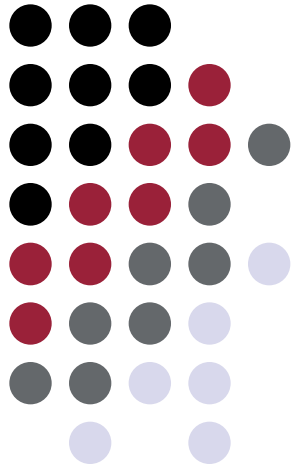
Mineração de Dados: Associação

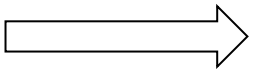
CEA462 – Sistemas de Apoio à Decisão

Prof. MSc. George H. G. Fonseca
Universidade Federal de Ouro Preto



Introdução

- Associação é a tarefa de descobrir relacionamentos interessantes entre registros em grandes bases de dados
  - Transações de cestas de compras
  - Identificar relações entre convicções e o voto de eleitores



Definições

- Um algoritmo de associação procura por regras de associação
  - Que atendam a um **suporte** e **confiança** mínimos preestabelecidos
  - Uma regra de associação é expressa na seguinte forma
    - $X \longrightarrow Y$
  - Seja $\sigma(X)$ a quantia de vezes que um conjunto X de itens aparece,
    - Suporte $s(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{N}$
    - Confiança $c(X \longrightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$

Definições

- Exemplo

TID	Itens
1	{Pão, Leite}
2	{Pão, Fraldas, Cerveja, Ovos}
3	{Leite, Fraldas, Cerveja, Cola}
4	{Pão, Leite, Fraldas, Cerveja}
5	{Pão, Leite, Fraldas, Cola}

- Regra: {Leite, Fraldas} ⟶ {Cerveja} (X ⟶ Y)
- s = $\frac{2}{5}$ “Número de vezes que a regra aparece sobre o número total de registros”
- c = $\frac{2}{3}$ “Número de vezes que a implicação (Y) aparece sobre o número de vezes que o precedente aparece (X)”

Definições

- Associação é um processo de duas etapas
  - Gerar o conjunto de itemsets frequentes
    - e.g. que atendem ao suporte mínimo
  - Gerar as regras apartir dos itemsets frequentes
    - e.g. que atendem à confiança mínima

Geração do Conjunto de Itens Frequentes por Força Bruta

- Listar todos os conjuntos de itens possíveis e calcular seu suporte

TID	Itens
1	{Pão, Leite}
2	{Pão, Fraldas, Cerveja, Ovos}
3	{Leite, Fraldas, Cerveja, Cola}
4	{Pão, Leite, Fraldas, Cerveja}
5	{Pão, Leite, Fraldas, Cola}

Candidatos
{Pão}
{Leite}
{Fraldas}
{Cerveja}
{Ovos}
{Cola}
{Pão, Leite}
{Pão, Fraldas}
{Pão, Cerveja}
...

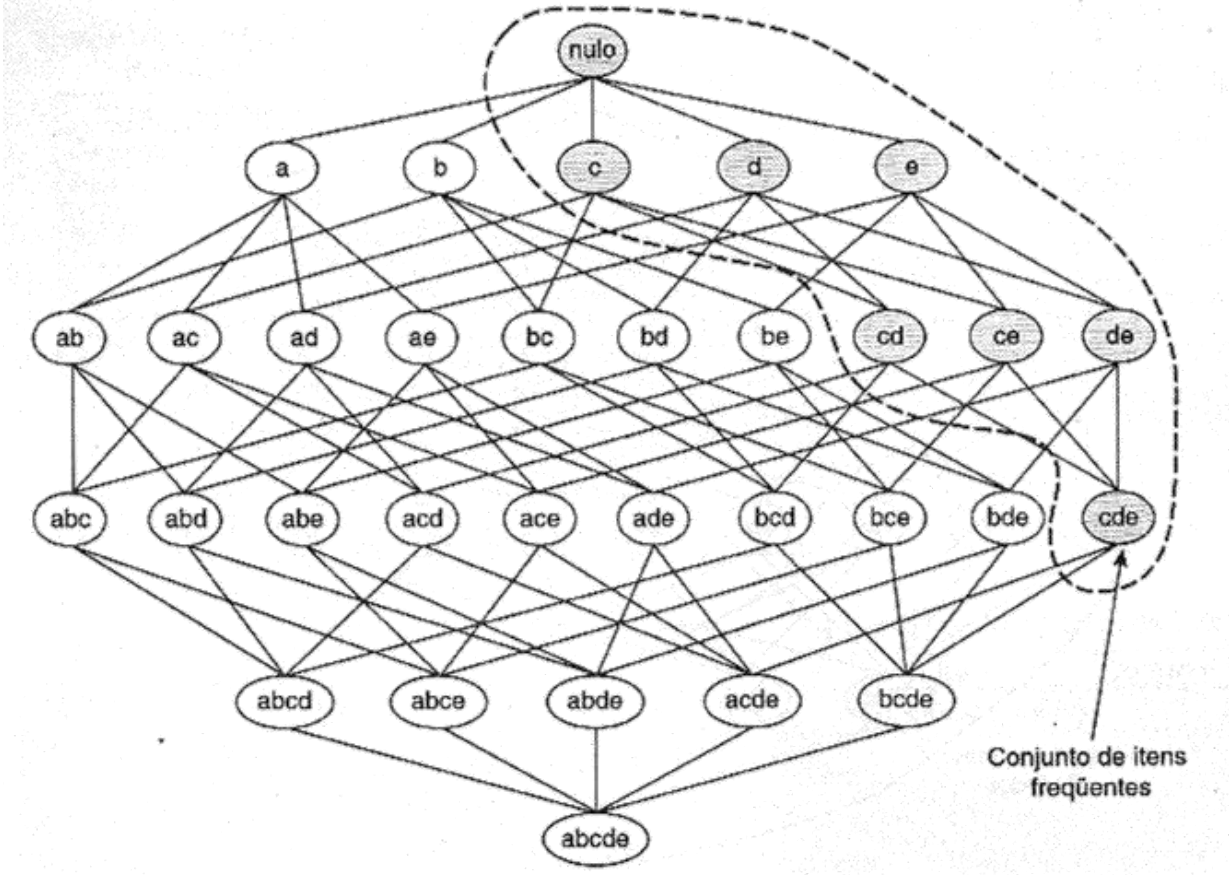
Geração do Conjunto de Itens Frequentes por Força Bruta

- Extremamente ineficiente!
  - O(N × M × w)
  - N = número de transações
  - M = número de conjuntos de itens candidatos (2^k - 1, k = itens)
  - w = Tamanho da maior transação

  - Nosso pequeno exemplo levaria a (5 * 63 * 4) = 1260 comparações

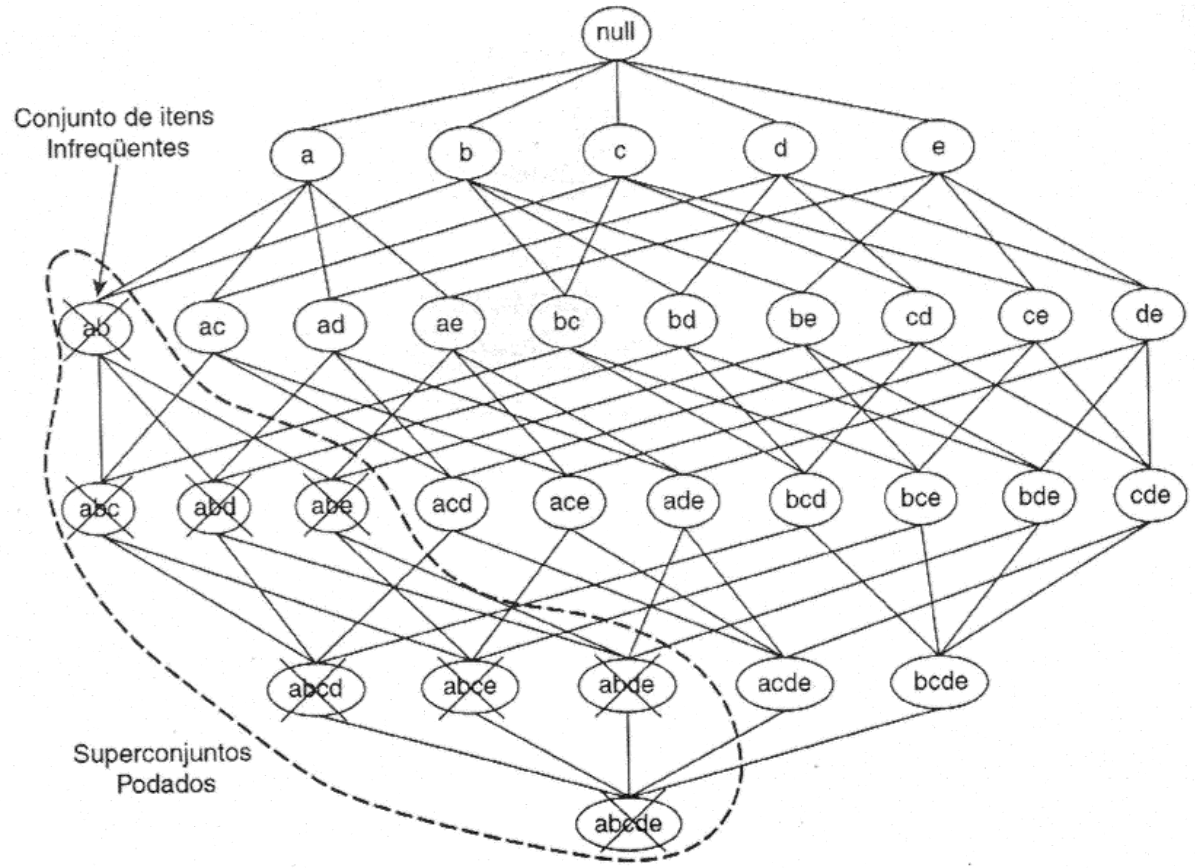
Princípio Apriori

- Se um conjunto de itens é frequente, então todos os seus subconjuntos também o são.



Propriedade Monotônica

- Se um conjunto de itens é infrequente, então todos os seus superconjuntos também o são.



Algoritmo Apriori

- Inicialmente cada item é considerado um conjunto candidato de 1 item
  - Os que não atendem o suporte mínimo são descartados
- Posteriormente, procura-se conjuntos candidatos de 2 itens usando apenas os conjuntos de 1 item
- O processo é repetido até não restem mais conjuntos de candidatos a serem testados

Algoritmo Apriori

- Algoritmo

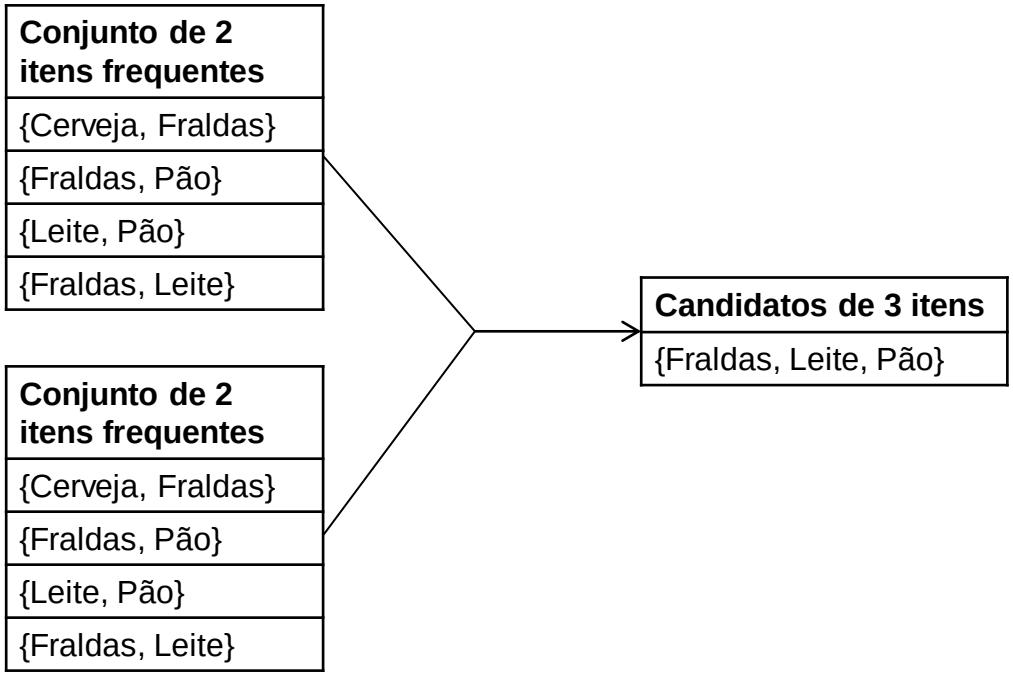
apriori(minSup, conj. de transações T, conj. de itens I)

1. k = 1;
2. $F = F_k = \{i \mid i \in I \wedge \sigma(i) \geq minSup \times |T|\}$; — Gera o conjunto de 1-itemsets frequentes
3. **faça**
4. ++k; — Gera o conjunto itens candidatos (a ser explicado)
5. $C_k = genCandidatos(F_{k-1})$;
6. **para cada** $t \in T$
7. $C_t = subset(C_k, t)$; — Separa os candidatos que estão contidos na transação t
8. para cada candidato $c \in C$
9. $++\sigma(c)$; — Incrementa seu valor de suporte
10. **fim-para**
11. **fim-para**
12. $F_k = \{c \mid c \in C_k \wedge \sigma(c) \geq minSup \times |T|\}$; — Adiciona os candidatos que atendem ao suporte mínimo
13. $F = F \cup F_k$;
14. **até que** $F_k = \emptyset$;
15. retorne F;

fimAssociador

Algoritmo Apriori

- A função genCandidatos(F_{k-1})
  - Funde pares de conjuntos frequentes de tamanho k-1, gerando conjuntos frequentes de tamanho k
  - Gerará um novo candidato se
    - $a_i = b_i$ para i = 1, ..., k-2 e $a_i \neq b_i$ para i = k-1



Algoritmo Apriori

- Exemplo
  - minsup = 0.5

TID	Itens
1	{Pão, Leite}
2	{Pão, Fraldas, Cerveja, Ovos}
3	{Leite, Fraldas, Cerveja, Cola}
4	{Pão, Leite, Fraldas, Cerveja}
5	{Pão, Leite, Fraldas, Cola}

- Obs.: Itens em azul compõem F (conj. de itemsets frequentes)

apriori(minSup, conj. de transações T, conj. de itens I)
1. **k = 1;**
2. **F = F_k = {i | i ∈ I ∧ σ(i) ≥ minSup × |T|};**
3. **faça**
4. ++k;
5. C_k = genCandidatos(F_{k-1});
6. **para cada** t ∈ T
7. C_t = subset(C_k, t);
8. para cada candidato c ∈ C
9. ++σ(c);
10. **fim-para**
11. **fim-para**
12. F_k = {c | c ∈ C_k ∧ σ(c) ≥ minSup × |T|};
13. F = F ∪ F_k;
14. **até que** F_k = ∅;
15. retorne F;

fimAssociador

Conjunto de 1 itens frequentes
{Cerveja} σ = 3
{Fraldas} σ = 4
{Leite} σ = 4
{Pão} σ = 4

Algoritmo Apriori

- Exemplo
  - minsup = 0.5

apriori(minSup, conj. de transações T, conj. de itens I)
1. k = 1;
2. F = F_k = {i | i ∈ I ∧ σ(i) ≥ minSup × |T|};
3. **faça**
4. **++k;**
5. **C_k = genCandidatos(F_{k-1});**
6. **para cada** t ∈ T
7. C_t = subset(C_k, t);
8. para cada candidato c ∈ C
9. ++σ(c);
10. **fim-para**
11. **fim-para**
12. F_k = {c | c ∈ C_k ∧ σ(c) ≥ minSup × |T|};
13. F = F ∪ $F_{k;}$
14. **até que** F_k = ∅;
15. retorne F;

fimAssociador

Conjunto de 1 itens frequentes
{Cerveja}
{Fraldas}
{Leite}
{Pão}

Conjunto de 1 itens frequentes
{Cerveja}
{Fraldas}
{Leite}
{Pão}

Candidatos de 2 itens
{Cerveja, Fraldas}
{Cerveja, Leite}
{Cerveja, Pão}
{Fraldas, Leite}
{Fraldas, Pão}
{Leite, Pão}

Algoritmo Apriori

- Exemplo

TID	Itens
1	{Pão, Leite}
2	{Pão, Fraldas, Cerveja, Ovos}
3	{Leite, Fraldas, Cerveja, Cola}
4	{Pão, Leite, Fraldas, Cerveja}
5	{Pão, Leite, Fraldas, Cola}

apriori(minSup, conj. de transações T, conj. de itens I)

1. k = 1;
2. F = F_k = {i | i ∈ I ∧ σ(i) ≥ minSup × |T|};
3. **faça**
4. ++k;
5. C_k = genCandidatos(F_{k-1});
6. **para cada t ∈ T**
7. **C_t = subset(C_k, t);**
8. **para cada candidato c ∈ C**
9. **++σ(c);**
10. **fim-para**
11. **fim-para**
12. F_k = {c | c ∈ C_k ∧ σ(c) ≥ minSup × |T|};
13. F = F ∪ F_k;
14. **até que** F_k = ∅;
15. retorne F;

fimAssociador

Candidatos de 2 itens (C_k)
{Cerveja, Fraldas} σ = 3
{Cerveja, Leite} σ = 2
{Cerveja, Pão} σ = 2
{Fraldas, Leite} σ = 3
{Fraldas, Pão} σ = 3
{Leite, Pão} σ = 3

TID	C_t
1	{{Leite, Pão}}
2	{{Cerveja, Fraldas}, {Cerveja, Pão}, {Fraldas, Pão}}
3	{{Cerveja, Fraldas}, {Cerveja, Leite}, {Fraldas, Leite}}
4	{{Cerveja, Fraldas}, {Cerveja, Leite}, {Cerveja, Pão}, {Fraldas, Leite}, {Fraldas, Pão}, {Leite, Pão}}
5	{{Fraldas, Leite}, {Fraldas, Pão}, {Leite, Pão}}

Algoritmo Apriori

- Exemplo
  - minsup = 0.5

apriori(minSup, conj. de transações T, conj. de itens I)
1. k = 1;
2. F = F_k = {i | i ∈ I ∧ σ(i) ≥ minSup × |T|};
3. **faça**
4. ++k;
5. C_k = genCandidatos(F_{k-1});
6. **para cada** t ∈ T
7. C_t = subset(C_k, t);
8. para cada candidato c ∈ C
9. ++σ(c);
10. **fim-para**
11. **fim-para**
12. **F_k = {c | c ∈ C_k ∧ σ(c) ≥ minSup × |T|};**
13. **F = F ∪ F_k;**
14. **até que** F_k = ∅;
15. retorne F;

fimAssociador

Candidatos de 2 itens (C_k)
{Cerveja, Fraldas} σ = 3
{Cerveja, Leite} σ = 2
{Cerveja, Pão} σ = 2
{Fraldas, Leite} σ = 3
{Fraldas, Pão} σ = 3
{Leite, Pão} σ = 3

→

Conjunto de 2 itens frequentes (F_k)
{Cerveja, Fraldas} σ = 3
{Fraldas, Leite} σ = 3
{Fraldas, Pão} σ = 3
{Leite, Pão} σ = 3

Algoritmo Apriori

```
apriori(minSup, conj. de transações T, conj. de itens I)
        1.k = 1;
        2.F = F_k = {i | i ∈ I ∧ σ(i) ≥ minSup × |T|};
        3.faça
        4.   ++k;
        5.   C_k = genCandidatos(F_k-1);
        6.   para cada t ∈ T
        7.      C_t = subset(C_k, t);
        8.      para cada candidato c ∈ C
        9.         ++σ(c);
        10.     fim-para
        11.  fim-para
        12.  F_k = {c | c ∈ C_k ∧ σ(c) ≥ minSup × |T|};
        13.  F = F ∪ F_k;
        14.até que F_k = ∅;
        15.retorne F;
fimAssociador
```

- Exemplo
  - minsup = 0.5

Conjunto de 2 itens frequentes (F_k)
{Cerveja, Fraldas}
{Fraldas, Leite}
{Fraldas, Pão}
{Leite, Pão}

Conjunto de 2 itens frequentes (F_k)
{Cerveja, Fraldas}
{Fraldas, Leite}
{Fraldas, Pão}
{Leite, Pão}

Candidatos de 3 itens
{Fraldas, Leite, Pão}

Algoritmo Apriori

```
apriori(minSup, conj. de transações T, conj. de itens I)
        1.k = 1;
        2.F = F_k = {i | i ∈ I ∧ σ(i) ≥ minSup × |T|};
        3.faça
        4.   ++k;
        5.   C_k = genCandidatos(F_{k-1});
        6.   para cada t ∈ T
        7.      C_t = subset(C_k, t);
        8.      para cada candidato c ∈ C
        9.         ++σ(c);
        10.     fim-para
        11.  fim-para
        12.  F_k = {c | c ∈ C_k ∧ σ(c) ≥ minSup × |T|};
        13.  F = F ∪ F_k;
        14.até que F_k = ∅;
        15.retorne F;
fimAssociador
```

- Exemplo

TID	Itens
1	{Pão, Leite}
2	{Pão, Fraldas, Cerveja, Ovos}
3	{Leite, Fraldas, Cerveja, Cola}
4	{Pão, Leite, Fraldas, Cerveja}
5	{Pão, Leite, Fraldas, Cola}

TID	C_t
1	
2	
3	
4	{Fraldas, Leite, Pão}
5	{Fraldas, Leite, Pão}

Candidatos de 3 itens
{Fraldas, Leite, Pão} σ = 2

Algoritmo Apriori

```
apriori(minSup, conj. de transações T, conj. de itens I)
        1.k = 1;
        2.F = F_k = {i | i ∈ I ∧ σ(i) ≥ minSup × |T|};
        3.faça
        4.   ++k;
        5.   C_k = genCandidatos(F_k-1);
        6.   para cada t ∈ T
        7.      C_t = subset(C_k, t);
        8.      para cada candidato c ∈ C
        9.         ++σ(c);
        10.     fim-para
        11.  fim-para
        12.  F_k = {c | c ∈ C_k ∧ σ(c) ≥ minSup × |T|};
        13.  F = F ∪ F_k;
        14.até que F_k = ∅;
        15.retorne F;
fimAssociador
```

- Exemplo
  - minsup = 0.5

Candidatos de 3 itens
{Fraldas, Leite, Pão} σ = 2

→

Conjunto de 3 itens frequentes (F_k)
∅

- Fim do algoritmo ($F_k = \emptyset$)
- Retorne F (conjunto de itemsets em azul)

Algoritmo Apriori

- E para gerar as regras!?
  - Para cada conjunto frequente f ∈ F, gere todas as possibilidades de regras e avalie sua confiança
    - Valores de suporte, previamente armazenados são utilizados para tal, assim não é necessário ler a base de dados novamente!
    - Conjuntos frequentes de tamanho 1 são descartados

Algoritmo Apriori

- E para gerar as regras!?
  - Seguindo o exemplo, suponha minConf = 0.9 (α)

Conjunto de itens frequentes (F)
~~{Cerveja} σ = 3~~
~~{Fraldas} σ = 4~~
~~{Leite} σ = 4~~
~~{Pão} σ = 4~~
{Cerveja, Fraldas} σ = 3
{Fraldas, Leite} σ = 3
{Fraldas, Pão} σ = 3
{Leite, Pão} σ = 3

Regras
{Cerveja} ⟶ {Fraldas} α = 1.00
~~{Fraldas} ⟶ {Cerveja} α = 0.75~~
~~{Fraldas} ⟶ {Leite} α = 0.75~~
~~{Leite} ⟶ {Fraldas} α = 0.75~~
~~{Fraldas} ⟶ {Pão} α = 0.75~~
{Pão} ⟶ {Fraldas} α = 1.00
~~{Leite} ⟶ {Pão} α = 0.75~~
~~{Pão} ⟶ {Leite} α = 0.75~~

Exercício

- Execute o manualmente algoritmo Apriori para a seguinte base de dados considerando
  - minSup = 0.5 e minConf = 0.6 (α)

Registros
{Imigracao_S, AjudaELSlavador_S, ReligiaoEscolas_S, Democrata}
{Imigracao_S, AjudaELSlavador_S, ReligiaoEscolas_N, Democrata}
{Imigracao_N, AjudaELSlavador_N, ReligiaoEscolas_S, Republicano}
{Imigracao_N, AjudaELSlavador_N, ReligiaoEscolas_N, Republicano}
{Imigracao_S, AjudaELSlavador_N, ReligiaoEscolas_N, Democrata}

Bibliografia

- *Introdução ao Data Mining*. Steinbach, Michael; Kumar, Vipin; Tan, Pang-ning, Rio de Janeiro: Ed. Ciência Moderna, 2009. Capítulo 6.