**Network Classification with Applications to Brain Connectomics**

Authors: Jesus D. Arroyo-Relion, Daniel Kessler, Elizaveta Levina, and Stephan F. Taylor

Published in: Annals of Applied Statistics

Abstract

Analysis of a single network is simple, but a sample of networks is much more difficult. An example of a sample of networks would be brain networks which represent functional connectivity. The goal of the authors is to use individual edge information as well as the network structure of the data to determine difference in brain connectivity patterns.

Proposed method: graph classification that uses edge weights as predictors while also taking into account the network by introducing penalties. Implemented via convex optimization.

Data set: two fMRI studies of schizophrenia

1. Introduction

Example setting is a sample of networks from multiple populations of interest (mentally ill and healthy control). Each unit (a patient) is represented by their network and the nodes (brain areas of interest) are labeled and shared through all the networks. The paper will aim to determine what the rules for predicting the class of a given network are and how to interpret these rules. They will achieve this by developing a high dimensional network classifier.

2. Framework for node selection in graph classification 🡪 the classifier and structured penalties

2.1 Penalized graph classification approach.

Assumptions are undirected graphs that contain no self-loops as these match neuroimaging standards. The goal is to predict the class label (Y) from adjacency matrix (A) and the coefficients (B) are estimated from training data by minimizing an objective consisting of a loss function plus a penalty. This set up is ideal for medium to large brain networks. The loss function used is the logistic loss function.

2.2 Selecting nodes and edges through group lasso.

The group lasso penalty is designed to eliminate a group of variables simultaneously. They penalize the number of active nodes by treating all edges connected to one node as a group. Removing a row of B is then the same as removing a node.

3. Optimization Algorithm

The algorithm uses two approaches to optimization, proximal algorithms and alternating direction method of multipliers (ADMM). The groups overlap which causes the optimization difficulty. The paper solves this by solving the proximal operator for the penalty directly using the ADMM method. This could give a more accurate sparsity pattern and allows for additional penalties. Additional penalties are helpful because they can incorporate more information (ex. spatial location).