Assignment #3

Paulvir Dhanda
CMPT459
Simon Fraser University

July 11, 2020

Question 1

Let's begin with letting X be the set all of items. We will first prove that if there exists an element x st. $x \in X$ then the $Sup(x) \geq Sup(Z)$ such that $x \in Z$ and $len(Z) > 1$. We can first let there be an arbitrary transaction t such that $x \in t$. Then if we consider a new arbitrary transaction $t^{'}$ such that $x \in t^{'}$ and we say that the then the $len(t^{'}) > 1$, that is to say the amount of elements in the transaction is in greater than 1. Then by definition:

$$set(t^{'}) \subseteq set(t) \implies count(set(t^{'})) \leq count(set(t)) \iff Sup(Z) \leq Sup(x)$$

With this knowledge, if we let $X^{'}$ be the k-item set such that $x_1, x_2, , , x_k \in X^{'}$, then:

$$min\{Sup(x_1), Sup(x_2), , , Sup(x_k)\} \geq Sup(X^{'})$$

Furthermore, by definition, X' is most frequented k-item set. This implies:

$$Sup(X^{'}) \geq Sup(Y) \implies min\{Sup(x_1), Sup(x_2), , , Sup(x_k)\} \geq Sup(Y) \square$$

Question 2

For this question we start with the transaction table that looks as follows:

```
Sales:
Key           Items
-----------------------------------------------------------
T01            {abc}
T02            {cde}
```

Now, if you let $X = \{a, b, c\}$ and $Y = \{c, d, e\}$ then, $Sup(X) = 1$, $Sup(Y) = 1$, and $X \cap Y \neq \emptyset$: meeting the requirements of the question. However:

$$Sup(X) = Sup(\{a, b, c, d, e\} = \emptyset \neq 1$$

Providing us with a counter example. $\square$

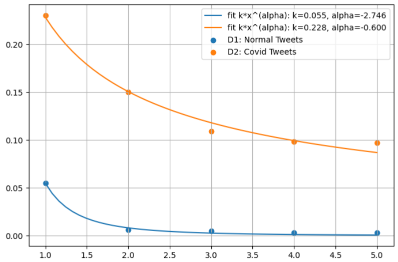

Figure 1: Top support value vs. Length

Question 3

For this question the supplied code is in directory labels with the question number. All code will run as supplied and as instructed by the question answers.

3.1

All code for this question in is the `q3` directory. First, the user will run `generate_tweets.py` to generate two CSV files, `covid.csv and normal.csv`. In these files the tweets will be saves as tokenized python lists. Next, the user runs `restructure.py` to generate 100 most frequented item set of length 1, 2, 3, 4, 5 in their own file respectively. The algorithm used is the FP growth algorithm: `from mlxtend.frequent_patterns import fpgrowth`.

3.2

The top 100 frequent patterns and their supports can be seen in the `normal-length-*.csv` files for the normal tweets data set (D1) and `covid-length-*.csv` for the covid tweets data set (D2). The length of the data sets is signified by the number at the end of each file.

3.3

Using the frequent items CSV files, I generated the graph in figure 1. For a user to do this they must run `analyze.py` (after the previously mentioned steps). The method used to generated the best fit line was `from scipy.optimize import curve_fit`. This method uses a least-squares method to fit the data. Accompanied with the graph, in the legend, are appropriate estimates for the parameters α and k to the power function: $y = kx^{\alpha}$ for their respective graphs. I do agree that the two data sets (D1 and D2) are some transformation of the power law distribution just by inspection. Further analysis can be done to validate using a Chi-Squared goodness of fit test of the parameters using the co variance matrix supplied by the `curve_fit` function.

Question 4

4.1

The method I used for this question starts with the `normal.csv` and `covid.csv` files generated in the previous question. They are supplied in the question directory. First I generate the item set using FP growth once again. This time however, the algorithm return only those values with $Sup_{D1}(X) > 5$ and the length is less than four. Once generated the frequent item are generated for D2 with no restrictions. Then the two data sets are inner joined on the items column. Once joined the $Odd(X)$ is calculated for each item in the joined data.

4.2

The odds are in `q4` directory, in the file: `final_data.csv`. The data was generated using the algorithm above and by running `main.py`.