EDAMI 22L - IMPLEMENTING CHARM ALGORITHM

Omer AMAC
Sunny ROZARIO

November 29, 2022

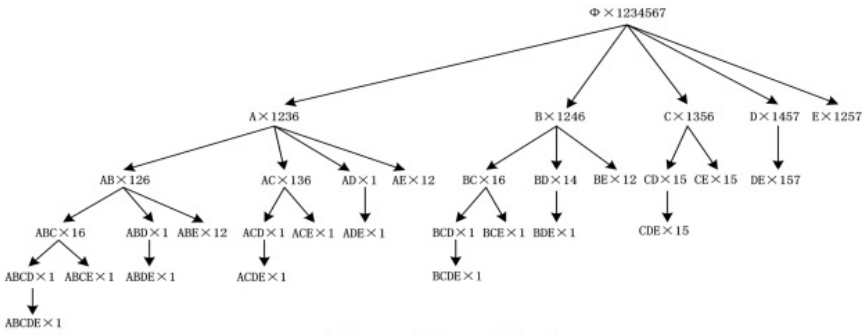
Contents

1 Introduction

One of the fundamental problem in Data mining is mining frequent closed patterns or itemsets in large transaction databases. An efficient Algorithm for closed itemsets mining was proposed by Mohammed J. Zaki and Ching-Jui Hsiao (2002). Our project is based on the CHARM algorithm. Charm is an algorithm for discovering frequent closed itemsets in a transaction database. The input is a transaction database that is a set of transactions and a threshold named minSup.[FV22] The Charm algorithm is an important algorithm because it is one of the first depth-first algorithm for mining frequent closed itemsets.

TID	Items
1	A B C D E
2	A B E
3	A C
4	B D
5	C D E
6	A B C
7	D E

(a) Transactional database



(b) Itemset-Tidset tree (IT-tree)

Figure 1: Example of a transactional database and corresponding itemset-tidset tree.[Che15]

IT-tree is a complete search space. It is such tree that every node on it represented by an itemset-tidset pair e.g. X×t(X) where X is an itemset. The whole IT-tree is constructed by all frequent itemsets with the common organization that all direct children of a node X are generated by combination of X and other nodes coming after it according to a total order and thus all children of this node share the same prefix X.

2 Outline

2.1 Objective

The objective of this project is to research the twitter data on the word 'covid' as the global conversations happening on Twitter. The project can help researchers on different fields to analyse the twitter activity of the coronavirus outbreak. Also, the research of this project can attract the attention of researchers related to different fields on knowledge such has social science, network science, infodemiology and others.The purpose of this project is finding the words which appears with the word "covid". During the pandemic, Twitter was the one of the most active social platform and people talked about covid all the time. However, we know that covid was effecting other things and these things were becoming the content of these tweets with covid title. We will try to find these contents. First, we will find the words which appears most with the covid word and then we will visually examine the result. With help of support values, we will not have to examine all the appearings, we will just exqmine the ones which appears with meaningful supports.

In this project we will use CHARM algorithm and Covid-19 tweets dataset from Twitter to examine the data. Aim of the project is checking the words appear with the "covid" word.We will try to find the these words and the supports of such transactions. minimum support value will be effective while checking the appearance of other words next to covid. We will analyze the results and try to express it in visual representations.

2.2 Method of Experiments

We will use CHARM algorithm and the advenced version of the CHARM algorithm. While reaching to our goal we will be also comparing these two versions of CHARM algorithm.

Algorithms will be implemented in python code and then we will use preprocessed twitter dataset as input of our algorithms. Then we will check if the both algorithms give the same result, and then we will compare the speed of these algorithms.

2.3 Description of Algorithm

CHARM builds the IT-tree and searches over it. Since lots of itemsets on IT-tree are not closed, CHARM utilizes 4 properties to skip many levels of the tree so that it can compress the search space. Let Xj and Xj be any two itemsets and X=XiXj, then the 4 properties are showed in the following and related proof can be found in r71.

If t(Xi)=t(Xj), then c(Xi)=c(Xj)=c(X)

If t(Xi)t(Xj), then c(Xi)=c(X), but c(Xj)c(X)

If t(Xi)t(Xj), then c(Xj)=c(X), but c(Xi)c(X)

If t(Xi)t(Xj), then c(Xi)c(Xj)c(X)

In the last step, CHARM uses a fast hash-based approach to check if the remaining itemset on the IT-tree are closed or not.[3]

Algorithm CHARM

```
1   procedure CHARM (D, min_sup)
2     [P] = { X_i × t(X_i) | X_i ∈ I and σ(X_i) ≥ min_sup}
3     Charm-Extend ([P], C = Φ)
4     return C  // a hash table saving all closed itemsets
5   end procedure

6   function Charm-Extend ([P], C)
7     for each X_i × t(X_i) in [P]
8       X = X_i and [P_i] = Φ
9       for each X_j × t(X_j) in [P] with X_j ≥_f X_i // f is a total order
10        X = X_i ∪ X_j and Y = t(X_i) ∩ t(X_j)
11        if (σ(X) ≥ min_sup) then // 4 properties
12          if t(X_i) = t(X_j) then
13            remove X_j from [P] and replace all X_i with X
14          else if t(X_i) ⊂ t(X_j) then
15            replace all X_i with X
16          else if t(X_i) ⊃ t(X_j) then
17            remove X_j from [P]
18            add X × Y to [P_i] with order f
19          else
20            add X × Y to [P_i] with order f
21        end if
22      end for
23      if ([P_i] ≠ Φ) then Charm-Extend ([P_i], C) end if
24      save X in C with related check
25    end for
26  end function
```

Figure 2: Pseudo code of CHARM algorithm.[FV22]

2.4 Data Sets and Tools

Twitter data set will be used for executing the experiments. It is preferred because of being a dynamic and more realistic database. Twitter data is the information collected by either the user, the access point, what's in the post and how users view or use your post. While this might sound somewhat vague, it's largely due to the massive amount of data that can be collected from a single Tweet.

External scraper library will be used for scraping the tweets. We will use snscraper in this project. We will store the tweets in a csv/excel file and then we will clean the data to make it suitable for give as an input to our algorithm.

3 Implementation of The CHARM Algorithm

Python language used while implementation of the CHARM algorithm. We used pandas library for dealing with the data frames.

3.1 CHARM Algorithm in Python Code

We have done the implementation of CHARM algorithm for mining closed frequent sets on the Twitter data. Implementation based on the scientific paper: CHARM: An Efficient Algorithm for Closed Itemset Mining Mohammed J. Zaki Ching-Jui Hsiao.
Our implementation is structured as follows:

The algorithm is implemented in Python using the Numpy library for Data Frame operations. Charm.py provides the algorithm functionality which is getting closed frequent sets given the input set of transactions. We have set the Support to 0.5 which is the minimal relative support of the itemset to be taken for the result. The support of item set is specified 0.5, which means that only itemsets that occur in 50 percent or more transaction will be included in the result.
The main script contains two classes which are Data Preparation class and CharmAlgorithm class. In the Data Preparation class we first import the twitter data and write it in a transactions form, as list of tuples i.e 'tid':tid,'item':element. The transform data then generates the list of transactions for each element in the input data and writes to the dataframe. Then we get the frequent items from the transactions form which returns the dataframe with one element itemsets with relative support -¿ min support.
The Algorithm class contains the implementation of the charm algorithm. The main algorithm is responsible for finding the closed itemsets and the output is stored in the dataframe, this method will chose the combination of itemset transaction id pairs appearing in the items grouped input dataframe and executes the 4 charm property method. The charm property then applies one of the 4 properties on the given itemset-transaction id pairs.

3.2 Input Data

We cleaned the data from the unncessary charcters and words.

Example format of the input is as follows:
1 : [covid poland regulation mask vaccine]
1 : [1 23 4 12 74]

With that, we gave unique transaction id/name for all worlds in the whole tweets.

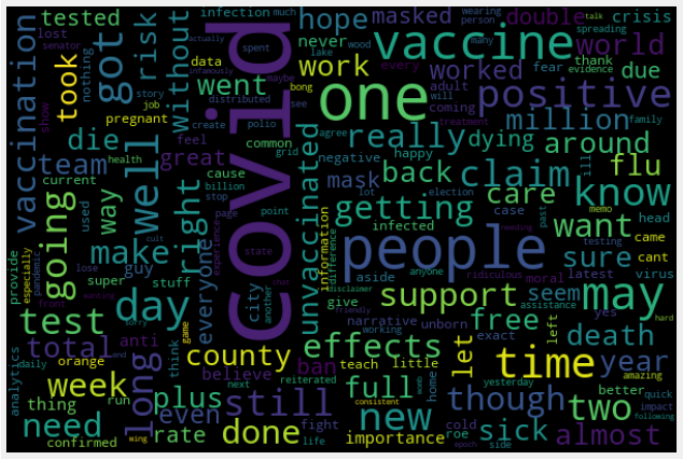


Figure 3: Example worcloud presentation of dataset.

We have taken the tweets from the Twitter and stored them into a dataframe. After saving the dataframe, we had to clean the tweets from the unuseful words and special characters. Main content that we cleaned from the tweets:

-Mentions: the words start with "@"
-URL links: The web links includes "http" at the begining.
-Emojies: The speacial characters, which people use for emotion expressions.
-The Numbers: Numerical values, i.e. date, time, natural number.
-Special Characters: The characters like "!,+,?,/,...".

After we cleaned the tweets from the special charachters, we used the nltk library while cleaning the stopwords. We had to have pure meaningful words to transform it to transaction dataset.

After having our preprocessed tweet dataset, we had to create a dictionary from uniqe words to map on the dataset. We gave numerşcal ids' to every uniqe word and when we mapped that on our dataset, it created our transactions dataset.

1 : ['covid', 'warsaw', 'mask', 'vaccine']
2 : ['cold', 'drink', 'covid', 'sick', 'mask']

transforms to:

1 : [1 3 45 123]
2 : [32 12 1 453 45]

3.3 Dictionary

Here we can have a look at to an example of dictionary.

{"clinical": 0, "used": 1, "machine": 2, "learning": 3, "explore": 4, "electronic": 5...

Figure 4: Dictionary Example.

Now we can run our algorithm over the dataset.

4 Results/Experiments

We will run our algorithm for different size of datasets. We will use 100 tweets, 1000 tweets and 5000 tweets. We will also change the minimum support values and try to observe the changes. We will print the wordcloud of dataset and the bar plot of the output of the algorithm. Bar plot will show transactions and the supports of these transactions.

4.1 Experiment 1

In this experiment we will check the 100 tweets and minimum support of 0.05. This small set also will give us a chance to have a detailed look over all dataset and the algorithm output.

Figure 5: Wordcloud representation of dataset.

Elapsed time for the dataset with 100 tweets.

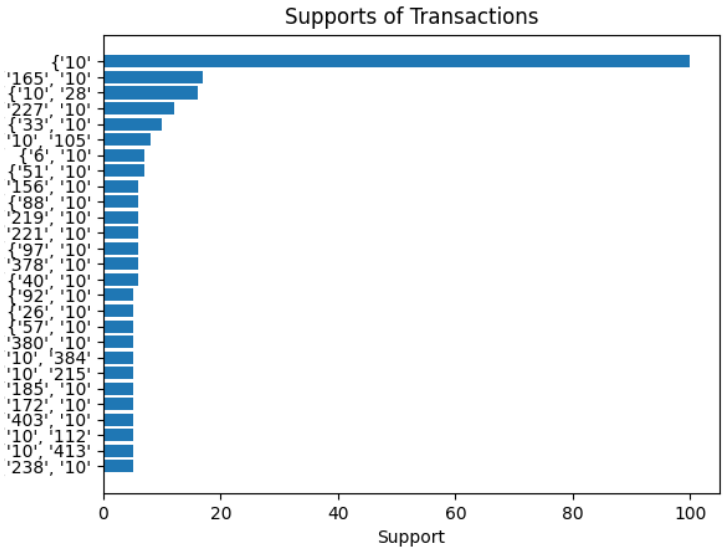


Figure 6: Supports of Transactions.

```
Elapsed time for CHARM Algorithm: 0.1875
```

Figure 7: Elapsed time for 100 tweets.

4.2 Experiment 2

In this experiment we used 1000 tweets and the minimum support of 0.05.

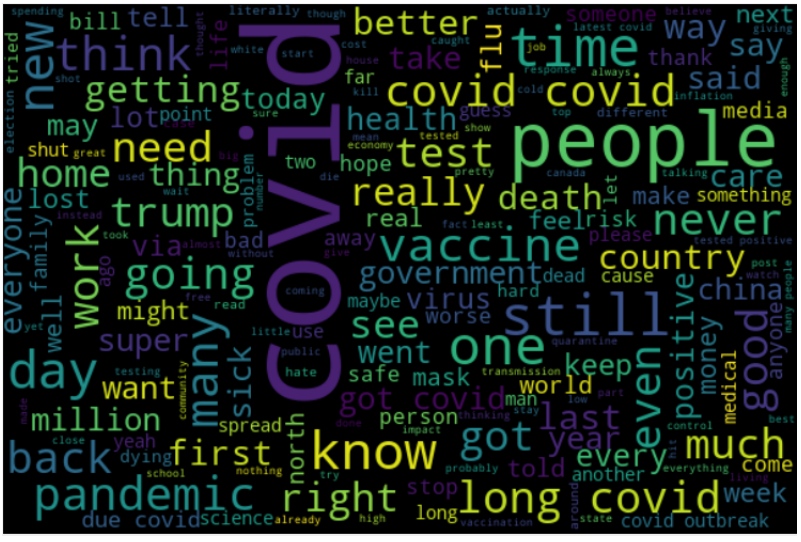


Figure 8: Wordcloud representation of dataset.

In below chart, 5 shows the word 'covid' and the other numbers define the words which appear with 'covid' word in our dataset with minimum 0.05 support. Elapsed time for the dataset with 1000 tweets.

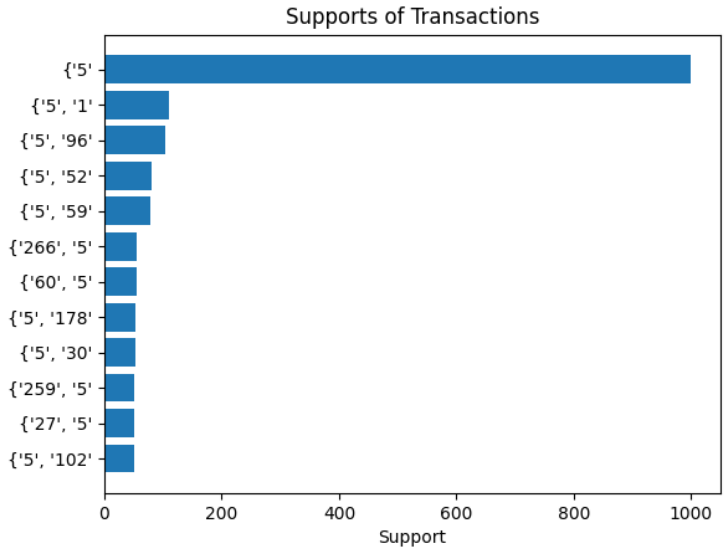


Figure 9: Supports of Transactions.

```
Elapsed time for CHARM Algorithm: 0.21875
```

Figure 10: Elapsed time for 100 tweets.

4.3 Experiment 3

Elapsed time for the dataset with 5000 tweets.

```
Elapsed time for CHARM Algorithm: 0.21875
```

Figure 11: Elapsed time for 100 tweets.

4.4 Experiment 4

In this experiment we used 7000 tweets and the minimum support of 0.05.

4.5 Experiment 5

4.6 Comments

The first thing we can see from the experiment results are the time of executions. As we could predict, depend on the size of the data set, execution time increases.

5 Implementation of the DCI-Closed Algorithm

5.1 DCI-Closed Algorithm

DCI Closed algorithm again a algorithm to find the frequent closed itemsets. Precudure is similar to Charm algorithm but not exactly the same.

Our algorithm will take 3 parameters. closed itemsets and two sets of items. They are named in pseudo code as CLOSED SET, PRE SET and POST SET.

As a result we are expecting to have a non-duplicate closed itemsets which contains the closed set. "In particular, the goal of the procedure is to deeply explore each valid new generator obtained from CLOSED SET by extending it with all the element in POST SET."[Per04]

However it is also very important to create the items for the PRE SET and POST SET. Because our algorithm will be working recursively and itemsets should be suitable for them. As

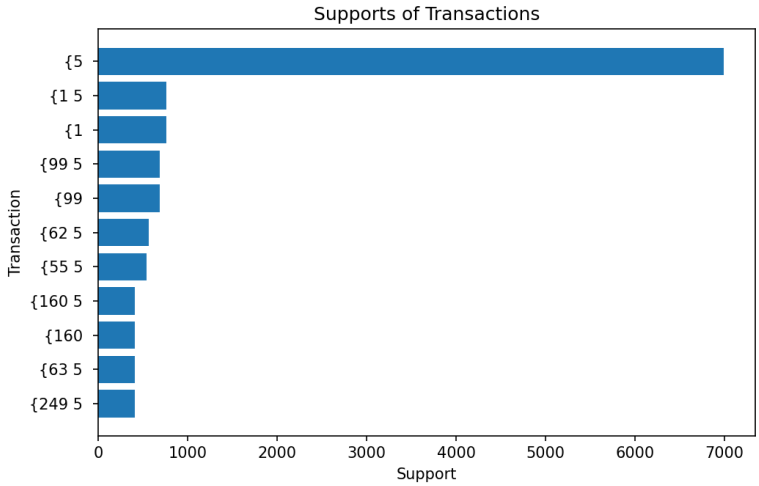


Figure 12: Supports of Transactions.

it is mentioned in our reference document too, while the recursive process, size of the POST SET should monotonically decrase and PRE SET should increase.

```
1: procedure DCI_Closed(CLOSED_SET, PRE_SET, POST_SET)
2:     for all i ∈ POST_SET do                                  ▷ Try to create a new generator
3:         new_gen ← CLOSED_SET ∪ i
4:         if supp(new_gen) ≥ min_supp then                     ▷ new_gen is frequent
5:             if is_dup(new_gen, PRE_SET) = FALSE then         ▷ Duplication check
6:                 CLOSED_SET_New ← new_gen
7:                 POST_SET_New ← ∅
8:                 for all j ∈ POST_SET, i ≺ j  do              ▷ Compute closure of new_gen
9:                     if g(new_gen) ⊆ g(j) then
10:                        CLOSED_SET_New ← CLOSED_SET_New ∪ j
11:                    else
12:                        POST_SET_New ← POST_SET_New ∪ j
13:                    end if
14:                end for
15:                Write out CLOSED_SET_New and its support
16:                DCI_Closed(CLOSED_SET_New, PRE_SET, POST_SET_New)
17:                PRE_SET ← PRE_SET ∪ i
18:            end if
19:        end if
20:    end for
21: end procedure
22:
23:
24: function is_dup(new_gen, PRE_SET)
25:    for all j ∈ PRE_SET do                                   ▷ Duplicate check
26:        if g(new_gen) ⊆ g(j) then
27:            return FALSE                                     ▷ new_gen is not order preserving!!
28:        end if
29:    end for
30:    return TRUE
31: end function
```

Figure 13: Pseudo code for the DCI-Closed Algorithm.[Per04]

5.2 Input Data

We cleaned the data from the unncessary charcters and words. We will use the same input data as we used in the Charm Algorithm.

Example format of the input is as follows:

1 : [covid poland regulation mask vaccine]

6 Results/Experiments

6.1 Experiment 1

In this experiment we used 100 tweets and the minimum support of 0.05 Elapsed time for DCI-Closed Algorithm: 0.03125

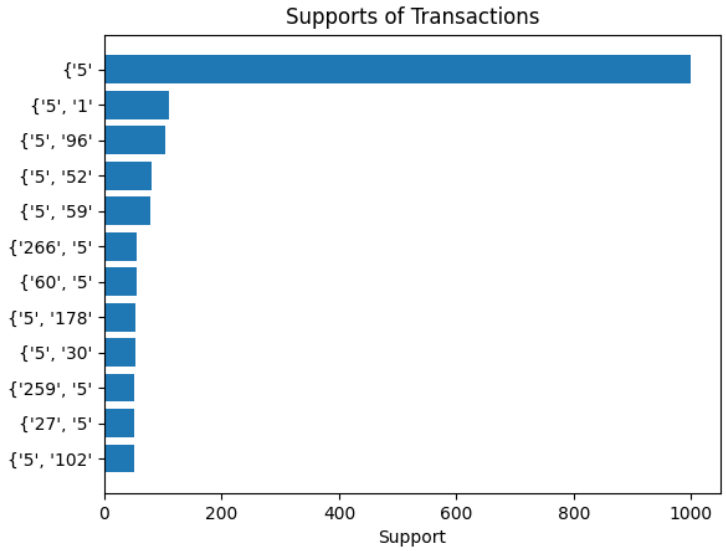


Figure 14: Supports of Transactions.

6.2 Experiment 2

In this experiment we used 1000 tweets and the minimum support of 0.05 Elapsed time for DCI-Closed Algorithm: 0.078125

6.3 Experiment 3

In this experiment we used 5000 tweets and the minimum support of 0.05 Elapsed time for DCI-Closed Algorithm: 0.34375

6.4 Experiment 4

In this experiment we used 7000 tweets and the minimum support of 0.05 Elapsed time for DCI-Closed Algorithm: 0.484375

6.5 Experiment 5

In this experiment we used 10000 tweets and the minimum support of 0.05 Elapsed time for DCI-Closed Algorithm: 0.78125

6.6 Comments

When we have a look at the outputs from the datasets with DCI-Closed algorithm. We can say that most repeating words with the "covid" word are same again.

7 Experiments with Different Minimum Support Values

In this section, we are going to experiment and compare the performance of the two algorithms using the same data-sets. The results are shown in below figures. In addition, we compare the run time for CHARM and DCI-Closed algorithm for different numbers of datasets.

7.1 Experiment 1

Experiment of CHARM and DCI-closed algorithm on different size of datasets respectively, 100, 1000, 5000, 7000, 1000 with minimum support value 0.01

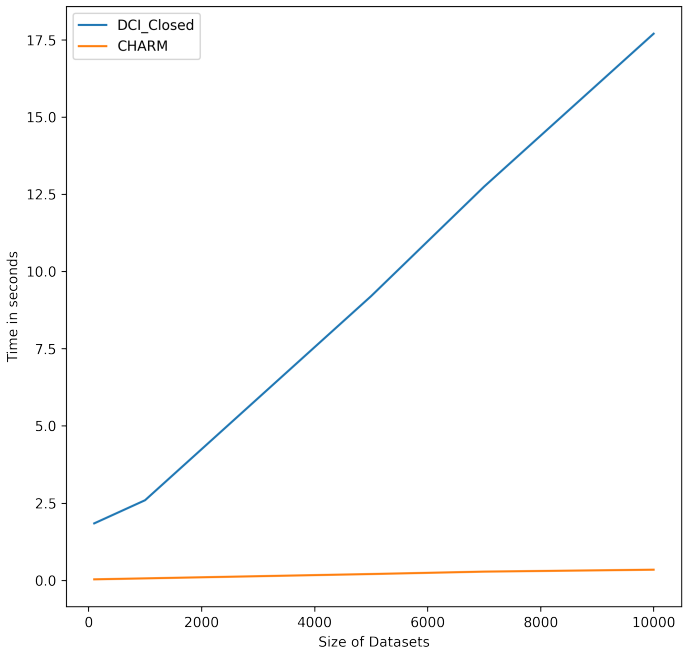


Figure 15: Minimum support value =0.01.

7.2 Experiment 2

Experiment of CHARM and DCI-closed algorithm on different size of datasets respectively, 100, 1000, 5000, 7000, 1000 with minimum support value 0.07

7.3 Experiment 3

Experiment of CHARM and DCI-closed algorithm on different size of datasets respectively, 100, 1000, 5000, 7000, 1000 with minimum support value 0.1

8 Comprasion of CHARM and DCI-Closed Algorithm

8.1 Comprasion of The Codes

The algorithms was coded in python using the dataset collected from twitter. We first implement the CHARM algorithm for mining frequent closed itemsets and in the second phase we implemented DCI-closed algorithm which is a famous algorithm for mining frequent closed itemsets. Our goal in this project was to implement a CHARM algorithm and compare it with a more efficient algorithm which is DCI-closed algorithm in this case.

8.2 Comprasion of the Results

We performed 3 experiments for performance comparison, we used the COVID-19 twitter datasets which we converted into transactional datasets. When comparing CHARM algorithm to DCI algorithm with mininum support value 0.01, it has been observed that the performance time of CHARM algorithm is faster than the DCI-closed algorithm for lower minimum support value. However, as we increase the minimum support value, the performance gaps between CHARM and DCI-close widens. For example experiment 2 and experiment 3, when the minimum support value was increased, DCI-closed algorithm was faster compared to CHARM algorithm. DCI-closed algorithm is faster then CHARM algorithm in execution time, which makes DCI-closed algorithm perform better on higher support values.

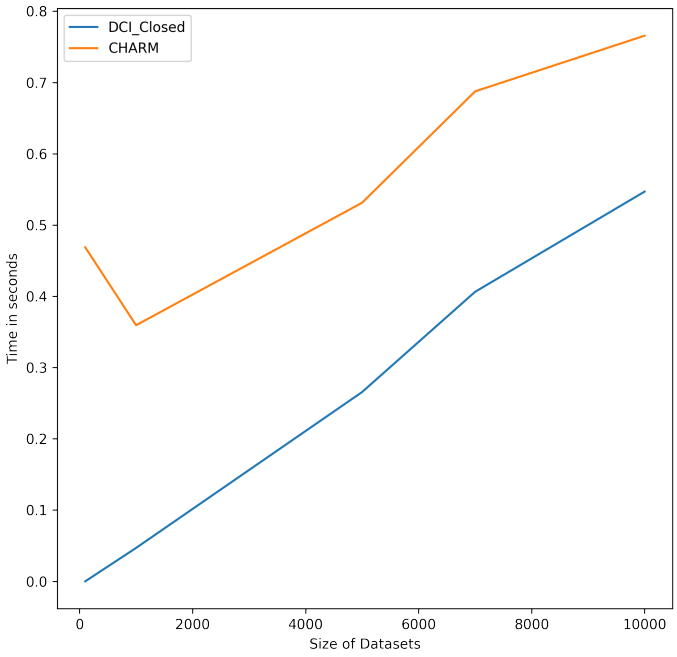


Figure 16: Minimum support value =0.07.

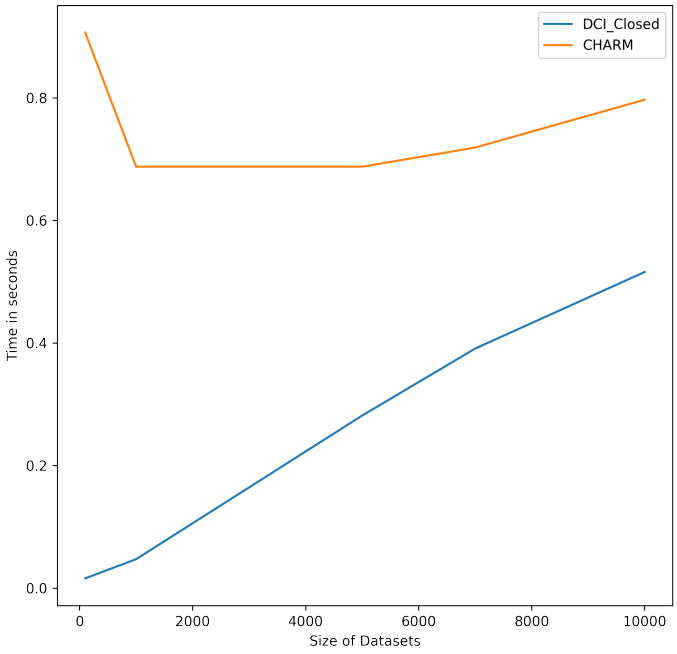


Figure 17: Minimum support value =0.1.

9 Conclusion

In this project, we implemented the CHARM and DCI-closed algorithms using the same Twitter dataset-COVID-19 tweets in two phases. One of the areas for future improvement is to improve the Charm algorithm. CHARM algorithm faces a memory-inefficient challenges since it needs to maintain all closet itemsets in memory to check if an itemset is closed or not. In addition, one should think about a different representation of the sequences for a dataset like we used. Because in our tweets data set most of the users only create one tweet. But when one user creates 10 tweets in the referred dataset, then the representation of all users would be of length 10 with lots of places which do not belong to any content. Therefore, one should develop a more effective data set representation by using our dataset. Furthermore, by reproducing our experiments one should adapt the assumed frequent words which are described in a hypothesis more to the date of the used tweets.

References

[Che15] Xin Ye; Feng Wei; Fan Jiang; Shaoyin Cheng. An optimization to charm algorithm for mining frequent closed itemsets. *IEEE*, 28 December 2015.

[FV22] Philippe Fournier-Viger. Mining frequent closed itemsets using the charm / dcharm algorithms. *SPMF*, 2018-2022.

[Per04] Claudio Lucchese; Salvatore Orlando; Raffaele Perego. Dci closed: a fast and memory efficient algorithm to mine frequent closed itemsets. *spmf*, Pisa; Italy-FIMI 2004.