Final Project Proposal - TeamMIK

Leading Question

> *Suppose we are the owners of Amazon, and we want to shrink the number of products that we are selling on the platform and we want to do so as optimally as possible. In other words, we want to find products that render the most number of other related purchases given the co-purchasing dataset.*

To answer the leading question, we will:

1) Find top-N nodes with the most co-purchase index[1] using IDDFT,
2) Decide which products to promote on using Tarjan's Algorithm
3) Find the products with high potential using Djisktra's Algorithm

Dataset Acquisition and Processing

We will be using "Amazon product co-purchasing network" from Stanford's network dataset (SNAP). The dataset is stored in a .txt file with each node representing an item and directed edges of node A to node B representing "customer bought item A, customer therefore also bought item B".

Data processing will involve collapsing multigraph into a simple graph, and performing iterative deepening depth first search on each node to determine which item links to the most number of nodes given a certain depth.

Link to dataset: http://snap.stanford.edu/data/amazon0505.html

Graph Traversal

The graph traversal that we need to implement should be iterative deepening depth first traversal(IDDFT) because we take the assumption that customers have a limit of co-purchasing. By using IDDFT we will calculate the number of unique nodes a certain node can traverse on. The number of nodes that can traverse on will be the co-purchase index of the product, which determines the extent of influence of the certain product. We will hence rank the top-N nodes with the most co-purchase index.

Graph Algorithm

Tarjan's Algorithm	Dijsktra's Algorithm

[1] Co-purchasing index is the number of related products purchased together with the current product.

- We want to use Trajan's Algorithm to find strongly connected graphs and strongly connected components - Given our output of graphs with maximum co-purchasing indices, find strongly connected components to determine which items should be offering promotions	- We want to use Dijsktra's Algorithm to find the shortest path between 2 nodes. - Based on how short the path of a node is from the top-N nodes with the highest co-purchase indices, we can determine which products have the most potential.

Timeline

Time	Description
April 09th	Final Project Proposal and Teammate Contract Due
April 16th	First group meeting to set up goals and weekly targets.
	Target: Finish implementing IDDFT
April 23rd	Second meeting
	Target: Finish implementing Tarjan's Algorithm
April 30th	Third meeting
	Target: Finish implementing Dijsktra's Algorithm
May 07th	Fourth meeting
	Target: Using the algorithms to answer topic question
May 08th - May 11th	Prepare for project Deliverables
May 12th	Final Project Deliverables