DS-GA HW2

Nischal Mainal

February 2018

1 Max-cut

The algorithm for max-cut that I ended up using is a mix of SDP and greedy, depending on the case it maximized the objective better. And the upper bound has been produced by maximizing the relaxed SDP and using the maximum of the relaxed objective as upper bound, except for the case of graph 5 and graph 6, where they are assigned a loose trivial bound.

1. Graph 1:
   Upper bound = 4
   Optimized objective value = 4

2. Graph 2
   upper bound = 68.5
   optimized objective value = 65

3. Graph 3
   upper bound = 380.35
   optimized objective value = 363

4. Graph 4
   upper bound = 104.54
   optimized objective value = 101

5. Graph 5
   upper bound = 62368
   optimized objective value = 33108

6. Graph 6
   upper bound = 1282
   optimized objective value = 943

2 Conductance

We start by noting that maximizing the conductance is essentially the same as minimizing the Cheeger constant, which itself is upper bound by normalized cut. I started by finding a solution for normalized cut which is an eigenvector problem (as has been described in the lecture notes).I used the solution obtained by minimizing the minimizing the normalized cut as my solution. And the upper bound was found by use of the Cheeger inequality since the objective at hand is essentially the reciprocal of the cheeger cut. The upper bound turns out to be $\frac{2}{second-largest-eigenvalue-of-normalized-laplacian}$

1. Graph 1:
   Upper bound = 2
   Optimized objective value = 1

2. Graph 2
   upper bound = 1.48
   optimized objective value = 0.189

3. Graph 3
   upper bound = 1.57
   optimized objective value = 0.076

4. Graph 4
   upper bound = 1.18
   optimized objective value = 0.4

5. Graph 5
   upper bound = 1.84
   optimized objective value = 0.007

6. Graph 6
   upper bound = 1.12
   optimized objective value = 0.37

3 Clique

I used the Bron–Kerbosch algorithm with recursive back-tracking that numerates all the maximal cliques. After listing all the maximal cliques, the cliques with largest no. of elements was identified as the optimal solution. Since this is an exhaustive process, the upper bound is the same as optimized value.

1. Graph 1:
   Upper bound = 2
   Optimized objective value = 2

2. Graph 2
   upper bound = 6
   optimized objective value = 6

3. Graph 3
   upper bound = 8
   optimized objective value = 8

4. Graph 4
   upper bound = 3
   optimized objective value = 3

5. Graph 5
   upper bound = 13
   optimized objective value = 13

6. Graph 6
   upper bound = 3
   optimized objective value = 3

4 Cluster

Firstly, I have assigned all upper bound to be infinity, which is actually reached in cases such as Graph 1, where we can find completely isolated cluster. Since there is no efficient algorithm to determine that for larger graphs, I have assigned inf to be the upper bound. As for the cluster, my output is simply the set of nodes with size as V/5 with nodes (V/5) of the nodes with smallest degree in the graph.

1. Graph 1:
   Upper bound = ∞
   Optimized objective value = ∞

2. Graph 2
   upper bound = ∞
   optimized objective value =0.033

3. Graph 3
   upper bound = ∞
   optimized objective value = 0.0057

4. Graph 4
   upper bound = ∞
   optimized objective value = 0.05

5. Graph 5
   upper bound = ∞
   optimized objective value = 5.25900604786e-05

6. Graph 6
   upper bound = ∞
   optimized objective value = 0.005

5 Dense

This algorithm starts with the whole graph and checks if the condition for densness is met. If not, it greedily, deletes the node with smallest number of edges such that of all choices, this choice has best chance of meeting the criteria of denseness. I am still unsure, but believe that this algorithm leads to optimal solution in all cases, and as such I have chosen the upper bound to be same as the optimized objective value.

1. Graph 1:
   Upper bound = 4
   Optimized objective value = 4
2. Graph 2
   upper bound = 20
   optimized objective value = 20
3. Graph 3
   upper bound = 49
   optimized objective value = 49
4. Graph 4
   upper bound = 15
   optimized objective value = 15
5. Graph 5
   upper bound = 499
   optimized objective value = 499
6. Graph 6
   upper bound = 21
   optimized objective value = 21