The Cut-Norm and Combinatorial Optimization

Alan Frieze and Ravi Kannan

Outline of the Talk

- The Cut-Norm and a Matrix Decomposition.
- Max-Cut given the Matrix Decomposition.
- Quadratic Assignment given the Matrix Decomposition.
- Constructing Matrix Decomposition via the Grothendieck Identity – Alon and Naor
- Multi-Dimensional Matrices

The *cut-norm* of the $R \times C$ matrix $\mathbf{A}$ is defined to be

$$||\mathbf{A}||_{\square} = \max_{\substack{S \subseteq R \\ T \subseteq C}} |\mathbf{A}(S, T)|.$$

The *cut-norm* of the $R \times C$ matrix $\mathbf{A}$ is defined to be

$$||\mathbf{A}||_\Box = \max_{\substack{S \subseteq R \\ T \subseteq C}} |\mathbf{A}(S,T)|.$$

Relation to regularity:
$\mathbf{D}$ is an $R \times C$ matrix with $\mathbf{D}(i,j) = d$.
W is an $R \times C$ matrix with $||W||_\Box \leq \epsilon |R|\,|C|$.
$\mathbf{A} = \mathbf{D} + \mathbf{W}$.

$$|\mathbf{A}(S,T) - d|S|\,|T|\,| \leq \epsilon |R|\,|C|.$$

Cut Matrices

Given $S \subseteq R$, $T \subseteq C$ and real value d:

$R \times C$ *Cut Matrix* $\mathbf{C} = CUT(S, T, d)$:

$$\mathbf{C}(i,j) = \begin{cases} d & \text{if } (i,j) \in S \times T, \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{bmatrix} d & d & d & d & 0 & 0 \\ d & d & d & d & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Matrix Decomposition

$$\mathbf{A} = \mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(s)} + \mathbf{W}.$$

$\mathbf{D}^{(t)} = CUT(R_t, C_t, d_t)$.

Matrix Decomposition

$$\mathbf{A} = \mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(s)} + \mathbf{W}.$$

$\mathbf{D}^{(t)} = CUT(R_t, C_t, d_t)$.

We want $s, \max_t\{|d_t|\}$ and $||\mathbf{W}||_\square$ to be small.

Matrix Decomposition

$$\mathbf{A} = \mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(s)} + \mathbf{W}.$$

$\mathbf{D}^{(t)} = CUT(R_t, C_t, d_t)$.

We want $s, \max_t\{|d_t|\}$ and $||\mathbf{W}||_\square$ to be small.

$$\begin{aligned} ||\mathbf{W}||_\square &\leq \epsilon mn \qquad\qquad m = |R|, n = |C| \\ s &= O(1/\epsilon^2) \\ \max_t\{|d_t|\} &= O(1) \end{aligned}$$

is achievable.

Let

$$||\mathbf{A}||_F = \left(\sum_{i,j} A(i,j)^2 \right)^{1/2}$$

be the Frobenius Norm of **A**.

Assume inductively that we have found cut matrices

$$\mathbf{D}^{(j)} = CUT(R_j, C_j, d_j),$$

such that $\mathbf{W}^{(t)} = \mathbf{A} - (\mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(t)})$ satisfies

$$||\mathbf{W}^{(t)}||_F^2 \leq (1 - \epsilon^2 t)||A||_F^2.$$

Assume inductively that we have found cut matrices

$$\mathbf{D}^{(j)} = CUT(R_j, C_j, d_j),$$

such that $\mathbf{W}^{(t)} = \mathbf{A} - (\mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(t)})$ satisfies

$$||\mathbf{W}^{(t)}||_F^2 \leq (1 - \epsilon^2 t)||A||_F^2.$$

Suppose there exist $S \subseteq R, T \subseteq C$ such that

$$|\mathbf{W}^{(t)}(S, T)| \geq \epsilon\sqrt{mn}||A||_F.$$

Let

$$R_{t+1} = S,\ C_{t+1} = T,\ d_{t+1} = \frac{\mathbf{W}^{(t)}(S, T)}{|S||T|}.$$

$$||\mathbf{W}^{(t+1)}||_F^2 - ||\mathbf{W}^{(t)}||_F^2 = ||\mathbf{W}^{(t)} - \mathbf{D}^{(t+1)}||_F^2 - ||\mathbf{W}^{(t)}||_F^2 =$$

$$\sum_{\substack{i \in R_{t+1} \\ j \in C_{t+1}}} ((\mathbf{W}^{(t)}(i,j) - d_{t+1})^2 - \mathbf{W}^{(t)}(i,j)^2) =$$

$$- |R_{t+1}||C_{t+1}|d_{t+1}^2 =$$

$$- \frac{\mathbf{W}^{(t)}(R_{t+1}, C_{t+1})^2}{|R_{t+1}|\,|C_{t+1}|} \leq -\epsilon^2 ||\mathbf{A}||_F^2.$$

$$||\mathbf{W}^{(t+1)}||_F^2 - ||\mathbf{W}^{(t)}||_F^2 = ||\mathbf{W}^{(t)} - \mathbf{D}^{(t+1)}||_F^2 - ||\mathbf{W}^{(t)}||_F^2 =$$

$$\sum_{\substack{i \in R_{t+1} \\ j \in C_{t+1}}} ((\mathbf{W}^{(t)}(i,j) - d_{t+1})^2 - \mathbf{W}^{(t)}(i,j)^2) =$$

$$-|R_{t+1}||C_{t+1}|d_{t+1}^2 =$$

$$-\frac{\mathbf{W}^{(t)}(R_{t+1}, C_{t+1})^2}{|R_{t+1}|\,|C_{t+1}|} \leq -\epsilon^2||\mathbf{A}||_F^2.$$

Conclusion: $\exists \mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(s)}, s \leq \epsilon^{-2}$ such that

$$||\mathbf{W}^{(s)}||_\square \leq \epsilon\sqrt{mn}||\mathbf{A}||_F$$

.

Refinements

Suppose that we can only compute R_{t+1}, C_{t+1} such that $|\mathbf{W}^{(t)}(R_{t+1}, C_{t+1})| \geq \rho ||\mathbf{W}^{(t)}||_{\square}$ where $\rho \leq 1$.

Refinements

Suppose that we can only compute R_{t+1}, C_{t+1} such that $|\mathbf{W}^{(t)}(R_{t+1}, C_{t+1})| \geq \rho||\mathbf{W}^{(t)}||_\square$ where $\rho \leq 1$.

Conclusion: We can compute $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(s)}, s \leq \rho^{-2}\epsilon^{-2}$ such that

$$||\mathbf{W}^{(s)}||_\square \leq \epsilon\sqrt{mn}||\mathbf{A}||_F$$

.

Suppose that $||W^{(t)}||_\square \geq \epsilon\sqrt{mn}||\mathbf{A}||_F$ and we have computed R_{t+1}, C_{t+1} such that $|\mathbf{W}^{(t)}(R, C_{t+1})| \geq \rho\epsilon\sqrt{mn}||A||_F$.

Suppose that $||W^{(t)}||_{\square} \geq \epsilon\sqrt{mn}||\mathbf{A}||_F$ and we have computed R_{t+1}, C_{t+1} such that $|\mathbf{W}^{(t)}(R, C_{t+1})| \geq \rho\epsilon\sqrt{mn}||A||_F$.

If $|R_{t+1}| < m/2$ then either (i) $|\mathbf{W}^{(t)}(R, C_{t+1})| \geq \frac{1}{2}\rho\epsilon\sqrt{mn}||A||_F$ or (ii) $|\mathbf{W}^{(t)}(R \setminus R_{t+1}, C_{t+1})| \geq \frac{1}{2}\rho\epsilon\sqrt{mn}||A||_F$

Suppose that $||W^{(t)}||_\square \geq \epsilon\sqrt{mn}||\mathbf{A}||_F$ and we have computed R_{t+1}, C_{t+1} such that $|\mathbf{W}^{(t)}(R, C_{t+1})| \geq \rho\epsilon\sqrt{mn}||A||_F$.

If $|R_{t+1}| < m/2$ then either (i) $|\mathbf{W}^{(t)}(R, C_{t+1})| \geq \frac{1}{2}\rho\epsilon\sqrt{mn}||A||_F$ or (ii) $|\mathbf{W}^{(t)}(R \setminus R_{t+1}, C_{t+1})| \geq \frac{1}{2}\rho\epsilon\sqrt{mn}||A||_F$

Conclusion: We can compute $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(s)}, s \leq 4\rho^{-2}\epsilon^{-2}$ such that

$$||\mathbf{W}^{(s)}||_\square \leq \epsilon\sqrt{mn}||\mathbf{A}||_F$$

and such that $|R_i| \geq m/2$ and $|C_i| \geq n/2$.

Suppose that $||W^{(t)}||_\square \geq \epsilon\sqrt{mn}||\mathbf{A}||_F$ and we have computed R_{t+1}, C_{t+1} such that $|\mathbf{W}^{(t)}(R, C_{t+1})| \geq \rho\epsilon\sqrt{mn}||A||_F$.

If $|R_{t+1}| < m/2$ then either (i) $|\mathbf{W}^{(t)}(R, C_{t+1})| \geq \frac{1}{2}\rho\epsilon\sqrt{mn}||A||_F$ or (ii) $|\mathbf{W}^{(t)}(R \setminus R_{t+1}, C_{t+1})| \geq \frac{1}{2}\rho\epsilon\sqrt{mn}||A||_F$

Conclusion: We can compute $\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(s)}, s \leq 4\rho^{-2}\epsilon^{-2}$ such that

$$||\mathbf{W}^{(s)}||_\square \leq \epsilon\sqrt{mn}||\mathbf{A}||_F$$

and such that $|R_i| \geq m/2$ and $|C_i| \geq n/2$.

Then

$$\sum_{t=1}^{s} |R_t|\,|C_t|d_t^2 \leq ||\mathbf{A}||_F^2 \Longrightarrow \sum_{t=1}^{s} d_t^2 \leq 4||\mathbf{A}||_\infty.$$

MAX-CUT

$G = (V, E)$ is a graph with $n \times n$ adjacency $\mathbf{A}$ and

$$\mathbf{A} = \mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(s)} + \mathbf{W}.$$

$$\mathbf{D}^{(t)} = CUT(R_t, C_t, d_t),\ |d_t| \leq 2,\ s = O(1/\epsilon^2) \textit{ and } ||\mathbf{W}||_\square \leq \epsilon n^2$$

$G = (V, E)$ is a graph with $n \times n$ adjacency $\mathbf{A}$ and

$$\mathbf{A} = \mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(s)} + \mathbf{W}.$$

$$\mathbf{D}^{(t)} = CUT(R_t, C_t, d_t),\ |d_t| \leq 2,\ s = O(1/\epsilon^2) \textit{ and } ||\mathbf{W}||_\Box \leq \epsilon n^2$$

If $(S, \bar{S})$ is a cut in G then the weight of this cut satisfies

$$|\mathbf{A}(S, \bar{S}) - (\mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(s)})(S, \bar{S})| \leq \epsilon n^2.$$

$G = (V, E)$ is a graph with $n \times n$ adjacency $\mathbf{A}$ and

$$\mathbf{A} = \mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(s)} + \mathbf{W}.$$

$$\mathbf{D}^{(t)} = CUT(R_t, C_t, d_t),\ |d_t| \leq 2,\ s = O(1/\epsilon^2) \textit{ and } ||\mathbf{W}||_\Box \leq \epsilon n^2$$

If $(S, \bar{S})$ is a cut in G then the weight of this cut satisfies

$$|\mathbf{A}(S, \bar{S}) - (\mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(s)})(S, \bar{S})| \leq \epsilon n^2.$$

$$\sum_{t=1}^{s} \mathbf{D}^{(t)}(S, \bar{S}) = \sum_{t=1}^{s} d_t f_t g_t$$

where

$$f_t = |S \cap R_t| \text{ and } g_t = |\bar{S} \cap C_t|$$

So we look for S to (approximately) minimize $\sum_{t=1}^{s} d_t f_t g_t$.

So we look for S to (approximately) minimize $\sum_{t=1}^{s} d_t f_t g_t$.

Let $\nu = \epsilon n$ and

$$\bar{f}_t = \left\lfloor \frac{f_t}{\nu} \right\rfloor \nu, \qquad \bar{g}_t = \left\lfloor \frac{g_t}{\nu} \right\rfloor \nu$$

Then

$$\sum_{t=1}^{s} |f_t g_t d_t - \bar{f}_t \bar{g}_t d_t| \leq 6\nu n s \leq 6\epsilon n^2.$$

So we look for S to (approximately) minimize $\sum_{t=1}^{s} d_t f_t g_t$.

Let $\nu = \epsilon n$ and

$$\bar{f}_t = \left\lfloor \frac{f_t}{\nu} \right\rfloor \nu, \qquad \bar{g}_t = \left\lfloor \frac{g_t}{\nu} \right\rfloor \nu$$

Then

$$\sum_{t=1}^{s} |f_t g_t d_t - \bar{f}_t \bar{g}_t d_t| \leq 6\nu n s \leq 6\epsilon n^2.$$

So we can look for S to (approximately) minimize $\sum_{t=1}^{s} d_t \bar{f}_t \bar{g}_t$.

There are $\leq (2/\epsilon)^{2s}$ choices for the sequence $(\bar{f}_t, \bar{g}_t)$ and so we enumerate all possiblities and see if there is a cut with (approximately) these parameters.

There are $\leq (2/\epsilon)^{2s}$ choices for the sequence $(\bar{f}_t, \bar{g}_t)$ and so we enumerate all possiblities and see if there is a cut with (approximately) these parameters.

$\mathcal{P} = V_1, V_2, \ldots, V_k$ is the coarsest partition of V (with at most 2^{2s} parts in it) such that each R_t, C_t is the union of sets in $\mathcal{P}$.

There are $\leq (2/\epsilon)^{2s}$ choices for the sequence $(\bar{f}_t, \bar{g}_t)$ and so we enumerate all possiblities and see if there is a cut with (approximately) these parameters.

$\mathcal{P} = V_1, V_2, \ldots, V_k$ is the coarsest partition of V (with at most 2^{2s} parts in it) such that each R_t, C_t is the union of sets in $\mathcal{P}$.

We check $(\bar{f}_t, \bar{g}_t)$ by solving the LP relaxation of the integer program

$$\begin{array}{lllll} 0 & \leq & x_P & \leq & |P| & \forall\, P \in \mathcal{P} \\ \bar{f}_t & \leq & \sum_{P \subseteq R_t} x_P & < & \bar{f}_t + \nu & 1 \leq t \leq s \\ \bar{g}_t & \leq & \sum_{P \subseteq C_t} (|P| - x_P) & \leq & \bar{g}_t + \nu & \end{array}$$

and doing some adjusting. ($x_P = |S \cap P|$).

There are $\leq (2/\epsilon)^{2s}$ choices for the sequence $(\bar{f}_t, \bar{g}_t)$ and so we enumerate all possiblities and see if there is a cut with (approximately) these parameters.

$\mathcal{P} = V_1, V_2, \ldots, V_k$ is the coarsest partition of V (with at most 2^{2s} parts in it) such that each R_t, C_t is the union of sets in $\mathcal{P}$.

The partition $\mathcal{P}$ has the property that for disjoint $S, T \subseteq V$ we have

$$\left| e(S,T) - \sum_{i \in [k]} \sum_{j \in [k]} d_{i,j} |S_i||T_j| \right| \leq 2||\mathbf{W}||_\square \leq 2\epsilon n^2$$

where $d_{i,j} = e(V_i, V_j)/(|V_i|\,|V_j|)$.

There are $\leq (2/\epsilon)^{2s}$ choices for the sequence $(\bar{f}_t, \bar{g}_t)$ and so we enumerate all possiblities and see if there is a cut with (approximately) these parameters.

$\mathcal{P} = V_1, V_2, \ldots, V_k$ is the coarsest partition of V (with at most 2^{2s} parts in it) such that each R_t, C_t is the union of sets in $\mathcal{P}$.

The partition $\mathcal{P}$ has the property that for disjoint $S, T \subseteq V$ we have

$$\left| e(S,T) - \sum_{i \in [k]} \sum_{j \in [k]} d_{i,j} |S_i||T_j| \right| \leq 2||\mathbf{W}||_\square \leq 2\epsilon n^2$$

where $d_{i,j} = e(V_i, V_j)/(|V_i|\,|V_j|)$.

We could replace $\mathcal{P}$ with an ordinary regular partition. The constants as a function of ϵ get worse.

Quadratic Assignment

$$\text{Minimise} \sum_{i,j,p,q} A_{i,j,p,q} z_{i,p} z_{j,q}$$

$$\text{subject to} \sum_k z_{i,k} = \sum_k z_{k,j} = 1$$

$$z_{i,j} = 0, 1. \forall i, j.$$

A set of n items V have to be assigned to a set of n locations X, one per location. $z_{i,p} = 1$: Place item i in position $p = \pi(i)$

$\mathbf{T}(i, i') \leq 1$ is the amount of *traffic* between item i and i'.
$\mathbf{D}(x, x')$ is the *distance* between location x and x'.

If item i is assigned to location $\pi(i)$ for $i \in [n]$ the total cost $c(\pi)$ is defined by

$$c(\pi) = \sum_{i=1}^{n} \sum_{i'=1}^{n} \mathbf{T}(i, i')\mathbf{D}(\pi(i), \pi(i')).$$

The problem is to minimise $c(\pi)$ over all bijections $\pi : V \to X$.

Metric QAP.

Metric space X with metric $\mathbf{D}$.

1. diam(X)=1 i.e. $\max_{x,y} \mathbf{D}(x,y) = 1$.
2. For all $\epsilon > 0$ there exists a partition $X = X_1 \cup X_2 \cup \cdots \cup X_\ell$, $\ell = \ell(\epsilon)$, such that diam(X_j) $\leq \epsilon$.
   We call this an ϵ – *refinement* of X.

   So there is a $\ell \times \ell$ matrix $\hat{\mathbf{D}}$ such that if $x \in X_j$ and $x' \in X_{j'}$ then
   $$|\mathbf{D}(x,x') - \hat{\mathbf{D}}(j,j')| \leq 2\epsilon.$$
   This partition must be computable in time polynomial in n and $1/\epsilon$.

We call this the *metric* QAP.

We decompose

$$\mathbf{T} = \mathbf{T}_1 + \mathbf{T}_2 + \cdots + \mathbf{T}_s + \mathbf{W}$$

where $||\mathbf{W}||_\square \leq \epsilon n^2$.
For bijection $\pi : V \to X$ we have

$$c(\pi) = \sum_{k=1}^{s} \sum_{i,j=1}^{n} \mathbf{T}_k(i,j)\mathbf{D}(\pi(i),\pi(j)) + \Delta_1$$

We compute an $O(\epsilon^{-3})$-refinement of X and let $S_i^{(\pi)} = \pi^{-1}(X_i)$.

$$\sum_{k=1}^{s} \sum_{i,j=1}^{n} \mathbf{T}_k(i,j)\mathbf{D}(\pi(i),\pi(j)) =$$

$$\sum_{k=1}^{s} \sum_{i,j=1}^{\ell} d_k |R_k \cap S_i^{(\pi)}|\,|C_k \cap S_j^{(\pi)}|\hat{\mathbf{D}}(i,j) + \Delta_2.$$

Computing a decomposition.

Computing a decomposition.

Reducible to finding an approximation to the cut-norm.

Computing a decomposition.

Reducible to finding an approximation to the cut-norm.

Our paper gave algorithms with a “small” additive error.

Computing a decomposition.

Reducible to finding an approximation to the cut-norm.

Our paper gave algorithms with a “small” additive error.

Alon and Naor gave an approximation algorithm with multiplicative error!

Let

$$||\mathbf{A}||_{\infty\to 1} = \max_{x_i, y_j \in \{\pm 1\}} \sum_{i,j} \mathbf{A}(i,j) x_i y_j .$$

Let

$$||\mathbf{A}||_{\infty\to 1} = \max_{x_i, y_j \in \{\pm 1\}} \sum_{i,j} \mathbf{A}(i,j) x_i y_j .$$

$$\sum_{i,j} \mathbf{A}(i,j) x_i y_j = \sum_{\substack{x_i=1 \\ y_j=1}} \mathbf{A}_{i,j} - \sum_{\substack{x_i=1 \\ y_j=-1}} \mathbf{A}_{i,j} - \sum_{\substack{x_i=-1 \\ y_j=1}} + \sum_{\substack{x_i=-1 \\ y_j=-1}} \mathbf{A}_{i,j} .$$

So

$$||\mathbf{A}||_{\infty\to 1} \leq 4||\mathbf{A}||_{\square}.$$

Let

$$||\mathbf{A}||_{\infty\to 1} = \max_{x_i, y_j \in \{\pm 1\}} \sum_{i,j} \mathbf{A}(i,j) x_i y_j .$$

$$\sum_{i,j} \mathbf{A}(i,j) x_i y_j = \sum_{\substack{x_i=1 \\ y_j=1}} \mathbf{A}_{i,j} - \sum_{\substack{x_i=1 \\ y_j=-1}} \mathbf{A}_{i,j} - \sum_{\substack{x_i=-1 \\ y_j=1}} + \sum_{\substack{x_i=-1 \\ y_j=-1}} \mathbf{A}_{i,j} .$$

So

$$||\mathbf{A}||_{\infty\to 1} \le 4||\mathbf{A}||_{\square}.$$

A similar argument gives

$$||\mathbf{A}||_{\square} \le ||\mathbf{A}||_{\infty\to 1}.$$

Grothendieck's Identity

u, v are unit vectors in a Hilbert space H. z is chosen uniformly from $B = \{x : \ ||x|| = 1\}$.

$$\frac{\pi}{2}\mathbf{E}[sign(u \cdot z)\, sign(v \cdot z)] = \arcsin(u.v).$$

Let (u_i^*, v_j^*) define

$$L_{\mathbf{A}} = \max_{u_i, v_j} \sum_{i,j} \mathbf{A}(i,j) u_i \cdot v_j \qquad (\geq ||\mathbf{A}||_{\infty \to 1})$$

where (u_i, v_j) lie in R^{m+n} and $||u_i|| = ||v_j|| = 1$.

Let (u_i^*, v_j^*) define

$$L_{\mathbf{A}} = \max_{u_i, v_j} \sum_{i,j} \mathbf{A}(i,j) u_i \cdot v_j \qquad (\geq ||\mathbf{A}||_{\infty \to 1})$$

where (u_i, v_j) lie in R^{m+n} and $||u_i|| = ||v_j|| = 1$.

(u_i^*, v_j^*) are computable via Semi-Definite Programming.

Let $c = \sinh^{-1}(1) = \ln(1+\sqrt{2})$.

$$\begin{aligned}
\sin(cu_i^* \cdot v_j^*) &= \sum_{k=0}^{\infty} (-1)^k \frac{c^{2k+1}}{(2k+1)!} (u_i^* \cdot v_j^*)^{2k+1} \\
&= \sum_{k=0}^{\infty} (-1)^k \frac{c^{2k+1}}{(2k+1)!} (u_i^*)^{\otimes(2k+1)} \cdot (v_j^*)^{\otimes(2k+1)} \\
&= S(u_i^*) \cdot T(v_j^*).
\end{aligned}$$

Here

$$S(u_i^*) = \sum_{k=0}^{\infty} (-1)^k \sqrt{\frac{c^{2k+1}}{(2k+1)!}} (u_i^*)^{\otimes(2k+1)}$$

$$T(v_j^*) = \sum_{k=0}^{\infty} \sqrt{\frac{c^{2k+1}}{(2k+1)!}} (v_j^*)^{\otimes(2k+1)}$$

and

$$(u_1, u_2, u_3, u_4)^{\otimes(3)} = (u_1^3, u_1^2 u_2, u_1^2 u_3, u_1^2 u_4, u_1 u_2 u_3, \ldots).$$

Here

$$
\begin{aligned}
S(u_i^*) &= \sum_{k=0}^{\infty}(-1)^k\sqrt{\frac{c^{2k+1}}{(2k+1)!}}(u_i^*)^{\otimes(2k+1)}\\
T(v_j^*) &= \sum_{k=0}^{\infty}\sqrt{\frac{c^{2k+1}}{(2k+1)!}}(v_j^*)^{\otimes(2k+1)}
\end{aligned}
$$

and

$$
(u_1,u_2,u_3,u_4)^{\otimes(3)} = (u_1^3, u_1^2u_2, u_1^2u_3, u_1^2u_4, u_1u_2u_3, \ldots).
$$

Note that c has been chosen so that $||S(u_i^*)|| = ||T(v_j^*)|| = 1$.

$$
\begin{aligned}
L_A &= \sum_{i,j} \mathbf{A}_{i,j} u_i^* \cdot v_j^* \\
&= c^{-1} \sum_{i,j} \mathbf{A}_{i,j} \arcsin(S(u_i^*) \cdot T(v_j^*)) \\
&= c^{-1} \frac{\pi}{2} \sum_{i,j} \mathbf{A}_{i,j} \mathbf{E}[sign(S(u_i^*) \cdot z)\, sign(T(v_j^*) \cdot z)]
\end{aligned}
$$

$$
\begin{aligned}
L_A &= \sum_{i,j} \mathbf{A}_{i,j} u_i^* \cdot v_j^* \\
&= c^{-1} \sum_{i,j} \mathbf{A}_{i,j} \arcsin(S(u_i^*) \cdot T(v_j^*)) \\
&= c^{-1} \frac{\pi}{2} \sum_{i,j} \mathbf{A}_{i,j} \mathbf{E}[sign(S(u_i^*) \cdot z)\, sign(T(v_j^*) \cdot z)]
\end{aligned}
$$

We embed the $S(u_i^*), T(v_j^*)$ in R^{m+n} and choose z randomly and put $x_i = sign(T(u_i^* \cdot z), y_j = sign(T(v_j^*) \cdot z))$.

By choosing many z we get a good estimate of $||\mathbf{A}||_{\infty \to 1}$.

One can recover a good solution (x_i, y_j) by first deciding whether $x_1 = 1$ or $x_1 = -1$ etcetera.

$$\begin{aligned} L_A &= \sum_{i,j} \mathbf{A}_{i,j} u_i^* \cdot v_j^* \\ &= c^{-1} \sum_{i,j} \mathbf{A}_{i,j} \arcsin(S(u_i^*) \cdot T(v_j^*)) \\ &= c^{-1} \frac{\pi}{2} \sum_{i,j} \mathbf{A}_{i,j} \mathbf{E}[sign(S(u_i^*) \cdot z)\, sign(T(v_j^*) \cdot z)] \end{aligned}$$

One can extend the idea to approximate the cut-norm, with the same guarantee.

In Frieze, Kannan we gave a randomised algorithm for computing a weak partition using only $2^{\tilde{O}(\epsilon^{-2})}$ time.

In Frieze, Kannan we gave a randomised algorithm for computing a weak partition using only $2^{\tilde{O}(\epsilon^{-2})}$ time.

Subsequently, Alon,Fernandez de la Vega,Kannan, Karpinski, Yuster show how to compute such a partition using only $\tilde{O}(\epsilon^{-4})$ probes.

(A similar but weaker result was obtained by Anderson, Engebretson).

See also Borgs, Chayes, Lovász, Sós, Vesztergombi.

In Frieze, Kannan we gave a randomised algorithm for computing a weak partition using only $2^{\tilde{O}(\epsilon^{-2})}$ time.

Subsequently, Alon,Fernandez de la Vega,Kannan, Karpinski, Yuster show how to compute such a partition using only $\tilde{O}(\epsilon^{-4})$ probes.

(A similar but weaker result was obtained by Anderson, Engebretson).

See also Borgs, Chayes, Lovász, Sós, Vesztergombi.

Above also applies to Multi-Dimensional Arrays.

A multi-dimensional version

Max-r-CSP is the following problem: We are given m Boolean functions f_i defined on $Y_i = (y_1, y_2, \ldots, y_r)$ where $\{y_1, y_2, \ldots, y_r\} \subseteq \{x_1, x_2, \ldots, x_n\}$ and the aim to choose a setting for the variables $x_1, x_2, \ldots, x_n$ that makes as many of the functions f_i as possible, true.

A multi-dimensional version

Max-r-CSP is the following problem: We are given m Boolean functions f_i defined on $Y_i = (y_1, y_2, \ldots, y_r)$ where $\{y_1, y_2, \ldots, y_r\} \subseteq \{x_1, x_2, \ldots, x_n\}$ and the aim to choose a setting for the variables $x_1, x_2, \ldots, x_n$ that makes as many of the functions f_i as possible, true.

For each $z \in \{0,1\}^r$ and $(i_1, i_2, \ldots, i_r) \in [n]^r$ we define

$$\mathbf{A}^{(z)}(i_1, i_2, \ldots, i_r) = |\{j : Y_j = x_{i_1}, \ldots, x_{i_r} \text{ and } f_j(z_1, \ldots, z_r) = T\}|.$$

A multi-dimensional version

Max-r-CSP is the following problem: We are given m Boolean functions f_i defined on $Y_i = (y_1, y_2, \ldots, y_r)$ where $\{y_1, y_2, \ldots, y_r\} \subseteq \{x_1, x_2, \ldots, x_n\}$ and the aim to choose a setting for the variables $x_1, x_2, \ldots, x_n$ that makes as many of the functions f_i as possible, true.

For each $z \in \{0,1\}^r$ and $(i_1, i_2, \ldots, i_r) \in [n]^r$ we define
$\mathbf{A}^{(z)}(i_1, i_2, \ldots, i_r) = |\{j : \; Y_j = x_{i_1}, \ldots, x_{i_r} \text{ and } f_j(z_1, \ldots, z_r) = T\}|.$

Problem becomes to maximize, over $x_1, x_2, \ldots, x_n$,

$$\sum_{\substack{z \\ i_1, i_2, \ldots, i_r}} \mathbf{A}^{(z)}(i_1, i_2, \ldots, i_r)(-1)^{r-|z|} \prod_{t=1}^{r} (x_{i_t} + z_t - 1).$$

For this it is useful to have a decomposition for r-dimensional matrices: An r-dimensional matrix $\mathbf{A}$ *on* $X_1 \times X_2 \dots X_r$ is a map

$$\mathbf{A}: \; X_1 \times X_2 \cdots \times X_r \to \mathbf{R}.$$

If $S_i \subseteq X_i$ *for* $i = 1, 2, \dots r$, and d is a real number the matrix $\mathbf{M}$ satisfying

$$\mathbf{M}(e) = \begin{cases} d & \text{for } e \in S_1 \times S_2 \cdots \times S_r \\ 0 & \text{otherwise} \end{cases}$$

is called a cut matrix and is denoted

$$\mathbf{M} = CUT(S_1, S_2, \dots S_r; d).$$

B is the (2-dimensional) matrix with rows indexed by $Y_1 = X_1 \times \cdots \times X_{\hat{r}}$, $\hat{r} = \lfloor r/2 \rfloor$ and columns indexed by $Y_2 = X_{\hat{r}+1} \times \cdots \times X_r$.

For

$$\begin{aligned} i &= (x_1, \ldots, x_{\hat{r}}) \in Y_1 \\ j &= (x_{\hat{r}+1}, \ldots, x_r) \in Y_2 \end{aligned}$$

let

$$\mathbf{B}(i,j) = \mathbf{A}(x_1, x_2, \ldots, x_r).$$

Applying a decompositon algorithm we obtain

$$\mathbf{B} = \mathbf{D}^{(1)} + \mathbf{D}^{(2)} + \cdots + \mathbf{D}^{(s_0)} + \mathbf{W}$$

where for $1 \leq t \leq s_0$,

$$\mathbf{D}^{(t)} = CUT(R_t, C_t, d_t),$$

and $||\mathbf{W}||_\square$ is “small”.

Each R_t defines an $\hat{r}$-dimensional 0-1 matrix $\mathbf{R}^{(t)}$ where $\mathbf{R}^{(t)}(x_1,\dots,x_{\hat{r}}) = 1$ iff $(x_1,\dots,x_{\hat{r}}) \in R_t$. $\mathbf{C}^{(t)}$ is defined similarly. Assume inductively that we can further decompose

$$
\begin{aligned}
\mathbf{R}^{(t)} &= \mathbf{D}^{(t,1)} + \cdots + \mathbf{D}^{(t,s_1)} + \mathbf{W}^{(t)} \\
\mathbf{C}^{(t)} &= \hat{\mathbf{D}}t, 1 + \cdots + \hat{\mathbf{D}}t, \hat{s}_1 + \hat{\mathbf{W}}^{(t)}
\end{aligned}
$$

Here

$$
\begin{aligned}
\mathbf{D}^{(t,u)} &= CUT(R_{t,u,1},\dots,R_{t,u,\hat{r}}, d_{t,u}) \\
\hat{\mathbf{D}}t, \hat{u} &= CUT(R_{t,\hat{u},\hat{r}+1},\dots,R_{t,\hat{u},r}, \hat{d}_{t,\hat{u}})
\end{aligned}
$$

Each R_t defines an $\hat{r}$-dimensional 0-1 matrix $\mathbf{R}^{(t)}$ where $\mathbf{R}^{(t)}(x_1,\ldots,x_{\hat{r}}) = 1$ iff $(x_1,\ldots,x_{\hat{r}}) \in R_t$. $\mathbf{C}^{(t)}$ is defined similarly. Assume inductively that we can further decompose

$$\begin{aligned} \mathbf{R}^{(t)} &= \mathbf{D}^{(t,1)} + \cdots + \mathbf{D}^{(t,s_1)} + \mathbf{W}^{(t)} \\ \mathbf{C}^{(t)} &= \hat{\mathbf{D}}t,1 + \cdots + \hat{\mathbf{D}}t,\hat{s}_1 + \hat{\mathbf{W}}^{(t)} \end{aligned}$$

Here

$$\begin{aligned} \mathbf{D}^{(t,u)} &= CUT(R_{t,u,1},\ldots,R_{t,u,\hat{r}},d_{t,u}) \\ \hat{\mathbf{D}}t,\hat{u} &= CUT(R_{t,\hat{u},\hat{r}+1},\ldots,R_{t,\hat{u},r},\hat{d}_{t,\hat{u}}) \end{aligned}$$

It follows that we can write

$$\mathbf{A} = \sum_{t,u,\hat{u}} CUT(R_{t,u,1},\ldots,R_{t,u,\hat{r}},\hat{R}_{t,\hat{u},\hat{r}+1},\ldots,\hat{R}_{t,\hat{u},r},d_{t,u,\hat{u}}) + \mathbf{W}_1,$$

where $||\mathbf{W}_1||_\square$ is “small”.

THANK YOU