Department of Computer Science and Engineering

University of Dhaka

A Comparison between trie-based Apriori algorithm and FP Growth algorithm for frequent pattern mining

Submitted By

Muntasir Wahed (Roll - 007)

Supervised By

Dr. Chowdhury Farhan Ahmed
Dr. Moinul Islam Zaber

Date: October 10, 2018

Introduction

Frequent itemset mining is one of the most widely analysed problem in the field of data mining. Two popular algorithm for mining frequent itemsets are Apriori algorithm and FP growth algorithm. In this study, we compare between these two algorithms with respect to their time runtime and memory limitations.

Experimental Setup

We compare the two algorithms with five different datasets collected from http://fimi.ua.ac.be/data/. The datasets used in this study are as follows.

1. Chess
2. Mashroom
3. Pumsb_star
4. Accidents
5. T10I4D100K

Among these datasets, Chess and Pumsb_star are highly dense. Mashroom and Accidents are less dense, whereas T10I4D100K is very sparse.

Results

Min Support	Time (Apriori)	Time (Fp-Growth)	# frequent Patterns
0.2	33.41	1.31	53583
0.25	1.33	0.91	5535
0.3	0.7	0.78	2735
0.35	0.35	0.7	1189
0.4	0.35	0.68	565
0.45	0.15	0.6	329
0.5	0.1	0.58	153
0.55	0.09	0.53	99
0.6	0.064	0.5	51

Figure 1: Comparison using Mashroom dataset

Min Support	Time (Apriori)	Time (Fp-Growth)	# frequent Patterns
0.6	792.87	53.75	254944
0.65	167.49	15.27	111239
0.7	35.902	3.7	48731
0.75	9.65	1.37	20993
0.8	2.48	0.753	8227
0.85	0.66	0.565	2669
0.9	0.18	0.522	622

Figure 2: Comparison using Chess dataset

Min Support	Time (Apriori)	Time (Fp-Growth)	# frequent Patterns
0.35	204.31	9.61	116787
0.4	72.89	6.24	27354
0.45	6.54	4.64	1913
0.5	4.08	2.97	679
0.55	2.53	2.02	305
0.6	1.65	1.53	167
0.65	1.12	1.39	90
0.7	0.63	1.19	29

Figure 3: Comparison using pumsb_star dataset

Min Support	Time (Apriori)	Time (Fp-Growth)	#f frequent Patterns
0.35	54.51	52.12	42538
0.4	22.27	7.95	20483
0.45	11.34	3.64	10558
0.5	6.26	1.93	5476
0.55	3.75	1.45	2968
0.6	2.14	1.02	1627
0.65	1.34	0.885	859
0.7	0.837	0.82	495

Figure 4: Comparison using Accidents dataset

Min Support	Time (Apriori)	Time (Fp-Growth)	# frequent Patterns
0.1	0.14	0.72	411
0.2	0.06	0.56	160
0.3	0.04	0.5	65
0.4	0.03	0.47	25
0.5	0.01	0.45	11
0.6	0.01	0.46	4
0.7	0.01	0.43	1
0.8	0.01	0.44	1
0.9	0.01	0.45	0

Figure 5: Comparison using T10I4D100K dataset

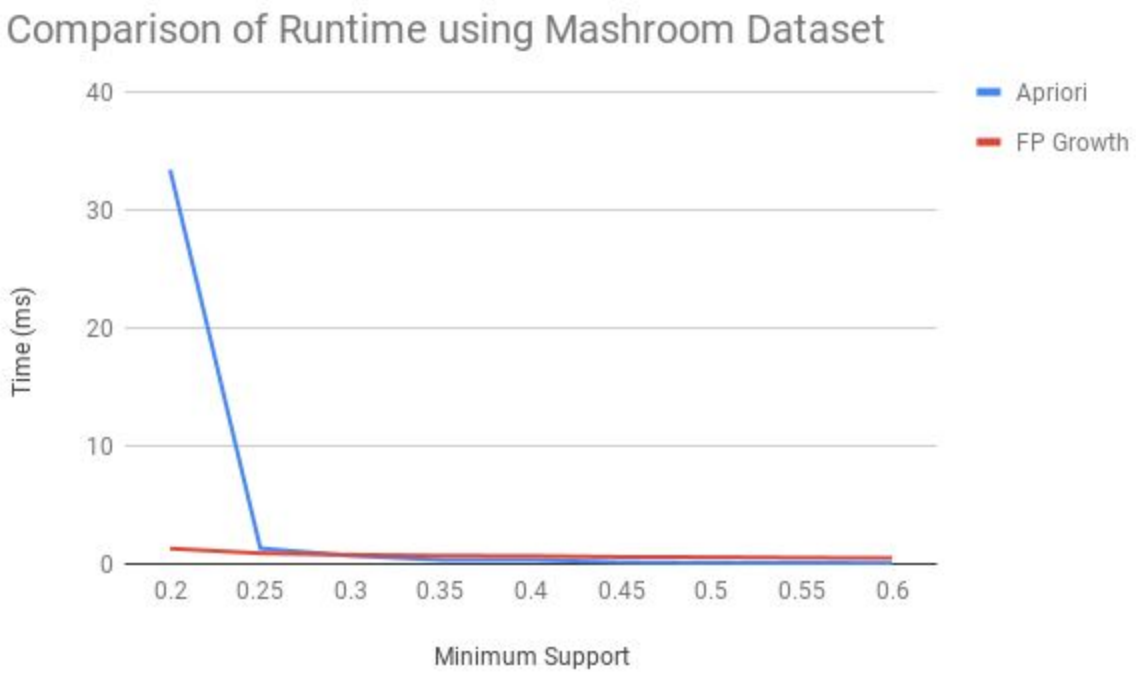

Figure 6: Comparison of runtime using Mashroom Dataset

Figure 7: Comparison of runtime using Chess Dataset

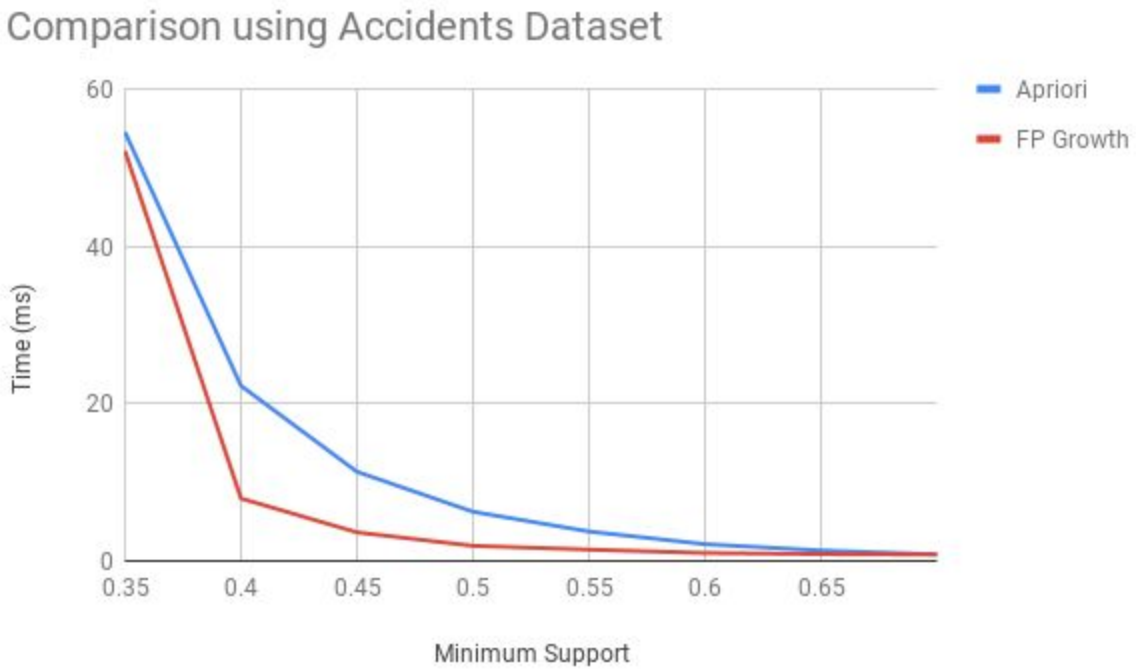

Figure 8: Comparison of runtime using Accidents Dataset

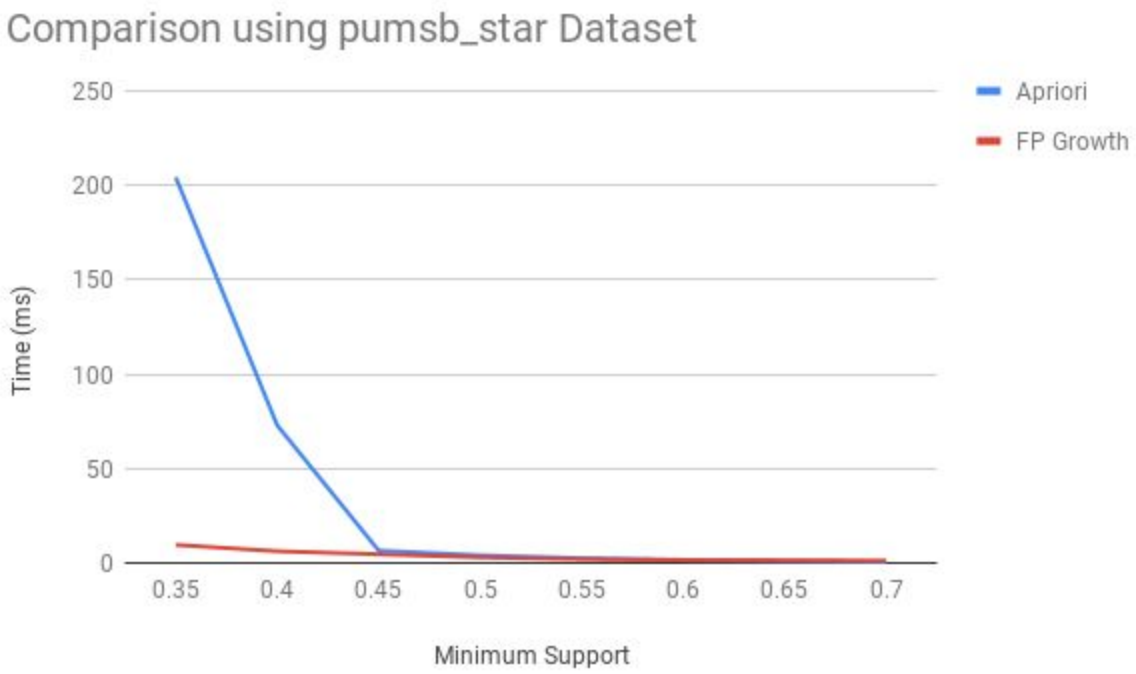

Figure 9: Comparison of runtime using pumsb_star Dataset

Figure 10: Comparison of runtime using T10I4D100K Dataset

Discussion

As seen from Figures 7 and 9, which correspond to the most dense datasets Chess and Pumsb_star, FP Growth algorithm works significantly better than Apriori algorithm in terms of runtime, almost 15 times faster in case of a minimum support of 60% for chess dataset.
On the other hand, in case of less dense datasets, as seen from Figures 6 and 8, there is no significant difference in terms of runtime.
Finally, as seen from Figure 10, Apriori algorithm works better than FP Growth algorithm. This is due to the fact that, the T10I4D100K dataset is very sparse. Hence the implementation overhead of FP Growth is larger than the overhead of reading the dataset multiple times, making Apriori algorithm more efficient than FP Growth algorithm in this particular case.

Conclusion

We can conclude from the study that, FP Growth algorithm is more efficient when the number of candidate patterns is very large. Otherwise the implementation overhead of FP Growth makes it less efficient than Apriori algorithm. On the other hand, one significant issue with Apriori algorithm is the memory overhead as well as the runtime in case of dense datasets.