Some Successes and Failures of Algebra in Constructing Extractors

David Zuckerman
University of Texas at Austin

Randomization extremely useful

- Algorithms (e.g. approximation algorithms, factoring polynomials, min cut, load balancing)
- Monte Carlo simulations
- Network constructions
- Cryptography
- Distributed computing

Problem with using randomization

Infeasible to get many truly random bits.

Two general approaches

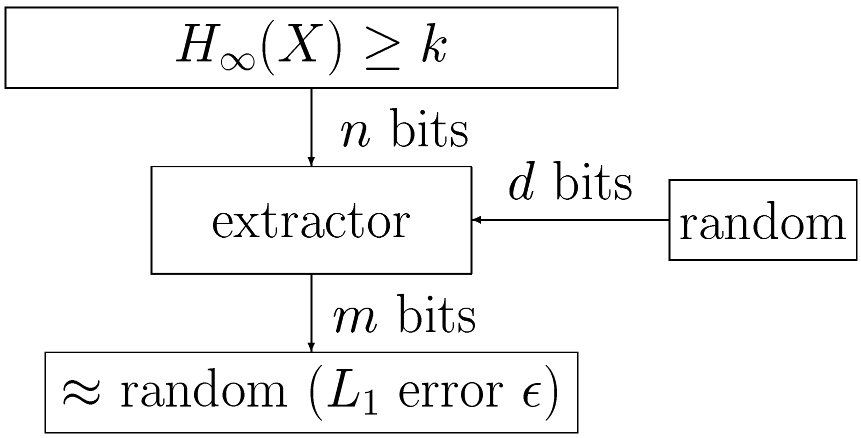
1. Minimize number of random bits used.
2. Minimize quality of randomness required.

Min-Entropy

Min-entropy $H_\infty(X) = \min_x\{-\log_2 \Pr[X = x]\}$.

Therefore, $H_\infty(X) \geq k \iff (\forall x) \Pr[X = x] \leq 2^{-k}$.

Extractor [Nisan-Z]

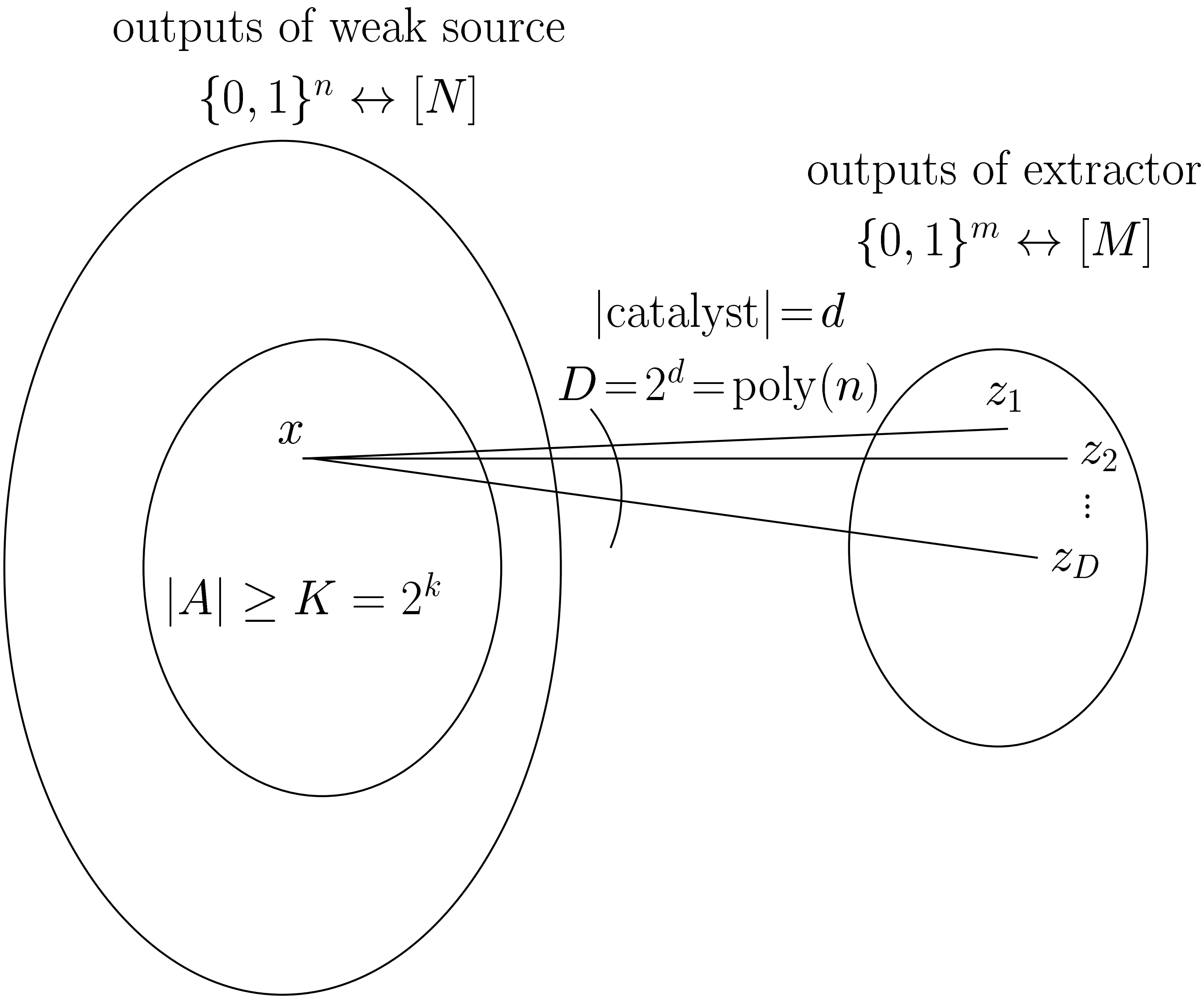


Extractor Parameters

error ϵ, $\alpha > 0$ arbitrary positive constants

due to	d random bits	output length m
non-explicit	$\log n + O(1)$	k
[LRVW] +[WZ]	$O(\log n)$ $O(\log^2 n)$	$(1-\alpha)k$ k
[TZS]	$\log n + O(\log(\log^* n))$	$k/(\sqrt{n}\log n)$
[SU]	$(1+\alpha)\log n$	$k^{\Omega(\alpha)}$

Graph-Theoretic View of Extractor

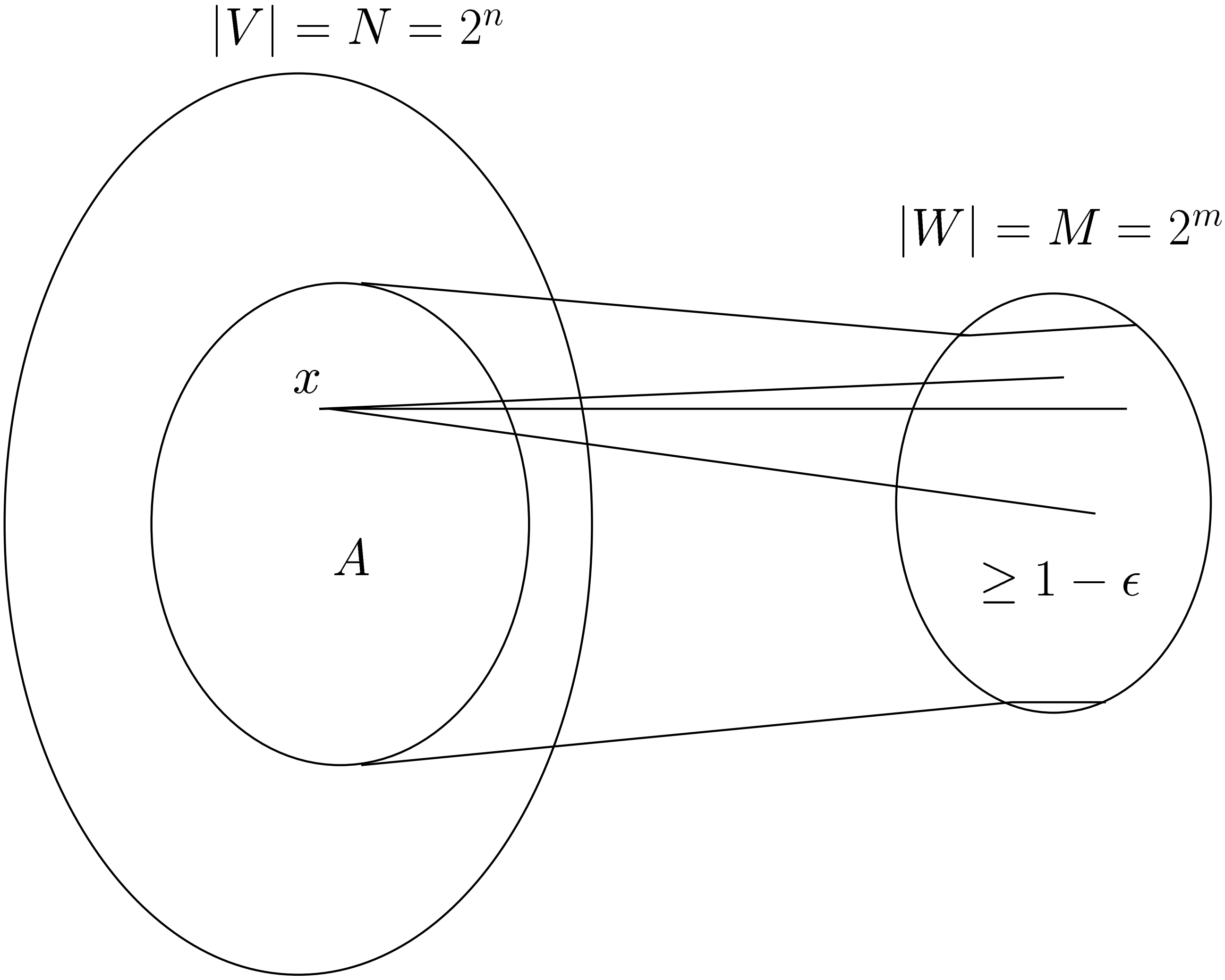


$$\Gamma(x) = \{E(x, y_1), \ldots, E(x, y_D)\} = \{z_1, \ldots, z_D\}$$

Extractor property:

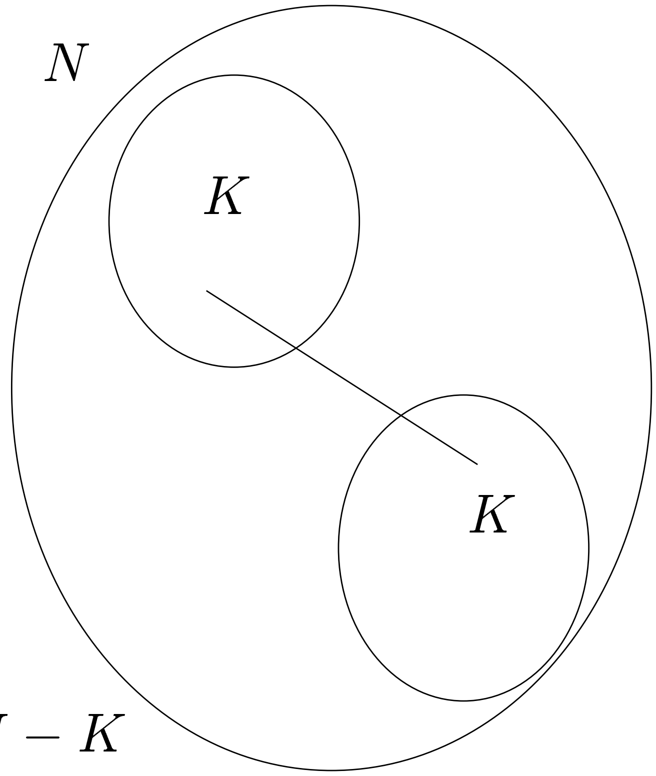
$x \in_R A, z \in_R \Gamma(x) \implies z$ ϵ-uniform

Extractor $\Longrightarrow$ Disperser



$$|A| \geq K \implies |\Gamma(A)| \geq 1 - \epsilon$$

Highly Expanding Graphs



Think of $K = \sqrt{N}$.

$|A| \geq K \implies |\Gamma(A)| > N - K$

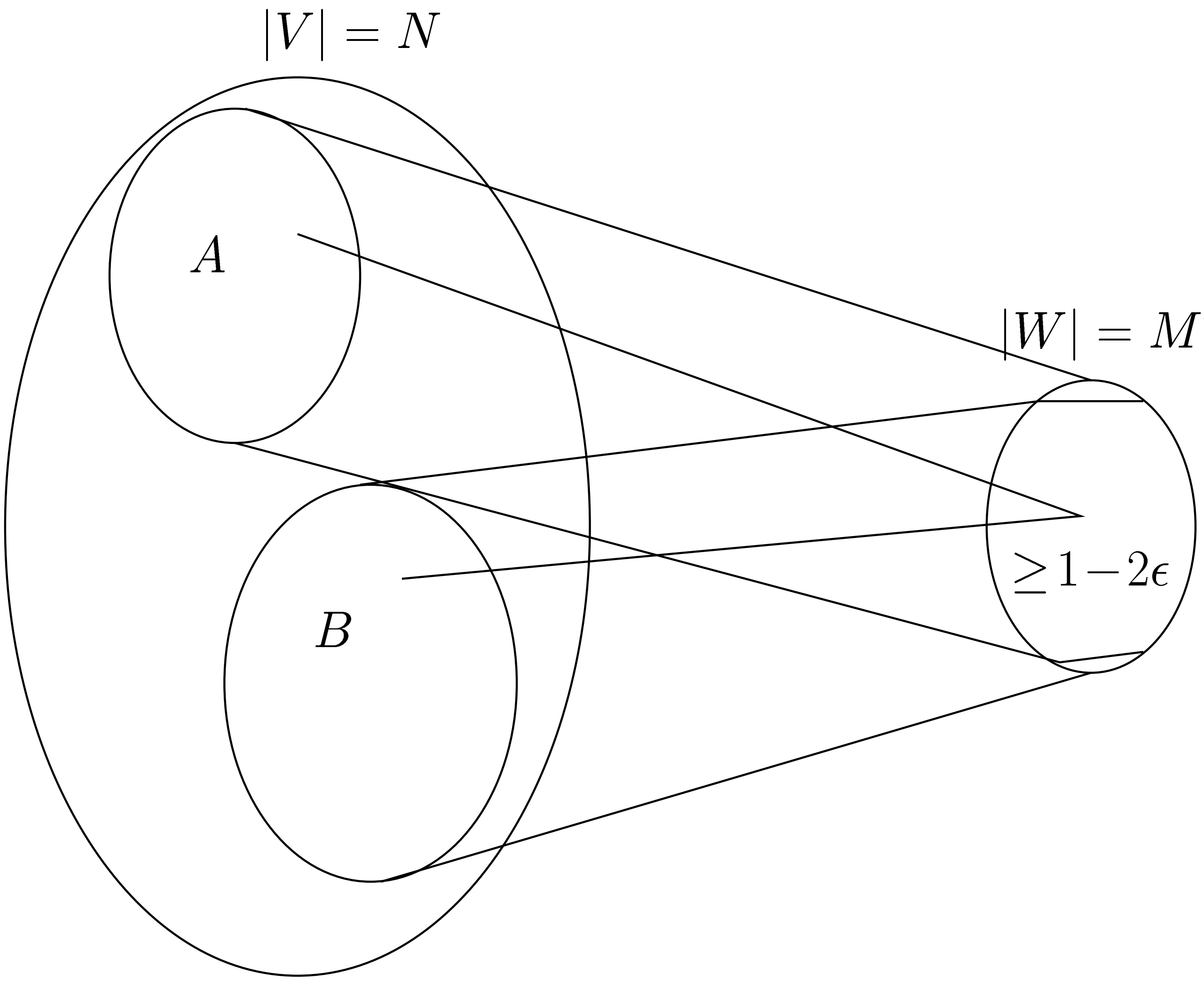
$D_{\text{avg}} > (N-K)/K \approx N/K$.

Second eigenvalue method: $D_{\text{avg}} \approx (N/K)^2$.
Tight e.g. for projective plane graphs.

Good extractors yield $D_{\text{avg}} = N^{1+o(1)}/K$ [WZ].

Disperser $\implies$ K-expanding graphs

[Wigderson-Z]



$|A|, |B| \geq K \implies A$, B have common neighbor.

$G =$ paths of length 2 on vertex set V.

Need to remove high degree vertices on right.

Applications of Extractors

Simulating randomized algorithms using weak random sources [Z]

Expanders that beat the eigenvalue bound [WZ]

Superconcentrators and non-blocking networks [WZ]

Pseudorandom generators for space-bounded computation [NZ]

Random sampling using few random bits [Z]

Sorting and selecting in rounds [Pip]

Time-Space tradeoffs [Sip]

Unapproximability of clique [Z] and VC dimension [MU]

Error-correcting codes with good list decoding properties [TZ]

Cryptography [L,V]

Extractor Constructions

Based on hashing [Z,NZ] [WZ,SZ,SSZ,Z,Ta-S,RSW,LRVW]

Based on pseudorandom generators [Trevisan] [RRV,ISW,TUZ]

Extractors from Polynomials [Ta-Shma-Z-Safra][SU]

Work over $\mathbb{F} = \mathbb{F}_q$, q prime.

First attempt:

$$\mathrm{Ext}(\overbrace{\text{poly } p,\, \deg(p) < h}^{\text{weak source}};\ \overbrace{a \in \mathbb{F}_q}^{\text{uniform}}) = p(a)$$

Fails:

- Squares of polys mapped to squares
- Output length $m = d$.

Second attempt

$$\mathrm{Ext}(\overbrace{\text{poly } p, \deg(p) < h}^{\text{weak source}}; \overbrace{a \in \mathbb{F}_q, r \in \{0,1\}^{\lceil \log q \rceil}}^{\text{uniform}})$$

$$= \langle p(a) \cdot r, p(a,b) \cdot r, \ldots, p(a+m-1) \cdot r \rangle$$

Tool in proof attempt: Yao's lemma

If $Z = Z_1 Z_2 \ldots Z_m \in \{0,1\}^m$ is ϵ-far from uniform, then

$$\exists i, \text{predictor } P : \Pr[P(Z_1, \ldots, Z_i) = Z_{i+1}] \geq \frac{1}{2} + \frac{\epsilon}{m}$$

Proof attempt idea

Suppose Ext not extractor. Then $\exists i, P$.
Suppose P predicts perfectly.
Knowledge of $p(a), p(a+1), \ldots, p(a+i-1)$
$\implies p(a+i) \implies p(a+i+1) \ldots$
Therefore, only $q^i < q^m$ possibilities for p.
Valid extractor if $K \geq q^m$ ($k \geq m \log q$), since $K = \#$ possibilities for p.

Problem: P not perfect.

Extractors from Bivariate Polynomials [TZS]

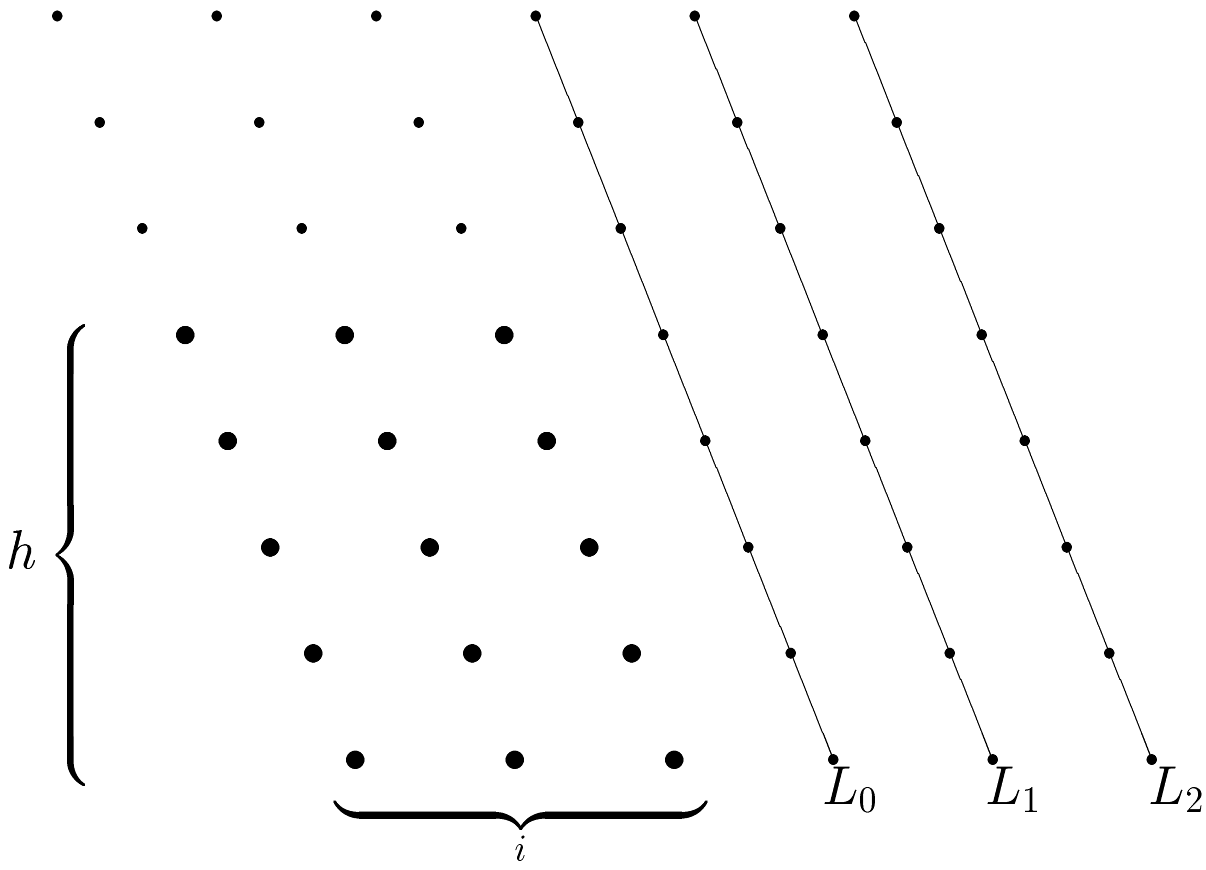
$$\mathrm{Ext}(\overbrace{\text{poly } p,\ \deg(p) < h}^{\text{weak source}};\ \overbrace{(a,b) \in \mathbb{Z}_q^2, r \in \{0,1\}^{\log q}}^{\text{truly random}})$$

$$= (p(a,b)\cdot r, p(a+1,b)\cdot r, \ldots, p(a+m-1,b)\cdot r)$$

Why bivariate polynomials help

- Predict successive lines.
- Many weak predictions for each point on line $\implies$ strong predictor for each line.
  - If Pr[predict point] $\geq 2/3$, then uniquely decode Reed-Solomon codes.
  - In fact, prediction probability smaller, so list decode RS codes.

Proof Outline



1. Given value of polynomial p on $hi \leq h(m-1)$ points of random "parallelogram."
2. Deduce value of p on qi points.
3. Yao's lemma + list decoding Hadamard code $\implies$ small number of guesses for $p(x), x \in L_0$.
4. List decoding Reed-Solomon code $\implies$ only ℓ possibilities for $p|_{L_0}$.

Proof Outline (continued)

1. Given value of polynomial p on $hi \leq h(m-1)$ points of random "parallelogram."
2. Deduce value of p on qi points.
3. Yao's lemma + list decoding Hadamard code $\implies$ small number of guesses for $p(x), x \in L_0$.
4. List decoding Reed-Solomon code $\implies$ only ℓ possibilities for $p|_{L_0}$.
5. Query which polynomial.
6. Successively learn p on $L_0, L_1, \ldots, L_{h-i}$
7. Deduce p.

Assuming Ext not an extractor, # possibilities for p is $\leq q^{hi}\ell^{h-i} \leq q^{2hm}$.

Therefore, Ext is extractor when $K > q^{2hm} \iff k > 2hm \log q$.

Parameters: $q = n^{\Theta(1)}, h = \Theta(\sqrt{n/\log n})$, so need $k > m\sqrt{n} \log n$.

List Decoding

Output list of close codewords.

List decoding possible for $> \frac{1}{2}$ minimum distance.

Combinatorial: few close codewords.
Algorithmic: find them.

Reed-Solomon codes:

- Algorithmic decoding within distance $n - \sqrt{kn}$ [Guruswami-Sudan, Sudan].

Lemma (Sudan): Given m distinct $(x_i, y_i) \in \mathbb{F}^2$, there are $< 2m/a$ degree h polys p such that $p(x_i) = y_i$ for at least a values of $i \in [m]$, provided $a \geq \sqrt{2hm}$.

Improved Algebraic Extractors[Shaltiel-Umans]

Disadvantage of previous construction:
Lines instead of points $\implies$ many more query points $\implies$ large entropy loss: $m < k/\sqrt{n}$.
Tight: polynomials p depending only on y.

Fix: better successor function; work in dimension $d > 2$.

- View $\mathbb{F}^d$ as extension field $\mathbb{F}_{q^d}$ of $\mathbb{F}$.
- g a generator of $\mathbb{F}_{q^d}$.
- $g \leftrightarrow d \times d$ matrix A over $\mathbb{F}$.
- $\mathrm{successor}(v) = Av$.

$$\mathrm{Ext}(\overbrace{\text{poly } p, \deg(p) < h}^{\text{weak source}}; \overbrace{v \in \mathbb{F}_q^d, r \in \{0,1\}^{\lceil \log q \rceil}}^{\text{uniform}})$$

$$= \langle p(v) \cdot r, p(Av) \cdot r, \ldots, p(A^{m-1}v) \cdot r \rangle$$

Complications

1. Error not small enough with lines.
   - Use higher degree curves.
2. In [TZS], need additional information about p with each prediction step.
   - Use interleaved curves.

Choose random $t_1, \ldots, t_{2r} \in \mathbb{F}$; $y_1, \ldots, y_{2r} \in \mathbb{F}^d$
Use degree $2r-1$ polynomials:

$$\begin{array}{cc} i \le r & i > r \\ \\ f_1(t_i) = y_i & f_1(t_i) = y_i \\ f_2(t_i) = Ay_i & f_2(t_i) = y_i \\ f_3(t_i) = Ay_i & f_3(t_i) = Ay_i \\ f_4(t_i) = A^2y_i & f_4(t_i) = Ay_i \\ \vdots & \vdots \end{array}$$

Start with values of p on $f_1, f_2, \ldots, f_{2i}$.

Conclusions: Algebra and Extractors

Successes of algebra:

- Simplest known extractor constructions use algebra.
- Best constructions for achieving $d \approx 1 \cdot \log n$
  - Useful when dealing with degree $D = 2^d$.
- Optimal pseudorandom generators from hard functions [Umans].

Failures of algebra:

- Bounding second eigenvalue yields weak results.
- Current algebraic constructions have $m = o(k)$.
- Proofs not so simple.

Open Questions

- Better algebraic constructions?
- Simpler proofs?
- $\exists$ poly-time algorithm deciding if graph is extractor?
  - Exact is NP-hard, but approximate?
  - Approximate K-expanding property?