Session-2

i22-2041

i22-1855

(Tashfeen) (Laiba)

Each basket is subset of items ↑

Date:

Association rules (market basket analysis)

↓ generates rules from counts

Support (S) => %.age of transactions

$P(A \cap B)$ } measures frequency of association

Confidence (C) →

$$\text{conf}(I \to j) = \frac{\text{support}(I \cup j)}{\text{support}(I)} \quad \cancel{\frac{P(A \cap B)}{P(A)}}$$ } strength of association

Parameters :-

i) Finding all items that appears frequently } min support count

ii) Find strong associations amoung frequent items } Confiden

=> Frequent itemsets (like treshold = 3)

⇓

Now, we see that in each basket the items which are repeated atleast 3 times.

=> Confidence

$$\text{conf}(I \to j) = \frac{\text{Support}(I \cup j)}{\text{Support}(I)}$$

=> Interest $(I \to j)$ = Conf $(I \to j)$ - Pr$[j]$

↓

Interesting rules are those with high interest value (> 0.5)

Date:

Example:

$B_1 = \{m, c, b\}$ $B_5 = \{m, p, j\}$

$B_2 = \{m, b\}$ $B_6 = \{c, j\}$

$B_3 = \{m, p, b\}$ $B_7 = \{m, c, b, j\}$

$B_4 = \{c, b, j\}$ $B_8 = \{b, c\}$

=> Association rule:- $\{m, b\} \rightarrow c$

$$\text{Confidence} = \frac{2}{4} = 0.5$$

$$\text{Interest} = \left(\left|0.5 - \frac{5}{8}\right|\right) = \frac{1}{8}$$

=>
- Support threshold = 3, Confidence = 0.75
- Frequent itemsets:-

  $\{b, m\}$, $\{b, c\}$, $\{c, m\}$, $\{c, j\}$, $\{m, c, b\}$

- Generate rules:-

$b \rightarrow m : c = 4/6$ $b \rightarrow c : c = 5/6$ $c \rightarrow m$

....

Date:

=> Compressing the output (Maximal / Closed)

i)

Items	Support	Maximal (S=3)	Closed
A	4		No
B	5		Yes
C	3		No
AB	4		Yes
AC	2		No
BC	3		Yes
ABC	2		Yes.

- 'A' Ko items wali list ma dakhna hy Kis Kay set me 'A' hy (e.g., AB, AC, ABC)
- Ab 'A' Ki support 4 hy. Avr Ab A(support) > AB(support), AC(support), ABC(support)

Closed => A (support) > Superset (support)

Maximal => • A (support) ≥ Threshold
• A (Supersets -> Support < Threshold)

Frequent pairs => no. of sets gets slow with size.

Date:

=> Naive Algo → fails if $(\text{items})^2$ · exceed main memory

=> Couting pairs in memory

- Approach 1:- Count pairs using matrix (uses 4 bytes per pair)

- Approach 2:- Table of triplets (12 bytes for pair)

=> A- Priori Algo (monotonicity)

Example:-

· Support = 2

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st Scan →

Itemset	Supp
A	2
B	3
C	3
D	1
E	3

L1 →

Itemset	Sup
A	2
B	3
C	3
E	3

C2

Itemset
A, B
A, C
A, E
B, C
B, E
C, E

← 2nd Scan

Itemset	Supp
A, B	1
A, C	2
A, E	1
B, C	2
B, E	3
C, E	2

←

Itemset	Supp
A, C	2
B, C	2
B, E	3
C, E	2

Date:

Item set	support
A, C, B	1
A, B, E	1
A, C, E	1
B, C, E	2

→

Itemset	Supp.
B, C, E	2

=> PCY (Park- Chen- Yu) Algo

s= 3

1. Items = [milk , coke , cookies , pepsi , ~~[illegible]~~ , juice]

milk = 1 , cookies = 3 , juice = 5

coke = 2 , pepsi = 4 ,

$B_1 = \{1,2,3\}$

$B_2 = \{1,4,5\}$

$B_3 = \{1,3\}$

$B_4 = \{2,5\}$

$B_5 = \{1,3,4\}$

$B_6 = \{1,2,3,5\}$

$B_7 = \{2,3,5\}$

$B_8 = \{2,3\}$

①

Items	frequency
1	5
2	5
3	6
4	2
5	4

② Remove elements having frequency less or equal than 2

Candidate set = { 1, 2, 3, 4, 5 }

(3) Map all candidates sets in pairs and calculate frequency (Sampling)

b1 = (1,2), (1,3), (2,3) = (2,4,4)

b2 = (1,4), (1,5), (4,5) = (2,2,1)

b3 = (1,3) = (4)

b4 = (2,5) = (3)

b5 = (1,3), (1,4), (3,4) = (4,2,1)

b6 : (1,2), (1,3) (1,5), (2,3), (2,5), (3,5) = (2,4,2,4,3,2)

b7 (2,3), (2,5), (3,5) = (4,3,2)

b8 (2,3) = (4)

Candidate pairs = (1,3) (2,3) (2,5)

(4) Apply hash function $h(i,j) = (i+j) \% 5 = K$

(1,3) = 4 % 5 = 4

(2,3) = 5 % 5 = 0

(2,5) = 7 % 5 = 2

Bucket no	pair	high freq	Candidate set
0	(2,3)	4	(2,3)
2	(2,5)	3	(2,5)
4	(1,3)	4	(1,3)

Candidate pair = 1, 2, 3, 5