



Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

# REGUŁY ASOCJACYJNE

Nguyen Hung Son

Wydział Matematyki, Informatyki i Mechaniki  
Uniwersytet Warszawski

25 lutego i 04 marca 2005



# OUTLINE

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

1 DANE TRANSAKCYJNE

2 Reguły asocjacyjne

3 Szukanie reguł asocjacyjnych

4 Ulepszenie algorytmu Apriori

5 FP-tree



# MOTYWACJE

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie algorytmu Apriori

FP-tree

## LISTA AUTORÓW (ITEMS)

A	Jane Austen
C	Agatha Christie
D	Sir Arthur Conan Doyle
T	Mark Twain
W	G. Wodehouse

## TRANSAKCJE

TID	Kupione książki				
10	A	C	T	W	
20		C	D	W	
30	A	C	T	W	
40	A	C	D	W	
50	A	C	D	T	W
60		C	D	T	

**Znajdź wzorce zachowań klientów., np.**

- Co jest często kupowane? (modne?, sezonowe?)
- Które tytuły są kupione razem?
- Co robić, aby przyciągać klientów?
- ...



# W JAKIEJ FORMIE WYRAZIĆ WZORZEC ZACHOWAŃ KLIENTÓW?

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- macierz kolokacji:

	<i>A</i>	<i>C</i>	<i>D</i>	<i>T</i>	<i>W</i>
<i>A</i>	4	4	2	3	4
<i>C</i>	4	6	4	4	5
<i>D</i>	2	4	4	2	<i>W</i>
<i>T</i>	3	4	2	4	<i>W</i>
<i>W</i>	4	5	3	3	5

- Reguły
  - $A \implies C$ ,
  - $C \implies W$ ,
  - $AC \implies T$ ,
  - $T \implies ACW$ .



# OUTLINE

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- 1 Dane transakcyjne
- 2 REGUŁY ASOCJACYJNE**
- 3 Szukanie reguł asocjacyjnych
- 4 Ulepszenie algorytmu Apriori
- 5 FP-tree



Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

## PODSTAWOWE OZNACZENIA

- Pozycje (ang. items) opisują dostępne towary. Zakłada się, że zbiorem wszystkich towarów jest  $I = \{i_1, i_2, \dots, i_m\}$  (*items*)
- baza transakcji  $D = \{(tid_1, T_1), (tid_2, T_2)\dots\}$  zawiera transakcje jako pary  $(tid_j, T_j)$ , gdzie:
  - $tid_j$  : unikalny identyfikator
  - $T_j \subset I$  : zbiór zakupionych towarów .
- itemset: każdy podzbiór zbioru towarów  $I$ ;
- $k$ -itemset: podzbiór o  $k$  elementach.
- $s(X)$  - liczba transakcji zawierających itemset  $X$ .



Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

## DEFINITION

**Regułą asocjacyjną** nazywamy każdą implikację typu

$$X \implies Y$$

gdzie  $X, Y$  są itemsetami. Jakość takiej reguły mierzymy  
jest funkcjami:

- wsparcie (support)

$$\text{support}(X \implies Y) = s(X \cup Y)$$

- wiarygodność (confidence)

$$\text{confidence}(X \implies Y) = \frac{s(X \cup Y)}{s(X)}$$



# PRZYKŁAD

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

## LISTA AUTORÓW (ITEMS)

A	Jane Austen
C	Agatha Christie
D	Sir Arthur Conan Doyle
T	Mark Twain
W	G. Wodehouse

## TRANSAKCJE

TID	Kupione książki				
10	A	C	T	W	
20		C	D	W	
30	A	C	T	W	
40	A	C	D	W	
50	A	C	D	T	W
60		C	D	T	

Reguły	wsparcie	st. wiar.
$A \implies C$	4	100%
$C \implies W$	5	83,3%
$AC \implies T$	4	75%
$T \implies ACW$	3	75%





# SFORMUŁOWANIE PROBLEMU

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

## PROBLEM

### DANE SĄ:

- zbiór pozycji  $I = \{i_1, i_2, \dots, i_m\}$
- baza transakcji  $D = \{(tid_1, T_1), (tid_2, T_2)\dots\}$
- stałe  $sup\_min$  = minimalna wartość wsparcia i  
 $conf\_min$  = minimalny stopień wiarygodności

**PROBLEM:** Znaleźć wszystkie reguły asocjacyjne o

- wsparciu  $\geq sup\_min$
- stopniu wiarygodności  $\geq conf\_min$



# OUTLINE

## Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

## Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

1 Dane transakcyjne

2 Reguły asocjacyjne

**3 SZUKANIE REGUŁ ASOCJACYJNYCH**

4 Ulepszenie algorytmu Apriori

5 FP-tree



# PROBLEM OBLICZENIOWY

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- Liczba wszystkich reguł asocjacyjnych wynosi  $3^n$ , gdzie  $n$  jest liczbą itemów.
- Sprawdzanie wszystkich reguł jest nie wykonywalne!
- Proponowano różne metody szukania z użyciem różnych technik obliczeń: sekwencyjne, równoległe.



# SCHEMAT WYSZUKIWANIA REGUŁ

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

Większość istniejących algorytmów działa w dwóch krokach:

**Znajdź częste zbiory:** znajdź itemsets o wsparciu większym niż  $min\_sup$  (frequent itemsets).

**Podział częstych zbiorów:** dla każdego częstego zbioru, znajdź podziały tego zbioru na 2 podzbiory w taki sposób, by powstały reguły o st. wiarygodności większym niż  $min\_conf$ .



# PRZYKŁAD

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie algorytmu Apriori

FP-tree

## TRANSAKCJE

TID	Kupione książki			
10	A	C	T	W
20		C	D	W
30	A	C	T	W
40	A	C	D	W
50	A	C	D	T
60		C	D	T

## CZĘSTE ZBIORY

wsparcie	Frequent itemsets
100% (6)	C
83% (5)	CW
67% (4)	A, D, T, <b>AC</b> , AW, CD, CT, ACW
50% (3)	AT, DW, TW, ACT, ATW, CDW, CTW, ACTW

- Min. wsparcie = 3 (50%)
- Min. wiarygodność = 75%
- Reguły dla **AC**:
  - $A \implies C$  (100% wiarygodności)
  - $C \implies A$  (66% wiarygodność)



Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- **Obserwacje:**

- Jeśli  $\{A, B\}$  jest częstym zbiorem, to  $\{A\}$  i  $\{B\}$  też muszą być częstymi zbiorami.
- Ogólniej: jeśli  $X$  częstym  $k$ -elementowym zbiorem, to wszystkie  $(k - 1)$ -elementowe podzbiory  $X$  też są częste.

- **Idea:**

- Znajdź wszystkie 1-elementowe częste zbiory
- Generuj 2-elementowe częste zbiory z 1-elementowych częstych zbiorów
- ...
- Generuj  $k$ -elementowe częste zbiory poprzez łączenie  $(k - 1)$ -elementowych częstych zbiorów



# IDEA ALGORYTMU APRIORI

Data mining

Nguyen Hung Son

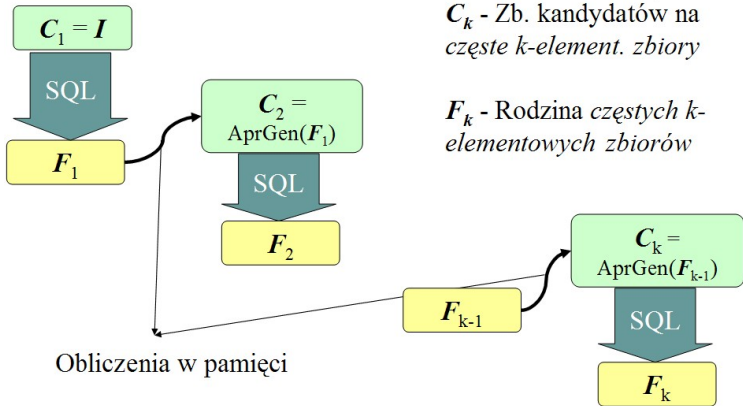
Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree





# ALGORYTM APRIORI

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

## ALGORYTM APRIORI

- 1:  $\mathbf{C}_1 := I$ ;  $\mathbf{F}_1 :=$  rodzina 1-elem. zbiorów częstych
- 2: **for** ( $k = 2$ ;  $\mathbf{F}_{k-1} \neq \emptyset$ ;  $k++$ ) **do**
- 3:      $\mathbf{C}_k := \text{AprioriGen}(\mathbf{F}_{k-1})$ ;
- 4:     //generowanie nowych kandydatów
- 5:      $\mathbf{F}_k := \{X \in \mathbf{C}_k : \text{support}(X) \geq \text{min\_sup}\}$
- 6: **end for**
- 7: Wynik :=  $\mathbf{F}_1 \cup \mathbf{F}_2 \cup \mathbf{F}_3 \dots \cup \mathbf{F}_k$ ;





# PRZYKŁAD

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie algorytmu Apriori

FP-tree

- minimalne wsparcie = 3
- ufność = 80%

## TRANSAKCJE

TID	Kupione książki				
10	A	C		T	W
20		C	D		W
30	A	C		T	W
40	A	C	D		W
50	A	C	D	T	W
60		C	D	T	

## CZĘSTE ZBIORY

	Frequent itemsets
<b>F<sub>1</sub></b>	A, C, D, T, W
<b>C<sub>2</sub></b>	AC, AD, AT, AW ...
<b>F<sub>2</sub></b>	AC, AT, AW, CD, CT, CW, DW, TW
<b>C<sub>3</sub></b>	ACT, ACW, ATW, CDW, CTW
<b>F<sub>3</sub></b>	ACT, ACW, ATW, CDW
<b>C<sub>4</sub></b>	



# FUNKCJA APRIORIGEN

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

**Funkcja AprioriGen( $F_{k-1}$ ) posiada dwa główne kroki:**

- **Łączenie:** do  $C_k$  wstawiamy sumy takich par  $X, Y \in F_{k-1}$ , które mają wspólne  $k - 2$  początkowych elementów. Np. dla

$$F_{k-1} = \{AB, AC, AD, AE, BC, BD, BE\}$$

mamy

$$C_k = \{ABC, ABD, ABE, ACD, ACE, \\ ADE, BCD, BCE, BDE\}$$

- **Obcinanie:** Usuwamy z  $C_k$  te zbiory, których nie wszystkie podzbiory ( $k - 1$ )-elementowe są w  $F_{k-1}$ . Np. możemy usuwać ACD, ponieważ CD nie znajduje się w  $F_{k-1}$ . Po obcinaniu otrzymujemy

$$C_k = \{ABC, ABD, ABE\}$$



Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- **Problem:** Niech  $X$  będzie zbiorem częstym. Znaleźć  $Y \subset X$  taki, że

$$confidence(X \setminus Y \implies Y) > min\_conf$$

- **Obserwacja:**  
"Jeśli  $AB \implies CD$  jest wiarygodną regułą, to reguły  $ABC \implies D$  i  $ABD \implies C$  też są"
- **Strategie:**
  - Przerzucać na prawą stronę po kolei pojedyncze elementy.
  - Stosować funkcję *AprioriGen()* do generowania zbioru warunkowanego  $Y$



# OUTLINE

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- 1 Dane transakcyjne
- 2 Reguły asocjacyjne
- 3 Szukanie reguł asocjacyjnych
- 4 ULEPSZENIE ALGORYTMU APRIORI**
- 5 FP-tree



# ULEPSZENIE ALGORYTMU APRIORI

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- Algorytm Apriori musi przeglądać całą bazę danych w celu obliczenia wsparcia dla kandydatów
- **Ulepszenie:** nowa struktura, która zawiera wyłącznie transakcje, które mogą wspierać aktualnych kandydatów.
  - 1 *counting\_base*: nowa struktura danych, która jest uaktualniana dla każdego kroku  $k$ ;
  - 2 Algorytm *AprioriTid*: oblicza wsparcie dla kandydatów skanując wyłącznie strukturę *counting\_base*;



# ALGORYTM APRIORI<sup>TID</sup>

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

## APRIORI<sup>TID</sup>

**Wejście:** zbiór transakcji  $D$ ,  $min\_sup$  - minimalne wsparcie

**Wyjście:** zbiór wszystkich częstych itemsetów  $F$

*//CB<sub>k</sub> - zbiór counting\_base obliczony w k-tym kroku*

- 1:  $C_1 := I$ ;  $F_1 :=$  rodzina 1-elem. zbiorów częstych
- 2: **for** ( $k = 2$ ;  $F_{k-1} \neq \emptyset$ ;  $k++$ ) **do**
- 3:      $C_k :=$  **AprioriGen**( $F_{k-1}$ );
- 4:     *//generowanie nowych kandydatów*
- 5:      $CB_k =$  **Counting\_base\_generate** ( $C_k, CB_{k-1}$ );
- 6:     **Support\_count**( $C_k, CB_k$ );
- 7:      $F_k := \{X \in C_k : support(X) \geq min\_sup\}$
- 8: **end for**
- 9: Wynik :=  $F_1 \cup F_2 \cup F_3 \dots \cup F_k$ ;



# GENEROWANIE STRUKTURĘ *Counting\_base*

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- $\mathbf{CB}_k$  = skojarzy każdą transakcję  $t$  z listą kandydatów występujących w  $t$ ;
- Elementami  $\mathbf{CB}_k$  są pary:

$$(t.TID, \{c \in \mathbf{C}_k | c \subset t\}) = (t.TID, S_k(t.TID))$$

- Jeśli jakaś transakcja nie zawiera kandydatów  $k$ -elementowych, to zostanie ona usunięta z  $\mathbf{CB}_k$  i ze wszystkich następnych zbiorów *counting\_base*;
- Można to wyznaczyć metodą iteracyjną:
  - $\mathbf{CB}_1 :=$  cała baza transakcji
  - $\mathbf{CB}_k := \{(i, S_k(i))\}$ , gdzie  $S_k(i)$  powstaje z  $S_{k-1}(i)$  w następujący sposób:

$$\begin{aligned} \text{JEŚLI } \{u_1 \dots u_{i-2}, a\} \text{ i } \{u_1 \dots u_{i-2}, b\} \in \mathbf{F}_{i-1} \cap \mathbf{CB}_{k-1} \\ \text{TO } \{u_1, \dots, u_{i-2}, a, b\} \in S_k(i) \end{aligned}$$



# PRZYKŁAD

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

$D = \{(1,acd), (2, bce), (3,abce), (4,be)\}$ .

$\text{min\_sup} = 0.5$

**Krok 1**

$\text{counting\_base} = \{(1, \{a,c,d\}), (2, \{b,c,e\}),$   
 $(3, \{a,b,c,e\}), (4, \{b, e\}) \}$

$F_1 = \{a, b, c, e\}$

$C_2 = \{ab, ac, ae, bc, be, ce\}$

**Krok 2**

$\text{counting\_base} = \{(1, \{ac\}), (2, \{bc, be, ce\}),$   
 $(3, \{ab, ac, ae, bc, be, ce\}), (4, \{be\}) \}$

$F_2 = \{ac, bc, be, ce\}$

$C_3 = \{bce\}$

**Krok 3**

$\text{counting\_base} = \{(2, \{bce\}), (3, \{bce\})\}$

$F_3 = \{bce\}$





- AprioriTid przeszukuje tablice  $CB_k$  zamiast skanowaia całą bazę transakcyjną;
  - Jest efektywny wtedy, gdy  $CB_k$  jest dostatecznie mała względem rozmiaru całej bazy.
- AprioriTid jest lepszy od Apriori wtedy, gdy
  - $CB_k$  mieści się w pamięci;
  - Częste zbiory mają rozkład z "długim ogonkiem"!
- AprioriHybrid
  - wykonuje Apriori w pierwszych iteracjach
  - przełączy na AprioriTid wtedy, gdy spodziewmy, że  $CB_k$  mieści się w pamięci.
- W praktyce, AprioriHybrid może być do 30% szybszy od Apriori i do 60% szybszy niż AprioriTid



# OUTLINE

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- 1 Dane transakcyjne
- 2 Reguły asocjacyjne
- 3 Szukanie reguł asocjacyjnych
- 4 Ulepszenie algorytmu Apriori
- 5 FP-TREE**



## Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- Baza danych jest zapamiętana w oszczędnej strukturze zwanej FP-tree.
- Częste zbiory są obliczone z tego drzewa;
- Jest to metoda “Dziel i rządź”;
- Baza danych jest przeskanowana dokładnie 2 razy;
  - pierwszy raz: częstość wystąpienia każdego przedmiotu (item);
  - drugi raz: Konstrukcja drzewa FP-tree
- O rząd wielkości szybszy niż Apriori.



# ILUSTRACJA

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- Baza transakcji ( $min\_sup = 3$ ):

TID	Items
1	f, a, c, d, g, i, m, p
2	a, b, c, f, l, m, o
3	b, f, h, j, o
4	b, c, k, s, p
5	a, f, c, e, l, p, m, n



# ILUSTRACJA

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie algorytmu Apriori

FP-tree

- Baza transakcji ( $min\_sup = 3$ ):

TID	Items
1	f, a, c, d, g, i, m, p
2	a, b, c, f, l, m, o
3	b, f, h, j, o
4	b, c, k, s, p
5	a, f, c, e, l, p, m, n

- Po pierwszym skanowaniu całej bazy:

f	c	a	b	m	p	l	o	d	e	g	h	i	j	k
4	4	3	3	3	3	2	2	1	1	1	1	1	1	1

- Po usunięciu przedmiotów o wsparciu  $< 3$  mamy posortowaną listę częstych przedmiotów (item):

Item	f	c	a	b	m	p
Frequency	4	4	3	3	3	3



## Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- Ponownie skanuje bazę. Dla każdej transakcji:



## Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- Ponownie skanuje bazę. Dla każdej transakcji:
  - usuwamy nieczęste items,



## Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- Ponownie skanuje bazę. Dla każdej transakcji:
  - 1 usuwamy nieczęste items,
  - 2 **sortujemy items, i**





## Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- Ponownie skanuje bazę. Dla każdej transakcji:
  - 1 usuwamy nieczęste items,
  - 2 sortujemy items, i
  - 3  **dodajemy ją do FP-tree**



- Ponownie skanuje bazę. Dla każdej transakcji:
  - 1 usuwamy nieczęste items,
  - 2 sortujemy items, i
  - 3 dodajemy ją do FP-tree

- przykład

1	f, c, a, m, p
2	f, c, a, b, m
3	f, b
4	c, b, p
5	f, c, a, m, p



- Ponownie skanuje bazę. Dla każdej transakcji:
  - 1 usuwamy nieczęste items,
  - 2 sortujemy items, i
  - 3 dodajemy ją do FP-tree

- przykład

1	f, c, a, m, p
2	f, c, a, b, m
3	f, b
4	c, b, p
5	f, c, a, m, p

- Dodajemy kolejną transakcję do drzewa:



Data mining

Nguyen Hung Son

Dane transakcyjne

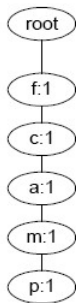
Reguły asocjacyjne

Szukanie

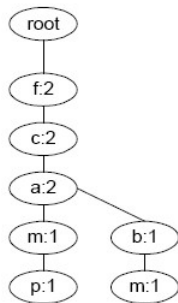
Ulepszenie  
algorytmu Apriori

FP-tree

tr1: f, c, a, m, p



tr2: f, c, a, b, m





Data mining

Nguyen Hung Son

Dane transakcyjne

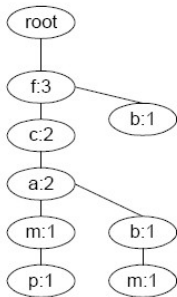
Reguły asocjacyjne

Szukanie

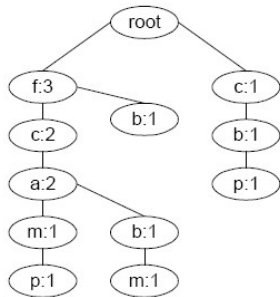
Ulepszenie  
algorytmu Apriori

FP-tree

tr3: f, b



tr4: c, b, p





Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

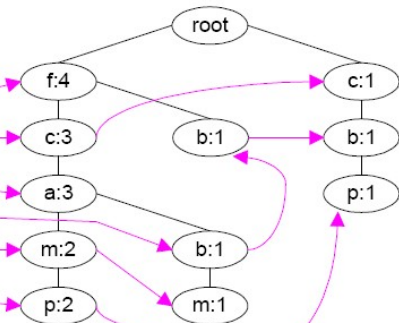
Ulepszenie  
algorytmu Apriori

FP-tree

tr5: f, c, a, m, p

Header Table

f:4	
c:4	
a:3	
b:3	
m:3	
p:3	





# SZUKANIE CZĘSTYCH ZBIORÓW Z DRZEWA FP-TREE

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

- Proces przeszukiwania sprawdza pojedyncze przedmioty w tablicy “header table” (od dołu do góry);
- Dla każdego przedmiotu  $x$ :
  - Skonstruować bazę warunkowych wzorców;
    - “Ścieżka prefiksowa” (prefix path) = ścieżka prowadząca od korzenia do wierzchołka  $x$ ;
    - Np. ścieżki prefiksowe dla  $p$ :  $[f : 2, c : 2, a : 2, m : 2]$  i  $[c : 1, b : 1]$ ;
  - Skonstruować warunkowe drzewo FP-tree
    - Traktujemy bazę warunkowych wzorców dla  $x$  jako małą bazę transakcji  $D(x)$ ;
    - Możemy skonstruować FP-tree dla  $D(x)$ ;
    - Jeśli drzewo posiada tylko jedną ścieżkę, to zatrzymujemy i wypisujemy częste zbiory;
    - W przeciwnym przypadku, powtarzamy ten proces.



# PRZYKŁAD

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie algorytmu Apriori

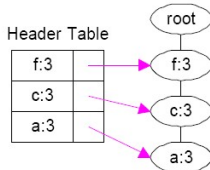
FP-tree

1 Dla p:

- Baza warunkowych wzorców dla p: [f:2, c:2, a:2, m:2], [c:1, b:1]
- c jest jedynym częstym przedmiotem i warunkowe FP-tree ma 1 wierzchołek (c:3).
- Więc {p,c} jest jedynym częstym zbiorem.

2 Dla m:

- Baza warunkowych wzorców dla p: [f:2, c:2, a:2], [f:1, c:1, a:1, b:1]
- f, c, a są częstymi przedmiotami



- warunkowe FP-tree:
- Odczytujemy częste zbiory: {f,m}, {c,m}, {a,m}, {f,c,m}, {f,a,m}, {c,a,m}, {f,c,a,m}.





# ALGORYTM FP-GROWTH

Data mining

Nguyen Hung Son

Dane transakcyjne

Reguły asocjacyjne

Szukanie

Ulepszenie  
algorytmu Apriori

FP-tree

## FP-GROWTH( $Tree, \alpha$ )

- 1: **if**  $Tree$  contains a single path  $P$  **then**
- 2:     **for** each combination  $\gamma$  of the nodes in  $P$  **do**
- 3:         generate pattern  $\gamma \cup \alpha$  with  
           support = minimum support of nodes in  $\gamma$ .
- 4:     **end for**
- 5: **else**
- 6:     **for** each  $a_i$  in the header table of  $Tree$  **do**
- 7:         generate pattern  $\beta = a_i \cup \alpha$  with  
           support =  $a_i.support$
- 8:         construct conditional pattern base for  $\beta$  and  
           conditional FP-tree  $Tree_\beta$
- 9:         **if**  $Tree_\beta \neq \emptyset$  **then**
- 10:             call FP-GROWTH( $Tree_\beta, \beta$ )
- 11:         **end if**
- 12:     **end for**
- 13: **end if**