Comparison of FP Growth & Apriori algorithm using chess & mushroom datasets

Kamruzzaman leeon ID: 2018-1-60-252 Email: 2018-1-60-252@std.ewubd.edu	Kamrul Hasan ID: 2018-1-60-039 Email: 2018-1-60-039@std.ewubd.edu	Nazmus Sakib ID: 2018-1-60-104 Email: 2018-1-60-104@std.ewubd.edu	Md. Tanbin Hossain Himel ID: 2018-1-60-151 Email: 2018-1-60-151@std.ewubd.edu	Md. Moniruzzaman Shanto ID: 2018-1-60-075 Email: 2018-1-60-075@std.ewubd.edu

TOOLS

- Colab research
- Python 3
- Minimum ram required 16GB

DATASETS DESCRIPTION

- Chess and mushroom datasets[1] were prepared by Roberto Bayardo from the UCI datasets and PUMSB.
- No specific description of attributes found.
- In the dataset every row is a single transaction.
- Every cell is an individual item.
- The number of transactions in chess is 3196 and the number of items is 75.
- The number of transactions in mushroom is 8124 and the number of items is 119.
- The datasets are preprocessed & no missing value is found.

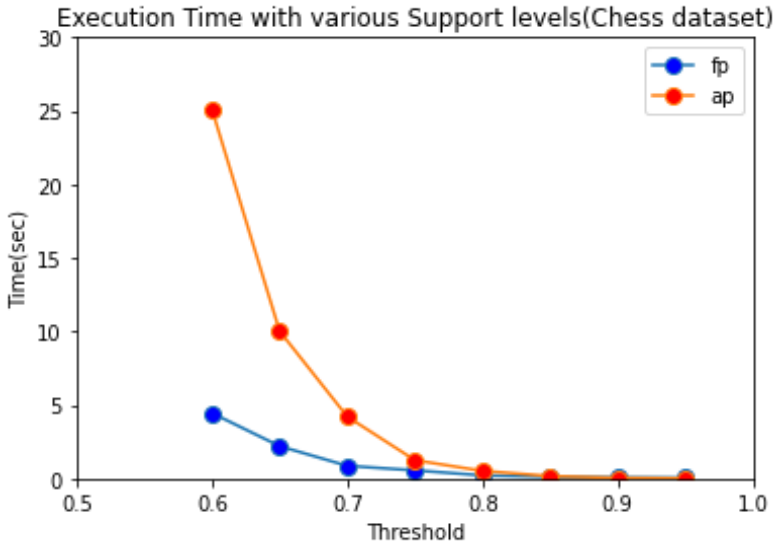
ALGORITHM COMPARISON

	FP Growth	**Apriori**
Speed	Faster, runtime increases linearly with increase in number of item sets.	Slower, runtime increases exponentially with increase in number of item sets.
Memory	Need small memory because it stores the compact version of database.	Need large memory because all the candidates from self-joining are stored in the memory.
Candidates	No candidate generation.	Use self-joining for candidate generation.
Frequent Patters	Pattern growth achieved by mining conditional FP trees.	Patterns selected from the candidates whose support is higher than minSup.
Scans	Only require two scans.	Scan the database over and over.

We have organized the main features of both algorithms and made them into the table above. From observing the table, we can tell that FP Growth is generally better than Apriori under most circumstances. That's why Apriori is just a fundamental method, and FP Growth is an improvement of it.

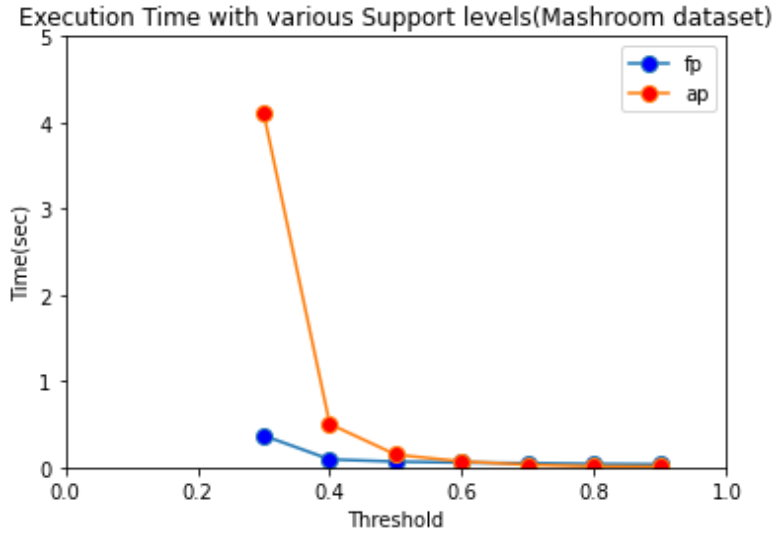
RESULT ANALYSIS

ANALYSIS OF EXECUTION TIME WITH VARIOUS SUPPORT FOR CHESS DATASET



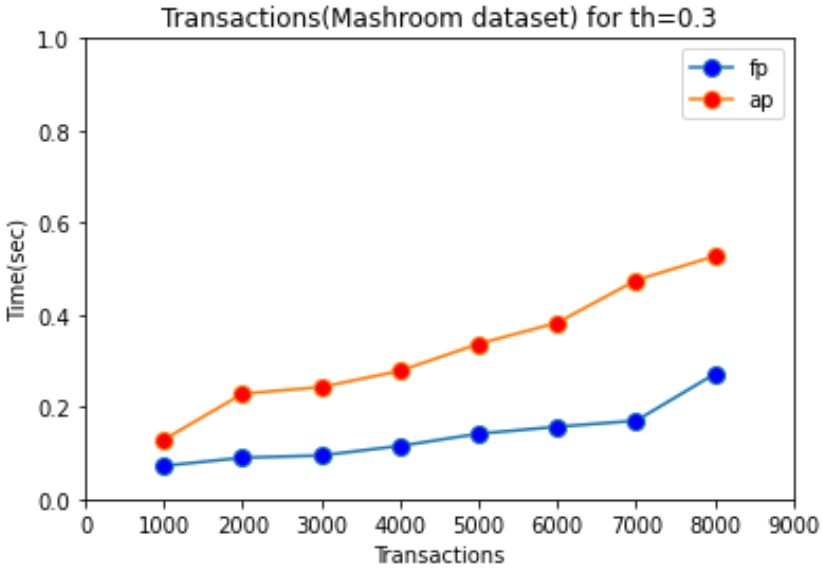
From the plotted graph for chess dataset, it is visible that execution time for Apriori and FP Growth algorithm is decreasing with respect to threshold. When the threshold is very low, the Apriori and FP Growth algorithm is taking much time on the other side when the threshold is high both algorithms are taking less execution time. Above graph shows that the Apriori algorithm is taking more time compared to the FP Growth algorithm. Which means the FP Growth algorithm is performing better for the chess dataset. Above graph also shows that for threshold 0.6 Apriori algorithm is taking 25 seconds to execute the operation on the other side FP Growth algorithm is taking around 5 second which is way better performance. Similarly, if we keep increasing the threshold value, time for both algorithms are decreasing but the Apriori algorithm is decreasing slowly compared to the FP Growth algorithm. So, the conclusion is that the FP Growth algorithm is performing better for the chess dataset because FP Growth algorithm scans the dataset only two times. In the first scan, it counts the frequency of item & generates header table. In the second scan, it sorts the item sets of the transactions according to header table.

ANALYSIS OF EXECUTION TIME WITH VARIOUS SUPPORT FOR MUSHROOM DATASET:



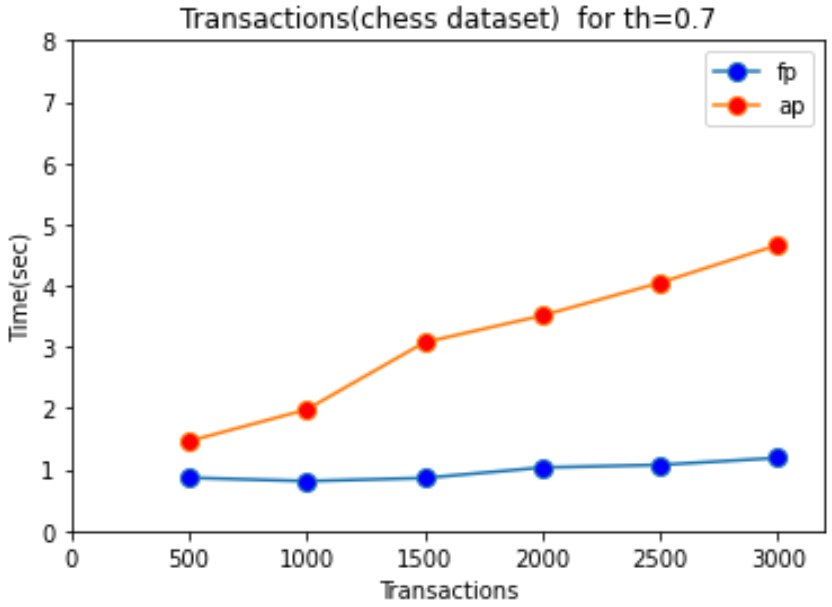
From the plotted graph for Mushroom dataset, it is visible that execution time for Apriori and FP Growth algorithm is decreasing with respect to threshold. When the threshold is very low, the Apriori and FP Growth algorithm is taking much execution time on the other side when the threshold is high both algorithms are taking less execution time. Above graph shows that the Apriori algorithm is taking more execution time compared to the FP Growth algorithm. Which means the FP Growth algorithm is performing better for the Mushroom dataset. Above graph also shows that for threshold 0.3 Apriori algorithm is taking 4 seconds to execute the operation on the other side FP Growth algorithm is taking less than 1 second to execute which is way better performance. Similarly, if we keep increasing the threshold value, time for both algorithms are decreasing but the Apriori algorithm is decreasing slowly compared to the FP Growth algorithm. So, the conclusion is that the FP Growth algorithm is performing better for the Mushroom dataset because FP Growth algorithm scans the dataset only two times. In the first scan, it counts the frequency of item & generates header table. In the second scan, it sorts the item sets of the transactions according to header table.

TRANSACTION ANALYSIS FOR MUSHROOM DATASET:



From the plotted graph for Mushroom dataset, it is visible that, when the number of transactions is increasing execution time is also increasing for Apriori algorithm. The Apriori algorithm scans all the frequent patterns which takes more time to execute. This is the reason the plotted line in the graph is increasing for Apriori algorithm. On the other hand, when the number of transactions is increasing execution time is also increasing for the FP Growth algorithm but compared to Apriori algorithm is taking less execution time because FP Growth algorithm scans the dataset only two times. In the first scan, it counts the frequency of item & generates header table. In the second scan, it sorts the item sets of the transactions according to header table. The graph shows that for 1000 transactions Apriori algorithm is taking more than 1 second and FP Growth algorithm is taking less than 1 second. Similarly, if we keep increasing the number of transactions, execution time for both algorithms are increasing. For 8000 transactions the Apriori algorithm is taking more than 5 seconds and the FP Growth algorithm is taking less than 3 seconds to execute. The conclusion is that the FP Growth algorithm is performing better for the Mushroom dataset for all the number of transactions.

Transaction Analysis for Chess dataset:



From the plotted graph for the chess dataset, it is visible that, when the number of transactions is increasing, execution time is also increasing for Apriori algorithm. The Apriori algorithm scans all the frequent patterns which takes more time to execute compared to the FP Growth algorithm. For this reason, the plotted line in the graph is increasing for the Apriori algorithm. On the other hand, when the number of transactions is increasing, execution time is also increasing for the FP Growth algorithm but compared to Apriori algorithm it is taking less execution time because FP Growth algorithm scans the dataset only two times. In the first scan, it counts the frequency of item & generates header table. In the second scan, it sorts the item sets of the transactions according to header table. The graph shows that for 1000 transactions the Apriori algorithm is taking more than 1 second and the FP Growth algorithm is taking less than 1 second. Similarly, if we keep increasing the number of transactions, execution time for both algorithms are increasing. For 8000 transactions the Apriori algorithm is taking about 5 seconds and the FP Growth algorithm is taking around 1 seconds to execute. The conclusion is that the FP Growth algorithm is performing better for the chess dataset for all the number of transactions.

CONCLUSION

After analyzing the results of the above datasets , we can say that FP Growth is faster than Apriori algorithm.

REFERENCE

1. http://fimi.uantwerpen.be/data/