Homework 2

Apriori Algorithm

Sagar Dhamija (50169364)

Varun Khandelwal (50168936)

Contents

Apriori Algorithm

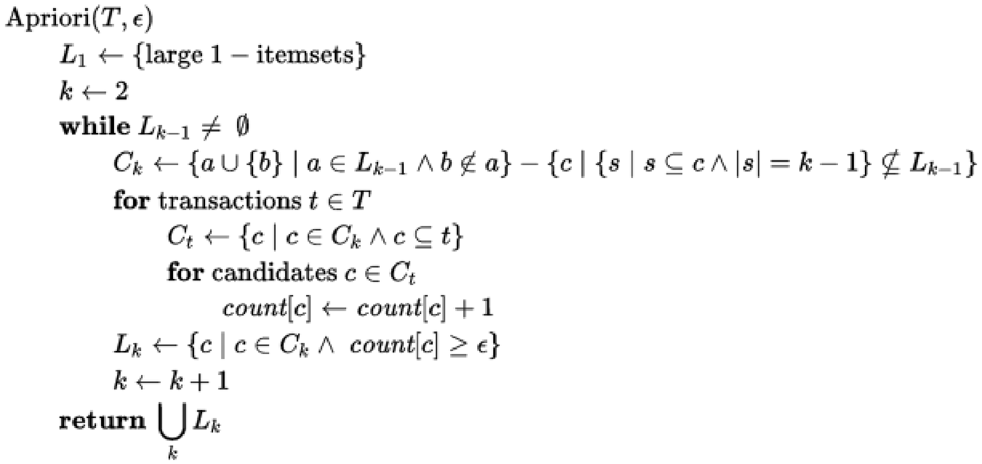
“Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

Apriori principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)$$

From the above mentioned equation, we can conclude that support of an item set never exceeds the support of its subsets. This is known as the anti-monotone property of support.

General Idea behind Apriori Algorithm:

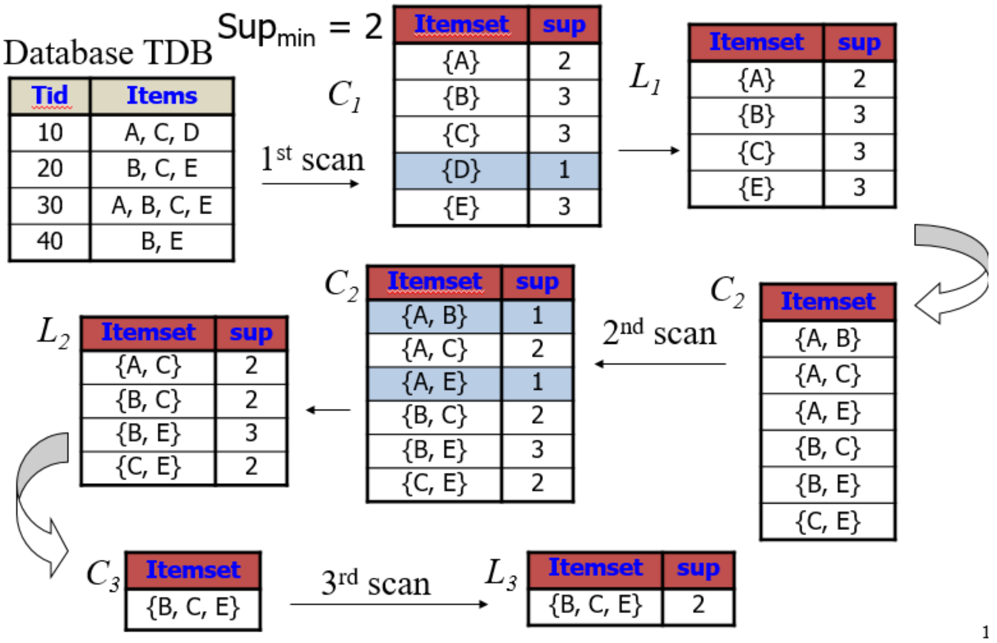


```
Apriori($T, \epsilon$)
    $L_1 \leftarrow \{\text{large } 1 - \text{itemsets}\}$
    $k \leftarrow 2$
    while $L_{k-1} \neq \emptyset$
        $C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \not\in a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k-1\} \nsubseteq L_{k-1}\}$
        for transactions $t \in T$
            $C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$
            for candidates $c \in C_t$
                $count[c] \leftarrow count[c] + 1$
        $L_k \leftarrow \{c \mid c \in C_k \wedge \ count[c] \geq \epsilon\}$
        $k \leftarrow k + 1$
    return $\bigcup_k L_k$
```

”

Examples depicting Apriori Algorithm:

Example 1:



Database TDB

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

$Sup_{min} = 2$

C_1 (1st scan)

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

L_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

C_2 (2nd scan)

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

L_3 (3rd scan)

Itemset	sup
{B, C, E}	2

Example 2:

“Consider the following database, where each row is a transaction and each cell is an individual item of the transaction:

alpha	beta	epsilon
alpha	beta	theta
alpha	beta	epsilon
alpha	beta	theta

The association rules that can be determined from this database are the following:

1. 100% of sets with alpha also contain beta
2. 50% of sets with alpha, beta also have epsilon
3. 50% of sets with alpha, beta also have theta”

Details of Apriori Implementation:

Given the fact that the size of the frequent item sets is never known therefore, we analyzed that the best data structure to use would be ArrayList. The code that we executed for implementing Apriori algorithm is broadly classified into 3 distinct easy to use and implement functions:

1) **background()**
   This function reads data from the file and also does preprocessing like converting Gene data to G1_UP and so on.
2) **set_length_1()**
   This function creates a frequent item set of size 1 only.
3) **set_length_n()**
   This function creates a list of frequent item sets of size 2 and above.

We have used the same algorithm as explained above in example 1.

Details of Association rule generation:

Association rule generation can be viewed broadly as follows:

1) **Rule set generation**
   This is the place where we have generated rules for a particular support and confidence value. We then store this rule set so that the end user can run formula (sample queries) on top of it.
2) **Formula parsing**
   This is where we went overboard and decided to accept the entire formula in a single string from the end user. Thus, the end user will just enter “BODY HAS ANY OF G1_UP AND HEAD HAS 1 OF G59_UP” and our code will automatically parse it and output the result.
3) **Formula execution**
   Our code is once again modular and we have created separate functions for:
   i. Template 2 queries
   ii. Queries that use the term “ANY”
   iii. Queries that work using “NUMBER” or “NONE”

Results:

Frequent itemsets for different supports

In order to maintain readability and clarity we have provided the results for different supports:

Support:30

Length of Item set	Item set Size
1	196
2	5340
3	5287
4	1518
5	438
6	88
7	11
8	1
9	0

Support:40

Length of Item set	**Item set Size**
1	167
2	753
3	149
4	7
5	1
6	0

Support:50

Length of Item set	**Item set Size**
1	109
2	63
3	2
4	0

Support:60

Length of Item set	Item set Size
1	34
2	2
3	0

Support:70

Length of Item set	Item set Size
1	7
2	0

Sample Query Output:

For the queries provided to us we have gone ahead summarized the results for better readability and understandability.

RULE HAS ANY OF G6_UP	Count:10
RULE HAS 1 OF G1_UP	Count:14
RULE HAS 1 OF (G1_UP, G10_DOWN)	Count:26
BODY HAS ANY OF G6_UP	Count:5
BODY HAS NONE OF G72_UP	Count:138
BODY HAS 1 OF (G1_UP, G10_DOWN)	Count:15
HEAD HAS ANY OF G6_UP	Count:5
HEAD HAS NONE OF (G1_UP, G6_UP)	Count:126
HEAD HAS 1 OF (G6_UP, G8_UP)	Count:6
RULE HAS 1 OF (G1_UP, G6_UP, G72_UP)	Count:48
RULE HAS ANY OF (G1_UP, G6_UP, G72_UP)	Count:50
SIZE OF RULE >= 3	Count:12
SIZE OF BODY >= 2	Count:6
SIZE OF HEAD >= 2	Count:6
BODY HAS ANY OF G1_UP AND HEAD HAS 1 OF G59_UP	Count:1
BODY HAS ANY OF G1_UP OR HEAD HAS 1 OF G6_UP	Count:12
BODY HAS 1 OF G1_UP OR HEAD HAS 2 OF G6_UP	Count:7
HEAD HAS 1 OF G1_UP AND BODY HAS 0 OF DISEASE	Count:7
HEAD HAS 1 OF DISEASE OR RULE HAS 1 OF (G72_UP, G96_DOWN)	Count:24
BODY HAS 1 of (G59_UP, G96_DOWN) AND SIZE OF RULE >= 3	Count:7

The actual results obtained from the code is shown below:

```
Enter Support
50
Enter Confidence
60

Support:50
Length of itemset:1
itemset size:109
Length of itemset:2
itemset size:63
Length of itemset:3
itemset size:2
Length of itemset:4
itemset size:0

Confidence:60
Length of Rule:2
Count:126
Length of Rule:3
```

```
Count:12

Enter formula
RULE HAS ANY OF G6_UP
Count:10

Enter formula
RULE HAS 1 OF G1_UP
Count:14

Enter formula
RULE HAS 1 OF (G1_UP, G10_DOWN)
Count:26

Enter formula
BODY HAS ANY OF G6_UP
Count:5

Enter formula
BODY HAS NONE OF G72_UP
Count:138

Enter formula
BODY HAS 1 OF (G1_UP, G10_DOWN)
Count:15

Enter formula
HEAD HAS ANY OF G6_UP
Count:5

Enter formula
HEAD HAS NONE OF (G1_UP, G6_UP)
Count:126

Enter formula
HEAD HAS 1 OF (G6_UP, G8_UP)
Count:6

Enter formula
RULE HAS 1 OF (G1_UP, G6_UP, G72_UP)
Count:48

Enter formula
RULE HAS ANY OF (G1_UP, G6_UP, G72_UP)
Count:50

Enter formula
SIZE OF RULE >= 3
Count:12

Enter formula
SIZE OF BODY >= 2
```

```
Count:6

Enter formula
SIZE OF HEAD >= 2
Count:6

Enter formula
BODY HAS ANY OF G1_UP AND HEAD HAS 1 OF G59_UP
Count:1

Enter formula
BODY HAS ANY OF G1_UP OR HEAD HAS 1 OF G6_UP
Count:12

Enter formula
BODY HAS 1 OF G1_UP OR HEAD HAS 2 OF G6_UP
Count:7

Enter formula
HEAD HAS 1 OF G1_UP AND BODY HAS 0 OF DISEASE
Count:7

Enter formula
HEAD HAS 1 OF DISEASE OR RULE HAS 1 OF (G72_UP, G96_DOWN)
Count:24

Enter formula
BODY HAS 1 of (G59_UP, G96_DOWN) AND SIZE OF RULE >= 3
Count:7
```

References

1. https://en.wikipedia.org/wiki/Apriori_algorithm
2. Lecture slides