



**Zbiory częste i reguły  
asocjacyjne**


Marzena Kryszkiewicz  
Politechnika Warszawska

1



**Definicje i własności**


2



**Reguły asocjacyjne - nieformalnie**

- Załóżmy, że pozycja {*ryba*} występuje w 5% transakcji sprzedaży, a zbiór pozycji {*ryba*, *białe wino*} występuje w 4% transakcji. Ta informacja umożliwia wyprowadzenie *reguły asocjacyjnej*, mówiącej, że w 80% transakcji, w których zakupiono *rybę*, zakupiono także *białe wino*.
- Do wyprowadzania takich reguł potrzebna jest informacja o tym, ile transakcji wspiera stosowne *zbiory pozycji* (ang. *itemsets*).

3




**Wsparcie zbioru pozycji**

- Niech  $D$  będzie zbiorem *transakcji*.
- *Wsparcie zbioru pozycji*  $X$ , oznaczane jako  $sup(X)$ , jest liczbą transakcji w  $D$ , które zawierają wszystkie pozycje z  $X$ , to jest:

$$sup(X) = |\{T \in D \mid X \subseteq T\}|.$$

4




**Przykład: Wsparcia zbiorów pozycji**

Przykładowy zbiór transakcji  $D$

Id	Transakcja
$T_1$	ABCDEG
$T_2$	ABCDEF
$T_3$	ABCDEH
$T_4$	ABDE
$T_5$	ACDEH
$T_6$	BCE

- $sup(\{ABC\}) = 3$ ,  $sup(\{EH\}) = 2$ .

5




**Względne wsparcie zbioru pozycji**

- *Względne wsparcie zbioru pozycji*  $X$ , oznaczane jako  $rSup(X)$ , jest stosunkiem transakcji w  $D$ , które zawierają wszystkie pozycje z  $X$ , do liczby wszystkich transakcji w  $D$ :

$$rSup(X) = sup(X) / |D|.$$

- **Uwaga:**  $rSup(X)$  może być traktowane jako oszacowanie prawdopodobieństwa wystąpienia zbioru pozycji  $X$  w  $D$ .


6



### Częste zbiory pozycji

- $X$  jest definiowany jako częsty zbiór pozycji (ang. frequent itemset), jeżeli
 
$$\text{sup}(X) > \text{minSup},$$
 gdzie  $\text{minSup}$  jest wartością progową wsparcia, zadaną przez użytkownika.
- Podstawowa własność zbiorów pozycji:** Wsparcia nadzbiorów zbioru  $X$  nie są większe od  $\text{sup}(X)$ .

7



### Przykład: Wsparcia zbiorów pozycji

Przykładowy zbiór transakcji D


Id	Transakcja
$T_1$	ABCDEG
$T_2$	ABCDEF
$T_3$	ABCDEH
$T_4$	ABDE
$T_5$	ACDEH
$T_6$	BCE

- $\text{sup}(\{ABC\}) = 3, \text{sup}(\{EH\}) = 2$ .
- Niech  $\text{minSup} = 2$ . Wtedy:  $\{ABC\}$  jest częsty,  $\{EH\}$  nie jest częsty (czyli jest rzadki).
- Wsparcia wszystkich nadzbiorów zbioru  $\{EH\}$  także nie są większe niż 2, stąd nadzbiory zbioru  $\{EH\}$  nie są częste.
- Jednakże wsparcia podzbiorów (właściwych) zbioru  $\{EH\}$  mogą być większe niż 2. Zatem może się zdarzyć, że (niektóre) podzbiory zbioru  $\{EH\}$  są częste.

8

7


8



### Reguły asocjacyjne (ARs)

- Reguła asocjacyjna jest wyrażeniem wiążącym dwa rozłączne zbiory pozycji:
 
$$X \rightarrow Y,$$
 gdzie  $\emptyset \neq Y \subseteq I \setminus X \subseteq I \setminus Y$ .
- O regule  $X \rightarrow Y$  mówi się, że jest oparta na zbiorze pozycji  $X \cup Y$ , przy czym:
  - zbiór  $X \cup Y$  jest nazywany bazą reguły  $X \rightarrow Y$ ,
  - $X$  – jej poprzednikiem,
  - $Y$  – jej następnikiem.

9




### Wsparcie reguły asocjacyjnej

- Wsparcie reguły  $X \rightarrow Y$  jest definiowane jako liczba transakcji zawierających bazę tej reguły, czyli:
 
$$\text{sup}(X \rightarrow Y) = \text{sup}(X \cup Y).$$
- Względne wsparcie reguły  $X \rightarrow Y$  jest definiowane jako względne wsparcie jej bazy:
 
$$r\text{Sup}(X \rightarrow Y) = r\text{Sup}(X \cup Y).$$

10

9

10



### Przykład: Reguły asocjacyjne

Zbiór transakcji D


Id	Transakcja
$T_1$	ABCDEG
$T_2$	ABCDEF
$T_3$	ABCDEH
$T_4$	ABDE
$T_5$	ACDEH
$T_6$	BCE

$\text{sup}(ABC) = 3,$   
 $\text{sup}(A) = 5.$

Stąd:

- $\text{sup}(\{A\} \rightarrow \{BC\}) = \text{sup}(\{ABC\}) = 3.$

11



### Zaufanie reguły asocjacyjnej

- Zaufanie reguły  $X \rightarrow Y$  jest definiowane jako stosunek liczby transakcji, które zawierają bazę  $X \cup Y$ , do liczby transakcji zawierających jej poprzednik  $X$ :
 
$$\text{conf}(X \rightarrow Y) = \text{sup}(X \rightarrow Y) / \text{sup}(X).$$
- Uwaga:**  $\text{conf}(X \rightarrow Y)$  może być traktowane jako oszacowanie prawdopodobieństwa warunkowego, że  $Y$  występuje w transakcji  $T$  pod warunkiem, że  $X$  występuje w  $T$ .

12

11

12

**Przykład: Reguły asocjacyjne**

Zbiór transakcji D

Id	Transakcja
$T_1$	ABCDEG
$T_2$	ABCDEF
$T_3$	ABCDEH
$T_4$	ABDE
$T_5$	ACDEH
$T_6$	BCE

$sup(ABC) = 3,$   
 $sup(A) = 5.$

Stąd:

- $sup(\{A\} \rightarrow \{BC\}) = sup(\{ABC\}) = 3,$
- $conf(\{A\} \rightarrow \{BC\}) = sup(\{ABC\}) / sup(\{A\}) = 3/5.$

13

**Silne reguły asocjacyjne**

- Silne reguły asocjacyjne (AR)** są definiowane jako te reguły asocjacyjne, których **wsparcie przekracza  $minSup$** , a **zaufanie przekracza  $minConf$** , czyli:

$$AR = \{r \in AR \mid sup(r) > minSup \wedge conf(r) > minConf\},$$

gdzie  $minSup \in [0, |D|)$  i  $minConf \in [0, 1)$ .

- Własność.** AR to te reguły asocjacyjne, których **bazy są zbiorami częstymi**, a zaufanie przekracza  $minConf$ .

14

**Zarys odkrywania silnych reguł asocjacyjnych**

Zazwyczaj AR jest odkrywane w dwóch etapach:

- Znajdź częste (ang. *frequent*) zbiory pozycji  $F$  i ich wsparcia w zbiorze transakcji D.
- Generuj AR wyłącznie z  $F$ : Niech  $Z \in F$ ,  $Z \neq \emptyset$  i  $Y \subseteq Z$ . Wtedy, kandydująca reguła  $Z \setminus Y \rightarrow Y$  jest silną regułą asocjacyjną, jeśli:

$$sup(Z) / sup(Z \setminus Y) > minConf.$$

15

**Odkrywanie zbiorów częstych przy użyciu algorytmu Apriori**

16

**Odkrywanie zbiorów częstych wg algorytmu Apriori**

- W ramach każdej iteracji  $i$ :
  - Wyznacz wsparcia kandydujących zbiorów pozycji o długości  $i$ .
  - Z kandydatów o długości  $i$ , którzy okazali się częstymi zbiorami, twórz kandydatów o długości  $i + 1$ .

17

**Przykład: Częste 1-zbiory pozycji**

Tid	Pozycje
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

- Niech  $minSup = 1$
- Iteracja 1:
  - $C_0 \rightarrow F_0: \emptyset_8$
  - $C_1 \rightarrow F_1: a_6 b_5 c_4 e_4 f_4 h_3$

18

**Przykład: Częste 2-zbiory pozycji**

Tid	Pozycje
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

- Niech minSup = 1
- Po iteracji 1:  
 $F_1: a_6 b_5 c_4 e_4 f_4 h_3$
- Iteracja 2:  
 $C_2 \rightarrow F_2: ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 bh_1$   
 $ce_2 cf_2 ch_2 ef_2 eh_0 fh_1$

Zbiory uznane za rzadkie po wyznaczeniu wsparcia.

19

**Przykład: Częste 3-zbiory pozycji**

Tid	Pozycje
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

- Niech minSup = 1
- Po iteracji 2:  
 $F_2: ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2$   
 $ce_2 cf_2 ch_2 ef_2$
- Iteracja 3:  
 $C_3 \rightarrow F_3: abc_3 abe_3 abf_1 abh_1 ace_2 acf_2 ach_2$   
 $ae_1 af_1 aeh_1 afh_1 bce_2 bcf_1 bef_2 cef_1 ceh_1$   
 $cfh_1$

Zbiory pozycji uznane za rzadkie jako nadzbiory rzadkich zbiorów.

20

**Przykład: Częste 4-zbiory pozycji**

Tid	Pozycje
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

- Niech minSup = 1
- Po iteracji 3:  
 $F_3: abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2$
- Iteracja 4:  
 $C_4 \rightarrow F_4: abce_2 acef_2 aceh_2 acfh_2$

21

**Przykład: Częste 5-zbiory pozycji**

Tid	Pozycje
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

- Niech minSup = 1
- Po iteracji 4:  
 $F_4: abce_2$
- Iteracja 5:  
 $C_5: -$

22

**Przykład: Znalezione częste zbiory pozycji**

Tid	Pozycje
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

$\emptyset_8$   
 $a_6 b_5 c_4 e_4 f_4 h_3$   
 $ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 ce_2 cf_2 ch_2 ef_2$   
 $abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2$   
 $abce_2$

- Uwaga: Niech  $n$  będzie długością najdłuższego częstego zbioru pozycji. Apriori znajduje wszystkie zbiory częste w  $n$  albo  $n+1$  iteracjach.

23

**Własności metody Apriori tworzenia kandydatów...**

Tid	Items
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

- Niech minSup = 1
- Po iteracji 3:  
 $F_3: abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2$
- Iteracja 4:  
 $C_4 \rightarrow F_4: abce_2 acef_2 aceh_2 acfh_2$
- Uwaga:  $abc$  i  $abe$  są rodzicami zbioru  $abce$ .
- Ogólna obserwacja. Rodzice każdego kandydata o dł.  $i$  są jego dwoma pierwszymi częstymi podzbiarami o dł.  $i-1$ .

24

### Własności metody Apriori tworzenia kandydatów...

- | Tid | Items |
|-----|-------|
| 1   | abce  |
| 2   | abcef |
| 3   | abch  |
| 4   | abe   |
| 5   | acfh  |
| 6   | bef   |
| 7   | h     |
| 8   | af    |
- Niech  $\text{minSup} = 1$
  - Po iteracji 3:  
 $F_3: abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2$
  - Iteracja 4:  
 $C_4 \rightarrow F_4: abce_2 acef aceh acfh$
  - Uwaga: *ace* i *acf* są rodzicami zbioru *acef*.

25

### Własności metody Apriori tworzenia kandydatów

- | Tid | Items |
|-----|-------|
| 1   | abce  |
| 2   | abcef |
| 3   | abch  |
| 4   | abe   |
| 5   | acfh  |
| 6   | bef   |
| 7   | h     |
| 8   | af    |
- Niech  $\text{minSup} = 1$
  - Po iteracji 3:  
 $F_3: abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2$
  - Iteracja 4:  
 $C_4 \rightarrow F_4: abce_2 acef aceh acfh$
  - Dlaczego zbiór *bcef* nie został utworzony?  
Ponieważ jego drugi rodzic *bcf*  $\notin F_3$ .

26



Wyznaczanie reguł asocjacyjnych na podstawie zbiorów częstych

27



### Odkrywanie reguł asocjacyjnych z użyciem algorytmu AprioriRuleGen...

- Kandydujące reguły budowane są z każdego niepustego zbioru częstego.
- Niech  $Z$  będzie danym niepustym zbiorem częstym. W iteracji  $i$  tworzone są reguły kandydujące postaci:

$$Z \setminus Y \rightarrow Y,$$

gdzie  $Y \subseteq Z$  i  $|Y| = i$ .

28



### Odkrywanie reguł asocjacyjnych z użyciem algorytmu AprioriRuleGen

- Własność.** Niech  $r_1: Z \setminus Y \rightarrow Y$  i  $r_2: Z \setminus Y' \rightarrow Y'$ , gdzie  $Y \subset Y'$ , będą regułami asocjacyjnymi.
  - $\text{conf}(r_1) \geq \text{conf}(r_2)$ ,
  - Jeśli  $\text{conf}(r_1) \leq \text{minConf}$ , to  $\text{conf}(r_2) \leq \text{minConf}$ .
- Aby zredukować liczbę reguł kandydujących, następni o długości  $i+1$  są budowane wyłącznie z następników o długości  $i$  silnych reguł asocjacyjnych.

29

29



### Przykład bezcelowego tworzenia reguły kandydującej

- Przykład.** Niech  $\text{minSup} = 1$ ,  $\text{minConf} = 60\%$ , baza =  $\{abce\}_2$ ,  $r_1: ce \rightarrow ab$  [2, 2/2] oraz  $r_2: be \rightarrow ac$  [2, 2/4] będą regułami kandydującymi, rozważanymi w iteracji 2. Oczywiście  $r_1$  jest regułą silną, a  $r_2$  nie jest silna.

Rozważmy regułę  $r_3$  także o bazie  $\{abce\}$ , której następnik będzie sumą teoriomnogościową następników reguł  $r_1$  i  $r_2$ . A zatem nowa reguła ma postać  $r_3: e \rightarrow abc$ . Stąd:

- $\text{sup}(r_3) = \text{sup}(\{abce\}) = 2$ ,
- $\text{conf}(r_3) = \frac{\text{sup}(\{abce\})}{\text{sup}(\{e\})} \leq \frac{\text{sup}(\{abce\})}{\text{sup}(\{be\})} = \text{conf}(r_2) \leq \frac{2}{4} < \text{minConf}$ .

A zatem, reguła  $r_3: e \rightarrow abc$  nie jest silna.

30

30

**Przykład: Odkrywanie ARs...**

Częste zbiory pozycji ( $\text{minSup} = 1$ ):  $\emptyset_8$   
 $a_6 b_5 c_4 e_4 f_4 h_3$   
 $ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 ce_2 cf_2 ch_2 ef_2$   
 $abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2$   
 $abce_2$

Niech  $\text{minConf} = 60\%$ ,  $Z = abce$ .

**Iteracja 1:**

- Następniki reguł kandydujących:  $Y_1 = \{a, b, c, e\}$ .
- Kandydujące reguły:
  - $bce \rightarrow a$  [2, 2/2];
  - $ace \rightarrow b$  [2, 2/2];
  - $abe \rightarrow c$  [2, 2/3];
  - $abc \rightarrow e$  [2, 2/3].

Silne reguły asocjacyjne:

- $bce \rightarrow a$  [2, 2/2];
- $ace \rightarrow b$  [2, 2/2];
- $abe \rightarrow c$  [2, 2/3];
- $abc \rightarrow e$  [2, 2/3].

31

31

**Przykład: Odkrywanie ARs...**

Częste zbiory pozycji ( $\text{minSup} = 1$ ):  $\emptyset_8$   
 $a_6 b_5 c_4 e_4 f_4 h_3$   
 $ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 ce_2 cf_2 ch_2 ef_2$   
 $abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2$   
 $abce_2$

**Iteracja 2 ( $\text{minConf} = 60\%$ ,  $Z = abce$ ):**

- Następniki dla ARs, znalezione w iteracji 1:  $Y_1 = \{a, b, c, e\}$ .
- Następniki reguł kandydujących:  $Y_2 = \{ab, ac, ae, bc, be, ce\}$ .
- Kandydujące reguły:
  - $ce \rightarrow ab$  [2, 2/2];  $ae \rightarrow bc$  [2, 2/3];
  - $be \rightarrow ac$  [2, 2/4];  $ac \rightarrow be$  [2, 2/4];
  - $bc \rightarrow ae$  [2, 2/3];  $ab \rightarrow ce$  [2, 2/4];

Silne reguły asocjacyjne:

- $ce \rightarrow ab$  [2, 2/2];
- $ae \rightarrow bc$  [2, 2/3];
- $be \rightarrow ac$  [2, 2/4];
- $ac \rightarrow be$  [2, 2/4];
- $bc \rightarrow ae$  [2, 2/3];
- $ab \rightarrow ce$  [2, 2/4];

32

32

**Przykład: Odkrywanie ARs**

Częste zbiory pozycji ( $\text{minSup} = 1$ ):  $\emptyset_8$   
 $a_6 b_5 c_4 e_4 f_4 h_3$   
 $ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 ce_2 cf_2 ch_2 ef_2$   
 $abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2$   
 $abce_2$

**Iteracja 3 ( $\text{minConf} = 60\%$ ,  $Z = abce$ ):**

- Następniki dla ARs, znalezione w iteracji 2:  $Y_2 = \{ab, ae, bc\}$ .
- Następniki reguł kandydujących:  $Y_3 = \{abe\}$ .
- Kandydujące reguły:
  - $c \rightarrow abe$  [2, 2/4]

Silne reguły asocjacyjne:

- Brak ARs w iteracji 3.

33

33

**Przykład: Odkryte ARs**

Częste zbiory pozycji ( $\text{minSup} = 1$ ):  $\emptyset_8$   
 $a_6 b_5 c_4 e_4 f_4 h_3$   
 $ab_4 ac_4 ae_3 af_3 ah_2 bc_3 be_4 bf_2 ce_2 cf_2 ch_2 ef_2$   
 $abc_3 abe_3 ace_2 acf_2 ach_2 bce_2 bef_2$   
 $abce_2$

Silne reguły asocjacyjne ( $\text{minConf} = 60\%$ ,  $Z = abce$ ):

- $bce \rightarrow a$  [2, 2/2];
- $ace \rightarrow b$  [2, 2/2];
- $abe \rightarrow c$  [2, 2/3];
- $abc \rightarrow e$  [2, 2/3];
- $ce \rightarrow ab$  [2, 2/2];
- $bc \rightarrow ae$  [2, 2/3];
- $ae \rightarrow bc$  [2, 2/3].

34

34

**Kwestie implementacyjne w algorytmie Apriori**

35

35

**Ważne operacje w algorytmach Apriori i AprioriRuleGen**

- Ważnymi, czasochłonnymi operacjami w *Apriori* są:
  - wyszukiwanie podzbiorów danej transakcji wśród zbiorów kandydujących o ustalonej długości,
  - sprawdzanie czy wszystkie podzbiory kandydata, krótsze od niego o jedną pozycję, są częste.
- Ważną, czasochłonną operacją w *AprioriRuleGen* jest wyszukiwanie podzbiorów częstych o zadanej długości (poprzedników reguł kandydujących) danego zbioru częstego (bazy reguły kandydującej) w celu poznania ich wsparć.
- W powyższych przypadkach wyszukiwane są ustalonej długości podzbiory danego zbioru (odpowiednio: transakcji, kandydata lub poprzednika kandydującej reguły).

36

36

**Użycie drzewa mieszającego**

- Drzewo mieszające (ang. *hash tree*) jest używane w celu wydajnego wyszukiwania i pozycyjnych podzbiorów danego zbioru.
- W szczególności, wszystkie i pozycyjne zbiory kandydujące są przechowywane w drzewie mieszającym.

37

**Przykład: Drzewo mieszające z kandydującymi 3-zbiorami**

Założenia:

- $h(x) = x \bmod 3$
- pojemność liścia – 2 zbiory

kandydat	reprezentacja
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

38

**Przykład: Tworzenie drzewa mieszającego z kandydującymi 3-zbiorami ...**

Założenia:

- $h(x) = x \bmod 3$
- pojemność liścia – 2 zbiory

kandydat	reprezentacja
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

39

**Przykład: Tworzenie drzewa mieszającego z kandydującymi 3-zbiorami ...**

Założenia:

- $h(x) = x \bmod 3$
- pojemność liścia – 2 zbiory

kandydat	reprezentacja
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

40

**Przykład: Tworzenie drzewa mieszającego z kandydującymi 3-zbiorami ...**

Założenia:

- $h(x) = x \bmod 3$
- pojemność liścia – 2 zbiory

kandydat	reprezentacja
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

41

**Przykład: Tworzenie drzewa mieszającego z kandydującymi 3-zbiorami ...**

Założenia:

- $h(x) = x \bmod 3$
- pojemność liścia – 2 zbiory

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

42

**Przykład: Tworzenie drzewa mieszającego z kandydującymi 3-zbiorami ...**

kandydat	reprezentacja
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Założenia:  
 •  $h(x) = x \bmod 3$   
 • pojemność liścia – 2 zbiory

43

43

**Przykład: Tworzenie drzewa mieszającego z kandydującymi 3-zbiorami ...**

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Założenia:  
 •  $h(x) = x \bmod 3$   
 • pojemność liścia – 2 zbiory

44

44

**Przykład: Tworzenie drzewa mieszającego z kandydującymi 3-zbiorami ...**

kandydat	reprezentacja
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Założenia:  
 •  $h(x) = x \bmod 3$   
 • pojemność liścia – 2 zbiory

45

45

**Przykład: Tworzenie drzewa mieszającego z kandydującymi 3-zbiorami ...**

candidate	coded candidate
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Założenia:  
 •  $h(x) = x \bmod 3$   
 • pojemność liścia – 2 zbiory

46

46

**Przykład: Tworzenie drzewa mieszającego z kandydującymi 3-zbiorami ...**

kandydat	reprezentacja
abc	123
abe	125
abf	126
ace	135
acf	136
ach	138
aef	156
bce	235
bcf	236
bef	256
beh	258
cef	356

Założenia:  
 •  $h(x) = x \bmod 3$   
 • pojemność liścia – 2 zbiory

47

47

**Przykład: Wyszukiwanie podzbiorów w drzewie mieszającym z kandydującymi 3-zbiorami...**

kandydat	reprezentacja
abc	123
abe	125
abf	126
...	...
transakcja	reprezentacja
...	...
bcef	2356
...	...

Założenia:  
 •  $h(x) = x \bmod 3$

• Znaleziono 4 podzbiory transakcji bcef (reprezentowanej przez: 2356).

48

48



**Przykład: Wyszukiwanie podzbiorów w drzewie mieszającym z kandydującymi 3-zbiorami**

kandydat	reprezentacja
abc	123
abe	125
abf	126
...	...

Assumption:  
•  $h(x) = x \bmod 3$

transakcja	reprezentacja
...	...
acde	1345
...	...

Znaleziono 1 podzbiór transakcji acde (reprezentowanej przez: 1345).

49

**Odkrywanie zbiorów częstych za pomocą algorytmu Eclat**

50

**Eclat: wyznaczanie tidlist...**

Id	Transaction
1	{abc}
2	{abc}
3	{abc}
4	{ab}
5	{bcd}

- Tidlisty (listy identyfikatorów transakcji zbiorów) kandydatów o długości 1:
  - $t(\{a\}) = \{1,2,3,4\}$ , stąd  $sup(\{a\}) = 4$
  - $t(\{b\}) = \{1,2,3,4,5\}$ , stąd  $sup(\{b\}) = 5$
  - $t(\{c\}) = \{1,2,3,5\}$ , stąd  $sup(\{c\}) = 4$
  - ...
- Tidlisty kandydatów o długości 2:
  - $t(\{ab\}) = t(\{a\}) \cap t(\{b\}) = \{1,2,3,4\} \cap \{1,2,3,4,5\} = \{1,2,3,4\}$ , stąd  $sup(\{ab\}) = 4$
  - $t(\{ac\}) = t(\{a\}) \cap t(\{c\}) = \{1,2,3,4\} \cap \{1,2,3,5\} = \{1,2,3\}$ , stąd  $sup(\{ac\}) = 3$
  - ...

$t(X \cup Y) = t(X) \cap t(Y)$   
 $sup(X \cup Y) = |t(X \cup Y)|$

51

**Eclat: wyznaczanie wyznaczanie tidlist**

Id	Transaction
1	{abc}
2	{abc}
3	{abc}
4	{ab}
5	{bc}

- Tidlisty kandydatów o długości 2:
  - $t(\{ab\}) = \{1,2,3,4\}$ ,  $sup(\{ab\}) = 4$
  - $t(\{ac\}) = \{1,2,3\}$ ,  $sup(\{ac\}) = 3$
  - ...
- Tidlisty kandydatów o długości 3:
  - $t(\{abc\}) = t(\{ab\}) \cap t(\{ac\}) = \{1,2,3,4\} \cap \{1,2,3\} = \{1,2,3\}$ , stąd  $sup(\{abc\}) = 3$
  - ...

$t(X \cup Y) = t(X) \cap t(Y)$   
 $sup(X \cup Y) = |t(X \cup Y)|$

52

**Eclat: wyznaczanie zbiorów częstych...**

Id	Transaction
1	{abc}
2	{abc}
3	{abc}
4	{ab}
5	{bcd}

Założenie:  
•  $minSup = 2$

$t(X \cup Y) = t(X) \cap t(Y)$   
 $sup(X \cup Y) = |t(X \cup Y)|$

53

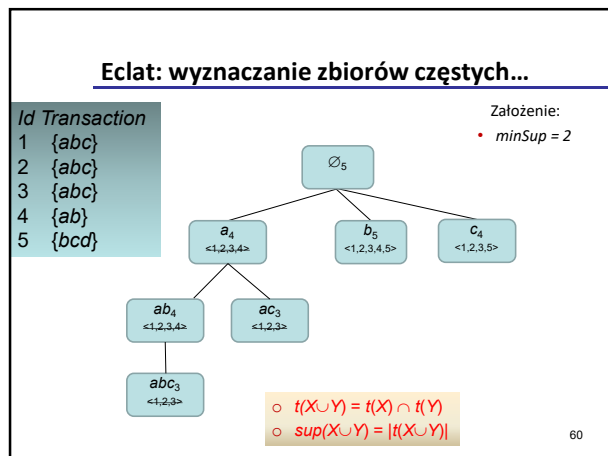
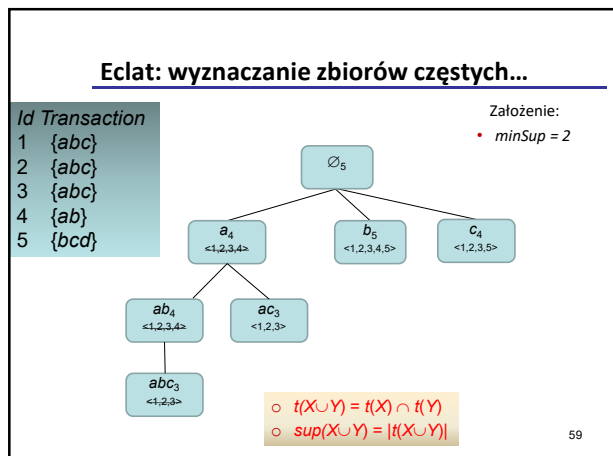
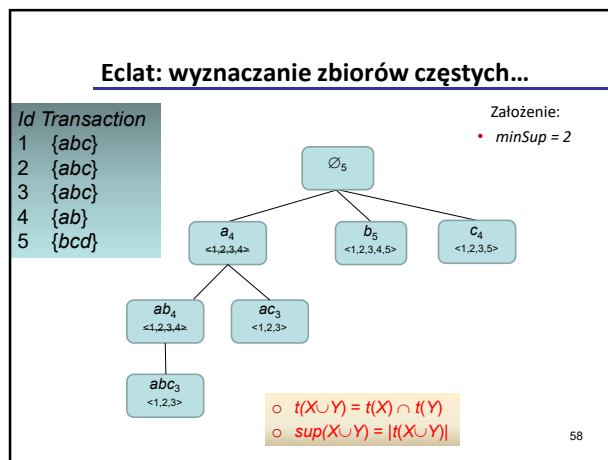
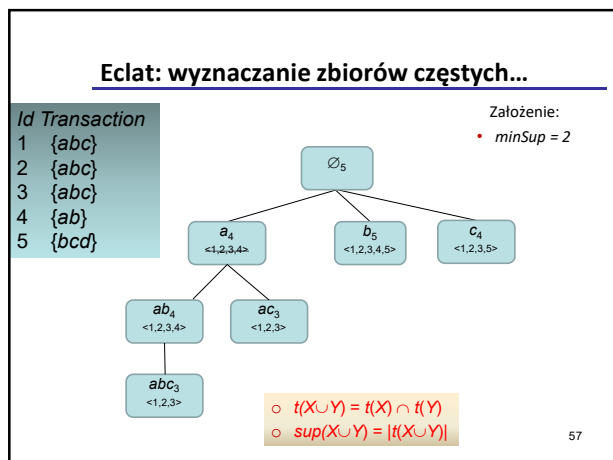
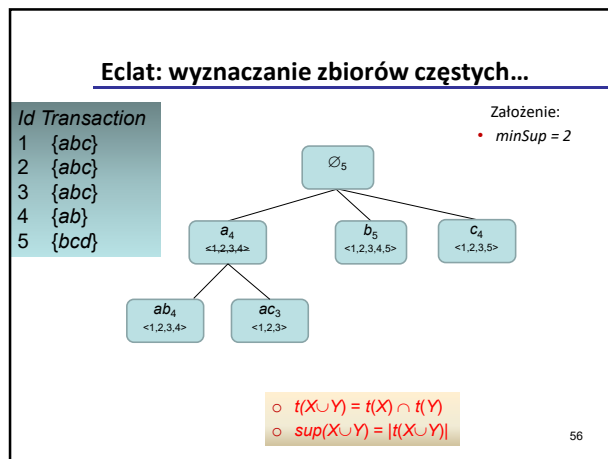
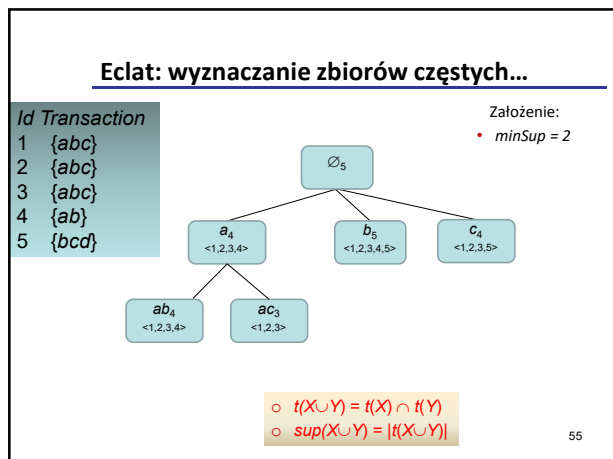
**Eclat: wyznaczanie zbiorów częstych...**

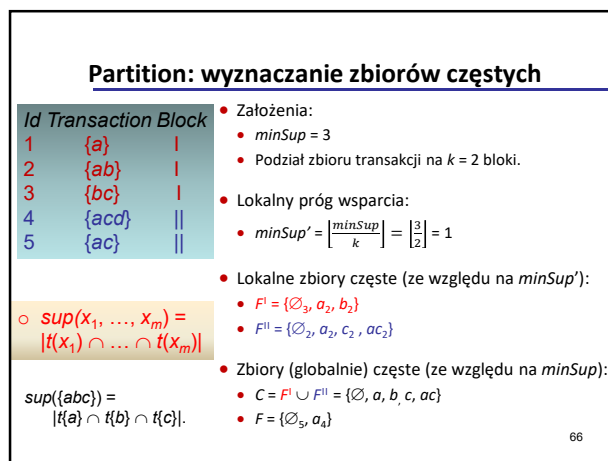
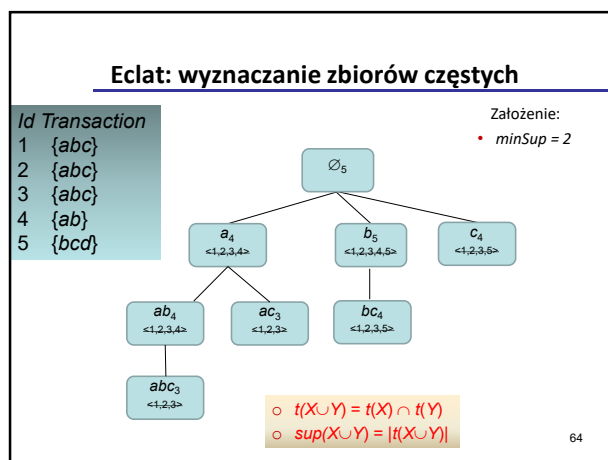
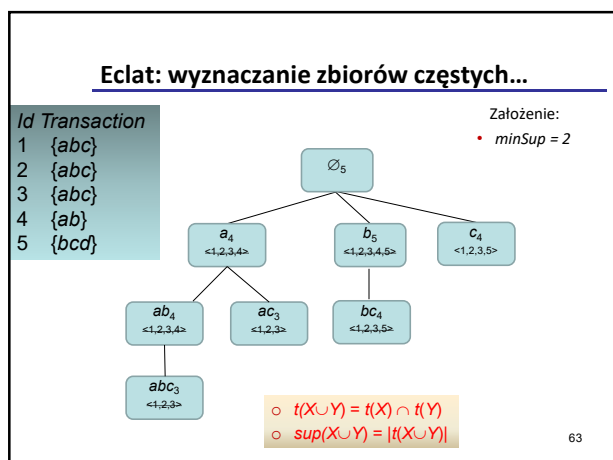
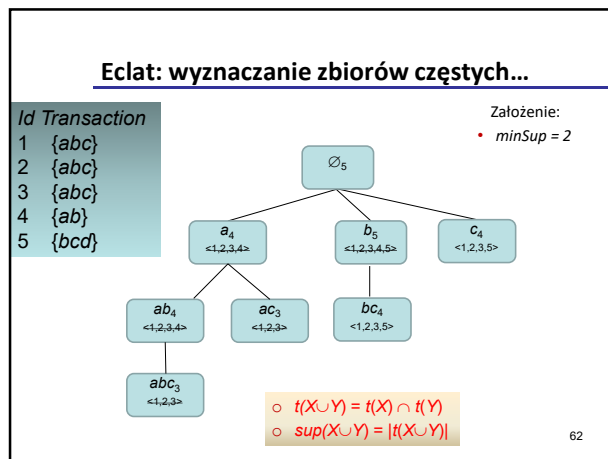
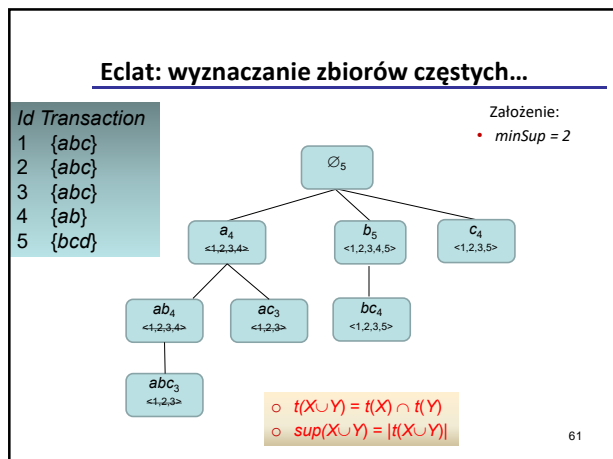
Id	Transaction
1	{abc}
2	{abc}
3	{abc}
4	{ab}
5	{bcd}

Założenie:  
•  $minSup = 2$

$t(X \cup Y) = t(X) \cap t(Y)$   
 $sup(X \cup Y) = |t(X \cup Y)|$

54





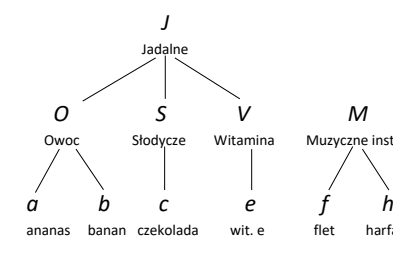


**Odkrywanie ARs z danych nie (tylko) transakcyjnych**

67

**Dane transakcyjne + taksonomie**

Tid	Items
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

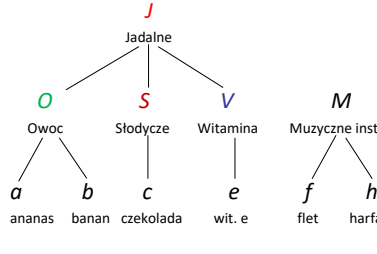


68

68

**Dane transakcyjne z uwzględnieniem taksonomii**

Tid	Items
1	abce <sup>OSVJ</sup>
2	abcef <sup>OSVMJ</sup>
3	abch <sup>OSMJ</sup>
4	abe <sup>OVJ</sup>
5	acfh <sup>OSMJ</sup>
6	bef <sup>OVMJ</sup>
7	h <sup>M</sup>
8	af <sup>OMJ</sup>

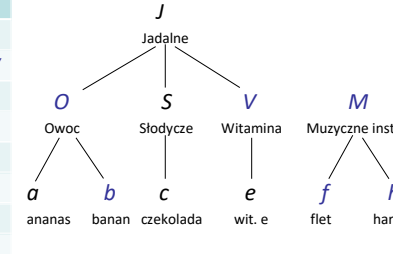


69

69

**Dane transakcyjne z uwzględnieniem taksonomii i nałożonych warunków**

Tid	Items
1	abce <sup>OV</sup>
2	abcef <sup>OMV</sup>
3	abch <sup>OM</sup>
4	abe <sup>V</sup>
5	acfh <sup>OM</sup>
6	bef <sup>OMV</sup>
7	h <sup>M</sup>
8	af <sup>OM</sup>



70

70

**Dane transakcyjne + pozycje zanegowane...**

Tid	Items
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

6 poz. pozycji {abcefh}

6 neg. pozycji {abcefh}

Tid	Items
1	abce <sup>fh</sup>
2	abcef <sup>fh</sup>
3	abch <sup>ef</sup>
4	abe <sup>cfh</sup>
5	acfh <sup>be</sup>
6	bef <sup>ach</sup>
7	h <sup>abcef</sup>
8	af <sup>bceh</sup>

71

71

**Dane transakcyjne + pozycje zanegowane...**

Tid	Items
1	abce
2	abcef
3	abch
4	abe
5	acfh
6	bef
7	h
8	af

6 poz. pozycji {abcefh}

6 neg. pozycji {abcefh}

$sup(\{befh\}) = 2$

$conf(\{bf\} \rightarrow \{eh\}) = 2/3$

$conf(\{bh\} \rightarrow \{ef\}) = 2/4$

Tid	Items
1	abce <sup>fh</sup>
2	abcef <sup>fh</sup>
3	abch <sup>ef</sup>
4	abe <sup>cfh</sup>
5	acfh <sup>be</sup>
6	bef <sup>ach</sup>
7	h <sup>abcef</sup>
8	af <sup>bceh</sup>

72

72

**Dane transakcyjne + pozycje zanegowane...**

Tid	Items
1	abcefh
2	abcefh
3	abchef
4	abecfh
5	acfhbe
6	befach
7	habcef
8	afbceh

→

Tid	Items
1	1 2 3 4 11 12
2	1 2 3 4 5 12
3	1 2 3 6 11 12
4	1 2 4 10 11 12
5	1 3 5 6 8 10
6	2 4 5 7 9 12
7	6 7 8 9 10 11
8	1 5 8 9 10 12

Item	a	b	c	e	f	h	a	b	c	e	f	h
item id	1	2	3	4	5	6	7	8	9	10	11	12

73

**Dane transakcyjne + pozycje zanegowane**

Niech  $n$  będzie liczbą wszystkich pozycji.

Maksymalna liczba zbiorów częstych bez negacji wynosi  $2^n$ .

Maksymalna liczba zbiorów częstych dopuszczających negację wynosi  $3^n$ .

n	maks. # zb. częstych bez negacji	maks. # zb. częstych dopuszczających negację	różnica rzędów wielkości
6	64	729	1
10	1024	59049	2
50	1.13E+15	7.17898E+23	8
100	1.27E+30	5.15378E+47	17
500	3.3E+150	3.636E+238	88

74

**Dane relacyjne → Dane transakcyjne...**

Wzrost	Kolor	Ocena
wysoki	zielony	5
niski	czarny	4
niski	zielony	4

→

Tid	Items
1	1 3 6
2	2 4 5
3	2 3 5

Item	(W=wys.)	(W=nis.)	(K=ziel.)	(K=czarny)	(O=4)	(O=5)
item id	1	2	3	4	5	6
atrybut	1	1	2	2	3	3

75

**Dane relacyjne → Dane transakcyjne**

Wzrost	Kolor	Ocena
wysoki	zielony	5
niski	czarny	4
niski	zielony	4

→

Tid	Items
1	1 3 6
2	2 4 5
3	2 3 5

Item	(W=w)	(W=n)	(K=z)	(K=c)	(O=4)	(O=5)
item id	1	2	3	4	5	6
atrybut	1	1	2	2	3	3

{2} → {5} [2, 2/2]. Stąd: (W=n) → (O=4) [2, 100%].

{3} → {5} [1, 1/2]. Stąd: (K=z) → (O=4) [1, 50%].

{2,3} → {5} [1, 1/1]. Stąd: (W=n) ∧ (K=z) → (O=4) [1, 100%].

76

**Literatura...**


- Rakesh Agrawal, Ramakrishnan Srikant: Fast Algorithms for Mining Association Rules in Large Databases. [VLDB 1994](#): 487-499
- Jiawei Han, Micheline Kamber, Jian Pei: Data Mining: Concept and Techniques, The Morgan Kaufmann Series in Data Management Systems, 2011
- Marzena Kryszkiewicz: Reasoning about Frequent Patterns with Negation. [Encyclopedia of Data Warehousing and Mining 2009](#): 1667-1674
- Tadeusz Morzy, Eksploracja danych: Metody i algorytmy, Wydawnictwo Naukowe PWN, 2013

77

**Literatura**

- Ashok Savasere, Edward Omiecinski, Shamkant B. Navathe: An Efficient Algorithm for Mining Association Rules in Large Databases. [VLDB 1995](#): 432-444
- Mohammed Javeed Zaki: Scalable Algorithms for Association Mining. IEEE Trans. Knowl. Data Eng. 12(3): 372-390 (2000)
- Frequent Itemset Mining Implementations Repository: <http://fimi.ua.ac.be/src/>

78



### Literatura dodatkowa - odkrywanie zbiorów częstych

- Ferenc Bodon: A fast APRIORI implementation. FIMI 2003
- E. Hikmawati, N.U. Maulidevi, K. Surendro: Minimum threshold determination method based on dataset characteristics in association rule mining. *J Big Data* 8, 146 (2021). <https://doi.org/10.1186/s40537-021-00538-3>
- CSK Selvi, A. Tamilarasi: An automated association rule mining technique with cumulative support thresholds. *Int J Open Probl Comput Math.* 2009;2:12
- Laszlo Szathmary, Superset Function with a Trie in PhD Thesis: Symbolic Data Mining Methods with the Coron Platform (<https://tel.archives-ouvertes.fr/tel-01754284/file/szathmary-thesis06.pdf>)

79



### Literatura dodatkowa - odkrywanie zbiorów częstych

- Chen Wang, Mingsheng Hong, Jian Pei, Haofeng Zhou, Wei Wang, Baile Shi: Efficient Pattern-Growth Methods for Frequent Tree Pattern Mining. *PAKDD* 2004: 441-451
- Marek Wojciechowski, Maciej Zakrzewicz: Dataset Filtering Techniques in Constraint-Based Frequent Pattern Mining. *Pattern Detection and Discovery* 2002: 77-91
- Oracle Apriori, <https://www.ibm.com/docs/pl/spss-modeler/SaaS?topic=mining-oracle-apriori>

80