**Department of Computer Science and Engineering**

**University of Dhaka**


---

$4^{th}$ *year* $2^{nd}$ *Semester*, 2021

CSE4255: Introduction to Data Mining and Warehousing Lab

Assignment: 01

# Implementation of Apriori and FP-Growth Algorithm.

**Submitted by:**

Md. Ahasanul Alam, Roll: 10

**Submission Date:**

28 January 2021

**Submitted to:**

Dr. Chowdhury Farhan Ahmed.

Professor, Department of Computer Science and Engineering,

University of Dhaka.

Frequent pattern mining searches for recurring relationships in a given data set. Two widely used frequent pattern mining Algorithms are (i) Apriori and (ii) FP Growth. In this assignment, we have implemented the Apriori and FP Growth algorithms and examined the runtime and memory usages of the algorithms on the following five data sets:

- Chess (Dense data set)
- Mushroom (Dense data set)
- T10I4D100K (Sparse data set)
- Retail (Sparse data set)
- Kosarak (Large data set)

All programs are written in Python programming language (version 3.8.5) in google colab with a 2vCPU @ 2.2GHzand and 13 GB memory. Runtime specifies the total execution time, i.e. CPU, I/Os, and it includes tree construction, tree restructuring, and mining time. Memory usages refers to the maximum memory that were required during the execution time.

| Size (MB) | #Transactions | #Items | Max TL | Avg TL | AVgTL/#Trans*(100) |
|---|---|---|---|---|---|
| 0.34 | 3196 | 75 | 37 | 37.00 | 49.33 |

Table 1: Properties of Chess dataset.

# 1 Chess

Chess is a dense dataset with the properties showed in Table 1. As Minimum support threshold, we used values in range [95% -60 %]. The comparisons between Apriori and FP Growth in terms of runtime and Memory usages are shown in Figure 1. From Figure 1, we can see Apriori takes more runtime and memory as the minimum support threshold decrease. This is because with lower support threshold, more patterns are generated and Apriori needs to scan the data several times. This adds extra runtime overhead.


Figure 1: Runtime and Memory comparison on Chess Data set

| Size (MB) | #Transactions | #Items | Max TL | Avg TL | AVgTL/#Trans*(100) |
|---|---|---|---|---|---|
| 0.83 | 8124 | 119 | 23 | 23.00 | 19.33 |

Table 2: Properties of Mushroom data set.

# 2 Mushroom

Mushroom is another dense data set whose properties are shown in Table 2. Minimum support thresholds are used in range [70% - 30%]. The comparisons between Apriori and FP Growth in terms of runtime and

Memory usages are shown in Figure 2. Similar to Chess data set, Apriori takes more time to generate the patterns. But FP-growth takes more memory than Apriori. This is because FP-growth need to maintain a header table, run recursively and sort the header tables in descending support count. As a result, extra memory are required. On the other hand,In Apriori, we removes the $(n-1)^{th}$ and above nodes in trie while working for $n^{th}$ nodes. This saves extra memory usages.

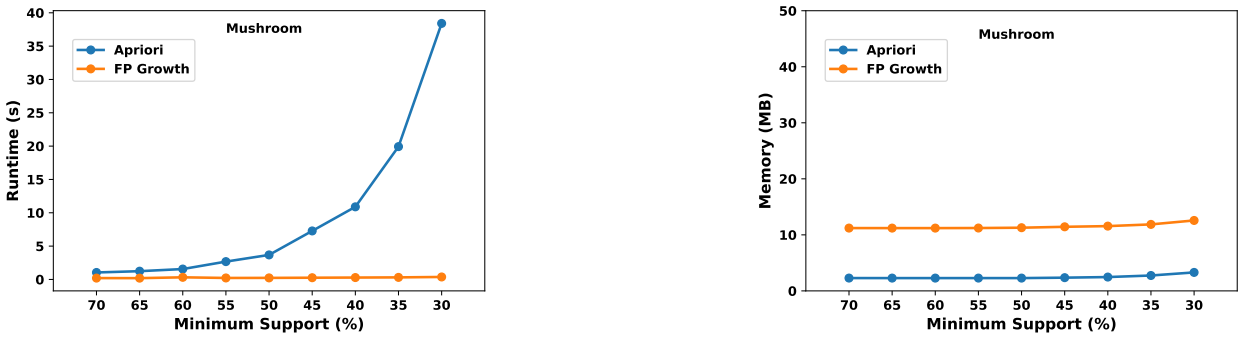


Figure 2: Runtime and Memory comparison on Mushroom Data set

| Size (MB) | #Transactions | #Items | Max TL | Avg TL | AVgTL/#Trans*(100) |
|---|---|---|---|---|---|
| 3.93 | 1,00,000 | 870 | 29 | 10.10 | 1.16 |

Table 3: Properties of T10I4D100K data set.

# 3 T10I4D100K

T10I4D100K is a sparse data set with the properties shown in Table 3. As Minimum support threshold, we used values in range [2% - 6%]. The comparisons between Apriori and FP Growth in terms of runtime and Memory usages are shown in Figure 3. Unlike the previous two data set here FP Growth takes more time with lower support threshold. This is because of the sparcity of the data set. In sparse data set lower support threshold generates more patterns but length of patterns (no of item in a single pattern) does increase much. As Apriori works as level by level, it doesn't needs to increase many levels. As a result can find solution faster than FP Growth.

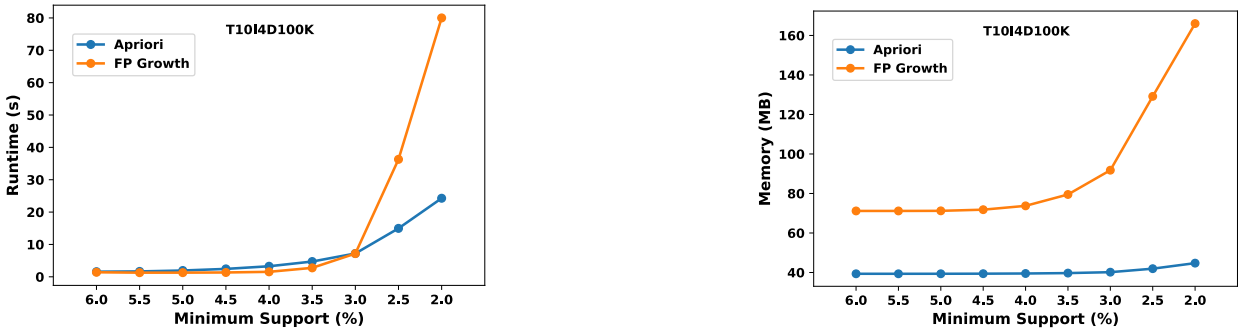


Figure 3: Runtime and Memory comparison on T10I4D100K Data set

# 4 Retail

Retail is another sparse data set with the properties shown in Table 4. Here we used values in range [0.1% - 0.5%] as Minimum support threshold. The comparisons between Apriori and FP Growth in terms of runtime

| Size (MB) | #Transactions | #Items | Max TL | Avg TL | AVgTL/#Trans*(100) |
|---|---|---|---|---|---|
| 3.97 | 88,162 | 16,470 | 76 | 10.31 | 0.06 |

Table 4: Properties of Retail data set.

and Memory usages are shown in Figure 4. Apriori takes more time for lowed support threshold. This is because for lower threshold Apriori needs to generate more levels in trie. This increases the runtime. FP growth takes more memory as it needs to maintain a header table, run recursively and sort data over and over again.

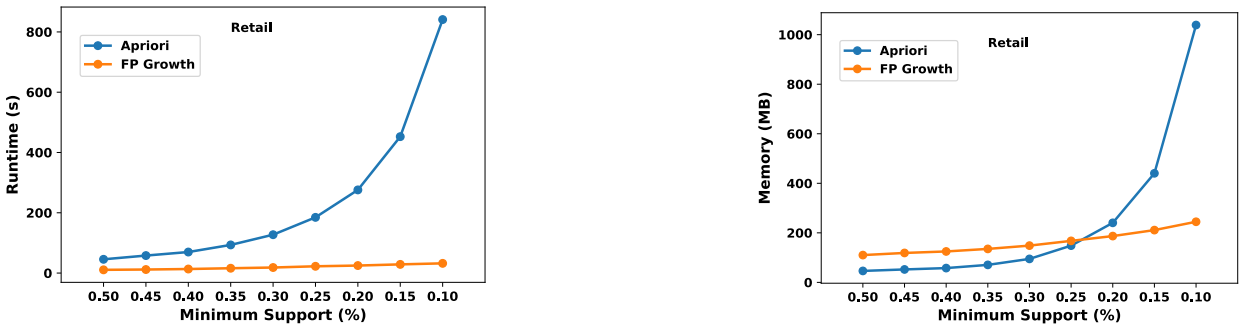


Figure 4: Runtime and Memory comparison on Retail Data set

| Size (MB) | #Transactions | #Items | Max TL | Avg TL | AVgTL/#Trans*(100) |
|---|---|---|---|---|---|
| 30.50 | 9,90,002 | 41,270 | 2498 | 8.10 | 0.02 |

Table 5: Properties of Kosarak data set.

## 5 Kosarak

Kosarak is a large and sparse data set5. As Minimum support threshold, we used values in range [2% - 8%]. The comparisons between Apriori and FP Growth in terms of runtime and Memory usages are shown in Figure 5. As the total number of transactions is huge here compared to the other four data sets, FP Growth outperforms Apriori. This is beacuse Apriori needs to scan this big database in every steps where FP Growth scans the database only twice. Thus runs faster than Apriori.


Figure 5: Runtime and Memory comparison on Kosarak Data set