Linear algebra and zero knowledge

Guillermo Angeris   Alex Evans

ZK Summit 9, Lisboa

Outline

What are we doing?

- We will generalize a number of important results used in ZK proofs

What are we doing?

- We will generalize a number of important results used in ZK proofs
- Most don't depend on polynomials!

What are we doing?

- We will generalize a number of important results used in ZK proofs
- Most don't depend on polynomials!
- Using (ideally) good notation, linear algebra
- And a sprinkling of error correcting codes

A warning for the brave

- Linear algebra over **R** and **C**:

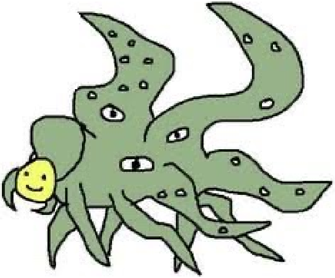
🙂

A warning for the brave

- Linear algebra over **R** and **C**:

  🙂

- Linear algebra over finite fields **F**:



Outline

Notation preliminaries

- We'll use 'probabilistic' implications
- Given propositions (depending on randomness r and r')

$$P_r \quad \underset{p}{\Longrightarrow} \quad Q_{r'}$$

  if $\Pr(P_r \wedge \neg Q_{r'}) \leq p$
- Number of downstream consequences

Implications

- Chaining implications

$$P_r \underset{p}{\Longrightarrow} Q_{r'} \quad \text{and} \quad Q_{r'} \underset{p'}{\Longrightarrow} T_{r''}$$

- Then

$$P_r \underset{p+p'}{\Longrightarrow} T_{r''}$$

Implications

- Chaining implications

$$P_r \underset{p}{\Longrightarrow} Q_{r'} \quad \text{and} \quad Q_{r'} \underset{p'}{\Longrightarrow} T_{r''}$$

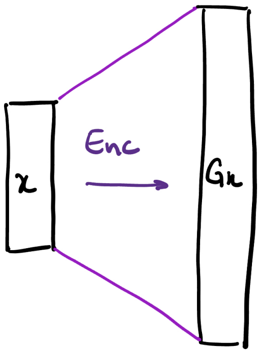
- Then

$$P_r \underset{p+p'}{\Longrightarrow} T_{r''}$$

- Many other things similar to ‘normal’ logic follow
- (With extra error)

Error correcting codes

- Given a linear code (matrix) $G \in \mathbf{F}^{m \times n}$
- We *encode* an n-vector x into a (much larger) codeword Gx



Examples of codes

- Trivial code

$$G = I$$

Examples of codes

- Trivial code

$$G = I$$

- Reed–Solomon code

$$G_{ij} = i^{j-1}$$

- *i.e.*, $(Gx)_r$ encodes a polynomial with coefficients x and evaluates it at r

Error correcting codes (cont)

- We will use one (and only one!) definition from coding theory
- The *distance* d of G is

$$d = \min_{x \neq 0} \|Gx\|_0,$$

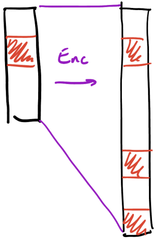
  where $\|\cdot\|_0$ is the number of nonzero entries

Error correcting codes (cont)

- We will use one (and only one!) definition from coding theory
- The *distance* d of G is

$$d = \min_{x \neq 0} \|Gx\|_0,$$

  where $\|\cdot\|_0$ is the number of nonzero entries
- In pictures:



Distances of codes

- The *distance* d of $G \in \mathbf{F}^{m \times n}$ is

$$d = \min_{x \neq 0} \|Gx\|_0$$

- Trivial code $G = I$

$$d = 1$$

- Reed–Solomon codes $G_{ij} = i^{j-1}$

$$d = m - n + 1 \qquad (\text{rows} - \text{cols} + 1)$$

  If $m = |\mathbf{F}|$ then $d = |\mathbf{F}| - n + 1$

Outline

Zero check

- The usual zero check:

$$(Gx)_r = 0 \quad \underset{p}{\Longrightarrow} \quad x = 0,$$

  where r is uniformly chosen from $1, \ldots, m$ and $p \leq 1 - d/m$

- For an RS code $d = |\mathbf{F}| - n + 1$ so

$$p \leq \frac{n-1}{|\mathbf{F}|}$$

Generalized zero check

- "Generalized" zero check, given *vectors* $y_1, \ldots, y_n$,

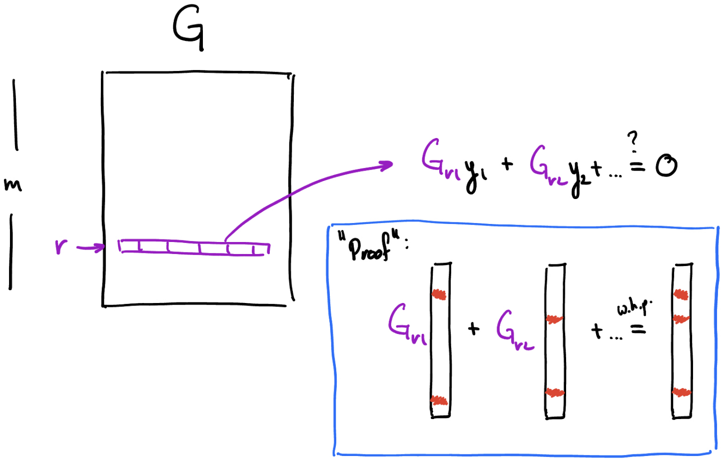
$$\sum_i G_{ri} y_i = 0 \quad \underset{p}{\Longrightarrow} \quad \text{every } y_i = 0,$$

  where $p \leq 1 - d/m$

- For an RS code, same bound as before

$$p \leq \frac{n-1}{|\mathbf{F}|}$$

Generalized zero check (picture!)



Folded zero check

- Take "generalized" zero check and apply the zero check again!

$$\left(G'\left(\sum_i G_{ri} y_i\right)\right)_{r'} = 0 \quad \underset{p'}{\Longrightarrow} \quad \sum_i G_{ri} y_i = 0,$$

and

$$\sum_i G_{ri} y_i = 0 \quad \underset{p}{\Longrightarrow} \quad \text{every } y_i = 0,$$

where $p \leq 1 - d/m$ and $p' \leq 1 - d'/m'$

- For an RS code, this is Schwarz–Zippel (with the same error!)

$$p + p' \leq \frac{(n-1) + (n'-1)}{|\mathbf{F}|}$$

Folded subspace check

- We can reduce checking n inclusions to just checking one

$$\sum_i G_{ri} y_i \in V \quad \underset{p}{\Longrightarrow} \quad \text{every } y_i \in V,$$

  where $p \leq 1 - d/m$ and $V \in \mathbf{F}^k$ is any subspace

- For an RS code $d = |\mathbf{F}| - n + 1$ so (again)

$$p \leq \frac{n-1}{|\mathbf{F}|}$$

Outline

Sparse zero check

- Sparse zero check

$$x_S = 0 \quad \underset{p}{\Longrightarrow} \quad \|x\|_0 \leq q,$$

where $S \subseteq \{1, \ldots, n\}$ uniformly and $p \leq (1 - q/n)^{|S|}$

Folded sparse check (Ligero lite(tm))

- We can 'fold' many vectors y_i and just check the sparsity of one

$$\left\| \sum_i G_{ri} y_i \right\|_0 \leq q \quad \underset{p}{\Longrightarrow} \quad \|y_i\|_0 \leq q,$$

where $p \leq (q+1)(1 - d/m)$

Folded subspace distance check (generalized Ligero)

- We can check that all y_i are q-close to a subspace V by checking a single vector is!

$$\left\| \sum_i G_{ri} y_i - V \right\|_0 \leq q \quad \underset{p}{\Longrightarrow} \quad \|y_i - V\|_0 \leq q,$$

where $p \leq (q+1)(1 - d/m)$ and $q < d'/2$, defined as

$$d' = \min_{v \in V \setminus \{0\}} \|v\|_0$$

Folded subspace distance check (cont)

- Part of the folded subspace distance proof is still open!
- Come and chat with us if this sounds interesting :)

A whirlwind tour

- We just did... 7 checks in 10 minutes
- Many generalizations are straightforward
- We can replace RS in parts with other ECCs that are more computationally efficient
- And we can understand many systems in one framework

A whirlwind tour

- We just did... 7 checks in 10 minutes
- Many generalizations are straightforward
- We can replace RS in parts with other ECCs that are more computationally efficient
- And we can understand many systems in one framework
- Paper (hopefully) soon!

Acknowledgments

- Assimakis Kattis