Mineração de Dados

Regras de Associação

Sumário

Introdução

Mineração de Itens Frequentes

- Em muitas situações, é importante conhecer as relações entre dois ou mais objetos da base de dados
  - Esses conjuntos de itens são chamados de *itemsets*
- Análises de cestas de compras é uma aplicação tradicional
  - Itens que são comprados juntos em mercados
  - Um caso popular é a compra de fraldas e cervejas
- A mineração de *itemsets* frequentes é uma tarefa exploratória básica
  - Busca por co-ocorrências
  - Fornece uma estimativa da probabilidade conjunta
- Uma vez determinado os *itemsets* frequentes, pode-se extrair regras de associação desses conjuntos
  - Fornece informação da ocorrência condicional dos itens
- Este conteúdo é baseado no material de Zaki & Meira Jr., *Data Mining and Analysis*

Mineração de Itens Frequentes: Terminologia

- *Itemsets*
  - Seja $I = \{x_1, x_2, \ldots, x_m\}$ um conjunto de elementos chamados itens
  - Um conjunto $X \subseteq I$ é chamado de *itemset* e um *itemset* de cardinalidade k é chamado de k-*itemset*
  - $I^{(k)}$ é o conjunto de todos os k-*itemsets*
- *Tidsets*
  - Seja $T = \{t_1, t_2, \ldots, t_n\}$ um conjunto de identificadores de transações ou *tids*
  - Um conjunto $\mathbf{T} \subseteq T$ é chamado de *tidset*
- Na prática, *itemsets* e *tidsets* são mantidos ordenados
- Transações
  - Uma transação é uma tupla $< t, X >$ onde $t \in T$ e X é um *itemset*
- Uma base de dados binária D é um conjunto que relaciona *tids* e itens, ou seja, $D \subseteq T \times I$

Representação de Bases de Dados

- Bases de dados binárias podem ser representadas por uma base transacional/horizontal ou vertical
- $\mathbf{i}(t)$ é o conjunto de itens do *tid* $t \in T$
- $\mathbf{t}(x)$ é o conjunto de *tids* que contem o item x

Bases Transacionais e Verticais

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

(a) Binary database

t	$\mathbf{i}(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

(b) Transaction database

x	A	B	C	D	E
$\mathbf{t}(x)$	1	1	2	1	1
	3	2	4	3	2
	4	3	5	5	3
	5	4	6	6	4
		5			5
		6			

(c) Vertical database

- A base de dados D tem 5 itens ($I = \{A, B, C, D, E\}$) e 6 *tids* ($T = \{1, 2, 3, 4, 5, 6\}$)
- A primeira transação é $< 1, \{A, B, D, E\} >$ ou $< 1, ABDE >$

Mineração de *Itemsets* Frequentes

Suporte e *Itemsets* Frequentes

- Suporte de um *itemset* X é a quantidade de transações em D que contem X
  $sup(X) = |\{t| < t, i(t) > \in D \text{ e } X \subseteq i(t)\}| = |t(X)|$
- O suporte relativo pode ser definido como
  $$rsup(X) = \frac{sup(X)}{|D|}$$
- O suporte relativo é uma estimativa da probabilidade conjunta dos itens em X
- X é chamado de *itemset* frequente quando $sup(X) \geq minsup$, one $minsup$ é um limiar de suporte mínimo (definido pelo usuário)
- O conjunto F denota o conjunto de todos os *itemsets* frequentes e $F^{(k)}$ é o conjunto de k-*itemsets* frequentes

Itemsets Frequentes

- $minsup = 3$

t	$\mathbf{i}(t)$
1	*ABDE*
2	*BCE*
3	*ABDE*
4	*ABCE*
5	*ABCDE*
6	*BCD*

Transaction Database

sup	itemsets
6	*B*
5	*E*, *BE*
4	*A*, *C*, *D*, *AB*, *AE*, *BC*, *BD*, *ABE*
3	*AD*, *CE*, *DE*, *ABD*, *ADE*, *BCE*, *BDE*, *ABDE*

Frequent Itemsets

The 19 frequent itemsets shown in the table comprise the set $\mathcal{F}$. The sets of all frequent k-itemsets are

$$\mathcal{F}^{(1)} = \{A, B, C, D, E\}$$
$$\mathcal{F}^{(2)} = \{AB, AD, AE, BC, BD, BE, CE, DE\}$$
$$\mathcal{F}^{(3)} = \{ABD, ABE, ADE, BCE, BDE\}$$
$$\mathcal{F}^{(4)} = \{ABDE\}$$

Regras de Associação

Regras de Associação

- Uma regra de associação é definida como $X \rightarrow Y$, onde X e Y são *itemsets* disjuntos
- Seja o suporte da regra o número de transações em que X e Y ocorrem conjuntamente ($X \cup Y$ ou XY)
  $sup(X \rightarrow Y) = sup(XY) = |t(XY)|$
- A fração em que XY ocorre é o suporte relativo da regra
  $$rsup(X \rightarrow Y) = \frac{sup(XY)}{|D|} = P(X \wedge Y)$$
- A confiança de uma regra é a probabilidade condicional das transações conterem Y tal que contém X
  $$conf(X \rightarrow Y) = P(Y|X) = \frac{P(X \wedge Y)}{P(X)} = \frac{sup(XY)}{sup(X)}$$

Algoritmo Força Bruta pra Minerar *Itemsets*

Algoritmo Força Bruta

- Determina todos os *itemsets*, computa seus valores de suporte e guarda os *itemsets* frequentes
- Complexidade computacional $O(|I|\ |D|\ 2^{|I|})$

Algoritmo Força Bruta

BRUTEFORCE (D, $\mathcal{I}$, *minsup*):

```
1 F ← ∅ // set of frequent itemsets
2 foreach X ⊆ I do
3     sup(X) ← COMPUTESUPPORT (X, D)
4     if sup(X) ≥ minsup then
5         F ← F ∪ {(X, sup(X))}
6 return F
```

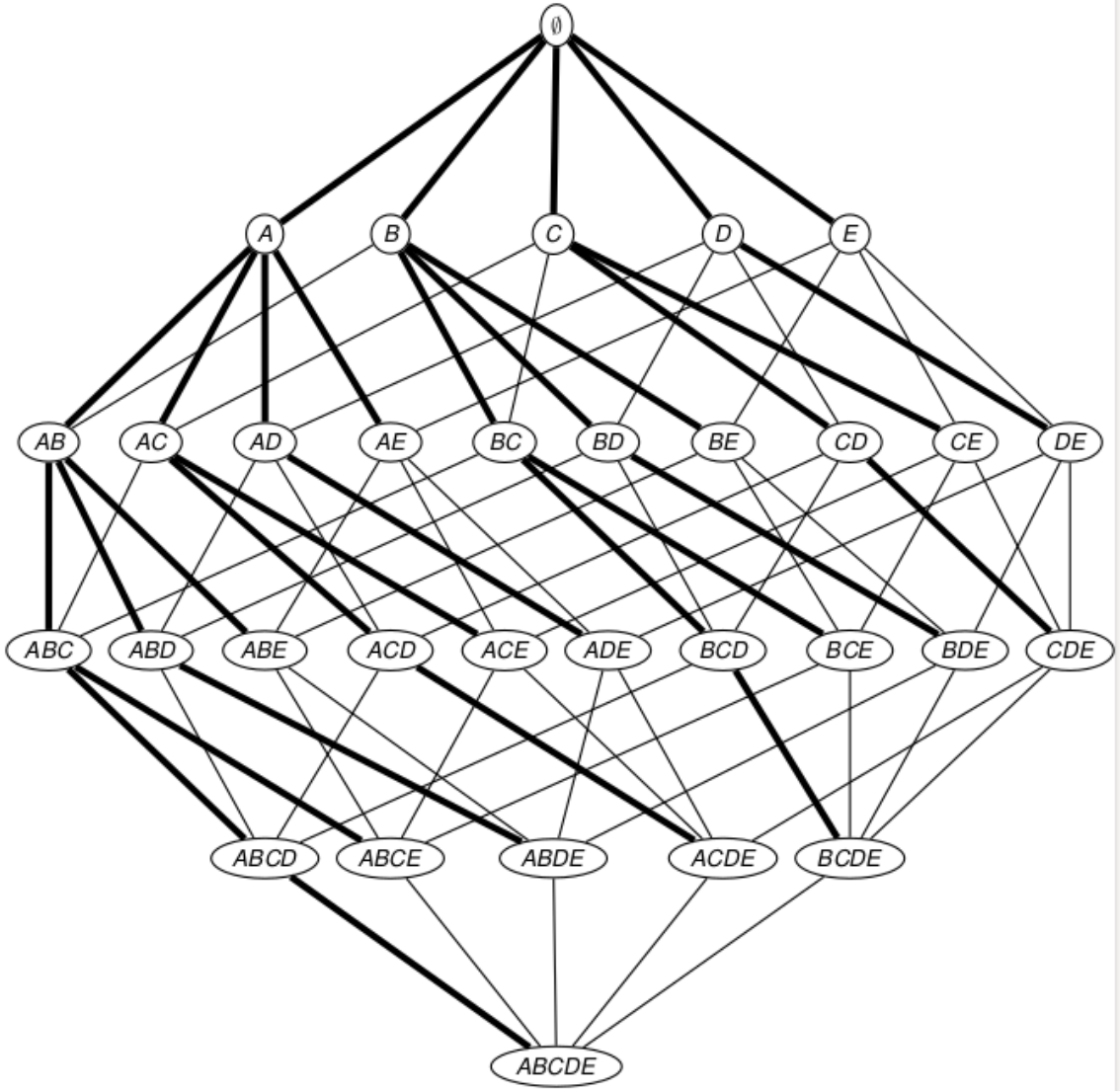
COMPUTESUPPORT (X, D):

```
1 sup(X) ← 0
2 foreach ⟨t, i(t)⟩ ∈ D do
3     if X ⊆ i(t) then
4         sup(X) ← sup(X) + 1
5 return sup(X)
```

Algoritmo Força Bruta

Itemset search space is a lattice where any two itemsets X and Y are connected by a link iff X is an *immediate subset* of Y, that is, $X \subseteq Y$ and $|X| = |Y| - 1$.

Frequent itemsets can enumerated using either a BFS or DFS search on the *prefix tree*, where two itemsets X, Y are connected by a link iff X is an immediate subset and prefix of Y. This allows one to enumerate itemsets starting with an empty set, and adding one more item at a time.



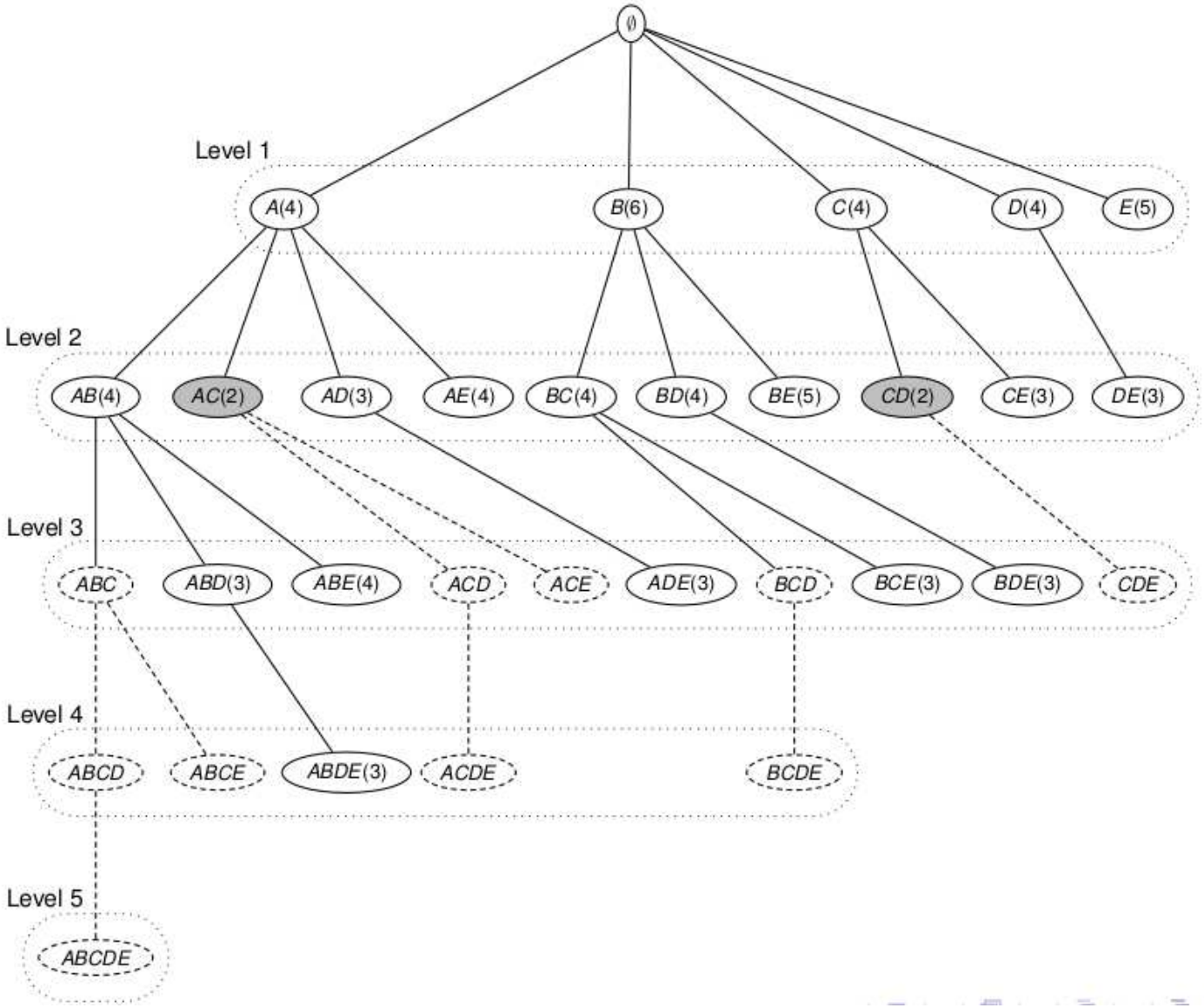
Apriori

Apriori

- Se $X \subseteq Y$ então $sup(X) \geq sup(Y)$ e, assim
  - se X é um *itemset* frequente então qualquer subconjunto $Z \subseteq X$ também é frequente
  - se X não é frequente então qualquer superconjunto $Z \supseteq X$ não é frequente

- O algoritmo *Apriori* explora essas propriedades
  - Reduz as buscas por evitar os candidatos a *itemsets* infrequentes
  - Subconjuntos infrequentes não compõem *itemsets* frequentes

Apriori


∅
Level 1
A(4)
B(6)
C(4)
D(4)
E(5)
Level 2
AB(4)
AC(2)
AD(3)
AE(4)
BC(4)
BD(4)
BE(5)
CD(2)
CE(3)
DE(3)
Level 3
ABC
ABD(3)
ABE(4)
ACD
ACE
ADE(3)
BCD
BCE(3)
BDE(3)
CDE
Level 4
ABCD
ABCE
ABDE(3)
ACDE
BCDE
Level 5
ABCDE

Apriori

APRIORI (D, $\mathcal{I}$, *minsup*):

1 $\mathcal{F} \leftarrow \emptyset$
2 $\mathcal{C}^{(1)} \leftarrow \{\emptyset\}$ // Initial prefix tree with single items
3 **foreach** $i \in \mathcal{I}$ **do** Add i as child of $\emptyset$ in $\mathcal{C}^{(1)}$ with $sup(i) \leftarrow 0$
4 $k \leftarrow 1$ // k denotes the level
5 **while** $\mathcal{C}^{(k)} \neq \emptyset$ **do**
6 COMPUTESUPPORT ($\mathcal{C}^{(k)}$, **D**)
7 **foreach** *leaf* $X \in \mathcal{C}^{(k)}$ **do**
8 **if** $sup(X) \geq minsup$ **then** $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X, sup(X))\}$
9 **else** remove X from $\mathcal{C}^{(k)}$
10 $\mathcal{C}^{(k+1)} \leftarrow$ EXTENDPREFIXTREE ($\mathcal{C}^{(k)}$)
11 $k \leftarrow k + 1$
12 **return** $\mathcal{F}^{(k)}$

Apriori

COMPUTESUPPORT ($\mathcal{C}^{(k)}$, **D**):

foreach $\langle t, \mathbf{i}(t) \rangle \in \mathbf{D}$ **do**
  foreach k-*subset* $X \subseteq \mathbf{i}(t)$ **do**
    if $X \in \mathcal{C}^{(k)}$ **then** $sup(X) \leftarrow sup(X) + 1$

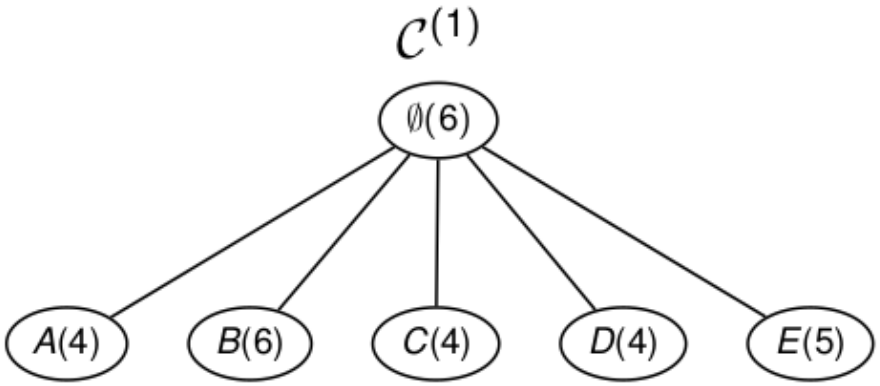
EXTENDPREFIXTREE ($\mathcal{C}^{(k)}$):

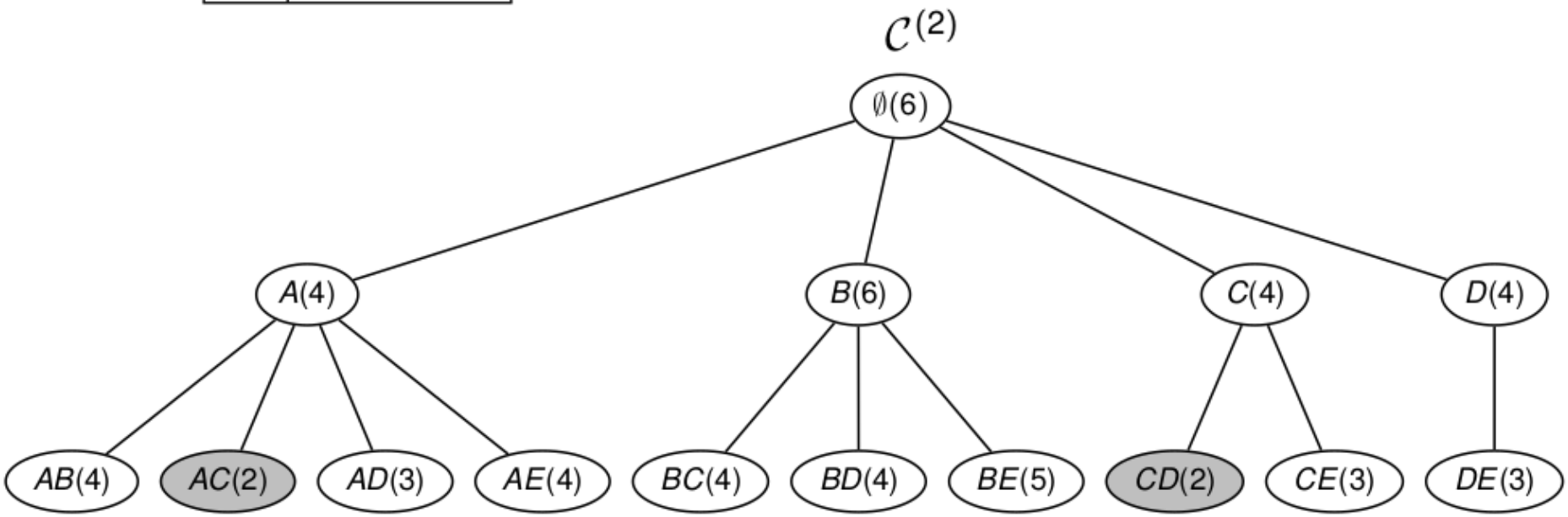
foreach *leaf* $X_a \in \mathcal{C}^{(k)}$ **do**
  foreach *leaf* $X_b \in \text{SIBLING}(X_a)$, *such that* $b > a$ **do**
    $X_{ab} \leftarrow X_a \cup X_b$
    `// prune candidate if there are any infrequent subsets`
    if $X_j \in \mathcal{C}^{(k)}$, ***for all*** $X_j \subset X_{ab}$, *such that* $|X_j| = |X_{ab}| - 1$ **then**
      Add X_{ab} as child of X_a with $sup(X_{ab}) \leftarrow 0$
  if *no extensions from* X_a **then**
    remove X_a, and all ancestors of X_a with no extensions, from $\mathcal{C}^{(k)}$
return $\mathcal{C}^{(k)}$

Apriori

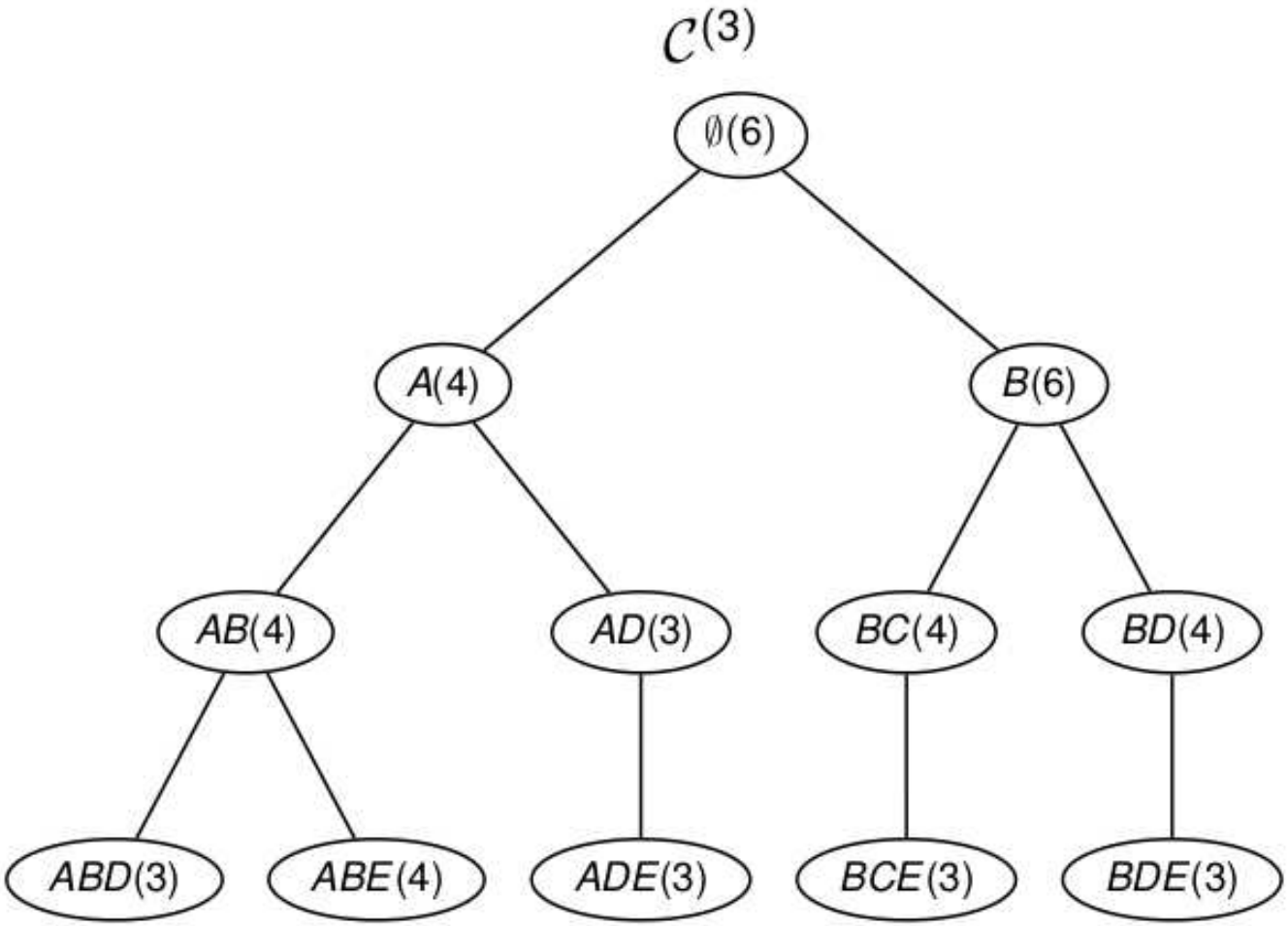
D

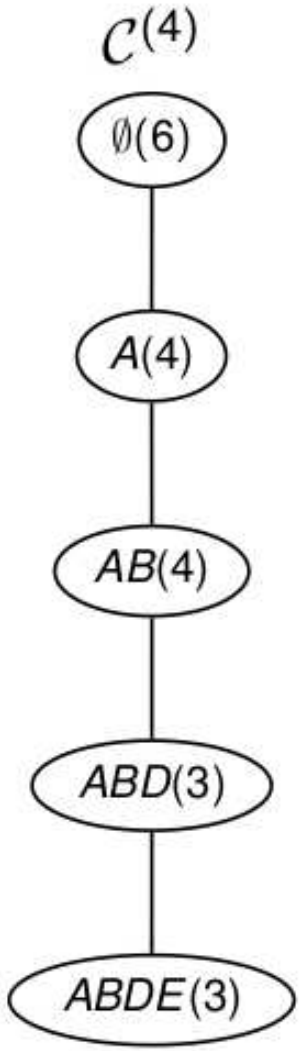
t	i(t)
1	*ABDE*
2	*BCE*
3	*ABDE*
4	*ABCE*
5	*ABCDE*
6	*BCD*





Apriori


$C^{(3)}$
∅(6)
A(4)
B(6)
AB(4)
AD(3)
BC(4)
BD(4)
ABD(3)
ABE(4)
ADE(3)
BCE(3)
BDE(3)

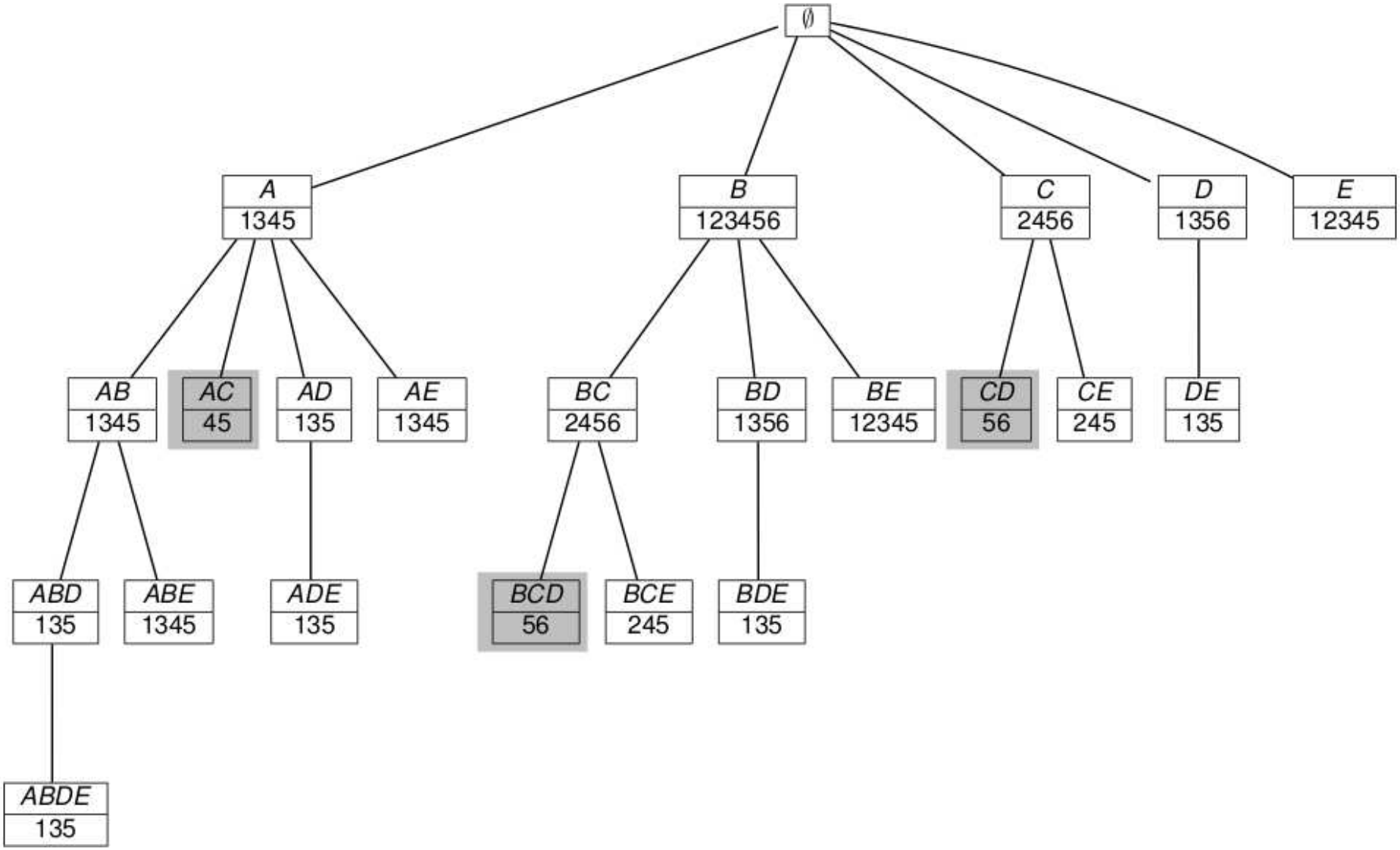

$C^{(4)}$
∅(6)
A(4)
AB(4)
ABD(3)
ABDE(3)

Eclat

Eclat

- Melhora o desempenho da contagem do suporte
- Método baseado na intersecção de *tidsets* e os indexa na estrutura de dados
- O suporte de um *itemset* candidato pode ser computado pela intersecção dos *tidsets* dos subconjuntos
  - Dados $t(X)$ e $t(Y)$, então $t(XY) = t(X) \cap t(Y)$
  - $sup(XY) = |t(XY)|$

Eclat


∅
A
1345
B
123456
C
2456
D
1356
E
12345
AB
1345
AC
45
AD
135
AE
1345
BC
2456
BD
1356
BE
12345
CD
56
CE
245
DE
135
ABD
135
ABE
1345
ADE
135
BCD
56
BCE
245
BDE
135
ABDE
135

Eclat

// Initial Call: $\mathcal{F} \leftarrow \emptyset, P \leftarrow \{\langle i, \mathbf{t}(i)\rangle \mid i \in \mathcal{I}, |\mathbf{t}(i)| \geq minsup\}$

ECLAT (*P*, *minsup*, $\mathcal{F}$):

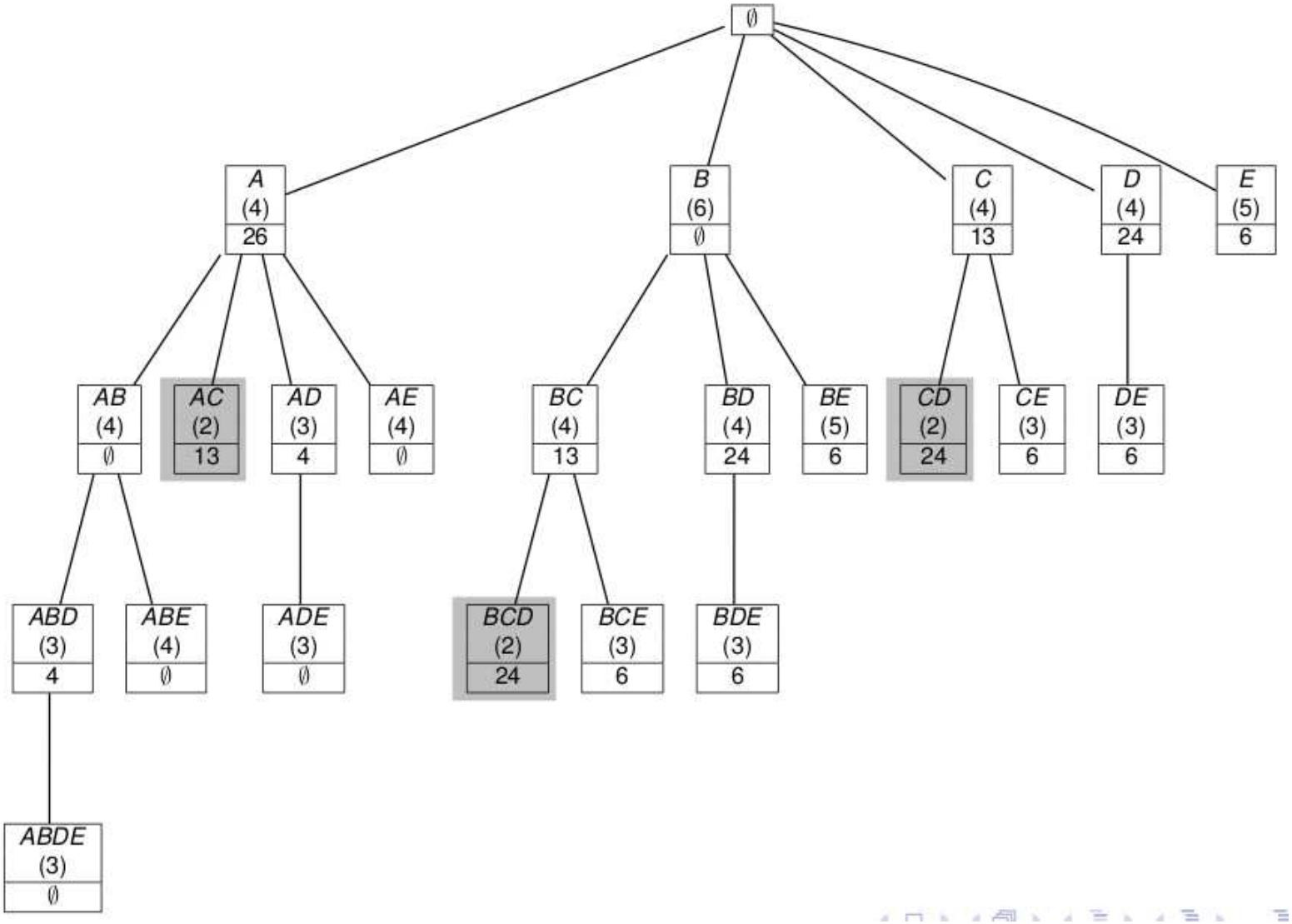
1 **foreach** $\langle X_a, \mathbf{t}(X_a)\rangle \in P$ **do**
2 $\quad \mathcal{F} \leftarrow \mathcal{F} \cup \{(X_a, sup(X_a))\}$
3 $\quad P_a \leftarrow \emptyset$
4 $\quad$ **foreach** $\langle X_b, \mathbf{t}(X_b)\rangle \in P$, *with* $X_b > X_a$ **do**
5 $\quad\quad X_{ab} = X_a \cup X_b$
6 $\quad\quad \mathbf{t}(X_{ab}) = \mathbf{t}(X_a) \cap \mathbf{t}(X_b)$
7 $\quad\quad$ **if** $sup(X_{ab}) \geq minsup$ **then**
8 $\quad\quad\quad P_a \leftarrow P_a \cup \{\langle X_{ab}, \mathbf{t}(X_{ab})\rangle\}$
9 $\quad$ **if** $P_a \neq \emptyset$ **then** ECLAT (P_a, *minsup*, $\mathcal{F}$)

dEclat

dEclat

- Melhora o desempenho do *Eclat* reduzindo os *tidsets* intermediários
- Guarda as diferenças ao invés de todos os *tidsets*
- Sejam $X_a = \{x_1, \ldots, x_j, x_a\}$ e $X_b = \{x_1, \ldots, x_j, x_b\}$, então $X_{ab} = X_a \cup X_b = \{x_1, \ldots, x_j, x_a, x_b\}$
- No processo de busca, pode-se guardar apenas as diferenças $d(X_{ab}) = t(X_a) \setminus t(X_{ab}) = t(X_a) \setminus t(X_b)$
- Pode-se determinar $d(X_{ab})$ com base nas diferenças como $d(X_{ab}) = d(X_b) \setminus d(X_a)$
- $sup(X_{ab}) = sup(X_a) - |d(X_{ab})|$
- Assim, é possível substituir as operações de intersecção do Eclat por operações de diferença entre conjuntos

dEclat


∅
A (4) 26
B (6) ∅
C (4) 13
D (4) 24
E (5) 6
AB (4) ∅
AC (2) 13
AD (3) 4
AE (4) ∅
BC (4) 13
BD (4) 24
BE (5) 6
CD (2) 24
CE (3) 6
DE (3) 6
ABD (3) 4
ABE (4) ∅
ADE (3) ∅
BCD (2) 24
BCE (3) 6
BDE (3) 6
ABDE (3) ∅

dEclat

// Initial Call: $\mathcal{F} \leftarrow \emptyset$,
$P \leftarrow \{\langle i, \mathbf{d}(i), sup(i)\rangle \mid i \in \mathcal{I}, \mathbf{d}(i) = \mathcal{T} \setminus \mathbf{t}(i), sup(i) \geq minsup\}$

dEclat (P, *minsup*, $\mathcal{F}$):

foreach $\langle X_a, \mathbf{d}(X_a), sup(X_a)\rangle \in P$ **do**
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X_a, sup(X_a))\}$
  $P_a \leftarrow \emptyset$
  foreach $\langle X_b, \mathbf{d}(X_b), sup(X_b)\rangle \in P$, *with* $X_b > X_a$ **do**
    $X_{ab} = X_a \cup X_b$
    $\mathbf{d}(X_{ab}) = \mathbf{d}(X_b) \setminus \mathbf{d}(X_a)$
    $sup(X_{ab}) = sup(X_a) - |\mathbf{d}(X_{ab})|$
    if $sup(X_{ab}) \geq minsup$ **then**
      $P_a \leftarrow P_a \cup \{\langle X_{ab}, \mathbf{d}(X_{ab}), sup(X_{ab})\rangle\}$
  if $P_a \neq \emptyset$ **then** dEclat (P_a, *minsup*, $\mathcal{F}$)

Gerando Regras de Associação

Gerando Regras de Associação

- Para cada *itemset* frequente $Z \in F$, deve-se gerar as regras $X \rightarrow Y$, onde $Y = Z \setminus X$
- A regra XY deve ser frequente $(sup(XY) > minsup)$
- Depois computa-se a confiança $conf(X \rightarrow Y) = \dfrac{sup(XY)}{sup(X)}$
  - $conf(X \rightarrow Y) \geq minconf$, ou seja, as confianças devem atender a um limiar de confiança mínima
  - Se $conf(X \rightarrow Y) < minconf$, então $conf(W \rightarrow Z \setminus W) < minconf \ \forall \ W \subset X$ pois $sup(W) \geq sup(X)$
  - Não é necessário investigar os subconjuntos de X

Gerando Regras de Associação

AssociationRules ($\mathcal{F}$, *minconf*):

1 **foreach** $Z \in \mathcal{F}$, *such that* $|Z| \geq 2$ **do**
2 $\mathcal{A} \leftarrow \{X \mid X \subset Z, X \neq \emptyset\}$
3 **while** $\mathcal{A} \neq \emptyset$ **do**
4 $X \leftarrow$ maximal element in $\mathcal{A}$
5 $\mathcal{A} \leftarrow \mathcal{A} \setminus X$ // remove X from $\mathcal{A}$
6 $c \leftarrow sup(Z)/sup(X)$
7 **if** $c \geq minconf$ **then**
8 print $X \longrightarrow Y, sup(Z), c$
9 **else**
10 $\mathcal{A} \leftarrow \mathcal{A} \setminus \{W \mid W \subset X\}$
 // remove all subsets of X from $\mathcal{A}$

Lift

- Uma vez encontradas as regras de associação que atendem às restrições impostas, essas podem ser ordenadas segundo um critério de interesse
- O *Lift* é uma medida de correlação
- $lift(X \rightarrow Y) = \frac{conf(X \rightarrow Y)}{sup(Y)} = \frac{sup(XY)}{sup(X)sup(Y)}$
  - se $lift(X \rightarrow Y) > 1$ então X e Y são positivamente correlacionados e a ocorrência de um implica na ocorrência do outro
  - se $lift(X \rightarrow Y) < 1$ então a ocorrência de X é negativamente correlata a Y
  - se $lift(X \rightarrow Y) = 1$, então X e Y são independentes $(P(XY) = P(X)P(Y))$