Beyond Worst Case Analysis **Fall 2018**

Lecture 5: Planted Clique in Random Graph 2

Scribe: Antares Chen, Wilson Wu Oct 12, 2018

5.1 Introduction

In this note, we discuss another algorithm for planted clique using semidefinite programming. To do so, we will first introduce SDP duality, then provide a SDP relaxation for planted clique, analyze the relaxation and end with a discussion on how the SDP algorithm is, in some sense, more powerful than the spectral algorithm we saw during Lecture 3.

5.2 SDP Duality

Duality is an extremely versatile tool in proving bounds on the performance of optimization-based algorithms. To begin, recall that for $X, C, A_1, \ldots, A_m \in \mathbb{R}^{n\times n}$ and $b_1, \ldots, b_m \in \mathbb{R}$, a semi-definite program is defined as

$$\begin{aligned} \text{minimize} \quad & \langle C, X\rangle \\ \text{subject to} \quad & \langle A_1, X\rangle = b_1 \\ & \cdots \\ & \langle A_m, X\rangle = b_m \\ & X \succeq 0 \end{aligned} \tag{5.1}$$

where $\langle A, B\rangle = \sum_{i=1}^n \sum_{j=1}^n A_{ij}B_{ij}$ (this is also sometimes denoted as $A \bullet B$). When given an optimization problem (be it SDP, LP, or otherwise), we are sometimes interested in bounding the optimal value of its objective function. One way of providing a bound is to look at its *dual* optimization problem. The reader may have encountered LP duality in a course such as CS170; here we will extend the same ideas to derive a dual optimization problem for SDPs.

Let us denote equation 5.1 as the *primal* SDP. Its corresponding dual is an SDP whose objective function computes the tightest lower-bound on the primal optimal value (and equivalently, the tightest upper-bound if the SDP is maximization problem). A familiar method of deriving the dual LP is to apply multipliers to each constraint, sum the constraints, then pull out a lower-bound for the primal objective function.

We can attempt this process to derive the dual SDP, but we will first need to massage the constraint $X \succeq 0$ into an inequality linear in X. Since X is positive semi-definite, we know that for any $\boldsymbol{v} \in \mathbb{R}^n$, the quadratic form of X evaluated on $\boldsymbol{v}$ is non-negative. That is

$$\boldsymbol{v}^\top X \boldsymbol{v} \geq 0$$

These are inequalities linear in X, so let's rewrite the primal SDP as the following.

$$\begin{aligned} \text{minimize} \quad & \langle C, X\rangle \\ \text{subject to} \quad & \langle A_1, X\rangle = b_1 \\ & \dots \\ & \langle A_m, X\rangle = b_m \\ & \boldsymbol{v}^\top X \boldsymbol{v} \geq 0 \qquad \forall \boldsymbol{v} \in \mathbb{R}^n \end{aligned}$$

Now that we have an LP on n^2 variables, albeit with infinite constraints, we can multipliers to each constraint.

$$y_i \ : \ \langle A_i, X\rangle y_i = b_i y_i \quad \forall i = 1, \dots, m \qquad\qquad c_v \ : \ c_v \boldsymbol{v}^\top X \boldsymbol{v} \geq 0 \quad \forall \boldsymbol{v} \in \mathbb{R}^n$$

We call these multipliers *dual variables* and constrain $c_v \geq 0$ in order to preserve the direction of the inequality. Summing over all inequalities derives:

$$\sum_{i=1}^n \langle A_i, X\rangle y_i + \sum_{\boldsymbol{v} \in \mathbb{R}^n} c_v \boldsymbol{v}^\top X \boldsymbol{v} \geq \sum_{i=1}^n b_i y_i$$

which further simplifies to

$$\left\langle A_i y_i + \sum_v c_v \boldsymbol{v}\boldsymbol{v}^\top, X \right\rangle \geq \sum_{i=1}^n b_i y_i$$

d Notice that if we constrain $A_i y_i + \sum_v c_v \boldsymbol{v}\boldsymbol{v}^\top = C$, then $\sum_{i=1}^n b_i y_i$ is a lower-bound on the primal objective. Finding the tightest lower-bound is then equivalent to maximizing the right-hand side of the above inequality with respect to y_i, c_v. The dual program is

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^n b_i y_i \\ \text{subject to} \quad & \sum_{i=1}^n A_i y_i + \sum_{\boldsymbol{v}} c_v \boldsymbol{v}\boldsymbol{v}^\top = C \\ & c_v \geq 0 \qquad\qquad\qquad\qquad \forall \boldsymbol{v} \in \mathbb{R}^n \end{aligned}$$

This is in fact an SDP! Recall that a non-negative linear combination of matrices with form $\boldsymbol{v}\boldsymbol{v}^\top$ is PSD. Because $c_v \geq 0$, we can introduce an additional constraint positing the existence of a positive semi-definite $Z \in \mathbb{R}^{n\times n}$ such that $Z = \sum_v c_v v v^\top$. This creates the constraints

$$\sum_{i=1}^n A_i y_i + Z = C \qquad\qquad Z \succeq 0$$

which is equivalent to $C - \sum_{i=1}^n A_i y_i \succeq 0$. The dual SDP to the primal SDP equation 5.1 is given by

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^n b_i y_i \\ \text{subject to} \quad & C - \sum_{i=1}^n A_i y_i \succeq 0 \end{aligned} \tag{5.2}$$

How will this be used to analyze planted clique? When we write down the SDP for planted clique, we will want to bound the optimal objective value of the SDP. One way to demonstrate that a primal solution with objective value z is optimal is to exhibit a separate dual solution that achieves a dual objective of z.

Let us suppose that p^* denotes the optimal primal objective value. As p^* is optimal, we have $p^* \leq z$. If we can exhibit a dual solution with the same objective value as z, then because the dual objective always lower-bounds the primal objective value, we have that $p^* \geq z$. Consequently, $p^* = z$ certifying that z is optimal! This technique is sometimes called *dual-fitting* and is commonly used to analyze the approximation ratio for algorithms based off of LP relaxations.

5.3 Planted Clique SDP Relaxation

In lecture 3, we introduced the planted clique model as follows:

Planted Clique Model
A graph sampled from the k-planted clique model is that constructed by

1. Sample $G \sim \mathcal{G}_{n,1/2}$.
2. For $G = (V, E)$, select $S \subseteq V$ where $|S| = k$ uniformly at random.
3. Plant a clique on vertices in S by connecting each vertex pair with an edge.

then provided a spectral algorithm due to Alon, Krivelevich, and Sudakov to recover S provided that $k \geq O(\sqrt{n})$. Another way of recovering a planted clique is to leverage the power of semi-definite programming. In particular, we will seek to demonstrate the following result due to Feige and Kilian:

Theorem 5.1. *There exists a constant $c > 0$ such that for the k-planted clique model where $k \geq c\sqrt{n}$, with high probability there exists an SDP relaxation for the planted clique problem with optimal value k.*

This theorem only discusses the optimal objective value when a more natural goal would be to actually recover the planted clique. There are many rounding procedures that one can use to achieve this. For example, one can look at the second largest eigenvector of SDP solution X, or solve the equivalent vector program solution and apply hyperplane rounding.

5.3.1 Primal SDP

We first need to exhibit an SDP relaxation of the planted clique problem and to do so, we follow the same steps taken to derive the SDP for max-cut in lecture 4. First, let's formulate a quadratic program for the problem.

Instead of directly solving planted clique, consider the equivalent question of finding the largest independent set on the complement of $G = (V, E)$ sampled from the planted clique model. We would like our optimal solution to take the form $z_i = 1$ if $i \in V$ is in the independent set and $z_i = 0$ otherwise. To enforce integrality, we add the constraint $z_i^2 = z_i$. We also want to enforce independence (i.e. no two vertices in the independent set are connected by an edge) so we add the constraint $z_i z_j = 0$ for all $(i, j) \in E$. Finally, we wish to maximize the size of the independent set or, equivalently, the sum of all z_i. Our initial quadratic program is

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in V} z_i \\ \text{subject to} \quad & z_i z_j = 0 \quad \forall (i,j) \in E \\ & z_i^2 = z_i \quad\ \ \forall i \in V \end{aligned} \tag{5.3}$$

The next step is to relax this into a vector program, but to do so, it would be easiest to have all summands in the program be quadratic. This way we can simply replace $z_i z_j$ with $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$ for a collection of $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$. Convert the objective function to $\left(\sum_{i \in V} z_i\right)^2 = \sum_{i,j \in V} z_i z_j$. The independence constraint remains the same, but we now need to replace the integrality constraint with $\sum_{i \in V} z_i^2 = 1$. Our quadratic program is now.

$$\begin{aligned} \text{maximize} \quad & \sum_{i,j \in V} z_i z_j \\ \text{subject to} \quad & z_i z_j = 0 \qquad \forall (i,j) \in E \\ & \sum_{i \in V} z_i^2 = 1 \end{aligned} \tag{5.4}$$

One can show that program 5.4 also solves the max independent set problem by arguing that a solution z^* to 5.4 is optimal if and only if it has the form

$$z_i^* = \begin{cases} \frac{1}{\sqrt{|S|}} & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

where S is the largest independent set of G. We can now relax the quadratic program into a vector program. Create vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$ and replace each instance of $z_i z_j$ with $\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$ to derive

$$\begin{aligned} \text{maximize} \quad & \sum_{i,j \in V} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle \\ \text{subject to} \quad & \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 \qquad \forall (i,j) \in E \\ & \sum_{i \in V} \|\boldsymbol{v}_i\|_2^2 = 1 \\ & \boldsymbol{v}_i \in \mathbb{R}^n \qquad\qquad \forall i \in V \end{aligned} \tag{5.6}$$

To derive the final SDP, we utilize the fact that if an $n \times n$ matrix $X \succeq 0$ then there exist $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n \in \mathbb{R}^n$ such that $X_{ij} = \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle$. The SDP for independent set is thus the following.

$$\begin{aligned} \text{maximize} \quad & \sum_{i,j \in V} X_{ij} \\ \text{subject to} \quad & X_{ij} = 0 \qquad \forall (i,j) \in E \\ & \sum_{i \in V} X_{ii} = 1 \\ & X \succeq 0 \end{aligned} \tag{5.7}$$

5.3.2 Dual SDP

Let's now consider the dual for this SDP. We will follow the same procedure outlined by the first section to derive the dual to program 5.7. Let's first rewrite the objective of program 5.7 into an equivalent form

$$\text{minimize} \quad \langle -J, X\rangle$$

where J denotes the $n \times n$ all-ones matrix. How did we derive the dual SDP? We first replaced $X \succeq 0$ with $v^\top X v \geq 0$ for each $v \in \mathbb{R}^n$ and applied multipliers to each constraint

$$y_{ij} \ : \ y_{ij}x_{ij} = 0 \qquad\qquad t \ : \ t\Big(\sum_{i\in V} x_{ii}\Big) = t \qquad\qquad c_v \ : \ c_v \boldsymbol{v}^\top X \boldsymbol{v} \geq 0$$

enforcing $c_v \geq 0$ to maintain the direction of the inequality. Next, we summed all constraints into a single inequality. This derives:

$$\sum_{i,j} y_{ij}x_{ij} + t\Big(\sum_{i\in V} x_{ii}\Big) + \sum_{v\in\mathbb{R}^n} c_v \boldsymbol{v}^\top X \boldsymbol{v} \geq t$$

Let's reformulate this inequality using the matrix inner-product. Define Y as the $n\times n$ matrix where $Y_{ij} = y_{ij}$ if $(i,j) \in E$ and 0 otherwise. The above inequality is thus

$$\langle Y, X\rangle + t\langle I, X\rangle + \Big\langle \sum_{v\in\mathbb{R}^n} c_v \boldsymbol{v}\boldsymbol{v}^\top, X\Big\rangle \geq t$$
$$\Big\langle Y + tI + \sum_{v\in\mathbb{R}^n} c_v \boldsymbol{v}\boldsymbol{v}^\top, X\Big\rangle \geq t$$

Equating $Y + tI + \sum_{v\in\mathbb{R}^n} c_v \boldsymbol{v}\boldsymbol{v}^\top = -J$ gives us our preliminary dual.

$$\begin{aligned} &\text{maximize} && t \\ &\text{subject to} && Y + tI + \sum_{v\in\mathbb{R}^n} c_v \boldsymbol{v}\boldsymbol{v}^\top = -J \end{aligned}$$

As noted above, $\sum_{v\in\mathbb{R}^n} c_v \boldsymbol{v}\boldsymbol{v}^\top$ is just some PSD matrix Z. We can again set $\sum_{v\in\mathbb{R}^n} c_v \boldsymbol{v}\boldsymbol{v}^\top = Z$ thus $Z = -J - Y - tI$. Enforcing $Z \succeq 0$ and replacing with a minimization, the dual program becomes

$$\begin{aligned} &\text{minimize} && -t \\ &\text{subject to} && -J - Y - tI \succeq 0 \end{aligned} \tag{5.8}$$

5.3.3 Dual as an Eigenvalue Minimization Problem

The dual SDP 5.8 simplifies quite nicely to a program on the eigenvalues of the matrix $J + Y$. Consider that the constraint $-J - tI - Y \succeq 0$ is equivalent to $-tI \succeq J + Y$, which states that the smallest eigenvalue

$\lambda_n(-tI)$ of $-tI$ is at least as large as the largest eigenvalue $\lambda_1(J+Y)$ of $J+Y$. However, the only eigenvalue of $-tI$ is $-t$, thus t in the dual SDP is feasible if and only if

$$-t \geq \lambda_1(J+Y)$$

Since we are minimizing over $-t$, an optimal solution must admit $-t = \lambda_1(J+Y)$. Let's define $M = J+Y$ and consider what M could look like. Recall that $y_{ij} = 0$ if $(i,j) \notin E$ and otherwise y_{ij} is unconstrained. It is equivalent to replace the positive semidefinite constraint with $M_{ij} = 1$ when $(i,j) \notin E$. In summary, the dual SDP for independent set is equivalent to the eigenvalue minimization problem

$$\begin{array}{ll} \text{minimize} & \lambda_1(M) \\ \text{subject to} & M_{ij} = 1 \quad \forall (i,j) \notin E \end{array} \tag{5.9}$$

Considering the dual SDP in this form will be critical to our proof of theorem 5.1, which we will now see!

5.4 Analysis of the SDP

To prove theorem 5.1, we first demonstrate that there exists a primal feasible solution that achieves objective value $k = |S|$ the size of the maximum independent set. Working from the vector program, assigning $\boldsymbol{v}_i$'s to the following satisfies the constraints and has objective value k:

$$\boldsymbol{v}_i = \begin{bmatrix} 1/\sqrt{k} & 0 & \cdots & 0 \end{bmatrix}^\top \quad \text{if } i \in S \qquad\qquad \boldsymbol{v}_i = \begin{bmatrix} 0 & 0 & \cdots & 0 \end{bmatrix}^\top \quad \text{otherwise}$$

Next, we exhibit a dual feasible M such that $\lambda_1(M) = k$ certifying that k is the primal optimal objective. Let's start by analyzing the structure of M. To simplify our analysis of M, we can first, without loss of generality, label the vertices such that those in the independent set are 1 to k. The matrix then looks like

$$M = \left[\begin{array}{c:c} J_k & \sim 50\% \text{ 1's} \\ \hdashline \sim 50\% \text{ 1's} & \sim 50\% \text{ 1's} \end{array}\right] \tag{5.10}$$

where it $i,j \in S$ then $M_{ij} = 1$, otherwise roughly $1/2$ of the $M_{ij} = 1$ (since $G \sim \mathcal{G}_{n,1/2}$) and the others have $M_{ij} = y_{ij}$ variables over whose values we are maximizing. Since y_{ij} are unconstrained, we have the freedom to set y_{ij} to anything we want.

How can we set y_{ij}'s such that M admits some eigenvector corresponding to eigenvalue k the size of the max independent set? A promising candidate eigenvector is $\boldsymbol{v} = [1, \ldots, 1, 0, \ldots, 0]^\top$ – the 0-1 indicator on the independent set with k ones and $n-k$ zeroes. The first k elements of $M\boldsymbol{v}$ are then all k, so for $\boldsymbol{v}$ to indeed be an eigenvector of M the rest must all be 0.

Let's now choose our y_{ij}'s. Say for a given row of M that ℓ of the first k values are equal to one. If each of the y_{ij} in that row among the first k columns are assigned value $-\frac{\ell}{k-\ell}$, then the first k elements of the row sum to 0. Since $\boldsymbol{v}$ is only nonzero at the first k entries, we have that $M\boldsymbol{v} = k\boldsymbol{v}$ or M has an eigenvalue $\lambda = k$. Note that the y_{ij} values corresponding to the last $n-k$ columns have not been assigned as they are irrelevant in $M\boldsymbol{v}$. To aid in the rest of the analysis, let's assign values of -1 to the bottom-right submatrix of 5.10 and symmetric values $-\frac{k-l}{k}$ to the top-right. Visually, M is now assigned to

$$M = \left[\begin{array}{c:c} J_k & 0 \\ \hdashline 0 & 0 \end{array}\right] + \left[\begin{array}{c:c} 0 & 1 \text{ or } -\frac{k-l}{k} \\ \hdashline 1 \text{ or } -\frac{k-l}{k} & 0 \end{array}\right] + \left[\begin{array}{c:c} 0 & 0 \\ \hdashline 0 & 1 \text{ or } -1 \end{array}\right]$$

where we have split M into the sum $M = M_1 + M_2 + M_3$. We want this M to have objective value k so let's show that k is in fact the largest eigenvalue of M by first applying the following lemma[1]

Lemma 5.2. For $n \times n$ symmetric matrices X, Y with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ and $\mu_1 \geq \ldots \geq \mu_n$ respectively, the eigenvalues $\nu_1 \geq \ldots \geq \nu_n$ of $X + Y$ each satisfy

$$\lambda_i + \mu_n \leq \nu_i \leq \lambda_i + \mu_1$$

which will allow us to control the spectrum of M as we add one component to the next. In particular, a direct application of lemma 5.2 for $X = M_1$ and $Y = M_2 + M_3$ admits

$$\lambda_2(M) \leq \lambda_2(M_1) + \lambda_1(M_2) + \lambda_1(M_3)$$

The matrix M_1 has rank one, so $\lambda_2(M_1) = 0$. On the other hand, M_2 and M_3 are random matrices and so we require a theorem of Vu [1] on the spectrum of symmetric random matrices to bound their eigenvalues:

Theorem 5.3. *If P is a random matrix such that*

i. P is symmetric with zeroes on the diagonal with probability 1

ii. For all entries P_{ij}, $\mathbb{E}[P_{ij}] = 0$ and $-1 \leq P_{ij} \leq 1$

iii. The entries P_{ij} are mutually independent.

then there is a constant $c > 0$ such that with high probability $\|P\| \leq c\sqrt{n}$ where $\|\cdot\|$ denotes the spectral norm.

Applying Theorem 5.3, we have that with high probability $\lambda_1(M_2) \leq c\sqrt{n}$ and $\lambda_1(M_3) \leq c\sqrt{n-k}$. (The matrix M_3 does not have zeroes along the diagonal, but the theorem can be adjusted such that the bound still holds). We thus have w.h.p. $\lambda_2(M) \leq C\sqrt{n}$ for some absolute constant C and, because we assume $k \geq O(\sqrt{n})$, we have $\lambda_2(M) < k$. Consequently, $\lambda_1(M) = k$ and so k is indeed feasible in the dual SDP. From what was discussed in section 5.2, $|S| = k$ is the optimal in the primal program thereby completing the argument for theorem 5.1.

5.5 SDP and the Monotone Adversary

We just saw that an SDP can be used to determine the exact size of a planted clique, a task which can also be accomplished by the spectral algorithm from Lecture 3. However, the SDP can be said to be more powerful

[1]We delegate the proof to the appendix

than this algorithm, in the sense that it is correct even for *semi-random* input, while the AKS algorithm does not have this guarantee.

In the semi-random model, nature generates a random input, e.g. a planted clique, which an adversary then modifies. Obviously the power of the adversary must be limited, otherwise this would be no different from worst-case analysis. Here, the adversary is *monotone* – that is, it is only allowed to add edges to dense areas of the graph and remove edges from sparse areas. In particular, the semi-random planted clique model is as follows:

Semi-random Planted Clique Model

1. Sample G from the planted clique model with clique S.
2. Arbitrarily remove a subset from $\{(i,j) \in E : i \notin S \vee j \notin S\}$, the edges not in the clique.

Intuitively, the spectral algorithm relies on the distribution of the adjacency matrix of G having a specific expectation and being concentrated tightly around this mean, and thus breaks when a monotone adversary is introduced. The SDP makes fewer such assumptions, and can be expected to work even in the semi-random case.

We can show this correctness more rigorously. Again, for convenience we will consider independent set, which is a clique in the complement of G. Given input graph G, define $h(G)$ to be the optimal value of the SDP relaxation and $b(G) = |S|$ the actual size of the independent set. In the semi-random model, nature generates a planted clique input G_0, from which the monotone adversary creates successive $G_1, \dots, G_n$ by adding one edge at a time outside the independent set (equivalently, removing edges outside the clique). Then, we can show the following:

Lemma 5.4. For each $1 \le i \le n$, $h(G_i) = b(G_i)$.

Proof. We show this by induction. The base case $h(G_0) = b(G_0)$ follows directly from Theorem 5.1. Suppose for some $1 \le k \le n$ that $h(G_k) = b(G_k)$. Note that adding an edge to any graph serves only to add one constraint to the SDP relaxation and does not change the objective function, meaning this objective can only decrease. (Any feasible solution in the more-constrained SDP must also be feasible in the original.) Meanwhile, since the monotone adversary only adds edges outside of the independent set, it does not affect the solution. Thus, $h(G_{k+1}) \le h(G_k)$ and $b(G_{k+1}) = b(G_k)$, and therefore $h(G_{k+1}) \le h(G_k) = b(G_k) = b(G_{k+1})$. However, since the SDP is a relaxation of the exact quadratic program, $h(G) \ge b(G)$ for any graph G, so we conclude $h(G_{k+1}) = b(G_{k+1})$. Thus, by induction, $h(G_i) = b(G_i)$ for any $1 \le i \le n$. □

From this, we conclude that the SDP is correct in the semi-random model.

References

[1] Van H. Vu, *Spectral norm of random matrices*, Combinatorica **27** (2007), no. 6, 721–736.

A Additional Proofs

A.1 Correctness of the Quadratic Program

For the SDP to be correct, it is necessary that it be a relaxation of an exact quadratic program. In other words,

Lemma 5.5. The size of the independent set $k = |S|$ is optimal in the quadratic program 5.4.

Proof. It is easy to check that the assignment in equation 5.5 is feasible with objective value k. To see that this is optimal, note first that by construction of the quadratic program the number of nonzero z_i is maximized when $z_i = 0$ iff i is not in the independent set. Once the nonzero variables are selected, they are constrained only by $\sum_{i \in V} z_i^2 = 1$. That is, the vector $\boldsymbol{z}'$ of nonzero z_i's must lie on the unit sphere. Expressing the objective function as $(\sum_{i \in V} z_i)^2$, it is clear that the objective increases with $|\sum_{i \in V} z_i| = \langle \boldsymbol{z}', \mathbf{1} \rangle$ (since $x \longmapsto x^2$ increases monotonically for positive x). This inner product is maximized at $\boldsymbol{z}' = \pm \frac{1}{\sqrt{k}} \mathbf{1}$, i.e. when all nonzero z_i are equal. Since the assignment 5.5 satisfies this property, it must be optimal. $\square$

A.2 Spectrum of a Sum of Matrices

The argument for lemma 5.2 utilizes the variational characterization of eigenvalues. Observe

Lemma 5.2. For $n \times n$ symmetric matrices X, Y with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ and $\mu_1 \geq \ldots \geq \mu_n$ respectively, the eigenvalues $\nu_1 \geq \ldots \geq \nu_n$ of $X + Y$ each satisfy

$$\lambda_i + \mu_n \leq \nu_i \leq \lambda_i + \mu_1$$

Proof. Recall that the variational characterization of eigenvalues admits the following

$$\mu_1 = \max_{\substack{\boldsymbol{x} \in \mathbb{R}^n \\ \boldsymbol{x} \neq 0}} \frac{\boldsymbol{x}^\top Y \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \qquad \mu_n = \min_{\substack{\boldsymbol{x} \in \mathbb{R}^n \\ \boldsymbol{x} \neq 0}} \frac{\boldsymbol{x}^\top Y \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \qquad \lambda_i = \max_{\boldsymbol{x} \in V_i} \frac{\boldsymbol{x}^\top X \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}$$

where V_i is the subspace orthogonal to the first $i-1$ eigenvectors of X. Also,

$$\nu_i = \max_{\boldsymbol{x} \in V_i} \frac{\boldsymbol{x}^\top (X+Y) \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = \max_{\boldsymbol{x} \in V_i} \left(\frac{\boldsymbol{x}^\top X \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} + \frac{\boldsymbol{x}^\top Y \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \right)$$

To demonstrate the lower-bound, observe

$$\begin{aligned} \lambda_i + \mu_n &= \max_{\boldsymbol{x} \in V_i} \left(\frac{\boldsymbol{x}^\top X \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \right) + \min_{\substack{\boldsymbol{x} \in \mathbb{R}^n \\ \boldsymbol{x} \neq 0}} \left(\frac{\boldsymbol{x}^\top Y \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \right) \\ &\leq \max_{\boldsymbol{x} \in V_i} \left(\frac{\boldsymbol{x}^\top X \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \right) + \min_{\boldsymbol{x} \in V_i} \left(\frac{\boldsymbol{x}^\top Y \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \right) \end{aligned}$$

$$\begin{aligned}
&\le \max_{\boldsymbol{x}\in V_i}\left(\frac{\boldsymbol{x}^\top X\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}\right)+\min_{\boldsymbol{x}\in V_i}\left(\frac{\boldsymbol{x}^\top Y\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}\right)\\
&=\max_{\boldsymbol{x}\in V_i}\left(\frac{\boldsymbol{x}^\top X\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}+\frac{\boldsymbol{x}^\top Y\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}\right)=\nu_i
\end{aligned}$$

And finally to demonstrate the upper-bound:

$$\begin{aligned}
\nu_i &= \max_{\boldsymbol{x}\in V_i}\left(\frac{\boldsymbol{x}^\top X\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}+\frac{\boldsymbol{x}^\top Y\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}\right)\\
&\le \max_{\boldsymbol{x}\in V_i}\left(\frac{\boldsymbol{x}^\top X\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}\right)+\max_{\boldsymbol{x}\in V_i}\left(\frac{\boldsymbol{x}^\top Y\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}\right)\\
&\le \max_{\boldsymbol{x}\in V_i}\left(\frac{\boldsymbol{x}^\top X\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}\right)+\max_{\substack{\boldsymbol{x}\in\mathbb{R}^n\\ \boldsymbol{x}\neq 0}}\left(\frac{\boldsymbol{x}^\top Y\boldsymbol{x}}{\boldsymbol{x}^\top\boldsymbol{x}}\right)=\lambda_i+\mu_1
\end{aligned}$$

which immediately implies the desired result. □

A.3 Spectrum of a Symmetric Random Matrix

Before we proceed, let us recall Hoeffding's inequality which bounds the probability that a sum of independent random variables deviates from its mean.

Theorem 5.6. *Let $X_1,\ldots,X_n$ be independent random variables with $a_i \le X_i \le b_i$ for each $1 \le i \le n$. Then, for $S=\sum_{i=1}^n X_i$,*

$$P(|S-\mu|\ge t)\le 2e^{-t^2/2\sum_{i=1}^n(b_i-a_i)^2}$$

where $\mu=\mathbb{E}[S]$. □

Now, the full proof of Theorem 5.3 is slightly involved and so instead we show a weaker claim while highlighting the use of ε-nets to prove concentration bounds on random matrices:

Theorem 5.7. *If P is a random matrix with the same properties as in Theorem 5.3, then there is a constant $c>0$ such that, for every $\delta>0$, $\|P\|\le c\sqrt{n\log n\log 1/\delta}$ with probability at least $1-\delta$.*

Proof. For any fixed unit vector $\boldsymbol{u}$, consider the quadratic form $\boldsymbol{u}^\top P\boldsymbol{u}=\sum_{i,j}u_iu_jP_{ij}$. Since by assumption $-1\le P_{ij}\le 1$, each $u_iu_jP_{ij}$ is bounded by $[-u_iu_j,u_iu_j]$, with the bound being $[0,0]$ if $i=j$ since P is zero on the diagonals. Note that $(\langle\boldsymbol{u},\boldsymbol{u}\rangle)^2=1=\sum_{i=1}^n u_i^4+\sum_{i\neq j}u_i^2u_j^2$, so $\sum_{i\neq j}(2u_iu_j)^2\le 4$. Then, since by assumption $\mathbb{E}[\sum_{i,j}u_iu_jP_ij]=0$, using the Hoeffding inequality we have

$$P(|\boldsymbol{u}^\top P\boldsymbol{u}|\ge t)\le 2e^{-t^2/8}$$

However, this in itself is not enough to show the result, as we want a bound over *all* unit vectors. We achieve this using an ε-net over the unit sphere in the following lemmas. (An ε-net of a metric space (M,d) is a subset $S\subseteq M$ such that $\forall m\in M,\exists s\in S$ with $d(m,s)\le\varepsilon$.)

Lemma 5.8. The unit sphere in $\mathbb{R}^n$ admits an ε-net N with $|N| \approx (2/\varepsilon)^n$.

Proof. Let Δ be a maximal ε-separated subset of the unit n-sphere – that is, for all $x, y \in \Delta$ such that $x \neq y$, their distance $d(x, y) > \varepsilon$, and there is no other point in the sphere that can be inserted into Δ without breaking this condition. Note that Δ is also an ε-net, for if any point x is not covered by Δ, then $\forall y \in \Delta, d(x, y) > \varepsilon$ and x can be added to Δ while preserving separation, contradicting the maximality of Δ. Now consider placing $\varepsilon/2$-radius n-spheres centered at each point in Δ. By the separation of Δ, these are all disjoint, and are all contained in the $(1+\varepsilon/2)$-radius n-sphere centered at the origin. This means Δ can have size at most $\dfrac{(1+\varepsilon/2)^n}{(\varepsilon/2)^n} \approx (2/\varepsilon)^n$. $\square$

Lemma 5.9. If $|\boldsymbol{u}^\top P \boldsymbol{u}| \leq t$ for every $\boldsymbol{u}$ in the ε-net of an n-sphere, then $|\boldsymbol{u}^\top P \boldsymbol{u}| \leq t + n\varepsilon$ for all unit vectors in $\mathbb{R}^n$.

Proof. Trivially, since $-1 \leq P_{ij} \leq 1$, we know that $||P|| \leq n$. Also, given an ε-net N of the unit sphere in $\mathbb{R}^n$, any unit vector $\boldsymbol{u}' \in \mathbb{R}^n$ can be expressed as $\boldsymbol{u} + \boldsymbol{v}$ where $\boldsymbol{u} \in N$ and $||\boldsymbol{v}|| \leq \varepsilon$. By the triangle inequality,

$$\boldsymbol{u}'^\top P \boldsymbol{u}' = (\boldsymbol{u} + \boldsymbol{v})^\top P (\boldsymbol{u} + \boldsymbol{v}) \leq \boldsymbol{u}^\top P \boldsymbol{u} + \boldsymbol{v}^\top P \boldsymbol{v} \leq t + n\varepsilon$$

As required. $\square$

Now, take ε to be $1/n$. Then by Lemma 5.8 the ε-net of the unit sphere is of size $\sim (2n)^n$. Union bounding over the ε-net, we have that $P(|\boldsymbol{u}^\top P \boldsymbol{u}| > t) \leq (2n)^n 2e^{-t^2/8}$. By setting $t = \Omega(\sqrt{n \log n \log 1/\delta}$, this probability is less than δ. Thus, using Lemma 5.9, we have that $||P|| \leq c\sqrt{n \log n \log 1/\delta}$ with probability at least $1 - \delta$. $\square$