Notes on "Approximating the Cut-Norm via Grothendieck's Inequality" by Alon and Naor

Ping-Ko Chiu

October 7, 2017

1 Cut Norm

Given a matrix $A = (a_{ij})_{i \in R, j \in S}$, the cut norm is given by $\|A\|_C = \max_{I \subset R, J \subset S} \sum_{i \in I, j \in J} a_{ij}$.

2 Result

Using SDP and rounding techniques, they concluded with a randomized approximation algorithm that solves for I, J such that

$$\left| \sum_{i \in I, j \in J} a_{ij} \right| = \rho \|A\|_C$$

where $\rho = 0.58$

3 Use of $\|A\|_{\infty \to 1}$

$$\|A\|_{\infty \to 1} = \max_{x_i, y_i \in \{-1,1\}} \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_i y_j$$

3.1 Why the $\|A\|_{\infty \to 1}$? and not other norms

Alon and Noar mentioned that it is "convenient" to study this norm. But why?

So I came up with two other optimization problems and evaluated why they are not relevant.

1. $\max_{x_{ij} \in \{-1,1\}} \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_{ij}$. This optimization problem is easy to solve since we just set $x_{ij} = 1$ if $a_{ij} > 0$ and $x_{ij} = -1$ otherwise. Solving this problem solves $\max_{Q \in R \times S} \sum_{(i,j) \in Q} a_{ij}$. In otherwords, this calculates the maximum subset sum of of A, which is trivial and irrelevant to the cutnorm problem.

2. $\max_{x_i \in \{-1,1\}} \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_i$. This is also easy since we just do a row sum and find out if the row is positive/negative. Set $x_i = 1$ if row sum positive, and $x_i = -1$ if row sum negative. Solving this problem solves $\max_{I \in R} \sum_{i \in I, j \in S} a_{ij}$. In other words, this solves the maximum sum of the subset of rows, which is trivial and not relevant to the cutnorm. The column version is equivalent.

We want to find $I \in R, J \in S$ that maximizes $\sum_{i \in I, j \in J} a_{ij}$. A natural optimization problem that seems similar the cutnorm problem would be the $\|A\|_{\infty \to 1}$. In plain words, this problem is seeking for an assignment of a multiplication of $\{-1, 1\}$ to the rows and the columns that maximizes the sum.

3.2 Bounds of $\|A\|_{\infty \to 1}$

The $\|A\|_{\infty \to 1}$ may look like the $\|A\|_C$ but it may not be the case that the assignment of x_i, y_j that gives the maximum of $\|A\|_{\infty \to 1}$ maps directly to the choice of $I \in R, J \in S$ that gives the maximum of $\|A\|_C$. Thus we prove the bounds of $\|A\|_{\infty \to 1}$ with respect to $\|A\|_C$.

Taken from proof in section 3 of Alon Noar paper with my added interpretation.

Bound 1: Without assumption of direct mapping from x_i, y_i to I, J

For any $x_i, y_i \in \{-1, 1\}$,

$$\sum_{i,j} a_{ij} x_i y_j = \sum_{\substack{i: x_i = 1 \\ j: y_j = 1}} a_{ij} - \sum_{\substack{i: x_i = 1 \\ j: y_j = -1}} a_{ij} - \sum_{\substack{i: x_i = -1 \\ j: y_j = 1}} a_{ij} + \sum_{\substack{i: x_i = -1 \\ j: y_j = -1}} a_{ij}$$

Since the absolute value of each term is $\|A\|_C$. By the triangle inequality

$$\|A\|_{\infty \to 1} \leq 4 \|A\|_C$$

Bound 2: Assume mapping $x_i = 1$ if $i \in I$ and $x_i = -1$ otherwise, and similarly, $y_j = 1$ if $j \in J$ and $y_j = -1$ otherwise.

$$\|A\|_C = \sum_{i,j} a_{ij} \frac{1 + x_i}{2} \frac{1 + y_j}{2} \qquad = \frac{1}{4} \sum_{i,j} a_{ij} + \frac{1}{4} \sum_{i,j} a_{ij} x_i + \frac{1}{4} \sum_{i,j} a_{ij} y_j + \frac{1}{4} \sum_{i,j} a_{ij} x_i y_j$$

(Sanity check: Only if $x_i = y_i = 1$, which corresponds to $i \in I, j \in J$ will it be included in the sum)

Since each of the terms in the sum is at most $\|A\|_{\infty \to 1} / 4$

$$\|A\|_C \leq \|A\|_{\infty \to 1}$$

General case: Combining bound 1 and bound 2, we have

$$\|A\|_C \leq \|A\|_{\infty \to 1} \leq 4 \|A\|_C$$

Special case of zero row sum and zero column sum matrix:

If row and column sum to zero, $\sum_{i,j} a_{ij} = \sum_{i,j} a_{ij} x_i = \sum_{i,j} a_{ij} y_j = 0$.

Then, $\|A\|_C = \frac{1}{4}\sum_{i,j} a_{ij}x_iy_j \leq \frac{1}{4}\|A\|_{\infty\to 1}$, this implies

$$\|A\|_{\infty\to 1} \geq 4\|A\|_C$$

Given Bound 1 and this special case, we have

$$\|A\|_{\infty\to 1} = 4\|A\|_C$$

■

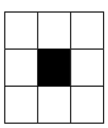
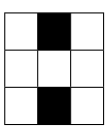
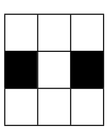
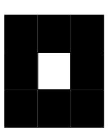
4 Cutnorm invariant transformation

Proposition 1: Given a real value matrix $A = (a_{i,j})_{i\in R, j\in S}$, let A^* be an $(n+1)\times(m+1)$ matrix where for all $1 \leq i \leq n, 1 \leq j \leq m, a^*_{i,j} = a_{i,j}$, for all $1 \leq j \leq m, a^*_{n+1,j} = \sum_i -a_{i,j}$, for all $1 \leq i \leq n, a^*_{i,m+1} = \sum_j -a_{i,j}$, and $a^*_{n+1,m+1} = \sum_{i,j} a_{ij}$. $\|A\|_C = \|A^*\|_C$

Proof: Since A is a submatrix of A^*, $\|A\|_C \leq \|A^*\|_C$.

Suppose, $\|A^*\|_C = \left|\sum_{i\in I, j\in J} a_{ij}\right|$, where $I \subset [n+1], J \subset [m+1]$. Let $P = \begin{cases} [n+1]\setminus I, & \text{if } n+1 \in I \\ I, & \text{otherwise} \end{cases}$, $Q = \begin{cases} [m+1]\setminus J, & \text{if } m+1 \in J \\ J, & \text{otherwise} \end{cases}$.

Let us examine the possibilities of P, Q.

1.  $P = I, Q = J$
2.  $P = [n+1]\setminus I, Q = J$
3.  $P = I, Q = [m+1]\setminus J$
4.  $P = [n+1]\setminus I, Q = [m+1]\setminus J$

Since the sum of each row and column is zero, $\|A^*\|_C = \left|\sum_{i\in I, j\in J} a_{ij}\right| = \left|\sum_{p\in P, q\in Q} a_{pq}\right|$. Furthermore, since $P \in R, Q \in S$, $\left|\sum_{p\in P, q\in Q} a_{pq}\right| \leq \|A\|_C$. Therefore $\|A^*\|_C \leq \|A\|_C$, and we have $\|A^*\|_C = \|A\|_C$. ■

4.1 Thought regarding $a^*_{n+1,m+1}$

If $a^*_{n+1,m+1} = 0$, we cannot ensure that $\|A\|_{\infty\to 1} = 4\|A\|_C$ since the row and columns dont sum to zero. It must be the case that $a^*_{n+1,m+1} = \sum_{i,j} a_{ij}$ in order to apply the assumption to use SDP to solve this problem.

5 Different rounding methods to approximate the cut norm

For any positive δ, and $u_i, v_j \in R^p$, where $p = n + m$, we can compute $\sum_{i,j} a_{ij} u_i$ to be at least the max value of the program minus δ.

Thus the important work is on the rounding which is finding $x_i, y_j \in \{-1, 1\}$ given u_i, v_j

$$\sum_{i,j} a_{ij} x_i y_j \geq \rho \sum_{i,j} a_{i,j} u_i \cdot v_j$$

Different rounding will achieve different $\rho -$ approximation.

Here are the key takeways without going into details of the proofs.

5.1 Method 1 - Four-wise indepndent $\{-1, 1\}^p$ vectors

Let V be a set of vectors $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_p) \in \{-1, 1\}^p$.

Let $H(q)$ denote the random variable by $[H(q)](\epsilon) = \sum_{j=1}^{p} \epsilon_j q_j$.

By markov's Inequality, for every positive real m, $Prob[|H(q)| \geq m] \leq \frac{3}{m^4}$.

Let $H^M(q)$ denote the M truncation of $H(q)$ where

$$[H^M(q)](\epsilon) = \begin{cases} [H(q)](\epsilon) & \text{if } |[H(q)](\epsilon)| \leq M \\ M & \text{if } [H(q)](\epsilon) > M \\ -M & \text{if } [H(q)](\epsilon) < -M \end{cases}$$

After some math, it is shown that

$$\sum_{i,j} a_{ij} H^M(u_i)(\epsilon) \cdot H^M(v_j)(\epsilon) \geq B(1 - \frac{2}{M}) - \delta$$

Choosing $M = 3$ and let $x_i = \frac{H^M(u_i)(\epsilon)}{M}, y_i = \frac{H^M(v_i)(\epsilon)}{M}$, for some ϵ, $|x_i|, |y_i| \leq 1$.

$$\sum_{i,j} a_{ij} x_i y_j \geq \frac{B}{27} - \frac{\delta}{9}$$

Paper claims that we can shift without decreasing the value of the sum?

Given this, they concluded that we can obtain $x_i, y_j \in \{-1, 1\}$ such that $\sum_{ij} a_{ij} x_i y_j \geq \frac{B}{27} - \frac{\delta}{9}$. Considering small δ, theorem 4.3 states:

"There is a deterministic polynomial time algorithm that finds, for a given real matrix $A = (a_ij)$, integers $x_i, y_j \in \{-1, 1\}$ such that the value of the sum $\sum_{ij} a_{ij}x_iy_j$ is at least $0.03B$ where B is the value of the semidefinite program."

5.2 Rounding with Gaussian Measure: Rietz' method

Let $u_i, v_j \in R^p$.

Let $g_1, g_2, \dots, g_p$ be standard indepndent Gaussian random variables and $G = (g_1, g_2, \dots, g_p)$.

By defining

$$x_i = sign(u_j \cdot G), y_j = sign(v_j \cdot G)$$

we have a solution that provides $\rho = \frac{4}{\pi} - 1 \approx 0.27$ approximation.

5.2.1 For positive definite matrices

For positive definite matrices, the approximation ratio is $\frac{2}{\pi} \approx 0.64$ from the work of Nesterov.

5.3 Rounding with random uniform vector: Krivine's argument

By Taylor's expansion of the Hilbert space H defined by $u, v \in H$, we expand the expression into a form involving the tensor power. By applying Grothendieck's identity to the result of the taylor expansion, we have $\frac{\pi}{2}E([sign(T(u_i) \cdot z] \cdot [sign(S(v_j) \cdot z]) = cu_i \cdot v_j$.

The k'th coordinates of $T(u), S(v)$ are given by:

$$T(u)_k = (-1)^k \sqrt{\frac{c^{2k+1}}{(2k+1)!}} \cdot u^{\otimes(2k+1)}$$

$$S(v)_k = \sqrt{\frac{c^{2k+1}}{(2k+1)!}} \cdot v^{\otimes(2k+1)}$$

where $c = sinh^{-1}(1) = ln(1+\sqrt{2})$ and $w^{\otimes j}$ denotes the j'th tensor power: $w \otimes w \otimes \ldots \otimes w$($j$ terms).

For given unit vectors u_i, v_j, let $u'_i = T(u_i)$ and $v'_i = S(v_i)$. Let H' be the span of u'_i, v'_j. Let z be a vector chosen randomly and uniformly in the unit sphere of H'.

By defining

$$x_i = sign(u'_i \cdot z), y_i = sign(v'_j \cdot z)$$

we have a solution that provides $\rho = \frac{2ln(1+\sqrt{2})}{\pi} > 0.56$ approximation.

Stated by Theorem 5.3: "There is a randomized polynomial time algoirthm that given an input n by m matrix $A = (a_{ij})$ and unit vectors $u_i, v_j \in R^p$ finds $x_i, y_j \in \{-1, 1\}$ such that the expected value of the sum $\sum_{ij} a_{ij}x_iy_j$ is

$$\frac{2ln(1+\sqrt{2})}{\pi} \sum_{ij} u_i \cdot v_j$$

Therefore there is a polynomial randomized $\rho-$ approximation algorithm for computing $\|A\|_{\infty\to 1}$, where $\rho = \frac{2ln(1+\sqrt{2})}{\pi} > 0.56$."

6 Connection to Wen & Yin's SDP Solver

The $\|A\|_{\infty\to 1}$ SDP relaxation

$$\begin{aligned} &\max_{u_i, v_i} \sum_{ij} a_{ij} u_i \cdot v_j \\ &\text{subject to } \|u_i\| = \|v_j\| = 1 \end{aligned}$$

We want to reduce the $\|A\|_{\infty\to 1}$ SDP problem to the SDP problem proposed in Wen & Yin's Paper

$$\begin{aligned} &\max_{V=[V_1,\ldots,V_n]} tr(CV^\intercal V) \\ &\text{subject to } \|V_i\| = 1, i = 1, \ldots, n \end{aligned}$$

Using the trace identity $\sum_{ij} w_{ij} x_{ij} = tr(WX)$, this is equivalent to

$$\begin{aligned} &\max_{V=[V_1,\ldots,V_n]} \sum_{ij} c_{ij} (V^\intercal V)_{ij} \\ &\text{subject to } \|V_i\| = 1, i = 1, \ldots, n \end{aligned}$$

Peter's Method: By creating $V = [u_1, \ldots, u_n, v_1, \ldots, v_n]$ and $C = [0A; A0]$, for the case where $n = m$, we have the optimization problem

$$\begin{aligned} &\max_{V=[u_1,\ldots,u_n,v_1,\ldots,v_n]} \sum_{pq} a_{pq} u_p \cdot v_q + \sum_{pq} a_{pq} v_p \cdot u_q \\ &\text{subject to } \|u_i\| = \|v_i\| = 1, i = 1, \ldots, n \end{aligned}$$

And for the case where A is symmetric and square, we have

$$\begin{aligned} &\max_{V=[u_1,\ldots,u_n,v_1,\ldots,v_n]} 2 \sum_{pq} a_{pq} u_p \cdot v_q \\ &\text{subject to } \|u_i\| = \|v_i\| = 1, i = 1, \ldots, n \end{aligned}$$

Alternative?: Peter uses this construction of C and V for solving the cutnorm. Perhaps another viable augmentation is $C = [0A; 00]$. In this case we actually have the original $\|A\|_{\infty\to 1}$ SDP problem and we might save a constant factor computational time. I need to verify this though.