Justin Zheng
U67912393

DS210 Project Report

Dataset Summary

I was unable to include the file of the dataset because the file was too large.
Link: https://snap.stanford.edu/data/amazon-meta.html

The title of my dataset is "Amazon product co-purchasing network metadata" and was gathered from the Stanford Network Analysis Platform (SNAP). The dataset contains information on 548,552 different products on Amazon. Each item contains sub information regarding its ID number, Amazon Standard Identification Number (ASIN), title, group, salesrank, similar products, categories, and reviews. I will be focusing only on each item's ID number, ASIN, title, group, and similar products. The main focus will be on the similar products aspect of the data. This part of the data contains information regarding the products that are frequently purchased with a specific product. If a product *i* is frequently co-purchased with product *j*, the graph contains an undirected edge from *i* to *j*.

Code Explanation

Data Cleaning and Processing

First, I had to process and clean the data I was working with. Since I only wanted specific parts of each item I decided to store the ID number, ASIN, title, and group as Strings in one vector and the list of edges of Strings in another vector. I did this by first reading and matching the data as a String and then splitting it by the string "Id:" and then collecting everything. This allowed me to split all the information into their individual components first before further processing. I then iterated through every item I collected from reading the data and further split each component by a new line ("\n"). This results in a vector of vectors that contains each part split by index (ID number at 0, ASIN 1, etc.). After splitting, I gathered the components I wanted based on their index and then stored them in a vector of vectors. I also gathered the edges and placed them into a separate vector of strings. I then returned these two vectors for further cleaning. This is done by the open_file() function. I then further cleaned the data by taking slices of the parts I wanted and making the data more usable overall. For example, I had to remove one character from the end of each String because there was a "/r" attached to the end (such as "ASIN: 1890626066/r" becoming "1890626066").

Since the list of edges was still a vector of Strings, I had to convert it into a vector of vectors of integers. To do this, I first had to assign the correct edge to each item. In this dataset, the list of products that are similar to a specific item is sometimes represented by their ASIN and not the id number. This means I first have to iterate through all the edges, find the corresponding

ASIN from the dataset, and then replace the ASIN edge with the ID number. This is done by the assign_edges() function.

Distance Computation and Most/Least Suggested

After converting all ASIN into ID numbers, I calculated the distance from all vertices to all vertices using the breadth-first search algorithm. Using this algorithm, I calculate the distance starting from every vertex and return it as a vector of vectors of Option<u32>. Then I compute the average distance at every vertex by unwrapping all the values of my vector of vectors of Option<u32>, adding them all up, and then dividing by the number of vertices. This is done by the functions compute_distance_bfs() and compute_average_distance(). To find the item that is most frequently bought with other items I iterated through all the edges and compared the lengths of each edge. I keep track of the id and length of the longest edge and the id and length of the shortest edge and then return them. This is done by the functions most_suggested() and least_suggested().

Results

The aim of this project was to determine the distances between each item and the average distance of each item. Additionally, I wanted to know which item had the highest and lowest average distance. Furthermore, I wanted to calculate which items are the most and least frequently bought with other items. Gathering this information could allow us to figure out why certain items are more or less frequently purchased with others compared to others.

Problems

Due to the size of the dataset I was unable to run the program on every vertex and compute distances from all points. As a result I only assigned edges to the first 200,000 items in the dataset. I then calculated the distance using breadth-first search for the first 9,000 items and then calculated the average distance for the first 5,000 items. This is because my program kept crashing before reaching 10,000 when computing the distances due to the size of the dataset, so I decided to cap it at 9,000. Additionally, my program would also crash when calculating the average distance so I capped it at 5,000.

Summary

The item with the greatest average distance is item 2775 with an average distance of 0.00067674316. The item with the least average distance is item 0 with an average distance of 0. The item that is most frequently suggested with other items is item 1 with 6 frequently bought together items. The item that is least frequently suggested with other items is item 150000 with 0 frequently bought together items.

Final Output

```
m_id: 2775 max: 0.00067674316 m_id2: 0 min: 0.0
max: [1, 6] min: [150000, 0] max dist: Some(0.00067674316)
```