Additional notes on planted clique and independent set

Uriel Feige

June 28, 2021

1 Computing the Lovasz theta function

Recall that the following was an upper bound on the size $\omega(G)$ of the maximum clique in graph $G(V, E)$ (based on the ϑ_2 formulation of the Lovasz theta function).

minimize $\lambda_1(M)$ **subject to:**

- M is a symmetric matrix of order n.
- $M_{ii} = 1$ for all $1 \leq i \leq n$.
- $M_{ij} = 1$ for all $(i, j) \in E$.

The number of variables in this program is $\binom{n}{2} - |E|$.

The program can be solved (up to arbitrary precision) using the ellipsoid algorithm. As no variable M_{ij} needs to be larger than $n - 1$ (by a Rayleigh quotient argument with an indicator vector for the set $\{i, j\}$), we have a bounding ball. Seeking a solution of value at most k, a matrix M violates it if $\lambda(M) > k$. The corresponding unit norm eigenvector v gives a violated constraint $v^T M v \leq k$.

2 An alternative formulation of the Lovasz theta function

Here is another upper bound on the size of the maximum clique in graph $G(V, E)$, this time based on the ϑ_3 formulation of the Lovasz theta function.

maximize $\sum_{i,j \leq n} B_{i,j}$ (equivalently, maximize $Tr(BJ)$) **subject to:**

- B is symmetric positive semidefinite (PSD).
- $Tr(B) = 1$ (namely, $\sum_i B_{ii} = 1$).
- $B_{ij} = 0$ for every $i \neq j$ with $(i, j) \notin E$.

If G has a clique K of size k, then having m composed of a k by k block of values of $\frac{1}{k}$ in the intersection of the rows and columns of K (and 0 elsewhere) shows that $\vartheta_3(G) \geq \omega(G)$.

In fact, for every graph it holds that $\vartheta_3(G) = \vartheta_2(G)$. Let us show the easier direction of the inequality, namely, $\vartheta_3(G) \leq \vartheta_2(G)$, as this is the direction more relevant to planted clique applications.

Let M and B be optimal solutions for ϑ_2 and ϑ_3, respectively. Observe that $C = \vartheta_2 I - M$ is symmetric PSD. Consider $Tr(BC) = Tr(B\vartheta_2 I) - Tr(BM) = \vartheta_2 - \vartheta_3$ (note that $BM = BJ$). The fact that B and C are symmetric PSD implies that $Tr(BC) \geq 0$. Hence $\vartheta_2 \geq \vartheta_3$.

(*Sketch of proof that* $Tr(BC) \geq 0$. For n by r and r by n matrices X and Y, term by term comparison shows that $Tr(XY) = Tr(YX)$. Represent C as QQ^T where Q is n by r. Then $Tr(BC) = Tr(BQQ^T) = Tr(Q^T BQ)$. Let q_i denote the ith column of Q. Then the r terms on the diagonal of $(Q^T B)Q$ can be seen to be the r values $(q_i^T B)q_i$. As B is PSD, each of these values is nonnegative.)

ϑ_2 can be approximated arbitrarily well using the ellipsoid algorithm. We refer to this as *semidefinite programming* (SDP) as the only nonlinear constraint is that B is PSD.

For the $G_{n,\frac{1}{2},k}$ model, the advantage of the ϑ_2 formulation is that being a minimization problem, we could prove that w.o.p. we have $\vartheta_2(G) = k$ (when k is sufficiently large). However, for actually finding the hidden clique, using the maximization version ϑ_3 is more elegant. We use the following two facts, that hold with w.o.p..

1. Removing any vertex $v \in K$ decreases ϑ_3 to $k-1$.

2. Removing and vertex $v \notin K$, the value of ϑ_3 remains k.

Let B^* be an optimal solution for ϑ_3. Being PSD, the diagonal of B^* is nonnegative (if $B^*_{ii} < 0$ then the vector 1_i has a negative Rayleigh quotient). No vertex i can have negative row sum in B^*. This is because zeroing its row and column would keep the matrix PSD but with higher sum of entries. The diagonal entries contribute exactly 1. For a vertex $i \in K$, let d_i be its diagonal value, and let s_i be its total off-diagonal row sum. Then $\frac{1}{1-d_i}(k - d_i - 2s_i) \leq k-1$ (because by removing v we can scale the remaining entries by $\frac{1}{1-d_i}$, but still will not pass $n-1$). Consequently, $2s_i \geq 1 + d_i(k-2)$. For $i \notin K$ we have $\frac{1}{1-d_i}(k - d_i - 2s_i) \leq k$, and consequently $2s_i \geq d_i(k-2) + d_i$. Summing over all vertices we have that $2\sum_i s_i \geq k + (k-2) + \sum_{i \notin K} d_i$. (We used the fact that $\sum d_i = 1$.) As $\sum s_i = k-1$ we infer that $\sum_{i \notin K} d_i = 0$. By nonnegativity of the diagonal, $d_i = 0$ for all $i \notin K$. As B^* is PSD, all rows and columns of vertices not in K are all 0.

Consider now the k by k block of K in B^*, which we will now refer to as the submatrix K (the rest of B^* is 0). K is PSD, and can be decomposed into QQ^T. Let q_i be the ith row of Q. Then $0 \leq \sum_{1 \leq i < j \leq k}(q_i - q_j)^2 = (k-1)\sum_{i \leq k} K_{ii} - \sum_{i \neq k} K_{ij} = 0$. Hence the first inequality must be an equality, implying that $q_i = q_j$ for all i and j. Hence all entries in K are $\frac{1}{k}$.

We can recognize the vertices of the planted clique by inspecting the diagonal of B^*. The vertices of K have value $1/k$, and the other vertices have value 0.

Clearly, it suffices to compute B^* within entrywise error smaller than $\frac{1}{2k}$ to exactly recover the planted clique.

It is somewhat surprising that ϑ_2 only gives the size of the planted clique whereas ϑ_3 also gives us the vertices. (In [2], ϑ_4 was used for this last purpose.)

3 Stronger model for hidden independent set

For $p \leq n^{\delta-1}$ with $0 < \delta < 1$, the value of the ϑ function for random $G_{n,p}$ graphs is known to be $\Theta(\sqrt{\frac{n}{p}})$ w.o.p. This suggests that we can find planted independent sets of size $k \geq c\sqrt{\frac{n}{p}}$ in $G_{n,p,k}$, where c is a sufficiently large constant (here the trick of guessing a few vertices from the independent set does not help in reducing the constant). The algorithms described in class indeed work for these parameters, and extend to the semirandom model in which an adversary is allowed to add edges outside the planted independent set.

An interesting case is that of a planted independent set K of size $k = \alpha n$, for a constant $0 < \alpha < 1$. For this regime Feige and Kilian [1] considered a stronger semi-random model, where only edges in $(K, V \setminus K)$ are included with probability p, and then an adversary can add arbitrary edges (but not within K). In this model K is not necessarily the largest independent set. The goal might be to find any independent set of size k, or alternatively, to output K as part of a list of independent sets. It is shown that this can be done if $p \geq \frac{(1+\epsilon)\ln n}{\alpha n}$, and NP hard (under randomized reductions) if $p \leq \frac{(1-\epsilon)\ln n}{\alpha n}$. We will show the hardness result in class (the algorithmic result is much more complicated).

For smaller values of k the tradoffs between k and p are only partly understood for the stronger semi-random model [3].

References

[1] Uriel Feige, Joe Kilian: Heuristics for Semirandom Graph Problems. J. Comput. Syst. Sci. 63(4): 639-671 (2001)

[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] Theo McKenzie, Hermish Mehta, Luca Trevisan: A New Algorithm for the Robust Semi-random Independent Set Problem. SODA 2020: 738–746.