Community Detection in the Setting of Generalized Random Dot Product Graphs

John Koo

Submitted to the faculty of the Univesity Graduate School
in partial fulfillment of the requirements for the degree
Doctor of Philosophy
in the Department of Statistics,
Indiana University
December 2022

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Approved:

Michael W. Trosset, Ph.D.

Minh Tang, Ph.D.

Julia Fukuyama, Ph.D.

Roni Khardon, Ph.D.

Fangzheng Xie, Ph.D.

December 1, 2022

Acknowledgements

Abstract

Graph and network data, in which samples are represented not as a collection of feature vectors but as relationships between pairs of observations, are increasingly widespread in various fields ranging from sociology to computer vision. One common goal of analyzing graph data is community detection or graph clustering, in which the graph is partitioned into disconnected subgraphs in an unsupervised yet meaningful manner (e.g., by optimizing an objective function or recovering unobserved labels). Because traditional clustering techniques were developed for data that can be represented as vectors, they cannot be applied directly to graphs. In this research, we investigate the use of a family of spectral decomposition based approaches for community detection in block models (random graph models with inherent community structure), first by demonstrating how under the Generalized Random Dot Product Graph framework, all graphs generated by block models can be represented as feature vectors, then applying clustering methods for these feature vector representations, and finally deriving the asymptotic properties of these methods.

Contents

1 Introduction

1.1 Graphs and Representations of Network Data

Graph and network data have become increasingly widespread in various fields including sociology, neuroscience, biostatistics, and computer science. This has resulted in various challenges for researchers who rely on traditional statistical and machine learning methods that are incompatible with graph data and instead assume that the data exist as feature vectors. To illustrate this challenge, consider the typical approach to building a statistical or machine learning model. Data are often represented as an $n \times p$ matrix $X = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$ in which each row $x_i \in \mathbb{R}^p$ is a vector of p features and each column is a set of n feature measurements. An analysis task for these data might be to come up with a classification model $\phi : \mathbb{R}^p \to \{1, 2, ..., K\}$ that uses the numerical values of each feature of a vector. For instance, $\phi(x)$ might first compute the distance from x to one of K points in $\mathbb{R}^p$ and assign x to the label of the nearest point. Examples of this include linear discriminant analysis (in the case of supervised learning) and Lloyd's algorithm or Gaussian mixture models (in the case of unsupervised learning). However, this type of approach is incompatible with data that are represented as graphs, in which each observation is not a vector of numerical features but a set of relationships to other observations. Instead of feature vector $x_i = \begin{bmatrix} x_{i1} & \cdots & x_{ip} \end{bmatrix}^\top$, we observe $a_i = \begin{bmatrix} a_{i1} & \cdots & a_{in} \end{bmatrix}$ where each a_{ij} is object i's relationship to object j.

A more formal description of graph data is as follows: Suppose we observe a network of n objects and pairwise relationships between them. This network is represented by a graph object $G = (V, E)$ with vertex set V, representing the objects, and edge set E, representing the pairwise relationships. The numeric representation of these data is in the form of *affinity matrix* $A \in \mathbb{R}^{n \times n}$ in which each A_{ij} is object i's relationship to object j. We assume that the entries of A represent affinities or similarities, i.e., the higher the value of A_{ij}, the stronger the relationship i has to j. If $A_{ij} = 0$, then i has no direct relationship to j. A is symmetric if it represents an undirected graph in which the relationship from i to j is the same as the relationship from j to i. A is binary, i.e., $A \in \{0, 1\}^{n \times n}$, if it represents an unweighted graph in which edges either exist or don't exist. If A is binary, we call it an *adjacency matrix.*

1.2 Probabilistic Models for Graphs

Given a sample or dataset, a typical analysis tasks are statistical inference, or the estimation of various parameters assuming the data come from a random distribution or process. These estimated parameters are often then used for making predictions or deriving insights about the population. For example, when fitting a Gaussian mixture model, the data are first assumed to come from a mixture of Gaussians. The model fitting process then involves estimating the means and standard deviations of each Gaussian component, along with the mixture weights. The resulting model provides insight into where each mixture component is located, how disperse each component is, and how the data are distributed between the components, as well as a prediction for which mixture a new observation belongs to. In order to perform a similar type of analysis for graphs, we must first define probability distributions from which such data can be sampled.

Within the scope of this dissertation, we focus primarily on unweighted and undirected graphs without self-loops. The adjacency matrix that describes such a graph is binary, symmetric, and hollow. In this setting, a plausible model is to sample each edge independently from a Bernoulli distribution, i.e., $A_{ij} \overset{\text{ind}}{\sim} P_{ij}$ for some $P_{ij} \in [0, 1]$ for each $i < j$ (setting $A_{ji} = A_{ij}$ since A is symmetric, and $A_{ii} = 0$ since A is hollow). Then similar to how the edges are compiled into an adjacency matrix A, the edge probabilities can be compiled into an edge probability matrix $P \in [0, 1]^{n \times n}$. In order to provide structure

1.3 Block Models for Community Detection

1.3.1 The Stochastic Block Model

1.3.2 Generalizations of the Stochastic Block Model: the Degree Corrected Block Model and the Popularity Adjusted Block Model

1.3.3 Review of Likelihood Maximization Approaches to Block Models

2 Random Dot Product Graphs and Generalized Random Dot Product Graphs

2.1 Definitions

2.2 Connecting the Stochastic Block Model to the Generalized Random Dot Product Graph

2.3 Connecting the Degree Corrected Block Model to the Generalized Random Dot Product Graph

3 Popularity Adjusted Block Models are Generalized Random Dot Product Graphs

4 Generalized Random Dot Product Graphs with Community Structure

Appendix A