SE ZG522: Big Data Systems

06 Incremental Data Mining

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

March 05, 2022 ONLINE (WILP @ BITS-Pilani Jan-May 2022)

http://ktiwari.in/bds

Association Rules

$t_1 = (1,3,4)$, $t_2 = (2,3,5)$, $t_3 = (1,2,3,5)$, $t_4 = (2,5)$, $t_5 = (1,3,5)$

Symb	Sup
{1}	3 ✓
{2}	3 ✓
{3}	4 ✓
{4}	1
{5}	4 ✓

Symb	Sup
{1,2}	1
{1,3}	3 ✓
{1,5}	2 ✓
{2,3}	2 ✓
{2,5}	3 ✓
{3,5}	3 ✓

Symb	Sup
{1,2,3}	1
{1,2,5}	1
{1,3,5}	2 ✓
{2,3,5}	2 ✓

Symb	Sup
{1,2,3,5}	1

An Example

Find **support** and **confidence** for $X \Rightarrow Y$ in following database

(Z)
(Z)
(Z)
(X,Y)
(X,Y)
(X,Y)
(X,Z)
(X,Z)
(Z)
(Z)

- **support** $= P(X \cup Y)$ $3/10$
- **confidence** $= P(Y|X)$ $3/5$

Association Rules

Association rules tries to do linked analysis.

Example: Whether same products are selling together?

- $I = \{i_1, i_2, i_3, ...i_m\}$, $T = \{t_1, t_2, t_3, ..., t_n\}$ and $t_i \subseteq I$
- Minimum support count should be maintained
- Can you see: Subset of frequent items is also frequent
- Apriori analysis [1]

Let's do it:

$t_1 = (1,3,4)$, $t_2 = (2,3,5)$, $t_3 = (1,2,3,5)$, $t_4 = (2,5)$, $t_5 = (1,3,5)$ and minimum support count be 2

[1] Mining association rules between sets of items in large databases, *R Agrawal, T Imielinski, and A Swami*, SIGMOD, 22(2), pp 207–216, ACM-1993

Association Rule Mining

Mathematical model of Association Rule Mining

- Let $I = \{i_1, i_2, ..., i_m\}$ be set of items
- Let $T = \{t_1, t_2, ..., t_n\}$ be set of transactions where $t_i \subseteq I$
- t_i is said to contain $X \subseteq I$ if $X \subseteq t_i$

An association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$ and $X \cap Y = \phi$

- An association rule $X \Rightarrow Y$ has a **support** s in set T if s% of the transactions in T contains $X \cup Y$

$$support(X \Rightarrow Y) = P(X \cup Y)$$

- The association rule $X \Rightarrow Y$ holds in the transaction set T with **confidence** c if c% of the transactions in T that contain X also contain Y

$$confidence(X \Rightarrow Y) = P(Y|X)$$

Association Rule Mining (contd..)

For a given *support* and *confidence* the problem of mining association rules is to find out all the association rules that have confidence and support greater than the corresponding thresholds.

It is a two-step process

1. *Find all frequent item sets:* $\{X : support(X) \geq S_{min}\}$
2. *Generate association rules from the frequent item set:* For any pair of frequent item set W and X satisfying $X \subset W$, of $support(X)/support(W) \geq C_{min}$, then $X \Rightarrow Y$ is a valid rule where $Y = W - X$.

- Second part is straight forward
- Most of the research interest lies in solving the first part

Prior work includes *Apriori*, *DHP*, *partition based*, *TreeProjection*, FP-Tree, and constraint-based ones.

Overview of Apriori

- Uses prior knowledge of k-item set to explore (k+1)-item set in a levelwise process.
- The set of frequent 1-item sets L_1 is initially found
- L_1 is then used by performing join and prune actions to form the set of candidate 2-items sets C_2
- In next data scan, the set of frequent 2-item sets L_2 are identified
- The whole process continues iteratively until there is no more candidate item sets

Example:

Consider $I = \{A, B, C, D, E, F\}$ and transaction $T = \{\ t_1 = (A, B, C)$, $t_2 = (A, F)$, $t_3 = (A, B, C, E)$, $t_4 = (A, B, D, F)$, $t_5 = (C, F)$, $t_6 = (A, B, C)$, $t_7 = (A, B, C, E)$, $t_8 = (C, D, E)$, $t_9 = (B, D, E)$, $\}$ and the minimum support be greater then 3.

Apriori at work

Consider transactions T

T_1	(A,B,C)
T_2	(A,F)
T_3	(A,B,C,E)
T_4	(A,B,D,F)
T_5	(C,F)
T_6	(A,B,C)
T_7	(A,B,C,E)
T_8	(C,D,E)
T_9	(B,D,E)

Scan T → C_1

Item	Supp
{A}	6
{B}	6
{C}	6
{D}	3
{E}	4
{F}	3

→ L_1

Item	Supp
{A}	6
{B}	6
{C}	6
{E}	4

C_2

Item
{A,B}
{A,C}
{A,E}
{B,C}
{B,E}
{C,E}

Scan T → C_2

Item	Supp
{A,B}	5
{A,C}	4
{A,E}	2
{B,C}	4
{B,E}	3
{C,E}	3

→ L_2

Item	Supp
{A,B}	5
{A,C}	4
{B,C}	4

C_3

Item
{A,B,C}

Scan T → C_3

Item	Supp
{A,B,C}	4

→ L_3

Item	Supp
{A,B,C}	4

Generate Association Rules

If $X \subset W$ & $support(X)/support(W) \geq C_{min}$, then $X \Rightarrow W - X$

Transactions

T_1	(A,B,C)
T_2	(A,F)
T_3	(A,B,C,E)
T_4	(A,B,D,F)
T_5	(C,F)
T_6	(A,B,C)
T_7	(A,B,C,E)
T_8	(C,D,E)
T_9	(B,D,E)

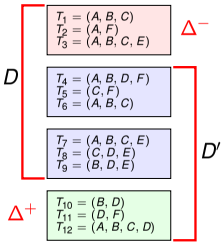
- Our frequent item contains
  - $\{A\}_6$ $\{B\}_6$ $\{C\}_6$ $\{E\}_4$ $\{A,B\}_5$ $\{A,C\}_4$ $\{B,C\}_4$ $\{A,B,C\}_4$
- Possibilities are
  $A \Rightarrow B$, $B \Rightarrow A$, $A \Rightarrow C$, $C \Rightarrow A$, , $B \Rightarrow C$, $C \Rightarrow B$, $A \Rightarrow \{B, C\}$, $B \Rightarrow \{A, C\}$, $C \Rightarrow \{B, A\}$, $\{A, B\} \Rightarrow C$, $\{A, C\} \Rightarrow B$, $\{B, C\} \Rightarrow A$
- Let's take $C_{min} = 1.22$
- Association rules that qualifies as valid rule are shown green
  $A \Rightarrow B$, $B \Rightarrow A$, $A \Rightarrow C$, $C \Rightarrow A$, $B \Rightarrow C$, $C \Rightarrow B$, $A \Rightarrow \{B, C\}$, $B \Rightarrow \{A, C\}$, $C \Rightarrow \{B, A\}$, $\{A, B\} \Rightarrow C$, $\{A, C\} \Rightarrow B$, $\{B, C\} \Rightarrow A$

Incremental Association Rule Mining

- Real databases are generally dynamic (not static)
- New transactions are generated and old transactions may be obsolete over time

Premitives:

- Let D be the initial database
- $\triangle^-$ portion of the database becomes obsolete
- Therefore, the database is left with $D^- = D - \triangle^-$
- $\triangle^+$ more transactions are added
- The database becomes $D' = D^- + \triangle^+ = D - \triangle^- + \triangle^+$



Incremental update can avoid redoing mining on updated database

Incremental Association Rule Mining (contd..)

1. The update problem can be reduced to finding the new set of frequent item sets. After that, the new association rules can be computed from the new frequent item sets.
2. An old frequent item set has the potential to become infrequent in the updated database.
3. Similarly, an old infrequent item set could become frequent in the new database.
4. In order to find the new frequent item sets "exactly", all the records in the updated database, including those from the original database, have to be checked against every candidate set.

We would evaluate two algorithms *viz*. FUP and FUP2

Algorithm FUP

- FUP stands for **F**ast **UP**date. [2]
- Can handle insertions only
- Specifically, given the original database D and its corresponding frequent item set $L = \{L_1, L_2, ..., L_k\}$.
  - The goal is to reuse the information, to efficiently obtain $L' = \{L'_1, L'_2, ..., L'_k\}$ for new database $D' = D \cup \triangle^+$

By utilizing the definition of support and constraint of minimum support S_{min}, following is used by FUP.

- An original frequent item set $X \in L$, becomes infrequent in D' iff $support(X)_{D'} < S_{min}$
- An item set $X \notin L$, becomes frequent in D' iff $support(X)_{\triangle^+} \geq S_{min}$
- If a k-item set X whose $(k-1)$-subset(s) becomes infrequent, *i.e.*, the subset is in L_{k-1} but not in L'_{k-1}, then X must be infrequent in D'.

[2] Maintenance of discovered association rules in large databases: An incremental updating technique, *DW Cheung, and J Han, and V Ng, and CY Wong*, International conference on data engineering, pp 106–114, IEEE, 1996

FUP at work

Consider the database D and the related frequent set discovered with Apriori

$T_1 = (A, B, C)$
$T_2 = (A, F)$
$T_3 = (A, B, C, E)$
$T_4 = (A, B, D, F)$
$T_5 = (C, F)$
$T_6 = (A, B, C)$
$T_7 = (A, B, C, E)$
$T_8 = (C, D, E)$
$T_9 = (B, D, E)$

Item set	Support
{A}	6/9
{B}	6/9
{C}	6/9
{E}	4/9
{A B}	5/9
{A C}	4/9
{B C}	4/9
{A B C}	4/9

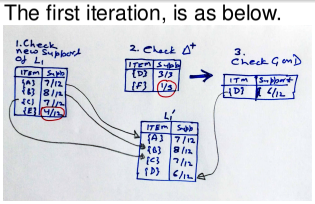
Consider the arrival of $\triangle^+$ more transactions

$T_1 = (A, B, C)$
$T_2 = (A, F)$
$T_3 = (A, B, C, E)$
$T_4 = (A, B, D, F)$
$T_5 = (C, F)$
$T_6 = (A, B, C)$
$T_7 = (A, B, C, E)$
$T_8 = (C, D, E)$
$T_9 = (B, D, E)$

$\triangle^+$

$T_{10} = (B, D)$
$T_{11} = (D, F)$
$T_{12} = (A, B, C, D)$

The first iteration, is as below.



FUP at work (contd...)

The second iteration, is as below.

1. Check new support of L_1

ITEM	Support
{A,B}	6/12
{A,C}	5/12
{B,C}	5/12

2. Check other itemsets in $\triangle^+$

ITEM	Support
{A,D}	1/3
{B,D}	2/3
{C,D}	1/3
{D,F}	1/3

3. Validate C_2 on database D

ITEM	Support
{B,D}	4/12

ITEM	Support
{A,B}	6/12
{A,C}	5/12
{B,C}	5/12

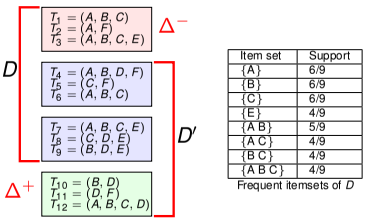
Similarly it is executed for next levels.

FUP_2

- FUP_2 [3] can work for both $\triangle^-$ and $\triangle^+$
- L_k from previous mining result is used for dividing candidate itemset C_k into two parts
  - $P_k = C_k \cap L_k$
  - $Q_k = C_k - P_k$
- Itemset that is frequent in $\triangle^-$, must be infrequent in D^-.
- Further if itemset in Q_k in infrequent in $\triangle^+$ then it is infrequent in D^-.
- This technique helps to effectively reduce number of candidate itemsets.

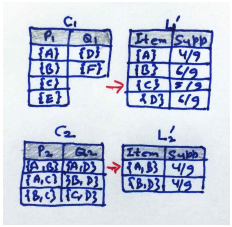
[3] A general incremental technique for maintaining discovered association rules, *DW Cheung, SD Lee, and B Kao*, Database Systems For Advanced Applications, pp: 185–194, World Scientific-1997

FUP_2 at work



Item set	Support
{A}	6/9
{B}	6/9
{C}	6/9
{E}	4/9
{A B}	5/9
{A C}	4/9
{B C}	4/9
{A B C}	4/9

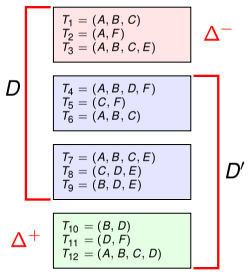
Frequent itemsets of D



- C_1 is set of all items. It is divided in P_i and Q_i
- Being frequent, support for all items in P_i is known. It could be updated using $\triangle^-$ and $\triangle^+$ only.
- $\text{Count}(\{A\})_{D'} = \text{Count}(\{A\})_D - \text{Count}(\{A\})_{\triangle^-} + \text{Count}(\{A\})_{\triangle^+}$
  $= 6 - 3 + 1 = 4$

FUP_2 at work

- In some cases only the scan of $\triangle^-$ and $\triangle^+$ is required.
- For example, $\text{Count}(\{F\})_{\triangle^+} - \text{Count}(\{F\})_{\triangle^-} = 0$ showing that support of {F} can not be improved.
- Consequently, fewer itemsets have to be further scanned
- An iteration finishes when all the itemsets in P_i and Q_i are verified, and new set of frequent itemsets L'_i is generated



Thank You!

Thank you very much for your attention!

Queries ?