Community Detection with Unnormalized Spectral Clustering

Brandon Tran, Dare Hunt, Andrew Moktarzadeh, Junjie Li

December 2022

1 Abstract

Recent work introduced the vast unfolding of communities in large networks, in which a heuristic methodology not only identifies communities, but also measures the density between nodes in modules that highlight the strength of a subcommunity. It was shown that such methodology can facilitate community detection, and exceed similar community detection algorithms in time complexity. In this paper, we introduce a more simplistic foundation based algorithm, in which communities are identified through the metric of common neighbors. We show how a collection of nodes with a large number of common neighbors have a higher probability of being deemed a community. Our algorithms are first trained from simple randomly generated graphs with ground truths. In training, we derive a 50% threshold for the proportion of common neighbors within nodes to be identified as a community. We apply these algorithms to real world graphs to visualize and represent our results.

2 Introduction

Technological innovations during the past few decades, including the rise of computers, the internet, and social media, have accelerated the size and strength of data networks. When analyzing the data behind various data networks, communities form naturally within them through connections between individual points of data, or nodes. These communities are typically defined by a common variable such as physical location, political alignment, or interest in a public figure. However, as more individual nodes of data are added to the data collection, the number of connections between nodes and the number of communities formed to represent these connections grows exponentially, creating difficult problems to overcome when analyzing the data in a timely manner.

It's important to note that grouping data has always been a problem that we have been trying to solve, and has been done through clustering algorithms, where using multiple attributes for each data entry can be used to find similarities and differences between them to create "clusters". However, the idea of locating and recovering communities is focused specifically on networks as analysis largely relies on a single attribute type - the edge. This is where the planted clique problem is presented: identifying the subset of nodes in a network that have something in common, all determined by edges. The challenge was constructing an algorithm to do so that could perform in efficient time. Methods to achieve this in polynomial time were introduced in 1995 by Luděk Kučera, and improved upon in 1998 by Alon, Krivelevich and Sudakov. Both of which proposed constraints to the size of the planted clique relative to the network, where the planted clique could be found with high probability. More recently, the paper “Computational Lower Bounds for Community Detection on Random Graphs” observes that there are calculations to clearly define three bounds that determine the level of difficulty to retrieve a planted clique: simple, hard, and impossible. This prior research exposes a drawback in graph data, given that some situations cannot be optimized at all.

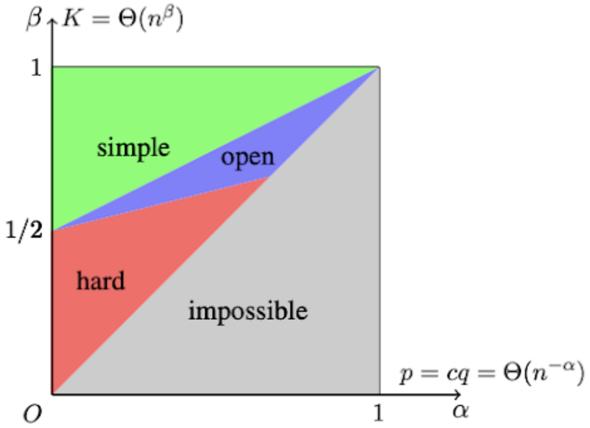

Figure 1: As seen in Hajek 15, “The simple (green), hard (red), impossible (gray) regimes for recovering planted dense subgraphs, and the hardness in the blue regime remains open.”

Aside from the general scope of community detection, we will be exploring the efficacy of the spectral embedding method. All community detection algorithms are designed to obtain the closely related nodes based on some calculations, but do so differently which could result in different predicted communities. Our focus is specifically around the application of the spectral embedding method based on different datasets. One dataset will be randomly generated nodes and relations, with a given ground truth to assess performance. The real-world dataset is one that documents a political article, the web source of that article, and a hyperlink seen on the article. The network is constructed so that each node is an article and edges are shared hyperlinks. Each article labels whether it came from a liberal or conservative source. Ideally, a community would be closely related sources suggesting that they belong to the same political party. Our model will use the spectral embedding algorithm to identify a community made of articles of the same political affiliation. The dataset contains the true affiliation of each article to check how spectral embedding performs in a network.

3 Data

The dataset [1] we will be using in this report is on a directed network of webblogs on U.S. politics. They used a crawler to extract blogposts from hyperlinks and manually labeled the political affiliation of each post. There are only two parties which would be the left wing and right wing, or democrat and republican

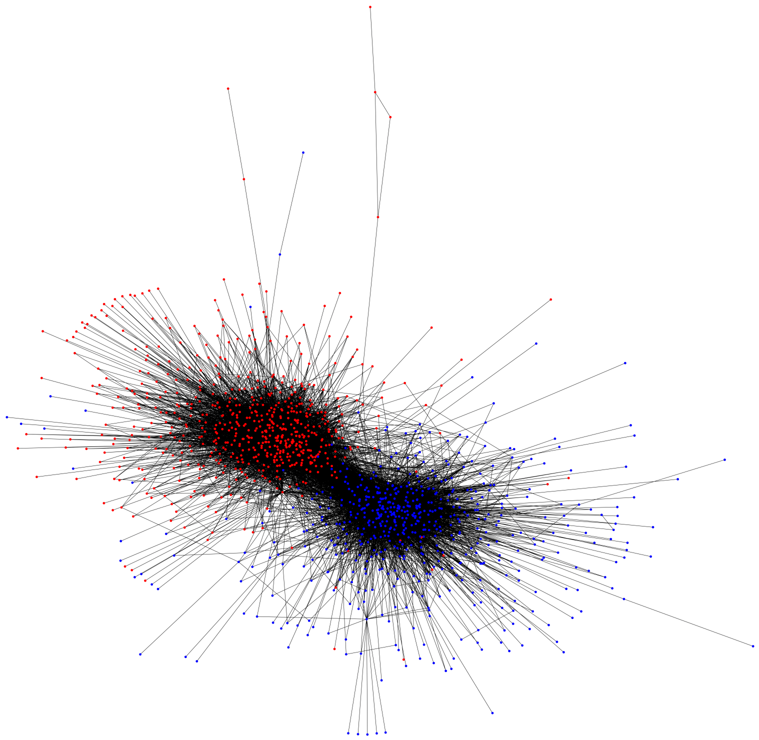

Figure 2: visualization of ground truths for the dataset

4 Method

The method of community detection is to use the spectral embedding algorithm, which refers to a set of operations performed on the graph data to extract communities. Spectral embedding uses dimensionality reduction through the use of eigenvalues so that we are able to find the most closely related nodes within a network. The network must be fully connected, so using the political dataset we must identify the largest First we use our data to find the adjacency matrix, which converts our graph data into n x n matrix where n is the number of nodes, and each number denotes whether there exists an edge between nodes corresponding to the column and row number. Converting our graph to an adjacency matrix allows us to now calculate the graph Laplacian, from which we can get the eigenvalues.

5 Results

Algorithm	Accuracy
Unnormalized Spectral Clustering	.7779

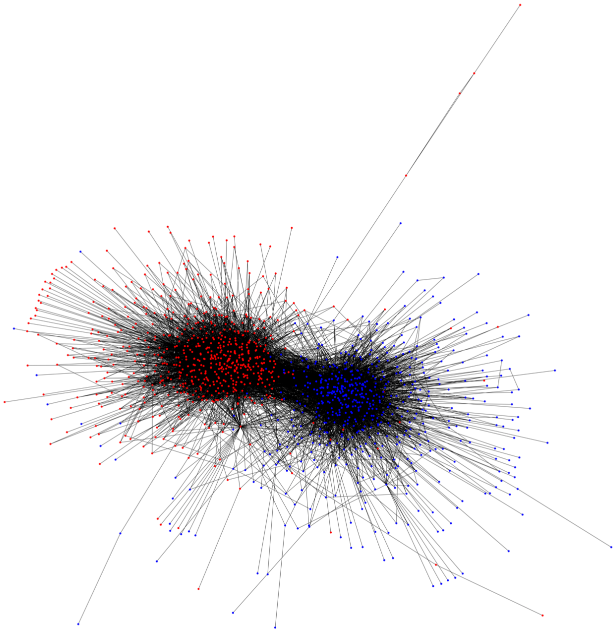

Figure 3: visualization of our predicted results

6 Discussion

Algorithm	Accuracy
Girvan-Newman	.6067
Louvain	.9584
Weighted Threshold	.6067

Table 1: results with different algorithms by another group [3]

When we compared the accuracy with our algorithm with the accuracies of different algorithms explored by another group, we can see that the unnormalized spectral clustering algorithm does better than the Girvan-Newman and Weighted Threshold algorithms but worse than the Louvain algorithm. Why these algorithms have varying performance metrics could be from different variables such as network size, small inner edge probabilities within the communities vs outer, or even the number of communities within the data itself.

Although The accuracy is worse than Louvain's, the spectral algorithm has the benefit of using high dimensional data which was not utilized within this project. Potentially, the unnormalized spectral clustering algorithm can perform the same or better than the Louvain algorithm. We also have to take into account the scalability of the algorithm when using it on a real world dataset. spectral algorithms tend to be slower and use more space when accounting for adjacency matrix calculations which makes it unfeasible to use on very large datasets.

References

[1] Lada A Adamic and Natalie Glance. “The political blogosphere and the 2004 US election: divided they blog”. In: *Proceedings of the 3rd international workshop on Link discovery*. 2005, pp. 36–43.

[2] William Fleshman. *Spectral Clustering*. URL: `https://towardsdatascience.com/spectral-clustering-aba2640c0d5b`. (accessed: 05.12.2022).

[3] Kudsi M et al. “Performance Evaluation of Clustering Algorithms on a Network of Political Blogs”. In: (2022).