Finding the Maximally-Connected k-subgraph of a Graph

Arezou Keshavarz and Brendan O'Donoghue
CME305, Stanford University

1 Problem Description

Consider a graph $G(V, E)$ with vertex set V and edge set E. We seek to find the set of k vertices that would construct a subgraph $\tilde{G}(\tilde{V}, \tilde{E})$ such that $\tilde{G}$ is the k-densest subgraph of G. Define $W \in \Re^{|V| \times |V|}$ as the connectivity matrix. Thus $W(i, j)$ represents the weight of the edge between node i and node j, and is 0 if there is no edge between node i and node j. The density of a graph is defined to be the sum of its weights. The goal of this project is to find a computationally tractable way to find (exactly or approximately) the set of k vertices $\tilde{V}$, and all of their corresponding edges $\tilde{E}$, such that the density of the subgraph $\tilde{G}$ is maximized.

2 Literature Review

The problem of finding the densest k-subgraph is shown to be NP-hard (as it is a reduction of the maximum clique problem [FS97], [ST05], [SW99], [FPK01]). [FS97] shows that while it is NP-hard to solve the densest k-subgraph problem, it is not NP-hard to distinguish between graphs that contain a k-clique and graphs in which the densest k-vertex subgraph has only $(1-\epsilon)\binom{k}{2}$ edges. With their SDP formulation, however, they show that the optimal solution need not detect the k-clique for $k \approx n^{1/3}$. The work of [FS97] and [SW99] also use semidefinite programming to solve the densest k-subgraph problem, whereas [FPK01], [BCC$^+$10] use combinatorial approximation algorithm. Another closely related problem is that of finding the k-subgraph with maximum algebraic connectivity (which is defined to be the second smallest eigenvalue of the Laplacian). For a review on using convex optimization of graph Laplacian eigenvalues, see [Boy06]. See [dA07] and [GB06b] for more discussions on algebraic connectivity. The work of [GB06a] considers the problem of growing a well-connected graph by adding edges to the graph such that the connectivity is maximally increased. They choose the edges based on the difference in the entries of the Fiedler vector. The work of [Hal70] considers the problem of placing k nodes in a space such that the total pairwise inter-node distance is minimized. It is shown that choosing the Fiedler vector can result in an optimal placement strategy. Furthermore, it shows that the entries of the Fiedler vector can be used to cluster the data points into groups.

We also apply our method to Erdös-Renyi random graphs with a hidden k-clique. If all edge weights are equal, the densest k-subgraph is the clique. In this special case, we see that the problem of finding the densest k-subgraph and the maximum clique are equivalent. Some of the literature on the maximum clique problem can be found at [CP90], [AKS98], [APR99]. The work of [AKS98] presents a polynomial time algorithm that finds the clique of size larger than $n^{1/2}$ almost surely. However, if the clique size is of size $n^{1/2-\epsilon}$, then the problem of finding the maximum clique is open. For random graphs, the maximum clique is likely to be of size very close to $2\log(n)$ [Jer06], therefore the problem of finding the maximum clique in a random graph is still an open problem.

3 Approach

In order to find the k-densest subgraph, we need to solve the following optimization problem:

We can define this problem more concisely as follows:

$$\begin{array}{ll} \text{maximize} & \sum_i \sum_j X_{ij} W_{ij} \\ \text{such that} & X = xx^T, \\ & \mathbf{1}^T x = k, \\ & x \in \{0,1\}^{|V|}. \end{array} \tag{1}$$

However, this is a combinatorial optimization problem. We try to solve this problem by relaxing it in multiple steps to a convex optimization problem so that we can make it computationally tractable.

The most obvious relaxation would be to relax the equality constraints to inequality constraints, and relax the integer constraints to bound constraints. Doing so would result in the following optimization problem:

$$\begin{array}{ll} \text{maximize} & \mathbf{Tr}(WX)/2 \\ \text{such that} & X \preceq xx^T, \\ & \mathbf{1}^T x = k, \\ & 0 \leq x \leq 1, \end{array} \tag{2}$$

The above optimization problem provides a lower bound on the optimal solution. To tighten this lower bound, we add some constraints to restrict the set of allowable X matrices. The new optimization problem is defined as:

$$
\begin{aligned}
\text{maximize} \quad & \mathbf{Tr}(WX)/2 \\
\text{such that} \quad & X \preceq xx^T, \\
& \mathbf{1}^T x = k, \\
& 0 \leq x \leq 1, \\
& X_{i,j} \leq \min(x_i, x_j), \\
& \mathbf{1}^T \, \mathbf{diag}(X) = k, \\
& \mathbf{1}^T X \mathbf{1} = k^2.
\end{aligned}
\tag{3}
$$

Here we try to interpret the added constraints:

- $X_{i,j} \leq \min(x_i, x_j)$: First, note that for a binary x_i, x_j, $\min(x_i, x_j) = x_i x_j$. Therefore, this constraint is also a relaxation on $X = xx^T$.
- $\mathbf{1}^T \, \mathbf{diag}(X) = k$: Note that $x_{ii} = 1$ if and only if node i is chosen to be included in the k-subgraph. Since we need to choose a total of k nodes in the subgraph, we require that the sum along the diagonal of X be equal to k.
- $\mathbf{1}^T X \mathbf{1} = k^2$: Note that in the original problem, we want to have $X = xx^T$; therefore, we would have $\mathbf{1}^T xx^T \mathbf{1} = (x^T \mathbf{1})^2 = k^2$. We enforce this constraint in the relaxed problem as well to tighten the lower bound.

4 Optimality in Erdös-Renyi Graphs Augmented with a k-Clique

Recall that the optimization problem in 3 provides a lower bound on the optimal solution of 1, and rounding this solution will result in an upper bound on 1. We now consider the case when $G(V, E)$ is a random $G(n, p)$ graph, augmented with a clique of size k. Then, our simulations indicate that solving the relaxed problem of 3 yields the optimal solution of 1., *i.e.*, the lower bound is tight. That is, solving the problem of 3 results in a binary solution, thus a zero sub-optimality gap. We have tested our method on multiple graphs. We varied n from 10 to 120 in increments of 10, and for each graph size we generated 10 random graphs. We tested three values for p: 1.1, 1.8, and 2.5. For each graph we augmented a clique of size $n^{0.45}$ inside the $G(n, p)$. The results are shown in Table 1.

5 Polishing

One way to improve the selection is to make one-step swaps of nodes such that the total density is increased. More specifically, we swap nodes i and j if doing so increases $x^T W x$, *i.e.*, if $x^T W x - (x - e_i + e_j)^T W (x - e_i + e_j) < 0$. Note that this is a very cheap computation: we can calculate and store Wx beforehand, and then at each step calculate $\text{sign}((Wx)_i - (Wx)_j - W_{ij})$, which is an $\mathcal{O}(1)$ operation. We go over all node pairs to determine if switching any of them will result in a better selection, and repeat the procedure until convergence (which

Table 1: Simulation results for Erdös-Renyi Graphs Augmented with a k-Clique. Each entry denotes the average sub-optimality gap for the given n and p.

	n											
	10	20	30	40	50	60	70	80	90	100	110	120
	1.1	0	0	0	0	0	0	0	0	0	0	0
p	1.8											
	2.5											

is when there are no changes in one complete pass through all node pairs). Therefore the polishing step is $\mathcal{O}(n^2)$, which is cheap compared to the $\mathcal{O}(n^6)$ convex optimization problem.

References

[AKS98] N Alon, M Krivelevich, and B Sudakov. Finding a large hidden clique in a random graph. *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, page 598, 1998.

[APR99] J Abello, PM Pardalos, and MGC Resende. On maximum clique problems in very large graphs. *External memory algorithms*, 50:119–130, 1999.

[BCC+10] A Bhaskara, M Charikar, E Chlamtac, U Feige, and A Vijayaraghavan. Detecting high log-densities–an $\mathcal{O}(n^{1/4})$ approximation for densest k-subgraph. *Arxiv preprint arXiv:1001.2891*, 2010.

[Boy06] Stephen Boyd. Convex optimization of graph laplacian eigenvalues. *International Congress of Mathematicians*, pages 1311–1319, 2006.

[CP90] R Carraghan and PM Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9(6):375–382, 1990.

[dA07] Nair Maria Maia de Abreu. Old and new results on algebraic connectivity of graphs, Jan 2007.

[FPK01] U Feige, D Peleg, and G Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[FS97] Uriel Feige and Michael Seltser. On the densest k-subgraph problem. *Algorithmica*, Sep 1997.

[GB06a] A Ghosh and S Boyd. Growing well-connected graphs. *Decision and Control, 2006 45th IEEE Conference on*, pages 6605–6611, 2006.

[GB06b] Arpita Ghosh and Stephen Boyd. Upper bounds on algebraic connectivity via convex optimization. *Linear Algebra Appl*, 418(2-3):693–707, Jan 2006.

[Hal70] Kenneth Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, Nov 1970.

[Jer06] M Jerrum. Large cliques elude the metropolis process. *Random Structures and Algorithms*, Jan 2006.

[ST05] A Suzuki and T Tokuyama. Dense subgraph problem revisited. *Joint Workshop New Horizons in Computing and Statistical Mechanical Approach to Probabilistic Information Processing*, 2005.

[SW99] Anand Srivastav and Katja Wolf. Finding dense subgraphs with mathematical programming. Jan 1999.