Chaining arguments and list decoding

Mary Wootters (based on work with Atri Rudra)

June 23, 2016

1. Error Correcting Codes and List Decoding

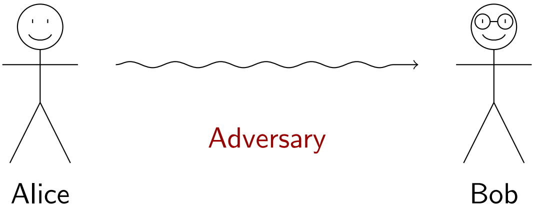
2. Random Codes

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

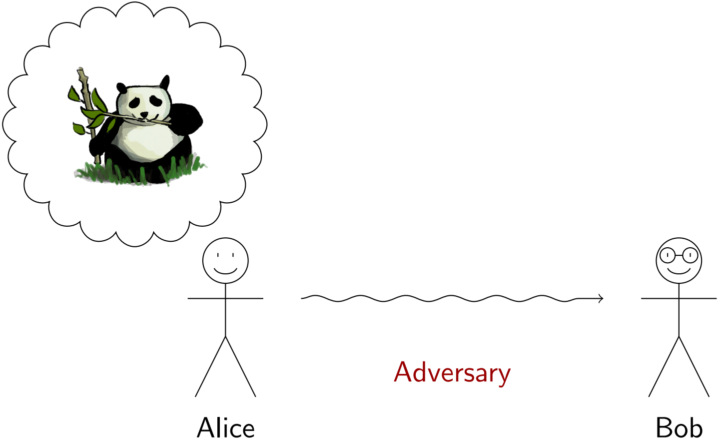
4. Conclusion

The point of this talk is the chaining argument

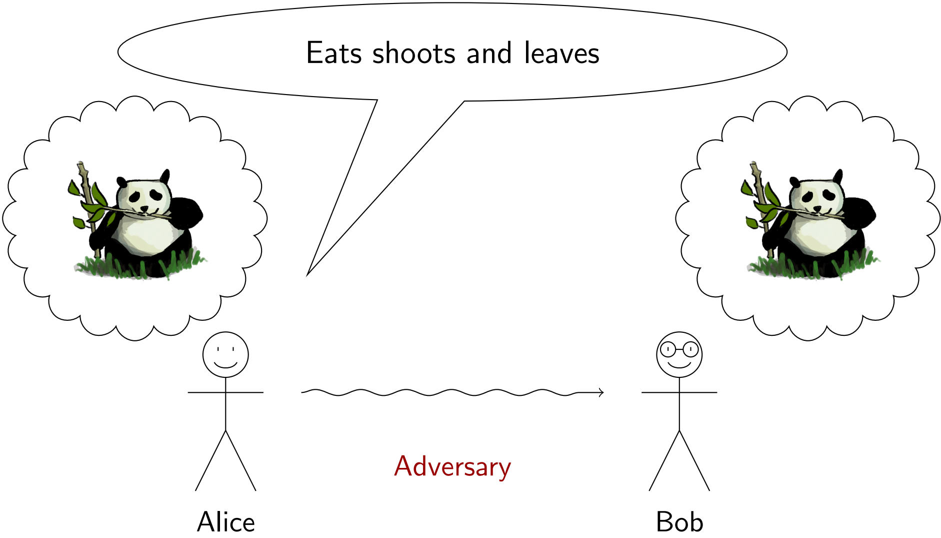
Error correcting codes



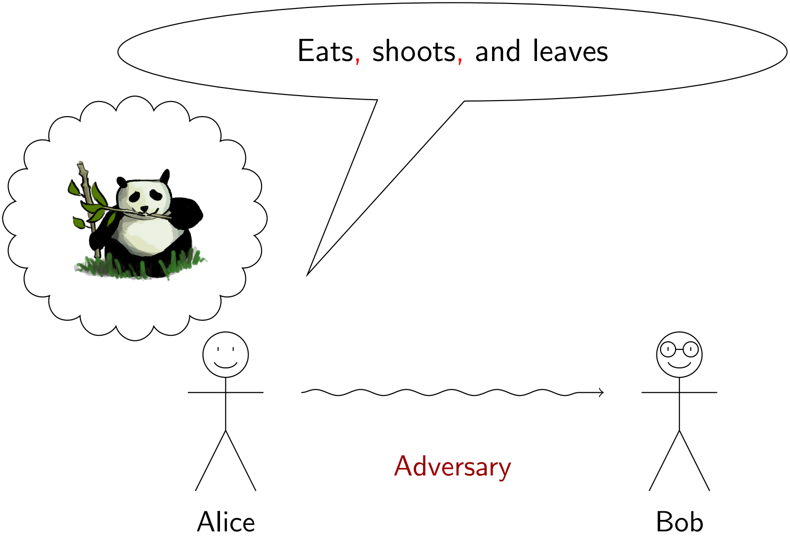
Error correcting codes



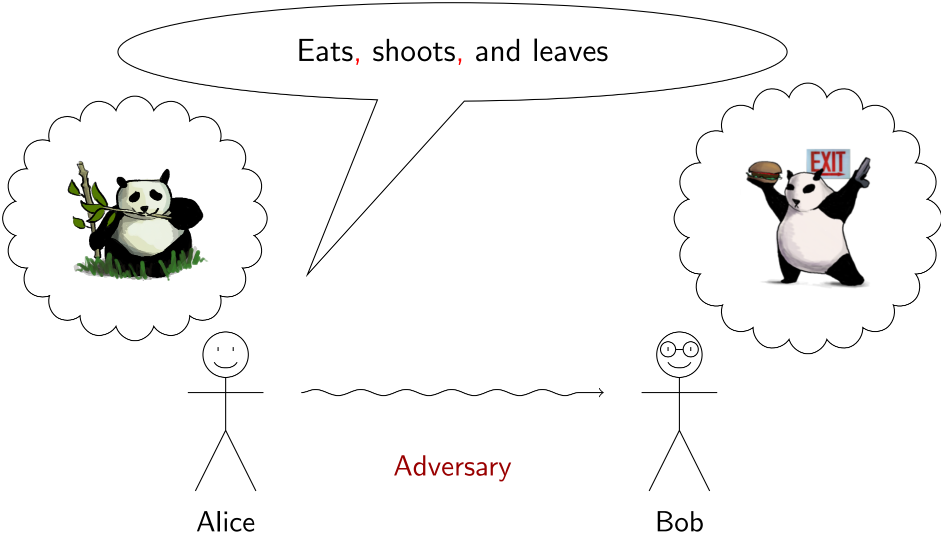
Error correcting codes



Error correcting codes

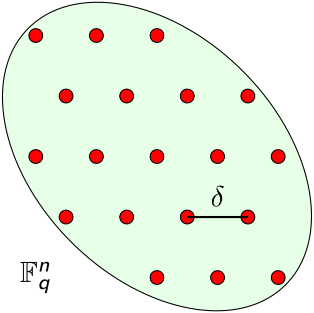


Error correcting codes



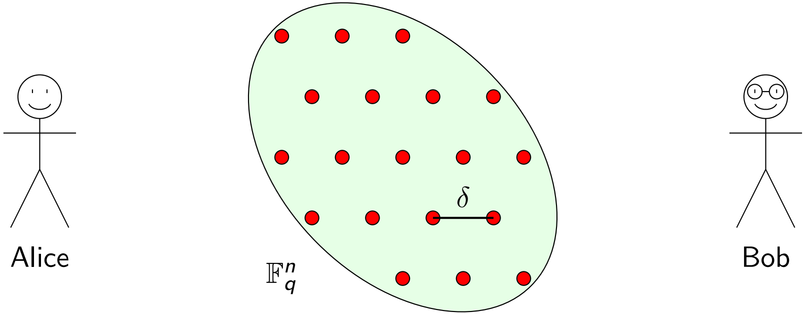
Error correcting codes

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



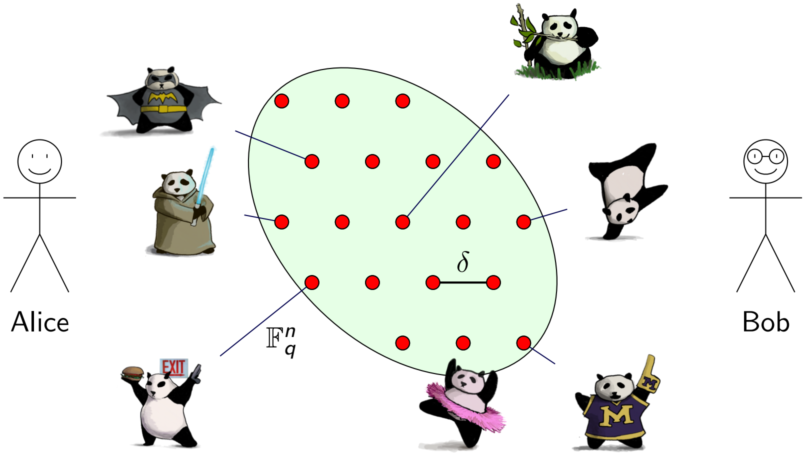
Error correcting codes

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



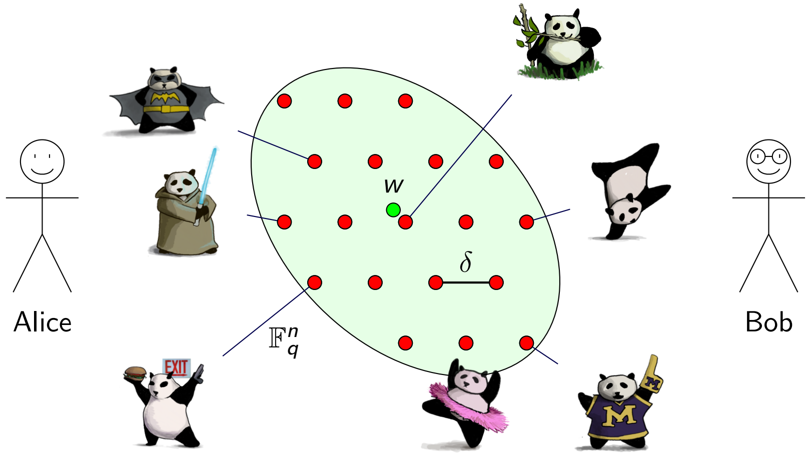
Error correcting codes

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



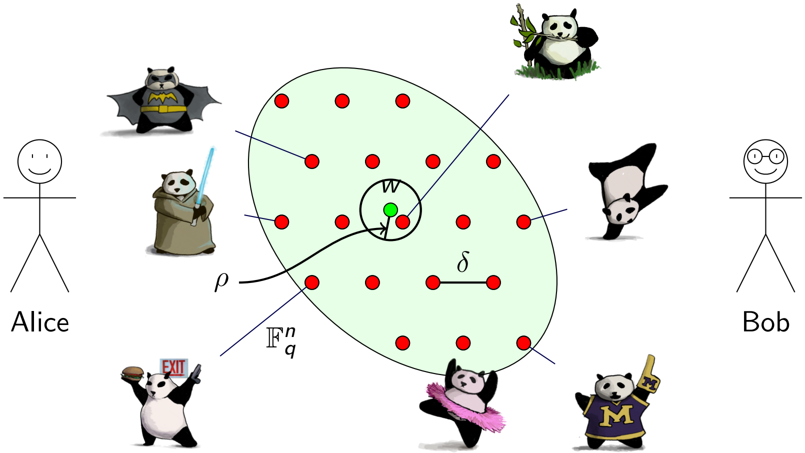
Error correcting codes

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



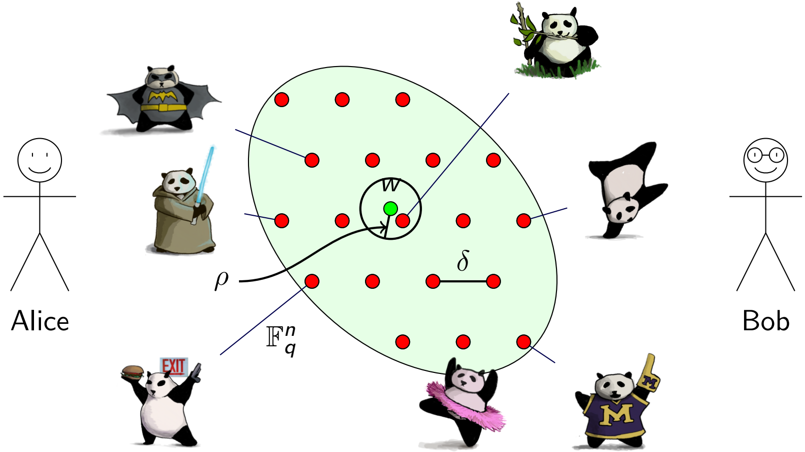
Error correcting codes

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



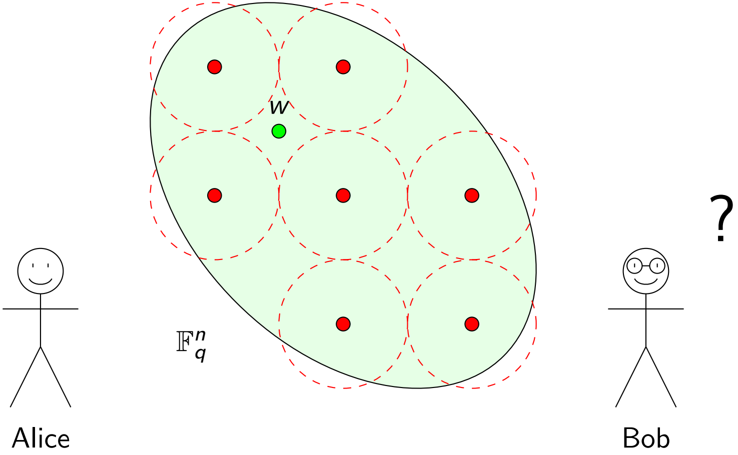
Error correcting codes

- $\mathcal{C} \subset \mathbb{F}_q^n$ is a code, with (relative Hamming) distance δ.
- Alice and Bob can use $\mathcal{C}$ to communicate:
  - error rate is ρ
  - The rate of $\mathcal{C}$ is $R = \log_q |\mathcal{C}|/n$.



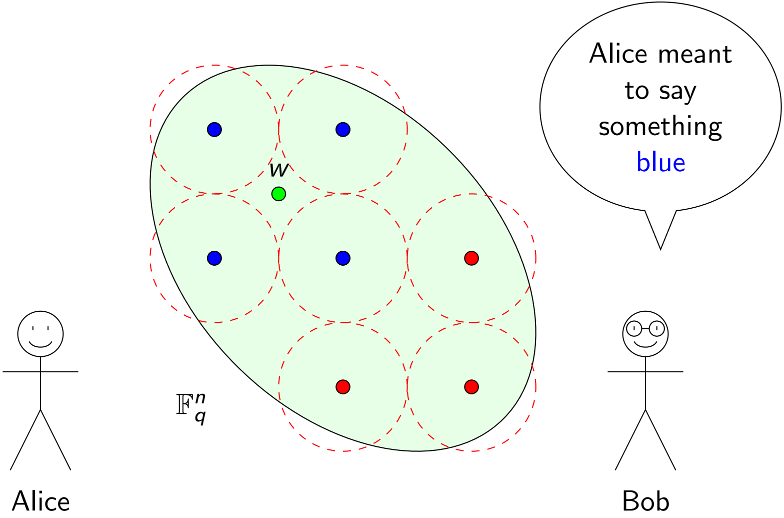
But what if the error rate $\rho > \delta/2$?

- 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!

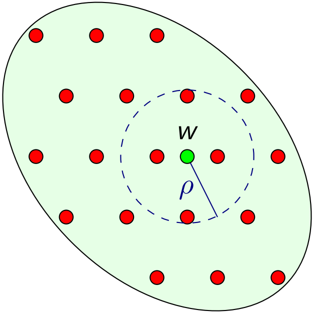


List decodable codes

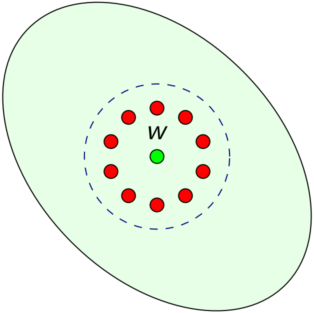
Definition

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

$$\max_{w \in \mathbb{F}_q^n} \left| \{ c \in \mathcal{C} \,:\, d(c, w) \leq \rho \} \right| \leq L.$$



$(\rho, 4)$-list-decodable



not $(\rho, 4)$-list-decodable

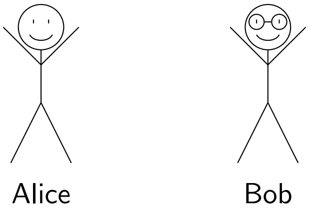
Now ρ can be big!

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

Now ρ can be big!

- Turns out (for large q), there are codes which are (ρ, L)-list decododable 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!



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? (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?

Questions

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

2. Where's the randomness here?

Questions

1. What codes are (near-)optimally list-decodable? (up to $\rho = 1 - \varepsilon$)
   - Completely random codes
   - A few special deterministic codes [Guruswami-Rudra'08...]
2. Where's the randomness here?

Questions

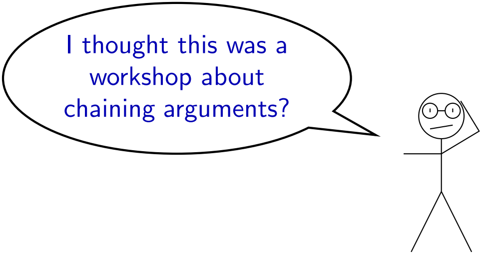
1. What codes are (near-)optimally list-decodable? (up to $\rho = 1 - \varepsilon$)
   - Completely random codes
   - Structured random codes
   - A few special deterministic codes [Guruswami-Rudra'08...]

2. Where's the randomness here?

Questions

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

2. Where's the randomness here?



Bob

Structured random codes

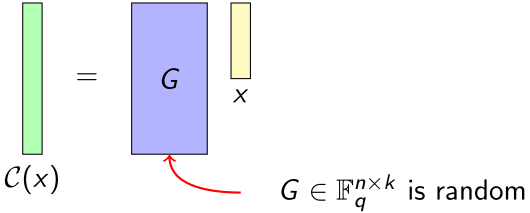
- Headline result: random Reed-Solomon Codes

Structured random codes

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

Structured random codes

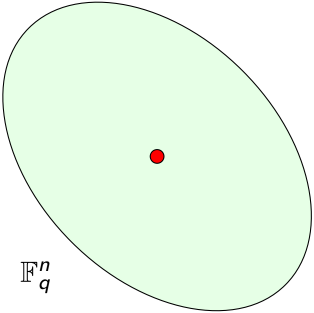
- Headline result: random Reed-Solomon Codes
- This talk: random linear codes.



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

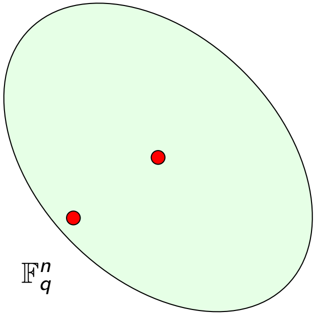
But first! Completely random codes

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



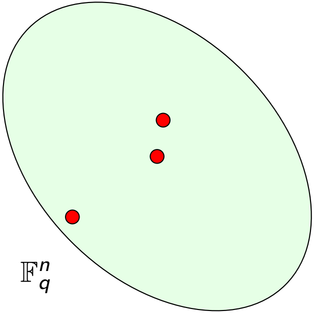
But first! Completely random codes

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



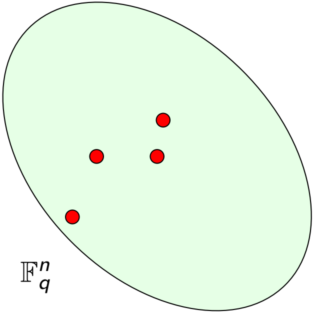
But first! Completely random codes

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



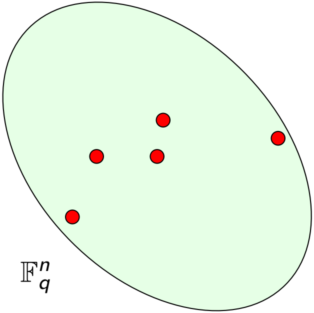
But first! Completely random codes

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



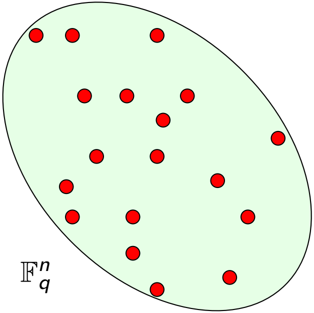
But first! Completely random codes

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



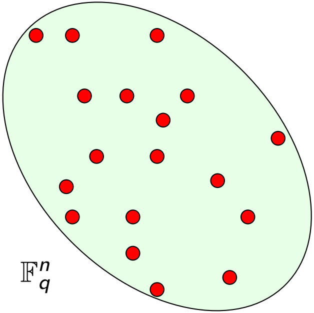
But first! Completely random codes

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



But first! Completely random codes

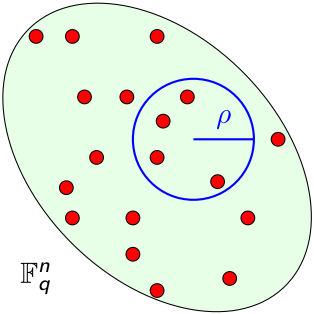
- Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a completely random code.



- Choose error rate $\rho = 1 - \varepsilon$.
- If the rate $R = \log_q |\mathcal{C}|/n$ is about ε, then $\mathcal{C}$ is $(\rho, 1/\varepsilon)$-list-decodable.

But first! Completely random codes

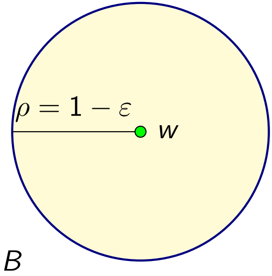
- Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a completely random code.



- Choose error rate $\rho = 1 - \varepsilon$.
- If the rate $R = \log_q |\mathcal{C}|/n$ is about ε, then $\mathcal{C}$ is $(\rho, 1/\varepsilon)$-list-decodable.

A random code is list-decodable to capacity

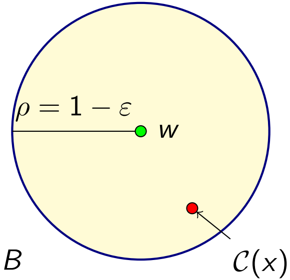
- Fix $w \in \mathbb{F}_q^n$.



A random code is list-decodable to capacity

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

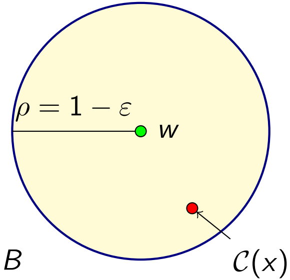
$$\mathbb{P}\left\{\mathcal{C}(x) \in B\right\} \leq \text{small} \approx q^{-\varepsilon n}$$



A random code is list-decodable to capacity

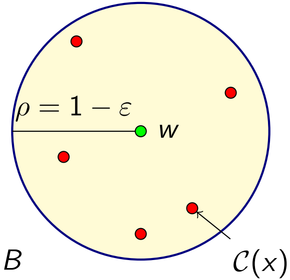
- Fix $w \in \mathbb{F}_q^n$.

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



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

A random code is list-decodable to capacity



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

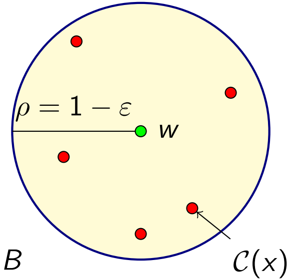
$$\mathbb{P}\left\{\mathcal{C}(x) \in B\right\} \leq \text{small} \approx q^{-\varepsilon n}$$

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

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

A random code is list-decodable to capacity

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



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

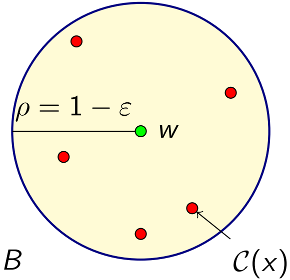
$$\mathbb{P}\left\{\mathcal{C}(x) \in B\right\} \leq \text{small} \approx q^{-\varepsilon n}$$

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

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

A random code is list-decodable to capacity

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



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

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

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

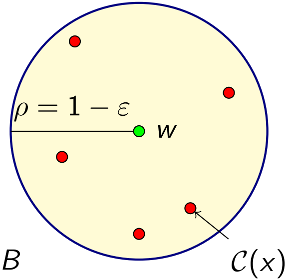
$$\mathbb{P}\left\{\mathcal{C}(x_0), \ldots, \mathcal{C}(x_L) \in B\right\} \leq \text{small}^{L+1}$$

- Union bound over all w, Λ:

$$\mathbb{P}\left\{\exists w, \Lambda, \text{bad event }\right\} \leq q^n \binom{q^k}{L+1} \text{small}^{L+1}$$
$$\approx q^{n+kL-n\varepsilon L}$$

A random code is list-decodable to capacity

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



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

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

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

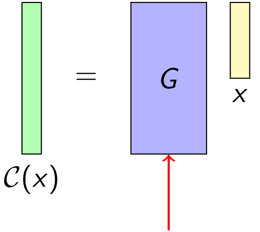
$$\mathbb{P}\{\mathcal{C}(x_0), \ldots, \mathcal{C}(x_L) \in B\} \leq \text{small}^{L+1}$$

- Union bound over all w, Λ:

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

If $L \approx 1/\varepsilon$ and $k/n > \varepsilon$, then this is small!

Try it for random linear codes

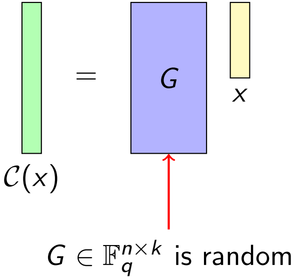


$G \in \mathbb{F}_q^{n \times k}$ is random

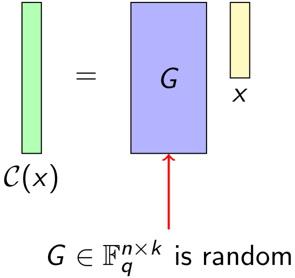
Try it for random linear codes

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

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



Try it for random linear codes

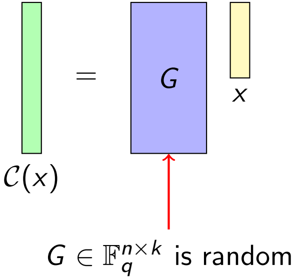


- Fix $w \in \mathbb{F}_q^n$.

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

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

Try it for random linear codes



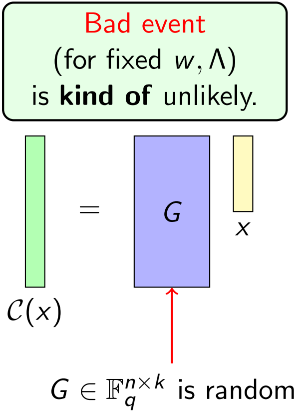
- Fix $w \in \mathbb{F}_q^n$.

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

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

$$\mathbb{P}\left\{\mathcal{C}(x_0), \ldots, \mathcal{C}(x_L) \in B\right\} \le \text{small}^{\log_q(L+1)}$$

Try it for random linear codes



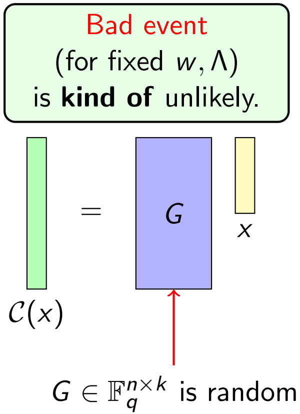
- Fix $w \in \mathbb{F}_q^n$.

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

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

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

Try it for random linear codes



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

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

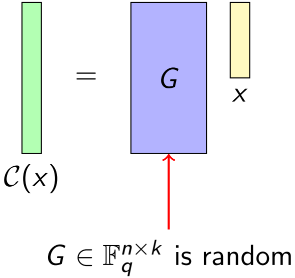
- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.

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

- Union bound over all w, Λ...

Try it for random linear codes

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



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

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

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

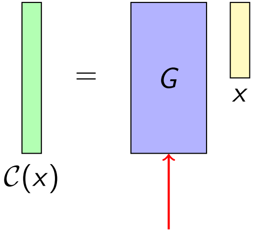
$$\mathbb{P}\{\mathcal{C}(x_0), \ldots, \mathcal{C}(x_L) \in B\} \leq \text{small}^{\log_q(L+1)}$$

- Union bound over all w, Λ...

  ...need $L \approx q^{1/\varepsilon}$ to get away with rate $R \approx \varepsilon$

Try it for random linear codes

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



$G \in \mathbb{F}_q^{n \times k}$ is random

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

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

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

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

- Union bound over all w, Λ...

  ...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 ρ.
  [Guruswami, Hastad, Kopparty'11]
- small q, large ρ.
  [Cheraghchi, Guruswami, Velingker'13],[W.'13]
- large q, large ρ.
  [Rudra-W.'14]

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

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

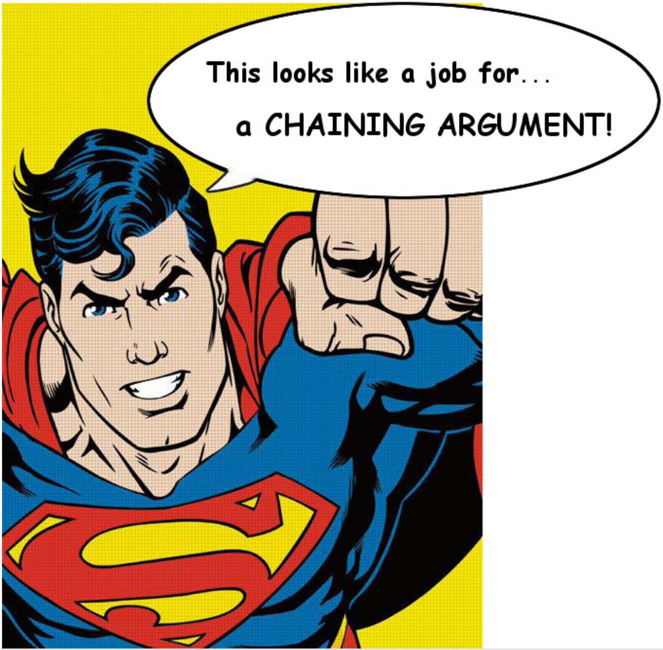
- Fix messages $\Lambda = \{x_0, \ldots, x_L\} \subset \mathbb{F}_q^k$.

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

- Union bound over all w, Λ...

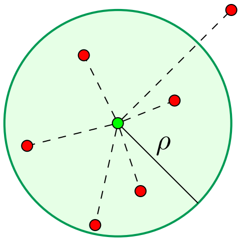
  ...need $L \approx q^{1/\varepsilon}$ to get away with rate $R \approx \varepsilon$

Question:
Are random linear codes as list-decodable as random codes?


This looks like a job for...
a CHAINING ARGUMENT!

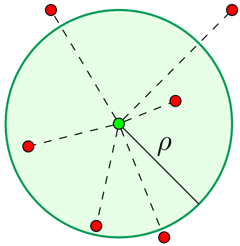
Average-Radius List Decodability

List Decodable



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

Average-Radius List Decodable

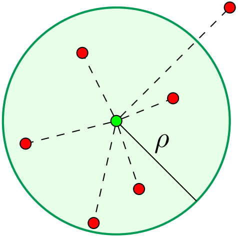


All lists of size $> L$ have average distance larger than ρ

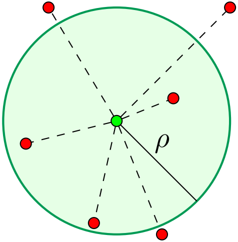
Average-Radius List Decodability

List Decodable

Average-Radius List Decodable



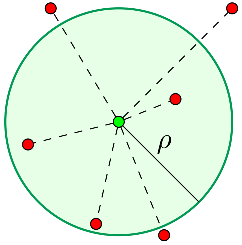
$\Longleftarrow$



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

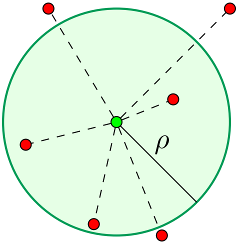
All lists of size $> L$ have average distance larger than ρ

Average-Radius List Decodability



All lists of size $> L$ have average distance larger than ρ.

Average-Radius List Decodability

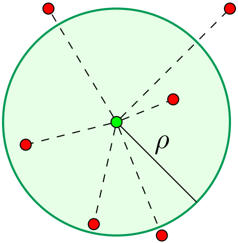


All lists of size $> L$ have average distance larger than ρ.

$$\Leftrightarrow$$

$$\max_{\Lambda \subset \mathbb{F}_q^k, |\Lambda| = L+1} \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \leq n \cdot (L+1) \cdot (1-\rho)$$

Average-Radius List Decodability

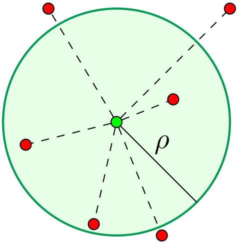


All lists of size $> L$ have average distance larger than ρ.

$$\Leftrightarrow$$

$$\max_{\Lambda \subset \mathbb{F}_q^k, |\Lambda| = L+1} \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \leq n \cdot (L+1) \cdot (1 - \rho)$$

Average-Radius List Decodability

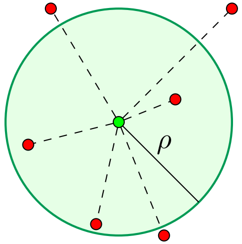


All lists of size $> L$ have average distance larger than ρ.

$$\Leftrightarrow$$

$$\max_{\Lambda \subset \mathbb{F}_q^k, |\Lambda| = L+1} \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \leq n \cdot (L+1) \cdot (1 - \rho)$$

Average-Radius List Decodability

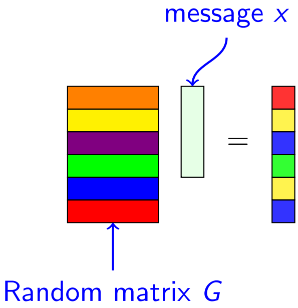


All lists of size $> L$ have average distance larger than ρ.

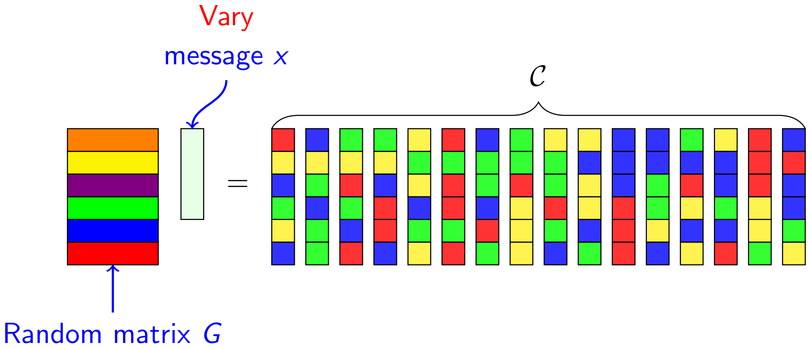
$$\Leftrightarrow$$

$$\max_{\Lambda \subset \mathbb{F}_q^k, |\Lambda| = L+1} \max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \leq n \cdot (L+1) \cdot (1 - \rho)$$

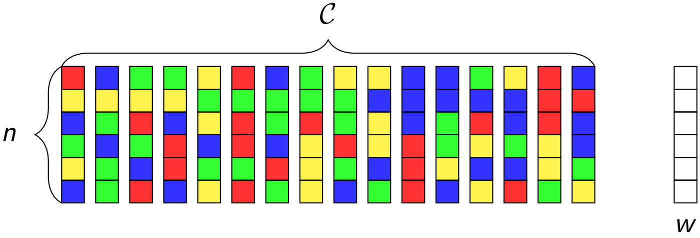
$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



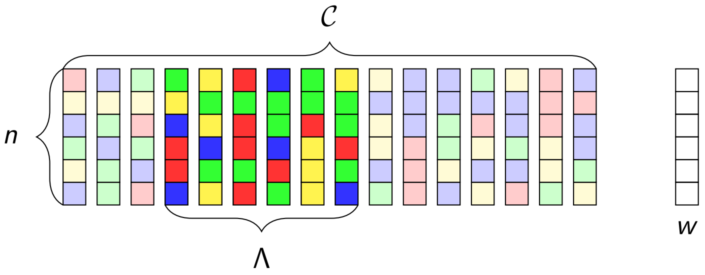
$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \quad \text{for a random linear code}$$



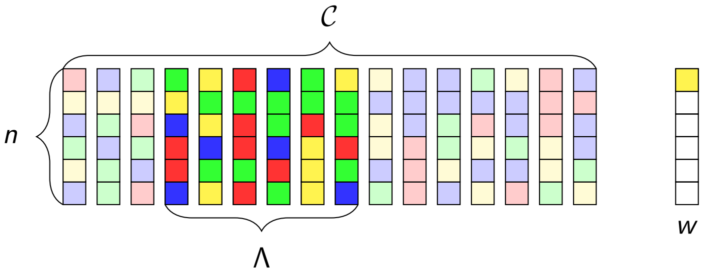
$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



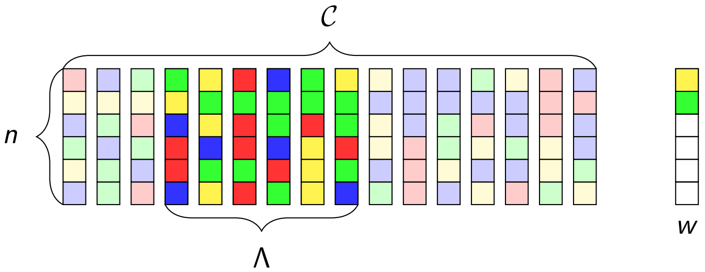
$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



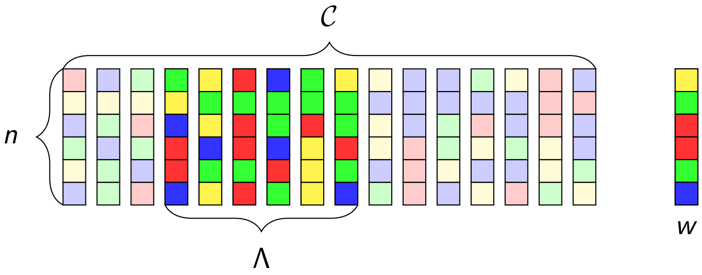
$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



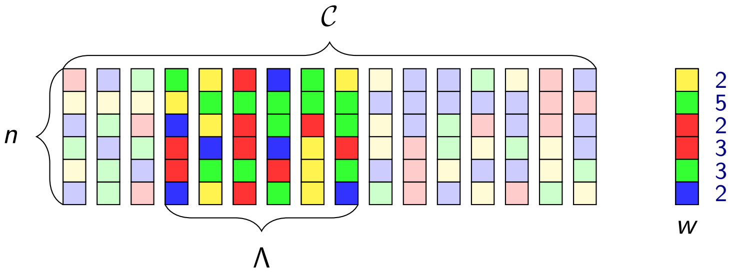
$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



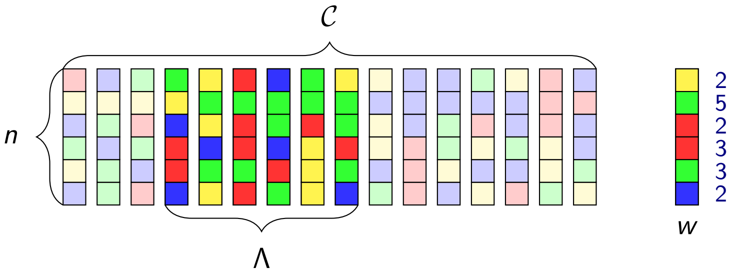
$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



$$\max_{w\in\mathbb{F}_q^n} \sum_{x\in\Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



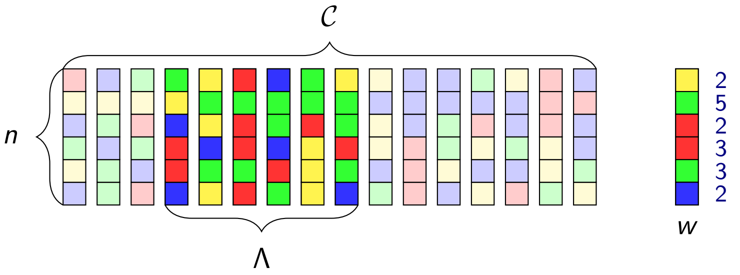
$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



Average agreement of Λ with the worst w:

$$\max_{w \in F_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) = 2 + 5 + 2 + 3 + 3 + 2 \qquad .$$

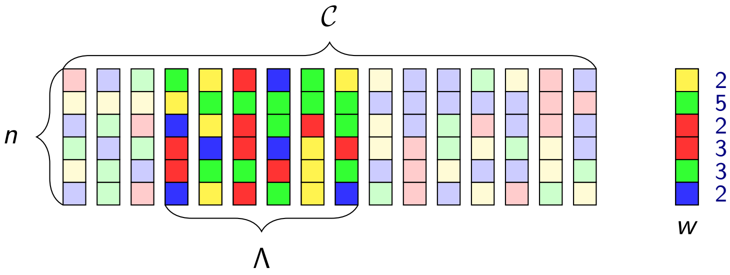
$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



Average agreement of Λ with the worst w:

$$\max_{w \in F_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) = 2 + 5 + 2 + 3 + 3 + 2 =: |\Lambda| \cdot \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda).$$

$$\max_{w \in \mathbb{F}_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) \text{ for a random linear code}$$



Average agreement of Λ with the worst w:

Independent!

$$\max_{w \in F_q^n} \sum_{x \in \Lambda} \mathrm{agr}(w, \mathcal{C}(x)) = 2 + 5 + 2 + 3 + 3 + 2 =: |\Lambda| \cdot \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda).$$

Yay! A sum of independent random variables

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

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

Yay! A sum of independent random variables

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

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

As usual:

- Show

$$\max_{\Lambda} \mathbb{E} \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) \leq \text{ small}$$

- Show

$$\mathbb{E} \max_{\Lambda} \left| \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) - \mathbb{E}\mathrm{pl}_j(\Lambda) \right| \leq \text{small.}$$

Yay! A sum of independent random variables

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

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

As usual:

- Show
$$\max_{\Lambda} \mathbb{E} \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) \leq \text{ small} \qquad \checkmark$$

- Show
$$\mathbb{E} \max_{\Lambda} \left| \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) - \mathbb{E}\mathrm{pl}_j(\Lambda) \right| \leq \text{small.}$$

Standard tricks

$$\mathbb{E} \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} \mathsf{pl}_j(\Lambda) - \mathbb{E}\mathsf{pl}_j(\Lambda) \right|$$

Standard tricks

Symmetrization and comparison

$$\mathbb{E} \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) - \mathbb{E}\mathrm{pl}_j(\Lambda) \right| \lesssim \mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} g_j \cdot \mathrm{pl}_j(\Lambda)$$

Standard tricks

Symmetrization and comparison

$$\mathbb{E} \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) - \mathbb{E}\mathrm{pl}_j(\Lambda) \right| \lesssim \mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} g_j \cdot \mathrm{pl}_j(\Lambda)$$

$$= \mathbb{E}_{\mathcal{C}} \left[\mathbb{E}_g \max_{|\Lambda|=L} g_j \cdot \mathrm{pl}_j(\Lambda) \mid \mathcal{C} \right]$$

Condition on $\mathcal{C}$ until further notice

Standard tricks

Symmetrization and comparison

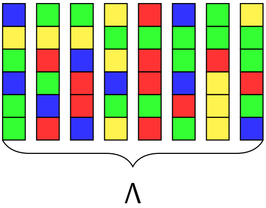
$$\mathbb{E} \max_{|\Lambda|=L} \left| \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) - \mathbb{E}\mathrm{pl}_j(\Lambda) \right| \lesssim \mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} g_j \cdot \mathrm{pl}_j(\Lambda)$$

$$= \mathbb{E}_{\mathcal{C}} \left[\mathbb{E}_g \max_{|\Lambda|=L} g_j \cdot \mathrm{pl}_j(\Lambda) \mid \mathcal{C} \right]$$

Condition on $\mathcal{C}$ until further notice

GOAL: bound the Gaussian mean width of

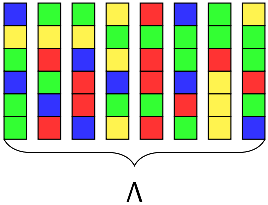
$$\left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = L \right\}$$



Attempt 1: Dudley's theorem

GOAL: bound the Gaussian mean width of

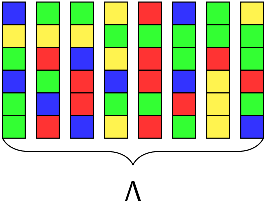
$$\left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = L \right\}$$



Attempt 1: Dudley's theorem

GOAL: bound the Gaussian mean width of

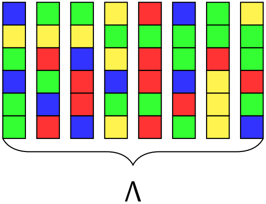
$$\left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} \; : \; |\Lambda| \leq L \right\}$$



Attempt 1: Dudley's theorem

GOAL: bound the Gaussian mean width of

$$\left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}$$



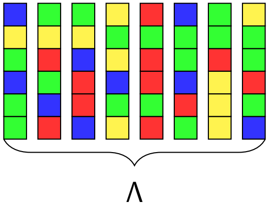
- Natural choice of nets:

$$\mathcal{N}_t = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$

Attempt 1: Dudley's theorem

GOAL: bound the Gaussian mean width of

$$\left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}$$



- Natural choice of nets:

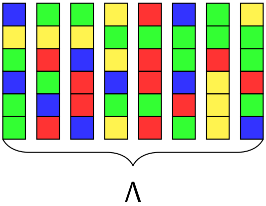
$$\mathcal{N}_t = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$

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

Attempt 1: Dudley's theorem

GOAL: bound the Gaussian mean width of

$$\left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\}$$



- Natural choice of nets:

$$\mathcal{N}_t = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$

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

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

Why should this work

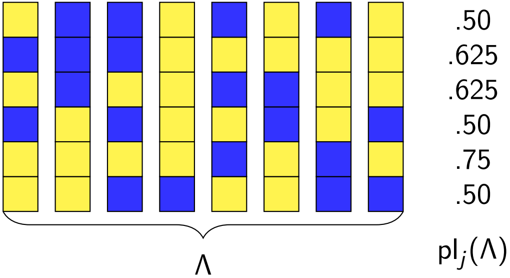
The goal is to show that $\mathcal{N}_t$ is a decent net of $\mathcal{X}$.
(w.r.t. ℓ_2, where "decent" degrades with t)

$$\mathcal{X} = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\} \qquad \mathcal{N}_t = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$

Why should this work

The goal is to show that $\mathcal{N}_t$ is a decent net of $\mathcal{X}$.
(w.r.t. ℓ_2, where "decent" degrades with t)

$$\mathcal{X} = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\} \qquad \mathcal{N}_t = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$

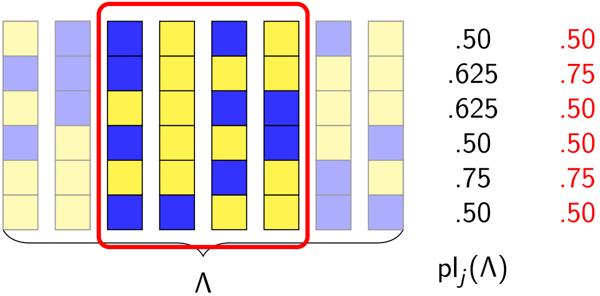


Why should this work

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

(w.r.t. ℓ_2, where "decent" degrades with t)

$$\mathcal{X} = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| \le L \right\} \qquad \mathcal{N}_t = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$



Why this shouldn't work

The goal is to show that $\mathcal{N}_t$ is a decent net of $\mathcal{X}$.
(w.r.t. ℓ_2, where "decent" degrades with t)

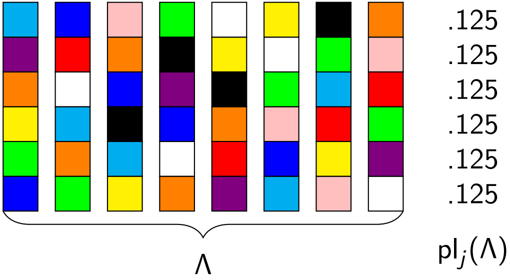
$$\mathcal{X} = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\} \qquad \mathcal{N}_t = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$

Why this shouldn't work

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

(w.r.t. ℓ_2, where "decent" degrades with t)

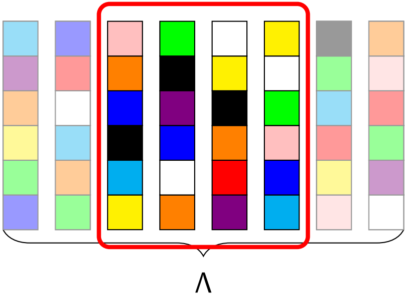
$$\mathcal{X} = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| \le L \right\} \qquad \mathcal{N}_t = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$



Why this shouldn't work

The goal is to show that $\mathcal{N}_t$ is a decent net of $\mathcal{X}$.
(w.r.t. ℓ_2, where "decent" degrades with t)

$$\mathcal{X} = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| \leq L \right\} \qquad \mathcal{N}_t = \left\{ \begin{pmatrix} \mathrm{pl}_1(\Lambda) \\ \mathrm{pl}_2(\Lambda) \\ \vdots \\ \mathrm{pl}_n(\Lambda) \end{pmatrix} : |\Lambda| = \frac{L}{2^t} \right\}$$



.125	.25
.125	.25
.125	.25
.125	.25
.125	.25
.125	.25

$\mathrm{pl}_j(\Lambda)$

Bummer!

Our great idea

to show that $\mathrm{pl}(\Lambda)$ is small

is foiled

Bummer!

Our great idea

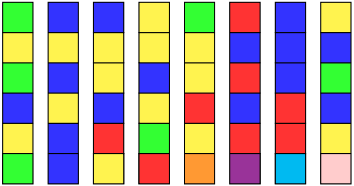
to show that $\mathrm{pl}(\Lambda)$ is small

is **foiled**

by sets Λ where $\mathrm{pl}(\Lambda)$ is small.

Solution

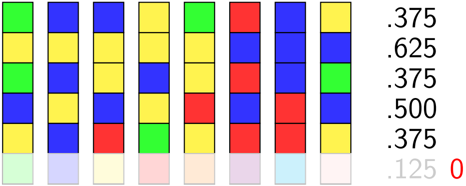
Nets are made up of pl(Λ), along with a set of active indices



.375
.625
.375
.500
.375
.125

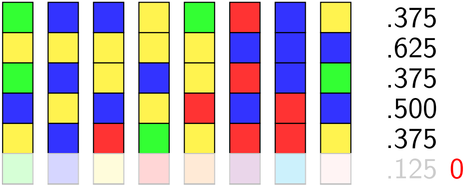
Solution

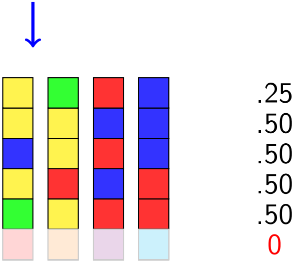
Nets are made up of pl(Λ), along with a set of active indices



Solution

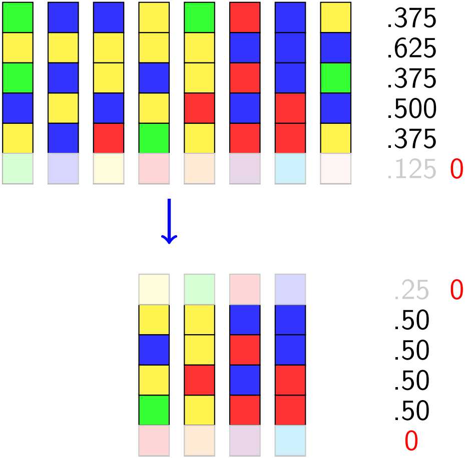
Nets are made up of pl(Λ), along with a set of active indices





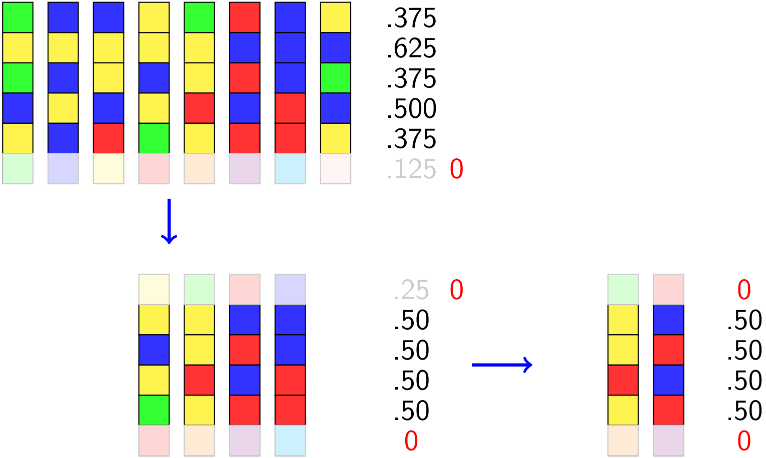
Solution

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



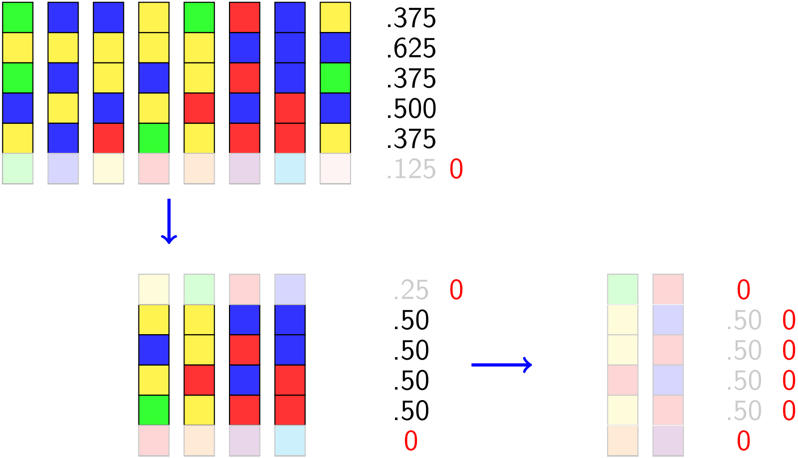
Solution

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



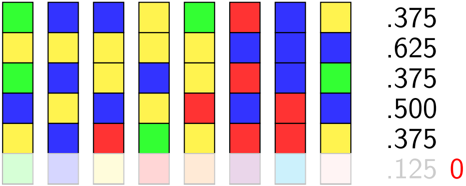
Solution

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



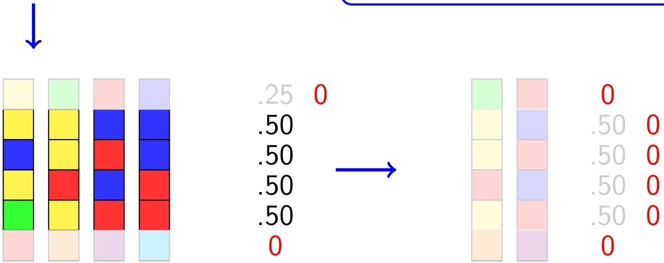
Solution

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



Guarantees:

- The active $\mathrm{pl}_j(\Lambda)$ never get too large.
- The distances between these vectors are not too big.



(Slightly) more precisely

(Slightly) more precisely

- Set Λ_0 of size L. Choose $I_0 = [n]$.

(Slightly) more precisely

- Set Λ_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 Λ_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:
  -
$$\|\mathrm{pl}_{I_t}(\Lambda_t) - \mathrm{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0)}.$$

(Slightly) more precisely

- Set Λ_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:
  - $$\|\mathrm{pl}_{I_t}(\Lambda_t) - \mathrm{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0)}.$$
  - $$\sum_{j \in I_t} \mathrm{pl}_j(\Lambda_t) \leq (1+\eta)^t \left(\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0) \right)$$

(Slightly) more precisely

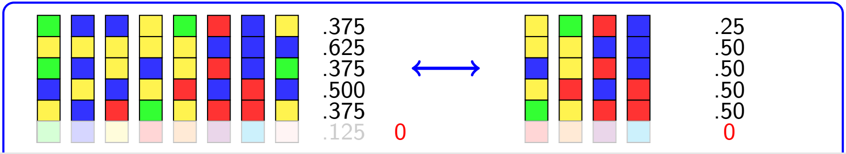
- Set Λ_0 of size L. Choose $I_0 = [n]$.
- There exists a chain $(\Lambda_0, I_0) \leftrightarrow (\Lambda_1, I_1) \leftrightarrow \ldots \leftrightarrow (\Lambda_T, I_T)$ so that:
  - $$\|\mathrm{pl}_{I_t}(\Lambda_t) - \mathrm{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0)}.$$
  - $$\sum_{j \in I_t} \mathrm{pl}_j(\Lambda_t) \lesssim \left(\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0) \right)$$

(Slightly) more precisely

- Set Λ_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:
  - $$\|\mathrm{pl}_{I_t}(\Lambda_t) - \mathrm{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0)}.$$
  - $$\sum_{j \in I_t} \mathrm{pl}_j(\Lambda_t) \lesssim \left(\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0) \right)$$
- Over all of the Λ_0's we could have started with, the number of pairs (Λ_t, I_t) that ever show up at level t is at most $\binom{|\mathcal{C}|}{L/2^t}^2$.

(Slightly) more precisely

- Set Λ_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:
  - $$\|\mathrm{pl}_{I_t}(\Lambda_t) - \mathrm{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0)}.$$
  - $$\sum_{j \in I_t} \mathrm{pl}_j(\Lambda_t) \lesssim \left(\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0) \right)$$
- Over all of the Λ_0's we could have started with, the number of pairs (Λ_t, I_t) that ever show up at level t is at most $\binom{|\mathcal{C}|}{L/2^t}^2$.



(Slightly) more precisely

- Set Λ_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:
  - $$\|\mathrm{pl}_{I_t}(\Lambda_t) - \mathrm{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0)}.$$
  - $$\sum_{j \in I_t} \mathrm{pl}_j(\Lambda_t) \lesssim \left(\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0) \right)$$
- Over all of the Λ_0's we could have started with, the number of pairs (Λ_t, I_t) that ever show up at level t is at most $\binom{|\mathcal{C}|}{L/2^t}^2$.

$$\boxed{\sum_j g_j \mathrm{pl}_j(\Lambda_0) \approx \sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T)}$$

(Slightly) more precisely

- Set Λ_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:
  - $$\|\mathrm{pl}_{I_t}(\Lambda_t) - \mathrm{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0)}.$$
  - $$\sum_{j \in I_t} \mathrm{pl}_j(\Lambda_t) \lesssim \left(\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0) \right)$$
- Over all of the Λ_0's we could have started with, the number of pairs (Λ_t, I_t) that ever show up at level t is at most $\binom{|\mathcal{C}|}{L/2^t}^2$.

$$\boxed{\sum_j g_j \mathrm{pl}_j(\Lambda_0) \approx \sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T)}$$

$$\boxed{\sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T) \text{ reasonable}}$$

(Slightly) more precisely

- Set Λ_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:
  - $$\|\mathrm{pl}_{I_t}(\Lambda_t) - \mathrm{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0)}.$$
  - $$\sum_{j \in I_t} \mathrm{pl}_j(\Lambda_t) \lesssim \left(\sum_{j=1}^{n} \mathrm{pl}_j(\Lambda_0) \right)$$

Call this Q
(this is what we originally wanted to bound)

- Over all of the Λ_0's we could have started with, the number of pairs (Λ_t, I_t) that ever show up at level t is at most $\binom{|\mathcal{C}|}{L/2^t}^2$.

$$\sum_j g_j \mathrm{pl}_j(\Lambda_0) \approx \sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T)$$

$$\sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T) \text{ reasonable}$$

Chaining argument

$$\boxed{\sum_j g_j \mathsf{pl}_j(\Lambda_0) \approx \sum_{j \in I_T} g_j \mathsf{pl}_j(\Lambda_T)}$$

- $$\|\mathsf{pl}_{I_t}(\Lambda_t) - \mathsf{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{Q}.$$
- Over all of the Λ_0's we could have started with, the number of pairs (Λ_t, I_t) that ever show up at level t is at most $\binom{|\mathcal{C}|}{L/2^t}^2$.

Chaining argument

$$\boxed{\sum_j g_j \mathrm{pl}_j(\Lambda_0) \approx \sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T)}$$

- $$\|\mathrm{pl}_{I_t}(\Lambda_t) - \mathrm{pl}_{I_{t+1}}(\Lambda_{t+1})\|_2 \lesssim \frac{1}{\sqrt{|\Lambda_t|}} \cdot \sqrt{Q}.$$
- Over all of the Λ_0's we could have started with, the number of pairs (Λ_t, I_t) that ever show up at level t is at most $\binom{|\mathcal{C}|}{L/2^t}^2$.

This implies (on board)

$$\mathbb{E} \max_{|\Lambda_0| = L} \left| \sum_{j=1}^n g_j \mathrm{pl}_j(\Lambda_0) - \sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T) \right| \lesssim \sqrt{Q \log |\mathcal{C}|}.$$

Base

$$\boxed{\sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T) \text{ reasonable}}$$

$$\sum_{j \in I_T} \mathrm{pl}_j(\Lambda_T) \lesssim Q \qquad \text{and} \qquad \mathrm{pl}_j(\Lambda_T) \leq 1 \quad \forall j$$

Base

$$\boxed{\sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T) \text{ reasonable}}$$

$$\sum_{j \in I_T} \mathrm{pl}_j(\Lambda_T) \lesssim Q \qquad \text{and} \qquad \mathrm{pl}_j(\Lambda_T) \leq 1 \quad \forall j$$

This implies (in your head)

$$\mathbb{E} \max_{\Lambda_0} \left| \sum_{j \in I_T} g_j \mathrm{pl}_j(\Lambda_T) \right| \lesssim \sqrt{Q}$$

Together

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

Together

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

Why did we want this again?

Together

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

Why did we want this again?

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

Together

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

Why did we want this again?

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

Together

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

Why did we want this again?

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

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

$$\mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) \lesssim n\varepsilon + \log |\mathcal{C}|.$$

Almost done

We just saw

$$\mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} \mathsf{pl}_j(\Lambda) \lesssim n\varepsilon + \log |\mathcal{C}|.$$

Almost done

We just saw

$$\mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) \lesssim n\varepsilon + \log|\mathcal{C}|.$$

We'd like

$$\log|\mathcal{C}| \leq n\varepsilon$$

Almost done

We just saw

$$\mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) \lesssim n\varepsilon + \log |\mathcal{C}|.$$

We'd like

$$\log |\mathcal{C}| \leq n\varepsilon$$

Almost done

We just saw

$$\mathbb{E} \max_{|\Lambda|=L} \sum_{j=1}^{n} \mathrm{pl}_j(\Lambda) \lesssim n\varepsilon + \log|\mathcal{C}|.$$

We'd like

$$\log|\mathcal{C}| \leq n\varepsilon$$

aka

$$R = \frac{\log_q(\mathcal{C})}{n} \lesssim \frac{\varepsilon}{\log(q)}.$$

Finally

Theorem

Suppose $q \geq \frac{k}{\varepsilon^2}$. Let $\mathcal{C}$ 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, $\mathcal{C}$ is (ρ, L)-list-decodable with

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

Finally

Theorem

Suppose $q \geq \frac{k}{\varepsilon^2}$. *Let* $\mathcal{C}$ *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, $\mathcal{C}$ *is* (ρ, L)*-list-decodable with*

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

Finally

Theorem

Suppose $q \geq \frac{k}{\varepsilon^2}$. Let $\mathcal{C}$ 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, $\mathcal{C}$ is (ρ, L)-list-decodable with

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

- What did we need from random linear codes?

Finally

Theorem

Suppose $q \geq \frac{k}{\varepsilon^2}$. *Let* $\mathcal{C}$ *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, $\mathcal{C}$ *is* (ρ, L)*-list-decodable with*

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

- What did we need from random linear codes?
  - Independent symbols.
  - $\max \mathbb{E}$ about right.

Finally

Theorem

Suppose $q \geq \frac{k}{\varepsilon^2}$. Let $\mathcal{C}$ 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, $\mathcal{C}$ is (ρ, L)-list-decodable with

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

- 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.

Recap

- 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 $\max \mathbb{E}$.

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!