Grothendieck problems on graphs

Talk by Nitya Mani
Notes by Sanjana Das
March 22, 2024

§1 Introduction

Today, we'll study the following problem on graphs (which we allow to have self-loops).

Definition 1.1. For a graph $G = (V, E)$, the Grothendieck constant of G, which we denote $K(G)$, is defined as the infimum of all K such that for every matrix $A = (A_{uv})$ we have

$$\sup_{\|f_u\|=1} \sum_{uv \in E} A_{uv} \langle f_u, f_v \rangle \leq K \sup_{z_u \in \{\pm 1\}} \sum_{uv \in E} A_{uv} z_u z_v. \tag{1}$$

In words, on the right-hand side we're trying to choose a sign $z_u \in \{\pm 1\}$ for each vertex u in G in order to maximize $A_{uv} z_u z_v$ over all edges of G; this is some optimization problem. And on the left-hand side we're considering a *semidefinite relaxation* of this optimization problem where instead of choosing a sign $z_u \in \{\pm 1\}$ for each vertex u, we're choosing a *vector* $f_u \in \mathbb{R}^d$ (for some arbitrary d) with norm 1. (And then we're replacing the product $z_u z_v$ with the inner product $\langle f_u, f_v \rangle$.) This will increase the maximum sum we can get, because we've only given ourselves more choices (we can think of the right-hand side as the special case of the left-hand side with $d = 1$). And we define $K(G)$ as the smallest constant K such that we can say this relaxation only increases the sum by a factor of K.

Remark 1.2. We can assume that $A_{uv} = 0$ for all $uv \notin E$, so that the right-hand side of (1) can be written as $z^\intercal A z$ where $z = (z_u)$; we'll often do this for notational convenience.

The reason the optimization problem on the left-hand side is called a *semidefinite* optimization problem is that it's equivalent to trying to maximize $\sum_{uv \in E} A_{uv} x_{uv}$ where X is a positive semidefinite matrix with diagonal entries 1 corresponding to the Gram matrix of the vectors f_u — so its entries are $X_{uv} = \langle f_u, f_v \rangle$. For this reason, we'll use the following notation.

Notation 1.3. Given G and A, we use SDP to denote the value of the supremum on the left-hand side of (1), and OPT to denote the value of the supremum on the right-hand side.

Then we have $\mathsf{SDP} \geq \mathsf{OPT}$, and proving upper bounds on $K(G)$ (which will be our focus for today) corresponds to proving inequalities in the reverse direction.

We'll also define a related notion, where we replace the condition $z_u \in \{\pm 1\}$ with $z_u \in [-1, 1]$ (and correspondingly replace $\|f_u\| = 1$ with $\|f_u\| \leq 1$).

Definition 1.4. For $G = (V, E)$, we define $K'(G)$ as the infimum of all K' such that for every matrix A,

$$\sup_{\|f_u\| \leq 1} \sum_{uv \in E} a_{uv} \langle f_u, f_v \rangle \leq K \sup_{z_u \in [-1,1]} \sum_{uv \in E} A_{uv} z_u z_v.$$

If G is simple (i.e., it has no self-loops), then $K'(G) = K(G)$ — the right-hand side is a multilinear form in the variables z_u, so if we want to maximize it, it's optimal to take each z_u to be an endpoint of $[-1, 1]$.

Here's one result regarding $K(G)$.

Theorem 1.5 (Alon–Makarychev–Makarychev–Naor)
For any simple graph G, we have
$$\log \omega(G) \lesssim K(G) \lesssim \vartheta(\overline{G}).$$

(We use ϑ to denote the Lovász theta number, which satisfies $\omega(G) \leq \vartheta(\overline{G}) \leq \chi(G)$.)

Today we're going to talk about upper bounds on $K(G)$ — we'll highlight the main ideas, but won't focus on getting the optimal constants. We'll consider three settings — the cases where G is bipartite, where G is a clique, and where G is a general graph.

§2 Bipartite graphs

We'll first consider the case where G is bipartite.

The best-known upper bound in this case is the following.

Theorem 2.1 (Krivine, Braverman–Makarychev–Makarychev–Naor)
For all bipartite G, we have
$$K(G) \leq \frac{\pi}{2\operatorname{arcsinh}(1)} - \varepsilon$$
(where $\varepsilon > 0$ is a small non-explicit constant).

One reason this is interesting is that the first term of $\frac{\pi}{2\operatorname{arcsinh}(1)}$ comes out of a natural semidefinite argument that's believed to be optimal for some problems, but the $-\varepsilon$ term means that it isn't optimal here.

§2.1 A first bound

First we're going to prove the following bound, which shows that $K(G) \leq (\frac{4}{\pi} - 1)^{-1}$.

Theorem 2.2 (Nesterov–Ye, Rielz)
We have $\mathsf{OPT} \geq (\frac{4}{\pi} - 1)\mathsf{SDP}$.

Let $\{f_u\}$ be the optimal collection of vectors for the SDP (i.e., the optimization problem on the left-hand side of (1)), and let X be their Gram matrix (defined by $X_{uv} = \langle f_u, f_v \rangle$ for all u and v).

At a very high level, the strategy for proving bounds of this form (where we lower-bound OPT in terms of SDP) is that we want to use the solution $\{f_u\}$ to the SDP to construct a feasible solution for the original optimization problem — i.e., a collection of ± 1-values $\{z_u\}$ — which is almost as good. This means we want a rounding scheme where we turn each vector f_u into a sign z_u.

To prove Theorem 2.2, we use the following rounding scheme — we first choose g to be a standard Gaussian vector (so g is a random vector in $\mathbb{R}^d$, where d is the dimension that the vectors f_u live in). Then we define
$$y_u = g^\intercal f_u \in \mathbb{R} \text{ and } z_u = \operatorname{sgn}(y_u) \in \{-1, 1\}.$$

(So we're choosing a random Gaussian vector g, turning the vectors f_u into real numbers y_u by projecting them onto g, and then rounding these real numbers to $\{\pm 1\}$-values using the sgn function.)

Fact 2.3 — We have $\mathbb{E}[yy^\intercal] = X$.

Here y denotes the (random) vector with entries y_u for all $u \in V$; this is some Gaussian vector (and in particular, each of its entries has distribution $\mathcal{N}(0,1)$), and it's not hard to check that its covariance matrix is X (which is what the above fact is saying).

And now we have $\mathsf{OPT} \geq \mathbb{E}_g[z^\intercal Az]$ (because for any choice of g, the resulting $z = (z_u)$ is a feasible solution to the optimization problem on the right-hand side of (1), with value $z^\intercal Az$; so $\mathsf{OPT} \geq z^\intercal Az$ for *every* g). So our goal is now to lower-bound $\mathbb{E}_g[z^\intercal Az]$ in terms of the value of the SDP.

The way we're going to do this is by a 'first-order approximation' strategy — we rewrite $\mathbb{E}_g[z^\intercal Az]$ as

$$\mathbb{E}_g[z^\intercal Az] = \mathbb{E}_g\left[(\gamma y + (z-y))^\intercal A(\gamma y + (z-y))\right], \tag{2}$$

where $\gamma = \sqrt{2/\pi}$. The reason for this choice of γ is the following calculation.

Fact 2.4 — If g_1 and g_2 are jointly Gaussian real numbers with mean 0 and variance 1, then

$$\mathbb{E}[g_1 \cdot \operatorname{sgn}(g_2)] = \gamma\mathbb{E}[g_1 \cdot g_2].$$

Proof. This holds in the case $g_1 = g_2$ (it's a fact that if $g_1 \sim \mathcal{N}(0,1)$ then $\mathbb{E}\,|g_i| = \gamma$) and the case where g_1 and g_2 are independent (where both sides are 0). And we can deduce the general statement from these two cases by writing g_1 as a linear combination of g_0 and g_2, where g_0 and g_2 are independent Gaussians — since this equality essentially respects taking linear combinations (for g_1). ☐

This means all the cross-terms in (1) disappear (meaning the terms $\mathbb{E}_g[y_uA_{uv}(z_v - y_v)]$), and we're left with

$$\mathbb{E}_g[z^\intercal Az] = \gamma^2\mathbb{E}_g[y^\intercal Ay] + \mathbb{E}_a[(z-\gamma y)^\intercal A(z-\gamma y)]. \tag{3}$$

And the first term $\mathbb{E}_g[y^\intercal Ay]$ exactly corresponds to SDP, since we saw that the covariance matrix of y is exactly X (so $\mathbb{E}[a_{uv}y_uy_v] = a_{uv}\langle f_u, f_v\rangle$ for all $u, v \in V$).

Meanwhile, it turns out that we can construct a feasible solution to the SDP from the second term of (3) as well! The matrix Y with entries $Y_{uv} = \mathbb{E}_a[(z_u - \gamma y_u)(z_v - \gamma y_v)]$ is positive semidefinite (since for any a the matrix with entries $(z_u - \gamma y_u)(z_v - \gamma y_v)$ is PSD — it's the 'Gram matrix' of a bunch of real numbers — and the expectation of a bunch of PSD matrices is still PSD), and any PSD matrix with diagonal entries 1 corresponds to a feasible solution to the SDP. So we can normalize Y to get a feasible solution — we have

$$\mathbb{E}[(z_u - \gamma y_u)^2] = 1 - \gamma^2$$

for every u (by a direct calculation using Fact 2.4), which means the matrix $Y' = (1-\gamma^2)^{-1}Y$ is a feasible solution to the SDP.

We're trying to get a *lower* bound on the right-hand side of (3), which means we want to know the *worst-case* value of the SDP (i.e., the minimum value of the left-hand side of (1), rather than the maximum). And this is the only place where we use bipartiteness — because G is bipartite, we know the worst-case feasible solution is the negative of the best (if we negate the vectors f_u for all u on one side of the vertex partition of G, this will negate each term $A_{uv}\langle f_u, f_v\rangle$). And so we get the bound

$$\mathbb{E}_a[(z-\gamma y)^\intercal A(z-\gamma y)] \geq -(1-\gamma^2)\mathsf{SDP}.$$

Combining the bounds on the two terms in (3), we get

$$\mathsf{OPT} \geq \mathbb{E}_g[z^\intercal Az] \geq \gamma^2\mathsf{SDP} - (1-\gamma^2)\mathsf{SDP} = \left(1 - \frac{4}{\pi}\right)\mathsf{SDP},$$

which proves Theorem 2.2.

Remark 2.5. One way to view this proof is that we want to start with $\mathbb{E}_g[z^\intercal A z]$ and make the SDP appear. We could do this if the z_u's were a linear function of the f_u's. Unfortunately sgn (which our rounding scheme uses) is *not* linear, so we instead try to approximate it by something linear — we take the first-order coefficient of the Hermite expansion and pull it out (this corresponds to the first term in (3)), and hope that the error (corresponding to the second term in (3)) is not too big.

§2.2 A second bound

Now we'll see another method, due to Krivine, that gets the following bound. (Our description will be somewhat different from how Krivine wrote it, which involved infinite-dimensional Hilbert spaces.)

Theorem 2.6
We have $\mathsf{OPT} \geq \frac{2}{\pi}\operatorname{arcsinh}(1)\cdot\mathsf{SDP}$.

We're again going to start with the optimal SDP solution X and try to use it to produce signs $z_u \in \{\pm 1\}$. But this time, the main idea is that we're going to try to first transform X so that when we apply the sgn function, we really *will* get something linear.

Let J be the all-1's matrix, and let J' be the block matrix

$$J' = \begin{bmatrix} +1 & -1 \\ & \\ -1 & +1 \end{bmatrix}$$

(with blocks corresponding to the vertex partition of G), which is PSD. Now we're going to define

$$\widetilde{X} = \frac{1}{2}J \circ (\sinh(\beta X) + \sin(\beta X)) + \frac{1}{2}J' \circ (\sinh(\beta X) - \sin(\beta X)),$$

where $\circ$ denotes the Hadamard (entrywise) product of two matrices, β is a real number we'll pick soon, and $\sin(\beta X)$ is the matrix obtained by applying sin entrywise to βX.

Claim 2.7 — The matrix $\widetilde{X}$ is PSD.

This is because if X and Y are PSD then $X + Y$ is too, and so is $X \circ Y$ by the Schur product theorem.

Remark 2.8. The choice of sin is motivated for reasons we'll see soon, and then the term sinh appears because it's the smallest thing we could add that ensures $\sinh(\beta X) \pm \sin(\beta X)$ are both PSD (the reason this is true is because $\sinh(\beta x) \pm \sin(\beta x)$ only has nonnegative Taylor coefficients).

We want $\widetilde{X}$ to be a feasible solution to the SDP, so we want its diagonals to be 1. And since X has diagonal entries 1, the diagonal entries of $\widetilde{X}$ are all

$$\widetilde{X}_{uu} = \frac{1}{2}(\sinh(\beta) + \sin(\beta)) + \frac{1}{2}(\sinh(\beta) + \sin(\beta)) = \sinh(\beta),$$

so we take $\beta = \operatorname{arcsinh}(1)$ to make this 1. (This is where the $\operatorname{arcsinh}(1)$ in the bound comes from.)

Now $\widetilde{X}$ corresponds to a feasible solution to the SDP (as it's a PSD matrix with diagonal entries 1); we'll let h_u denote the corresponding vector for each vertex u on the left and h'_v the corresponding vector for each vertex v on the right, so that the value of the SDP at $\widetilde{X}$ is $\sum_{uv} A_{uv}\langle h_u, h'_v\rangle$.

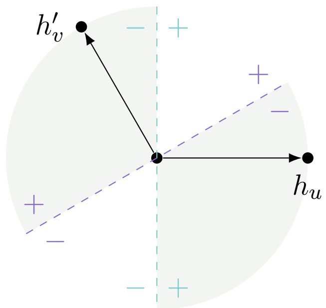
And now we can apply a similar rounding scheme as before, but to these vectors h_u and h'_v — we choose a random vector g which is uniform on the unit sphere, and we define $z_u = \operatorname{sgn}(g^\intercal h_u)$ and $z_v = \operatorname{sgn}(g^\intercal h'_v)$ for each vertex u on the left and v on the right. This gives us some feasible solution to the original optimization problem (i.e., with $\{\pm 1\}$-values), so

$$\mathsf{OPT} \geq \sum_{uv} A_{uv} \mathbb{E}_g[\operatorname{sgn}(g^\intercal h_u) \operatorname{sgn}(g^\intercal h'_v)]. \tag{4}$$

And if we let θ_{uv} be the angle between h_u and h'_v, then

$$\mathbb{P}_g[\operatorname{sgn}(g^\intercal h_u) \neq \operatorname{sgn}(g^\intercal h'_v)] = \frac{\theta_{uv}}{\pi}.$$

This can be seen geometrically — we can consider the plane through h_u and h'_v, and imagine drawing the perpendiculars to the two vectors. Then $\operatorname{sgn}(g^\intercal h_u)$ and $\operatorname{sgn}(g^\intercal h'_v)$ depend on which sides of these perpendiculars g lies on, and we get two sectors with angles θ_{uv} on which the two signs are different.



This means we have

$$\mathbb{E}_g[\operatorname{sgn}(g^\intercal h_u) \operatorname{sgn}(g^\intercal h'_v)] = 1 \cdot \left(1 - \frac{\theta_{uv}}{\pi}\right) + (-1) \cdot \frac{\theta_{uv}}{\pi} = 1 - \frac{2}{\pi}\theta_{uv}.$$

And we can write this in terms of $\widetilde{X}_{uv}$ — by definition $\widetilde{X}_{uv} = \langle h_u, h'_v \rangle$, so $\theta_{uv} = \arccos(\widetilde{X}_{uv})$, which means

$$\mathbb{E}_g[\operatorname{sgn}(g^\intercal h_u) \operatorname{sgn}(g^\intercal h'_v)] = \frac{2}{\pi} \arcsin(\widetilde{X}_{uv}),$$

and plugging this into (4) gives

$$\mathsf{OPT} \geq \frac{2}{\pi} \sum_{uv} A_{uv} \arcsin(\widetilde{X}_{uv}).$$

And now we can see why we defined $\widetilde{X}_{uv}$ to have a sin — we have $\widetilde{X}_{uv} = \sin(\beta X_{uv})$, so we get

$$\mathsf{OPT} \geq \frac{2}{\pi}\beta \sum_{uv} A_{uv} X_{uv} = \frac{2}{\pi}\beta \cdot \mathsf{SDP}.$$

(We defined X to be the optimal solution to the SDP, so $\sum_{uv} A_{uv} X_{uv}$ is SDP by definition.)

Remark 2.9. The way we use bipartiteness here is that when we define $\widetilde{X}$ by transforming X, only the entries on the 'off-diagonal blocks' will affect the value of the solution to the $\{\pm 1\}$-optimization problem we get from rounding $\widetilde{X}$. So we want the values on these off-diagonal blocks to be 'nice' (specifically, we want them to be sin functions, so that the above computation works out), but we don't actually care what happens on the diagonal blocks, and this allows us to put the sinh function in the diagonal blocks (using J') to make $\widetilde{X}$ PSD.

§3 Complete graphs

Now we'll consider complete graphs. This means we've got a generic quadratic form on a graph, and we want to see how far the semidefinite relaxation is from the original problem.

Theorem 3.1 (Nemiraski, Charikar–Wirth, AMMN)
We have $K(K_n) = \Theta(\log n)$.

(There may be more papers that also contemporaneously proved this.)

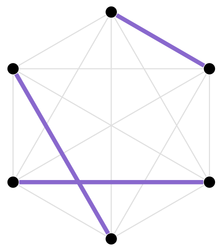
Remark 3.2. We're only going to consider *simple* graphs (i.e., graphs without self-loops). (For graphs with loops, $K(G)$ can become arbitrarily bad.) This means we can use the formulation of $K'(G)$ instead of $K(G)$ (since they're equal for graphs without self-loops); so when we try to use a rounding scheme to produce z_u's from the optimal SDP solution X, we only need to ensure $z_u \in [-1, 1]$ rather than $z_u \in \{\pm 1\}$.

§3.1 A proof

To prove Theorem 3.1 (or at least the upper bound), we'll start with a lemma.

Lemma 3.3
We have $\mathsf{OPT} \geq \frac{1}{n} \sum_{uv} |A_{uv}|$.

Proof. We'll prove this using a probabilistic argument — imagine we construct a random matching of K_n by selecting an edge uniformly at random and deleting it, then selecting another edge uniformly at random and deleting it, and so on.



And imagine we assign each edge uv a weight of $|A_{uv}|$; then the expected total weight of the matching is either $\frac{1}{n} \sum_{uv} |A_{uv}|$ or $\frac{1}{n-1} \sum_{uv} |A_{uv}|$ depending on whether n is even or odd (since it either contains $\frac{n-1}{2}$ or $\frac{n}{2}$ edges, and each edge is equally likely to be present); either way, it's at least $\frac{1}{n} \sum_{uv} |A_{uv}|$, so we can find some matching M with total weight at least this quantity.

Now we want to use M to produce ± 1's (in order to get a feasible solution to the optimization problem). To do so, for each edge $uv \in M$, we first choose $x_u \in \{\pm 1\}$ randomly, and then let $x_v = x_u \operatorname{sgn}(A_{uv})$. (If n is odd, so there's an umatched vertex, then we choose its sign randomly as well.)

Then each $uv \in M$ contributes exactly $|A_{uv}|$ to the value of the optimization problem, while each $uv \notin M$ contributes an expected value of 0 (since the signs of the two vertices are chosen independently). So in expectation the value of the optimization problem we get is the weight of M, which by construction is at least $\frac{1}{n} \sum_{uv} |A_{uv}|$. $\square$

Now we're going to use this to get a rounding scheme — let X be the matrix corresponding to the optimal SDP solution, and factor it as $X_{uv} = \langle f_u, f_v\rangle$ (so the f_u's are the vectors corresponding to the optimal SDP solution). We're again going to sample a standard Gaussian vector g, and define $y_u = g^\intercal f_u$ for each vertex u. This is some real number, but it's not necessarily in $[-1, 1]$, so we need to fix this, and the way we'll do so is by setting

$$z_u = \frac{y_u}{\max_v |y_v|}$$

for each u (so that $z_u \in [-1, 1]$).

How much do we lose by doing this? The point is that if we've got n standard Gaussians, their maximum is expected to be at most on the order of $\sqrt{\log n}$. And Gaussians have nice tails, so we get a bound something like the following.

> **Fact 3.4** — We have $\mathbb{P}[\max_v |g^\intercal f_v| \geq 4\sqrt{2\log 2n} + t] \leq 2\exp(-t^2/2)$.

(It should be possible to prove this by considering the probability that $g^\intercal f_v$ — which is a Gaussian of variance at most 1, since $\|f_v\| \leq 1$ — is at least this large for *each* v separately, and then union-bounding over all v.)

And now we have

$$\mathsf{OPT} \geq \sum_{uv} A_{uv}\mathbb{E}_u[z_u z_v] = \sum_{uv} A_{uv}\mathbb{E}_g\left[\frac{(g^\intercal f_u)(g^\intercal f_v)}{\max_w (g^\intercal f_w)^2}\right].$$

We'll split this into two cases, depending on whether $\max_w (g^\intercal f_w)^2$ is much larger than what we'd expect it to be or not — we'll think of the 'good case' as the event that $\max_w (g^\intercal f_w)^2 \leq 100\log n$, and the 'bad case' as the event that $\max_w (g^\intercal f_w)^2 > 100\log n$. Let p be the probability of the bad case, so that $p \leq 2n^{-4}$.

In the good case, noting that $\mathbb{E}_g[(g^\intercal f_u)(g^\intercal f_v)] = \langle f_u, f_v\rangle$, we get a total contribution of at least

$$\frac{1-p}{100\log n}\sum_{uv} A_{uv}X_{uv} \geq \frac{1-2n^{-4}}{100\log n}\cdot \mathsf{SDP}.$$

Meanwhile, for the contribution of the bad case (this contribution could potentially be negative, and we're trying to show it can't be too negative), we can upper-bound $|g^\intercal f_u|$ and $|g^\intercal f_v|$ by 1 and lower-bound $\max_w (g^\intercal f_w)^2$ by $100\log n$, so the bad case contributes at least

$$-p\cdot \sum_{uv} |A_{uv}|\cdot \frac{1}{100\log n} \geq -\frac{1}{50n^4\log n}\sum_{uv} |A_{uv}|.$$

And the expression we have here is exactly the one in Lemma 3.3, so plugging in that bound gives that this is at least

$$-\frac{1}{50n^3\log n}\cdot \mathsf{OPT}.$$

Putting these together, we get that

$$\mathsf{OPT} \geq \frac{1-2n^{-4}}{100\log n}\cdot \mathsf{SDP} - \frac{1}{50n^3\log n}\cdot \mathsf{OPT},$$

and moving things around gives $\mathsf{SDP} \leq 120\log n \cdot \mathsf{OPT}$ (which proves the upper bound of Theorem 3.1; the constant 120 is not important).

> **Remark 3.5.** Here, the place we lose the factor of $\log n$ is in taking the maximum of n Gaussians. And this loss is necessary — there are constructions showing that we need the $\log n$.

§3.2 A different perspective

Here's a different perspective on the same rounding scheme which involves a somewhat simpler argument (one that doesn't require something like Lemma 3.3).

Let X be the PSD matrix which is the optimal solution to the SDP, and imagine that we choose a Gaussian vector $h \sim \mathcal{N}(0, X)$ — i.e., instead of choosing a standard Gaussian, we choose one with covariance matrix X — and define $z(h) \in [-1,1]^n$ as the vector

$$z(h) = \frac{1}{\|h\|_\infty} \cdot h.$$

Then its entries $z(h)_u$ form a solution to the $[-1,1]$-optimization problem (for any choice of h).

> **Remark 3.6.** This is actually the same solution to the $[-1,1]$-optimization problem as the one we constructed in Subsection 3.1, just phrased differently — in that proof, the vector $y = (y_u)$ is distributed according to $\mathcal{N}(0, X)$.

Now we have

$$\mathsf{SDP} = \sum_{uv} A_{uv} X_{uv} = \sum_{uv} A_{uv} \mathbb{E}_{h \sim \mathcal{N}(0,X)}[h_u h_v]$$

(since we defined h to have covariance matrix X). And we can rewrite this as

$$\mathsf{SDP} = \sum_{uv} A_{uv} \mathbb{E}_h[z(h)_u z(h)_v \|h\|_\infty^2].$$

And now if we pull out $\|h\|_\infty^2$, what we're left with is the value of the $[-1,1]$-optimization problem at z. The difference with the argument in Subsection 3.1 is that instead of bounding the *expectation* of this quantity by OPT (which is essentially what we did in Subsection 3.1), we can even bound its *maximum* by OPT — so we get that

$$\mathsf{SDP} \leq \sup_h \sum_{uv} A_{uv} z(h)_u z(h)_v \mathbb{E}_{h \sim \mathcal{N}(0,X)} \|h\|_\infty^2 \leq \mathsf{OPT} \cdot \mathbb{E}_{h \sim \mathcal{N}(0,X)} \|h\|_\infty^2.$$

And similarly to Fact 3.4, we can show $\mathbb{E}\|h\|_\infty^2 \leq C \log n$ for some constant C, giving $K(K_n) \leq C \log n$.

> **Remark 3.7.** This bound is true for all graphs without self-loops.

§4 General graphs

Finally, we'll consider the case of general graphs. Things are more technical here, so we'll focus on the main ideas but not go into the nitty-gritty details.

> **Theorem 4.1** (AMMN)
> For any G, we have $K(G) = O(\log \vartheta(\overline{G}))$.

Here ϑ denotes the Lovász theta number, which we will define later (when it comes up).

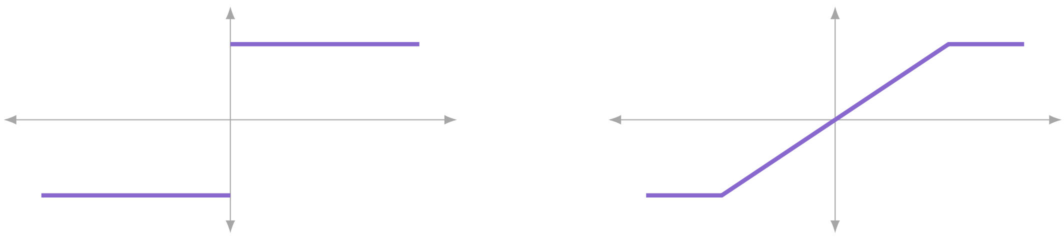
(We work with graphs without self-loops, so we can again consider $K'(G)$ instead.)

In all the proofs we've seen so far, we took our optimal PSD matrix X, a random vector g, and a rounding scheme (so we factored X as $X_{uv} = \langle f_u, f_v \rangle$, defined the real numbers $y_u = g^\intercal f_u$, and used the rounding scheme to turn these numbers into $z_u \in [-1,1]$).

Here, our random vector g is again going to be a standard Gaussian. And our rounding scheme will be a mixture of sgn and the rounding function we used for complete graphs (which was essentially $\max_u |y_u|$). The sgn function looks like -1 until we reach 0 and becomes 1 after. So instead, we're going to take a linear interpolation between -1 and 1 — we'll let m be a parameter we'll decide on later, and we define

$$z_u = \begin{cases} y_u/m & \text{if } |y_u| \leq m \\ \operatorname{sgn}(y_u) & \text{otherwise} \end{cases}$$

(where $y_u = g^\intercal f_u$, as before).



> **Remark 4.2.** We could think of the rounding scheme we used for complete graphs as being of this form with $m = \max |y_u|$ (so that we're always in the first case); but here we're generally going to take m to be smaller than this.

In order to analyze this, we're going to use a slightly similar approach to the first proof in the bipartite setting — where we 'approximate' our rounding scheme by a linear function (which corresponds directly to the value of the SDP) and try to control the error. Here, we'll define $x_u = y_u/m$, so that x_u is often but not always equal to z_u. Then we have

$$\mathsf{OPT} \geq \sum_{uv \in E} A_{uv} \mathbb{E}_g[z_u z_v] = \sum_{uv \in E} A_{uv} \mathbb{E}_g[x_u x_v] - \sum_{uv \in E} A_{uv} \mathbb{E}_g[x_u x_v - z_u z_v] \tag{5}$$

(the first term is the linear term we pull out, and the second is the 'error'). For the first term, we have

$$x_u x_v = \frac{1}{m^2} y_u y_v = \frac{1}{m^2} \langle f_u, f_v \rangle,$$

which means the first term is precisely $\frac{1}{m^2} \cdot \mathsf{SDP}$.

Now we'll try to deal with the error term. In the first proof in the bipartite setting, the way we did so was by saying that the error terms (over all pairs (u, v)) formed a PSD matrix whose diagonal entries were not too large, and so we could scale this matrix to get another feasible solution to the SDP (allowing us to upper-bound the subtraction in terms of SDP).

We want to do something similar here — we want to construct vectors h_u such that $\langle h_u, h_v \rangle = x_u x_v - z_u z_v$ for all $uv \in E$. And our goal is to do so in such a way that $\|h_u\|$ is 'small' for all u — this is because then we can scale them down by $\max_v \|h_v\|$ in order to get a feasible solution to the SDP. In other words, we take the Gram matrix of the h_u's and scale it down so its diagonal entries are at most 1, and this gives a feasible solution to the SDP; and the bound we get for the error term will be the amount we scaled down by times SDP, which will be better if all the original diagonal entries are small (so that we don't have to scale down by too much).

So we're trying to construct vectors h_u such that the diagonal of their Gram matrix is as small as possible, given some constraints on the off-diagonal entries (each corresponding to an edge). Intuitively, this is where the Lovász theta number comes in — if we've got lots of blank space (i.e., not many constraints) or more control on the structure of where these constraints are, then we might expect it to be easier to construct this Gram matrix.

Here's one way to define the Lovász theta number (this is sort of the dual formulation of the usual one, but it's more helpful here).

Definition 4.3. We define $\vartheta(\overline{G})$ as the minimum value of κ such that there is a PSD matrix W with diagonal entries 1 and with $W_{uv} = -1/(\kappa - 1)$ for all $uv \in E$.

Intuitively, it makes sense that this quantity comes into play, because this is very similar to the setup we've got here (where we're trying to minimize the diagonal given constraints on the entries corresponding to edges). But we have to do some hijinks to get things to exactly line up. We first rewrite

$$x_u x_v - z_u z_v = \frac{1}{2}(x_u + z_u)(x_v + z_v) - \frac{1}{2}(x_u - z_u)(x_v + z_v).$$

And $x_u = \frac{1}{m} \cdot g^\intercal f_u$ while z_u is just the version of x_u capped at ± 1, so we can compute

$$x_u - z_u = \mathbf{1}_{|g^\intercal f_u| > m} \cdot g^\intercal f_u \left(\frac{1}{m} - \frac{1}{|g^\intercal f_u|}\right),$$

and similarly

$$x_u + z_u = g^\intercal f_u \cdot \left(\frac{2}{m} + \mathbf{1}_{|g^\intercal f_u| > m}\left(\frac{1}{|g^\intercal f_u|} - \frac{1}{m}\right)\right).$$

Now let W be a solution to the SDP defined by the Lovász θ function (as in Definition 4.3), and factor it as the Gram matrix of vectors w_u. Then using W and this representation of $x_u x_v - z_u z_v$, it's possible to get some vectors h_u with $\langle h_u, h_v \rangle = x_u x_v - z_u z_v$ for all $uv \in E$ and with $\|h_u\| \leq (\frac{1}{2} + \vartheta(\overline{G}))(x_u - z_u)^2$. And we can get bounds on $(z_u - y_u)^2$ using the above expression for it and Gaussian tail bounds (as $g^\intercal f_u$ is a Gaussian).

Then we can rearrange this to a bound on the error term in (5), which ends up giving $\mathsf{OPT} \geq C_m \cdot \mathsf{SDP}$ where C_m is some function depending on m; and optimizing over m gives a bound of $\log \vartheta(\overline{G})$ (where the log comes from Gaussian tail bounds).