CS769: Course Project

Optimization in Machine Learning

April 26, 2023

Clustering Partially Observed Graphs via Convex Optimization

Instructor: Prof Ganesh Ramakrishnan

GROUP MEMBERS
Sakshi Heda, 200070071
Priyanshi Gupta, 200070061

Contents

1 Introduction

This paper seeks to study clusters in a partially observed unweighted graph, where for some node pairs, an edge can be identified and for other pairs, uncertainty exists. The goal is to group the nodes into distinct clusters that are densely connected to each other and sparsely connected between. To do this, we can use different approaches from basic graph-based clustering algorithms to more complex machine learning-based methods. We focus on the graph-based clustering method which uses the graph's structure to detect clusters. This method can identify clusters with high accuracy and give insights into the graph's structure.

Partial observations of graphs are encountered in a variety of contexts. For instance, in social networking sites, an edge will be present between two users if they accept or decline each other as friends. Conversely, for other pairs of users, there is simply no friendship information available, thus representing an unobserved case. In general, partial observations occur when obtaining similarity data is difficult or costly (e.g. requiring human involvement). In these settings, it is quite common for the majority of pairs to be unobserved, which is the scenario of our interest.

2 Problem Statement

Given a graph represented as an adjacency matrix, where $a_{ij} = 1$ if there exists an edge between node i and node j, we aim to form disjoint clusters with dense connections within clusters and sparse connections among clusters. Initially, we address the problem for fully observable graphs and then extend the approach to partially observed graphs. To achieve this, we employ a technique to sample full graphs and perform clustering on the sampled graph, aiming to approximate the fully observed case as closely as possible. The adjacency matrix is of size $n \times n$, where n is the number of nodes in the graph.

3 Methodology

We consider the following disagreement conditions:

1. Node pairs are in different clusters but have edge between them
2. Node pairs are in same cluster but no edge between them

Our aim is to minimize the number of disagreements in order to achieve optimal clustering. When we try to solve this problem systematically, it is NP-hard. Therefore, we tried to use the following method of convex optimization to do matrix splitting. We split the given matrix into a block diagonal (low-rank) matrix and a sparse matrix. The block diagonal matrix represents the ideal case of perfect clustering, where there are no disagreements. However, since this is not always possible, we use the sparse matrix to separate the points of disagreement and achieve optimal clustering. We can reorder the rows and column of the matrix to represent it in the form of perfect block diagonal matrix Given a matrix D we seek to split it into a low rank matrix A and a sparse matrix E by minimizing the objective

function $\min_{A,E} \lambda||A||_1 + ||K||_*$ subject to $A + K = D + I$.Here, I is added to D as the diagonal elements of D are zero, as there can be no edge between the same node. But, the diagonal element must be 1 for the matrix A to be low-rank (block diagonal). To solve this equation, we use the method of Augmented Lagrange Multipliers (ALM) as proposed in [1]. Specifically, we employ the Inexact ALM for Robust PCA to handle the problem of noise in the data.

Sampling

We propose a scalable and provable randomized framework for clustering large partially observed graphs generated from the stochastic block model in an unsupervised manner. Firstly, we construct a reduced graph sketch using random sampling and apply clustering to a sub-matrix of the graph's adjacency matrix associated with the sketch. The goal is to reduce computational complexity and improve efficiency while maintaining the accuracy of the clustering. Next, we infer the clusters of the full graph based on the clusters extracted from the sketch using a correlation-based retrieval step. This step enables us to determine the clusters of the full graph accurately and efficiently. To further improve computational efficiency, we explore different sampling techniques. Firstly, we demonstrate that uniform random node sampling can significantly reduce computational complexity without compromising the accuracy of the clustering. Secondly, we present a random degree-based node sampling algorithm to improve the quality of the clustering results.

4 Code

Inexact ALM for low rank matrix recovery

```
def InexactALM(D, lamb, mu, rho):
    Y0 = np.zeros(D.shape)
    rpca = R_pca(D, Y0, mu, lamb, rho)
    A, E = rpca.fit()
    return (A, E)
```

The rpca function (which is a part if class RPCA) is as follows

```
    def fit(self, tol=None, max_iter=1000, iter_print=100):
        iter = 0
        err = np.Inf
        Ek = self.E
        Yk = self.Y
        Ak = np.zeros(self.D.shape)
        while iter < max_iter:
            Ak = self.svd_threshold(self.D - Ek + self.mu_inv * Yk, self.
   mu_inv)
            Ek = self.shrink(self.D - Ak + (self.mu_inv * Yk), self.mu_inv
    * self.lmbda)
            Yk = Yk + self.mu * (self.D - Ak - Ek)
            err = self.norm_p(np.abs(self.D - Ak - Ek), 2)
            self.mu *= self.rho
            iter += 1
```

```
        self.A = Ak
        self.E = Ek
        return Ak, Ek
```

Clustering

```
    def Clustering(D):
    n = len(D)
    A = np.zeros(D.shape)
    lamb = 1/(32*np.sqrt(n))
    i = 0
    while(i < 10):
        A, E = InexactALM(D+np.identity(n), lamb, mu=1.5, rho=1.3)
        if(A.trace() > n):
            lamb /= 2
        elif(A.trace() < n):
            lamb *= 1
        else:
            pass
        i += 1
    A = np.around(A)
    E = np.around(E)
    A = A.astype(int)
    E = E.astype(int)
```

5 Results

Random symmetric matrix was generated and following is the result:

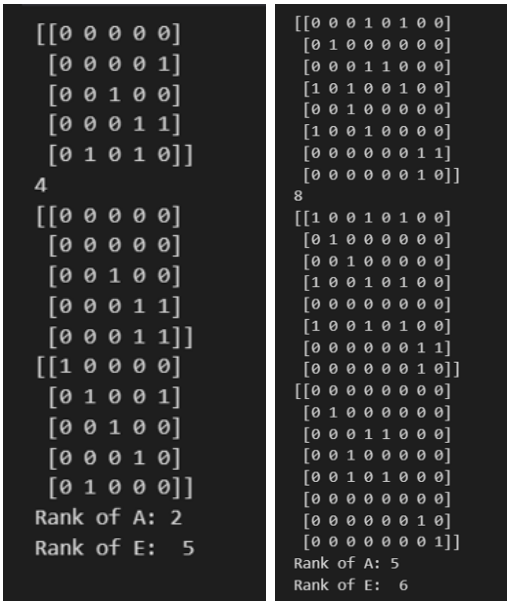

Figure 1: Results for matrices of different sizes

6 Observations and Conclusions

We implemented [2] for fully observed matrix and currently working on different sampling strategies as given in [3]

References

[1] O. Kuybeda, G. A. Frank, A. Bartesaghi, M. Borgnia, S. Subramaniam, and G. Sapiro, "A collaborative framework for 3d alignment and classification of heterogeneous subvolumes in cryo-electron tomography," *Journal of Structural Biology*, vol. 181, pp. 116–127, feb 2013.

[2] Y. Chen, A. Jalali, S. Sanghavi, and H. Xu, "Clustering partially observed graphs via convex optimization," 2014.

[3] M. Rahmani, A. Beckus, A. Karimian, and G. K. Atia, "Scalable and robust community detection with randomized sketching," *IEEE Transactions on Signal Processing*, vol. 68, pp. 962–977, 2020.

[4] A. Beckus and G. K. Atia, "Scalable community detection in the heterogeneous stochastic block model," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, 2019.