Chaining arguments and list decoding

Mary Wootters (based on work with Atri Rudra)

June 23, 2016
1 Error Correcting Codes and List Decoding

2 Random Codes

3 Setting up a Chaining Argument
   - Average-radius list-decodability
   - Pass to a Gaussian process
   - Controlling the Gaussian process

4 Conclusion
1 Error Correcting Codes and List Decoding

2 Random Codes

3 Setting up a Chaining Argument
   - Average-radius list-decodability
   - Pass to a Gaussian process
   - Controlling the Gaussian process

4 Conclusion

The point of this talk is the chaining argument
Error Correcting Codes and List Decoding

Random Codes

Setting up a Chaining Argument
- Average-radius list-decodability
- Pass to a Gaussian process
- Controlling the Gaussian process

Conclusion
Error correcting codes

Alice \quad \rightarrow \quad \text{Adversary} \quad \rightarrow \quad \text{Bob}
Error correcting codes

Alice  Adversary  Bob
Error correcting codes

Eats shoots and leaves

Alice

Adversary

Bob
Error correcting codes

Eats, shoots, and leaves

Adversary

Alice

Bob
Error correcting codes

Eats, shoots, and leaves

Adversary

Alice

Bob
Error correcting codes

- $C \subset \mathbb{F}_q^n$ is a code, with (relative Hamming) distance $\delta$. 

\[
C \subset \mathbb{F}_q^n
\]
Error correcting codes

- $C \subset \mathbb{F}_q^n$ is a code, with (relative Hamming) distance $\delta$.
- Alice and Bob can use $C$ to communicate:

Alice

$\mathbb{F}_q^n$

Bob
Error correcting codes

- $C \subset \mathbb{F}_q^n$ is a code, with (relative Hamming) distance $\delta$.
- Alice and Bob can use $C$ to communicate.
Error correcting codes

- $C \subset \mathbb{F}_q^n$ is a code, with (relative Hamming) distance $\delta$.
- Alice and Bob can use $C$ to communicate:

\[
\rho = \text{error rate}
\]

\[
R = \frac{\log_q |C|}{n}
\]
Error correcting codes

- $\mathcal{C} \subset \mathbb{F}_q^n$ is a code, with (relative Hamming) distance $\delta$.
- Alice and Bob can use $\mathcal{C}$ to communicate:

\[ \rho \]
\[ \mathbb{F}_q^n \]
Error correcting codes

- $C \subset \mathbb{F}_q^n$ is a code, with (relative Hamming) distance $\delta$.
- Alice and Bob can use $C$ to communicate:
  - error rate is $\rho$
  - The rate of $C$ is $R = \log_q |C|/n$. 
But what if the error rate $\rho > \delta/2$?

- Bob cannot uniquely decode Alice’s message.
But what if the error rate $\rho > \delta/2$?

- Bob cannot uniquely decode Alice’s message.
- List decoding to the rescue!

Alice meant to say something blue.
List decodable codes

Definition

A code $C \subset \mathbb{F}_q^n$ is $(\rho, L)$-list-decodable if

$$\max_{w \in \mathbb{F}_q^n} |\{c \in C : d(c, w) \leq \rho\}| \leq L.$$
Now $\rho$ can be big!

- Turns out (for large $q$), there are codes which are $(\rho, L)$-list decodable with:
  - Error rate $\rho = 1 - \varepsilon$
  - Rate $R = \Omega(\varepsilon)$
  - List size $L = O(1/\varepsilon)$
Now $\rho$ can be big!

- Turns out (for large $q$), there are codes which are $(\rho, L)$-list decodable with:
  - Error rate $\rho = 1 - \varepsilon$
  - Rate $R = \Omega(\varepsilon)$
  - List size $L = O(1/\varepsilon)$

- That means Alice and Bob can win...
  ...even when almost all of the symbols are corrupted!

![Alice](image1.png) ![Bob](image2.png)
Not just for Alice and Bob

- Also lots of connections to complexity theory:
  - Hardness amplification
  - Hardcore predicates from one-way functions
  - Extractors
  - Expanders
  - Pseudorandom generators
  - Hardness of computing permanents

- (See [Sudan’00] or [Vadhan’11] for good surveys).
Questions

1. What codes are (near-)optimally list-decodable? (up to $\rho = 1 - \varepsilon$)
Questions

1. What codes are (near-)optimally list-decodable? \((\text{up to } \rho = 1 - \varepsilon)\)

2. Where’s the randomness here?
Questions

1. What codes are (near-)optimally list-decodable? (up to $\rho = 1 - \varepsilon$)

2. Where’s the randomness here?

I thought this was a workshop about chaining arguments?

Bob
Questions

1. What codes are (near-)optimally list-decodable? (up to $\rho = 1 - \varepsilon$)
   - Completely random codes

2. Where's the randomness here?

   I thought this was a workshop about chaining arguments?
Questions

1. What codes are (near-)optimally list-decodable? \( (\text{up to } \rho = 1 - \varepsilon) \)
   - Completely random codes
   - A few special deterministic codes [Guruswami-Rudra’08...]

2. Where’s the randomness here?

   I thought this was a workshop about chaining arguments?

Bob
Questions

1. What codes are (near-)optimally list-decodable? (up to $\rho = 1 - \varepsilon$)
   - Completely random codes
   - Structured random codes
   - A few special deterministic codes [Guruswami-Rudra’08...]

2. Where’s the randomness here?

I thought this was a workshop about chaining arguments?

Bob
Questions

1. What codes are (near-)optimally list-decodable? \((\text{up to } \rho = 1 - \varepsilon)\)
   - Completely random codes ← This talk
   - Structured random codes ← This talk
   - A few special deterministic codes [Guruswami-Rudra’08…]

2. Where’s the randomness here?

I thought this was a workshop about chaining arguments?

Bob
1 Error Correcting Codes and List Decoding

2 Random Codes

3 Setting up a Chaining Argument
   - Average-radius list-decodability
   - Pass to a Gaussian process
   - Controlling the Gaussian process

4 Conclusion
Structured random codes

- **Headline result:** random Reed-Solomon Codes
Structured random codes

- **Headline result:** random Reed-Solomon Codes
- **This talk:** random linear codes.
Structured random codes

- **Headline result:** random Reed-Solomon Codes
- **This talk:** random linear codes.

A random linear code $C \subset \mathbb{F}_q^n$ is a random $k$-dimensional subspace of $\mathbb{F}_q^n$. 

$$C(x) = G \times$$ 

$G \in \mathbb{F}_{q}^{n \times k}$ is random
Let $C \subset \mathbb{F}_q^n$ be a completely random code.
But first! Completely random codes

- Let $C \subset \mathbb{F}_q^n$ be a completely random code.
But first! Completely random codes

Let $C \subseteq \mathbb{F}_q^n$ be a completely random code.
But first! Completely random codes

Let $C \subseteq \mathbb{F}_q^n$ be a completely random code.

- Choose error rate $\rho = 1 - \varepsilon$.
- If the rate $R = \log_2 |C|/n$ is about $\varepsilon$, then $C$ is ($\rho$, $1/\varepsilon$)-list-decodable.
Let $C \subset \mathbb{F}_q^n$ be a completely random code.
But first! Completely random codes

- Let $C \subseteq \mathbb{F}_q^n$ be a completely random code.
But first! Completely random codes

- Let $C \subseteq \mathbb{F}_q^n$ be a completely random code.
- Choose error rate $\rho = 1 - \varepsilon$.
- If the rate $R = \log_q |C|/n$ is about $\varepsilon$, then $C$ is $(\rho, 1/\varepsilon)$-list-decodable.
Let $C \subset \mathbb{F}_q^n$ be a completely random code.

Choose error rate $\rho = 1 - \varepsilon$.

If the rate $R = \log_q |C|/n$ is about $\varepsilon$, then $C$ is $(\rho, 1/\varepsilon)$-list-decodable.
A random code is list-decodable to capacity

- Fix \( w \in \mathbb{F}_q^n \).

\[ \rho = 1 - \varepsilon \]

\( B \)

\[ \mathbb{P}\{ C(x) \in B \} \leq \text{small} \approx q^n \varepsilon \]

Fix messages \( \Lambda = \{ x_0, \ldots, x_L \} \subset \mathbb{F}_q^k \).

\[ \mathbb{P}\{ C(x_0), \ldots, C(x_L) \in B \} \leq \text{small} \]

Bad event (for fixed \( w, \Lambda \)) is very unlikely!

Union bound over all \( w, \Lambda \):

\[ \mathbb{P}\{ \exists w, \Lambda, \text{bad event} \} \leq q^n (q^k L + 1) \text{small} \approx q^n + kL - n \varepsilon L \]

If \( L \approx \frac{1}{\varepsilon} \) and \( \frac{k}{n} > \varepsilon \), then this is small!
A random code is list-decodable to capacity

- Fix $w \in \mathbb{F}_q^n$.

\[
P\{C(x) \in B\} \leq \text{small} \approx q^{-\varepsilon n}
\]
A random code is list-decodable to capacity

- Fix $w \in \mathbb{F}_q^n$.

$$\mathbb{P} \{C(x) \in B\} \leq \text{small} \approx q^{-\varepsilon n}$$

- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.
A random code is list-decodable to capacity

- Fix $w \in \mathbb{F}_q^n$.

  $$\mathbb{P}\{C(x) \in B\} \leq \text{small} \approx q^{-\varepsilon n}$$

- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.

  $$\mathbb{P}\{C(x_0), \ldots, C(x_L) \in B\} \leq \text{small}^{L+1}$$
A random code is list-decodable to capacity

- Fix $w \in \mathbb{F}_q^n$. 
  \[
  \mathbb{P} \{ C(x) \in B \} \leq \text{small} \approx q^{-\varepsilon n}
  \]

- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$. 
  \[
  \mathbb{P} \{ C(x_0), \ldots, C(x_L) \in B \} \leq \text{small}^{L+1}
  \]

**Bad event**
(for fixed $w, \Lambda$)
is very unlikely!
A random code is list-decodable to capacity

Fix $w \in \mathbb{F}_q^n$.

$$\mathbb{P} \{ C(x) \in B \} \leq \text{small} \approx q^{-\varepsilon n}$$

Fix messages $\Lambda = \{ x_0, \ldots, x_L \} \subset \mathbb{F}_q^k$.

$$\mathbb{P} \{ C(x_0), \ldots, C(x_L) \in B \} \leq \text{small}^{L+1}$$

Union bound over all $w, \Lambda$:

$$\mathbb{P} \{ \exists w, \Lambda, \text{bad event} \} \leq q^n \binom{q^k}{L+1} \text{small}^{L+1}$$

$$\approx q^{n+kL-n\varepsilon L}$$
A random code is list-decodable to capacity

- Fix $w \in \mathbb{F}_q^n$.
  $$\mathbb{P}\{C(x) \in B\} \leq \text{small} \approx q^{-\varepsilon n}$$

- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.
  $$\mathbb{P}\{C(x_0), \ldots, C(x_L) \in B\} \leq \text{small}^{L+1}$$

- Union bound over all $w, \Lambda$:
  $$\mathbb{P}\{\exists w, \Lambda, \text{bad event}\} \leq q^n \left( \frac{q^k}{L + 1} \right)^{\text{small}^{L+1}} \approx q^{n + kL - n\varepsilon L}$$

If $L \approx 1/\varepsilon$ and $k/n > \varepsilon$, then this is small!
Try it for random linear codes

\[ C(x) = G \]

\[ G \in \mathbb{F}_q^{n \times k} \text{ is random} \]
Try it for random linear codes

- Fix \( w \in \mathbb{F}_q^n \).

\[
\mathbb{P} \{ C(x) \in B \} \leq \text{small} \approx q^{-\varepsilon n}
\]

\[
C(x) \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \q
Try it for random linear codes

- Fix $w \in \mathbb{F}_q^n$.

\[ \mathbb{P}\{C(x) \in B\} \leq \text{small} \approx q^{-\varepsilon n} \]

- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.

$G \in \mathbb{F}_q^{n \times k}$ is random.
Try it for random linear codes

- Fix $w \in \mathbb{F}_q^n$.

\[ \mathbb{P}\{C(x) \in B\} \leq \text{small} \approx q^{-\varepsilon n} \]

- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.

\[ \mathbb{P}\{C(x_0), \ldots, C(x_L) \in B\} \leq \text{small}^{\log_q(L+1)} \]

$G \in \mathbb{F}_q^{n \times k}$ is random
Try it for random linear codes

\[ C(x) = Gx \]

- Fix \( w \in \mathbb{F}_q^n \).
  \[ \mathbb{P}\{C(x) \in B\} \leq \text{small} \approx q^{-\varepsilon n} \]

- Fix messages \( \Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k \).
  \[ \mathbb{P}\{C(x_0), \ldots, C(x_L) \in B\} \leq \text{small}^{\log_q(L+1)} \]

**Bad event** (for fixed \( w, \Lambda \)) is kind of unlikely.

\( G \in \mathbb{F}_q^{n \times k} \) is random
Try it for random linear codes

- Fix $w \in \mathbb{F}_q^n$.

\[
P \{ C(x) \in B \} \leq \text{small} \approx q^{-\varepsilon n}
\]

- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.

\[
P \{ C(x_0), \ldots, C(x_L) \in B \} \leq \text{small}^{\log_q(L+1)}
\]

- Union bound over all $w, \Lambda$...

**Bad event** (for fixed $w, \Lambda$) is kind of unlikely.

$C(x)$

\[
G \in \mathbb{F}_q^{n \times k} \text{ is random}
\]
Try it for random linear codes

- Fix $w \in \mathbb{F}_q^n$.
  \[
P \{ C(x) \in B \} \leq \text{small} \approx q^{-\varepsilon n}
  \]

- Fix messages $\Lambda = \{ x_0, \ldots, x_L \} \subset \mathbb{F}_q^k$.
  \[
P \{ C(x_0), \ldots, C(x_L) \in B \} \leq \text{small} \log_q(L+1)
  \]

- Union bound over all $w, \Lambda$...
  ...need $L \approx q^{1/\varepsilon}$ to get away with rate $R \approx \varepsilon$

Bad event (for fixed $w, \Lambda$) is kind of unlikely.

$G \in \mathbb{F}_q^{n \times k}$ is random
Try it for random linear codes

- Fix $w \in \mathbb{F}_q^n$.
  \[ \mathbb{P}\{C(x) \in B\} \leq \text{small} \approx q^{-\varepsilon n} \]

- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.
  \[ \mathbb{P}\{C(x_0), \ldots, C(x_L) \in B\} \leq \text{small}^{\log_q(L+1)} \]

- Union bound over all $w, \Lambda$...
  \dots need $L \approx q^{1/\varepsilon}$ to get away with rate $R \approx \varepsilon$

Question:
Are random linear codes as list-decodable as random codes?
Try it for random linear codes

2-second lit review:
- Asked by [Elias'91].
- small $\rho$.
  - [Guruswami, Hastad, Kopparty'11]
- small $q$, large $\rho$.
  - [Cheraghchi, Guruswami, Velingker'13], [W.'13]
- large $q$, large $\rho$.
  - [Rudra-W.'14]

- Fix $w \in \mathbb{F}_q^n$.
  $\mathbb{P}\{C(x) \in B\} \leq \text{small} \approx q^{-\varepsilon n}$

- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.
  $\mathbb{P}\{C(x_0), \ldots, C(x_L) \in B\} \leq \text{small}^{\log_q(L+1)}$

- Union bound over all $w, \Lambda$...
  ...need $L \approx q^{1/\varepsilon}$ to get away with rate $R \approx \varepsilon$

Question:
Are random linear codes as list-decodable as random codes?
This looks like a job for...

a CHAINING ARGUMENT!
Error Correcting Codes and List Decoding

Random Codes

Setting up a Chaining Argument
- Average-radius list-decodability
- Pass to a Gaussian process
- Controlling the Gaussian process

Conclusion
1 Error Correcting Codes and List Decoding

2 Random Codes

3 Setting up a Chaining Argument
   - Average-radius list-decodability
   - Pass to a Gaussian process
   - Controlling the Gaussian process

4 Conclusion
Average-Radius List Decodability

List Decodable

All lists of size $> L$ have at least one codeword further than $\rho$

Average-Radius List Decodable

All lists of size $> L$ have average distance larger than $\rho$
Average-Radius List Decodability

List Decodable

All lists of size $> L$ have at least one codeword further than $\rho$

Average-Radius List Decodable

All lists of size $> L$ have average distance larger than $\rho$
All lists of size $> L$ have **average distance** larger than $\rho$. 

- Average-Radius List Decodability

\[
\max_{\Lambda \subset F^k q, |\Lambda| = L+1} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \leq n \cdot (L+1) \cdot (1 - \rho)
\]
All lists of size $> L$ have **average distance** larger than $\rho$. 

$$\Leftrightarrow$$

$$\max_{\Lambda \subseteq \mathbb{F}_q^k, |\Lambda| = L+1} \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \leq n \cdot (L + 1) \cdot (1 - \rho)$$
All lists of size $> L$ have average distance larger than $\rho$.

\[ \max_{\Lambda \subset \mathbb{F}_q^k, |\Lambda| = L+1} \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \leq n \cdot (L + 1) \cdot (1 - \rho) \]
All lists of size $> L$ have **average distance** larger than $\rho$.

\[
\max_{\Lambda \subseteq \mathbb{F}_q^k, |\Lambda| = L+1} \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \leq n \cdot (L + 1) \cdot (1 - \rho)
\]
All lists of size $> L$ have **average distance** larger than $\rho$.

\[
\max_{\Lambda \subseteq \mathbb{F}_q^k, |\Lambda| = L + 1} \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \leq n \cdot (L + 1) \cdot (1 - \rho)
\]
\[
\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} agr(w, C(x)) \quad \text{for a random linear code}
\]
\[ \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \] for a random linear code
\[
\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \quad \text{for a random linear code}
\]
\[ \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \] for a random linear code
\[ \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \] for a random linear code

\[ \text{Average agreement of } \Lambda \text{ with the worst } w: \]

\[ \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) = 2 + 5 + 2 + 3 + 3 + 2 = |\Lambda| \cdot n \sum_{j=1}^{\text{pl}_j(\Lambda)} \]
\[
\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \quad \text{for a random linear code}
\]
\[
\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \quad \text{for a random linear code}
\]
\[
\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \quad \text{for a random linear code}
\]
\[
\max_{w \in F_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \quad \text{for a random linear code}
\]

Average agreement of \( \Lambda \) with the worst \( w \):

\[
\max_{w \in F_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) = 2 + 5 + 2 + 3 + 3 + 2
\]
\[
\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \quad \text{for a random linear code}
\]

Average agreement of \( \Lambda \) with the worst \( w \):

\[
\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) = 2 + 5 + 2 + 3 + 3 + 2 =: |\Lambda| \cdot \sum_{j=1}^{n} \text{pl}_j(\Lambda).
\]
\[
\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) \quad \text{for a random linear code}
\]

Average agreement of \( \Lambda \) with the worst \( w \):

\[
\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \text{agr}(w, C(x)) = 2 + 5 + 2 + 3 + 3 + 2 =: |\Lambda| \cdot \sum_{j=1}^{n} \text{pl}_j(\Lambda).
\]

Independent!
1. Error Correcting Codes and List Decoding

2. Random Codes

3. Setting up a Chaining Argument
   - Average-radius list-decodability
   - Pass to a Gaussian process
   - Controlling the Gaussian process

4. Conclusion
Yay! A sum of independent random variables

To show that $C$ is list-decodable, suffices to show that

$$\mathbb{E} \max_{\Lambda} \sum_{j=1}^{n} pl_j(\Lambda) \leq \text{small}.$$
Yay! A sum of independent random variables

To show that $\mathcal{C}$ is list-decodable, suffices to show that

$$\mathbb{E} \max_{\Lambda} \sum_{j=1}^{n} pl_j(\Lambda) \leq \text{small}. \quad (1)$$

As usual:

- Show

$$\max_{\Lambda} \mathbb{E} \sum_{j=1}^{n} pl_j(\Lambda) \leq \text{small} \quad (2)$$

- Show

$$\mathbb{E} \max_{\Lambda} \left| \sum_{j=1}^{n} pl_j(\Lambda) - \mathbb{E} pl_j(\Lambda) \right| \leq \text{small}. \quad (3)$$
Yay! A sum of independent random variables

To show that \( \mathcal{C} \) is list-decodable, suffices to show that

\[
\mathbb{E} \max_{\Lambda} \sum_{j=1}^{n} pl_j(\Lambda) \leq \text{small}.
\]

As usual:

- Show

\[
\max_{\Lambda} \mathbb{E} \sum_{j=1}^{n} pl_j(\Lambda) \leq \text{small}
\]

- Show

\[
\mathbb{E} \max_{\Lambda} \left| \sum_{j=1}^{n} pl_j(\Lambda) - \mathbb{E} pl_j(\Lambda) \right| \leq \text{small}.
\]
Standard tricks

\[
\mathbb{E} \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} p_l(\Lambda) - \mathbb{E} p_l(\Lambda) \right|
\]
Standard tricks

\[ \mathbb{E} \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} p_{j}(\Lambda) - \mathbb{E}p_{j}(\Lambda) \right| \lesssim \mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} g_{j} \cdot p_{j}(\Lambda) \]

Symmetrization and comparison
Standard tricks

\[ \mathbb{E} \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} pl_j(\Lambda) - \mathbb{E} pl_j(\Lambda) \right| \lesssim \mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} g_j \cdot pl_j(\Lambda) \]

\[ = \mathbb{E}_C \left[ \mathbb{E}_g \max_{|\Lambda|=L} g_j \cdot pl_j(\Lambda) \mid C \right] \]

Symmetrization and comparison

Condition on \( C \) until further notice
Standard tricks

\[ \mathbb{E} \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} p_{l_j}(\Lambda) - \mathbb{E} p_{l_j}(\Lambda) \right| \lesssim \mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} g_{j} \cdot p_{l_j}(\Lambda) \]

\[ = \mathbb{E} C \left[ \mathbb{E}_{g} \max_{|\Lambda|=L} g_{j} \cdot p_{l_j}(\Lambda) \mid C \right] \]

Symmetrization and comparison

GOAL: bound the Gaussian mean width of

\[ \left\{ \begin{pmatrix} p_{l_1}(\Lambda) \\ p_{l_2}(\Lambda) \\ \vdots \\ p_{l_n}(\Lambda) \end{pmatrix} : |\Lambda| = L \right\} \]
1 Error Correcting Codes and List Decoding

2 Random Codes

3 Setting up a Chaining Argument
   - Average-radius list-decodability
   - Pass to a Gaussian process
   - Controlling the Gaussian process

4 Conclusion
Attempt 1: Dudley’s theorem

**GOAL:** bound the Gaussian mean width of

\[
\left\{ \left( \begin{array}{c}
pl_1(\Lambda) \\
pl_2(\Lambda) \\
\vdots \\
pl_n(\Lambda)
\end{array} \right) : |\Lambda| = L \right\}
\]
Attempt 1: Dudley’s theorem

**GOAL:** bound the Gaussian mean width of
\[
\left\{ \begin{pmatrix} p_1(\Lambda) \\ p_2(\Lambda) \\ \vdots \\ p_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}
\]
Attempt 1: Dudley’s theorem

**GOAL:** bound the Gaussian mean width of

\[
\left\{ \begin{pmatrix} p_1(\Lambda) \\ p_2(\Lambda) \\ \vdots \\ p_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}
\]

- Natural choice of nets:

\[
\mathcal{N}_t = \left\{ \begin{pmatrix} p_1(\Lambda) \\ p_2(\Lambda) \\ \vdots \\ p_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}
\]

- Natural way to show that any $\vec{p}_t(\Lambda)$ is close to some element of $\mathcal{N}_t$:
Choose a random subset of $\Lambda$ of size $L/2^t$. 

Mary Wootters
Chaining and list decoding
Attempt 1: Dudley’s theorem

**GOAL:** bound the Gaussian mean width of

\[
\left\{ \begin{pmatrix} pl_1(\Lambda) \\ pl_2(\Lambda) \\ \vdots \\ pl_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}
\]

▶ Natural choice of nets:

\[
\mathcal{N}_t = \left\{ \begin{pmatrix} pl_1(\Lambda) \\ pl_2(\Lambda) \\ \vdots \\ pl_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}
\]

▶ Natural way to show that any \(\vec{pl}(\Lambda)\) is close to some element of \(\mathcal{N}_t\):
Attempt 1: Dudley’s theorem

GOAL: bound the Gaussian mean width of
\[
\left\{ \begin{pmatrix} pl_1(\Lambda) \\ pl_2(\Lambda) \\ \vdots \\ pl_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}
\]

- Natural choice of nets:

\[
\mathcal{N}_t = \left\{ \begin{pmatrix} pl_1(\Lambda) \\ pl_2(\Lambda) \\ \vdots \\ pl_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}
\]

- Natural way to show that any $\vec{pl}(\Lambda)$ is close to some element of $\mathcal{N}_t$:

Choose a random subset of $\Lambda$ of size $L/2^t$. 

Mary Wootters
Chaining and list decoding
Why should this work

The goal is to show that $\mathcal{N}_t$ is a decent net of $\mathcal{X}$.

(w.r.t. $\ell_2$, where “decent” degrades with $t$)

$$\mathcal{X} = \left\{ \begin{pmatrix} p_l_1(\Lambda) \\ p_l_2(\Lambda) \\ \vdots \\ p_l_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}$$

$$\mathcal{N}_t = \left\{ \begin{pmatrix} p_l_1(\Lambda) \\ p_l_2(\Lambda) \\ \vdots \\ p_l_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$
Why should this work

The goal is to show that $\mathcal{N}_t$ is a decent net of $\mathcal{X}$. (w.r.t. $\ell_2$, where “decent” degrades with $t$)

$$
\mathcal{X} = \left\{ \begin{pmatrix} p_{l_1}(\Lambda) \\ p_{l_2}(\Lambda) \\ \vdots \\ p_{l_n}(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}
$$

$$
\mathcal{N}_t = \left\{ \begin{pmatrix} p_{l_1}(\Lambda) \\ p_{l_2}(\Lambda) \\ \vdots \\ p_{l_n}(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}
$$
Why should this work

The goal is to show that $N_t$ is a decent net of $\mathcal{X}$.
(w.r.t. $\ell_2$, where “decent” degrades with $t$)

$$\mathcal{X} = \left\{ \begin{pmatrix} p_{l_1}(\Lambda) \\ p_{l_2}(\Lambda) \\ \vdots \\ p_{l_n}(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}$$

$$N_t = \left\{ \begin{pmatrix} p_{l_1}(\Lambda) \\ p_{l_2}(\Lambda) \\ \vdots \\ p_{l_n}(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$
Why this shouldn’t work

The goal is to show that $\mathcal{N}_t$ is a decent net of $\mathcal{X}$.
(w.r.t. $\ell_2$, where “decent” degrades with $t$)

$$
\mathcal{X} = \left\{ \begin{pmatrix} pl_1(\Lambda) \\ pl_2(\Lambda) \\ \vdots \\ pl_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\} \\
\mathcal{N}_t = \left\{ \begin{pmatrix} pl_1(\Lambda) \\ pl_2(\Lambda) \\ \vdots \\ pl_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}
$$
Why this shouldn’t work

The goal is to show that $\mathcal{N}_t$ is a decent net of $\mathcal{X}$.

(w.r.t. $\ell_2$, where “decent” degrades with $t$)

\[
\mathcal{X} = \left\{ \begin{pmatrix} p_l_1(\Lambda) \\ p_l_2(\Lambda) \\ \vdots \\ p_l_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}
\]

\[
\mathcal{N}_t = \left\{ \begin{pmatrix} p_l_1(\Lambda) \\ p_l_2(\Lambda) \\ \vdots \\ p_l_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}
\]
Why this shouldn’t work

The goal is to show that $\mathcal{N}_t$ is a decent net of $\mathcal{X}$.

(w.r.t. $\ell_2$, where “decent” degrades with $t$)

$$\mathcal{X} = \left\{ \begin{pmatrix} p_{l_1}(\Lambda) \\ p_{l_2}(\Lambda) \\ \vdots \\ p_{l_n}(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}$$

$$\mathcal{N}_t = \left\{ \begin{pmatrix} p_{l_1}(\Lambda) \\ p_{l_2}(\Lambda) \\ \vdots \\ p_{l_n}(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$
Our great idea to show that $\text{pl}(\Lambda)$ is small is foiled.
Bummer!

Our great idea
to show that $\text{pl}(\Lambda)$ is small
is foiled

by sets $\Lambda$ where $\text{pl}(\Lambda)$ is small.
Solution

Nets are made up of $\text{pl}(\Lambda)$, along with a set of active indices

 Guarantees:
▶ The active $\text{pl}_j(\Lambda)$ never get too large.
▶ The distances between these vectors are not too big.
Solution
Nets are made up of $pl(\Lambda)$, along with a set of active indices

Guarantees:
▶ The active $pl_j(\Lambda)$ never get too large.
▶ The distances between these vectors are not too big.
Solution

Nets are made up of \( pl(\Lambda) \), along with a set of active indices.

 Guarantees:

▶ The active \( pl_j(\Lambda) \) never get too large.
▶ The distances between these vectors are not too big.
Solution

Nets are made up of $p_l(\Lambda)$, along with a set of active indices.
Solution

Nets are made up of pl(Λ), along with a set of active indices

Guarantees:
▶ The active pl(Λ) never get too large.
▶ The distances between these vectors are not too big.
Solution

Nets are made up of $pl(\Lambda)$, along with a set of active indices.

Guarantees:

- The active $pl_j(\Lambda)$ never get too large.
- The distances between these vectors are not too big.
Solution
Nets are made up of $\text{pl}(\Lambda)$, along with a set of active indices

Guarantees:
- The active $\text{pl}_j(\Lambda)$ never get too large.
- The distances between these vectors are not too big.
(Slightly) more precisely
(Slightly) more precisely

- Set Λ₀ of size L. Choose ℓ₀ = [n].
(Slightly) more precisely

- Set $\Lambda_0$ of size $L$. Choose $I_0 = [n]$.
- There exists a chain $(\Lambda_0, I_0) \leftrightarrow (\Lambda_1, I_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, I_T)$ so that:
(Slightly) more precisely

- Set $\Lambda_0$ of size $L$. Choose $I_0 = [n]$.
- There exists a chain $(\Lambda_0, I_0) \leftrightarrow (\Lambda_1, I_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, I_T)$ so that:
  \[ \| p_{I_t}(\Lambda_t) - p_{I_{t+1}}(\Lambda_{t+1}) \|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} p_{j}(\Lambda_0)}. \]
(Slightly) more precisely

- Set $\Lambda_0$ of size $L$. Choose $I_0 = [n]$.
- There exists a chain $(\Lambda_0, I_0) \leftrightarrow (\Lambda_1, I_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, I_T)$ so that:
  
  \[
  \| \text{pl}_{I_t}(\Lambda_t) - \text{pl}_{I_{t+1}}(\Lambda_{t+1}) \|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} \text{pl}_j(\Lambda_0)}.\]

\[
\sum_{j \in I_t} \text{pl}_j(\Lambda_t) \leq (1 + \eta)^t \left( \sum_{j=1}^{n} \text{pl}_j(\Lambda_0) \right).
\]
(Slightly) more precisely

- Set \( \Lambda_0 \) of size \( L \). Choose \( i_0 = [n] \).
- There exists a chain \( (\Lambda_0, i_0) \leftrightarrow (\Lambda_1, i_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, i_T) \) so that:

\[
\|p_{i_t}(\Lambda_t) - p_{i_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{\|\Lambda_t\|}} \cdot \sqrt{\sum_{j=1}^n p_j(\Lambda_0)}.
\]

\[
\sum_{j \in i_t} p_j(\Lambda_t) \lesssim \left( \sum_{j=1}^n p_j(\Lambda_0) \right)
\]
(Slightly) more precisely

- Set $\Lambda_0$ of size $L$. Choose $I_0 = \lceil n \rceil$.
- There exists a chain $(\Lambda_0, I_0) \leftrightarrow (\Lambda_1, I_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, I_T)$ so that:

\[
\|p_l^t(\Lambda_t) - p_l^{t+1}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^n p_l^j(\Lambda_0)}.
\]

\[
\sum_{j \in I_t} p_l^j(\Lambda_t) \lesssim \left( \sum_{j=1}^n p_l^j(\Lambda_0) \right)^{1/2}.
\]

- Over all of the $\Lambda_0$'s we could have started with, the number of pairs $(\Lambda_t, I_t)$ that ever show up at level $t$ is at most $\left( \frac{|C|}{L/2^t} \right)^2$. 
(Slightly) more precisely

- Set $\Lambda_0$ of size $L$. Choose $I_0 = [n]$.
- There exists a chain $(\Lambda_0, I_0) \leftrightarrow (\Lambda_1, I_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, I_T)$ so that:
  - $\|p_{l_t}(\Lambda_t) - p_{l_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} p_l(\Lambda_0)}$.
  - $\sum_{j \in I_t} p_l(\Lambda_t) \lesssim \left( \sum_{j=1}^{n} p_l(\Lambda_0) \right)$
- Over all of the $\Lambda_0$'s we could have started with, the number of pairs $(\Lambda_t, I_t)$ that ever show up at level $t$ is at most $(\frac{|C|}{L/2^t})^2$.
(Slightly) more precisely

- Set $\Lambda_0$ of size $L$. Choose $I_0 = [n]$.
- There exists a chain $(\Lambda_0, I_0) \leftrightarrow (\Lambda_1, I_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, I_T)$ so that:
  \[ \|p_{l_t}(\Lambda_t) - p_{l_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} p_{l_j}(\Lambda_0)}. \]

  \[ \sum_{j \in I_t} p_{l_j}(\Lambda_t) \lesssim \left( \sum_{j=1}^{n} p_{l_j}(\Lambda_0) \right) \]

- Over all of the $\Lambda_0$'s we could have started with, the number of pairs $(\Lambda_t, I_t)$ that ever show up at level $t$ is at most $(|C|L/2^t)^2$.

\[ \sum_j g_j p_{l_j}(\Lambda_0) \approx \sum_{j \in I_T} g_j p_{l_j}(\Lambda_T) \]
(Slightly) more precisely

- Set $\Lambda_0$ of size $L$. Choose $l_0 = [n]$.
- There exists a chain $(\Lambda_0, l_0) \leftrightarrow (\Lambda_1, l_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, l_T)$ so that:

$$
\|p_{l_t}(\Lambda_t) - p_{l_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} p_l(\Lambda_0)}.
$$

$$
\sum_{j \in l_t} p_l(\Lambda_t) \lesssim \left( \sum_{j=1}^{n} p_l(\Lambda_0) \right).
$$

- Over all of the $\Lambda_0$’s we could have started with, the number of pairs $(\Lambda_t, l_t)$ that ever show up at level $t$ is at most $(\frac{|C|}{L/2^t})^2$.

$$
\sum_{j} g_j p_l(\Lambda_0) \approx \sum_{j \in l_T} g_j p_l(\Lambda_T)
$$

$$
\sum_{j \in l_T} g_j p_l(\Lambda_T) \text{ reasonable}
$$
(Slightly) more precisely

- Set $\Lambda_0$ of size $L$. Choose $I_0 = [n]$.
- There exists a chain $(\Lambda_0, I_0) \leftrightarrow (\Lambda_1, I_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, I_T)$ so that:
  - $\|p_{I_t}(\Lambda_t) - p_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sum_{j=1}^{n} pl_j(\Lambda_0)$.
  - $\sum_{j \in I_t} pl_j(\Lambda_t) \lesssim \left( \sum_{j=1}^{n} pl_j(\Lambda_0) \right)^2$.
  - $\sum_{j} g_j pl_j(\Lambda_0) \approx \sum_{j \in I_T} g_j pl_j(\Lambda_T)$ reasonable

Over all of the $\Lambda_0$'s we could have started with, the number of pairs $(\Lambda_t, I_t)$ that ever show up at level $t$ is at most $\left( \frac{|C|}{L/2^t} \right)^2$. 

Call this $Q$ (this is what we originally wanted to bound).
Chaining argument

\[ \sum_{j} g_{j} p_{j}(\Lambda_{0}) \approx \sum_{j \in I_{T}} g_{j} p_{j}(\Lambda_{T}) \]

\[ \| p_{l_{t}}(\Lambda_{t}) - p_{l_{t+1}}(\Lambda_{t+1}) \|_{2} \lesssim \frac{1}{\sqrt{|\Lambda_{t}|}} \cdot \sqrt{Q}. \]

➤ Over all of the \( \Lambda_{0} \)'s we could have started with, the number of pairs (\( \Lambda_{t}, l_{t} \)) that ever show up at level \( t \) is at most \( \left( \frac{|C|}{L/2^{t}} \right)^{2} \).
Chaining argument

\[
\sum_{j} g_j p_{l_j}(\Lambda_0) \approx \sum_{j \in I_T} g_j p_{l_j}(\Lambda_T)
\]

\[
\|p_{l_t}(\Lambda_t) - p_{l_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{Q}.
\]

Over all of the \(\Lambda_0\)'s we could have started with, the number of pairs \((\Lambda_t, l_t)\) that ever show up at level \(t\) is at most \(\left(\frac{|C|}{L/2^t}\right)^2\).

This implies (on board)

\[
\mathbb{E} \max_{|\Lambda_0|=L} \left| \sum_{j=1}^{n} g_j p_{l_j}(\Lambda_0) - \sum_{j \in I_T} g_j p_{l_j}(\Lambda_T) \right| \lesssim \sqrt{Q \log |C|}.
\]
Base

\[
\sum_{j \in I_T} g_j p_l(j(\Lambda_T)) \text{ reasonable}
\]

\[
\sum_{j \in I_T} p_l(j(\Lambda_T)) \lesssim Q \quad \text{and} \quad p_l(j(\Lambda_T)) \leq 1 \quad \forall j
\]
Base

\[
\sum_{j \in I_T} p_l_j(\Lambda_T) \lesssim Q \quad \text{and} \quad p_l_j(\Lambda_T) \leq 1 \quad \forall j
\]

This implies (in your head)

\[
\mathbb{E}_{\Lambda_0} \left| \sum_{j \in I_T} g_j p_l_j(\Lambda_T) \right| \lesssim \sqrt{Q}
\]
Together

\[
\mathbb{E} \max_{|\Lambda_0| = \mathcal{L}} \sum_{j=1}^{n} g_j \cdot pl_j(\Lambda_0) \lesssim \sqrt{Q \log |C|} = \sqrt{\left( \sum_{j=1}^{n} pl_j(\Lambda) \right) \log |C|}
\]
Together

\[ \mathbb{E} \max_{|\Lambda_0|=L} \sum_{j=1}^{n} g_j \cdot pl_j(\Lambda_0) \lessgtr \sqrt{Q \log |C|} = \sqrt{\left( \sum_{j=1}^{n} pl_j(\Lambda) \right) \log |C|} \]

Why did we want this again?
Together

\[ E \max_{|\Lambda_0|=L} \sum_{j=1}^{n} g_j \cdot pl_j(\Lambda_0) \lesssim \sqrt{Q \log |C|} = \sqrt{\left( \sum_{j=1}^{n} pl_j(\Lambda) \right) \log |C|} \]

Why did we want this again?

\[ E_C \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} pl_j(\Lambda) - E \sum_{j=1}^{n} pl_j(\Lambda) \right| \lesssim E_C \left[ E_g \max_{|\Lambda|=L} \sum_{j=1}^{n} g_j pl_j(\Lambda) \right] \]
Together

$$\mathbb{E} \max_{|\Lambda_0|=L} \sum_{j=1}^{n} g_j \cdot p_{l_j}(\Lambda_0) \lesssim \sqrt{Q \log |C|} = \sqrt{\left( \sum_{j=1}^{n} p_{l_j}(\Lambda) \right) \log |C|}$$

Why did we want this again?

$$\mathbb{E}_c \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} p_{l_j}(\Lambda) - \mathbb{E} \sum_{j=1}^{n} p_{l_j}(\Lambda) \right| \lesssim \mathbb{E}_c \left[ \mathbb{E}_g \max_{|\Lambda|=L} \sum_{j=1}^{n} g_j p_{l_j}(\Lambda) \right] \lesssim \mathbb{E}_c \left( \sum_{j=1}^{n} p_{l_j}(\Lambda) \right) \log |C|$$
Together

\[
\mathbb{E} \max_{|\Lambda_0| = L} \sum_{j=1}^{n} g_j \cdot \text{pl}_j(\Lambda_0) \lesssim \sqrt{Q \log |C|} = \sqrt{\left( \sum_{j=1}^{n} \text{pl}_j(\Lambda) \right) \log |C|}
\]

Why did we want this again?

\[
\mathbb{E}_C \max_{|\Lambda| = L} \left| \sum_{j=1}^{n} \text{pl}_j(\Lambda) - \mathbb{E} \sum_{j=1}^{n} \text{pl}_j(\Lambda) \right| \lesssim \mathbb{E}_C \left[ \mathbb{E}_g \max_{|\Lambda| = L} \sum_{j=1}^{n} g_j \text{pl}_j(\Lambda) \right] \lesssim \mathbb{E}_C \left( \sum_{j=1}^{n} \text{pl}_j(\Lambda) \right) \log |C| \]

Solving… (and using the fact that \( \max \mathbb{E} \) is fine):

\[
\mathbb{E} \max_{|\Lambda| = L} \sum_{j=1}^{n} \text{pl}_j(\Lambda) \lesssim n\varepsilon + \log |C|.
\]
Almost done

We just saw

\[ \mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} \text{pl}_j(\Lambda) \lesssim n\varepsilon + \log |C|. \]
Almost done

We just saw

\[ \mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} \text{pl}_j(\Lambda) \lesssim n\epsilon + \log |C|. \]

We’d like

\[ \log |C| \leq n\epsilon \]
Almost done

We just saw

\[ \mathbb{E} \max_{|\Lambda| = L} \sum_{j=1}^{n} p_{l_j}(\Lambda) \lesssim n\varepsilon + \log |C|. \]

We'd like

\[ \log |C| \leq n\varepsilon \]
Almost done

We just saw

\[ \mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} p(l_j(\Lambda)) \lesssim n\varepsilon + \log |C|. \]

We’d like

\[ \log |C| \leq n\varepsilon \]

aka

\[ R = \frac{\log_q |C|}{n} \lesssim \frac{\varepsilon}{\log(q)}. \]
Finally

**Theorem**

Suppose \( q \geq \frac{k}{\epsilon^2} \). Let \( \mathcal{C} \) be a random linear code over \( \mathbb{F}_q \) with

\[
R = \frac{k}{n} = \frac{C\epsilon}{\log(q) \log^5(1/\epsilon)}.
\]

Then w.h.p, \( \mathcal{C} \) is \((\rho, L)\)-list-decodable with

- error rate \( \rho = 1 - \epsilon \)
- list size \( L = O\left(\frac{1}{\epsilon}\right) \).
Finally

Theorem

Suppose \( q \geq \frac{k}{\varepsilon^2} \). Let \( C \) be a random linear code over \( \mathbb{F}_q \) with

\[
R = \frac{k}{n} = \frac{C\varepsilon}{\log(q) \log^5(1/\varepsilon)}.
\]

Then w.h.p, \( C \) is \((\rho, L)\)-list-decodable with

error rate \( \rho = 1 - \varepsilon \)    list size \( L = O \left( \frac{1}{\varepsilon} \right) \).
Finally

Theorem

Suppose $q \geq \frac{k}{\epsilon^2}$. Let $C$ be a random linear code over $\mathbb{F}_q$ with

$$R = \frac{k}{n} = \frac{C\epsilon}{\log(q) \log^5(1/\epsilon)}.$$ 

Then w.h.p, $C$ is $(\rho, L)$-list-decodable with

error rate $\rho = 1 - \epsilon$ \quad list size $L = O\left(\frac{1}{\epsilon}\right)$. 

What did we need from random linear codes?
Finally

Theorem

Suppose $q \geq \frac{k}{\varepsilon^2}$. Let $C$ be a random linear code over $\mathbb{F}_q$ with

$$R = \frac{k}{n} = \frac{C\varepsilon}{\log(q) \log^5(1/\varepsilon)}.$$ 

Then w.h.p, $C$ is $(\rho, L)$-list-decodable with

error rate $\rho = 1 - \varepsilon$ \quad list size $L = O\left(\frac{1}{\varepsilon}\right).$

What did we need from random linear codes?

- Independent symbols.
- max $\mathbb{E}$ about right.
Finally

**Theorem**

Suppose \( q \geq \frac{k}{\epsilon^2} \). Let \( C \) be a random linear code over \( \mathbb{F}_q \) with

\[
R = \frac{k}{n} = \frac{C\epsilon}{\log(q) \log^5(1/\epsilon)}.
\]

Then w.h.p, \( C \) is \((\rho, L)\)-list-decodable with

- error rate \( \rho = 1 - \epsilon \)
- list size \( L = O\left(\frac{1}{\epsilon}\right) \).

- What did we need from random linear codes?
  - Independent symbols.
  - \( \max \mathbb{E} \) about right.
- Also works for Reed-Solomon Codes with random evaluation points.
1. Error Correcting Codes and List Decoding

2. Random Codes

3. Setting up a Chaining Argument
   - Average-radius list-decodability
   - Pass to a Gaussian process
   - Controlling the Gaussian process

4. Conclusion
Chaining method for establishing **list-decodability** of structured random codes.
Recap

- Chaining method for establishing list-decodability of structured random codes.

- Some punchlines:
Recap

- Chaining method for establishing list-decodability of structured random codes.

- Some punchlines:
  - Random linear codes are (nearly) optimally list-decodable.
Recap

- Chaining method for establishing list-decodability of structured random codes.

- Some punchlines:
  - Random linear codes are (nearly) optimally list-decodable.
  - There are Reed-Solomon codes list-decodable beyond the Johnson bound.
Recap

- Chaining method for establishing **list-decodability** of structured random codes.

- Some punchlines:
  - Random linear codes are (nearly) optimally list-decodable.
  - There are Reed-Solomon codes list-decodable beyond the Johnson bound.
  - More generally, technique works for anything with independent symbols and reasonable max $\mathbb{E}$. 

Mary Wootters

Chaining and list decoding

39 / 41
Many questions

- Remove superfluous log factors?
  the log\((q)\) is especially obnoxious

- Related: find a simpler proof?
  I mean, I like this one, but...

- Nice characterization of codes list-decodable to capacity?
  Or a nice sufficient condition?

- Applications to other pseudorandom objects?
  Is a random linear extractor optimal?
The end

Thanks for listening!