LECTURE NOTES ON APPLICATIONS OF GROTHENDIECK'S INEQUALITY

APPROXIMATING THE CUT NORM

JOP BRIËT

Abstract. In this lecture we will discuss Grothendieck's inequality in the context of combinatorial optimization. In particular, we will cover a result of Alon and Naor [AN06] on approximating the cut norm of a matrix in polynomial time.

1. Approximating the cut norm

The *cut norm* of a matrix $A \in \mathbb{R}^{m\times n}$ is defined by

$$\|A\|_{\mathrm{cut}} = \max_{S\subseteq[m],T\subseteq[n]} \Big|\sum_{i\in S}\sum_{j\in T} A_{ij}\Big|.$$

The problem of computing the cut norm of a given matrix is relevant in a variety of problems. Examples include finding regular (or Szemerédi) partitions of graphs [ADL+94] and so-called cut decompositions of matrices [FK99]. Unfortunately, this problem is unlikely to be tractable, even in an approximate sense. Say that an algorithm ALG approximates the cut norm of a matrix A to within a factor $c \in (0, 1]$ if it returns a number $\mathrm{ALG}(A)$ whose value lies between $c\|A\|_{\mathrm{cut}}$ and $\|A\|_{\mathrm{cut}}$.

Proposition 1.1 (Alon–Naor)**.** *If $P \neq NP$, then there is no polynomial-time algorithm that, given a matrix $A \in \mathbb{R}^{m\times n}$, approximates $\|A\|_{\mathrm{cut}}$ to within a factor greater than $16/17 + \varepsilon$ for any fixed $\varepsilon > 0$.*

The proof of this proposition uses a simple reduction from the maxcut problem. Given a graph $G = (V, E)$ and a bi-partition (S, S^c) of the vertex set, define the *cut value* of (S, S^c) to be the number of edges with one endpoint in S and one endpoint in S^c. The maxcut problem asks to compute the maximum cut value among all bi-partitions. A famous result of Håstad [Hås01] asserts that it is NP-hard to approximate

MAXCUT to within a factor $16/17 + \varepsilon$ for any fixed $\varepsilon > 0$. However, things don't get much worse than Proposition 1.1.

Theorem 1.2 (Alon–Naor). *There exists a randomized polynomial-time algorithm that approximates the cut norm to within a factor* 0.56*.*

The key to Theorem 1.2 is a connection between Grothendieck's inequality and semidefinite programming.

2. Semidefinite programming

Recall that a symmetric matrix $A \in \mathbb{R}^{n\times n}$ is *positive semidefinite* if all of its eigenvalues are nonnegative, in which case we write $A \succeq 0$. Another characterization of positive semidefinite matrices is given by the set of Gram matrices. For $d, n \in \mathbb{N}$ and a set of vectors $x_1, \dots, x_n \in \mathbb{R}^d$, define $\mathrm{Gram}(x_1, \dots, x_n)$ to be the $n \times n$ matrix given by $(\langle x_i, x_j\rangle)_{i,j=1}^n$. A matrix is positive semidefinite if and only if it is a Gram matrix. Given a positive semidefinite matrix, a set of Gram vectors can be found in polynomial time (due to the fact that there is a polynomial-time algorithm for the Cholesky decomposition).

A important tool in optimization is a polynomial-time algorithm for maximizing linear functionals over positive semidefinite matrices subject to linear constraints. A simple generic semidefinite program has the following form: Let $A, C_1, \dots, C_k \in \mathbb{R}^{n\times n}$ be symmetric matrices and $b_1, \dots, b_k \in \mathbb{R}$ be real numbers. Denote by $\langle A, X\rangle = \sum_{i,j=1}^n A_{ji}X_{ij}$ the trace inner product.

$$\begin{aligned} \text{maximize} \quad & \langle A, X\rangle \\ \text{subject to} \quad & X \succeq 0 \\ & \langle X, C_i\rangle = b_i \quad \forall i \in \{1, \dots, k\}. \end{aligned}$$

The function $X \mapsto \langle A, X\rangle$ is referred to as the *objective function* and a matrix X is *feasible* if it simultaneously satisfies all the constrains $X \succeq 0$ and $\langle X, C_i\rangle \leq b_i$ for each $i \in \{1, \dots, k\}$. The maximum possible value objective value over the set of feasible solutions is the *optimum.* A feasible solution whose objective value is within an additive error $\varepsilon > 0$ of the optimum can be found in polynomial time (in the size of the input $(A, C_1, \dots, C_k, b_1, \dots, b_k)$ and the logarithm of $1/\varepsilon$.)

3. The Alon–Naor algorithm

The starting point for Theorem 1.2 is the following simple proposition.

Proposition 3.1. *Let m, n be positive integers and let $n' = \max\{m, n\}$. For any matrix $A \in \mathbb{R}^{m\times n}$ there exists a matrix $B \in \mathbb{R}^{n'\times n'}$ such that*

$$\|A\|_{\text{cut}} = \frac{1}{4}\|B\|_{\infty\to 1}.$$

It thus suffices to approximate the $\infty \to 1$ norm of a matrix. This is where the meat is.

Theorem 3.2. *For any $\varepsilon > 0$, there exists a randomized polynomial-time algorithm that, given a matrix $A \in \mathbb{R}^{n\times n}$, returns random vectors $a, b \in \{-1, 1\}^n$ such that*

$$0.56\,\|A\|_{\infty\to 1} - \varepsilon \le \mathbb{E}\Big[\sum_{i,j=1}^{n} A_{ij} a_i b_j\Big] \le \|A\|_{\infty\to 1}. \tag{1}$$

Theorem 3.2 follows from the following link between the Grothendieck norm and semidefinite programming.

Proposition 3.3. *For any fixed $\varepsilon > 0$, there is a polynomial-time algorithm that, given a matrix $A \in \mathbb{R}^{n\times n}$, returns unit vectors $x_1, \ldots, x_n, y_1, \ldots, y_n \in S^{2n-1}$ such that*

$$\Big|\sum_{i,j=1}^{n} A_{ij}\langle x_i, y_j\rangle - \|A\|_G\Big| \le \varepsilon. \tag{2}$$

Proof: Define the $2n \times 2n$ block matrix $B = \left[\begin{smallmatrix} 0 & A \\ 0 & 0 \end{smallmatrix}\right]$. Consider the semidefinite program

$$\begin{aligned} \text{maximize} \quad & \langle B, Z\rangle \\ \text{subject to} \quad & Z \succeq 0 \\ & Z_{kk} = 1 \quad \forall k \in [2n]. \end{aligned}$$

By the remarks above, this program can be solved up to error ε in polynomial time. Let Z be a feasible solution. Then since Z is positive semidefinite, it is a Gram matrix, and so there exist vectors $z_1, \ldots, z_{2n}$ such that $Z = \text{Gram}(z_1, \ldots, z_{2n})$. Moreover, since $Z_{kk} = 1$ for each $k \in [2n]$, it follows that each z_k is a unit vector. Rename these vectors to $x_i = z_i$ for each $i \in [n]$ and $y_j = z_j$ for each $j \in \{n+1, \ldots, 2n\}$. From this, it is easy to see that the optimum equals $\|A\|_G$. □

The main idea behind the algorithm in Theorem 3.2 is to use Krivine's proof of Grothendieck's inequality. Recall that this proof used the following lemma.

Lemma 3.4 (Krivine). *Let $x_1, \dots, x_n, y_1, \dots, y_n \in S^{2n-1}$ be unit vectors. Then, there exist unit vectors $u_1, \dots, u_n, v_1, \dots, v_n \in S^{2n-1}$ such that for all $i, j \in [n]$, we have*

$$\langle u_i, v_j \rangle = \sin(c\langle x_i, y_j \rangle), \tag{3}$$

where $c = \sinh^{-1}(1)$.

In addition, we had Grothendieck's identity.

Lemma 3.5 (Grothendieck's identity). *Let x, y be n-dimensional real unit vectors and let $g = (g_1, \dots, g_n) \sim N(0, I_n)$ be an n-dimensional standard Gaussian vector. Then,*

$$\mathbb{E}\big[\operatorname{sign}(\langle x, g \rangle) \operatorname{sign}(\langle y, g \rangle)\big] = \frac{2}{\pi} \arcsin(\langle x, y \rangle). \tag{4}$$

Proof of Theorem 3.2: We first use Proposition 3.3 to efficiently find vectors $x_1, \dots, x_n$, $y_1, \dots, y_n \in S^{2n-1}$ such that

$$\sum_{i,j=1}^{n} A_{ij} \langle x_i, y_j \rangle \geq \|A\|_G - \varepsilon \geq \|A\|_{\infty \to 1} - \varepsilon.$$

By Lemma 3.4, there exist vectors $u_1, \dots, u_n, v_1, \dots, v_n \in S^{2n-1}$ such that (3) holds for all $i, j \in [n]$. Hence, such vectors can be found in polynomial time using semidefinite programming. Now sample a random vector $g \in \mathbb{R}^{2n}$ from the standard Gaussian distribution $N(0, I_{2n})$ and let $a_i = \operatorname{sign}(\langle u_i, g \rangle)$ and $b_j = \operatorname{sign}(\langle v_j, g \rangle)$. The claim now follows from Grothendieck's identity (Lemma 3.5). □

4. Exercises

Exercise 4.1. In the proof of Theorem 1.2 we claimed that the vectors u_i, v_j from Lemma 3.4 can be found efficiently using semidefinite programming. Give a semidefinite program that produces such vectors.

References

[ADL+94] N. Alon, R. A. Duke, H. Lefmann, V. Rödl, and R. Yuster. The algorithmic aspects of the regularity lemma. *J. Algorithms*, 16(1):80–109, 1994.

[AN06] Noga Alon and Assaf Naor. Approximating the cut-norm via Grothendieck's inequality. *SIAM J. Comput.*, 35(4):787–803 (electronic), 2006. Preliminary version in STOC'04.

[FK99] Alan Frieze and Ravi Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

[Hås01] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.