Data Mining (COL 761)

Assignment: 1 Transactional Data Compression

Team: Data Crashers		
Ajay Tomar	Anshul Patel	Pulkit Singal
2023AIB2071	2023AIB2072	2023AIB2064

Project Structure:

Scripts:

- compile.sh: For compilation of all .cpp files
- interface.sh: For compression / decompression of data

.cpp files:

- fptree.cpp: For calculating minimum support threshold (*minsup*) and generating frequent itemsets using FP-Tree algorithm
- comp.cpp: For creating compressed dataset (containing compressed transactions and mappings)
- decomp.cpp: For decompression of compressed dataset to get back the original dataset.

Data files (naming convention for this report):

- f_itemset.dat: data file containing frequent itemsets
- compressed_dataset.dat: data file containing compressed data and mappings
- reconstructed.dat: data file containing decompressed data

Explanation:

Frequent Itemset Mining (fptree.cpp):

- *FP-Tree algorithm has been used for the generation of frequent itemsets.
- The minimum support threshold (*minsup*) value is calculated as per the Heuristic Function eq (1).
- *The frequent itemsets are written in the f_itemset.dat file.

Compression of Data (comp.cpp):

- The frequent itemsets are read from f_itemset.dat.
- The frequent itemsets are mapped to keys (negative integers) and stored in RAM.
- The mappings are arranged in descending order of the number of items in each frequent itemset.
- Transactions are read (one at a time) from the original transaction database.
- The subsets of each transaction, which are equal to any of the frequent itemsets in the mapping are replaced with their corresponding keys. If two frequent itemsets, one of which is a subset of another, occur in the same transaction, then the larger frequent itemset is used for replacement with its key.
- The mappings in which the frequent itemsets have not been replaced by their corresponding keys in any of the transactions are deleted from RAM.
- Writing the data to compressed_dataset.dat:
    - The first line of compressed_dataset.dat contains an integer (meta data) which is calculated based on the number of transactions in the transaction database. This represents the line number of the compressed_dataset.dat on which the first mapping is written.
    - Each transaction (in compressed form) is written on the compressed_dataset.dat file, starting from the second line.
    - After leaving a single blank line, only the mappings which are used for compression are written on the compressed_dataset.dat file.

Decompression of Data (decomp.cpp):

- The integer on the first line of compressed_dataset.dat is used for identifying the line number from which the mappings are written.
- All the mapping are loaded in the RAM.
- Starting from the second line of compressed_dataset.dat, each of the compressed transactions is read one by one.
- The negative integers in each compressed transaction are then replaced by the corresponding frequent itemset (as per the mapping), which is appended at the end of each transaction.
- Each of the decompressed transactions is written on the reconstructed.dat file.

Heuristic Function used for determining the minimum support threshold (*minsup*):

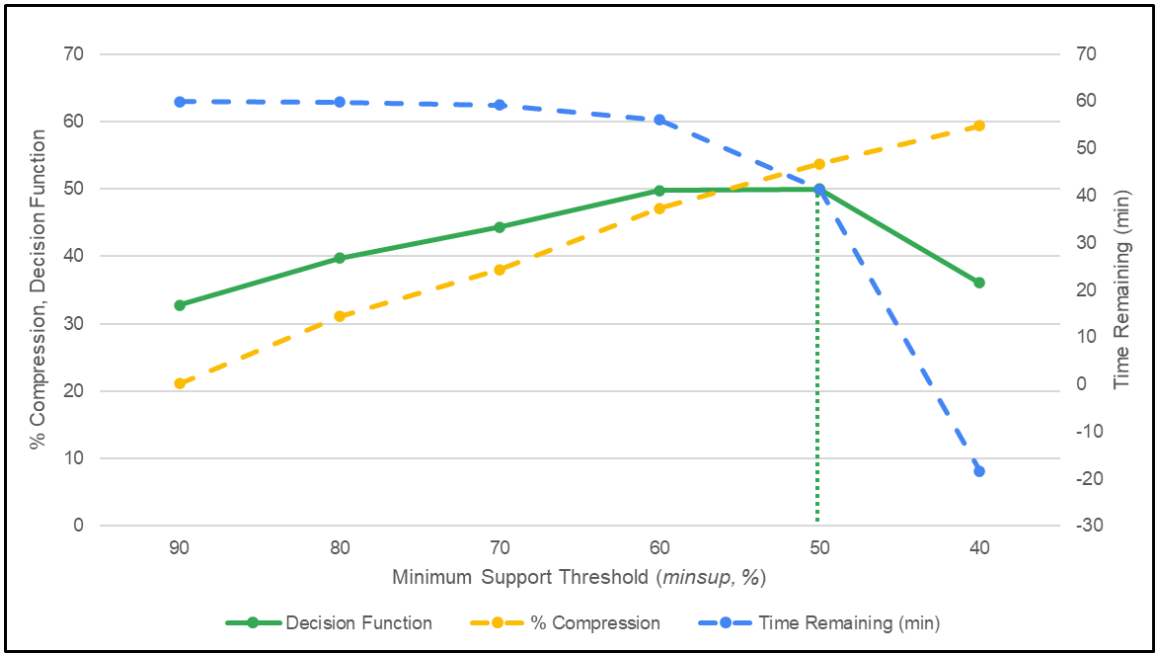
minsup = *(mean of support of top 50% individual items) * sigma / mu* (1)
(Note: In the above formula, the items are arranged in descending order of support)
where,

mu = Mean of the supports of individual items
sigma = standard deviation of the supports of individual items

Results/Observations:

- The below plot is obtained for the transaction dataset **D_small.dat**, by varying the value of *minsup* and plotting the values of the following dependent variables:

1. **% Compression:** Percentage of original transaction database compressed (in terms of individual integers)
2. **Time Remaining (min):** Time remaining in minutes (from the total time limit of 60 minutes) after the code is run completely
3. **Decision Function:** Objective function that is to be maximized. It is defined as follows:
   Decision Function = 0.7[% Compression] + 0.3*[Time Remaining (min)]* (2)



From the above plot, it is observed that the Decision Function has a maximum value corresponding to *minsup* = 50%. As per the Heuristic Function, the *minsup* value obtained is 55.77%, which is quite close to the optimal value of 50% obtained using the Decision Function.

- The decompressed data obtained in reconstructed.dat, is found to be completely lossless.

***Credits:**
fptree.cpp (For generating frequent itemsets using FP-Tree algorithm)