CSCI 4961/6961: Project

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.

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:
     - $X \succcurlyeq 0$
     - $X_{ii} = 1$

   -Solve the SDP Problem by minimizing $Tr(AX)$, where A is an adjacency matrix for representation for the smallest graph in Facebook dataset.
   -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
   $$\frac{e^T Ae - y^T Ay}{4}$$
   where, e is the column vector whose entries are all 1's. Record the size of the cut.
   -Compare the size of the Maxcut to the exact Maxcut for the following facebook datasets:
   # 3980.edges - graph size: (52, 52) - 83
   # 698.edges - graph size: (61, 61) - 144
   # 414.edges - graph size: (150, 150) - 889
   # 686.edges - graph size: (168, 168) - 881
   # 348.edges - graph size: (224, 224) - 1654
   # 0.edges - graph size: (333, 333) - 1339
   # 3437.edges - graph size: (534, 534) - 2515
   # 1912.edges - graph size: (747, 747) - 15175
   # 1684.edges - graph size: (786, 786) - 7285
   Check that the size of your max cut is within the approximation guaranteed by the Goeman-Williamson Algorithm.
   -Repeat this process for two other graphs from the Facebook datasets of your choosing, except 107.edges. 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

$$\mathbb{E}[W] \geq \alpha \sum_{i,j} w_{ij} \frac{1 - < u_i, u_j >}{2}$$

would not be reversed. Prove the inequality for a graph with negative edge weights is

$$\mathbb{E}[W] - W_{-} \geq \alpha \left\{ \sum_{i<j} w_{ij} \frac{1 - < u_i, u_j >}{2} - W_{-} \right\}$$

where

$$W_{-} = \sum_{i<j} w_{ij}^{-} \ , where \ w_{ij}^{-} = \min(0, w_{ij}) \ .$$

Hint: $\mathbb{E}[W] - W_{-}$ can be rewritten as

$$\mathbb{E}\left[\sum_{i,j} w_{ij} \left(\frac{1 - u_i u_j}{2}\right)\right]_{w_{ij}>0} + \mathbb{E}\left[\sum_{i,j} |w_{ij}| \left(\frac{u_i u_j}{2}\right)\right]_{w_{ij}<0}$$

by splitting W into its positive and negative weights