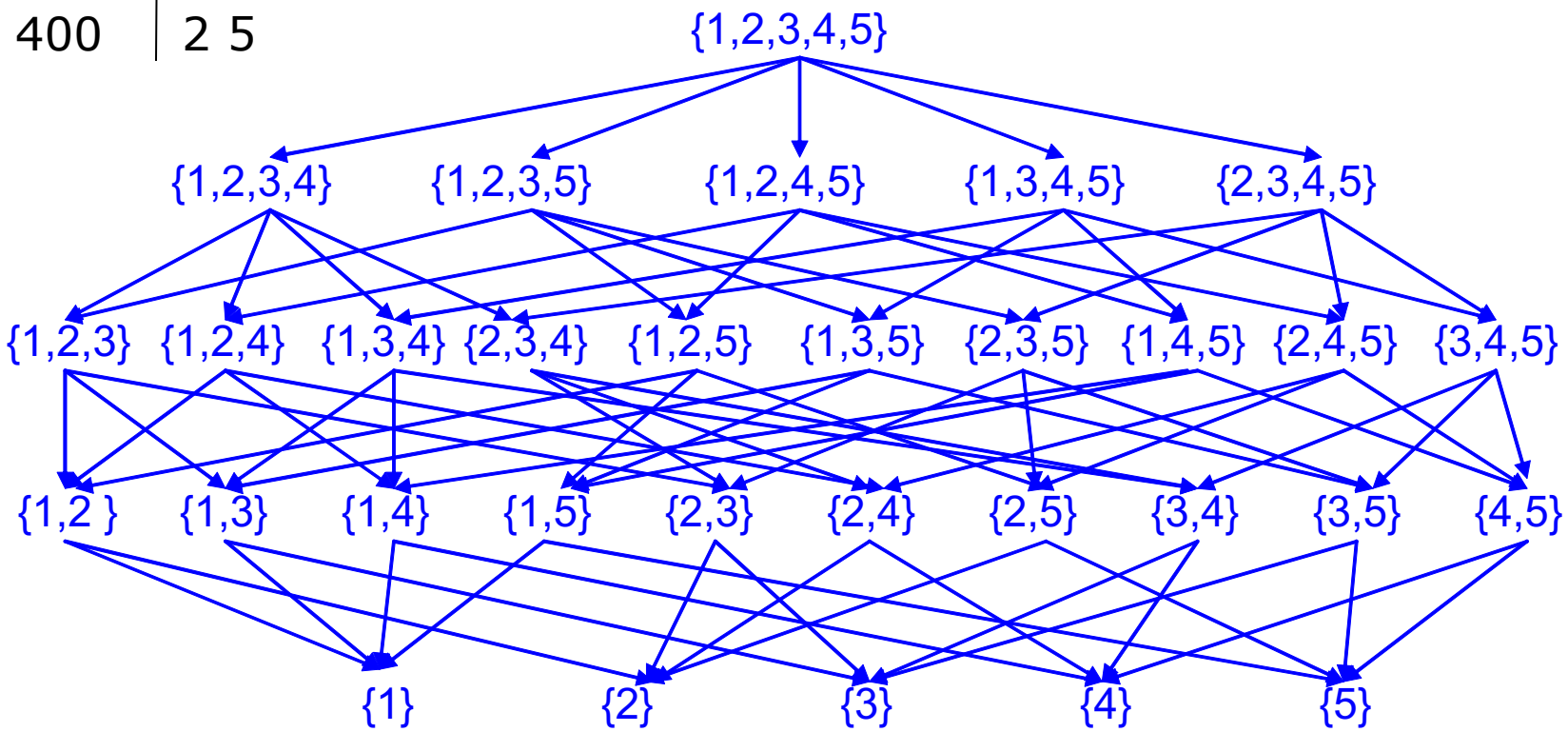
Aproximação exaustiva:

- Listar todas as possíveis regras de associação
- Calcular o suporte e a confiança para cada regra
- Cortar as regras que não verificam $\mathbf{Sup_{min}}$ **e** $\mathbf{Conf_{min}}$

⇒ **Computacionalmente impossível !**

Espaço de Procura – Conjuntos de Itens

Cliente	Itens
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5



Complexidade Extração de Regras de Associação

A Geração de conjuntos frequentes de itens é **computacionalmente intensivo**. Para d itens:

Número total conjuntos de itens = 2^d-1

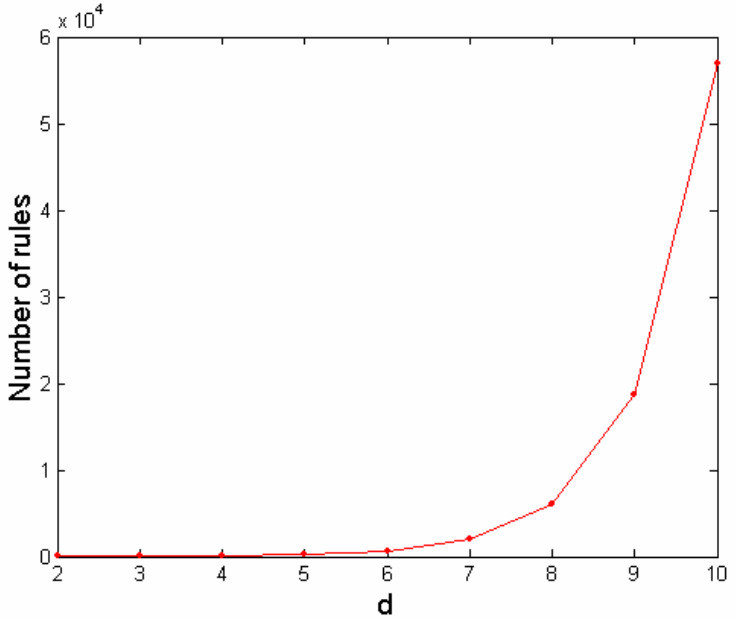
- Número total regras de associação:

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

D = 5

→ C = 2^5 -1 = 31 Conj^{tos} itens

→ R = 3^5 - 2^6 -1 = 180 Regras



Algoritmo Apriori [Agrawal et al. 93]

Envolve dois passos:

1. Gerar **conjuntos de itens frequentes**
   - Selecionar todos os conj^{tos} de items com suporte $\geq Sup_{min}$

2. Gerar **Regras de Associação Fortes**
   - A partir dos conj^{tos} de items frequentes, gerar regras com confiança $\geq Conf_{min}$, através de partições binárias dos conj^{tos} de items frequentes

Princípio do Algoritmo Apriori

Se um conjunto de itens não satisfaz o Suporte mínimo então podemos ignorar todos os seus super-conjuntos

$$\forall_{X,Y} : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

ou seja, o suporte de um conjunto de itens nunca excede o suporte dos seus subconjuntos

➥ **propriedade anti monótona do Suporte**

1º: Construção de Conjuntos Frequentes

1. Começa-se com conjuntos de tamanho 1 (**Sup_{Min} = 50%**)

Cliente	Itens
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Conjto de Transações

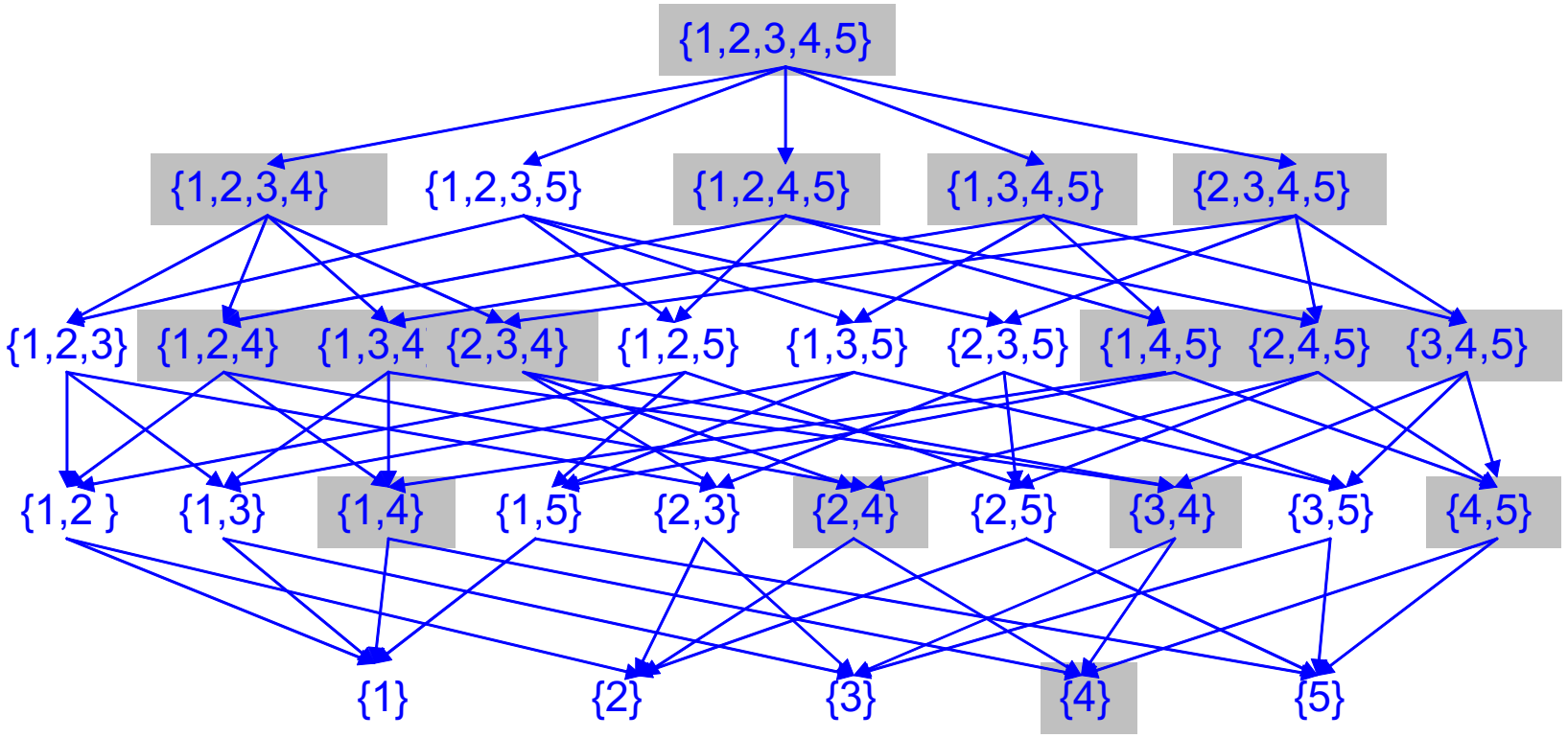
Item-Set	Suporte
{1}	0,5
{2}	0,75
{3}	0,75
{4}	0,25
{5}	0,75

Eliminado

Conjtos frequentes de tamanho 1 - F_1

Redução do Espaço de Procura

Através da **Propriedade anti monótona do Suporte**



1º: Construção de Conjuntos Frequentes

1. Conjuntos frequentes de tamanho 1 - F_1: {1},{2},{3},{5}
2. Os conjuntos de tamanho 2 são obtidos a partir dos conjuntos frequentes de tamanho 1

Cliente	Itens
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Item-Set	Suporte
{1,2}	0,25
{1,3}	0,5
{1,5}	0,25
{2,3}	0,5
{2,5}	0,75
{3,5}	0,5

Conjuntos frequentes de tamanho 2 - F_2

1º: Construção de Conjuntos Frequentes

1. F_1: {1},{2},{3},{5}
2. F_2: {1,3},{2,3},{2,5},{3,5}
3. Os conj^{tos} de tamanho 3 são obtidos a partir dos itens frequentes dos conj^{tos} tamanho 1 e 2 – F_2

Cliente	Itens
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Item-Set	Suporte
{1,2,3}	0,25
{1,2,5}	0,25
{1,3,5}	0,25
{2,3,5}	0,50

Conjuntos frequentes de tamanho 3 (F_3)

Item-Set	Suporte
{1,2,3,5}	0,25

Conjuntos frequentes de tamanho 4 (F_4)

1º: Construção de Conjuntos Frequentes

Se os conjuntos forem gerados por enumeração exaustiva, o nº de item-sets no nível k é dado por:

$$\binom{d}{k} = \frac{d!}{(d-k)! \times k!}$$

$^5C_1 + {}^5C_2 + {}^5C_3 + {}^5C_4 = 5 + 10 + 10 + 5 + 1 = 31$

Com a poda de item-sets baseada no suporte: 4 + 4 + 1= 9

1. F_1: {1},{2},{3},{5}
2. F_2: {1,3},{2,3},{2,5},{3,5}
3. F_3: {2,3,5}

2º Passo: Geração de Regras

A partir dos conjuntos frequentes F_2, F_3 obtidos constroem-se as regras

Cliente	Itens
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$F_2 = \{ \{1, 3\} \{2, 3\} \{2, 5\} \{3,5\} \}$

Regra	Conf
{1}⇒{3}	1
{3}⇒{1}	2/3
{2}⇒{3}	2/3
{3}⇒{2}	2/3
{2}⇒{5}	1
{5}⇒{2}	1
{3}⇒{5}	2/3
{5}⇒{3}	2/3

Regras com origem no mesmo conjto frequente têm o mesmo Suporte, mas podem ter diferentes valores de Confiança

2º: Geração de Regras

A partir do conjunto frequente F_3 constroem-se as regras

Cliente	Itens
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

$F_3 = \{2, 3, 5\}$

Regras	**Conf**
{2,3} ⇒ {5}	1
{2,5} ⇒ {3}	2/3
{3,5} ⇒ {2}	1
{2} ⇒ {3,5}	2/3
{3} ⇒ {2,5}	2/3
{5} ⇒ {2,3}	2/3

Resultado Final do Algoritmo APRIORI

Do conjunto regras geradas selecionam-se aquelas que apresentam confiança = $Conf_{min}$ = 100% - **Regras de Associação Fortes**

Cliente	Itens
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

{1} => {3} (c: 1, s: 0.5)

{2} => {5} (c: 1, s: 0.75)

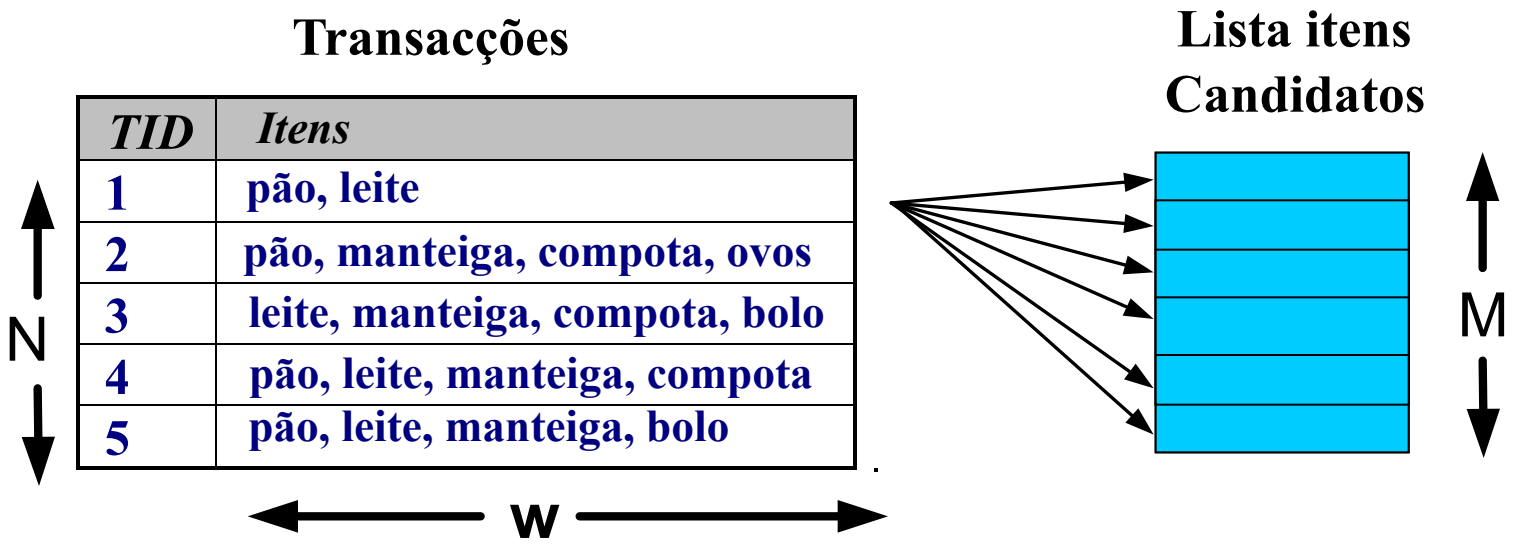
{5} => {2} (c: 1, s: 0.75)

{2,3} => {5} (c: 1, s: 0.5)

{3,5} => {2} (c: 1, s: 0.5)

Geração dos Conjuntos Frequentes

- A geração dos conj[tos] frequentes é computacionalmente mais exigente do que a geração de regras



- O suporte de cada item-set é calculado fazendo o scanning à BD
- Complexidade ~ O(N×M×W)
  - ➥ Computacionalmente intensivo M = 2^d

Construção de Conjuntos Frequentes

Existem diferentes maneiras de gerar conjuntos frequentes. Um método é efectivo se:

- Evitar o mais possível a geração de conjuntos candidatos não necessários, ou seja, conjuntos com pelo menos um item infrequente
- Assegurar que o conjunto de candidatos gerado é completo
- Não gerar o mesmo conjunto candidato mais do que uma vez

A geração de conjuntos frequentes por enumeração exaustiva torna o processo de poda mais intensivo, pois um número elevado de conj^{tos} candidatos é gerado e como tal tem de ser analisado

Construção de Conjuntos Frequentes

Método $F_{k-1} \times F_1$

Cada k-itemset é criado por extensão do (k-1)-itemset com o **k1**-itemset

Este método produz $O(|F_{k-1}| \times |F_1|)$ conjuntos candidatos

O procedimento é completo

De modo a evitar a geração de conj^{tos} repetidos **é importante manter os conj^{tos} ordenados** e cada (k-1)-itemset é estendido apenas com os conj^{tos} frequentes maiores

F2: {1,3},{2,3},{2,5},{3,5}

F1: {1},{2},{3},{5}

$\Rightarrow$ F3: {1,2,3} {1,2,5} {1,3,5} {2,3,5}

F3: {1,2,3} {1,2,5} {1,3,5} {2,3,5}

F1: {1},{2},{3},{5}

$\Rightarrow$ F4: {1,2,3,5}

Optimização da Geração de Regras

- Como gerar eficientemente regras a partir dos conjuntos frequentes de itens?
  - Em geral, a Confiança não obedece à propriedade anti-monótona que o Suporte verifica

    Conf(ABC →D) pode ser maior ou menor do que Conf(AB →D)
  - A Confiança é anti-monótona, pois depende do nº de items do lado esquerdo da regra
  - Mas, **regras geradas a partir do mesmo conjunto de itens frequentes apresentam a propriedade anti-monótona**
  - Por exemplo, L = {A,B,C,D}:

$$Conf(ABC \rightarrow D) \geq Conf(AB \rightarrow CD) \geq Conf(A \rightarrow BCD)$$

Geração de Regras

- Dado um conjunto frequente de itens L, encontrar todos os subconjuntos não vazios $f \subset L$ tais que $f \rightarrow L - f$ satisfaz a confiança mínima
  - Se {A,B,C,D} conjunto frequente de itens, começam-se por gerar as **regras com elevada confiança – regras com apenas um item no lado direito da regra**:

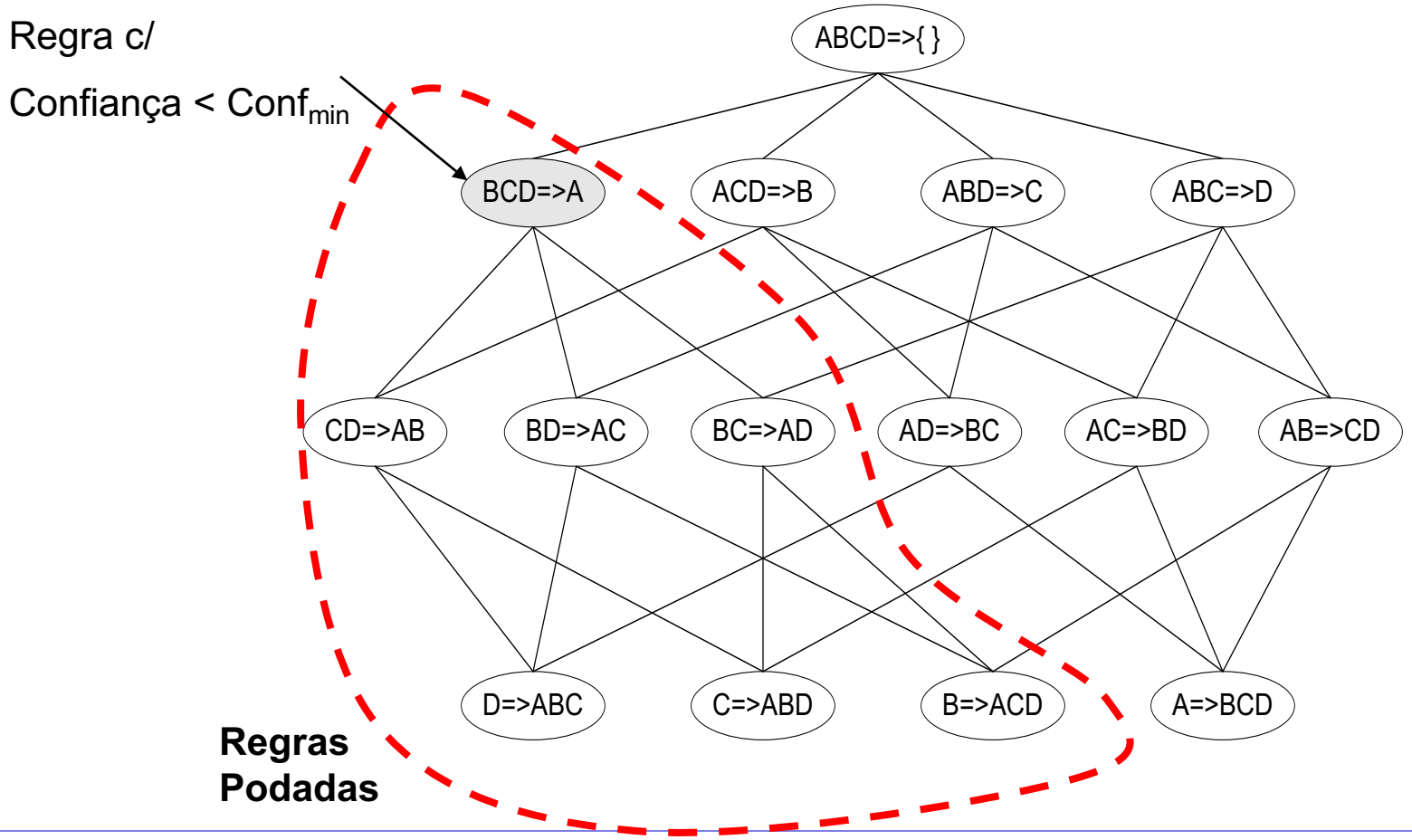
ABC $\rightarrow$D,	ABD $\rightarrow$C,	ACD $\rightarrow$B,	BCD $\rightarrow$A
AB $\rightarrow$CD,	AC $\rightarrow$ BD,	AD $\rightarrow$ BC	
BC $\rightarrow$AD,	BD $\rightarrow$AC,	CD $\rightarrow$AB	
A $\rightarrow$BCD,	B $\rightarrow$ACD,	C $\rightarrow$ABD,	D $\rightarrow$ABC

- Se **|L| = k**, existem **$2^k - 2$** regras de associação candidatas (ignorando $L \rightarrow \varnothing$ e $\varnothing \rightarrow L$)

Geração de Regras – Algoritmo Apriori

Teorema: Seja o conj^{to} Frequente $F=\{I_1,I_2,..,I_n\}$

Se a regra $I_1,I_2,..,I_{n-1} \rightarrow I_n$ não satisfaz a $Conf_{min}$, então qualquer regra $I_1 \rightarrow I_2,..,I_{n-1},I_n$ também não satisfaz a $Conf_{min}$



Avaliação APRIORI

Fatores de complexidade

- Número total de itens
- Média de itens por transação
- Número de conjuntos frequentes
- Número médio de itens por conjunto frequente
- Suporte mínimo
- Número de transações
- Número de passagens pela base de dados

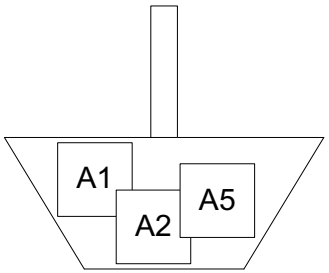
  = k ou k+1 em que k é o tamanho do maior conjunto frequente

- Rápido
- Necessita grande espaço de memória
- Faz muitos acessos à base de dados (“hard disk”)

Variantes do Algoritmo APRIORI

- Existem muitas variantes e extensões
  - Menos acessos ao disco: minimizar o número de passagens pela base de dados (I/O), exemplos:
    - *Dynamic Item-set Counting* (DIC) [Brin et al. 97]
    - *Partition* [Savasere 95]
  - Implementações paralelas (mais rápidas), exemplos:
    - PDM – Parallel Data Mining [Park 1995]
  - **FP-growth Algorithm** – representação mais eficiente (até à data!)
    - A BD é representada usando uma FP-tree
    - Um algoritmo recursivo *"divide-and-conquer"* é usado para extrair os conjuntos frequentes

Preparação dos Dados (Basket Data)



Cliente #1

A1 ..... 2 ... 45
A7 ..... 3 ... 760
A5 ..... 4.... 120
...
An ..... 7... 350
12-3-01

Cliente #2

A4 ..... 1.... 170
A8 ..... 5.... 750
A9 ..... 3.... 25

20-3-01

Cliente #10

A4 ..... 5..... 250
A1 ..... 1..... 430
A3 ..... 4..... 35
A6.....12..... 610

14-3-01

Representação horizontal

Representação vertical

#Cliente	...	A1	A2	A3	A4	A5	...	An
1		1	0	0	0	1		1
2		0	0	0	1	0		0
...								
10		1	0	1	1	0		0

Python: Apriori

```
from mlxtend.frequent_patterns import apriori

freqItemsets = apriori(basket, min_support=0.05, use_colnames=True)

freqItemsets
```

	support	itemsets	length
0	0.324115	(beer)	1
1	0.335177	(canned)	1
2	0.324115	(charcuterie)	1
3	0.305310	(clothes)	1
4	0.323009	(dairy)	1
...	...	...	...
115	0.055310	(charcuterie, frozen_meals, dairy, fish)	4
116	0.054204	(charcuterie, dairy, fish, fruitvegs)	4
117	0.053097	(dairy, canned, beer, charcuterie, fish)	5
118	0.055310	(dairy, frozen_meals, beer, charcuterie, fish)	5

Python: Apriori

```
from mlxtend.frequent_patterns import association_rules

rules = association_rules(freqItemsets, metric="support", min_threshold=0.01)
```

	antecedents	consequents	support	confidence	lift	leverage	conviction
0	(canned)	(beer)	0.184735	0.551155	1.700492	0.076099	1.505832
1	(beer)	(canned)	0.184735	0.569966	1.700492	0.076099	1.545977
2	(charcuterie)	(beer)	0.324115	1.000000	3.085324	0.219064	inf
3	(beer)	(charcuterie)	0.324115	1.000000	3.085324	0.219064	inf
4	(beer)	(clothes)	0.070796	0.218430	0.715438	-0.028159	0.888840
...	...	...	...	...	...	...	...
649	(dairy)	(fish, fruitvegs, beer, charcuterie)	0.054204	0.167808	3.095890	0.036695	1.136513
650	(fruitvegs)	(dairy, fish, beer, charcuterie)	0.054204	0.163880	1.742908	0.023104	1.083544
651	(beer)	(dairy, fruitvegs, charcuterie, fish)	0.054204	0.167235	3.085324	0.036635	1.135731
652	(fish)	(dairy, fruitvegs, beer, charcuterie)	0.054204	0.167808	3.095890	0.036695	1.136513
653	(charcuterie)	(dairy, fruitvegs, beer, fish)	0.054204	0.167235	3.085324	0.036635	1.135731

Avaliação de Regras de Associação

Limitações do Modelo Suporte/Confiança

O Modelo Suporte/Confiança tem recebido muitas críticas ao longo dos últimos anos:

- O número de regras geradas pelo modelo é geralmente muito grande, dificultando o processo de análise por parte do utilizador
- Grande parte dos resultados minerados é composto por regras óbvias, redundantes ou, até mesmo, contraditórias

Medidas Objectivas

- Devem depender apenas dos dados - *data-driven*
- Devem ser independentes do domínio – *domain-independent*
- Devem necessitar do mínimo de informação do utilizador (apenas valores limite para filtrar relações de baixa qualidade)

	B	$\overline{B}$	
A	f_{11}	f_{10}	f_{1+}
$\overline{A}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	N

$\overline{A}$ $(\overline{B})$ Indica a não presença de A (B) na transação

f_{11} indica o nº de vezes que ambos os itens A e B ocorrem

…

f_{01} indica nº transações que contêm B mas não A

Propriedades das Medidas Objectivas

As medidas de avaliação de regras de associação devem obedecer às seguintes propriedades:

P1: Propriedade de Inversão

Uma medida objectiva deve ser constante face à operação inversão, ou seja, por troca dos valores $f_{00} \leftrightarrows f_{11}$ e $f_{01} \leftrightarrows f_{10}$ deve permanecer invariável

P2: Propriedade Adição Nula

Uma medida objectiva deve ser constante face à adição nula, ou seja, não deve ser afectada pelo aumento de f_{00} enquanto as restantes frequências f_{11} , f_{01} , f_{10} permanecem constantes

P3: Propriedade de Escala

Uma medida objectiva deve ser constante relativamente a mudanças de escala, ou seja, se M(T) = M(T') em que T é uma tabela de contigência com as frequências $[f_{11},f_{01},f_{10},f_{00}]$ e T' é uma tabela de contigência com as frequências $[k_1k_3f_{11}, k_2k_3f_{01}, k_1k_4f_{10}, k_2k_4f_{00}]$

Regras Fortes Não Interessantes

	Café	$\overline{\text{Café}}$	
Chá	15	5	20
$\overline{\text{Chá}}$	75	5	80
	90	10	100

Chá $\Rightarrow$ Café

Chá $\Rightarrow$ Café Sup (15/100)=15%, Conf (15/20)=75%

- Estes valores relativamente altos podem induzir que quem compra chá também compra café

mas

Sup (Café) **90%** > Conf(Chá $\Rightarrow$ Café) **75%**

➥ **regra enganadora**

- A medida Confiança não tem em conta o Suporte dos itens do lado direito da regra

Outras Medidas: Interesse [Brin S. *et al*]

Interesse (Lift)

$$Interesse\ (A \Rightarrow B) = \frac{Conf\,(A \Rightarrow B)}{Sup(B)} = \frac{Sup(A,B)}{Sup(A) \times Sup(B)} \qquad [0..+\infty[$$

Interesse (A,B) $\begin{cases} = 1, & \text{A e B são independentes} \quad Sup(A,B) = S(A) \times S(B) \\ > 1, & \textbf{A e B são positivamente correlacionados} \\ < 1, & \text{A e B são negativamente correlacionados} \end{cases}$

Interesse é uma medida que indica o quão mais provável é um item ser comprado em relação à sua taxa de compra típica, uma vez que se sabe que outro item foi comprado

Um grande valor de Interesse é, portanto, um forte indicador de que uma regra é importante, e reflete uma verdadeira conexão entre os itens

Interesse

	Café	$\overline{\text{Café}}$	
Chá	15	5	20
$\overline{\text{Chá}}$	75	5	80
	90	10	100

Chá $\Rightarrow$ Café

Sup (15/100)=15%,
Conf (15/20)=75%

Interesse (Chá $\Rightarrow$ Café) = 0.8333 < 1

➥ negativamente correlacionados
Chá e Café são itens concorrentes

Outras Medidas: Leverage [Piatestky-Shapiro 1991]

Leverage (Influência)

$$Leverage\ (A \Rightarrow B) = Sup\ (A \Rightarrow B) - SupEsp\ (A \Rightarrow B)$$

$$\text{Leverage}(A \Rightarrow B) = Sup(A \Rightarrow B) - Sup(A) \times Sup(B) \quad [-0{,}25 \ldots +0{,}25]$$

- Esta medida indica o valor da diferença entre o suporte real e o suporte esperado de uma regra de associação
- Quanto maior o valor da medida, mais interessante é a regra

Leverage (A,B) {
- = 0, A e B são independentes S(A,B) = S(A)×S(B)
- > 0, **A e B são positivamente correlacionadas**
- < 0, A e B são negativamente correlacionadas

Leverage (Chá ⇒ Café) = -0,03 < 1 → negativamente correlacionados

Outras Medidas: Conviction [Brin et al. 1997]

- As medidas *Interesse e Leverage possuem como característica o* facto de serem medidas simétricas:

$$Interesse\ (A \Rightarrow B) = Interesse\ (B \Rightarrow A)$$
$$Leverage\ (A \Rightarrow B) = Leverage\ (B \Rightarrow A)$$

- Estes índices medem a dependência entre os itens, e não a implicação

- O principal objectivo da medida Convinction é **avaliar** a **força da associação**

$$Convinction\ (A \Rightarrow B) = \frac{1 - Sup(B)}{1 - Conf(A \Rightarrow B)} \quad [0.5,\ \ldots,\ +\infty[$$

- Valores da medida Conviction: 0.5, …, 1, …, ∞
  - Convinction = 1, A e B são independentes
  - 1,01 < Convinction < 5 Regras mais interessantes
  - Convinction > 5 .. ∞ Regras óbvias

Convinction (Chá ⇒ Café) = 0,4 → associação fraca