Data Mining

algoritmo **apriori**

Jomi F. Hübner

Universidade Federal de Santa Catarina
Departamento de Automação e Sistemas
http://jomi.das.ufsc.br

Apriori

- algoritmo simples de *data mining* em banco de dados com transações
- valor histórico

- entrada: transações (conjunto de itens comprados)
- saída *regras* de associação
  $\rightsquigarrow$ "quem compra pepino e hamburger também compra cerveja"

Exemplo de Banco de Dados

Transação	Pepino	Batata	Hamburger	Leite	Cerveja
t_1	1	1	1	0	0
t_2	0	1	1	1	0
t_3	0	0	0	1	1
t_4	1	1	0	1	0
t_5	1	1	1	0	1
t_6	1	1	1	1	1

[fonte]

Notação

- I: conjunto de itens
- D: conjunto de transações,
  cada transação $t \in D$ é um conjunto de itens ($t \subseteq I$)
- suporte a um conjunto $x \subset I$ de itens

$$supp(x) = \frac{|\{t \mid x \subseteq t \wedge t \in D\}|}{|D|}$$

- $supp(\{b\}) = 5/6$ (b = batata)
- $supp(\{b, p\}) = 4/6$ (p = pepino)
- $supp(\{b, p, l\}) = 2/6$ (l = leite)
- $supp(\{b, p, l, c\}) = 1/6$ (c = cerveja)

Algoritmo

1 **function** apriori(D, ϵ)

Input: conjunto de transações D; frequência mínima ϵ.
Output: cjto de itemsets frequentes
Data: C_k: itemsets candidatos; L_k: itemsets frequentes de tamanho k.

2 $L_1 \leftarrow \{\{i\} \mid i \in I \wedge supp(\{i\}) \geq \epsilon\}$
3 $k \leftarrow 2$
4 **while** $L_{k-1} \neq \emptyset$ **do**
5 $\quad C_k \leftarrow \{a \cup b \mid a \in L_{k-1} \ \wedge \ b \in L_{k-1}\}^a$
6 $\quad L_k \leftarrow \{c \mid c \in C_k \ \wedge \ |c| = k \ \wedge \ supp(c) \geq \epsilon\}$
7 $\quad k \leftarrow k + 1$
8 **return** $\bigcup_k L_k$

[a] assume-se que se um cjto é frequente, seus sub-cjtos também são. Por isso C_k pode ser construídos a partir de L_{k-1}.

Regras

- Uma regra $X \rightarrow Y$ é um par onde $X \subset I$, $Y \subset I$, $X \cap Y = \emptyset$
- A confiança em uma regra é dada por

$$conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$$

- Exemplo:

$$conf(\{p, b\} \rightarrow \{h\}) = \frac{supp(\{p, b, h\})}{supp(\{p, b\})} = \frac{3/6}{4/6} = 0,75$$

Em 75% das compras de pepino e batata, se comprou hamburger

Geração de Regras

$$\{< X, Y > \mid X \subset L \ \wedge \ X \neq \emptyset \wedge \quad (1)$$
$$Y \subset L \ \wedge \ Y \neq \emptyset \wedge \quad (2)$$
$$X \cap Y = \emptyset \wedge \quad (3)$$
$$conf(X \rightarrow Y) \geq \delta\} \quad (4)$$
$$L \in apriori(D, \epsilon) \quad (5)$$

para $L = \{p, b, h\}$ temos:

$$pb \rightarrow h, ph \rightarrow b, hb \rightarrow p, \ p \rightarrow bh, b \rightarrow ph, h \rightarrow pb$$

regras potenciais:

$$\sum_{L \in apriori(D,\epsilon)} 2^{|L|} - 2$$

Resumo

- processo bottom-up
- lento
- gera muitas regras
- funciona com valores discretos

- tem várias extensões que mitigam essas limitações