PHYICS CITATION NETWORK

MGMT59000 CFA Project

Austin Bohlin, Dae Sung Kim

Description of the Dataset

The dataset consists of three files.

1. Paper citation network, from node and to node
2. Time of nodes
3. Paper meta information

This dataset is hosted on the Stanford Network Analysis Platform. The paper citation network dataset is a text file that covers all the citations of the 27770 papers and 352807 edges. The edges represent the direction of citations from *i* citing paper to *j* cited paper. Each node represents specific a paper. If the cited or citing paper is not considered to be in this topic, it is not included in the network. The dataset specifically provides network information about high energy physics theory papers that were published between January 1993 and April 2003.

The time of nodes dataset is a text file information that provides each nodes' time of submission, representing the submission time for a paper. However, we are not interested in this particular dataset.

The paper meta information includes information like author, date of paper submission, report number, journal reference, and brief description about the paper. However, this meta information is not included in the network nodes. It is a separate file for a refence only.

All data in this set is from arXiv, an open-access scholarly paper archive.

Data Pre-Processing

After graphing the data, we found that it was not a connected graph. There were around 370 nodes that were not connected to the largest component. We decided to remove the unconnected components and only use the largest component of the network. In addition, we removed self-referencing edges because a paper cannot self-reference itself. This would allow us to have a meaningful cluster analysis.

Insights from the Dataset

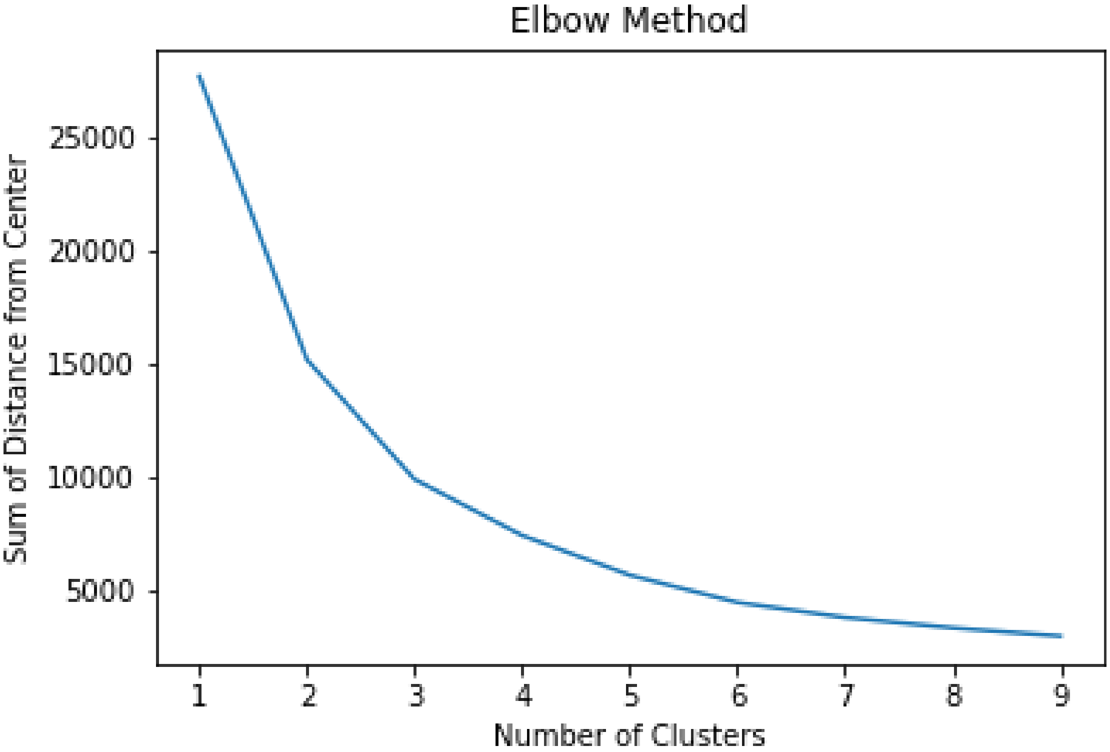
K Means Clustering

Below graph shows the elbow method for finding optimal K clusters in the physics citation network. The graph shows that there are five optimal clusters within the network. In this particular dataset, a cluster could represent a sub-discipline within the high energy physics theory field. The reasoning behind the clusters' representation is that if papers are in a same sub-field, their characteristics must be also similar, which is represented in the distance between the nodes. In addition, papers in a same sub-field will tend to reference each other more than other sub-fields.

However, it is crucial to understand the limitation of K means clustering method. It is sensitive to random initialization and the optimization is based around visual approximation. Therefore, for future analysis, we only assume that there are five sub-fields within the dataset, while the actual value could be different.

To conduct cluster analysis below steps were taken:

1. Converted network graph into vectorized 2-dimensional graph through using node2vec library. This particular method was chosen due to lack of information on weighted edges.
2. Imported sklearn library to loop through increasing number of K's for KMeans function.
3. Based on the cluster assignment, we calculated the sum of the distances within the cluster.
4. Plotted the relationship between the sum of distances within the cluster to number of K through matplotlib library.



It is important to note that many of the interpretation for this project is based upon the cluster analysis and cluster labels. However, due to the nature of K means algorithm, the cluster label changes on each execution of the code. Therefore, all the information is based on the first iteration of the code execution.

Betweenness Centrality

Betweenness centrality analysis quantifies the potential influence of the node. For this analysis, we wanted to gain insight on the most influential paper within the sub-discipline. Here, we assume that if a paper is influential, it will be cited in multiple different papers. So, an important paper will be in multiple different pathways within the sub-network graphs. The results show that the most influential papers were:

- For cluster label 0, Seiberg's *String Theory and Noncommutative Geometry*.
- For cluster label 1, Maldacena's *The Large N Limit of Superconformal Field Theories and Supergravity*.
- For cluster label 2, Phong's *Lectures on Supersymmetric Yang-Mills Theory and Integrable Systems*.
- For cluster label 3, Sen's *Stable Non-BPS States in String Theory*
- For cluster label 4, Maldacena's *Large N Field Theories, String Theory and Gravity*

The author and name of the papers were attained from the paper meta information dataset by comparing the node number.

An interesting note to make from this insight is that author named Maldacena published the most influential paper in two different sub-discipline of high energy physics theory.

Following steps were taken to obtain the information on the most influential paper:

1. Create clusters based on predefined number from the cluster analysis.
2. Assigned cluster labels to each node as an attribute that will be used as a filter.
3. Create a sub-graph based on the cluster labels.
4. Run betweenness centrality function on each cluster to extract the node with the max amount.

Depth First Search (DFS)

After gaining insight on the most influential papers in each sub-discipline, we were also curious about which paper is the most prolific in the high energy physics theory field. In this case, the most prolific paper represents a paper that was continuously developed upon. We assumed that this kind of paper will produce the greatest number of successor nodes. The reason is that in order to progressively develop, there should be constant efforts from other authors to develop upon the paper's idea through citing. Therefore, in network terms, the most prolific node will create child nodes and its child node will create more child nodes and the pattern continues.

DFS successor algorithm provides just that information. It provides the whole tree of "family" in list format. Based on this algorithm, Kol's *Thermal Monopoles* is the most prolific paper in this dataset.

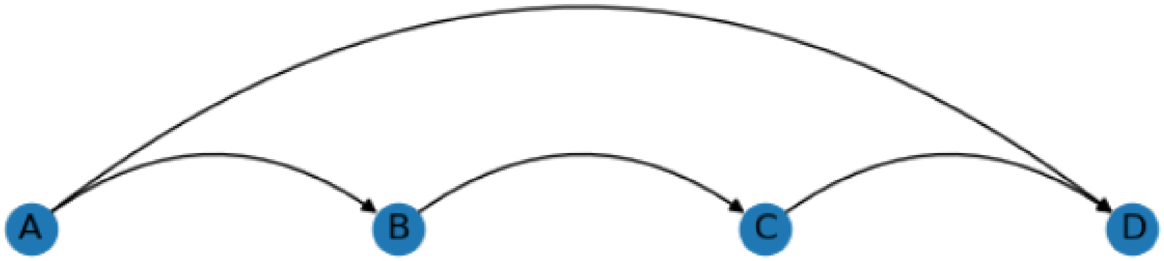
To gain this insight following steps were taken:

1. Initialize maximum number and prolific variable
2. Loop through each cluster to find its prolific paper using DFS algorithm (this method reduced the time of execution significantly).
3. Find the most prolific variable through comparing each cluster's most prolific node

Longest Path

After learning about the prolific paper, we wanted to find the articles that had the longest path away from the center of each cluster. These papers would be the outliers of those clusters. Since these nodes are the outer edges of the cluster, they can't be cited by any other paper, because if they were than that paper would be the outer edge instead. We assume these would either be papers that are on the cutting edge, or the end of the road for that topic.

For directed graphs, networkx has a built-in function to find the maximum distance from 1 node to another. However, after trying this, we learned that our graph had cycles, and this causes max distance function to fail. A cycle in this graph would mean that a paper A would have to cite a paper B that was also citing paper A. We did not expect this to be a possibility as we assumed you can't cite a paper that hasn't been released yet. From here we had 2 options: figure out how to remove the cycles or find a new way to get the maximum non cycled path. We decided to find the length of the Dijkstra path for each node in the cluster from the cluster's central node. With this list we could find the nodes that had the largest shortest distance from the central node, which is like the longest path. This is different in 1 main way; the shortest path will account for a paper citing the parent of a cite chain:



In this chain the shortest path would be 1, but the maximum path would be 4. So, by using the shortest path we lose this chain length, but since we are looking for outlier papers, it might be better that we do. In the picture above D would be the oldest paper as citations go from the citer to the cited, so for Dijkstra to work the way we wanted we needed to reverse the direction of our graph. From this we were able to get a list of nodes for each cluster that had the longest Dijkstra path from each cluster's central node.

Linear Programming

Another question we wanted to find was the minimum number of papers someone would need to read to get an understanding of a particular subfield. For this question, we thought about how this could relate to the graph of connection we had. We assumed that reading a paper should give someone some understanding of each of its neighbors, as they wouldn't be citing them if they weren't somewhat related. So, we framed the problem as:

What is the minimum number of nodes that would let you move from any selected node to any other node in the graph in 1 move.

This problem is best suited to be solved by a linear program. However, we would not want to read papers that had only a few citations, since those could be seen as less impactful. So, we removed all papers that had been cited less than 10 times. We also reselected the largest component from this new graph in case we created any disconnects.

The program straightforward, using a binary decision list for each node S_i and the set of edges E_{ij}, we set the constraints as:

$$\sum_{i \in nodes} S_i * E_{ij} + S_i \geq 1 \quad \forall\, j \in nodes$$

Then the objection function is just:

$$\min \sum_{i \in nodes} S_i$$

From this we were able to obtain a list of papers that connected their clusters together. We found out later that this type of set is called a dominating set, so this LP finds the smallest dominating set of a graph.

Based on the linear programming, we noticed that for clusters 2, there is only one dominating overall book to learn about this particular sub-field. For cluster 3, there are only seven books to understand.