Contents

1 List decoding of Reed-Solomon codes

1.1 Introduction

An error-correcting code is $C \subseteq \Sigma^n$, where $\Sigma, |\Sigma| = q$ is the alphabet and n is the encoding length.

Define the normalized Hamming distance

$$d(x, y) = \frac{1}{n} |\{i : x_i \neq y_i\}|.$$

We want

$$\delta = \min_{x,y \in C, x \neq y} d(x, y)$$

to be as large as possible (constant as $n \to \infty$). Imagine balls of radius $\delta/2$ around each point. We can send one of $|C|$ messages by mapping $[|C|] \to \Sigma^n$; they can withstand $\delta/2$ errors. The rate of the code is $\frac{\lg |C|}{n}$ (how many bits of actual information are sent compared with encoding length); we want the rate and distance to be high, but there are fundamental limits to what can be achieved.

$\delta/2$ is the unique decoding radius. Unique decoding is not possible beyond $\delta/2$.

In many cases, even if there is not a unique codeword within a given radius ($\frac{\delta}{2}+\varepsilon$, say), there may be a only a small number of codewords.

Definition 1.1: C is (δ, L)-list decodable if for all $x \in \Sigma^n$, $|B(x,\delta)\cap C| \le L$.

If you choose your code randomly—that is, choose a codeword, removing a $\frac{\delta}{2}$ ball around it, and repeat—with high probability it will have good list decodability, up to distance $(1-\varepsilon)\delta$. List decoding is a better way to handle talk about how good the code is than stochastically, because it's worst-case.

The maximum δ is the list-decoding radius. Here $L(n)$ is a function of n. The ideal setting is $L(n) = \text{poly}(n)$. Informally the list decoding radius is the maximum δ such that $L(n) = \text{poly}(n)$.

What does a random code give us; what is possible and not possible? What is the capacity of list decoding?

Theorem 1.2: Let $0 < \delta < 1 - \frac{1}{q}$. (Above this error, the noisy word is essentially random, so we can say nothing.) Then there exists a (δ, L)-list decodable code with rate

$$1 - H_q(\delta) - \frac{1}{L}.$$

Here H_q is defined so that

$$|B(0^n,\delta)| \sim q^{H_q(\delta)n}.$$

Proof. Choose $M = q^{Rn}$ codewords at random. We want

$$\mathbb{P}(C \text{ not } (\delta, L)\text{-list decodable}) < 1.$$

Do a union bound. The probability is at most, by the union bound,

$$\binom{M}{L+1} q^n \left(\frac{|B(x,\delta)|}{q^n}\right)^{L+1} < 1.$$

(q^n is the number of points; the probability of $L+1$ points in a ball centered at a point is $\le \binom{M}{L+1}\left(\frac{|B(x,\delta)|}{q^n}\right)^{L+1}$.) □

This is almost tight.

Theorem 1.3: If a code is (δ, L)-list decodable with

$$R \ge 1 - H_q(\delta) + \varepsilon$$

then $L(n) \ge q^{\frac{\varepsilon n}{2}}$.

A random ball will contain too many points.

Proof.

$$\begin{aligned}\mathbb{E}_{x\in_R\Sigma^n}[|B(x,\delta)\cap C|] &= |C|\frac{|B(0^n,\delta)|}{q^n}\\ &\geq 2^{Rn}q^{n(H_q(\delta)-1)}\\ &\geq q^{\varepsilon n/2}.\end{aligned}$$

□

When $q\to\infty$, $H_q(\delta)\to\delta$. We can achieve rate $R\geq 1-\delta-\varepsilon$ with alphabet of size $2^{O\left(\frac{1}{\varepsilon}\right)}$ and list size $O\left(\frac{1}{\varepsilon}\right)$.

Graph: the achievable rates are below the line $\delta+R=1$.

This is what is known existentially.

The current known explicit codes can achieve $R\geq 1-\delta-\varepsilon$ with alphabet of size $2^{\text{poly}\left(\frac{1}{\varepsilon}\right)}$ and list size $n^{O\left(\frac{1}{\varepsilon}\right)}$. The idea is a folded Reed-Solomon code concatenated with an asymptotically good list-decodable code.

Theorem 1.4 (Johnson bound)**:** Given $C\subseteq\Sigma^n$ with min-distance $\delta=1-\varepsilon$, then C is $(1-\sqrt{\varepsilon},\text{poly}(n))$ list-decodable.

The Johnson bound says that if you slightly increase the radius, there cannot be an exponential number of codewords.

Theorem 1.5 (Singleton bound)**:** For any codewith distance δ, $R\leq 1-\delta$.

MDS (maximal distance separable) codes are where $R+\delta=1$, $\delta=1-R$, R= ε. MDS codes are $(1-\sqrt{R},\text{poly}(N))$ list-decodable.

1.2 Reed-Solomon Codes

We introduce the Reed-Solomon code $RS_{k,n,\mathbb{F}}$, $k<n\leq q$.

These are MDS codes. They achieve $R+\delta=1$ so are $(1-\sqrt{R},\text{poly}(n))$-list decodable.

Let $\Sigma=\mathbb{F}_q=:\mathbb{F}$. Let

$$C=\{\text{evaluations of degree } k \text{ polynomials on } \{\alpha_1,\ldots,\alpha_n\}\subseteq\mathbb{F}\}$$

where $\alpha_1,\ldots,\alpha_n$ are distinct and fixed. Codewords correspond to degree k polynomials in $\mathbb{F}[x]$. 2 distinct degree k polynomials can only agree on k points, so $\delta=\frac{n-k}{n}=1-\frac{k}{n}$. The rate is $\frac{\log_q(q^{k+1})}{n}=\frac{k+1}{n}$. The Reed-Solomon code is $(1-\sqrt{\frac{k}{n}},\text{poly}(n))$-list decodable.

What is the algorithm? Given a set of points, we need to find all polynomials passing through enough of those points.

Problem 1.6: Given $(\alpha_1,y_1),\ldots,(\alpha_n,y_n)$, find all polynomials of degree $\leq k$ such that $p(\alpha_i)=y_i$ for at least $\sqrt{nk}$ indices $i\in[n]$.

Madha Sudan (90's) showed that you can do this with $\sqrt{2nk}$.

Theorem 1.7: There is a polynomial-time algorithm (given in the proof) that given n points as above, finds all degree $\le k$ polynomials agreeing on $\ge 2\sqrt{nk}$ points.

Proof. The algorithm is as follows. Note this algorithm will work even if the α_i are not distinct.

Define the $(1, k)$-weighted degree of a polynomial $Q(x, y)$ as $\deg Q(X, Y^k)$. The strategy is as follows.

We will find a low $(1, k)$-weighted degree polynomial $Q(x, y)$ of degree D such that $Q(\alpha_i, y_i) = 0$ for all i.

Look at $R(x) = Q(x, p(x))$ where $p \in L$. We have

$$\deg R \le D.$$

For all i such that $p(\alpha_i) = y_i$,

$$R(\alpha_i) = Q(\alpha_i, p(\alpha_i)) = Q(\alpha_i, y_i) = 0.$$

Suppose R has at least t roots and $\deg R \le D$. If we arrange so that $t > D$, then $R(x) \equiv 0$ identically. Then $Q(x, p(x)) \equiv 0$ implies $y - p(x) \mid Q(x, y)$. (There is a deterministic algorithm to factor bivariate polynomials.) Factor R to find all factors in the form $y - p(x)$; then output those polynomials $p(x)$.

Now we just need to find the polynomial $R(x)$; this is polynomial interpolation.

The number of coefficients in $Q(x, y)$ is $\frac{1}{k}\binom{D+2}{2}$. We need to satisfy n linear constraints; they are linear homogeneous equations in the coefficients. If $\frac{\binom{D+2}{2}}{k} > n$ then there is a nonzero solution: a nonzero $\varphi(x, y)$ of $(1, k)$ weight degree $\le D$ with $\varphi(\alpha_i) = y_i$ for all i.

If the number of roots is $t > D \approx \sqrt{2kn}$, then we can find all polynomials with agreement t. □

Guruswami and Sudan improved this to $\sqrt{kn}$.

Consider $k = 1$, $\mathbb{F} = \mathbb{R}$, find all lines which pass through at least 3 points. If you can interpolate a degree 2 polynomial through all these points, get all lines as lines within the curve. The degree 2 curve will be 2 lines.

Let's look at a small example to see how to improve the bound.

Consider the following picture (see notebook). Here $n = 10$ and $t = 4$. Here we can choose $k = 4$. However it can only have 4 linear factors. The maximum D is 3. Peculiar: through each point there are 2 lines. An algebraic curve passing all points should vanish at the points with multiplicity 2. Fit a polynomial which vanishes with degree 2 at the points.

First define multiplicity.

Definition 1.8: $Q(x, y)$ vanishes with multiplicity r at (α, β) if $Q(x+\alpha, y+\beta)$ doesn't have any monomial of degree $\le r$.

Lemma 1.9: Let $Q(x, y)$ be with $(1, k)$ degree $\le D$, vanishing at (α_i, y_i) with multiplicity r for all $i \in [n]$. Let P be a degree k polynomial with agreement $t > D/r$. Then $y - p(x) \mid Q(x, y)$.

Proof. Let $p \in L$, $P(\alpha_i) = y_i$. Define

$$Q^i(x,y) = Q(x+\alpha_i, y+y_i);$$

it has no monomials of degree $< r$. Then

$$\begin{aligned} R(x) &= Q(x, P(x)) \\ &= Q^i(x-\alpha_i, P(x)-y_i) \\ &= Q^i(x-\alpha_i, \underbrace{P(x)-p(\alpha_i)}_{x-\alpha_i | P(x)-P(\alpha_i)}). \end{aligned}$$

Q^i has no monomials of degree $< r$. Thus $(x-\alpha_i)^r \mid R(x)$. The number of linear factors is $tr > D$. Thus $R(x) \equiv 0$ and $y - p(x) \mid Q(x,y)$. □

The number of coefficients in $Q(x,y)$ of $(1,k)$-degree D is $\frac{1}{k}\binom{D+2}{D}$. The number of homogeneous linear equations is $n\binom{r+1}{2}$. So a nonzero Q exists if $\frac{1}{k}\binom{D+2}{2} > n\binom{r+1}{2}$.

Choose $D = \sqrt{knr(r+1)}$. Let $t = \frac{D}{r} = \sqrt{kn(1+\frac{1}{r})}$.

Make r large enough, we approach the Johnson bound.

The number of polynomials in the list is at most $\frac{D}{k} = \sqrt{\frac{nr(r+1)}{k}}$, the y-degree.

Conclusion:

1. If $t > \sqrt{kn}$ the list size is $\le n^{\varepsilon}$ and we can find all of them.
2. The Reed Solomon code of rate R can be list decoded up to $1 - \sqrt{(1+\varepsilon)R}$ errors with best size $\frac{1}{\varepsilon\sqrt{R}}$.

This method (the polynomial method) is very flexible. We give another application, the list recovering problem. Given $x \in C$, a noisy channel turns each coordinate into a set s_i and spits out $s_1, \dots, s_n$, $|s_i| \le \ell$. We have $|s_i| \le l$ such that for at least $1-\delta$ fraction of i's, $x_i \in s_i$. Find all x such that $x_i \in s_i$ for at least $1-\delta$ fraction of i's.

$$\left| \left(\bigcup_{y \in S_1 \times \cdots \times S_n} B(y,\delta) \right) \cap C \right| \le L.$$

Reed-Solomon codes are also good list-recoverable codes.

If $t > \sqrt{knl}$, where t is the number of i such that $x_i \in S_i$, then the Reed-Solomon code is $(1-\sqrt{kl}, l, O(n^2l^2) = L)$ list recoverable.

We only achieve $1-\sqrt{R}$. We want to attain $(1-R, L)$. Guruswami and Rudra came up with folded Reed-Solomon codes. Take a generator γ of $\mathbb{F}_q^\times$, $n = q-1$. $\alpha_1, \dots, \alpha_n$ are $S = \{1, \gamma, \dots, \gamma^{q-1}\}$. Consider blocks. $\mathbb{F}^m$ by m. $P(1), \dots P(r), \dots, P(r^{q-1})$.

It's an open problem to make $n^{O(\frac{1}{\varepsilon})}$ independent of n. Use concatention to get length down.

You can concatenate a list-recoverable code with a list-decodable codes to get a list-decodable code. $C_{\text{out}} \circ C_{\text{in}}$.

2 $SL = L$

Reingold, Vadhan, and Wigderson.

There are many stories in this problem.

One of the most fundamental questions in theoretical CS is the following.

Claim 2.1: Randomness is useless.

This claim comes from very recent years. 20 years ago people actually believe it's useful (Papadimitriou gave $BPP \neq P$ as a homework exercise in his book).

There are 2 questions: is randomness useful in time and in space?

1. $BPP \overset{?}{P}$
2. $RL \overset{?}{L}$

We focus on the second problem. The first problem is wide open: we don't know $BPP \overset{?}{\in} DTIME(2^{o(n)})$. We know more about the second question: Savitch proves

$$BPP \subseteq DSPACE(\ln^2 n).$$

We can do better: Saks and Zhou in 1999 show

$$RL \subseteq DSPACE(\ln^{\frac{3}{2}} n).$$

(They actual show this for BPP.) Think of RL as (undirected) $s-t$ nonconnectivity problem. $coRL$ is interesting because it contains $s-t$ connectivity. (RL doesn't have a complete problem.) For $P \overset{?}{=} NP$, just look at a complete problem; here we don't have one.

Here we show $SL = L$. Reingold showed this in 2004.

For simplicity, just think of SL as one problem, undirected $s-t$ connectivity.

Consider a graph.

Jieming starts from a vertex and wants to find a way to Nanjing. Jieming has very little memory, and cannot remember the whole structure of the graph. He can't remember much more than the name of a city.

The standard way to solve this is to take a uniform random walk on the graph. If the graph has $|V| = n$, after $n^2 \ln n$ steps, there is a high probability that he will have visited Nanjing.

But for a general graph, it's necessary to flip $n^2 \ln n$ coins: Consider 2 complete graphs with a bridge: It takes order of n time to hit the bridge vertex, and it has $O\left(\frac{1}{n}\right)$ chance of crossing the bridge.

For which graphs can we do this randomized routing faster than the worst case? A natural family is the family of expaner graphs.

Expander graphs have rapid mixing under a random walk. There are 2 definitions.

Definition 2.2: A graph $G = (V, E)$ is a **λ-edge expander** if for all sets $S \subseteq V$, $|S| \le \frac{|V|}{2}$,

$$\frac{E(S, \overline{S})}{\text{Vol}(S)} \ge \lambda.$$

Here, $\text{Vol}(S) = \sum_{v\in S} \deg(v)$. (We'll focus on the simple case when the graph is d-regular, i.e, for all $v \in V$¡ $\deg(v) = d$.) (Pictorially, a subset of $G = (V,E)$ is very spiky, like a sea urchin.)

A graph $G = (V,E)$ is a **λ-spectral expander**if $\lambda_2(L(G)) \ge \lambda$. The Laplacian $L(G)$ is defined from the adjacency matrix A_G. A_G is defined by $(A_G)_{ij} = 1$ if there is a edge between i and j. M is the random walk matrix $\frac{1}{d}A_G$, and

$$L = I - M.$$

(Thinking of this matrix as an operator, $(Lx)_i = \frac{1}{d}\sum_{(i,j)\in E} x_{ij}$, it flows from a vertex to adjacent vertices.)

The Laplacian originates in differential geometry, $L = \text{div}\,\nabla$.
These 2 definitions are quite close.

Theorem 2.3: Let $h(G), \lambda(G)$ denote the edge and spectral expansions of G. Then

$$\lambda(G) \le h(G) \le \sqrt{2\lambda(G)}.$$

The LHS is trivial, the RHS is Cheeger's inequality.
People in CS use a third definition that is more useful.

Definition 2.4: A graph is a λ-expander if for all $x \perp u = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, x \ne 0$,

$$\left|\frac{x^T M x}{x^T x}\right| \le \lambda.$$

(For edge and spectral expansion we want λ to be large. For random walk expansion we want λ to be close to 0.) $\lambda(L)$ range from 0 to 2, while $\lambda(M)$ range from 1 to -1.

Note that

$$\left|\frac{x^T(I-L)x}{x^T x}\right| = \left|1 - \frac{x^T L x}{x^T x}\right|$$

so spectral and random-walk expansion are not quite the same. For bipartite graphs, $\lambda(M) = 1$: random walks are not mixing at all. The side you're on depends on the parity.

For the zig-zag product they consider the third definition.

2.1 Zig-zag product

Think of G_1 having large size and degree (N_1, D_1), G_2 having small size and degree (N_2, D_2), and suppose $D_1 = N_2$.

$$G_3 = G_1 \text{Ⓩ} G_2$$

has large size N_1N_2 and small degree, D_2^2.

Let $\lambda_M(G_i) = \lambda_i$ denote the random walk expansion. Then (a naive bound)

$$\lambda_M(G_3) \le \lambda_1 + \lambda_2 + \lambda_2^2.$$

It's easy to construct a large degree expander. It's easy to construct small-size expanders (try all possible graphs). It's easy to construct G_1, G_2, but an expander like G_3 is hard to construct. It takes as input 2 easy-to-construct expanders and outputs a harder to construct graph that's an expander.

Replace each vertex of the original graph with a copy of the second graph G_2. Its size is hence $N_1 N_2$.

Each vertex in G_3 is labeled $(v, u) \in G_1 \times G_2$. The (i, j)th neighbor is defined as follows. Take u' the ith neighbor of u in G_2, $u \xrightarrow{i} u'$. Take the u'th neighbor of v in G_1; move to the cloud of w; $v \xrightarrow{u'} w$ in G_2. Now consider $\widetilde{u}$ such that $w \xrightarrow{\widetilde{u}} v$ in G_1; we move to $(w, \widetilde{u})$. Now move $\widetilde{u} \xrightarrow{j} \widetilde{u}'$ in G_2. Define

$$(v, u) \xrightarrow{(i,j)} (w, \widetilde{u}').$$

What's the magic of the zig-zag product? We claim

$$A_3 := A(G_3) = \widetilde{A_2}\widetilde{A_1}\widetilde{A_2}$$

where $\widetilde{B_2} = I \otimes A_2$ has N_1 blocks and each is a copy of A_2, and $\widetilde{A_1}$ is a permutation (actually matching) matrix where $(\widetilde{A_1})_{((v,u),(w,u'))} = 1$ if

$$v \xrightarrow{u, G_1} w \xrightarrow{u', G_1} v.$$

Consider when $(u, v) \xrightarrow{(i,j)} (w, \widetilde{u}')$:

$$\begin{aligned} &u \xrightarrow{i, G_2} u' \\ &v \xrightarrow{u', G_1} w \xrightarrow{\widetilde{u}, G_1} v \\ &\widetilde{u} \xrightarrow{j} \widetilde{u}' \end{aligned}$$

What is the action of $\widetilde{A_2}\widetilde{A_1}\widetilde{A_2}$ on $e_{(v,u)}$? It goes to $\sum_{uu' \in G_2} e_{(v,u')} = e_v \otimes A_2 e_u$. "Stay in the block of v, move to all the neighbors of u."

In the second step, we move across blocks as given by $\widetilde{A_1}$.

We need to show

$$\forall x \in \mathbb{R}^{N_1 \times N_2}, x \perp u \implies \frac{x^T M_3 x}{x^T x} \le \lambda_1 + \lambda_2 + \lambda_2^2.$$

Decompose x as a vector that is uniform on each block,

$$x = \underbrace{\alpha \otimes u}_{\alpha^{\|}} + \underbrace{x'}_{\alpha^{\perp}}$$

where x' has blocks $\widetilde{\alpha_i} \perp u_i$. Then (noting $\widetilde{A_2}\alpha^{\|} = \alpha^{\|}$),

$$\begin{aligned}
\left\langle x, \widetilde{A_2}\widetilde{A_1}\widetilde{A_2}x \right\rangle &= \left\langle \widetilde{A_2}x, \widetilde{A_1}\widetilde{A_2}x \right\rangle \\
&= \left\langle \widetilde{A_2}(\alpha^{\|} + \alpha^{\perp}), \widetilde{A_1}\widetilde{A_2}(\alpha^{\|} + \alpha^{\perp}) \right\rangle \\
&= \left\langle \alpha^{\|} + \widetilde{A_2}\alpha^{\perp}, \widetilde{A_1}(\alpha^{\|} + \widetilde{A_2}\alpha^{\perp}) \right\rangle \\
&= \left\langle \alpha^{\|}, \widetilde{A_1}\alpha^{\|} \right\rangle + \left\langle \alpha^{\|}, \widetilde{A_1}\widetilde{A_2}\alpha^{\perp} \right\rangle + \left\langle \widetilde{A_2}\alpha^{\perp}, \widetilde{A_1}\alpha^{\|} \right\rangle + \left\langle \widetilde{A_2}\alpha^{\perp}, \widetilde{A_1}\widetilde{A_2}\alpha^{\perp} \right\rangle \\
&\le \lambda_1 + 2\lambda_2 + \lambda_2^2
\end{aligned}$$

Because $\alpha_1 + \cdots + \alpha_{N_1} = 0$ we can use the expansion properties of G_1 to get the λ_1 bound for the first term.

The idea:

1. Suppose G has parameters (N, D) and expansion λ. (say $1 - \frac{1}{ND}$)

2. G^2 has parameters (N, D^2), expansion λ^2.

3. Take H with parameters $(D^2, \sqrt{D}), \lambda_2$. Then

   $$G^2 \textcircled{Z} H = G_{\text{new}}$$

   has parameters (ND^2, D) expansion $\lambda(G_{\text{new}}) \le \lambda^2 + 2\lambda_2 + \lambda_2^2 \le \lambda^2 + \varepsilon$.

We go from G with parameters (N, D, λ) to G_{new} with parameters $(ND, D, \lambda^2 + \varepsilon)$. Call this operator Z. Doing this operator t times, $Z^t G = (ND^{2t}, D, \lambda^{2^t} + \varepsilon')$. Then set $t = \lg\left(\frac{ND}{2}\right)$ to get $Z^t G$ with parameters $(N^2 D, D, \frac{1}{e} + \varepsilon)$. (We're cheating a little, using the naive bound. We need a better bound to get ε small: G_3 has expansion $\frac{1}{2}(1 - \lambda_2^2)\lambda_1 + \frac{1}{2}\sqrt{(1-\lambda_2^2)^2\lambda_1^2 + t\lambda_2^2}$. To go from our proof to the real proof, you just need to note $\alpha^{\|} \perp \alpha^{\perp}$, $\alpha^{\|} \perp \widetilde{A_2}\alpha^{\perp}$. Plug in this picture into the formula.

3 $O(\sqrt{\log n})$ approximation for sparsest cut by Arora, Rao and Vazirani

The outline of the talk is as follows.

1. Define the problem.

2. Give a LP relaxation (L, R) giving a $O(\ln n)$ approximation.

3. L_1 metrics and sparsest cut

4. Give the SDP relaxation due to Arora, Rao, and Vazirani.

3.1 Problem

Definition 3.1: Given a graph $G = (V, E)$, a capacity function $c_e : E \to \mathbb{R}^+$, and $d : V \times V \to \mathbb{R}^+$, find

$$\min_{S \subseteq V} \frac{\sum_{(i,j)\in\delta(S)} c_{(i,j)}}{\sum_{i,j\in\delta(S)??} d_{(i,j)}}.$$

For simplicity, we consider uniform sparsest cut $c_e = 1, d_{ij} = 1$, so we want

$$\phi(S) = \min_{S\subseteq V} \frac{\delta(S)}{|S||V-S|}.$$

This is closely connected to the edge expansion $\alpha(G) = \min_{S\subseteq V, |S|\le \frac{n}{2}} \frac{\delta(S)}{|S|}$. Approximating ϕ, α are equivalent up to a constant factor.

It's NP-hard to find the sparsest cut exactly.

3.2 Metrics

Definition 3.2: A **metric** is (X, d) with

$$d(x,z) \le d(x,y) + d(y,z),$$

$d(x,y) = 0$ iff $x = y$. (A semimetric only needs satisfy $d(x,y) = 0$ if $x = y$.)

There is a natural semimetric one can define on the space of cuts. Assign a distance of 1 if they are separated by 1:

$$d_S(x,y) = |1_S(x) - 1_S(y)|.$$

There is now a nice interpretation

$$\phi(G) = \min_S \frac{\sum_{e\in E,(i,j)} d_S(i,j)}{\sum_{i,j} d_S(i,j)}.$$

We can instead relax this to minimizing over all semimetrics d, rather than just metrics of the form d_S. Call this $LR(G)$. We show

$$LR(G) \le \phi(G) \le O(\ln n) LR(G).$$

This can be formulated as a LP.

Find

$$\min \sum_{(i,j)\in E} d(i,j)$$

subject to $d(i,k) \le d(i,j) + d(j,k).$, $\sum_{i,j} d(i,j) = n^2$, $d(i,j) \ge 0$, $d(i,i) = 0$, $d(i,j) = d(j,i)$. This can be solved in polynomial time.

Now we prove the inequality.

A finite L_1 metrics is $X \subseteq \mathbb{R}^d$ with $d(x,y) = \|x-y\|_1$. Given any metric d we say there is an embedding $f : d \to L_1$ with distortion α if

$$\|f(x) - f(y)\|_1 \le d(x,y) \le \|f(x) - f(y)\|_1 \alpha.$$

L_1 metrics capture the sparsest cut problem.

Given any embedding f to L_1, there exists S, d_S,

$$\frac{\sum_e d_S(i,j)}{\sum_{i,j} d_S(i,j)} \le \frac{\sum_e \|f(x) - f(y)\|_1}{\sum_{i,j} \|f(i) - f(j)\|_1}.$$

RHS is

$$\ge \min_i \frac{\sum_e |f_i(x) - f_i(y)|}{\sum_{i,j} |f_i(i) - f_i(j)|}.$$

$f_i(i) = g_i$. You can scale and shift so that $|\max g(i) - \min g(i)| = 1$. Fix a threshold d, take the embedding on a line, and take the cut defined by that threshold. Let i^* be where the minimum is attained. It equals

$$\frac{\mathbb{E}_t \sum_{e=\{i,j\}} d_{s_t}(i,j)}{\mathbb{E}_t \sum_{i,j} d_S(i,j)} \ge \frac{\sum d_{s_t}(i,j)}{\sum_{i,j}}.$$

Now we use

Theorem 3.3 (Bourgain)**:** For every semimetric d there is an embedding $f : d \to L_1$ with distortion $O(\ln n)$.

The tight example comes from analyzing the LP. A constant degree expander.

Then

$$\|f^*(x) - f^*(y)\|_1 \le d(x,y) \le O(\ln n) \|f^*(x) - f^*(y)\|_1 .$$

We get

$$\le \frac{\sum_e \|f^*(x) - f^*(y)\|_1}{\sum_{x,y} \|f^*(x) - f^*(y)\|_1} \le O(\ln n) \frac{\sum_e d(x,y)}{\sum_{x,y} d(x,y)}$$

Cone generated by semimetrics contains every L_1 metric.

Another relaxation is the spectral relaxation. Consider the normalized adjacency matrix A. The spectral gap is $1 - \lambda_2$. The spectral gap is

$$\sum_{x:V\to\mathbb{R}} \frac{\sum_C |x(i) - x(j)|^2}{\sum_{i,j} |x(i) - x(j)|^2}.$$

If we embed $x : V \to \mathbb{R}^n$ we use the L_2 norm.

What if we combine these two relaxations: impose the triangle inequality constraint, etc. We ask that $\|x(i) - x(k)\|^2 \le \|x(i) - x(j)\|^2 + \|x(j) - x(k)\|$.

$$ARV(G) = \sum_{x:V\to\mathbb{R}^n, \triangle \text{ inequalities}} \frac{\sum_C \|x(i) - x(j)\|^2}{\sum_{i,j} \|x(i) - x(j)\|^2}.$$

ARV showed

$$ARV(G) \leq \phi(G) \leq O(\sqrt{\ln n})ARV(G).$$

The lower bound is

$$\phi(G) \geq \Omega(\ln \ln n)ARV(G).$$

A cycle makes Cheeger's inequality tight. If you take a cycle and use the L_1 relaxation, it's not tight. (?) Tight ones are constant degree expanders. What is the balance between these two? ARV shows the union does give you some improvement.

We prove $\phi(G) \leq O(\sqrt{\ln n})ARV(G)$. We show the inequality $\phi(G) \leq O((\ln n)^{\frac{2}{3}})ARV(G)$ as it contains the main ideas but is similar.

We show ARV can be computed in polynomial time.

Assign vectors $x(i) \in \mathbb{R}^n$ to minimize:

$$\min \sum_e \|x(i) - x(j)\|^2$$

such that

$$\sum_{i,j} \|x(i) - x(j)\|^2 = n^2$$

and for all i, j, k,

$$\|x(i) - x(k)\|^2 \leq \|x(i) - x(j)\|^2 + \|x(i) - x(k)\|^2 .$$

Find a cut such that the sparsity of the cut is at most $O(\sqrt{\ln n})$.

The rounding strategy is as follows. Suppose we can find an embeding $f : V \to \mathbb{R}$

$$\frac{\sum_e |f(i) - f(j)|}{\sum_{i,j} |f(i) - f(j)|} \leq O(\sqrt{\ln n})ARV(G).$$

Then we're done. The strategy for finding such an embedding: Let v_i the vector associated with vertex i by the SDP. Look at the metric

$$d(i, j) = \|v_i - v_j\|^2 .$$

The strategy is to find a set S such that defining $f(i) = d(i, S) = \min_{j \in S} d(i, j)$.

An easy inequality by the triangle inequality.

$$f(i) - f(j) \leq \|v_i - v_j\|^2$$

$$ARV(G) = \frac{\sum_e \|v_i - v_j\|^2}{\sum_{i,j} \|v_i - v_j\|^2}.$$

Numerator inequality ok. Average distortion. Need to show

$$\sum_{i,j} f(i) - f(j) \geq \frac{\sum \|v_i - v_j\|^2}{\sqrt{\ln n}}.$$

later proved by ALN (2008): $L_2^2 \to L_1$ with distortion $O(\sqrt{\ln n} \ln \ln n)$.

$L_2^2 \to L_2 \to L_1$. Second one is isometric.

Here we only need average case distortion.

We're trying to produce 2 disjoint sets L and R of vertices both of size $\Omega(n)$ separated by $\Delta = \frac{1}{\sqrt{\ln n}}$. Two sets that are far away. Let's do this for $(\ln n)^{\frac{2}{3}}$. (Separated in L_2^2 metric.) Then the equation is satisfied: look at summation between pairs coming from L and R. (Define $S = L$.)

Supposer there exists some vertex i^* such that $|B(i^*, \frac{1}{4})| \geq \frac{n}{4}$. Set $S = B(i^*, \frac{1}{4})$. Then the inequality holds:

$$\begin{aligned} n^2 &= \sum_{i,j} d(i,j) \\ &\leq \sum_{i,j} d(i,i^*) + d(j,i^*) \\ &= 2n \sum_i d(i,i^*) \\ &\leq 2n(\sum_i d(i,S) + \frac{1}{4}). \end{aligned}$$

Gives $\sum_i d(i,S) \geq \frac{n}{4}$. This gives

$$\sum_{j \in S} d(j,S) \geq \frac{n}{4}.$$

Get $|S| \sum_{j\in S} d(j,S)$. At least $\frac{n^2}{16}$.

Suppose there does not exist a point like this.

Algorithm to produce L, R: Extension of Goemans-Williamson rounding. Put points depending on side of hyperplane. Take a random line, project, if $> \sigma$ put on one side, if $< -\sigma$ but on other side.

Define $O = \operatorname{argmax}_{O \in V} B(O, 4)$. Now pick a random Gaussian $r_i \sim N(0,1)$. Let

$$\begin{aligned} L &= \{i : (v_i - v_o)\cdot\gamma \geq \sigma\} \\ R &= \{j : (v_j - v_o)\cdot \geq\leq -\sigma\}\,. \end{aligned}$$

Assume $\Delta = \frac{1}{(\ln n)^{\frac{3}{2}}}$. Find a nonseparated pair, throw it out. Prune until Δ-separated. By definition, 2σ-separated in sense $(v_i - v_j)\cdot\gamma \geq 2\sigma$. σ is a constant. $\sigma > \delta$. $\gamma_i \sim N(0,1)$ not unit vector. But expectation is $\sqrt{n}$. In expectation

$$\mathbb{E}[\|(v_i - v_j)\cdot\gamma\|^2] = \|v_i - v_j\|^2\,.$$

easy to show $\ln n$ approximation by concentration. Without needing Bourgain's result.

Let L', R' be the bad pairs.

Theorem 3.4: With constant probability L and R are both $\Omega(n)$.

2 constant size disjoint which are both constant distance away from o. Project own to... large.

Condition on it.

Theorem 3.5: With good probability, L', R' are $\Omega(n)$ large.

Show by contradiction. Given L, R, suppose L', R' are small, i.e., $L', R' \leq \frac{\alpha}{2}n$. Show that for constant fraction, can come up with v_i, v_j such that $(v_i - v_j)\cdot\gamma$... Why $\ln n$ approximation? Take $\delta = \frac{1}{\ln n}$. ? cannot satisfy approximation. Suppose v_i, v_j, $\mathbb{P}((v_i - v_j)\cdot\gamma \geq t) \leq e^{-\frac{t^2}{\|v_i - v_j\|^2}}$.

$$\|v_i - v_j\|^2 \leq \frac{C}{\Delta} \leq \frac{C}{\ln n}$$

$$e^{-\sigma^2(\ln n)} \leq n^{-4}.$$

Any 2 pairs less than $\ln n$ away cannot have projection $> \sigma$. Whp no pairs from it removed.

Proving $(\ln n)^{\frac{3}{2}}$: construct points that are Tke many pairs throw out. Each time throw out from fixed L, R, consider it a matching. Put matchings together, becomes a weighted graph. Find path $v_{i_1}, \ldots, v_{i_k}$. k-length paths

$$\mathbb{P}((v_{i_k} - v_{i_1}) \geq \sigma k).$$

Show $\|v_i - v_j\|^2 \leq k\Delta$. Discrepance between squares.

$$e^{-\frac{\sigma^2 k}{\Delta}}.$$

As you build paths. With constant probability can find these paths. Concentration of measure, reduce projection, increase probability... extend paths... At least Δ fraction of edges going out of each vertex. Directed graph.

Vertex i, edge $2j$. fixing γ, bipartite directed graph, matching. Find paths in this graph which have these large projections.

4 Cryptography in NC^0

By Applebaum, Ishai, Kushilevitz, 2004.

Question: Are there OWF's in NC^0?

Definition 4.1: A **one-way function** satisfies the following.

1. easy to compute: there is a polynomial time algorithm to compute it. (Here, though, we want each bit to be computable in NC^0.)
2. hard to invert: for any probabilistic polynomial time algorithm A,

$$\mathbb{P}_x[f(A(f(x))) = f(x)] < O\left(\frac{1}{n^k}\right).$$

NC^0_c means NC^0 with locality c (each output bit depends on c input bits).

4.1 Previous work

Linial, Mansour, and Nisan in 1989 showed there is no PRF in AC^0. (PRF's imply one-way functions.)

Cryan and Mitterson, 2001 showed there is no PRG in NC_2^0, and no PRG in NC_3^0 chieving superlinear stretch.

On the positive side, Impagliazzo and Naor in 1996 showed there is a PRG in AC^0 based on the hardness of subset sum: is there a subset of a set that sums to a certain number?

Goldreich in 2000 showed there is no OWF in NC_2^0.

Result: Any OWF in NC^1 or even $\bigoplus L/\operatorname{poly}$ can be compiled into an OWF in NC_4^0.

Barrington's Theorem shows $NC^1 \subseteq L/\operatorname{poly}$. L/poly (log space with polynomial-sized advice) is exactly the class of PBP's. NC-circuit, can also be computed by a poly-size branching program.

4.2 Mod-2 branching program

Definition 4.2: A **mod-2 branching program** is $(G = (V, E), \phi, s, t)$ where G is an acyclic directed graph, $s, t \in V$, ϕ is a labeling assigning each edge $x_i, \overline{x_i}, 1$. On input $(w_1, \ldots, w_n)$, define $G_w = (V, E_w), e \in E_w$ iff $\phi(e)$ is satisfied by w. Let $f(w)$ be the number of paths from s to t mod 2.

The main technique is randomized encodings.

Definition 4.3: A **randomized encoding** of $f : B^n \to B^l$ is

$$\widehat{f} : B^t \times B^m \to B^s$$

that satisfies the following.

1. δ-correctness: there exists a deterministic algorithm C (the decoder) such that for all $x \in B^n$,
$$\mathbb{P}[C(\widehat{f}(x, U_m)) \neq f(x)] \leq \delta.$$
2. ε-privacy: there exists an algorithm S (simulator) such that for all $x \in B^n$,
$$\left\| S(f(x)) - \widehat{f}(x, U_m) \right\| \leq \varepsilon.$$
(It does not have additional information about the input.)

A perfect randomized encoding has

1. $\delta = \varepsilon = 0$,
2. is balanced: $S(U_L) = U_S$.
3. stretch-preserving: $L - n = S - n - m$.

Lemma 4.4: Let $\widehat{f}$ be a perfect randomized encoding for f. If f is hard to invert, then $\widehat{f}$ is hard to invert.

Proof. Suppose $\widehat{B}$ inverts $\widehat{f}(x, r)$. We obtain B that inverts $f(x)$ as follows.

1. Get input $y = f(x)$.
2. Let $\widehat{y} = S(y)$.
3. Run $\widehat{B}$ on $\widehat{y}$ and get (x', r').
4. Output x'.

Perfect correctness gives the following property: If $f(x) \neq f(x')$ then the support of $\widehat{f}(x, U_m)$ and $\widehat{f}(x', U_m)$ are disjoint. □

Now we only have to construct, given $f \in \oplus L/\text{poly}$, a perfect RE $\widehat{f}$ in NC^0_4. The steps are as follows.

1. Given a mod-2 BP, construct a degree-3 randomizing polynomials.

   Letting the BP be (G, ϕ, s, t) of size L, let $A(x)$ be the $L \times L$ adjacency matrix of G_x. Now let $L(x)$ be the submatrix of $A(x) - I$ obtained by removing the first column and last row.

   Fact: $f(x) = \det(L(x))$. Proof:

$$\begin{aligned} f(x) &= (I + A(x) + \cdots + A(x)^L)_{s,t} \\ &= (\overset{(I - A(x)^{L+1})}{\nearrow}\!\!\!\!\emptyset(1 - A(x))^{-1})_{s,t} \\ &= (I - A(x))^{-1}_{s,t} \\ &= (-1)\frac{\det(-L(x))}{\det(I - A(x))} = \det(L(x)). \end{aligned}$$

   $L(x)$ is not very private; it tells it a lot about the input.

   Fact: Let M, M' be $(L-1) \times (L-1)$ matrices with -1 on the second diagonal (below the diagonal) with 0's below. Then

$$\det(M) = \det(M')$$

   iff there exist $r^{(1)}, r^{(2)}$ such that $R_1(r^{(1)}) M R_2(r^{(2)}) = M'$. Here R_1 packs the entries of $r^{(1)}$ above the diagonal and has 1's on the diagonal, and R_2 packs the entries of $r^{(2)}$ into the last column and has 1's on the diagonal.

   Proof: $\Leftarrow$ is trivial.

   $\implies$: there exist $r^{(1)}$ and $r^{(2)}$ with $R_1(r^{(1)}) M R_2(r^{(2)})$ having $\det(M)$ in the upper-right corner, -1 on the diagonal, and 0's everywhere else. (?) M' satisfies the same condition.

   $R_1(\cdot), R_2(\cdot)$ form a multiplicative group.

   Lemma 4.5: Suppose a BP computes f. Let $\widehat{f}(x, (r^{(1)}, r^{(2)}))$ be the $\binom{L}{2}$ entries of $R_1(r^{(1)}) L(x) R_2(r^{(2)}) = M$. Then $\widehat{f}$ is a perfect randomized encoding of f.

Proof. (a) Perfect correctness: The decoder outputs $\det(M) = \det(L(x)) = f(x)$.

(b) Simulator: $S(y) = R_1(r^{(1)})MR_2(r^{(2)})$ where M has y in the upper right-hand corner.

$S(f(N))$ and $\widehat{(}f, U_m)$ have same support. The support size is $2^{\#\text{ of random bits}}$.

$S(U_1) = U_{\binom{L}{2}}$. Each output bit is a degree-3 polynomial. □

2. Given a degree 3 polynomial, construct a function in NC_4^0. Write

$$f(x) = T_1(x) + \cdots + T_k(x)$$

where T_i re degree 3 monomials. Then

$\widehat{f}(x_1(r_1,\ldots,r_k,r'_1,\ldots,r'_{k-1})) = (T_1(x)-r_1,\ldots,T_k(x)-r_k, r-r'_1, r'_1+r_2-r'_2,\ldots,r'_{k-2}+r_{k_1}-r'_{k-1}, r'_{k-1}+r$

Proof. (a) Decoder sum.

(b) $y = (r_1,\ldots,r_{2k}, y - \sum_{i=1}^{2k} r_i$.

(c) Balanced length preserving.

□

Impractical: 3 digit times 3 digits requires 10000 modes. Blowup is n^7. Number of output bits too large.

Problem: NC_3^0?

Where use randomness? If some function is OWF, rhan.. lso.

5 PRIMES is in P

By Agarwal, K, and Saxena.

We start with this observation.

Theorem 5.1: n is prime iff (for any $a \perp n$, a relatively prime to n)

$$(X+a)^n \equiv X^n + a \pmod{n}.$$

(Equality of polynomials: each coefficient matches.)

Proof. If n is prime then $\binom{n}{i} \equiv 0 \pmod{n}$ for all $1 \le i \le n-1$ and $a^n \equiv a \pmod{n}$ (Fermat's little theorem).

If n is not prime consider some prime $p \mid n$. Suppose $p^r || n$. We claim that $n \nmid \binom{n}{p^r}$. We show this with Lucas's Theorem. □

This gives a primality test, but takes time $O(n)$.

The trick is the following. Instead check if

$$(X+a)^n \equiv X^n + a \pmod{n, X^r - 1}.$$

We show that if there exists r such that for $O(\ln^{O(1)} n)$ many a the above holds, then n is prime, and conversely.

The algorithm is as follows.

1. Check if n is a perfect power $a^b, b > 1$. (There are $\lg n$ possible values for b.)
2. Find the smallest r such that $\operatorname{ord}_r n \geq (\ln n)^2$. Here $\operatorname{ord}_r n = \min\{a > 0 : n^a \equiv 1 \pmod{r}\}$.
3. Check for all $1 \leq a \leq \sqrt{\varphi(r)} \ln n =: l$ that

   $$(X+a)^n \equiv X^n + a \pmod{n, X^r - 1}.$$

   (Note you do repeated squaring, there are only r coefficients to remember.)

If n passes all these checks, n is prime.

We want to show that $r = O(\ln^{O(1)} n)$.

If n is composite; we show it can't pass all the tests.

Theorem 5.2: $r \leq (\ln n)^5$.

Proof. Suppose by way of contradiction that for all $r \leq (\ln n)^5$, $\operatorname{ord}_p n < (\ln n)^2$. If $\operatorname{ord}_p n = a$, $r \mid n^a - 1$. Hence

$$\operatorname{lcm}(1, \ldots, (\ln n)^5) \mid \prod_{i=1}^{(\ln n)^2} (n^i - 1) \leq n^{(\ln n)^4} = 2^{(\ln n)^5}.$$

However

$$\operatorname{lcm}(1, \ldots, m) \geq 2^{m-1}.$$

□

Proof. Assume $p \mid n$, $p < n$. WLOG $\operatorname{ord}_r p > 1$. Assume the first equation is true. The second is also true since p is prime.

$$\begin{aligned}(X+a)^n &\equiv X^n + a \pmod{n, X^r - 1}\\ (X+a)^p &\equiv X^p + a \pmod{n, X^r - 1}\end{aligned}$$

The first equation is

$$((X+a)^p)^{\frac{n}{p}} \equiv (X^p)^{\frac{n}{p}} + a \iff (X^p + a)^{\frac{n}{p}} \qquad = (X^p)^{\frac{n}{p}} + a.$$

Choose q such that $pq \equiv 1 \pmod{r}$. Then

$$\begin{aligned}(X^{pq} + a)^{\frac{n}{p}} &\equiv (X^{pq})^{\frac{n}{p}} + a\\ (X+a)^{\frac{n}{p}} &\equiv X^{\frac{n}{p}} + a.\end{aligned}$$

We want to look at exponents for which things like this are true.

Definition 5.3: Given a polynomial $f(X)$ and an integer m, m is **introspective** for $f(X)$ if

$$f(X^m) \equiv f(X)^m \pmod{n, X^r - 1}.$$

We showed $n, p, \frac{n}{p}$ are introspective for $X + a$ for all $a, 1 \le a \le l$. Introspective numbers are closed under multiplication.

Theorem 5.4: If m and m' are introspective for f then so is mm'.

Proof.

$$\begin{aligned} f(X)^{mm'} &\equiv f(X^m)^{m'} \pmod{n, X^r - 1} \\ f(X^m)^{m'} &\equiv f(X^{mm'}) \pmod{n, X^{mr} - 1} \\ f(X^m)^{m'} &\equiv f(X^{mm'}) \pmod{n, X^r - 1} \end{aligned}$$

becuse $X^r - 1 \mid X^{mr} - 1$. □

Theorem 5.5: If m is introspective for f_1 and f_2, then m is introspective for $f_1 f_2$.

Proof.

$$f_1(X^m) f_2(X^m) \equiv f_1(X)^m f_2(X)^m \equiv (f_1(X) f_2(X))^m.$$

□

These generate a lot of introspection. This gives many polynomial relations, which is bad.

We define 2 groups based on introspection.

- Define $H \subseteq (\mathbb{Z}/r)^\times$ generated by $p, \frac{n}{p}$ (both these are relatively prime to r).
- (Annoyingly, $\mathbb{Z}[X]/\langle n, X^r - 1\rangle$ is not a field. We convert it to a field in the following way.) Let $h(X)$ be an irreducible factor of the rth cyclotomic polynomial modulo p. (It is not linear because the order is > 1.) Let $\frac{F_p[X]}{\langle h(X)\rangle}$. (I.e., adjoin a rth root of unity to $\mathbb{F}_p$.) Let $G \subseteq F^\times$ be generated by $(X+1), \dots, (X+l)$ in F.

We upper-bound and lower-bound G to get a contradiction.

Let $t = |H|$. Since $\operatorname{ord}_r n \ge (\ln n)^2$, $t \ge (\ln n)^2$.

Theorem 5.6: $|G| \ge 2^t$.

Proof. Consider all products of $(X + i)$ of degree $\le t$. We claim all these products are distinct in G.

By way of contradiction, suppose we have 2 products $f(X) = g(X)$ in G (F). Choose $m \in H$. By construction of H, m is introspective for $f = g$: $f(X)^m = g(X^m)$ so $f(X^m) = g(X^m)$. Let $Q(Y) = f(Y) - g(Y)$ in $\mathbb{F}_p[Y]$. ($f, g \in \mathbb{F}_p[X]$.)

This implies X^m is a root of $Q(Y)$ for all $m \in H$. The X^m are all distinct because everything in H is relatively prime to r.

Thus $Q(Y)$ has at least t roots. This is bad since the degree is $< t$.

We get a contradiction since $l \ge t$; there are at least 2^t different products. □

Theorem 5.7: $|G| \leq n^{\sqrt{t}}$.

Proof. Consider the set

$$\widetilde{I} = \left\{ p^i \left(\frac{n}{p}\right)^j : 0 \leq i, j \leq \sqrt{t} \right\}.$$

By Pigeonhole there exist $m_1, m_2 \in \widetilde{I}$ such that $m_1 \equiv m_2 \pmod{r}$. Then $X^{m_1} = X^{m_2}$. Since $f(X) \in G$,

$$f(X)^{m_1} = f(X^{m_1}) = f(X^{m_2}) = f(X)^{m_2}.$$

Let $Q(Y) = Y^{m_1} - Y^{m_2}$. Then $f(X)$ is a root of Q for all $f \in G$. But $\deg Q \leq \max(m_1, m_2) \leq n^{\sqrt{t}}$. Hence $|G| \leq n^{\sqrt{t}}$.

(This only works if n is not a power of p.) □

These 2 theorems together imply that $2^t \leq |G| \leq n^{\sqrt{t}}$. However, $t > (\ln n)^2$, so

$$n^{\sqrt{t}} < 2^t \leq |G| \leq n^{\sqrt{t}}$$

That's it! □

In the paper they get $\frac{t+l}{t-1}$. Bur for $l \geq t$, it's $\geq 2^t$.

6 Optimal inapproximability of MAXCUT from UGC

This is a paper from [KKMO04].

Goemans-Williamson achieves $\alpha \approx 0.878$-approximation to MAXCUT.

If UGC is true than approximating MAXCUT better than $\alpha_{GW} + \varepsilon$ is NP-hard for every $\varepsilon > 0$.

The way to show it is the following: Look at $\mathsf{Gap} - \mathsf{MaxCut}_{(c,s)}$, $c > s$. If this is NP-hard for some parameters c, s, then a $\frac{c}{s}$ approximation is NP-hard.

They prove that for all $\varepsilon > 0, -1 < \rho < 0$, $\mathsf{Gap} - \mathsf{MaxCut}_{(\frac{1-\rho}{2} - \varepsilon, \frac{\cos^{-1}\rho}{\pi} + \varepsilon)}$ is NP-hard assuming UGC.

Unconditionally, $(\frac{16}{17} + \varepsilon)$-approximation of Gap-MaxCut is NP-hard.

Maximizing the bound gives the GW bound:

$$\max_{-1<\rho<0} \frac{\left(\frac{1-\rho}{2}\right)}{\cos^{-1}\frac{\rho}{\pi}} = \alpha_{GW} \approx 0.878.$$

The bad case for GW gives the same bad case here.

We construct a PCP-like theorem. For every $-1 < \rho < 0, \varepsilon > 0$, construct a 2-query PCP system with alphabet $\{-1, 1\}$ and completeness $c = \frac{1-\rho}{2} - \varepsilon$ and soundness $s = \frac{\cos^{-1}\rho}{\pi} + \varepsilon$ in which the verifier.

The verifier looks at 2 bits and depending on the 2 bits, accepts or rejects. If x is in the language, accept with probability $\geq c$ and if not, accept with probability $\leq s$, using only $\neq$ checks.

What is MaxCut? It's some kind of equality constraint.

Given a PCP, estimate the proof that will make the verifier accept with the highest probability. The probability of acceptance is the max cut in the graph constructed as follows: the vertices are the number of bits. The weight on v_iv_j is $\mathbb{P}\left(\text{verifier queries } i, j\right)$. The max cut is the probability of acceptance.

If there is PCP system, you can convert it into a max cut system. Estimate $\max_{\text{proofs}} \mathbb{P}(V \text{ accepts } p)$.

If L is a NP-hard language, then we are done. We don't know existing NP-hard problems with this property, so we start with UGC instead.

Consider Gap-UniqueLabelCover$(\Sigma)_{(1-\delta,\delta)}$. Consider a bipartite graph $G = (V \cup W, E)$. Each edge has a permutation $\pi_{vw} : \Sigma \to \Sigma$ that is left regular. We want $\sigma : V \cup W \to E$ such that for all $(v, w) \in E$, $\pi_{vw}(\sigma(w)) = \sigma(v)$. The value is the maximum fraction of edges satisfiable.

For example, say $\Sigma = \mathbb{F}_q$. The constraint is that $\sigma(v) + \sigma(w) = c_{vw}$. $\mathsf{Max2Lin}(\mathbb{F}_q)$. Label cover is NP-hard.

Conjecture 6.1 (UGC (Khot))**:** For all $\delta > 0$ there exists $m = |\Sigma|$ such that $GapUGC_{(1-\delta,\delta)}(\Sigma)$ is NP-hard.

We give a 2-query PCP system for this problem.

We need the long code. It is $E : [m] \to \{f : \{-1,1\}^m \to \{-1,1\}\} \cong B^{B^m}$, mapping $i \mapsto e_i = \mathsf{Dict}_i$ where $e_i(x) = x_i$.

Proofs only has -1 and 1. Can't give labels, so encode labels using encoding to turn into bits. Use long code to turn labels into bits.

Find a random edge, check to see if the permutation is satisfied. However, only bits can be checked!

We give the dictatorship test. Suppose we want to know whether the function is (close to) a long code or not. Given $f : \{-1,1\}^m \to \{-1,1\}$ we want to check if f is a dictator using 2 queries.

The simplest attempt is to take $x \sim \{-1,1\}^n$. Check if $f(x) \neq f(-x)$. This is bad because all odd functions satisfy this.

Attempt 2: Instead, we randomly flip some bits. $x \sim \{-1,1\}^n$. Take $y \sim_\rho x$ where

$$y_i = \begin{cases} x_i, & \text{with probability } \frac{1+\rho}{2} \\ -x_i, & \text{with probability } \frac{1-\rho}{2}. \end{cases}$$

Check if $f(x) \neq f(y)$. For the dictator, we get

1. $C = \frac{1-\rho}{2}$, the probability a single fixed bit is flipped.

2. Soundness:

$$\begin{aligned} s(f) &= \mathbb{P}(f \text{ is accepted}) \\ &= \mathbb{P}_{x \sim_\rho y}(f(x) \neq f(y)) \\ &= \frac{1}{2} - \frac{1}{2} \underbrace{\mathbb{E}_{x \sim_\rho y}[f(x)f(y)]}_{\text{Stab}_\rho(f)}. \end{aligned}$$

For example, consider $f = \mathsf{Maj}(x)$.

$$\mathbb{P}(\mathsf{Maj} \text{ is accepted}) = \frac{1}{2} - \frac{1}{2}\operatorname{Stab}_\rho(\mathsf{Maj}_m) \to \frac{\cos^{-1}\rho}{\pi}$$

as $m \to \infty$. Proof: Use the central limit theorem to get $\mathsf{Maj}\left(\frac{\sum x_i}{\sqrt{m}}\right)$ approaches a Gaussian.

Theorem 6.2 (Majority is stablest, MOO05)**:** Let $-1 < \rho < 0, \varepsilon > 0$. There exists τ, C such that if f passes the test with probability $> \frac{\cos^{-1}\rho}{\pi} + \varepsilon$, then there exists $i \in [m]$ such that $\operatorname{Inf}_i^{\le c}(f) \ge \tau$.

Definition 6.3: Define $x^{\oplus i}$ is x with the ith coordinate flipped. Note the Fourier expansion is $f(x) = \sum_{S\subseteq[m]} \widehat{f}(S)\chi_S(x)$, $\widehat{f}(S) = \mathbb{E}_x f(x)\chi_S(x)$, $\chi_S(x) = \prod_{i\in S} x_i$. Note $\sum_{S\subseteq[m]}\widehat{f}(S)^2 = \mathbb{E}[f(x)^2] = 1$.

$$\begin{aligned}\operatorname{Inf}_i(f) &= \mathbb{P}_{x\sim\{-1,1\}^n}[f(x) \ne f(x^{\oplus i})]\\ &= \sum_{S\ni i}\widehat{f}(S)^2.\end{aligned}$$

For example $f(x) = x_i$ has $\operatorname{Inf}_i(e_i) = 1$. We also have $\operatorname{Inf}_i(f) = 1$ for all i, which we want to exclude. Thus we look at

$$\operatorname{Inf}_i^{\le c}(f) = \sum_{S\ni s, |S|\le c}\widehat{f}(S)^2.$$

Now we come to the 2-query PCP for $\text{GapULC}(\Sigma)_{(1-\delta,\delta)}$. Given a graph, the verifier asks the prover to encode the optimal labelling using the long code. How do we check it satisfies $1-\delta$ fraction? This is inspired by the dictatorship test. We can only test whether one of the symbols is a dictator or not. I give you a sequence of functions, how can you test? The test combines!

Attempt 1:

- Pick a random $(v, w) \sim E$. Let f_v, f_w be the supposed long codes of $\sigma(v)$ and $\sigma(w)$.
- Let $\pi = \pi_{vw}$.
- Choose $x \sim_\rho y$ (ρ-correlated). (We're trying to combine the 2 test.)
- Accept if $f_v(x) = f_w(y\circ\pi)$ where $(y\circ\pi)_i = y_{\pi(i)}$. Here $y \in \{-1,1\}^m$ and $\pi : [m] \to [m]$.

This doesn't work because the prover can set $f_v = 1$ for all $v \in V$ and $f_w = -1$ for all $w \in W$ and by bipartiteness the verifier always accepts.

$x_{\sigma(v)} = e_{\sigma(v)}(x) \ne e_{\sigma(w)}(y\circ\pi) = e_{\pi(\sigma(w))}(y) = y_{\sigma(v)}$.

We modify the test.

- Pick $v \sim V$ and $w, w' \sim N(v)$. (We assume left-regularity.)
- Let $\pi = \pi_{vw}, \pi' = \pi_{vw'}$.
- Let $x \sim_\rho y$.

- Accept if $f_w(x \circ \pi) \neq f_{w'}(y \circ \pi')$.

Completeness: An optimal labeling will satisfy $\geq 1-\delta$ of the edges. The distribution of $(v,w),(v,w')$ are marginally uniform.

$$\mathbb{P}(\geq 1 \text{ of } (v,w),(v,w')) \leq \mathbb{P}((v,w) \text{ not}) + \mathbb{P}((v,w') \text{ not}) \leq 2\delta.$$

Write $f_w = e_{\sigma(w)}$, $f_{w'} = e_{\sigma(w')}$,

$$x_{\sigma(v)} = x_{\pi_{vw}(\sigma(w))} = e_{\sigma(w)}(x \circ \pi) \neq e_{\sigma(w')}(y \circ \pi') = y_{\pi_{vw'}(\sigma(w'))} = y_{\sigma(v)}$$

For every δ, some m for which NP-hard. Set δ to be small. Completeness is

$$C \geq (1-2\delta)\left(\frac{1-\delta}{2}\right) \geq \frac{1-\rho}{2} - \varepsilon,$$

$\delta \leq \frac{\varepsilon}{2}$.

Soundness: Suppose $\mathbb{P}(V \text{ accepts}) \geq C + \varepsilon$. We want to show there exists a labeling which satisfies $> \delta$ fraction of edges.

This is the main part of the proof.

For at least $\frac{\varepsilon}{2}$ fraction of $v \in V$, the test conditioned on selecting v accepts with probability $> \left(\frac{\cos^{-1}\rho}{\pi}\right) + \frac{\varepsilon}{2}$.

Otherwise by averaging argument

$$\mathbb{P}(\text{accept}) \leq \frac{\varepsilon}{2}I + (1-\frac{\varepsilon}{2})(\frac{\cos^{-1}}{\rho} + \frac{\varepsilon}{2}) < \cos^{-1}\frac{\rho}{\pi} + \varepsilon.$$

Call such v's good. Then

$$\mathbb{P}\left(\text{accept}\right) = \mathbb{E}_{w,w'}\left[\mathbb{E}_{x\sim_\rho y}\left[\frac{1}{2} - \frac{1}{2}f_w(x\circ\pi)f_{w'}(y\circ\pi')\right]\right].$$

Let

$$g_v(z) = \mathbb{E}_{w\sim N(v)}[f_w(z \circ \pi_{vw})].$$

If this is a proper labeling, $e_{\sigma(w)}(z \circ \pi_{vw}) = z_{\pi_{vw'}(\sigma(w))} = z_{\sigma(v)}$. We'll extract the most influential coordinates.

The expectation goes inside. Each expectation you can replace with g.

$$\begin{aligned}\mathbb{P}\left(\text{accept}\right) &= \mathbb{E}_{x\sim_\rho y}[\frac{1}{2} - \frac{1}{2}g_v(x)g_w(x)] \\ &= \frac{1}{2} - \frac{1}{2}\operatorname{Stab}_\rho(g_v) > \cos^{-1}\left(\frac{\rho}{\pi}\right) + \frac{\varepsilon}{2}.\end{aligned}$$

Now we use Majority is Stablest. There exists $j \in [m]$ such that

$$\operatorname{Inf}_j^{\leq C}(g_v) \geq \tau.$$

Assign $\sigma^*(v) = j_v$.

Assign a label for w independent of v. The label of v depend on its neighbors, the label for w depends only on its long code.

We show that the label for w should also have high influence for f_w. Expand

$$\begin{aligned} f_w \circ \pi_{vw} &= \sum_{S\subseteq[m]} \widehat{f}_w(S)\chi_S(x\circ\pi_{vw}) \\ &= \sum_T \widehat{f_w}(\pi^{-1}(T))\chi_T \\ g_v &= \mathbb{E}_{w\sim N(v)}[f_w(x\circ\pi_{vw})] \\ &= \sum_T \mathbb{E}_v \widehat{f}_w(\pi^{-1}(T))\chi_T. \end{aligned}$$

Influence high says that influence of neighbors f_w for these neighbors high for some coordinates. By Jensen,

$$\begin{aligned} \tau &\le \mathrm{Inf}^{\le C}_{j_v}(g_v) \\ &= \sum_{|S|\le C, S\ni j_v} \widehat{g}_v(S)^2 \\ &= \sum_{|S|\le C, S\ni j_v} \left(\mathbb{E}_w\left[\widehat{f_w}(\pi^{-1}(S))\right]^2\right) \\ &\le \sum_{|S|\le C, S\ni j_v} \mathbb{E}_w\left[\widehat{f_w}(\pi^{-1}(S))^2\right] \\ &= \mathbb{E}_w\left[\sum_{|S|\le C, S\ni j_v} \widehat{f_w}(\pi^{-1}(S)^2)\right] \\ &= \mathbb{E}_w\left[\mathrm{Inf}^{\le C}_{\pi^{-1}_{vw}(i_v)}(f_w)\right]. \end{aligned}$$

By another averaging argument, for at least $\frac{\tau}{2}$ fraction of v's neighborhood, $\mathrm{Inf}^{\le C}_{\pi_{vw}(j_v)}(f_w) \ge \frac{\tau}{2}$.

Look at all coordinates with high influence and choose one with high influence randomly: Let

$$S_w = \left\{k : \mathrm{Inf}^{\le C}_k(f_w) \ge \frac{\tau}{2}\right\}.$$

Assigning $\sigma^*(w) \sim_R S_w$, $|S_w| \le \frac{C}{\left(\frac{\tau}{2}\right)} = \frac{2C}{\tau}$.

Proof. We have

$$\sum_{i=1}^m \mathrm{Inf}^{\le C}_i(f_w) = \sum_{|S|\le C} |S|\widehat{f_w}(S)^2.$$

($\mathrm{Inf}^{\le C}_i(f_w) = \sum_{S\ni i, |S|\le C} \widehat{f_w}(S)^2$. The number of $S \ni i$ is $|S|$.) Choose a random $(v,w) \sim E$. What is $\mathbb{P}(\sigma^*(v) = \pi_{vw}(\sigma^*(w)))$? It's

$$\begin{aligned} \mathbb{P}(\sigma^*(v) = \pi_{vw}(\sigma^*(w))) &= \mathbb{P}(v \text{ good})\,\mathbb{P}(w \text{ good for } v\text{—}v \text{ is good})\,\mathbb{P}(\sigma^*(v) = \pi_v w(\sigma^*(w))\text{—}...) \\ &\ge \frac{\varepsilon}{2}\frac{\tau}{2}\frac{1}{\left(\frac{2C}{\tau}\right)} \\ &= \frac{\varepsilon\tau^2}{8C} = \delta. \end{aligned}$$

□

m depends on δ depends on τ, C depends on ε, ρ. We reduced $\mathsf{Gap} - \mathsf{ULC}_{(1-\delta,\delta)}(\varepsilon)$ to $\mathsf{Gap} - \mathsf{MaxCut}_{(\frac{1-\rho}{2}-\varepsilon, \frac{\cos^{-1}\rho}{\pi}+\varepsilon)}$.

In MaxCSP, maximize the constraints satisfied. $C_i(x|_S)$, $|S| \le q$ constant arity. Maximize number of constraints satisfied. There is an SDP that achieves best approximation ratio. Generalizes GW.

Charikar: given ULC with $\ge 1-\delta$ satisfiable, there is an algorithm to satisfy $\frac{1}{|\varepsilon|^{\frac{\delta}{2}+O(S^2)}}$ fraction of constraints.

UGC captures limitations of SDP methods. Assuming UGC is like saying SDP is the best.

There is a reason to disbelieve UGC if you think for some problem there can be something better than SDP.

Matching, determinants: require exponential size SDP.

Lower bounds: cannot compute parity using SDP?

7 Hastad's 3-bit PCP

This was proved by Hastad in 1997. It's based on hardness of label cover.

7.1 Hardness of label cover

Definition 7.1: An instance I of label cover consists of

1. labels L, R,
2. $G = (U, V, E)$,
3. for each edge E in G, a function. $\Pi = \{\pi_e : L \to R : e \in E\}$. (In unique label cover the functions have to be permutations, but not here.)

A labeling is a pair of functions $A : U \to L, B : V \to R$. An edge (u, v) is satisfied iff $\pi_{(u,v)}(A(u)) = B(v)$.

Define the value $\mathrm{val}(I)$ to be the maximum possible fraction of satisfied edges.

Theorem 7.2 (Hardness of label cover)**:** For all $\delta \in (0, 1)$, there exist L, R such that $|L|, |R| \le \exp\left(\frac{1}{\delta}\right)$ such that $\mathsf{Gap} - \mathsf{LC}(L, R)_{(1,\delta)}$ is NP-hard, where

- Yes instances are I with $\mathrm{val}(I) = 1$,
- No instances are I with $\mathrm{val}(I) \le \varepsilon$.

The proof is by Ran Raz in 1995, by the Parallel Repetition Theorem.

Theorem 7.3 (3 bit PCP)**:** For all $\varepsilon, \delta \in (0, 1)$, there exists a PCP for Gap-LC$_{(1,\delta)}$ over the boolean alphabet satisfying the following.

1. The verifier queries 3 bits and checks $x_{i_1} \oplus x_{i_2} \oplus x_{i_3} = b$.
2. Completeness is $1 - \varepsilon$. (If it is a yes instance, there exists a proof that is accepted with probability $\geq 1 - \varepsilon$.)
3. Soundness is $\frac{1}{2} + \delta$. (If it is a no instance, all proofs are accepted with probability $\leq \frac{1}{2} + \delta$.)

Corollary 7.4 (Hardness of MAX3LIN2)**:** It is NP-hard to distinguish: given a system of linear equations over $\mathbb{F}_2$ with 3 variables, are

- $\geq 1 - \varepsilon$ satisfiable,
- $\leq \frac{1}{2} + \varepsilon$ satisfiable.

This is optimal because a random assignment gives a 2-approximation.

Corollary 7.5 (Hardness of MAX3SAT)**:** For all $\varepsilon, \delta \in (0, 1)$, it is NP-hard to distinguish a $(1 - \varepsilon)$-satisfiable instance and $(\frac{7}{8} + \delta)$-satisfiable instance.

A random assignment will give a $\frac{7}{8}$ approximation.

If we can solve MAX3SAT with better than $\frac{7}{8}$ approximation, we can get better than 2-approximation for MAX3LIN. This gives a reduction from MAX3SAT to MAX3LIN2. Encode $x_{i_1} \oplus x_{i_2} \oplus x_{i_3} = 1$ with

$$\begin{gathered} x_{i_1} \vee x_{i_2} \vee x_{i_3} \\ x_{i_1} \vee \overline{x_{i_2}} \vee \overline{x_{i_3}} \\ \overline{x_{i_1}} \vee \overline{x_{i_2}} \vee x_{i_3} \\ \overline{x_{i_1}} \vee x_{i_2} \vee \overline{x_{i_3}} \end{gathered}$$

We use the long code,

$$\operatorname{long}_L(x_1, \ldots, x_L) = \chi_c.$$

7.2 Hastad's 3 bit PCP

Hastad's 3-bit PCP is as follows. The input is a LC instance I. The proof is

$$\begin{aligned} f_u &: \{-1, 1\}^L \to \{1, -1\}, \forall u \in U \\ f_v &: \{-1, 1\}^R \to \{1, -1\}, \forall v \in V \end{aligned}$$

The verifier does the following.

1. Pick $(u, v) \sim_R E$.
2. Pick $x \sim_R \{-1, 1\}^R$, $y \sim_R \{1, -1\}^L$.

3. Pick $\mu \sim \{1,-1\}^L$ such that
$$\mu_i = \begin{cases} 1, & \text{w. p. } 1-\varepsilon \\ -1, & \text{w. p. } \varepsilon. \end{cases}$$
4. For all $i \in L$, take $z_i = X_{\pi_{(u,v)}(i)} y_i \mu_i$.
5. Accept if $f_u(z) = f_v(x) f_u(y)$.

Completeness. Let A, B be labelings satisfying all edges. Let $f_u(x_1, \dots, x_L) = X_{A(u)}$ and $f_v(x_1, \dots, x_R) = X_{B(V)}$. Then

$$\begin{aligned} \mathbb{P}\,(\text{accept}) &= \mathbb{P}(f_u(z) = f_v(x) f_u(y)) \\ &= \mathbb{P}[Z_{A(u)} = X_{B(V)} Y_{A(U)} \\ &= \mathbb{P}(X_{\pi_{(u,v)}(A(u))} Y_{A(u)} = X_{B(v)} Y_{A(u)}) \\ &= \mathbb{P}(\mu_{A(u)} = 1) = 1 - \varepsilon. \end{aligned}$$

□

Problem: if you set the bits all to 1, the verifier will always accept the proof.

Soundness. The valid long code should satisfy the following condition: $f_u(-x) = -f_u(x)$ (we say it is folded).

We don't ask the prover to give all the information; we just ask the prover to give half the information. For all f_u, and a pair $(x, -x)$, only one of $f_u(x), f_u(-x)$ is given in the proof.

Claim: if a function $f : \{-1,1\}^L \to \{-1,1\}$ is folded, then $\widehat{f}(\phi) = 0$.

Proof: $\widehat{f}(\phi) = \mathbb{E}_x[f(x)] = 0$.

We have to show that if $\mathbb{P}\,(\text{accept}) \geq \frac{1}{2} + \delta$ then there exists a labelling (A, B) that satisfies $\geq \delta'$ fraction of edge constraints.

Notation: Given $S \subseteq L$, $\pi : L \to R$,

$$\begin{aligned} \pi(S) &= \{j \in R : \exists L \in S, \pi(L) = j\} \\ \pi_2(S) &= \{j \in R : \text{odd number of } L\text{'s in } S,\ \pi(L) = j\} \end{aligned}$$

Claim: For all $x \in \{-1,1\}^R$, $\chi_S(x \circ \pi) = \chi_{\pi_2(S)}(x)$. $((x \circ \pi)_i = x_{\pi(i)})$

Proof: $\chi_S(x \circ \pi) = \prod_{i \in S} x_{\pi(i)} = \prod_{j \in \pi(S)} x_j$. We have

$$\begin{aligned} \frac{1}{2} + \delta \leq \mathbb{P}\,(\text{accept}) &= \mathbb{E}_{u,v,x,y,\mu}\left[\frac{1 + f_u(x) f_v(x) f_u(y)}{2}\right] \\ \implies \mathbb{E}[f_u(z) f_v(x) f_u(y)] &\geq 2\delta. \end{aligned}$$

This implies at for $\geq \delta$ fraction of edges (u, v), the value $\mathbb{E} f_u(z) f_v(x) f_u(y) \geq \delta$.

To prove soundness, set $f_u = f, f_v = g$ for notational convenience. Then

$$\begin{aligned}
\mathbb{E}[f(z)g(x)f(y)] &= \sum_{S,T,U} \widehat{f}_S \widehat{g}_T \widehat{f}_U \mathbb{E}_{x,y,u}[\chi_S(x)\chi_T(x)\chi_U(y)] && z = (\chi \circ \pi)y\mu \\
&= \sum_{S,T,U} \widehat{f}_S \widehat{g}_T \widehat{f}_U \mathbb{E}_x[\chi_S(x \circ \pi)\chi_T(x)]\mathbb{E}_y[\chi_U(y)\chi_S(y)]\mathbb{E}_\mu[\chi_S(\mu)] \\
&= \sum_S \widehat{f}_S^2 \widehat{g}_{\pi_2(S)}(1-2\varepsilon)^{|S|}.
\end{aligned}$$

This is $\geq \delta$ for δ fraction of edges.

Decoding a labeling is done as follows.

1. For all u, pick S with probability $\widehat{f}_u^2(S)$.
2. Set $A(u) \sim_R S$.

For all v,

1. pick T with probability $\widehat{f}_v^2(T)$.
2. Sample $B(v) \sim_R T$.

Why does this labeling satisfy some fraction of edges?

Fix (u, v).

$$\begin{aligned}
\mathbb{P}[(u,v) \text{ is satisfied}] &= \mathbb{P}[\pi_{(u,v)}(A(u)) = B(v)] \\
&\geq \sum_S \sum_{T \subseteq \pi(S)} \widehat{f}_u^2(S)\widehat{f}_v^2(T)\frac{1}{|S|}.
\end{aligned}$$

What is the probability that a random edge is satisfied?

$$\begin{aligned}
\mathbb{P}_{(u,v)\in E}[(u,v) \text{ is satisfied}] &\geq \sum_S \widehat{f}_u^2(S)\widehat{f}_v^2(\pi_v(S))\frac{1}{|S|} \\
&= \sum_S (\widehat{f}_u(S)\widehat{f}_2(S)\frac{1}{\sqrt{|S|}})^2 \sum_S \widehat{f}_u^2(S) \\
&\overset{\text{CS}}{\leq} \left(\sum_S \widehat{f}_u^2(S)\widehat{f}_v(\pi_2(S))\frac{1}{\sqrt{|S|}}\right)^2 \\
&\geq 4\varepsilon(\sum_S \widehat{f}_u^2(S)\widehat{f}_v(\pi_2(S))(1-2\varepsilon)^{|S|})^2 \\
&\geq \delta \cdot 4\varepsilon, \delta^2 = 4\varepsilon\delta^2 = \delta'
\end{aligned}$$

Use $\frac{1}{\sqrt{x}} \geq \sqrt{4\varepsilon}(1-2\varepsilon)^x$. (Take derivatives to see this.) □

Easier than PCP we saw last timel. Do

8 ZKP for NP

What is a zero-knowledge proof? IP plus some notion of zero knowledge.

Recap: In IP

- Prover (Merlin), computationally unbounded
- Verifier (Arthur), computationally bounded, probabilistic

Prover wants to convince verifier that $x \in L$. This procedure should be

- Complete: if $x \in L$, V should accept (with perfect completeness—can always make one-sided (?))
- Sound: if $x \notin L$, V should reject with high probabilistic

What is zero knowledge? Prover should convince $x \in L$ without giving any other info. The verifier can simulate the entire procedure, in the following sense. There exits an efficient algorithm S such that

$$\mathbb{P}(S \text{ outputs } \pi) = \mathbb{P}((P, V) \text{ outputs } \pi).$$

This is called Perfect Zero Knowledge. Statistically Zero Knowledge is when the distributions are $\approx$: superpoly samples are required. Computational Zero Knowledge means for all $x \in L$, no polytime algorithm can distinguish between output of S and (P, V). This is the version used in crypto.

We give some canonical examples.

1. Graph nonisomorphism (PZK):

   Input: G_1, G_2. Prover wants to convince verifier $G_1 \ncong G_2$.

   Verifier chooses random permutation H of either G_1 or G_2 and sends it to the prover.

   Prover says whether $H \cong G_1$ or $H \cong G_2$.

   If $G_1 \cong G_2$, prover can't be right with probability $> \frac{1}{2}$.

   Here is the simulator: picks G_1, G_2, chooses random isomorphism, and outputs index chosen: $(\pi(G_b), b)$.

2. Graph isomorphism (PZK): Input G_1, G_2.

   Prover sends V a random permutation H of G.

   Verifier says 1 or 2.

   Prover sends permutation π such that $\pi(H) = G_b$.

   Simulator: You might think this is impossible, but the simulator doesn't need to solve graph isomorphism. With probability $\frac{1}{2}$ output $(\pi(G_1), 1, \pi^{-1})$. With probability $\frac{1}{2}$ output $(\pi(G_2), 2, \pi^{-1})$.

We show 3-coloring has a zero-knowledge proof, conditioned on the existence of 1-way functions. (Actually, you need bit commitment. Encrypt a bit and send it to the verifier and then reveal later.)

Let G be the input graph with a 3-coloring C.

P randomly permutes colors in C and then commits. Each vertex is colored.

Verifier chooses a random edge. The verifier reveals the colors on that edge. Accept if the vertices have different colors.

$\frac{1}{n^2}$ probability of failure.

If $|G| = n$, then repeat n^3 times. Issues in parallel?

Question: Can you do in parallel? (Concurrent zero-knowledge.)

This is not perfect zero-knowledge because of bit-commitment. If it is perfectly binding, the verifier could figure all the colors from the commitment. for a computationally bounded verifier.

OWF $\implies$ PRG $\implies$ bit commitments. (OWP: h hardcore predicate. $f(x), h(x \oplus b)$. Given $f(x)$ hard to predict $h(x)$.)

If has a PZK proof, what happens? If any NP-complete problem has PZK proof, the poly hierarchy collapses to the 2nd level.

Simulator outputs random edge. Permutation ensures random colors. $e, (u, v)$, encrypts those. For the rest, assign random bit commitments. It doesn't have to start at the beginning, it just has to output a plausible transcript. It's not perfect; if you had enough time you can check the bit commitments are bogus.