INDIAN INSTITUTE OF TECHNOLOGY

KANPUR

Statistical & AI Techniques in Data Mining

MTH552A

FP GROWTH ALGORITHM AND ITS IMPROVEMENT BASED ON ADJACENCY TABLE

SUCHISMITA ROY (201440)

KOYEL PRAMANICK (201333)

Guided by: Dr. Amit Mitra

Department of Mathematics and Statistics

ACKNOWLEDGEMENT

Our journey of accomplishing this project really involves many ones to whom we are highly obliged. We would like to express our deepest appreciation to all those who have provided us the possibility to complete this project. We give a special gratitude to our respected instructor Dr.Amit Mitra, Department of Mathematics and Statistics, IIT KANPUR, whose contribution in stimulating suggestion, valuable guidance, constructive criticism and encouragement help us to coordinate our project.

We take the privilege to thank the authors and publishers of the various papers we have consulted. Also thanks to the various other free website from which we got help. At last we would like to thank our seniors and batch mates for their co-operation throughout the project. Without their guidance and supervision this project would not have been completed.

ABSTRACT

In this project, we have discussed three different methods of Association Rule Mining. We started from Apriori algorithm and illustrated using a hypothetical example. To overcome its drawback, we move towards FP Algorithm. FP Algorithm performs good in most situations. However, if the frequent itemsets are too many, then it is not that much effective. To deal with this situation, we have disussed a association rule mining technique based on adjacency table. At the end, we have illustrated these algorithms using a real life dataset and have tried to predict which characteristics increases the chance of heart attack.

Contents

1 Introduction

Data Mining can be defined as a technique for exploration and analysis large volume of data to discover meaningful patterns and rules that was previously unrevealed. The applications of Data Mining is expanding day by day. It is widely used in marketing, fraud detection, credit risk management, spam Email filtering, or even to discern the sentiment or opinion of users. In today's era of big data, we always try to find relationships among different variables. Association rule mining is one of such methods that finds interesting relationships, specifically associations or frequent patterns among the different itemset occurs in a transaction database. To find marketing strategies, we try to study the behaviour of customers regarding choice or preferences of products and services. To make profits, the provider needs to satisfy the customer. So an easy way is to arrange items in such a way so that the customer finds it convenient as well as tempted to buy. It maybe a supermarket or recommendation lists on E-Commerce platforms, it is a powerful key of profit. In this article, we will basically discuss about different algorithms used for Market Basket Analysis. Market Basket Analysis is a famous Association Rule Mining Technique that helps to identify which products a customer buys together.

In this article, we will basically talk about different algorithms of Market Basket Analysis. Today, researchers are spending time to improve association rule mining technique. There are many well known algorithms such as Apriori Algorithm, Eclat algorithm, OPUS search, FP-growth algorithm etc. *Apriori Algorithm* is one of the oldest algorithms of Association Rule Mining. It was proposed by Agarwal in 1993. It uses a breadth first search (3). Performance of Apriori Algorithm decreases as the complexity and the number of frequent itemsets increases. Then we moved to use *FP Algorithm.* FP stands for Frequent pattern, and it uses a recursive processing approach (3). In the paper (4), Yin et al. have argued that FP Algorithm demands for a lot of space and becomes inefficient in the case of Sparse data or dense dataset. And they have proposed a new method based on adjacency table. Throughout this report, we mainly want to compare these three techniques.

Our report is structured as follows. In the next section we will introduce these three algorithms. In section 3, we will make a theoretical comparison of these three methods and in the last section we will apply these algorithms to a real life dataset.

2 Different Algorithms

To proceed further, we first need to introduce some basic definitions.

- **Support Count:** Number of occurances of an itemset in the database T. It is denoted by $\sigma(\{itemset\})$
- **Support:** Fraction of transactions containing the itemset. Denoted by, $S(\{itemset\}) = \frac{\sigma\{itemset\}}{|T|}$.
- **Frequent Itemset:** An itemset which has support greater than or equal to a threshold, is called a frequent itemset.
- **Confidence:** Confidence is a measure used to find how often B appears in transactions containing A.

$$C(A \implies B) = \frac{S(A,B)}{S(A)} \times 100\%$$

Let us first introduce the algorithms.

2.1 Apriori Algorithm

It is one of the most popular algorithms. As it uses prior knowledge of frequent itemset properties to generate association rules, it is named as *Apriori Algorithm.* This algorithm is based on Anti-Monotonicity property which states that *Any subset of a frequent itemset is frequent.* There are two steps in Apriori Algorithm.

Step:1 Generate all frequent itemsets.

Step:2 Generate association rules using these frequent itemsets.

The first step can be divided into two steps-Join Step and Prune Step. Let l_k and c_k respectively denote the frequent itemsets and set of all candidates at level k.

1. Items in l_{k-1} are listed in an order.
2. **Self Joining:** Let two itemsets are in the following order, $\{p.item_1, p.item_2, \ldots, p.item_{k-1}\}$ and $\{q.item_1, q.item_2, \ldots, q.item_{k-1}\}$,
   where, $p.item_i = q.item_i$, for all $i = 1(1)k-2$
   and $p.item_{k-1} < q.item_{k-1}$ according to the order mentioned above, Then insert the itemset $p.item_1 p.item_2 \ldots, p.item_{k-1} q.itqmk - 1$ into c_k.
3. **Pruning of C_k Set** Let c be a specific item in C_k and s be any of $(k-1)$ subsets of C. If any of the subset s is not present in l_{k-1}, delete C from C_k.

Now we can move towards the next step, step 2, to develop association rule. Let L denotes the set of any frequent itemset.

1. Find all non-empty subsets F of L.
2. Each rule will be stated as $(F \implies \{L - F\})$ that satisfies the minimum confidence threshold level.

Let me now illustrate the Algorithm, using the following example. This is a hypothetical data that we have made that helps us to illustrate three methods.

Transactions	Items
T_1	I_1, I_3, I_5, I_8
T_2	I_2, I_4, I_6
T_3	I_1, I_4, I_5, I_7
T_4	I_1, I_4, I_8
T_5	I_5, I_7, I_8
T_6	I_1, I_3, I_5, I_7
T_7	I_3, I_8
T_8	I_1, I_7, I_8

The minimum support Count is 2. Let the minimum confidence is 60%.
Now we will continue our analysis as follows.

Candidate Sets (C_1)	Support Counts	Frequent Itemset (L_1)
I_1	5	I_1
I_2	1	
I_3	3	I_3
I_4	3	I_4
I_5	4	I_5
I_6	1	
I_7	4	I_7
I_8	5	I_8

Now we will move towards the second level.

Candidate Sets (C_2)	Support Counts	Frequent Itemset (L_2)
(I_1, I_3)	2	(I_1, I_3)
(I_1, I_4)	2	(I_1, I_4)
(I_1, I_5)	3	(I_1, I_5)
(I_1, I_7)	3	(I_1, I_7)
(I_1, I_8)	3	(I_1, I_8)
(I_3, I_4)	0	
(I_3, I_5)	2	(I_3, I_5)
(I_3, I_7)	1	
(I_3, I_8)	2	(I_3, I_8)
(I_4, I_5)	1	
(I_4, I_7)	1	
(I_4, I_8)	1	
(I_5, I_7)	3	(I_5, I_7)
(I_5, I_8)	2	(I_5, I_8)
(I_7, I_8)	2	(I_7, I_8)

Candidate Sets (C_3)	Support Counts	Frequent Itemset (L_3)
(I_1, I_3, I_4)	0	
(I_1, I_3, I_5)	2	(I_1, I_3, I_5)
(I_1, I_3, I_7)	1	
(I_1, I_3, I_8)	1	
(I_1, I_4, I_5)	1	
(I_1, I_4, I_7)	1	
(I_1, I_4, I_8)	1	
(I_1, I_5, I_7)	2	(I_1, I_5, I_7)
(I_1, I_5, I_8)	1	
(I_1, I_7, I_8)	1	
(I_3, I_5, I_8)	1	
(I_5, I_7, I_8)	1	

Now no element for next level. Thus the frequent itemsets are (I_1, I_3, I_7) and (I_1, I_5, I_7). Now it is time to move towards step 2.

Rule ($F \implies \{L - F\}$)	Confidence
$(I_1 \implies \{I_3, I_5\})$	$\frac{2}{5}$
$(I_3 \implies \{I_1, I_5\})$	$\frac{2}{3}$
$(I_5 \implies \{I_1, I_3\})$	$\frac{2}{4}$
$(\{I_1, I_3\} \implies I_5)$	$\frac{2}{2}$
$(\{I_1, I_5\} \implies I_3)$	$\frac{2}{3}$
$(\{I_3, I_5\} \implies I_1)$	$\frac{2}{2}$
$(I_1 \implies \{I_5, I_7\})$	$\frac{2}{5}$
$(I_5 \implies \{I_1, I_7\})$	$\frac{2}{4}$
$(I_7 \implies \{I_1, I_5\})$	$\frac{2}{4}$
$(\{I_1, I_5\} \implies I_7)$	$\frac{2}{3}$
$(\{I_1, I_7\} \implies I_5)$	$\frac{2}{3}$
$(\{I_5, I_7\} \implies I_1)$	$\frac{2}{3}$

Since, minimum confidence is 50% we get the following rules, $\{(I_3 \implies \{I_1, I_5\})$, $(\{I_1, I_3\} \implies I_5)$, $(\{I_1, I_5\} \implies I_3)$, $(\{I_3, I_5\} \implies I_1)$, $(I_5 \implies \{I_1, I_7\})$, $(\{I_1, I_5\} \implies I_7)$, $(\{I_1, I_7\} \implies I_5)$, $(\{I_5, I_7\} \implies I_1).\}$

2.2 FP Algorithm

The main drawback of Apriori Algorithm is it builds the candidate set at each step to generate the frequent itemset and for that it scans the entire data again and again. And if the dataset is large enough then it took a lot of time. To overcome this drawback we use *FP Algorithm*.
FP Algorithm or Frequent Pattern Growth Algorithm generates the frequent itemset without generating the candidate sets. It mainly comprises of two steps. In the first step, it builds a compact data structure known as *FP-Tree*; In the next step, it directly finds the frequent itemsets from the FP Tree. FP Tree was proposed by Han.(com) The advantage of using FP-Tree is that the overlapping itemsets share a common path and make the data highly compressed.To apply this algorithm, we mainly require **two passes** throughout the entire dataset. In the first scan, it calculates the supports of each item and identify the frequent itemsets and discard the infrequent ones. Using this step, they arrange the frequent itemsets in decreasing order based on their support. Using pass 2, it generates the association rules. Now let me describe the detailed steps of FP Algorithm using an example
. We will apply FP Algorithm on the same dataset mentioned above.

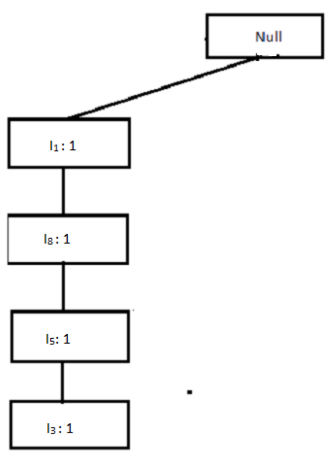
1. Just as the Apriori Algorithm, we need to build a table with frequency of individual items **??**. Here FP Algorithm uses its first pass. Since minimum support count is 2, we will first choose all frequent items. These elements are stored in descending order of their respective frequencies. When two elements have the same support count, we have placed I_i before I_j if $i < j$. Now using the frequent items, the data is looking as follows.
$$D = \{I_1(5), I_8(5), I_5(4), I_7(4), I_3(3), I_4(3)\} \tag{1}$$

This is called Frequent Pattern Set. Since I_2 and I_6 have support count less than 2, they have excluded. We can organize our data as follows.

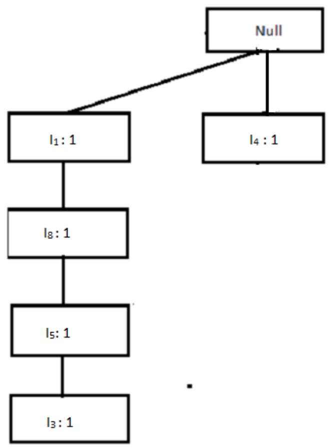
Transactions	Items	Ordered-Item Set
T_1	I_1, I_3, I_5, I_8	I_1, I_8, I_5, I_3
T_2	I_2, I_4, I_6	I_4
T_3	I_1, I_4, I_5, I_7	I_1, I_5, I_7, I_4
T_4	I_1, I_4, I_8	I_1, I_8, I_4
T_5	I_5, I_7, I_8	I_8, I_5, I_7
T_6	I_1, I_3, I_5, I_7	I_1, I_5, I_7, I_3
T_7	I_3, I_8	I_8, I_3
T_8	I_1, I_7, I_8	I_1, I_8, I_7

2. Now all the ordered itemsets are inserted into a Trie Data Structure.

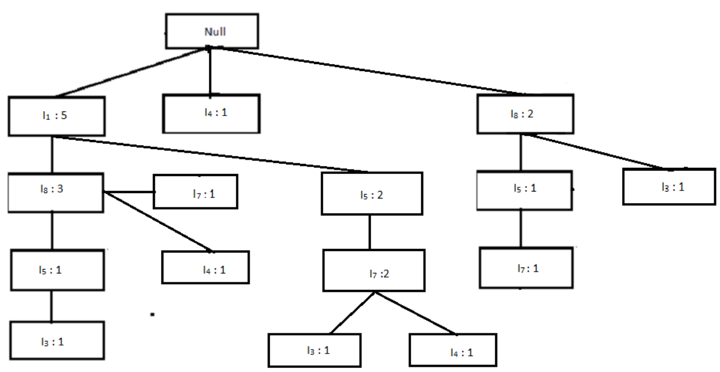
   - we will insert the set $\{I_1, I_8, I_5, I_3\}$.



   - Then we will insert $\{I_4\}$



- Next we will insert all the other 5 orered set chronologically and finally get the image like this.



3. Conditional Pattern Base is the path labels of all the paths which lead to any node of the given item in the frequent -pattern tree. For each item, now we have to calculate that.

Items	Conditional Pattern Base
I_4	$\{ \{I_1, I_5, I_7 : 1\}, \{I_1, I_8 : 1\} \}$
I_3	$\{\{I_1, I_8, I_5 : 1\}, \{I_1, I_5, I_7 : 1\} \{I_8 : 1 \} \}$
I_7	$\{\{I_1, I_5 : 2\}, \{I_1, I_8 : 1\}, \{I_8, I_5 : 1\} \}$
I_5	$\{\{I_1, I_8 : 1\}, \{I_1 : 2\}, \{I_8 : 1\}\}$
I_8	$\{\{I_8 : 3\}\}$
I_1	

4. From the Conditional pattern base, we will look into, which are the elements common in all the paths of a particular item. Taking the set of all such elements, we can make conditional frequent pattern Tree. We need to calculate the support count by summing up the support counts of all the paths in the conditional pattern base. For our data, Conditional Pattern Tree is structured as follows.

Items	Conditional Pattern Base	Frequent Pattern Tree
I_4	$\{\ \{I_1, I_5, I_7 : 1\}, \{I_1, I_8 : 1\}\ \}$	$\{I_1 : 2\}$
I_3	$\{\{I_1, I_8, I_5 : 1\}, \{I_1, I_5, I_7 : 1\}\ \{I_8 : 1\ \}\ \}$	$\{I_1 : 3\}$
I_7	$\{\{I_1, I_5 : 2\}, \{I_1, I_8 : 1\}, \{I_8, I_5 : 1\}\ \}$	$\{I_1 : 3\}$
I_5	$\{\{I_1, I_8 : 1\}, \{I_1 : 2\}, \{I_8 : 1\}\}$	$\{I_1 : 2\}$
I_8	$\{\{I_1 : 3\}\}$	$\{I_1 : 3\}$
I_1		

5. Pairing the items of the conditional Frequent Pattern Tree set, we can generate the frequent Pattern Rules corresponding to each item from conditional Frequent Pattern tree. It is given in the table below.

Items	Frequent Pattern Generated
I_4	$\{\ \{I_1, I_4 : 2\}\}$
I_3	$\{\{I_1, I_3, : 3\}\}$
I_7	$\{\{I_1, I_7 : 3\}\}$
I_5	$\{\{I_1, I_5 : 2\}\}$
I_8	$\{\{I_1, I_8 : 3\}\}$
I_1	

Then we will calculate all possible rules from each frequent item set and will finally consider those, which rules have confidence greater than or equal to minimum confidence just as we have illustrated for Apriori algorithm.

2.3 Improvement of FP Algorithm based on an Adjacent Table

We have already discussed that FP Algorithm requires two database scans and create a FP-Tree that contains all the itemsets. In the paper (4), Yin et al. have argued that FP Tree requires a lots of memory to store it. Moreover, "if the frequent itemsets is too many and the memory can't load the mapping information of all the items in the FP-Tree, the algorithm will not be effective."(4)(2) For a huge dataset, scanning it twice deteriorates the performance of the algorithm.

Thus they have proposed a new method based on adjacency table. Now let us explain the algorithm using our hypothetical dataset.

1. **Generation of Adjacency Table:**
   Here, we assume that items of each itemsets is related to each other. They can form a complete graph. One the same pair of items is transacted twice, the weight of the edge is incremented by one. The weight of the final edge is termed as *Association Frequency.* After the first scan of the database, the following graph is obtained.

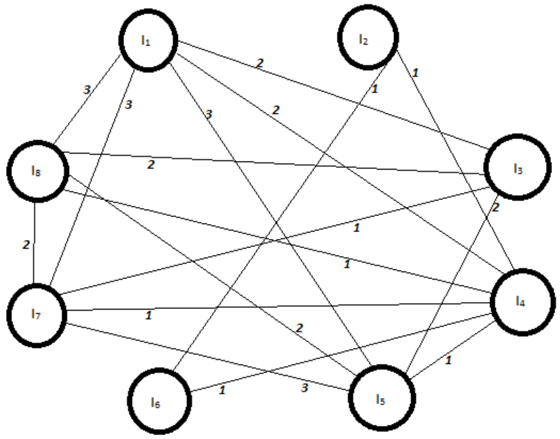


Figure 1: Graph Showing All Combinations along with Frequency

2. **The Mining of frequent itemsets:**

   Since our minimum support count is 2, we have excluded the itemsets whose support counting is less than 2 and can get the frequent itemsets as follows:

ItemSets	Support Counts
(I_1, I_3)	2
(I_1, I_4)	2
(I_1, I_5)	3
(I_1, I_7)	3
(I_1, I_8)	3
(I_3, I_4)	0
(I_3, I_5)	2
(I_3, I_7)	1
(I_3, I_8)	2
(I_4, I_5)	1
(I_4, I_7)	1
(I_4, I_8)	1
(I_5, I_7)	3
(I_5, I_8)	2
(I_7, I_8)	2

   Here also, we will prune the infrequent items. We can again plot the graph and continue to mine the adjacency table. Now we will get the following,

Candidate Sets (C_3)	Support Counts	Frequent Itemset (L_3)
(I_1, I_3, I_4)	0	
(I_1, I_3, I_5)	2	(I_1, I_3, I_5)
(I_1, I_3, I_7)	1	
(I_1, I_3, I_8)	1	
(I_1, I_4, I_5)	1	
(I_1, I_4, I_7)	1	
(I_1, I_4, I_8)	1	
(I_1, I_5, I_7)	2	(I_1, I_5, I_7)
(I_1, I_5, I_8)	1	
(I_1, I_7, I_8)	1	
(I_3, I_5, I_8)	1	
(I_5, I_7, I_8)	1	

Thus, we can get the frequent items.

Next, we will calculate all possible candidate rules from each frequent itemset and will consider only those which has confidence greater than or equal to minimum confidence just as we have illustrated for Apriori algorithm. In the paper(4), the authors have argued that this mehods take less time than FP Algorithm.

3 Application

3.1 Data Description

We have applied these algorithms to a real dataset named "Heart Attack Analysis & Prediction" Dataset. Here is the source of the dataset. We have divided the entire data in two sets, one is for those who has more chance to heart attack and other is for the remaining who has less chance of heart attack. Description of the dataset is given as follows.

1. **Age:** Age of patient is classified as follows.
   - Class 1: < 44
   - Class 2: $44 - 52$
   - Class 3: $52 - 59$
   - Class 4: > 59
2. **Sex:** Males and Females.
3. **Exang:** exercise induced angina (1 = yes; 0 = no).
4. **ca:** number of major vessels (0-3). Here we have assumed that number of vessels corresponds to each class.
5. **cp:** Chest Pain type is classified as follows.
   - Value 1: typical angina.

- Value 2: atypical angina
- Value 3: non-anginal pain
- Value 4: asymptomatic

6. **trtbps:** resting blood pressure (in mm Hg). We have classified the variable as follows.
   - Class 1: ≤ 120
   - Class 2: $120 - 130$
   - Class 3: $130 - 140$
   - Class 4: > 140
7. **chol:** cholestoral in mg/dl fetched via BMI sensor. It is classified as follows.
   - Class 1: ≤ 208
   - Class 2: $208 - 234$
   - Class 3: $234 - 267$
   - Class 4: > 267
8. **fbs:** (fasting blood sugar $>$ 120 mg/dl) (1 = true; 0 = false)
9. $rest_ecg:$ resting electrocardiographic results. It is classified as follows.
   - Class 1: ST-T wave abnormality
   - Class 2: definite left ventricular hypertrophy
   - Class 3: normal
10. **thalachh**
11. **thalach:** maximum heart rate achieved. It is classified as follows.
    - Class 1: ≤ 149
    - Class 2: $149 - 161$
    - Class 3: $161 - 172$
    - Class 4: > 172
12. **oldpeak** It is classified as follows.
    - Class 1: $oldpeak_0.2 < oldpeak <= 1$
    - Class 2: $oldpeak_0 < oldpeak <= 0.2$
    - Class 3: $oldpeak_oldpeak > 1$
13. **slp** Slope. It is classified as follows.
    - Class 1: $downslopping$
    - Class 2: $flat$
    - Class 3: $upsloping$

14. **thall**

    - Class 1: $noeffect$
    - Class 2: $normal$
    - Class 3: $reversibledefect$

15. **target:** Chance of heart attack. It is classified as follows.

    - 0 = less chance of heart attack
    - 1 = more chance of heart attack

Here age, trtbps, chol, thalach, oldpeak - these are continuous features. They are classified according <Q1, Q1-Q2, Q2-Q3 and >Q3 for all features. Rest features are categorical and classified according to their respective class.

Here each class of the each variables denotes a unique item. Thus we have 43 items in total. We have applied these algorithms to two sets of data and we have tried to find association among these items. **Basically our aim is to find, which are variables, specifically which class of which variable appears more frequently for a person, who has a higher chance of heart attack.** Indirectly, we are trying to find the causes of heart attack.

In our algorithm, we have named our **itemsets** following the rule below:

3.2 Results:

3.2.1 First we are representing results for target = 1

From Apriori Algorithm:

- Combinations
  - (trtbps$>$ 140, Class No of exng, 4 major vassels)
- Minimum Support: 30
- Minimum Confidence: 0.5

From FP Growth Algorithm:

- Combinations
  - {(Class No of exng,oldpeak $>$ 1), (fbs $\leq$ 120 mg/dl,oldpeak $>$ 1), (caa = 0, Class No of exng), (caa = 0, fbs $\leq$ 120 mg/dl), (Class Normal of thall, Class No of exng), (Class Normal of thall, fbs $\leq$ 120 mg/dl), (Class NO of exng, fbs $\leq$ 120 mg/dl)}
- Minimum Support Ratio: 0.65
- Minimum Confidence: 0.5

From Advanced Algorithm over FP Growth:

  - Combinations

      * (trtbps> 140, Class No of exng, 4 major vassels)
  - Minimum Support: 30
  - Minimum Confidence: 0.5

Thus we have got some sets of frequent items. We can say those who are prone to heart attack, belongs to these classes. Although, we cannot say that these are the cause of heart attack but these frequent characteristics can be seen tto the patients who are more prone to heart attack.

3.2.2 Next we are representing results for target = 0

From Apriori Algorithm:

- Combinations
  - {(male, fbs $\leq$ 120 mg/dl, oldpeak $>$ 1) (Class asymptomatic of cp, fbs $\leq$ 120 mg/dl, oldpeak $>$ 1)}
- Minimum Support: 30
- Minimum Confidence: 0.5

From FP Growth Algorithm:

- Combinations
  - {(Class asymptomatic of cp, oldpeak $>$ 1), (oldpeak $>$ 1, male), (fbs $\leq$ 120 mg/dl, male), (fbs $\leq$ 120 mg/dl, oldpeak $>$ 1)}
- Minimum Support: 0.65
- Minimum Confidence: 0.5

From Advanced Algorithm over FP Growth:

- Combinations
  - {(male, fbs $\leq$ 120 mg/dl, oldpeak $>$ 1) (Class asymptomatic of cp, fbs $\leq$ 120 mg/dl, oldpeak $>$ 1)}
- Minimum Support: 30
- Minimum Confidence: 0.5

Thus we have got some sets of frequent items for the patients who are less prone to heart attack. Although, we cannot say that these are the cause of heart attack but these frequent characteristics can be seen to the patients who are not prone to heart attack.

We can further note that patients with less blood pressure are less prone to Heart attack, and with high blood pressure are more prone to heart attack. So it is likely that high blood pressure plays a role in heart attack.

4 Conclusion

In this project, we have basically tried to compare the performance of three algorithms. Every algorithm has its own pros and cons. We have tried to mention the drawbacks of these algorithms and have searched for the methods to overcome the drawbacks in literature. Since we had difficulty in finding dense dataset, we have not shown any real life application of the third method.However, we can conclude that FP Algorithm, performs faster than Apriori and gives anticipated result in most of the situations.

References

[com]

[2] Jiang, H. and Meng, H. (2017). A parallel fp-growth algorithm based on gpu. *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, pages 97–102.

[3] Yadav, C., Wang, S., and Kumar, M. (2013). An approach to improve apriori algorithm based on association rule mining. pages 1–9.

[4] Yin, M., Wang, W., Liu, Y., and Jiang, D. (2018). An improvement of fp-growth association rule mining algorithm based on adjacency table. *MATEC Web of Conferences*, 189:10012.