Finding a semi-randomly hidden clique

Uriel Feige

June 7, 2021

1 Introduction

We recall some bounds from spectral graph theory that apply to random graphs. Let $p \leq \frac{1}{2}$ satisfy $p \geq \frac{n^\epsilon}{n}$ for some $\epsilon > 0$ (we think of ϵ as fixed as n grows). Then for the adjacency matrix of a random $G \in_R G_{n,p}$ graph, the following bounds hold with overwhelming probability. (The probability is over the choice of G. We abbreviate the term *with overwhelming probability* to w.o.p., and interpret it to mean that the probability is at least $1-O(2^{n^{-\delta}})$, for some $\delta > 0$.)

1. $\lambda_1(G) \simeq pn$, with the corresponding eigenvector being roughly the all 1 vector.
2. $\max[\lambda_2(G), |\lambda_n(G)|] \leq c\sqrt{pn}$, where $c > 0$ is some universal constant independent of n and p.

Let $G_{n,\frac{1}{2},k}$ be the distribution over random $G_{n,\frac{1}{2}}$ graphs with a randomly planted clique K of size k. Our goal is to design an algorithm that w.o.p. finds a clique of size k in a graph $G \in_R G_{n,\frac{1}{2},k}$. We have seen that if $k \geq c_1\sqrt{n \log n}$ for a sufficiently large constant c_1, then almost surely (with probability tending to 1 as n grows) K is composed of the k vertices of highest degree in G. We also noted that if $k \geq c_1\sqrt{n}$ then w.o.p. $\lambda_2(G) > c\sqrt{\frac{n}{2}}$, and hence we can distinguish between the distribution $G_{n,\frac{1}{2}}$ and the distribution $G_{n,\frac{1}{2},k}$. We noted that [1] showed how to use this fact in order to actually find K in polynomial time (w.o.p.). In this lecture we will show a different algorithm for finding K, based on [2]. This algorithm has the advantage of being more robust compared to the algorithm of [1], an issue that will be discussed in Section 3.

2 The algorithm

Let A denote the adjacency matrix of G, let I denote the identity matrix, and let J denote the all 1 matrix. Consider the matrix $B = 2(A + I) - J$. It has 1 along the diagonal and in entries B_{ij} for which $(i, j) \in E$, and -1 elsewhere.

To intuitively understand the spectrum of B, consider $G \in_R G_{n,\frac{1}{2}}$, and suppose that the all 1 vector 1_V is an eigenvector of A (indeed, it is a good approximation for the eigenvector corresponding to $\lambda_1(A)$). Then 1_V is also an eigenvector of B, with eigenvalue $2(\lambda_1(A) + 1) - n$, which is smaller than $\sqrt{n}$. Every other eigenvalue of A is orthogonal to 1_V, and consequently is also an eigenvalue of B, with eigenvalue $2\lambda_i + 2 \leq c\sqrt{2n}$. Indeed, this intuition is correct, and w.o.p., $\max[\lambda_1(B), |\lambda_n(B)|] \leq c\sqrt{n}$, for some sufficiently large constant c. See [3] for more details on bounds on eigenvalues of random symmetric matrices.

Observe that if $G \in_R G_{n,\frac{1}{2},k}$, then $\lambda_1(B) \geq k$, as can be seen by considering a Raleigh quotient for the vector 1_K (entries corresponding to the planted clique are 1, the remaining entries are 0). Hence if k is sufficiently large, the planted clique might be found by inspecting the eigenvector corresponding to $\lambda_1(B)$. Indeed, this is the approach followed by [1]. We will instead follow the approach of [2], based on the theta function of Lovasz [4].

Given a graph $G \in_R G_{n,\frac{1}{2},k}$, consider the following optimization problem, that we refer to as $\bar{\vartheta}(G)$:

Minimize $\lambda_1(M)$ subject to:

1. Matrix M is a symmetric matrix of order n.
2. $M_{ij} = 1$ whenever $B_{ij} = 1$. Namely, $M_{ii} = 1$ for every i, and $M_{ij} = 1$ for every edge $(i, j) \in E$.

The optimal value for $\bar{\vartheta}(G)$ is at least k (as the Raleigh quotient argument holds for M). The key to our algorithm is the following theorem proved in [2].

Theorem 1 *If $k \geq c_2\sqrt{n}$ (for a sufficiently large constant $c_2 > 0$) then w.o.p., $\bar{\vartheta}(G) = k$.*

To use Theorem 1, we need to efficiently compute $\bar{\vartheta}(G)$. This can be done using semi-definite programming (SDP). We defer the details to Section 4.

Given Theorem 1, finding K is straightforward. W.o.p., for every vertex $v \in V$, we have that if $v \in K$ then $\bar{\vartheta}(G_{-v}) = \bar{\vartheta}(G) - 1$, and if $v \notin K$ then $\bar{\vartheta}(G_{-v}) = \bar{\vartheta}(G)$ (here G_{-v} denotes the subgraph of G induced on all vertices but v). These statements follow from the fact that G_{-v} is distributed either like $G_{n-1,\frac{1}{2},k-1}$ or $G_{n-1,\frac{1}{2},k}$ (depending on whether $v \in K$), and from (applying a union bound on) Theorem 1. Hence K can be found by computing the function $\bar{\vartheta}$ on n subgraphs of G. In fact, $O(\frac{n}{k})$ computatons of $\bar{\vartheta}$ suffice in expectation, because for every vertex v that is detected to be in K, all its non-neighbors can be simultaneously marked as not belonging to K. A further improvement is to define G_{-v} as the subgraph of G induced only on the neighbors of v. In this case we have that if $v \in K$ then $\bar{\vartheta}(G_{-v}) = \bar{\vartheta}(G)-1$, and if $v \notin K$ then $\bar{\vartheta}(G_{-v}) \simeq \frac{\bar{\vartheta}(G)}{2}$ (if c_2 is sufficiently large).

One might hope that a single computation of $\bar{\vartheta}(G)$ suffices in order to find K, using the matrix M returned by this computation. For this matrix M we have that $\lambda_1(M) = \bar{\vartheta}(G) = k$. Moreover, the indicator vector 1_K for K has Raleigh quotient equal to $k = \lambda_1(M)$, and hence 1_K is an eigenvector for M, corresponding to λ_1. Hence if $\lambda_2(M)$ is smaller than k (better still, smaller than $k-1$, so that we do not need very high precision in our computations), then K can be recovered from M by computing the eigenvector that corresponds to $\lambda_1(M)$. Indeed, the proof of Theorem 1 shows that such a matrix M with $\lambda_1(M) = k$ and $\lambda_2(M) < k-1$ exists. However, the optimization problem for $\bar{\vartheta}(G)$ is likely to have multimple solutions. For other matrices M that solve it optimally (giving value k) λ_1 may have multiplicity larger than 1. This is exemplified in the homework assignment. For such matrices, there is a supspace of dimension larger than 1 for the eigenvectors corresponding to λ_1, and it might not be as easy to extract K from vectors in this subspace. Without adding more constraints to the formulation of $\bar{\vartheta}(G)$, we are not guaranteed to obtain a matrix M with $\lambda_2 \leq k-1$. (Interestingly, for matrix B it does hold w.o.p. that $\lambda_1(B) \geq k$ and $\lambda_2(B) \leq k-1$.)

The Lovasz theta function has several alternative formulations. Our $\bar{\vartheta}$ is based on a formulation referred to as ϑ_2. Using a different formulation, referred to as ϑ_4, it is shown in [2] that a single computation of $\bar{\vartheta_4}(G)$ suffices w.o.p. in order to extract all vertices of K.

3 A semi-random model

Consider a semirandom model $AG_{n,p,k}$ (here A stands for *adversarial*) in which one first generates at random a graph $G' \in_R G_{n,p,k}$, and then an adversary can remove from G' edges of its choice, provided that K remains a clique. In a sense, this only makes the task of the algorithm easier, as there are fewer non-clique edges to be confused with clique edges. However, it is not difficult to see that both the algorithm listing vertices by their degrees and the algorithm of [1] are fooled by such an adversary. (The adversary can easily increase the second eigenvalue of the adjacency matrix of G. For example, by removing edges, the subgraph induced on $V \setminus K$ can be broken into t connected components, each of size roughly n/t, giving t eigenvalues each of size rounghly $\frac{n}{2t}$.)

In contrast, such an adversary cannot increase the value of $\bar{\vartheta}(G)$ (as the adversary only removes constraints from the corresponding optimization problem), and cannot decrease its value (as the clique of size k remains). Hence the adversary has no effect at all on Theorem 1, and on algorithms that are based on it.

4 Solving SDPs

To be written.

5 Notes on the proof of Theorem 1

By permuting the order of vertices (this does not effect the eigenvalues, and only permutes coordinates in its eigenvectors), we may assume that the vertices of K are numbered 1 to k. We can partiton B (and later M) into four blocks C, D^T, D, F. The top-left corner C is an order k all 1 matrix. The bottom-right corner F is an order $n-k$ symmetric matrix with ± 1 values. D (bottom-left) is an $n-k$ by k matrix with ± 1 values.

We know that for M the vector 1_K will have Raleigh quotient k. Hence we would like it to be an eigenvector of M. This dictates that in B row sums are 0. To achieve this, do the following. Let n_i denote the number of -1 entries in row i of D. Then the row sum is $k - 2n_i$. To make it 0, add to every -1 entry of row i the value $x_i = \frac{2n_i - k}{n_i}$, obtaining a matrix D'. The

matrix M is the same as B, but with D replaced by D' (and D^T replaced by D'^T).

A tool useful for bounding $\lambda_2(M)$ is the following (simplified version) of Weyl's theorem.

Theorem 2 *Let A and B be two symmetric matrices of order n, and let $C = A + B$. Then for every $1 \leq i \leq n$ and $j + k \leq i + 1$ it holds that:*

$$\lambda_i(C) \leq \lambda_j(A) + \lambda_k(B)$$

Likewise, for every $1 \leq i \leq n$ and $j + k \geq i + n$ it holds that:

$$\lambda_i(C) \geq \lambda_j(A) + \lambda_k(B)$$

To use Theorem 2, we decompose M into a sum of three matrices.

Matrix X describes the graph G before planting of K, and has $+1$ entries along the diagonal and for every edge of G, and -1 entries elsewhere. In particular, X coincides with M in the bottom-right order $n - k$ corner.

Matrix Y describes the planting process, and has $+2$ entry for every edge added by the planting process. In particular, $X + Y$ coincides with M both in the bottom-right order $n - k$ corner, and in the top-left order k corner.

Matrix Z is $M - (X + Y)$. If has non-zero corresponding to the changes $D' - D$ (and $D'^T - D^T$).

All matrices X, Y, Z, M are symmetric, and $M = X + Y + Z$.

Applying Theorem 2 (twice) we get that:

$$\lambda_2(M) \leq \lambda_1(X) + \lambda_2(Y) + \lambda_1(Z)$$

Being a random ± 1 order n matrix (with 1 along the diagonal) we have that w.o.p. $\lambda_1(X) \leq c\sqrt{n}$.

As Y is a random $\{0, 2\}$ order k matrix (and 0 elsewhere) we have that w.o.p. $\lambda_2(Y) \leq 2c\sqrt{k}$.

To bound $\lambda_1(Z)$, note that $(\lambda_1(Z))^2$ is smaller than the trace of Z^2. This trace is the sum of square norms of the rows of $D' - D$. Every such row i is expected to have $n_i \simeq \frac{k}{2}$ non-zero entries, and each such entry is expected to have absolute value $|x_i| = O(\frac{1}{\sqrt{k}})$. Hence the square norm of a row is expected to be $O(1)$, and the trace of Z^2 is expected to be $O(n)$. Consequently, we expect that $\lambda_1(Z) = O(\sqrt{n})$, which is smaller than $\frac{k}{2}$ when k is a sufficiently

large constant times $\sqrt{n}$. Indeed, this event happens with overwhelming probability (see Lemma 3 in [2]).

Consequently, we have w.o.p. that $\lambda_2(M) \leq c\sqrt{n} + 2c\sqrt{k} + \frac{k}{2} < k$, as desired.

6 Homework

Hand in by June 21, 2021.

Consider the $G_{n,\frac{1}{2},k_1,k_2}$ model in which one first generates a random graph $G' \in_R G_{n,\frac{1}{2}}$, and then plants in it two random disjoint cliques, K_1 of size k_1, and K_2 of size k_2.

1. Prove the following Analogue of Theorem 1. If $k_1 \geq k_2 \geq c_2\sqrt{n}$ (for a sufficiently large constant $c_2 > 0$) then w.o.p., $\bar{\vartheta}(G) = k_1$. You may use without proof bounds on the eigenvalues of random $\{0,1\}$ and random ± 1 matrices (similar to uses made in the lecture), and need not repeat the proof of Lemma 3 from [2] (though you may need to explain why the constant 96 there can be changed to a different constant).

2. Using the above, give a polynomial time algorithm that actually finds K_1 and K_2.

3. Getting back to the $G_{n,\frac{1}{2},k}$ model with $k \geq c_2\sqrt{n}$ (for a sufficiently large constant $c_2 > 0$), show that the optimization problem underlying $\bar{\vartheta}(G)$ is likely to have solutions for which the multiplicity of $\lambda_1(M)$ is larger than 1.

References

[1] Noga Alon, Michael Krivelevich, Benny Sudakov: Finding a large hidden clique in a random graph. Random Struct. Algorithms 13(3-4): 457–466 (1998).

[2] Uriel Feige, Robert Krauthgamer: Finding and certifying a large hidden clique in a semirandom graph. Random Struct. Algorithms 16(2): 195–208 (2000).

[3] Zoltan Furedi, Janos Komlos: The eigenvalues of random symmetric matrices. Comb. 1(3): 233–241 (1981).

[4] Laszlo Lovasz: On the Shannon capacity of a graph. IEEE Trans. Inf. Theory 25(1): 1–7 (1979).