Linear algebra with errors, coding theory, crypto and Fourier analysis on finite groups

Steven Galbraith

Mathematics Department, University of Auckland

Outline

- Alexander Aitken
- Linear algebra with errors
- Linear algebra modulo p with errors
- Coding theory
- Cryptography
- Fourier analysis on finite groups
- Open questions

Thanks: LMS and NZMS.

Alexander Aitken

- Born in Dunedin, NZ.
- Studied Otago Boys' High School and Otago University.
- Served in WWI at Gallipoli and the Somme.
- PhD Edinburgh 1926.

GALLIPOLI
TO THE SOMME

Recollections of a
New Zealand Infantryman

BY

ALEXANDER AITKEN
8/2524 N.Z.E.F.

WITH AN INTRODUCTION BY

SIR BERNARD FERGUSSON
D.S.O., O.B.E.
Governor-General of New Zealand

LONDON
OXFORD UNIVERSITY PRESS
WELLINGTON MELBOURNE
1963

Alexander Aitken

"Professor Aitken's first year mathematics lectures were rather unusual. The fifty minutes were composed of forty minutes of clear mathematics, five minutes of jokes and stories and five minutes of tricks."

Aitken's mathematics

- Aitken's mathematical work was in statistics, numerical analysis, and algebra.
- He wrote books about matrices and determinants, and wrote important papers about generalised least squares.
  He was also very interested in group theory.
- My research is in computational number theory and applications in cryptography.
- This talk is designed to connect Aitken's work with current hot topics in cryptography.

A game: Learning my secret vector

Your goal is to compute my secret $\underline{s} \in \mathbb{Z}^3$.

- **Chosen queries**:
- You give me $\underline{a} \in \mathbb{Z}^3$.
- I return $\underline{a} \cdot \underline{s}$.

- **Random queries**:
- You ask me for a clue.
- I choose a "random" $\underline{a} \in \mathbb{Z}^3$.
- I return $(\underline{a}, \underline{a} \cdot \underline{s})$.

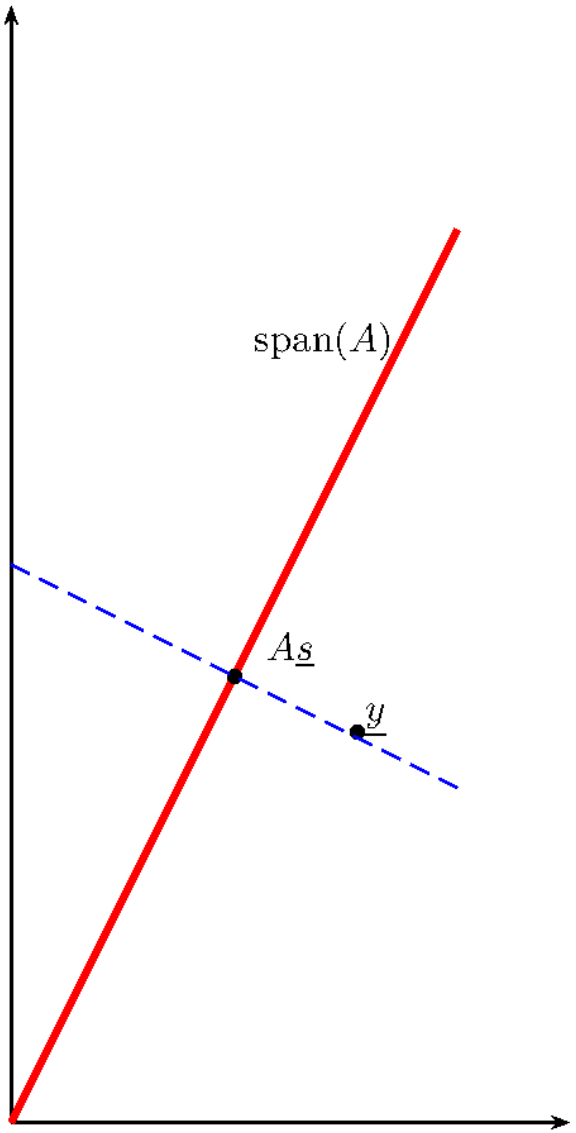
The power of linear algebra

- Let $\underline{s} \in \mathbb{R}^n$ (column vector).
- It doesn't matter if you choose A and I give you $A\underline{s}$, or I choose random A and give you $(A, A\underline{s})$.
- Once A has rank n, you can compute $\underline{s}$.
- This works over any field, and is efficient.

Linear algebra with errors

- Suppose I introduce "errors" into my computation, so you receive $\underline{a} \cdot \underline{s} + e$ for some "small" e that is more likely to be zero than any other value.
  (I will be more precise later.)
- Suppose the "errors" are deterministic, so if you ask me $\underline{a}$ again I return the same value $\underline{a} \cdot \underline{s} + e$.
  Hence, you can't "average away" the errors by repeated queries.
- How can you compute my secret $\underline{s}$? Does the query model matter?

Classical least squares



- Given $m \times n$ matrix A (with $m > n$) and $\underline{y} = A\underline{s} + \underline{e}$, the problem is to compute $\underline{s}$.
- $\underline{y}$ lies in a high dimensional space $\mathbb{R}^m$.
  We want to find point in the n-dimensional subspace (column span of A) that is close to $\underline{y}$.
- Orthogonal projection minimises the error.

Classical least squares

- Let $\underline{s} \in \mathbb{R}^n$, A be an $m \times n$ matrix (with $m > n$) and $\underline{e}$ a "short" error vector in $\mathbb{R}^m$.
- Given A and $\underline{y} = A\underline{s} + \underline{e}$ the problem is to compute $\underline{s}$.
- If $A^T A$ has rank n then the orthogonal projection is

$$\hat{\underline{s}} = (A^T A)^{-1} A^T \underline{y}.$$

- So $\underline{y} - A\hat{\underline{s}}$ is a "short" error vector.
- In other words, solving linear regression is "easy".

Error distributions modulo p

- Let p be a prime (mostly $p = 2$ in this talk).
- We consider error distributions on $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$ such that 0 is the most likely value.
- In the case $p = 2$, we have $\Pr(e = 0) = \mathfrak{p} > \frac{1}{2}$ and $\Pr(e = 1) = 1 - \mathfrak{p}$.
- In the case $p > 2$ we take a discrete normal distribution on $\mathbb{Z}$ with standard deviation much less than p (see later for details).
- Problem: Given $m \times n$ matrix A (with $m > n$) and $\underline{y} = A\underline{s} + \underline{e} \pmod{p}$, where $\underline{e} = (e_1, \ldots, e_m)^T$ for e_i chosen with this distribution, to compute $\underline{s} \in \mathbb{Z}_p^n$.

Error correcting codes

- An (m, n) linear code (binary) is a vector subspace of $\mathbb{Z}_2^m$ of dimension n.
- The **Hamming weight** of a vector $\underline{e} \in \mathbb{Z}_2^m$ is the number of ones in the vector.
- A **code word** is an element $\underline{c} = A\underline{s}$, where the columns of A are a basis for the subspace.
- One transmits the code word $\underline{c}$ along a noisy channel and hopes that only a few bits get corrupted.
- The received word $\underline{y} \in \mathbb{Z}_2^m$ is therefore of the form $\underline{y} = A\underline{s} + \underline{e}$ where $\underline{e}$ has low Hamming weight.
- For certain special matrices A, and low enough error-rates, there are efficient decoding algorithms (more details given later).

Error correcting codes

Let $m = 7$, $n = 4$ and

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

$\underline{s} = (1, 0, 1, 0)^T$ is encoded as $(1, 0, 1, 0, 1, 0, 1)^T$.

$\underline{y} = (1, 1, 1, 0, 1, 0, 1)^T$ and $(1, 0, 1, 0, 1, 0, 0)^T$ are both decoded as $(1, 0, 1, 0)^T$.

Decoding linear codes

- Let A be an arbitrary $m \times n$ binary matrix.
- Then A defines a linear code.
- Given $\underline{y} \in \mathbb{Z}_2^m$ one can ask:
  - Is there a vector $\underline{e}$ of weight less than some given bound w such that $\underline{y} - \underline{e}$ is in the column span of A?
  - Find a vector $\underline{s} \in \mathbb{Z}_2^n$ such that $\underline{y} - A\underline{s}$ has minimal weight.
- Both these computational problems are NP-hard. (Berlekamp, McEliece, van Tilborg, 1978)
- This suggests that linear algebra modulo 2 with errors is hard: we do not expect there to exist an algorithm to solve it with polynomial-time worst-case complexity.

Summary

	Chosen queries	Random queries
Exact Lin. Alg.	Easy	Easy
Lin. Alg. $\mathbb{R}$ with errors	Easy	Easy
Lin. Alg. $\mathbb{Z}_2$ with errors	Easy?	Hard

Remark about "hard" and "easy"

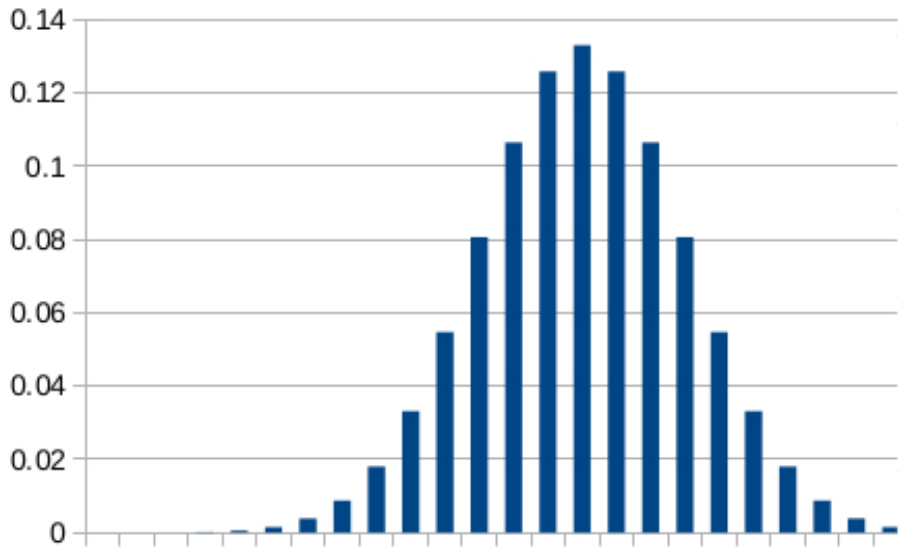
- Input is an $m \times n$ binary matrix A (with $m > n$) and $\underline{y} = A\underline{s} + \underline{e} \pmod 2$.
- So input size is $m(n+1)$ bits.
- An algorithm is efficient (and the problem is considered "easy") if the number of steps is polynomial in the input size.
- A problem is "hard" if we do not know any efficient algorithm.
- So an algorithm that needs n queries to compute $\underline{s} \in \mathbb{Z}_2^n$ is efficient.
- Whereas an algorithm that needs to try all 2^n choices for $\underline{s} \in \mathbb{Z}_2^n$ is not efficient.

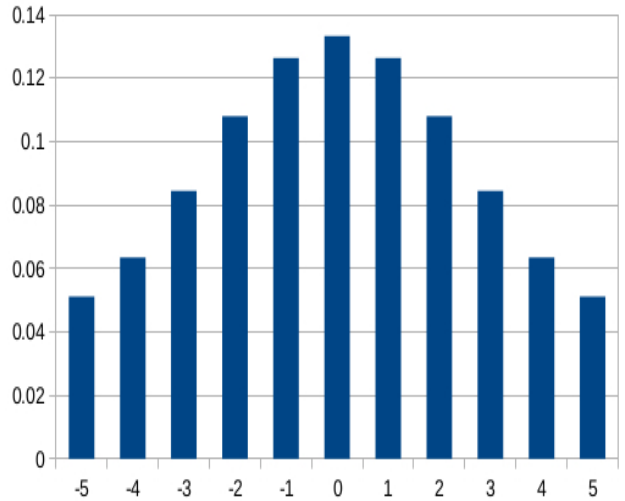
Error distributions modulo p

In the case $p > 2$ we take a discrete normal distribution on $\mathbb{Z}$ with standard deviation much less than p.

Precisely, $\Pr(e = x)$ is proportional to $\exp(-x^2/(2\sigma^2))$ where $\sigma \ll p$.

We then reduce this to $\{-(p-1)/2, \ldots, -1, 0, 1, \ldots, (p-1)/2\}$ by summing over congruence classes modulo p.





Linear algebra modulo $p > 2$ with errors

- Let p be a prime.
- Let $\underline{s} \in \mathbb{Z}_p^n$, A be an $m \times n$ matrix with entries in $\mathbb{Z}_p$.
- Let $\underline{e}$ be an error vector in $\mathbb{Z}_p^m$ with entries chosen independently from the error distribution.
- Given A and $\underline{y} = A\underline{s} + \underline{e} \pmod{p}$ the problem is to compute $\underline{s}$.
- **Exercise:** Explain why linear regression mod p doesn't work. In other words,

$$\hat{\underline{s}} = (A^T A)^{-1} A^T \underline{y} \pmod{p}$$

  is not usually a good estimator for $\underline{s}$, since $\underline{y} - A\hat{\underline{s}}$ is not usually "small".
- **Exercise:** Explain why Gaussian elimination doesn't work.

Linear algebra modulo p with errors

- As already mentioned, the case $p = 2$ is the problem of decoding a random linear code.
- This is also often called learning parity with noise.

- When p is large and the errors are chosen from a discrete normal distribution with parameter $\sigma \ll p$ the problem is called the **learning with errors problem** (LWE).
- This was studied by Oded Regev in 2005.
- Regev proved some strong hardness results.

Summary

	Chosen queries	Random queries
Exact Lin. Alg.	Easy	Easy
Lin. Alg. $\mathbb{R}$ with errors	Easy	Easy
Lin. Alg. $\mathbb{Z}_p$ with errors	?	Hard

Can we use these hard problems for something?

Public key cryptography

- Cryptosystems provide privacy for communication over an insecure channel.
- Traditional cryptography is symmetric: both sender and receiver have the same "key".
- Public key cryptography is asymmetric: the sender requires no secret information to send a private message to the receiver.

Public key cryptography

- Concept was first proposed by James Ellis at GCHQ in 1970.
- First cryptosystems by Clifford Cocks (1973) and Malcolm Williamson (1974).
- In the non-classified community, the first public key cryptosystems were due to Whit Diffie, Martin Hellman, Ralph Merkle, Ron Rivest, Adi Shamir and Len Adleman.
- Public key cryptography is based on hard computational problems.

McEliece public key cryptosystem

- Public key: Generator matrix A for an $m \times n$ linear binary code.
- Private key: A decoding algorithm for the code.
- To encrypt $\underline{s} \in \mathbb{Z}_2^n$:
  - Choose $\underline{e} \in \mathbb{Z}_2^m$ of low Hamming weight
  - Set $\mathrm{c} = A\underline{s} + \underline{e}$.
- To decrypt: Run the secret decoding algorithm to get $\underline{s}$.

Public key cryptography from LWE (Regev)

- Private key: $\underline{s}$
- Public key: A $(m \times n)$, $\underline{y} = A\underline{s} + \underline{e} \pmod{p}$
- To encrypt $\mathrm{M} \in \{0, 1\}$:
  - Choose $\underline{u} \in \{0, 1\}^m$
  - Set $\mathrm{c}_1 = \underline{u}A \pmod{p}, \mathrm{c}_2 = \underline{u} \cdot \underline{y} + \mathrm{M}(p-1)/2 \pmod{p}$
- To decrypt: Compute $v = \mathrm{c}_2 - \mathrm{c}_1 \cdot \underline{s} \pmod{p}$ reduced to the interval $\{-(p-1)/2, \ldots, -1, 0, 1, \ldots, (p-1)/2\}$.
  If $|v| < p/4$ then output 0, else output 1.
- Features: Post-quantum and homomorphic.

Summary

	Chosen queries	Random queries
Exact Lin. Alg.	Easy	Easy
Lin. Alg. $\mathbb{R}$ with errors	Easy	Easy
Lin. Alg. $\mathbb{Z}_p$ with errors	?	Hard

Chosen query model

- I have secret $\underline{s} \in \mathbb{Z}_2^n$.
- You give me $\underline{a} \in \mathbb{Z}_2^n$.
- I return $\underline{a} \cdot \underline{s} + e$ where $e \in \mathbb{Z}_2$ satisfies $\Pr(e = 0) = \mathfrak{p} > \frac{1}{2}$. Note: e is a deterministic function of $\underline{a}$.
- **Basic trick:** Choose unit vector $\underline{u}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ and random $\underline{s}$ and query on $\underline{a} + \underline{u}_i$ and $\underline{a}$.
- Get $y_1 = (\underline{a} + \underline{u}_i) \cdot \underline{s} + e_1$ and $y_2 = \underline{a} \cdot \underline{s} + e_2$.
- With probability $\mathfrak{p}^2 + (1 - \mathfrak{p})^2 > \frac{1}{2}$ we have $y_1 - y_2 = s_i$. (This analysis assumes independent errors; worst case needs $\mathfrak{p} > \frac{3}{4}$.)

Summary

	Chosen queries	Random queries
Exact Lin. Alg.	Easy	Easy
Lin. Alg. $\mathbb{R}$ with errors	Easy	Easy
Lin. Alg. $\mathbb{Z}_p$ with errors	Easy	Hard

Extensions of the problem

- Given A and $\underline{y}$ find all pairs $(\underline{s}, \underline{e})$ such that $\underline{y} = A\underline{s} + \underline{e}$ and $\underline{e}$ is "small".
- In coding theory this is called list decoding.
- Fourier analysis turns out to be a helpful way to think about this problem in the "chosen-query model".

Fourier Analysis on Finite Groups

- Consider $G = \mathbb{Z}_2^n$, a finite additive group of order 2^n.
- The set of functions $f : G \rightarrow \mathbb{C}$ is a $\mathbb{C}$-vector space of dimension 2^n.
- There is an inner product

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{\underline{x} \in G} f(\underline{x}) \overline{g(\underline{x})}$$

- An orthonormal basis for this set of functions is

$$\chi_{\underline{a}}(\underline{x}) = (-1)^{\underline{a} \cdot \underline{x}}$$

where $\underline{a}$ runs over all elements of $\mathbb{Z}_2^n$.

Fourier analysis on finite groups

- Let $f : G \to \mathbb{C}$ be given, $G = \mathbb{Z}_2^n$.
- Then f has a Fourier expansion

$$f(\underline{x}) = \sum_{\underline{a} \in \mathbb{Z}_2^n} \hat{f}(\underline{a}) \chi_{\underline{a}}(\underline{x})$$

  where the Fourier coefficients are $\hat{f}(\underline{a}) = \langle f, \chi_{\underline{a}} \rangle$.
- Parseval's identity: $\langle f, f \rangle = \sum_{\underline{a} \in G} |\hat{f}(\underline{a})|^2$.
- We call a character $\chi_{\underline{a}}$ **heavy** if $|\hat{f}(\underline{a})|$ is "relatively large".
- Parseval implies there cannot be many heavy Fourier coefficients.

Lemma

- Let $X \subseteq \mathbb{Z}_2^n$ such that $|X| = \delta 2^n$.
- Let $f : \mathbb{Z}_2^n \rightarrow \{1, -1\}$ be such that

$$f(\underline{x}) = (-1)^{\underline{x} \cdot \underline{s}} = \chi_{\underline{s}}(\underline{x})$$

for all $\underline{x} \in X$, and

$$f(\underline{x}) = (-1)^{\underline{x} \cdot \underline{s} + 1} = -\chi_{\underline{s}}(\underline{x})$$

for all $\underline{x} \in \overline{X} = \mathbb{Z}_2^n \setminus X$.

- Then $\langle f, f \rangle = 1$ and $\hat{f}(\underline{s}) = 2\delta - 1$.

Linear algebra with errors

- Consider $\underline{y} = A\underline{s} + \underline{e} \in \mathbb{Z}_2^m$.
- Think of this as m pairs $(\underline{a}_i, y_i = \underline{a}_i \cdot \underline{s} + e_i)$.
- Then $(-1)^{y_i}$ is the value $f(\underline{a}_i)$ of a function f like the one on the previous slide.
- Since e_i is usually 0 then $\delta \gg \frac{1}{2}$ and so $\hat{f}(\underline{s})$ is heavy.
- Hence, the problem of computing $\underline{s}$ is re-phrased as computing a heavy Fourier coefficient.
- It is not efficient to compute all the Fourier coefficients $\hat{f}(\underline{x})$, as there are 2^n characters $\chi_{\underline{x}}$.

Goldreich-Levin/Kushilevitz-Mansour

- In 1989 Goldreich and Levin published a landmark paper in cryptography and learning theory.
  They sketched a learning algorithm for heavy Fourier coefficients.
- In 1993 Kushilevitz and Mansour presented a more general algorithm.
- There is a community in Engineering that studies related algorithms under the name "sparse Fourier transform" (see recent survey paper by Gilbert, Indyk, Iwen and Schmidt).
- Hence, we know an efficient algorithm to compute a list of heavy Fourier coefficients of a function f, by querying the function at certain points.

Prefix/Filter function

- Fix $f : \mathbb{Z}_2^n \rightarrow \mathbb{C}$.
- Let $1 \leq k \leq n$.
- For $\underline{a} \in \mathbb{Z}_2^n$ write $\underline{a} = \alpha\beta$ where $\alpha \in \mathbb{Z}_2^k$ and $\beta \in \mathbb{Z}_2^{n-k}$.
- For fixed $\alpha \in \mathbb{Z}_2^k$ define $f_\alpha : \mathbb{Z}_2^{n-k} \rightarrow \mathbb{C}$ by

$$f_\alpha(\underline{x}) \ = \ \frac{1}{2^{n-k}} \sum_{\beta \in \mathbb{Z}_2^{n-k}} \hat{f}(\alpha\beta)\chi_\beta(\underline{x}).$$

- Parseval states

$$\langle f_\alpha, f_\alpha \rangle \ = \ \sum_{\beta \in \mathbb{Z}_2^{n-k}} |\hat{f}(\alpha\beta)|^2.$$

Estimating a prefix function

- Recall the definition: Given $\alpha \in \mathbb{Z}_2^k$ we set

$$f_\alpha(\underline{x}) = \frac{1}{2^{n-k}} \sum_{\beta \in \mathbb{Z}_2^{n-k}} \hat{f}(\alpha\beta)\chi_\beta(\underline{x}).$$

- **Lemma:**

$$f_\alpha(\underline{x}) = \frac{1}{2^k} \sum_{\underline{y} \in \mathbb{Z}_2^k} f(\underline{y}\underline{x})\chi_\alpha(\underline{y}).$$

- Given $\underline{x}$ one can estimate $f_\alpha(\underline{x})$ by choosing some $\underline{y} \in \mathbb{Z}_2^k$ and sampling the function on $\underline{y}\underline{x}$.

The Kushilevitz-Mansour algorithm

- We want to find heavy Fourier coefficients of $f : \mathbb{Z}_2^n \to \mathbb{C}$.
- The algorithm computes a list $\mathcal{L}$ of "candidate prefixes" $\alpha \in \mathbb{Z}_2^k$.
- Suppose $\underline{s} = (s_1, \ldots, s_n)$ is a heavy fourier coefficient for f and that $\alpha = (s_1, \ldots, s_k)$.
  Then $\langle f_\alpha, f_\alpha \rangle \geq |\hat{f}(\underline{s})|^2$.
- Similarly, for $b = s_{k+1}$,

$$|\langle f_{\alpha b}, f_{\alpha b} \rangle| \ \geq \ |\hat{f}(\underline{s})|^2.$$

- Write $\overline{b} = 1 - b$. Parseval implies

$$\langle f_\alpha, f_\alpha \rangle = \langle f_{\alpha b}, f_{\alpha b} \rangle + \langle f_{\alpha \overline{b}}, f_{\alpha \overline{b}} \rangle.$$

- Hence, if $\langle f_{\alpha \overline{b}}, f_{\alpha \overline{b}} \rangle$ is small, and we can estimate its value accurately, then we can eliminate $\alpha \overline{b}$.

The Kushilevitz-Mansour algorithm

- At each step the algorithm takes $\alpha \in \mathcal{L}$ and computes approximations to $\langle f_{\alpha b}, f_{\alpha b} \rangle$ for $b \in \{0,1\}$.
- The algorithm then extends the list $\mathcal{L}$ of "candidate prefixes" from $\mathbb{Z}_2^k$ to $\mathbb{Z}_2^{k+1}$.
- Parseval implies the list does not become too large.
- The algorithm runs in polynomial-time and requires polynomially-many **chosen** samples of the function f.
- Mansour has developed this algorithm for functions on $\mathbb{Z}_{2^n}$ and Akavia to $\mathbb{Z}_p$ and more general Abelian groups.

Connections with my work

- In collaboration with my former post-doc Shi Bai I have worked on algorithms using lattices to solve LWE in special cases.

S. Bai and S. D. Galbraith, "Lattice Decoding Attacks on Binary LWE", in W. Susilo and Y. Mu (eds.), ACISP 2014, Springer LNCS 8544 (2014) 322–337.

S. Bai, S. D. Galbraith, L. Li and D. Sheffield, "Improved Combinatorial Algorithms for the Inhomogeneous Short Integer Solution Problem". Submitted.

- In collaboration with my PhD student Barak Shani I have used the Goldreich-Levin/Kushilevitz-Mansour algorithm in the case of $G = \mathbb{Z}_p$ to prove some "hardness" results relevant for cryptography.

S. D. Galbraith and B. Shani, "The Multivariate Hidden Number Problem", in A. Lehmann and S. Wolf (eds), 8th International Conference on Information-Theoretic Security (ICITS), Springer LNCS 9063 (2015) 250-268.

Open Questions

- What are the best algorithms for solving linear algebra with errors for various parameter ranges?
- Improve the analysis of these algorithms and get better tools for determining parameters for high-security cryptography.
- Ditto for the Ring-LWE problem, which is where $\underline{a} \cdot \underline{s}$ is replaced by multiplication in the cyclotomic ring $\mathbb{Z}_p[x]/(x^n + 1)$.
- Are Fourier learning algorithms optimal from a concrete point of view?

Thank you for your attention

Solution to the question about least squares

- Least squares computes an orthogonal projection of $\underline{y} = A\underline{s} + \underline{e}$ onto the space spanned by A.
- Over $\mathbb{R}$, orthogonal projection minimises the error.
- A first problem is that $\mathbb{Z}_p^m$ is not a metric space.
- Over $\mathbb{Z}_p$, orthogonal projection still makes sense, but it does not behave well with respect to the intuitive sense of "distance".

Example

$$\underline{s} = \begin{pmatrix} 5 \\ 76 \end{pmatrix}, \quad A = \begin{pmatrix} 22 & 102 \\ 191 & 176 \\ -26 & 104 \end{pmatrix}, \quad \underline{e} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}.$$

Least squares on $\underline{b} = A\underline{s} + \underline{e}$ computes

$$\hat{\underline{s}} \approx \begin{pmatrix} 4.993 \\ 76.003 \end{pmatrix}.$$

Now work over $\mathbb{Z}_{311}$. The formula gives

$$\hat{\underline{s}} = \begin{pmatrix} 274 \\ 223 \end{pmatrix}.$$

Why can't we use Gaussian elimination?

- Given $m \times n$ matrix A and $\underline{y} = A\underline{s} + \underline{e} \pmod{p}$ the problem is to compute $\underline{s}$.
- Perform row operations on A to construct a matrix U such that UA starts with identity matrix.
- Then $U\underline{y} = UA\underline{s} + U\underline{e}$.
- The problem is that the entries of U behave like random elements of $\mathbb{Z}_p$ and so $U\underline{e}$ looks like a uniform vector.
- The discrete nature of arithmetic in $\mathbb{Z}_p$ is relevant here. In traditional numerical analysis we think of Gaussian elimination being relatively stable with respect to errors.