

Noisy Interpolating Sets for Low Degree Polynomials

Zeev Dvir

Princeton U.

Amir Shpilka

Technion

Interpolating sets

- $\mathbb{F}_d[x_1, \dots, x_m]$ = m-variate polynomials of **total degree** $\leq d$ over \mathbb{F}
- **In this talk** $d = \text{constant}$, $|\mathbb{F}|$ constant (**except in the examples...**)
- $S \subset \mathbb{F}^m$ is interpolating for $\mathbb{F}_d[x_1, \dots, x_m]$ if the mapping $P(x_1, \dots, x_m) \rightarrow (P(\alpha))_{\alpha \in S}$ is 1-1
- I.e. any two polynomials in $\mathbb{F}_d[x_1, \dots, x_m]$ must differ on some point from S
- $\Leftrightarrow P \in \mathbb{F}_d[x_1, \dots, x_m]$ can be recovered from its set of values on S

Example

- Assume $\{0, 1, \dots, d\} \subset \mathbb{F}$
- $S = \{0, 1, \dots, d\}$ is interpolating for $\mathbb{F}_d[x]$
- Interpolation is easy:
 - $P(x) = \sum_{i=0}^d P(i) \cdot \prod_{j \neq i} (x-j)/(i-j)$
- **More generally:** $S = \{0, \dots, d\}^m$ is interpolating for n -variate polynomials with degree $\leq d$ in each variable
- **Recall:** we are interested in total degree d

Noisy interpolating sets

- S is a ε -noisy interpolating set for P if $P(x_1, \dots, x_m)$ can be recovered (efficiently) from its set of values on S even if an adversary corrupts ε -fraction of the values
- **In other words:** noisy interpolating sets allow (efficient) error correction
- **Note:** unlike noiseless case, no guarantee for an efficient interpolation algorithm
- **Goal:** construct ε -noisy interpolating sets, with efficient recovery, for $\mathbb{F}_d[x_1, \dots, x_m]$

Example

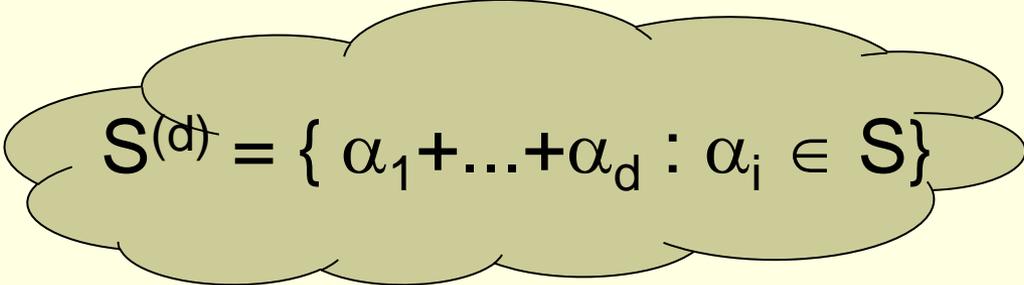
- $S = \{0, 1, \dots, n-1\}$ noisy interpolating set for $\mathbb{F}_d[x]$ for $\varepsilon = (n-d)/2n$
- **Proof:** minimal distance of degree d Reed-Solomon codes
- **Note:** efficient interpolation algorithm

Our results

- **Theorem:** Let S be ε -noisy interpolating set for degree 1 polynomials over \mathbb{F} . Then

$$S^{(d)} := S + S + \dots + S \text{ (d times)}$$

is $(\varepsilon/2)^d$ -noisy interpolating set for degree d polynomials (i.e. $\mathbb{F}_d[x_1, \dots, x_m]$)


$$S^{(d)} = \{ \alpha_1 + \dots + \alpha_d : \alpha_i \in S \}$$

Our results

- **Theorem:** Let S be ε -noisy interpolating set for degree 1 polynomials over \mathbb{F} . Then

$$S^{(d)} := S + S + \dots + S \text{ (d times)}$$

is $(\varepsilon/2)^d$ -noisy interpolating set for degree d polynomials (i.e. $\mathbb{F}_d[x_1, \dots, x_m]$)

- **Moreover:** if S has efficient recovery algorithm then so does $S^{(d)}$
- Works for any \mathbb{F}
- **Note:** $S^{(d)}$ may be a multiset
- **Theorem:** Can find S s.t. $\{S^{(d)}\}$ is a noisy interpolating set

Punctured Reed-Muller codes

- RM(\mathbb{F}, d, m) code is RM: $\mathbb{F}_d[x_1, \dots, x_m] \rightarrow \mathbb{F}^{|\mathbb{F}|^m}$
 - $P(x_1, \dots, x_m) \rightarrow \{P(\alpha)\}_{\alpha \in \mathbb{F}^m}$
- **Fact:** rate $\sim m^d/p^m$, distance $(1-1/|\mathbb{F}|)^d = \exp(-d)$
- **Question:** can we make $RM_{d,m}$ a good code (linear rate, constant relative distance)?
- **Corollary:** if $|S|=O(m)$ is ε -noisy interpolating set for degree 1 polynomials, then $RM_S: \mathbb{F}_d[x_1, \dots, x_m] \rightarrow \mathbb{F}^{|S^{(d)}|}$ is a good code
- **Proof:** $|S^{(d)}| = O_d(m^d) = O(\dim(\mathbb{F}_d[x_1, \dots, x_m]))$ Can correct $(\varepsilon/2)^d = \exp(-d)$ frac. of errors

PRGs for degree d polynomials

- **Def:** T is ε -pseudo-random for $\mathbb{F}_d[x_1, \dots, x_m]$ if for any $P(x_1, \dots, x_m)$ and $\alpha \in \mathbb{F}$

$$|\Pr_{x \in_R \mathbb{F}}[P(x) = \alpha] - \Pr_{x \in_R T}[P(x) = \alpha]| < \varepsilon$$

- **In particular** T is noisy interpolating set
- **However** no clear efficient recovery
- **Note:** Our result does not imply pseudo-randomness
- **Corollary:** our result+ [Viola`08] $S^{(d)}$ is pseudo-random and has efficient recovery

What's next

- Noisy interpolating sets for linear functions
 - Linear error-correcting codes
- Partial derivatives of polynomials
- Noisy interpolating sets for deg 2 polynomials

Noisy interpolating sets for linear functions (deg 1 polynomials)

- **Def:** $C: \mathbb{F}^m \rightarrow \mathbb{F}^n$ linear error correcting code of rate n and relative distance δ if
 - C is a linear mapping
 - $\forall v, u \in \mathbb{F}^m \quad d_H(C(v), C(u)) \geq \delta \cdot n$
- C can be represented by $n \times m$ matrix G
- Let $S = \{\text{rows of } G\}$
- **Observation:** S is $\delta/2$ -noisy interpolating set for degree 1 polynomials (linear functions)

Noisy interpolating sets for linear functions cont.

- $S = \text{rows of } G = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$
- **Def:** $\forall \mathbf{a} \in \mathbb{F}^m, L_{\mathbf{a}}(x_1, \dots, x_m) = a_1 x_1 + \dots + a_m x_m = \langle \mathbf{a}, \mathbf{x} \rangle$
- $G \cdot \mathbf{a} = (\langle \mathbf{s}_1, \mathbf{a} \rangle, \dots, \langle \mathbf{s}_n, \mathbf{a} \rangle) = (L_{\mathbf{a}}(\mathbf{s}_1), \dots, L_{\mathbf{a}}(\mathbf{s}_n))$
- I.e. encoding of \mathbf{a} = evaluation of $L_{\mathbf{a}}$ on S
- **Note:** minimal distance = $\delta n \Rightarrow$
can recover $L_{\mathbf{a}}$ from $< \delta n/2$ errors \Rightarrow
 S is $\delta/2$ -noisy interpolating set for degree 1 polynomials
- Efficient decoding algorithm \Leftrightarrow efficient noisy interpolating algorithm

Proof sketch of main theorem

- **Theorem:** Let S be ε -noisy interpolating set for degree 1 polynomials over \mathbb{F} . Then

$$S^{(d)} := S + S + \dots + S \text{ (d times)}$$

is $(\varepsilon/2)^d$ -noisy interpolating for degree d polynomials

- **Proof idea:** induction on d
- **Induction basis:** $d=1$ is the assumption
- **Induction step:** learn partial derivatives of P

Partial derivatives of polynomials

- $M = x^{d_1} y^{d_2} z^{d_3}$ ($d_i < |\mathbb{F}|$)
- $\partial M / \partial x = d_1 \cdot x^{d_1-1} y^{d_2} z^{d_3}$
- **Additivity**: $\forall \mathbf{a} \in \mathbb{F}^m, \partial_{\mathbf{a}} P(\mathbf{x}) = \sum_i a_i \cdot \partial P / \partial x_i$
- **Equivalently**: $\partial_{\mathbf{a}-\mathbf{b}} P(\mathbf{x}) = \sum_i (a_i - b_i) \cdot \partial P / \partial x_i$
- **Note**: $(x+a)^d - (x+b)^d = (a-b) \cdot d \cdot x^{d-1} + \{\text{deg} < d-1\}$
- $P(\mathbf{x}+\mathbf{a}) - P(\mathbf{x}+\mathbf{b}) = \sum_i (a_i - b_i) \cdot \partial P / \partial x_i + E(\mathbf{x})$
where $\text{deg}(E(\mathbf{x})) < d-1$
- **Lesson**: estimating P on two shifts of S
gives access to a lower degree polynomial

The case $d=2$

- $S+S = \{s_1+s_2 : s_1, s_2 \in S\} = \cup_{s_i \in S} S+s_i$
- Assume $(P(a))_{a \in S+S}$ has $\varepsilon^2/2$ errors
- Call $S_a = S+a$ **good** if contains $\leq \varepsilon/2$ errors
- S_a, S_b good \Rightarrow the degree 1 poly $P(x+a)-P(x+b)$ ($\approx \partial_{a-b}P$) has $\leq \varepsilon$ errors
- Can reconstruct the deg 1 poly $\partial_{a-b}P$ (ignore constant term for now)
- **New goal**: reconstruct P from the set $\{\partial_{a-b}P\}$

The case $d=2$ cont.

- **Recall:** $\partial_{\mathbf{a}-\mathbf{b}} P(X) = \sum_{i=1 \dots m} (\mathbf{a}_i - \mathbf{b}_i) \cdot \partial P / \partial \mathbf{x}_i$

We have:

$$\begin{pmatrix} \partial_{s_1 - s_2} P \\ \partial_{s_1 - s_3} P \\ \vdots \\ \partial_{s_{n-1} - s_n} P \end{pmatrix} + \begin{pmatrix} N \\ O \\ I \\ S \\ E \end{pmatrix}$$

The case $d=2$ cont.

- Recall: $\partial_{\mathbf{a}-\mathbf{b}} P(X) = \sum_{i=1 \dots m} (a_i - b_i) \cdot \partial P / \partial x_i$

$$\begin{pmatrix} s_1 - s_2 \\ s_1 - s_3 \\ \vdots \\ \vdots \\ s_{n-1} - s_n \end{pmatrix} \begin{pmatrix} \partial_{x_1} P \\ \vdots \\ \partial_{x_m} P \end{pmatrix} + \begin{pmatrix} \text{N} \\ \text{O} \\ \text{I} \\ \text{S} \\ \text{E} \end{pmatrix} = \begin{pmatrix} \partial_{s_1 - s_2} P \\ \partial_{s_1 - s_3} P \\ \vdots \\ \vdots \\ \partial_{s_{n-1} - s_n} P \end{pmatrix} + \begin{pmatrix} \text{N} \\ \text{O} \\ \text{I} \\ \text{S} \\ \text{E} \end{pmatrix}$$

- Matrix contains many (shifted) copies of S
- Can use decoder for S to find $(\partial_{x_1} P, \dots, \partial_{x_m} P)$ (recall, S is NIS for deg 1 polynomials!)
- Comparing coefficients we can recover P

The case of general d

- $S^{(d)} = S^{(1)} + S^{(d-1)} = \cup_{s_i \in S} S^{(d-1)} + s_i$
- Assume $(P(a))_{a \in S^{(d)}}$ has $(\epsilon/2)^d$ errors
- $a \in S$ is good if $S_a = S^{(d-1)} + a$ contains $\leq \frac{1}{2}(\epsilon/2)^{d-1}$ errors
- S_a, S_b good \Rightarrow the degree $d-1$ poly $P(x+a) - P(x+b)$ ($\approx \partial_{a-b}P$) has $\leq (\epsilon/2)^{d-1}$ errors
- Can reconstruct the deg $d-1$ poly $\partial_{a-b}P$ (will fix lower order terms later)
- As before can reconstruct P from the set $\{\partial_{a-b}P\}$

Running time analysis

- We make $|S|^2$ calls to the decoding algorithm for degree $d-1$
- After that we make $|S|$ calls to the decoding algorithm for S (for each of the $\binom{n}{d}$ monomials)
- Then, we take a majority vote for each monomial
- After that we recover monomials of degree $< d$
- $t(d) = |S|^2 \cdot t(d-1) + \binom{n}{d} \cdot |S| \cdot \text{Dec}(S) + |S|^2 \cdot \binom{n}{d} + t(d-1)$
 $= O(n^{2d-1})$
- I.e., algorithm runs in (less than) quadratic time

Summary

- Showed construction of noisy Interpolating Set for degree d polynomials over small fields
- Gave decoding algorithm for $\exp(-d)$ fraction of errors
- Q: improve decoding radius to $2^{-d}/2$
- Q: list decoding for radius 2^{-d}
- Q: noisy interpolating sets for **sparse univariate polynomials!** (see Saraf-Yekhanin)
- Applications?



Thank You