Aula 14 – Mineração de regras de associação

1001524 – Aprendizado de Máquina I

2023/1 - Turmas A, B e C

Prof. Dr. Murilo Naldi

naldi@ufscar.br

Agradecimentos

- Parte do material utilizado nesta aula foi cedido pelo professor Diego Silva e, portanto, os agradecimentos

Frequent itemset mining

Problema clássico: análise da lista de compras

- Quais itens são frequentemente comprados juntos?
- Isso pode ser utilizado, por exemplo, para:
  - Deixar produtos na mesma prateleira ou corredor
  - Fazer promoções específicas

Frequent itemset mining

Por exemplo:

Frequent itemset mining

Outro exemplo:

Mas é sempre “óbvio” assim?

Fato ou *fake*?

Exemplo comum em livros didáticos

- Aos fins de semana, um cliente que compra cerveja também compra… fraldas!
- Esses são os “pais-de-família” que vão ao mercado comprar fraldas e aproveitam para comprar a cerveja
  - Ou o contrário
- Nesse caso, deixar um dos produtos no caminho entre o outro produto e o caixa pode aumentar o lucro

Frequent itemset mining

Outros exemplos

- Páginas de um portal acessadas na mesma seção
- Minerais encontrados na mesma região
- Pessoas que vão aos mesmos eventos
- Filmes que são assistidos pelos mesmos usuários
- etc

Outro exemplo

Famoso caso do Walmart

- Há anos, a rede de mercados decidiu minerar seus dados
- Encontrou diversos padrões interessantes e não tão óbvios:
  - Há regiões nos EUA com vários casos de tornado
  - Analisando esses períodos, o Walmart percebeu que havia em muitas compras um conjunto de produtos “óbvios”, mas também percebeu...

Outro exemplo

Pop tarts de morango!

A história conta que aumentar a disponibilidade do produto quando há alerta de tornado aumenta sua venda em 7 vezes

Exemplo ilustrativo

Transação	Item 1	Item 2	Item 3
1	Açúcar	Leite	Café
2	Açúcar	Leite	Café
3	Açúcar	Leite	Café
4	Açúcar	Leite	
5	Açúcar	Leite	

Exemplo ilustrativo

Transação	Item 1	Item 2	Item 3
1	Açúcar	Leite	Café
2	Açúcar	Leite	Café
3	Açúcar	Leite	Café
4	Açúcar	Leite	
5	Açúcar	Leite	

Se açúcar é comprado, **então** leite também é comprado.
Se açúcar e leite são comprados, **então** café também é comprado (em 60% das transações).
Etc

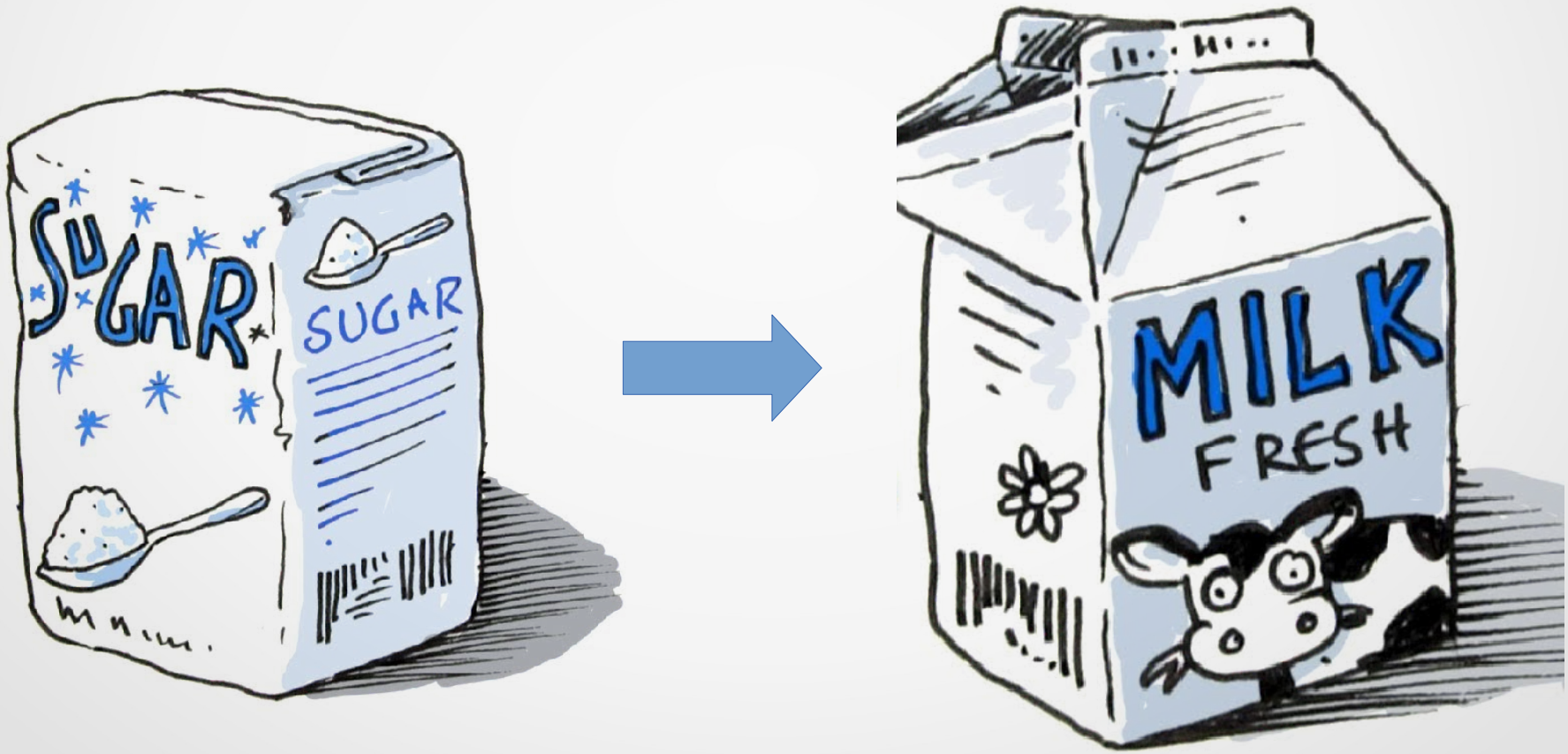
Regras

Nesse caso, chamamos os termos ***“se-então”*** de regras

- Ainda, dizemos que o “se” é o lado esquerdo
- O “então” representa o lado direito

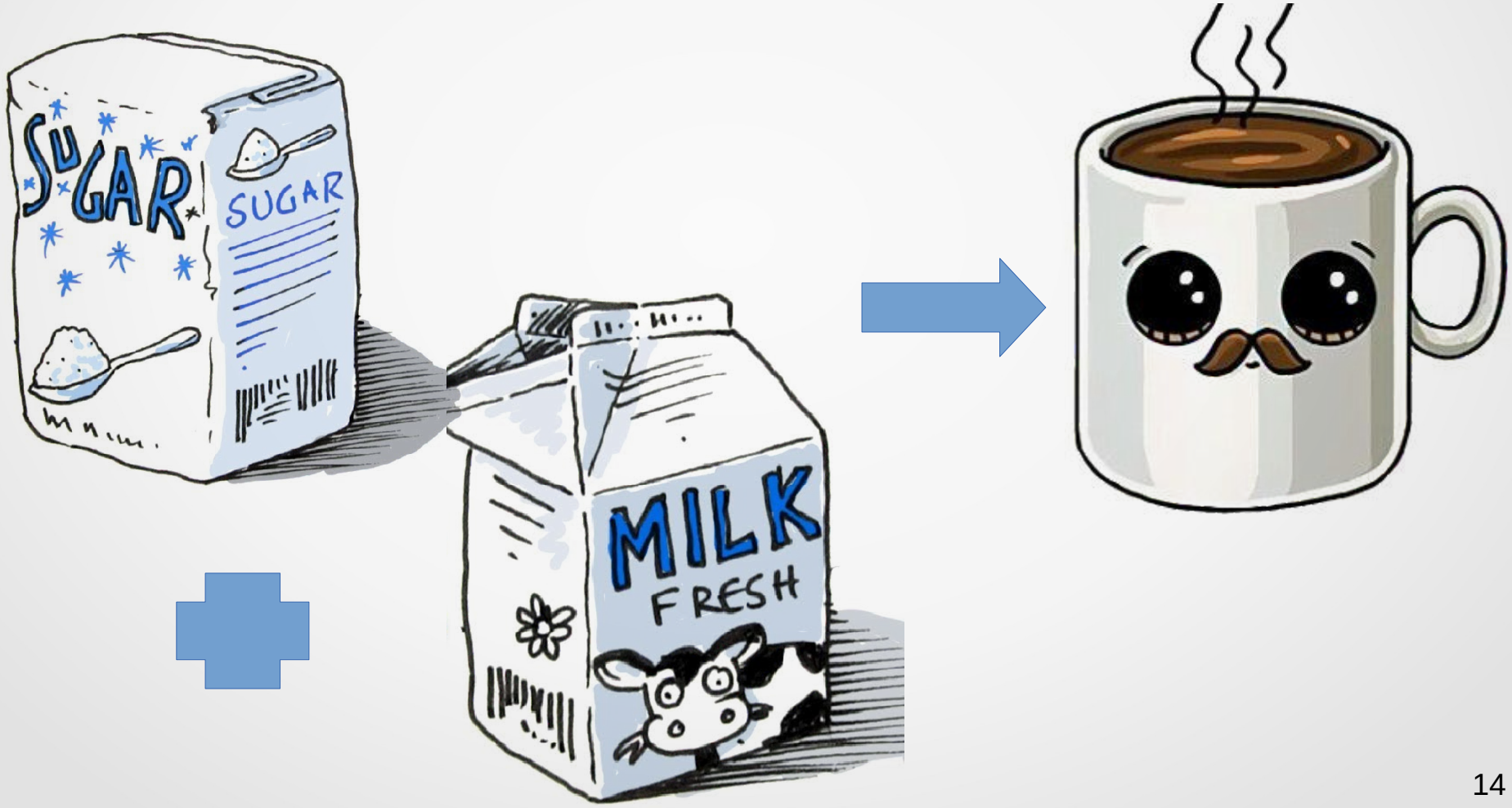
Exemplos de regras

- {Açúcar} ⇒ {Leite}



Exemplos de regras

- {Açúcar, Leite} ⇒ {Café}



Frequent itemset mining

O nome dado ao conjunto de itens que compõe um lado da regra é chamado de… conjunto de itens.

- Mais comum em inglês mesmo: *itemset*

Frequent itemset mining

Formalizando:

- $\mathbf{A} = \{\mathbf{a_1}, \mathbf{a_2}, \ldots, \mathbf{a_m}\}$ é o universo de *m* itens
- $\mathbf{I} \subseteq \mathbf{A}$ é um *itemset*
- $\mathbf{T} = (\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_n})$ é um conjunto de *n* transações
- Cada transação é um par $\langle \mathbf{tid_i}, \mathbf{itens_i} \rangle$
- $\mathbf{itens_i} \subseteq \mathbf{A}$ e $|\mathbf{itens_i}| = k$

Frequent itemset mining

Uma transação **t** ∈ **T** suporta (dá **suporte**) um *itemset* **I** se **I** ⊆ **t**.

- Ou seja, **t** contém todos os elementos de **I**
- Exemplo: **I** = {a,b,d} e **t** = {a,b,c,d}

Intuitivamente: dar suporte significa “testemunhar a favor”

Frequent itemset mining

O conjunto de transações K(I) que suporta I é dito conjunto suporte do *itemset*.

Frequent itemset mining

Suporte absoluto ($\mathbf{s_T}$) e relativo (σ_T)

$$s_T(I) = |K_T(I)|$$

ou seja, número de transações que suportam I

$$\sigma_T(I) = \frac{1}{n} s_T(I)$$

O problema de encontrar *itemset* frequentes pode ser definido por:

Frequent itemset mining

- Dados:
  - Um conjunto A de itens, uma tabela T de transações sobre A e um número $0 < \sigma_{min} \leq 1$ (**suporte mínimo**)
- Encontrar:
  - Conjunto de *itemsets* com suporte maior ou igual a um valor σ_{min} definido pelo usuário
  - Conjunto de regras de associações com **confiança** maior do que a estabelecida pelo usuário

Frequent itemset mining

TID	Itens
1	{a,d,e}
2	{b,c,d}
3	{a,c,e}
4	{a,c,d,e}
5	{a,e}
6	{a,c,d}
7	{b,c}
8	{a,c,d,e}
9	{b,c,e}
10	{a,d,e}

0 itens	1 item	2 itens	3 itens
Ø: 10	{a}: 7	{a,c}: 4	{a,c,d}: 3
	{b}: 3	{a,d}: 5	{a,c,e}: 3
	{c}: 7	{a,e}: 6	{a,d,e}: 4
	{d}: 6	{b,c}: 3	
	{e}: 7	{c,d}: 4	
		{c,e}: 4	
		{d,e}: 4	

Figura 10.1 *Um banco de dados de transações, com 10 transações, e a enumeração de todos os conjuntos de itens frequentes usando o suporte mínimo de* $s_{\min} = 3$.

Espaço de busca

O espaço de busca para o conjunto **A** tem $2^{|\mathbf{A}|}$ *itemsets*

- Devemos reduzir o espaço
- Usualmente, considera-se o fato que
  - Seja **X, Y** $\subseteq$ **I**
  - Se **X** $\subseteq$ **Y** $\Rightarrow$ suporte(**Y**) $\leq$ suporte(**X**)
  - Portanto, se um *itemset* é pouco frequente, todos seus superconjuntos também serão

Espaço de busca

Em outras palavras: adicionar um item no *itemset* diminui seu suporte. Portanto o suporte é monotonicamente não crescente ao se fazer seguidas operações desse tipo.

Espaço de busca

TID	Item set
1	{a,d,e}
2	{b,c,d}
3	{a,c,e}
4	{a,c,d,e}
5	{a,e}
6	{a,c,d}
7	{b,c}
8	{a,c,d,e}
9	{b,c,e}
10	{a,d,e}

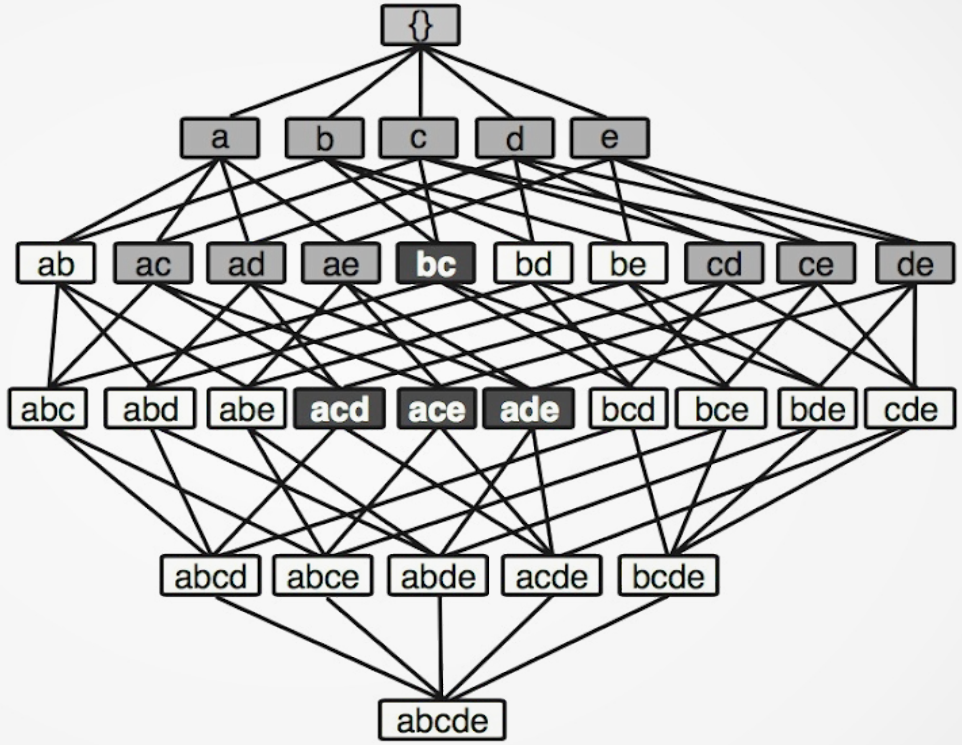


Figura 10.2 *Um banco de dados de transações, com 10 transações, e o espaço de busca para encontrar todos os possíveis itemsets frequentes usando o suporte mínimo de* $s_{\min} = 3$.

Apriori

Esse fato é utilizado pelo primeiro (e mais conhecido) algoritmo de descoberta de *itemsets* frequentes: ***Apriori***

- Baseado em busca em largura
- A cada nível, se cria *itemsets* a partir dos *itemsets* do nível anterior
- A frequência é novamente calculada para os novos *itemsets*

Apriori

Inicialmente, cada elemento é um *itemset* (candidato) unitário

- Os *itemsets* de tamanho k+1 são obtidos a partir dos *itemsets* de tamanho k
- Isso é obtido pela combinação do k-ésimo conjunto de candidatos ($\mathbf{F}_k$) com ele mesmo

Apriori

- **X** U **Y** dos itens **X**, **Y** $\in$ $\mathbf{F}_k$ é gerada se eles têm o mesmo prefixo de *k*-1 elementos
  - Fácil de se obter se os itens forem mantidos em ordem
- Varre-se o banco recalculando os suportes e inserindo os *itemsets* com suporte mínimo em $\boldsymbol{F}_{k+1}$

Exemplo - Apriori

- Dado o conjunto de itens, selecionamos os com suporte mínimo 0.4

	support	itemsets	length
0	0.444444	(Beer)	1
1	0.777778	(Diaper)	1
2	0.222222	(Gum)	1
3	0.555556	(Soda)	1
4	0.444444	(Snack)	1

Exemplo - Apriori

- Selecionados

	support	itemsets	length
0	0.444444	(Beer)	1
1	0.777778	(Diaper)	1
2	0.555556	(Soda)	1
3	0.444444	(Snack)	1

Exemplo - Apriori

- Repete cálculo para os pares com suporte mínimo 0.4

	support	itemsets	length
0	0.444444	(Diaper, Beer)	2
1	0.222222	(Soda, Beer)	2
2	0.222222	(Snack, Beer)	2
3	0.333333	(Soda, Diaper)	2
4	0.222222	(Snack, Diaper)	2
5	0.333333	(Soda, Snack)	2

Exemplo - Apriori

- Repete cálculo para os pares com suporte mínimo 0.4

	support	itemsets	length
0	0.444444	(Diaper, Beer)	2
1	0.222222	(Soda, Beer)	2
2	0.222222	(Snack, Beer)	2
3	0.333333	(Soda, Diaper)	2
4	0.222222	(Snack, Diaper)	2
5	0.333333	(Soda, Snack)	2

Exemplo - Apriori

- Sobraram com suporte mínimo 0.4

	support	itemsets	length
0	0.444444	(Beer)	1
1	0.777778	(Diaper)	1
2	0.555556	(Soda)	1
3	0.444444	(Snack)	1
4	0.444444	(Diaper, Beer)	2

Exemplo - Apriori

- Combinação dos *itemsets* que sobraram não gera nenhuma combinação com suporte mínimo de 0.4
  - O algoritmo pára

	support	itemsets	length
0	0.222222	(Soda, Diaper, Beer)	3
1	0.222222	(Snack, Diaper, Beer)	3

Exemplo - Apriori

- Resultado:

	support	itemsets	length
0	0.444444	(Beer)	1
1	0.777778	(Diaper)	1
2	0.555556	(Soda)	1
3	0.444444	(Snack)	1
4	0.444444	(Diaper, Beer)	2

Exemplo - Apriori

TID	Item set
1	{a,d,e}
2	{b,c,d}
3	{a,c,e}
4	{a,c,d,e}
5	{a,e}
6	{a,c,d}
7	{b,c}
8	{a,c,d,e}
9	{b,c,e}
10	{a,d,e}

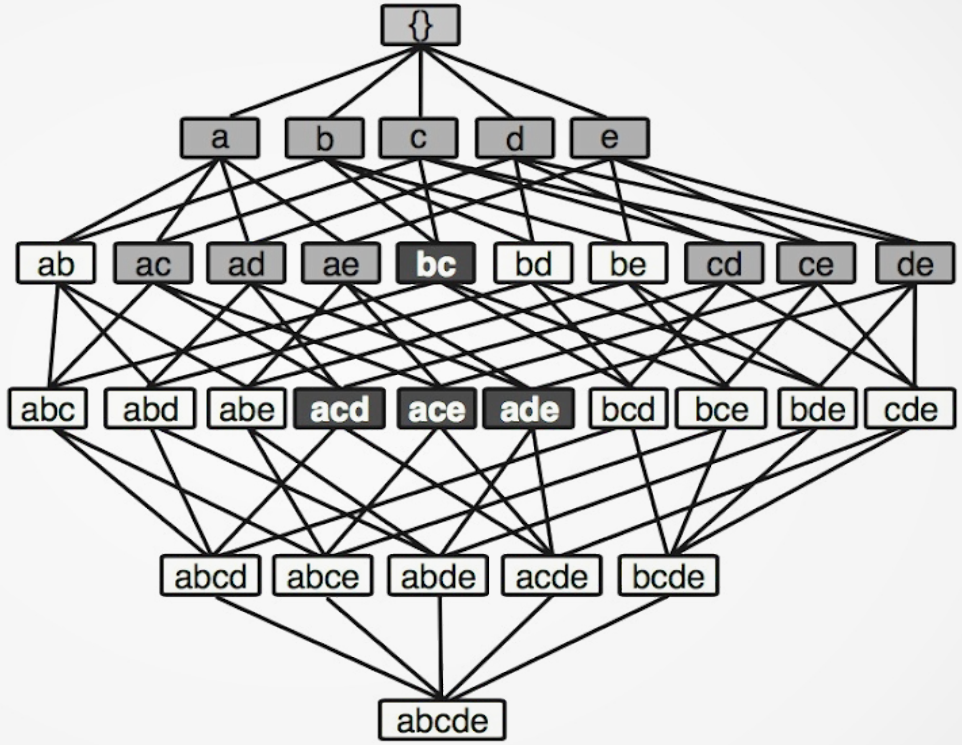


Figura 10.2 *Um banco de dados de transações, com 10 transações, e o espaço de busca para encontrar todos os possíveis itemsets frequentes usando o suporte mínimo de* $s_{\min} = 3$.

Apriori - Regras de associação

Além de encontrar *itemsets* frequentes, o Apriori tem um segundo passo para encontrar regras de associação

- Combina-se termos para construir regras da forma A→ B, tal que A U B é um *itemset* frequente
- O grau de interesse da regra é dado pela sua confiança

$$confiança(A \rightarrow B) = \frac{P(A \cup B)}{P(A)} = \frac{suporte(A \cup B)}{suporte(A)}$$

Apriori - Regras de associação

- O algoritmo gera todos os subconjuntos não vazios do *itemset*
- Para cada subconjunto s, verifica a confiança da regra dada por s → {l\s}

Apriori - Regras de associação

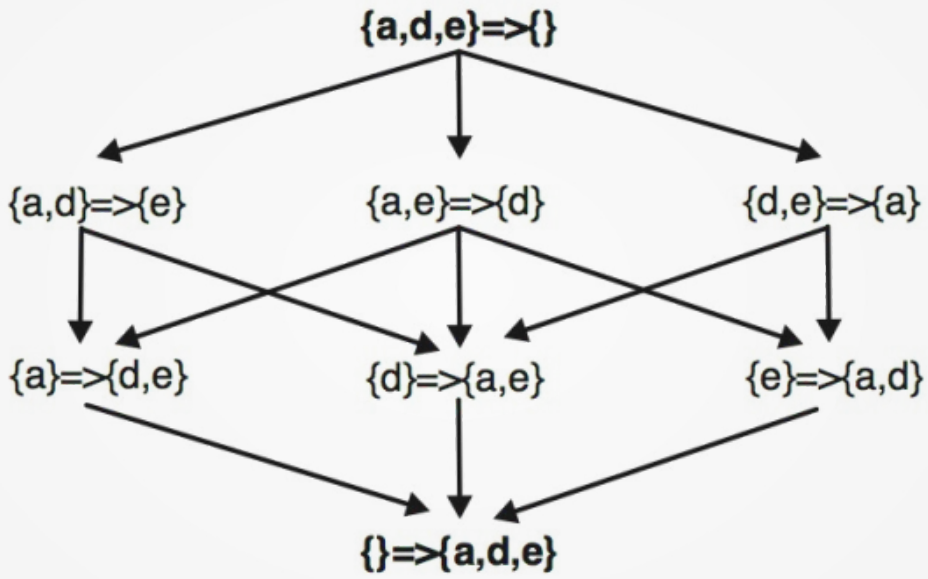


Figura 10.3 *Espaço de busca de regras de associação para um itemset frequente.*

Apriori - Regras de associação

Há diversas soluções para melhorar a eficiência da solução

- O livro da Katti apresenta o FP-Growth
  - Utiliza uma trie (árvore de sufixos)

Mas há outro problema:
número excessivo de regras

Pós-processamento

Vamos pós-processar a base de regras

- Ex: eliminar *itemsets* que são subconjuntos de outros *itemsets* frequentes

Pós-processamento

Algumas definições:

- *Itemset* frequente maximal: se é frequente, mas nenhum dos seus superconjuntos é frequente
- *Itemset* frequente fechado: se e somente se ele não possui superconjuntos com a mesma frequência

Pós- Processamento

- *itemsets* frequentes maximais

TID	Itens
1	{a,d,e}
2	{b,c,d}
3	{a,c,e}
4	{a,c,d,e}
5	{a,e}
6	{a,c,d}
7	{b,c}
8	{a,c,d,e}
9	{b,c,e}
10	{a,d,e}

0 itens	1 item	2 itens	3 itens
Ø: 10	{a}: 7	{a,c}: 4	{a,c,d}: 3
	{b}: 3	{a,d}: 5	{a,c,e}: 3
	{c}: 7	{a,e}: 6	{a,d,e}: 4
	{d}: 6	{b,c}: 3	
	{e}: 7	{c,d}: 4	
		{c,e}: 4	
		{d,e}: 4	

Pós- Processamento

- *itemsets* frequentes fechados

TID	Itens
1	{a,d,e}
2	{b,c,d}
3	{a,c,e}
4	{a,c,d,e}
5	{a,e}
6	{a,c,d}
7	{b,c}
8	{a,c,d,e}
9	{b,c,e}
10	{a,d,e}

0 itens	1 item	2 itens	3 itens
Ø: 10	{a}: 7	{a,c}: 4	{a,c,d}: 3
	{b}: 3	{a,d}: 5	{a,c,e}: 3
	{c}: 7	{a,e}: 6	{a,d,e}: 4
	{d}: 6	{b,c}: 3	
	{e}: 7	{c,d}: 4	
		{c,e}: 4	
		{d,e}: 4	

Pós- Processamento

- Exemplo: avaliar a regra {chá} → {café}
  - Suporte = 150/1000 = 0,15
  - Confiança = 0,15/0,2 = 0,75

Tabela 10.2 *Preferências sobre o consumo de Chá e Café de 1000 consumidores*

	Café	Não Café	Total
Chá	150	50	200
Não Chá	650	150	800
Total	800	200	1000

Porém, a probabilidade de alguém tomar café (independente do chá) é 80%. Portanto, a regra é enganosa.

Coeficiente de interesse (lift)

$$Lift(A \rightarrow B) = \frac{confiança(A \rightarrow B)}{suporte(B)} = \frac{suporte(A \cup B)}{suporte(A) \times suporte(B)}$$

- Lift = 1 significa que A e B são independentes
- Valores menores que um indicam correlação negativa
- Valores maiores que um, correlação positiva
  - No exemplo, Lift({chá} → {café}) = 0,9375 (< 1)

Coeficiente de interesse (lift)

$$Lift(A \rightarrow B) = \frac{confiança(A \rightarrow B)}{suporte(B)} = \frac{suporte(A \cup B)}{suporte(A) \times suporte(B)}$$

Quociente entre confiança e valor esperado para a confiança

- Lift = 1 significa que A e B são independentes
- Valores menores que um indicam correlação negativa
- Valores maiores que um, correlação positiva

  - No exemplo, Lift({chá} → {café}) = 0,9375 (< 1)

Coeficiente de interesse (lift)

Em outras palavras:

- > 1 indica que A tem efeito positivo sobre a ocorrência de B e, portanto, tendem a aparecer junto quando A ocorre
- < 1 indica o contrário, ou seja, A tem efeito negativo sobre a ocorrência de B, inibindo-a
- = 1 indica que A não influencia na ocorrência de B e que eles ocorrem independentemente

Convicção

$$convicção(A \rightarrow B) = \frac{1 - suporte(B)}{1 - confiança(A \rightarrow B)}$$

Também demonstra efeito negativo da regra quando seu valor é menor que 1. Tende a infinito quando a confiança da regra tende a 1. Zero indica que B está em todas as transações.

No exemplo, convicção({chá} → {café}) = 0,8 (< 1)

Convicção

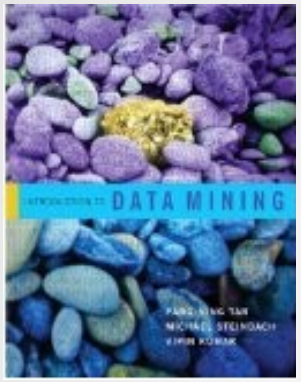
$$convicção(A \rightarrow B) = \frac{1 - suporte(B)}{1 - confiança(A \rightarrow B)}$$

Quociente da frequência esperada de A ocorrer sem B

Também demonstra efeito negativo da regra quando seu valor é menor que 1. Tende a infinito quando a confiança da regra tende a 1. Zero indica que B está em todas as transações.

No exemplo, convicção({chá} → {café}) = 0,8 (< 1)

Bibliografia



V. TAN, STEINBACH, M., KUMAR, P. Introdução ao Data Mining (Mineração de Dados). Edição 1. Ciência Moderna 2009. ISBN 9788573937619.

Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina. Katti Faceli, Ana Carolina Lorena, João Gama, André C. P. L. F. de Carvalho. Grupo Gen 2011

Quinlan, J. R., C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993