Scribed by Haaris Khan
Last modified 10/3/2017

Lecture 5

In which we study the SDP relaxation of Max Cut in random graphs.

1 Quick Review of Chernoff Bounds

Suppose $X_1, ..., X_n$ are mutually independent random variables with values $0, 1$.
Let $X := \sum_{i=1}^{n} X_i$. The Chernoff Bounds claim the following:

1. $\forall\ \epsilon$ such that $0 \leq \epsilon \leq 1$,

$$\mathbb{P}(|X - \mathbb{E}[X]|) > \epsilon \cdot \mathbb{E}[X]) \leq \exp(\Omega(\epsilon^2 \cdot \mathbb{E}[X]))$$

2. $\forall\ t\ > 1$,

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq t \cdot \mathbb{E}[X]) \leq \exp(-\Omega((t \log(t)) \cdot \mathbb{E}[X]))$$

3. When we do not know $\mathbb{E}[X]$, we can bound as follows:

$$\mathbb{P}(|X - \mathbb{E}[X]| \geq \epsilon \cdot n) \leq \exp(-\Omega(\epsilon^2 \cdot n))$$

2 Cutting a Near-Optimal Number of Edges in $G_{n,p}$ Via SDP Rounding

Consider $G_{n,p}$ where $p > \frac{\log(n)}{n}$. We show that with $1 - o(1)$ probability, the max-degree will be $O(d)$

- Fix v
- For some constant c,

$$\begin{aligned}
\mathbb{P}(\text{v has degree} > c \cdot d) &= \mathbb{P}(|deg(v) - \mathbb{E}[v]| > (c-1)\,\mathbb{E}[deg(v)]) \\
&\leq \exp(-\Omega((c-1)\log(c-1) \cdot d)) \text{ (by Chernoff Bounds)} \\
&\leq \exp(-\Omega((c-1)\log(c-1) \cdot \log(n)) \\
&\leq \frac{1}{n^2}, \text{ for some choice of constant c}
\end{aligned}$$

So $\mathbb{P}(\exists\, v \text{ with degree } > c \cdot d) \leq n \cdot \frac{1}{n^2} \leq \frac{1}{n}$

Next, we compute the number of vertices that participate in a triangle. Recall that degree d can be bounded by $o(n^{\frac{1}{3}})$

$$\mathbb{E}[\text{number vertices in triangles}] = n \cdot \mathbb{P}(\text{v participates in a triangle})$$

If a vertex participates in a triangle, there are $\binom{n-1}{2}$ ways of choosing the other two vertices that participate with v in the triangle.
So the expected number of vertices in triangles can be bounded by

$$\begin{aligned}\mathbb{E}[\text{number vertices in triangles}] &\leq n \cdot p^3 \cdot \binom{n-1}{2} \\ &\leq n^3 \cdot p^3 \\ &= o(n) \text{ if } p = o\left(\frac{1}{n^{\frac{2}{3}}}\right),\ d = o(n^{\frac{1}{3}})\end{aligned}$$

So with $o(1)$ probability,

- All vertices have degree $O(d)$
- $o(n)$ vertices participate in triangles.

3 Eigenvalue Computations and SDP

Problems like finding the largest / smallest eigenvalue can be solved using SDP

Let M be symmetric, $\lambda_{\max}$ be the largest eigenvalue of M: $\lambda_{\max} = \max_x \frac{\boldsymbol{x}^T M \boldsymbol{x}}{\|\boldsymbol{x}\|^2}$ We can formulate this as Quadratic Programming:

$$\begin{aligned}\max_{i,j} \quad & \sum_{i,j} M_{i,j} x_i y_j \text{s.t.} \\ & \sum_i x_i^2 = 1\end{aligned}$$

We showed previously that we can relax a Quadratic Program to SDP:

$$\begin{aligned}\max_{i,j} \quad & \sum_{i,j} M_{i,j} \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle \text{s.t.} \\ & \sum_i \|\boldsymbol{x_i}\|^2 = 1\end{aligned}$$

In fact, it happens that these two are equivalent. To show this, we must show that a vector solution x of SDP can hold as a solution to the QP and vice versa.

Proving x for QP is valid for SDP: Trivial. Any solution x to our Quadratic Program must be a solution for our SDP since it is a relaxation of the problem; then the optimum of our QP must be less than or equal to the optimum of our SDP

Proving x for SDP is valid for QP: Consider $x :=$ vector solution of cost c. We note that our SDP can be transformed into an unconstrained optimization problem as follows:

$$\max_{i,j} \qquad \frac{\sum_{i,j} M_{i,j} \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle}{\sum_i \|\boldsymbol{x_i}\|^2}$$

The cost c can be defined as the value of our solution:

$$c = \frac{\sum_{i,j} M_{i,j} \sum_k \boldsymbol{x_k}^i \boldsymbol{x_k}^j}{\sum_i \sum_k \|\boldsymbol{x_k}^i|^2}$$

$$\leq \max_k \frac{\sum_{i,j} M_{i,j} \boldsymbol{x_k}^i \boldsymbol{x_k}^j}{\sum_i \|\boldsymbol{x_k}^i\|^2}$$

We get a one-dimensional solution when we use the k^{th} element of x, and wish to find the k that maximizes this.

We use the following inequality:

$$\frac{a_1 + ... + a_m}{b_1 + ... + b_m} \leq \max_{k=1,...,m} \frac{a_k}{b_k}, b_k > 0$$

Proof:

$$\sum_i a_i = \sum_i b_i \cdot \frac{a_i}{b_i} \leq \sum_i b_i \cdot \max_k \frac{a_k}{b_k}$$

$$= \max_k \frac{a_k}{b_k} \cdot \sum_i b_i$$

4 SDP Max-Cut: Spectral Norm as a SDP Certificate

Consider the SDP relaxation of Max-Cut on Graph G:

$$\max \qquad \sum_{(i,j) \in E} \frac{1}{4} \|\boldsymbol{X_i} - \boldsymbol{X_j}\|^2$$

s.t.

$$\forall v \in V, \|\boldsymbol{X_v}^2\| = 1$$

Let the optimum value for this SDP be $SDPMaxCut(G)$. It's obvious that $MaxCut(G) \leq SDPMaxCut(G)$. Under our constraints, we can rewrite our SDP as

$$\sum_{(i,j)\in E} \frac{1}{2} - \frac{1}{2}\langle \boldsymbol{X_i}, \boldsymbol{X_j} \rangle$$

So our new optimization problem is

$$\begin{aligned} \max \quad & \frac{|E|}{2} - \sum_{(i,j)\in E} \frac{1}{2}\langle \boldsymbol{X_i}, \boldsymbol{X_j} \rangle \\ \text{s.t.} \quad & \\ & \forall i \in V, \|\boldsymbol{X_i}\|^2 = 1 \end{aligned}$$

We can relax our constraint to the following: $\forall i \in V, \sum_i \|\boldsymbol{X_i}\|^2 = n$. Relaxing our constraint will yield an optimization problem with a solution less than the stricter constraint (call this $SDP'MaxCut(G)$):

$$\begin{aligned} \max \quad & \frac{|E|}{2} - \sum_{(i,j)\in E} \frac{1}{2}\langle \boldsymbol{X_i}, \boldsymbol{X_j} \rangle \\ \text{s.t.} \quad & \\ & \sum_v \|\boldsymbol{X_v}\|^2 = n \end{aligned}$$

Clearly, we have the following inequalities: $MaxCut(G) \leq SDPMaxCut(G) \leq SDP'MaxCut(G)$. We can rewrite $SDP'MaxCut(G)$ as

$$\begin{aligned} \max \quad & \frac{|E|}{2} + \frac{n}{4} \cdot \sum_{i,j} \frac{-A_{i,j}\langle \boldsymbol{X_i}, \boldsymbol{X_j} \rangle}{\sum_i \|\boldsymbol{X_i}\|^2} \\ & \sum_v \|\boldsymbol{X_v}\|^2 = n \end{aligned}$$

Note that our objective function computes the largest eigenvalue of $-A$:

$$= \frac{|E|}{2} + \frac{n}{4} \cdot \lambda_{\max}(-A)$$

For every graph $G_{n,p}$ with $0 \leq p \leq 1$,

$$MaxCut(G) \leq SDPMaxCut(G) \leq \frac{|E|}{2} + \frac{n}{4} \cdot \lambda_{\max}(-A)$$

$$\leq \frac{|E|}{2} + \frac{n}{4} \cdot \lambda_{\max}(pJ - A)$$
$$\leq \frac{|E|}{2} + \frac{n}{4} \cdot \|pJ - A\|$$

Recall from previous lectures that for $p > \frac{\log(n)}{n}$, the adjacency matrix of A sampled from $G_{n,p}$ has $\|pJ - A\| \leq O(\sqrt{np})$ with high probability. This implies that $SDPMaxCut(G) \leq \frac{|E|}{2} + O(n \cdot \sqrt{d})$. Semantically, this means that $SDPMaxCut(G)$ computes in poly-time a correct upper-bound of $MaxCut(G)$.

5 Trace and Eigenvalues

Suppose matrix M is symmetric with eigenvalues $\lambda_1 \dots \lambda_n$. The following are true:

- M^k eigenvalues are $\lambda_1^k \dots \lambda_n^k$
- $trace(M) := \sum_{i,i} M_{i,i}$; $trace(M) = \sum_i \lambda_i$

Then, for $M^{2k}, trace(M^{2k}) = \lambda_1^{2k} + \ldots + \lambda_n^{2k}$.

$$(\max_i |\lambda_i|)^{2k} \leq trace(M^{2k}) \leq n \cdot (\max_i |\lambda_i|)^{2k}$$

Also,

$$\|M\| \leq (trace(M^{2k})^{\frac{1}{2k}} \leq n^{\frac{1}{2k}} \cdot \|M\|$$

$A_{i,j}$ is defined as the number of expected paths from i to j that take k steps (not necessarily simple paths in a graph)

$= \sum_{\text{paths from i to j}} M_{i,a_1} \dots M_{a_n,j}$

Our goal with this is to compute the eigenvalues λ. Since traces relates the sum of the diagonal and the sum of eigenvalues for symmetric M, we can use this to provide an upper bound for symmetric M.