Introdução a Sistemas Inteligentes

Noções de Preparação de Dados e Mineração de Regras de Associação

Prof. Ricardo J. G. B. Campello

ICMC / USP

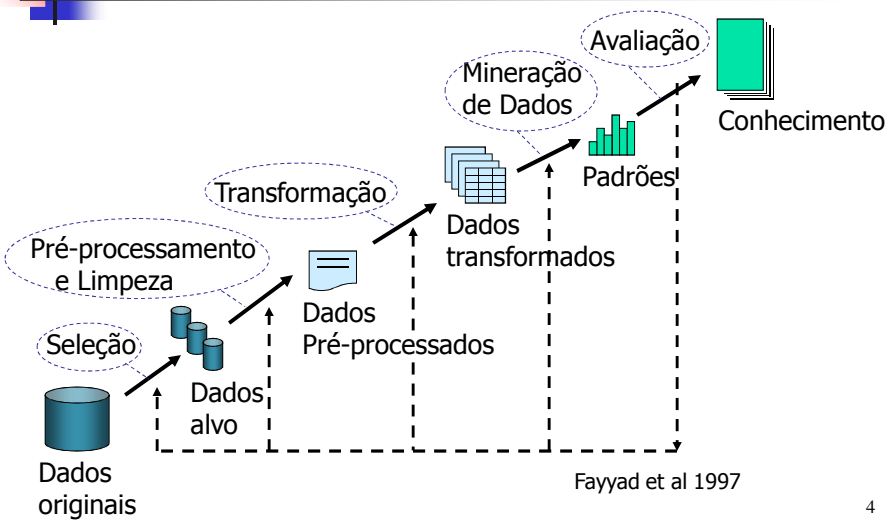
Créditos

- Parte deste material consiste de adaptações e extensões dos originais:
  - gentilmente cedidos pelos professores Eduardo R. Hruschka (baseados no curso de Gregory Piatetsky-Shapiro, disponível no sítio http://www.kdnuggets.com) e André C. P. L. F. de Carvalho
  - do livro de (Tan et al., 2006)

Aula de Hoje

- Noções de pré-processamento de dados
- Introdução à mineração de regras de associação
  - Medidas de suporte e confiança
  - Princípio "Apriori"
  - Regras de associação

Relembrando KDD...



Fayyad et al 1997

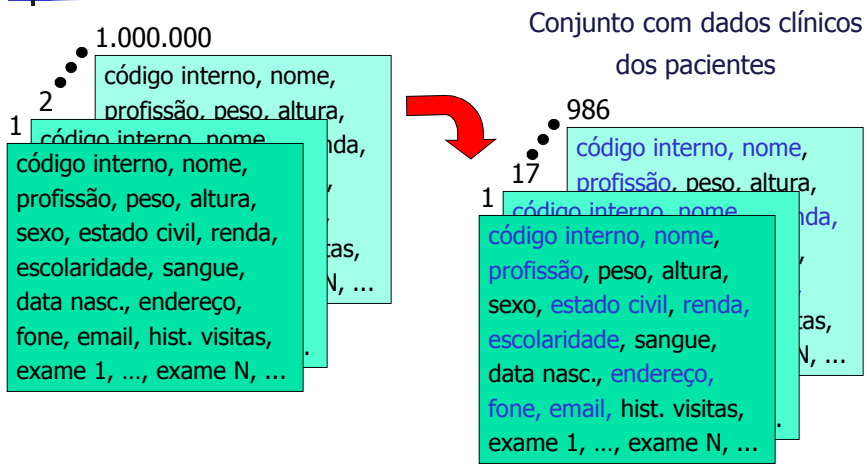
Preparação de Dados

- Tornar os dados adequados para utilização na etapa de mineração
- Envolve basicamente as etapas de:
  - Seleção
  - Limpeza e Pré-Processamento
  - Transformação

Seleção

- Toma-se um subconjunto de interesse dentre os dados disponíveis
  - Subconjunto de registros
    - exemplos, instâncias ou objetos
  - Subconjunto de atributos
    - campos, variáveis ou características
    - considerados relevantes para o problema
      - os demais são claramente irrelevantes e descartados "manualmente"

Exemplo



Pré-Processamento e Limpeza

- Melhorar a qualidade dos dados e facilitar sua posterior utilização
- As principais operações são
  - Eliminar dados duplicados
  - Agregar dados
  - Lidar com valores ausentes
  - Lidar com ruído

Transformações

- Inclui diversas operações, tais como:
  - Seleção "automática" de atributos
    - p/ eliminar atributos irrelevantes e/ou redundantes
  - Extração de características
    - p/ obter um menor no. de atributos mais relevantes a partir dos dados brutos
  - Discretizações e conversões
    - entre atributos de diferentes naturezas
  - Normalizações
    - para que os atributos ou mesmo os objetos apresentem determinadas propriedades (usualmente estatísticas) de interesse

Exemplo

- Normalizações:
  - Re-escalonamento
    - Converter todos os valores de um atributo para [0, 1] ou [-1,+1]
  - Padronização
    - Fazer com que cada atributo possua média nula e variância 1
- Podem ser fundamentais em algoritmos que veremos
  - p. ex. KNN

$$x' = \frac{(x - \min_x)}{(\max_x - \min_x)}$$

$$x' = \frac{x - \mu_x}{\sigma_x}$$

Análise de Associação

- Descoberta e interpretação de padrões de inter-relacionamento interessantes que podem estar escondidos em grandes bases de dados de "transações"

Exemplos de Transações:

ID	Produto
1	leite, pão, ovos
2	pão, açúcar
3	pão, cereal
4	leite, pão, açúcar
5	leite, cereal
6	pão, cereal
7	leite, cereal
8	leite, pão, cereal, ovos
9	leite, pão, cereal

Análise de Associação

- Variados Campos de Aplicação
- Por exemplo:
  - Mercados: relações entre produtos, perfis de consumo, etc.
  - Meteorologia: relações entre fenômenos atmosféricos, terrestres, marítimos, etc.
  - Medicina: relações entre exames, sintomas, doenças, etc.
  - Bioinformática...

Exemplo de Base de Dados de Transações:

T	Produtos
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, C

Exemplos = Transações

Itens:

A = leite
B= pão
C= cereal
D= açúcar
E= ovos

Exemplo de Base de Dados de Transações:

T	Produtos
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, C

Produtos convertidos em atributos binários assimétricos:

T	A	B	C	D	E
1	1	1	0	0	1
2	0	1	0	1	0
3	0	1	1	0	0
4	1	1	0	1	0
5	1	0	1	0	0
6	0	1	1	0	0
7	1	0	1	0	0
8	1	1	1	0	1
9	1	1	1	0	0

NOTA

- Todas as discussões a seguir presumem que as bases de dados de transações em questão são **binárias assimétricas**
- Métodos para análise de associação sobre bases não binárias, tais como, por exemplo, aquelas contendo transações de consumo com a **quantidade** e/ou **preço** dos itens envolvidos, estão além do escopo deste curso.

Definições:

- **Conjunto de Itens (*itemset*) I**: um subconjunto de itens possíveis
  - Exemplo: I = {A, B, E} (ordem não é importante)
  - Pode ser parte de (ou toda) uma **transação** (t)
- **Suporte(I)** = nº de transações t que contêm I
  - Na base de dados anterior temos que: sup({A, B, E}) = 2, sup({B, C}) = 4
- **Conjunto de Itens Freqüentes**: sup(I) ≥ *sup_mín*, onde *sup_mín* é o suporte mínimo, definido pelo usuário

Propriedade do Subconjunto:

- **Todo subconjunto de um conjunto freqüente é também freqüente**
- Por quê?
- Exemplo: suponhamos que {A,B} seja freqüente. Dado que cada ocorrência de {A,B} inclui A e B, então A e B tem de ser eles próprios freqüentes
- Quase todos os algoritmos para extrair regras de associação são baseados nesta propriedade

Regras de Associação:

- **Regra de Associação R** :
  - Conjunto de itens 1 $\Rightarrow$ Conjunto de itens 2
    - Conjuntos de itens 1 e 2 disjuntos
  - **Interpretação:** se determinada transação inclui o conjunto de itens 1, então esta também inclui (ou provavelmente inclui) o conjunto de itens 2
    - Cuidado! Não deve ser interpretada como causa – efeito!
- Exemplos:
  - A,B $\Rightarrow$ E,C
  - A $\Rightarrow$ B,C

Como Obter Regras de Associação?

- *Dado um conjunto de itens freqüentes {A, B, E}, quais são as possíveis regras de associação?*
  - A $\Rightarrow$ {B, E}
  - {A, B} $\Rightarrow$ E
  - {A, E} $\Rightarrow$ B
  - B $\Rightarrow$ {A, E}
  - {B, E} $\Rightarrow$ A
  - E $\Rightarrow$ {A, B}

Suporte e Confiança:

- Seja R: I $\Rightarrow$ J uma regra de associação
  - sup(R) = sup(I $\cup$ J) / N
    - N = No de transações (fixo)
    - Logo, a união dos conjuntos I $\cup$ J define o suporte
  - conf(R) = sup(I $\cup$ J) / sup(I) é a confiança de R
    - Número de transações que possuem I e J dividido pelo número de transações que possuem I
- Regras de associação com mínimo suporte são às vezes chamadas de "regras fortes"

Exemplo de Regras de Associação Formadas por 3 Itens:

- ***Dado um conjunto de itens {A,B,E} com suporte = 2, quais regras de associação possuem conf_mín = 50% ?***

  A, B $\Rightarrow$ E : conf=2/4 = 50%

  A, E $\Rightarrow$ B : conf=2/2 = 100%

  B, E $\Rightarrow$ A : conf=2/2 = 100%

  E $\Rightarrow$ A, B : conf=2/2 = 100%

T	Lista de itens
1	**A, B, E**
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	**A, B, C, E**
9	A, B, C

Confiança é menor do que a mínima requerida:

A $\Rightarrow$ B, E : conf=2/6=33% < 50%

B $\Rightarrow$ A, E : conf=2/7=28% < 50%

Exemplo (Freitas & Lavington):

Se "café" então "pão"; Suporte=0,3 / Confiança=1

Se "café" então "manteiga"; Suporte=0,3 / Confiança=1

Se "pão" então "manteiga"; Suporte=0,4 / Confiança=0,8

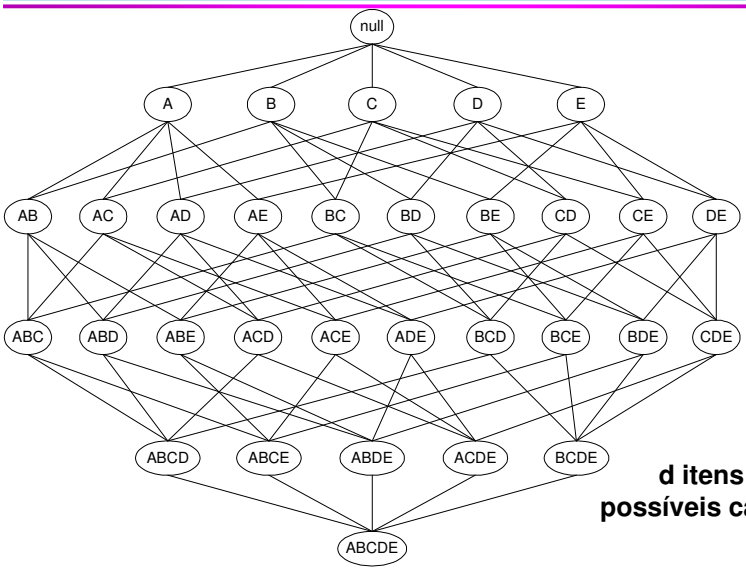
Se "café E pão" então "manteiga"; Suporte=0,3 / Confiança=1

R	Leite	Café	Cerveja	Pão	Manteiga	Arroz	Feijão
1	N	S	N	S	S	N	N
2	S	N	S	S	S	N	N
3	N	S	N	S	S	N	N
4	S	S	N	S	S	N	N
5	N	N	S	N	N	N	N
6	N	N	N	N	S	N	N
7	N	N	N	S	N	N	N
8	N	N	N	N	N	N	S
9	N	N	N	N	N	S	S
10	N	N	N	N	N	S	N

Encontrando Regras de Associação:

- Uma regra possui dois parâmetros: *sup_mín* e *conf_mín*;
  - sup(R) ≥ *sup_mín* & conf(R) ≥ *conf_mín*
- Problema:
  - Encontrar todas as regras que forneçam *sup_mín* e *conf_mín* pré-estabelecidos
- Inicialmente, encontrar todos os conjuntos de itens freqüentes
- Em seguida, extrair regras com elevada confiança a partir desses conjuntos

Geração de Itens Frequentes



d itens ⇒ 2^d possíveis candidatos ...

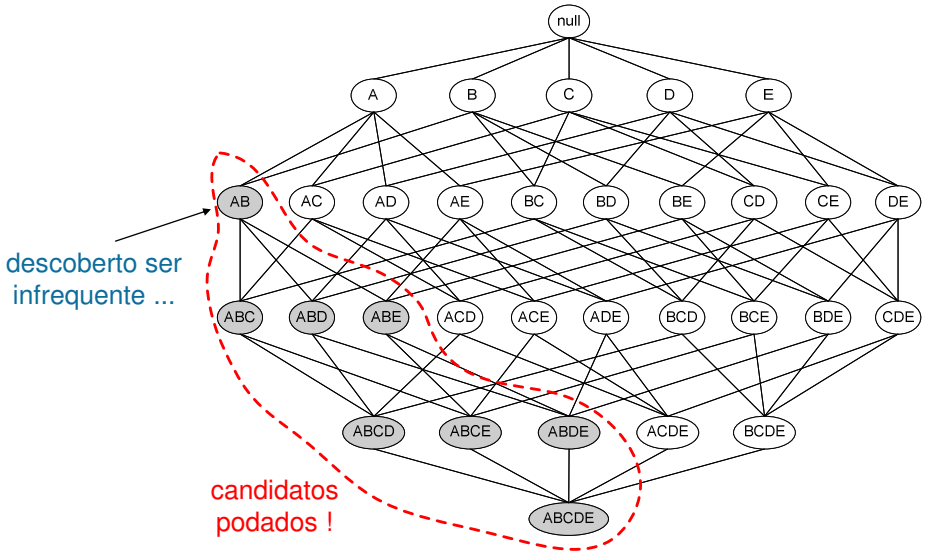
Principal Forma de Reduzir o No. de Candidatos

- Princípio Apriori:
  - Se um conjunto de itens é frequente, então todos os seus subconjuntos também devem ser
- Válido devido à seguinte propriedade do suporte:

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

  - Propriedade anti-monotônica
    - suporte de um conjunto de itens nunca excede os suportes dos seus subconjuntos !

Ilustrando o Princípio



Ilustrando o Princípio

sup_mín = 3

Item	Suporte
Bread	**4**
Coke	**2**
Milk	**4**
Beer	**3**
Diaper	**4**
Eggs	**1**

Conjuntos de Tamanho 1

Conj. de Itens	Suporte
{Bread,Milk}	**3**
{Bread,Beer}	**2**
{Bread,Diaper}	**3**
{Milk,Beer}	**2**
{Milk,Diaper}	**3**
{Beer,Diaper}	**3**

Conjuntos de Tamanho 2

(não é necessário gerar candidatos envolvendo Coke ou Eggs!)

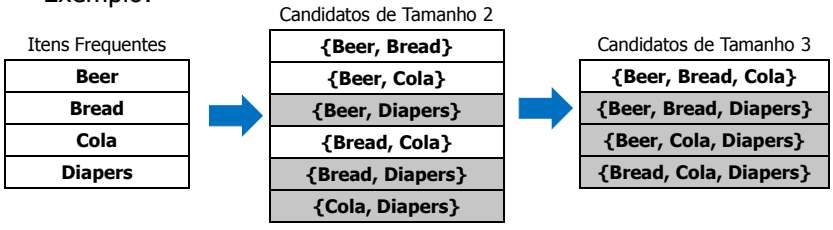
Conj. de Tamanho 3

Geração de Candidatos

- **Método F_{k-1} x F_1 :**
  - Gera conjuntos de itens de tamanho k unindo conjuntos de itens frequentes de tamanho k – 1 e de tamanho 1
    - **Nota:** não é o método mais eficiente de geração de candidatos, mas é simples e basta para as necessidades do nosso curso !
  - Exemplo (k = 3):
    - {Bread, Diapers} ∪ {Milk} → {Bread, Diapers, Milk}
  - Mas como evitar redundâncias...?
    - Ex.: {Milk, Diapers} ∪ {Bread} → {Milk, Diapers, Bread}

Geração de Candidatos

- **Método F_{k-1} x F_1 :**
  - Para evitar redundâncias, basta manter os itens ordenados internamente a cada conjunto e gerar os candidatos de forma organizada, unindo cada conj. frequente de tamanho (k – 1) apenas com os itens freq. de ordem superior na lista ordenada
  - Exemplo:



Geração de Candidatos

sup_mín = 3

Item	Suporte
Bread	4
Coke	2
Milk	4
Beer	3
Diaper	4
Eggs	1

Conjuntos de Tamanho 1

Conj. de Itens	Suporte
{Bread,Milk}	3
{Bread,Beer}	2
{Bread,Diaper}	3
{Milk,Beer}	2
{Milk,Diaper}	3
{Beer,Diaper}	3

Conjuntos de Tamanho 2

Geração de candidatos deve seguir até que, para algum k, não haja conjuntos frequentes de k itens

Conj. de Tamanho k

∅

Poda de Candidatos

- Note que todo o procedimento demanda computar o suporte de cada conjunto candidato gerado...
  - Varrer a BD e tentar casar cada candidato com cada transação...
  - Pode ser bem mais eficiente:
    - mantendo ordenadas as transações e os candidatos
    - utilizando estruturas de dados apropriadas
  - Mesmo assim é computacionalmente caro !
  - Custo computacional pode ser amenizado se for possível eliminar alguns candidatos sem computar diretamente seu suporte

Poda de Candidatos (Exemplo)

- Dados 5 conjuntos freqüentes de 3 itens:

  {A B C}, {A B D}, {A C D}, {A C E}, {B C D}

- Conjunto candidato formado por 4 itens:

  {A B C D}

  → Pode ser freqüente, pois todos os seus subconjuntos de 3 itens o são

- E o que dizer sobre o conjunto {A C D E} ?

  → Como {C D E} não é freqüente, o conjunto {A C D E} também não é !

  → Pode ser descartado

Rotina Básica

- Algoritmo:
  - Seja k=1
  - Gere conjuntos de itens frequentes de tamanho 1
  - Repita até que não haja mais conjuntos frequentes
    - **Gere** conjuntos candidatos de tamanho (k+1)
    - **Pode** os candidatos que possuem subconjuntos de tamanho k que não são frequentes
    - Conte o suporte dos candidatos remanescentes varrendo a base de dados e **elimine** os candidatos infrequentes , ou seja, aqueles com contagem menor que sup_mín

Exercício:

- Gere os conjuntos de itens frequentes com sup_mín = 5 na BD abaixo:

(Witten and Frank, 2005)

Nota: cada par atributo = valor é um item !

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Voltando ao Exemplo Anterior (Freitas & Lavington)...

R	Leite	Café	Cerveja	Pão	Manteiga	Arroz	Feijão
1	N	S	N	S	S	N	N
2	S	N	S	S	S	N	N
3	N	S	N	S	S	N	N
4	S	S	N	S	S	N	N
5	N	N	S	N	N	N	N
6	N	N	N	N	S	N	N
7	N	N	N	S	N	N	N
8	N	N	N	N	N	N	S
9	N	N	N	N	N	S	S
10	N	N	N	N	N	S	N

- Passo 1 – Suporte p/ Conjuntos com 1 Item:
  - Arroz: 2; Café: 3; Cerveja: 2; Feijão: 2; Leite: 2; Manteiga: 5; Pão: 5.
  - Considerando sup_mín = 3:
  - **Café**, **Manteiga** e **Pão** seriam os itens freqüentes!
- Passo 2 – Suporte p/ Conjuntos com 2 Itens:

  ⇒Procurar considerando somente os itens freqüentes

  ⇒ Café, Manteiga, Pão

- Passo 2 ...

{Café, Manteiga}: Suporte = 3;

{Café, Pão}: Suporte = 3;

{Manteiga, Pão}: Suporte = 4;

Conjuntos de itens freqüentes para sup_mín = 3:

{Café, Manteiga}, {Café, Pão}, {Manteiga, Pão}

- Passo 3 – Suporte p/ Conjuntos com 3 Itens:
  - A partir dos conjuntos anteriores obtém-se:
    - {Café, Manteiga, Pão}: Suporte = 3;
  - Nota:
    - Antes de calcular o suporte deste conjunto, ele foi submetido (e sobreviveu) ao procedimento de poda que verificou que todos os seus subconjuntos de tamanho 2 são freqüentes:
      - {Manteiga, Pão}, {Café, Pão}, {Café, Manteiga}

R	Leite	Café	Cerveja	Pão	Manteiga	Arroz	Feijão
1	N	S	N	S	S	N	N
2	S	N	S	S	S	N	N
3	N	S	N	S	S	N	N
4	S	S	N	S	S	N	N
5	N	N	S	N	N	N	N
6	N	N	N	N	S	N	N
7	N	N	N	S	N	N	N
8	N	N	N	N	N	N	S
9	N	N	N	N	N	S	S
10	N	N	N	N	N	S	N

(Freitas & Lavington)

- Em Resumo:
  - Sup({Café}) = 3 e Sup({Manteiga}) = Sup({Pão}) = 5
  - Sup({Café, Manteiga}) = 3
  - Sup({Café, Pão}) = 3
  - Sup({Manteiga, Pão}) = 4
  - Sup({Café, Manteiga, Pão}) = 3

Gerando Regras de Associação:

- Dados os conjuntos de itens freqüentes:
  - Para cada conjunto I:
    - Para cada subconjunto J de I:
      - Determinar todas as regras de associação da forma:

        $(I - J) \Rightarrow J$

      - Eliminar aquelas com medida de **confiança** menor que o limiar mínimo pré-estabelecido (conf_mín)

Voltando ao Exemplo Anterior (Freitas & Lavington)...

Sup({Café) = 3; Sup({Manteiga}) = Sup({Pão}) = 5

Sup({Café, Manteiga}) = 3; Sup({Café, Pão}) = 3; Sup({Manteiga, Pão}) = 4

Sup({Café, Manteiga, Pão}) = 3

Calcula-se, então, a **Confiança** das regras candidatas:

a) {Café, Manteiga} :
   - Se "café" então "manteiga" – conf.=1,0
   - Se "manteiga" então "café" – conf.=0,6

b) {Café, Pão} :
   - Se "café" então "pão" – conf.=1,0
   - Se "pão" então "café" – conf.=0,6

c) {Manteiga, Pão} :
   - Se "manteiga" então "pão" – conf.=0,8
   - Se "pão" então "manteiga" – conf.=0,8

d) {Café, Manteiga, Pão} :
   - Se "café, pão" então "manteiga" – conf.=1,0
   - Se "café, manteiga" então "pão" – conf.=1,0
   - Se "manteiga, pão" então "café" – conf.=0,75
   - Se "café" então "pão,manteiga" – conf.=1,0
   - e assim por diante, escolhendo-se depois as regras que respeitam conf_mín.

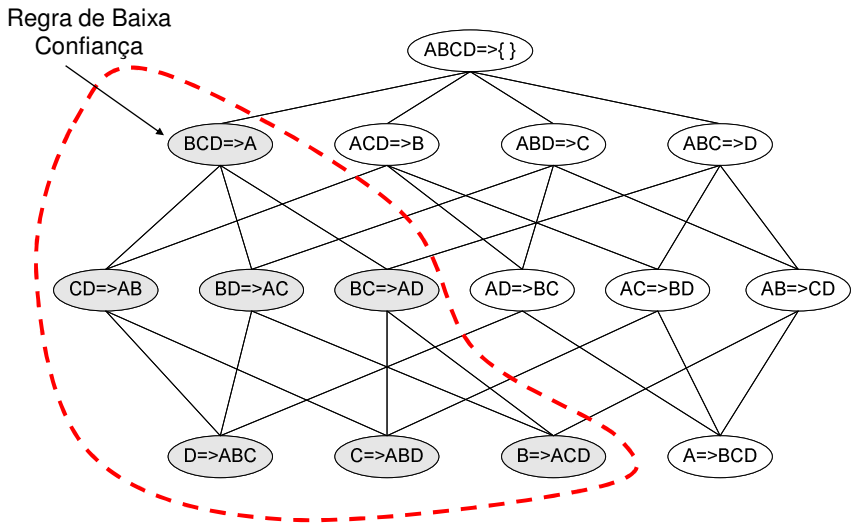
Exercícios

- Complete o exemplo anterior selecionando todas as regras de associação que se pode extrair da BD que tenham conf_mín = 0.8
- Explique porque não é preciso calcular o suporte das regras candidatas (apenas a confiança) para saber que essas regras necessariamente possuem suporte maior ou igual ao mínimo (0,3 neste exemplo)

Nota Sobre a Geração de Regras

- A geração do conjunto de regras também pode se beneficiar do princípio apriori
- É simples observar que o princípio implica a seguinte propriedade da confiança:
  - Confiança de regra de um dado conj. de itens não cresce se passamos itens da esquerda para a direita da regra
  - Por exemplo, dado um conj. de itens {A B C D}:
    - conf(ABC $\rightarrow$ D) $\geq$ c(AB $\rightarrow$ CD) $\geq$ c(A $\rightarrow$ BCD)

Geração de Regras no Algoritmo Apriori



"Filtrando" Regras de Associação:

- Problema: grandes BDs (e.g. supermercados) podem produzir um número elevado de regras de associação, mesmo com valores razoáveis para suporte e confiança...
  - Esse problema é ainda mais crítico em BDs com distribuições de suporte desbalanceadas...
- Possíveis soluções:
  - Pré-processar a base e/ou **filtrar regras**...
- Medidas de interesse (objetivas e subjetivas) para filtragem:
  - Tópico fundamental em análise de associação, mas está além do escopo deste curso...

"Filtrando" Regras de Associação:

- Medidas de Interesse Subjetivas:
  - Geralmente são dependentes do problema
  - Tipicamente são especificadas por um especialista de domínio e utilizadas para filtrar regras que não satisfazem as especificações
    - podem ser formalizadas em termos lógicos ou matemáticos e inseridas no processo automático de filtragem; ou
    - podem ser aplicadas de forma iterativa pelo próprio especialista em um ambiente iterativo amigável (visual data mining)
- Medidas de Interesse Objetivas:
  - Alternativas a Suporte - Confiança

Exercícios Adicionais

- Tome alguns conjuntos de itens de diferentes tamanhos que você gerou no exercício envolvendo a base de (Witten & Frank, 2005), gere regras de associação a partir desses conjuntos e calcule a confiança de cada uma dessas regras.

Exercícios Adicionais

- A propriedade que a confiança de uma regra de um dado conj. de itens não cresce se passamos itens da esquerda para a direita da regra é denominada anti-monotônica. Com base na definição da medida de confiança, explique porque essa propriedade é válida.

Referências

- P.-N. Tan, Steinbach, M., and Kumar, V., *Introduction to Data Mining*, Addison-Wesley, 2006
- I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd Edition, Morgan Kaufmann, 2005