Bocconi University — 41000: Graph Theory Handout 10
Luca Trevisan March 1, 2022

Lecture 10: Semidefinite Programming for Planted Clique

In which we show that, in the computationally tractable regime $k \gg \sqrt{n}$ of the planted clique problem, the planted clique is the unique optimum solution of a Semidefinite Programming relaxation of the maximum clique problem. We also study the robustness of SBM and planted clique algorithms to adversarial manipulations of the input.

1 A Semidefinite Programming Relaxation of Maximum Clique and its Dual

Following what we did for the SBM, we are going to formulate a Semidefinite Programming relaxation of the maximum clique problem, formulate its dual, and then use duality to argue that, for graphs coming from the planted clique distribution, the planted clique is, with high probability, the unique optimum of the Semidefinite Program.

To formulate a Semidefinite Programming relaxation of a combinatorial problem, it is helpful to first formulate the combinatorial problem as a homogeneous quadratic optimization problem, because we can then mechanically relax the homogeneous quadratic optimization problem to a semidefinite program.

Given a graph $G = (V, E)$, the maximum clique problem can be formulated as the quadratic programming problem

$$\begin{array}{ll} \text{maximize} & \sum_{v \in V} x_v \\ \text{subject to} & \\ & x_v^2 = x_v \qquad \forall v \in V \\ & x_u x_v = 0 \qquad \forall (u,v) \notin E \end{array} \tag{1}$$

The constraint $x_v^2 = x_v$ forces each x_v to be either 0 or 1, and the other constraints require the set of vertices v such that $x_v \neq 0$ to be a clique, because two non-zero edges cannot be the endpoints of a non-edge (sorry for all the negations, this discussion would be a bit cleaner if we talked about independent set instead of clique). The cost function is equal to the number of v such that $x_v \neq 0$, and hence counts the number of vertices in a maximum clique.

The challenge in translating (1) to a semidefinite program is that is it not an homogenous quadratic program. There is a standard technique to "homogenize" quadratic programs, which involves adding a new variable x_0, require $x_0^2 = 1$, and then proceed as if $x_0 = 1$. With these ideas, the quadratic program becomes:

$$\begin{aligned} &\text{maximize } \sum_{v \in V} x_v^2 \\ &\text{subject to} \\ &\qquad x_0^2 = 1 \\ &\qquad x_v^2 = x_v x_0 \qquad \forall v \in V \\ &\qquad x_u x_v = 0 \qquad \forall (u,v) \notin E \end{aligned} \tag{2}$$

A feasible solution of the above quadratic program is such that $x_0 = \pm 1$ and each x_v equals either 0 or x_0. The non-edge constraints enforce that the vertices such that $x_v \neq 0$ form a clique, and the cost function counts the number of such vertices.

The above quadratic program can be mechanically relaxed to the semidefinite program below.

$$\begin{aligned} &\text{maximize } \sum_{v \in V} ||x_v||^2 \\ &\text{subject to} \\ &\qquad ||x_0||^2 = 1 \\ &\qquad ||x_v||^2 = \langle x_v, x_0 \rangle \qquad \forall v \in V \\ &\qquad \langle x_u, x_v \rangle = 0 \qquad \forall (u,v) \notin E \end{aligned} \tag{3}$$

The above SDP is called the *Lovasz Theta function* of the (complement of the) graph, and it has been extensively studied. Lovasz found several equivalent formulations of the above relaxation and of its dual. Without developing the whole theory of the Theta function, let us start again with another quadratic formulation of maximum clique, which is "natively" homogenous.

$$\begin{aligned} &\text{maximize } \left(\sum_{v\in V} x_v\right)^2 \\ &\text{subject to} \\ &\qquad \sum_{v\in V} x_v^2 = 1 \\ &\qquad x_u x_v = 0 \qquad \forall (u,v) \notin E \end{aligned} \tag{4}$$

It might not be immediately clear that the above quadratic program is a formulation of the maximum clique problem, but if C is a clique in the graph and $k = |C|$, then we can assign $x_v = \frac{1}{\sqrt{k}}$ if $v \in C$ and $x_v = 0$ otherwise, which is feasible and has cost k. Thus the optimum of (2) is at least the size of the maximum clique in the graph. On the other hand, if $\{x_v\}_{v\in V}$ is a feasible solution, and $C = \{v : x_v \neq 0\}$, then C is a clique because of the non-edge constraints, and the cost of the solution is

$$\left(\sum_{v\in V} x_v\right)^2 = \left(\sum_{v\in C} x_v\right)^2 \leq |C| \cdot \sum_{v\in C} x_v^2 = |C|$$

and so the size of the maximum clique in the graph is an upper bound to the optimum of (2), and hence the two quantities are equal.

The standard semidefinite programming relaxation of (2) is

$$\begin{aligned} &\text{maximize } \left\|\sum_{v\in V} x_v\right\|^2 \\ &\text{subject to} \\ &\qquad \sum_{v\in V} ||x_v||^2 = 1 \\ &\qquad \langle x_u, x_v\rangle = 0 \qquad \forall (u,v) \notin E \end{aligned} \tag{5}$$

It possible to prove that (5) is equivalent to (3), although we will not prove this fact. We can also write (5) as

$$\begin{aligned} &\text{maximize } J \bullet X \\ &\text{subject to} \\ &\qquad I \bullet X = 1 \\ &\qquad X_{u,v} = 0 \qquad \forall (u,v) \notin E \\ &\qquad X \succeq \mathbf{0} \end{aligned} \tag{6}$$

and its dual is very simple:

$$\begin{aligned}
&\text{minimize } y \\
&\text{subject to} \\
&\qquad y \cdot I + M \succeq J \\
&\qquad M_{u,v} = 0 \qquad \forall (u,v) \in E \\
&\qquad M_{v,v} = 0 \qquad \forall v \in V
\end{aligned} \tag{7}$$

Indeed, we see that if y, M are feasible for (7) and X is feasible for (6), then

$$J \bullet X \leq (y \cdot I + M) \bullet X = y$$

where the first inequality follows from the fact that $X \succeq \mathbf{0}$ and $J \preceq yI + M$ and the final equality follows from $M \bullet X = 0$ (the two matrices have disjoint sets of non-zero entries) and $I \bullet X = 1$ (this is a constraint on X).

2 Constructing a Dual Solution

We need to find a matrix M whose non-zero entries are concentrated on the non-edges of G and such that this matrix plus $k \cdot I$ dominates J in the PSD sense, where k is the size of the planted clique.

We know that if A is the adjacency matrix of G and $\bar{A}$ is the adjacency matrix of the complement graph (that is $\bar{A}_{u,v} = 1$ for the non-edges (u,v) of G) we have

$$\mathbb{E}\,\bar{A} = \frac{1}{2}J - \frac{1}{2}\mathbf{1}_C\mathbf{1}_C^T$$

where C is the planted clique. Furthermore we have, from previously discussed concentration results, that with high probability:

$$\left\| \bar{A} - \frac{1}{2}J - \frac{1}{2}\mathbf{1}_C\mathbf{1}_C^T \right\| \leq O(\sqrt{n})$$

and so

$$J \preceq 2\bar{A} - \mathbf{1}_C\mathbf{1}_C^T + O(\sqrt{n})I$$

which is almost what we are looking for, because $2\bar{A}$ is a matrix that is non-zero only on non-edges of G and $O(\sqrt{n}) \leq k$.

The main problem is the $\mathbf{1}_C\mathbf{1}_C^T$ term. Following what we did for the SBM, we could hope to prove

$$J \preceq 2\bar{A} + kI$$

by arguing that if $x \perp \mathbf{1}_C$, then

$$x^T(2\bar{A} + kI - J)x = x^T(2\bar{A} - \mathbf{1}_C\mathbf{1}_C^T + kI - J)x > 0 \tag{8}$$

and then proving that

$$(2\bar{A} + kI - J)\mathbf{1}_C = \mathbf{0} \tag{9}$$

We definitely have (8), but there is a problem with (9): the expression on the left is actually 0 in the coordinates of C, but for coordinates $u \notin C$ we have

$$((2\bar{A} + kI - J)\mathbf{1}_C)_u = 2 \cdot \text{ (number of non-neighbors of } v \text{ in } C) \ - k$$

We can introduce a "correction" term, by exploiting the fact that, with high probability, all the above quantities are in the range $\pm O(\sqrt{k \log n})$. Define the matrix N as follows:

$$N_{u,v} = \begin{cases} 2 - \frac{k}{\text{(number of non-neighbors of } u \text{ in } C)} & \text{if } u \notin C, v \in C, \text{ and } (u,v) \notin E \\ \\ 0 & \text{otherwise} \end{cases}$$

Tracing the definitions shows that N is non-zero only on non-edges, and that

$$(N \cdot \mathbf{1}_C)_u = 2 \cdot \text{ (number of non-neighbors of } v \text{ in } C) \ - k$$

so that

$$((2\bar{A} - N + kI - J)\mathbf{1}_C = \mathbf{0}$$

We also have, with high probability, $||N|| \leq O(\sqrt{k \log n})$, and so, provided that $k \gg \sqrt{k \log n} + \sqrt{n}$ we derive that for all $x \perp \mathbf{1}_C$

$$x^T(2\bar{A} - N + kI - J)x = x^T(2\bar{A} - N - \mathbf{1}_C\mathbf{1}_C^T + kI - J)x > 0$$

So we have obtained a dual solution, where $M = 2\bar{A} - N$ and $y = k$ that proves that the size of the maximum clique is an upper bound to the value of the SDP (under conditions that hold with high probability) and $\mathbf{1}_C$ is the unique optimum solution (because every other feasible solution will have a strictly smaller objective value).

3 Robustness

We have already seen some advantages of Semidefinite Programming over spectral methods for problems such as SBM and planted clique: an approximate solution for SBM can be recovered even in the regime of constant average degree, in which spectral methods do not work (unless one adds a pre-processing step that regularizes the graphs), and in both SBM and planted cliques there are regimes in which SDP finds the hidden solution without any post-processing step, since the hidden solution is the unique optimum of the SDP.

An additional advantage of SDP is that it does not "overfit" to the probabilistic models that we discussed, and it keeps working even in the presence of deviations from the distribution.

To provide a rigorous approach to the study of robustness, it is possible to study *semi-random* generative models in which an instance is produced by first sampling an instance from a probabilistic model (for example a graph from the SBM or from the planted clique distribution), and then allowing an adversary to modify the instance in a certain limited way.

In such models, the spectral algorithms that we described and that work, in some regimes, in the SBM and the planted clique model, fail even in semi-random models in which the adversary is "helpful." In the SBM, an adversary is "helpful" if, after a graph is sampled from the SBM, the adversary only deletes edges that cross the partition and adds edges that do not cross the partition. In the planted clique case, an adversary is "helpful" if it only deletes non-clique edges (and does not add any edge). Such adversaries are called "helpful" because they change the instance in ways that, intuitively, should make the task of finding the hidden solution easier.

We will not prove it, but it is possible to prove that the spectral algorithm for the SBM and the spectral algorithm for planted clique that we described can be made to fail by a helpful adversary.

In the case of SDP, the analysis for the random case holds even in the semi-random helpful case. Consider the regime in which the SDP has the hidden partition as its unique optimal solution. It is not difficult to show that any change made by a helpful adversary preserves the invariant that the hidden partition is the unique optimal solution. The same is true for the SDP for planted clique described in these notes. Suppose that G is a graph with a planted clique C of size k, and suppose that $\frac{1}{k}\mathbf{1}_C\mathbf{1}_C^T$ is the unique optimal solution of the SDP (5). If a helpful adversary deletes any non-clique edge, then $\frac{1}{k}\mathbf{1}_C\mathbf{1}_C^T$ is still a feasible solution of value k; furthermore, the process of removing edges adds constraints to the SDP, and hence restricts the set of feasible solution. If, before the helpful adversary intervened, all feasible solutions different from $\frac{1}{k}\mathbf{1}_C\mathbf{1}_C^T$ had cost $< k$, then the same will be true for a stronger reason after the intervention of the helpful adversary, and so the planted clique remains the unique optimal solution even after the adversarial changes.