Recovery of Communities in Large Networks

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.

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.

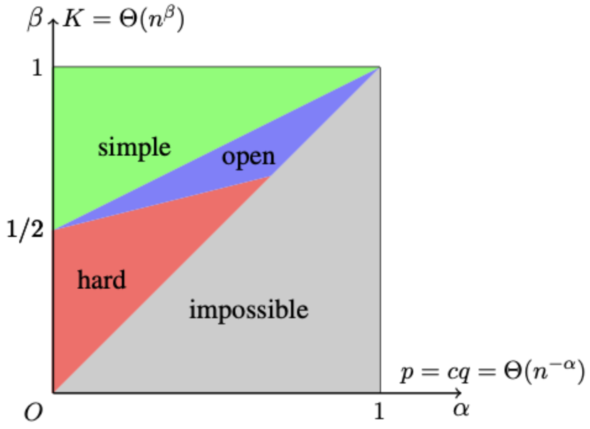

Figure 1: As seen in Hajek 15, “The simple (green), hard (red), impossible (gray) regimes for recovering planted dense subgraphs, and the hardness in the blue regime remains open.”

Data in networks are useful as they can naturally be transcribed in a visual manner that expresses more information than tabular data. However, as data networks in use today grow larger than they’ve ever been, the communities or cliques within them are not necessarily growing in size at the same rate. This, along with the gap in knowledge for recovering communities in large networks, is the problem we plan to explore. While exploring approaches to accomplish this goal, we will first conduct our research on random generated graphs with communities built around ground truths to confirm the validity of our methods. We will then apply these methods to real world data-networks and assess our methods to see if these methods scale to real world scenarios with vast networks and smaller communities. Recovering communities would allow us to summarize and visualize vast amounts of data efficiently, an important goal in today’s data driven age.

Why it’s interesting

Community detection in a network is important and interesting because it can provide useful insights to the structural organization of a network that can be applied to many diverse real-world networks. Since there is a tremendous amount of information stored in each network, if we could detect communities in each network it would provide us with important information and allow the study of the network easier. Furthermore, it could help us improve efficiency for processing and analyzing network data. For example, in social media each user is a node, and the users’ friends whom they interact with form a connection and thus become a network. Social media companies could use community detection algorithms to keep people with common friends,common interests, and background tightly connected, so they could better personalize and establish a more efficient recommendation system and advertisements. By analyzing the existence of communities, we can also learn about the processes of how a network is spreading in various settings. Another useful and important application of community detection is the prediction of missing links and identifying false links in a network because of errors. By applying a community detection algorithm it would allow users to assign and fix these links.

References

http://proceedings.mlr.press/v40/Hajek15.pdf
https://samgrosen.github.io/files/recovering-network-signal.pdf