CSCI 4961/6961: Homework 5

Assigned Wednesday December 2 2020. Due by 11:59pm Wednesday December 9 2020.

Create a Jupyter notebook for this assignment, and use Python 3. Write documented, readable and clear code (e.g. use reasonable variable names). Submit this notebook along with a pdf in which the answers to each question are legible, and clearly labeled. You will be graded primarily based on the solutions and answers in the pdf, but the notebook must be runnable. Name the files RPIid\_pr1.ipynb and RPIid\_pr1.pdf, where RPIid is your six letter RPI id.

In this homework, you will ...

1. Download the file  
   <https://github.com/Libsmj/CSCI-4961-GoemansWilliamson/blob/main/dataset.ipynb>  
   rename it to RPIid\_ pr1.ipynb, and use it as the starting point of your solution. The code should download the dataset and parse the data into an adjacency matrix.  
   Answer the following problems in your pdf in full sentences and provide the plots asked for in the pdf. The TA will not look in your Python code for answers that are not present in the pdf.  
   - Install the SDP solvers from <https://www.cvxpy.org/install/index.html> and <http://cvxopt.org/install/index.html>  
   - Create a symmetric matrix variable, X, of the same size of and set the following constraints:  
    -   
    -   
   -Solve the SDP Problem by minimizing , where is an adjacency matrix for representation for the smallest graph in Facebook dataset. Record the amount of time it took for the SDP solver to run.  
   -Solve for the Maxcut by finding y, the sign of the inner product between the Cholesky Factorization of X and a random variable from 0 to 1. The size of the cut is given by  
   where, e is the column vector whose entries are all 1’s. Record the size of the cut.  
   -Find the exact value of the max cut by using …  
   -Repeat this process for two other graphs from the Facebook datasets of your choosing. Modify the parser cell to open a different “.edges” file.
2. [For CSCI6961 students.] For the Goemans-Williamson Maxcut Approximation Algorithm, we required that our graph G has non-negative edge weights. This was so that the inequality

would not be reversed. Prove the inequality for a graph with negative edge weights is  
where  
Hint: can be rewritten as

by splitting W into its positive and negative weights