Graph entropy and the zero-error capacity of channels

Jaikumar Radhakrishnan

Tata Institute of Fundamental Research, Mumbai

Based on joint work with Siddharth Bhandari (TIFR), Venkat Guruswami (CMU) and Marco Dalai (Brescia)

The binary difference problem

Fix N. What is the minimum n such that we can find N **different** sequences in $\{0,1\}^n$?

Answer: $\lceil \log_2 N \rceil$

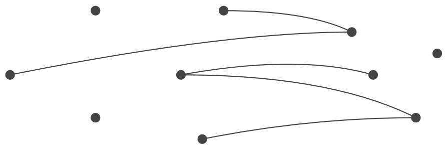
What is the maximum number of sequences in $\{0,1\}^n$ such that every **two** of them are **different**?

Answer: 2^n

Hansel's lemma about graph covering

Suppose $[M] = \{1, 2, \ldots, M\}$, K_M = the complete graph on $[M]$, and

- G_i ($i \in I$) a finite sequence of bipartite graphs on $[M]$
- τ_i fraction of non-isolated vertices in G_i



$\tau_i = 7/10$

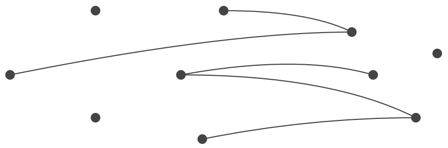
Lemma (Hansel '64)

$$\bigcup_{i\in I} G_i = K_M \implies \sum_{i\in I} \tau_i \geq \log_2(M).$$

Hansel's lemma about graph covering

Suppose $[M] = \{1, 2, \ldots, M\}$, K_M = the complete graph on $[M]$, and

- G_i ($i \in I$) a finite sequence of bipartite graphs on $[M]$
- τ_i fraction of non-isolated vertices in G_i



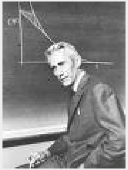
$\tau_i = 7/10$

Lemma (Hansel '64)

$$\bigcup_{i \in I} G_i = K_M \quad \Longrightarrow \quad \sum_{i \in I} \tau_i \geq \log_2(M).$$

Entropy of a random variable

Definition (Shannon entropy)



Claude Shannon (1916-2001)

Let $\mathbf{X}$ be a random variable taking values in the set $[n] = \{1, 2, \ldots, n\}$. Then, its entropy is given by

$$H[\mathbf{X}] = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

- $H[\mathbf{X}]$ measures the uncertainity in $\mathbf{X}$.
- $H[\mathbf{X}]$ is a function of the distribution of $\mathbf{X}$, not the actual values it takes.
- $H[\mathbf{X}] \leq \log_2 n$.

Pragmatics

Alice

Observes **X**

Sends Bob a message **M**

$\Leftarrow$

$\Rightarrow$

Bob

Recovers **X** from **M**

Goal

- Alice and Bob exchange bits.
- Bob must recover **X** exactly.
- Goal: minimize the (expected) total number of bits transmitted.

Transmission cost

Let $T[\mathbf{X}]$ denote the minimum (expected) cost of transmitting **X**.

Pragmatics

Alice

Observes **X**
Sends Bob a message **M**

$\Leftarrow$
$\Rightarrow$

Bob

Recovers **X** from **M**

Goal

- Alice and Bob exchange bits.
- Bob must recover **X** exactly.
- Goal: minimize the (expected) total number of bits transmitted.

Transmission cost

Let $T[\mathbf{X}]$ denote the minimum (expected) cost of transmitting **X**.

Entropy and transmission

Theorem

$$H[\mathbf{X}] \;\leq\; T[\mathbf{X}] \;\leq\; H[\mathbf{X}] + 1\,.$$

Long years ago . . . 1948

Shannon's source coding theorem

Let p be a probability distribution on $[n]$. For $\epsilon > 0$ and positive integer k, let

$$N(k,\epsilon) = \min_{A \subset [n]^k : p^k(A) \geq 1-\epsilon} |A|.$$

Theorem (Shannon)

For all $\epsilon \in (0,1)$, $\lim_{k \to \infty} \frac{1}{k} \log_2 |N(k,\epsilon)| = H(p)$.

Long years ago . . . 1948

Shannon's source coding theorem

Let p be a probability distribution on $[n]$. For $\epsilon > 0$ and positive integer k, let

$$N(k,\epsilon) = \min_{A \subset [n]^k : p^k(A) \geq 1-\epsilon} |A|.$$

Theorem (Shannon)

For all $\epsilon \in (0,1)$, $\displaystyle\lim_{k\to\infty} \frac{1}{k} \log_2 |N(k,\epsilon)| = H(p)$.

Long years ago . . . 1948

Shannon's source coding theorem

Let p be a probability distribution on $[n]$. For $\epsilon > 0$ and positive integer k, let

$$N(k, \epsilon) = \min_{A \subset [n]^k : p^k(A) \geq 1-\epsilon} |A|.$$

Theorem (Shannon)

For all $\epsilon \in (0, 1)$, $\lim_{k \to \infty} \frac{1}{k} \log_2 |N(k, \epsilon)| = H(p)$.

. . . not wholly or in full measure, but very substantially!

Conditional entropy

Definition

$(\mathbf{X}, \mathbf{Y})$: a pair of random variables with some joint distribution.

$$H[\mathbf{Y} \mid \mathbf{X}] = \sum_i p_X(i) H[\mathbf{Y}_i]$$

Fact

- *Conditioning reduces uncertainity:* $H[\mathbf{Y} \mid \mathbf{X}] \leq H[\mathbf{Y}]$.
- $H[\mathbf{XY}] = H[\mathbf{X}] + H[\mathbf{Y} \mid \mathbf{X}]$.

Conditional entropy

Definition

$(\mathbf{X}, \mathbf{Y})$: a pair of random variables with some joint distribution.

$$H[\mathbf{Y} \mid \mathbf{X}] = \sum_i p_X(i) H[\mathbf{Y}_i]$$

Fact

- *Conditioning reduces uncertainity:* $H[\mathbf{Y} \mid \mathbf{X}] \leq H[\mathbf{Y}]$.
- $H[\mathbf{XY}] = H[\mathbf{X}] + H[\mathbf{Y} \mid \mathbf{X}]$.

The noisy channel

Specification

Input alphabet: $[m]$

Output alphabet: $[n]$

Characteristics: $\Pr[\text{output} = j \mid \text{input} = i] = p_{j|i}$.

Code of conduct

Encoding: $\{0,1\}^k \to [m]^t$

Decoding: $[n]^t \to \{0,1\}^k$

Goal

Error: $\Pr[\text{input} \neq \text{output}] \leq \epsilon$.

Rate: $\dfrac{k}{t}$ should be as large as possible.

The noisy channel

Specification

Input alphabet: $[m]$

Output alphabet: $[n]$

Characteristics: $\Pr[\text{output} = j \mid \text{input} = i] = p_{j|i}$.

Code of conduct

Encoding: $\{0,1\}^k \to [m]^t$

Decoding: $[n]^t \to \{0,1\}^k$

Goal

Error: $\Pr[\text{input} \neq \text{output}] \leq \epsilon$.

Rate: $\dfrac{k}{t}$ should be as large as possible.

The noisy channel

Specification

Input alphabet: $[m]$

Output alphabet: $[n]$

Characteristics: $\Pr[\text{output} = j \mid \text{input} = i] = p_{j|i}$.

Code of conduct

Encoding: $\{0,1\}^k \to [m]^t$

Decoding: $[n]^t \to \{0,1\}^k$

Goal

Error: $\Pr[\text{input} \neq \text{output}] \leq \epsilon$.

Rate: $\dfrac{k}{t}$ should be as large as possible.

Capacity

Input to the channel: $\mathbf{X} \in [m]$
Ouput of the channel: $\mathbf{Y} \in [n]$.

Definition (Capacity of a channel E)

$$C(E) = \max_{X} H[\mathbf{X}] + H[\mathbf{Y}] - H[\mathbf{XY}].$$

Jaane kya toone kahi . . . jaane kya meine suni

Theorem (Shannon)

Let C be the capacity of the channel. Then, for all $\epsilon > 0$ and all k, there exist encoders and decoders such that

Encoding rate: $\frac{k}{t} \geq C - \epsilon$.

Error: $\Pr[error] \to 0$ as $k \to \infty$.

Optimality: *Can't replace* $C - \epsilon$ *by* $C + \delta$ *for any* $\delta > 0$.

Jaane kya toone kahi . . . jaane kya meine suni

Theorem (Shannon)

Let C be the capacity of the channel. Then, for all $\epsilon > 0$ and all k, there exist encoders and decoders such that

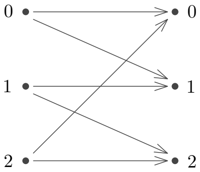
Encoding rate: $\frac{k}{t} \geq C - \epsilon$.

Error: $\Pr[error] \to 0$ *as* $k \to \infty$.

Optimality: *Can't replace* $C - \epsilon$ *by* $C + \delta$ *for any* $\delta > 0$.

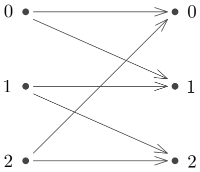
. . . baat kuchch ban hi gayi!

The 3/2 channel: trifference



- Any two input symbols are confusable (share a common output).
  - Any two codewords are confusable (in any position).
  - Zero-error capacity 0.
- However, the three inputs have no common output.

The 3/2 channel: trifference



- Any two input symbols are confusable (share a common output).

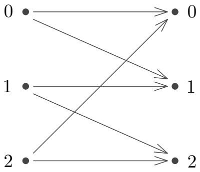
  $\Longrightarrow$ Any two codewords are confusable (in any position).

  $\Longrightarrow$ Zero-error capacity 0.
- However, the three inputs have no common output.

  $\Longrightarrow$ We can build triplets of codewords which cannot be confused (i.e., we can narrow down the intended codeword to a list of size $L = 2$).

$$
\begin{array}{ccccccccc}
x & = & 0 & 0 & 0 & 2 & 1 & 2 & \cdots \\
y & = & 0 & 1 & 1 & 1 & 2 & 1 & \cdots \\
z & = & 1 & 1 & 2 & 2 & 1 & 2 & \cdots
\end{array}
$$

The 3/2 channel: trifference



- Any two input symbols are confusable (share a common output).

  $\Longrightarrow$ Any two codewords are confusable (in any position).

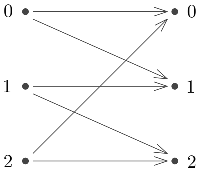
  $\Longrightarrow$ Zero-error capacity 0.

- However, the three inputs have no common output.

  $\Longrightarrow$ We can build triplets of codewords which cannot be confused (i.e., we can narrow down the intended codeword to a list of size $L = 2$).

$$\begin{array}{ccccccccc} x & = & 0 & 0 & 0 & 2 & 1 & 2 & \cdots \\ y & = & 0 & 1 & 1 & 1 & 2 & 1 & \cdots \\ z & = & 1 & 1 & 2 & 2 & 1 & 2 & \cdots \end{array}$$

The 3/2 channel: trifference



- Any two input symbols are confusable (share a common output).

$\Longrightarrow$ Any two codewords are confusable (in any position).

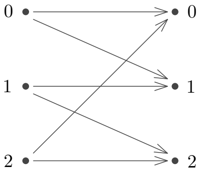
$\Longrightarrow$ Zero-error capacity 0.

- However, the three inputs have no common output.

$\Longrightarrow$ We can build triplets of codewords which cannot be confused (i.e., we can narrow down the intended codeword to a list of size $L = 2$).

$$
\begin{array}{ccccccccc}
x & = & 0 & 0 & 0 & 2 & 1 & 2 & \cdots \\
y & = & 0 & 1 & 1 & 1 & 2 & 1 & \cdots \\
z & = & 1 & 1 & 2 & 2 & 1 & 2 & \cdots
\end{array}
$$

The 3/2 channel: trifference



- Any two input symbols are confusable (share a common output).

$\Longrightarrow$ Any two codewords are confusable (in any position).

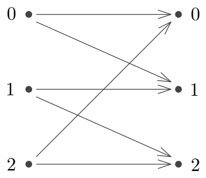
$\Longrightarrow$ Zero-error capacity 0.

- However, the three inputs have no common output.

$\Longrightarrow$ We can build triplets of codewords which cannot be confused (i.e., we can narrow down the intended codeword to a list of size $L = 2$).

$$\begin{array}{cccccccc} x & = & 0 & 0 & 0 & 2 & 1 & 2 & \cdots \\ y & = & 0 & 1 & 1 & 1 & 2 & 1 & \cdots \\ z & = & 1 & 1 & 2 & 2 & 1 & 2 & \cdots \end{array}$$

The 3/2 channel: trifference



- Any two input symbols are confusable (share a common output).

  $\Longrightarrow$ Any two codewords are confusable (in any position).

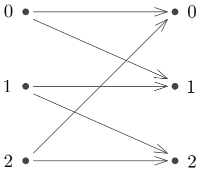
  $\Longrightarrow$ Zero-error capacity 0.

- However, the three inputs have no common output.

  $\Longrightarrow$ We can build triplets of codewords which cannot be confused (i.e., we can narrow down the intended codeword to a list of size $L = 2$).

$$
\begin{array}{lllllllll}
x & = & 0 & 0 & \color{red}{0} & 2 & 1 & 2 & \cdots \\
y & = & 0 & 1 & \color{red}{1} & 1 & 2 & 1 & \cdots \\
z & = & 1 & 1 & \color{red}{2} & 2 & 1 & 2 & \cdots
\end{array}
$$

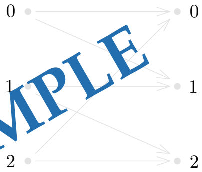
The 3/2 channel: trifference



- Any two input symbols are confusable (share a common output).
  - $\Longrightarrow$ Any two codewords are confusable (in any position).
  - $\Longrightarrow$ Zero-error capacity 0.
- However, the three inputs have no common output.
  - $\Longrightarrow$ We can build triplets of codewords which cannot be confused (i.e., we can narrow down the intended codeword to a list of size $L = 2$).

$$\begin{array}{ccccccccc} x & = & 0 & 0 & 0 & 2 & 1 & 2 & \cdots \\ y & = & 0 & 1 & 1 & 1 & 2 & 1 & \cdots \\ z & = & 1 & 1 & 2 & 2 & 1 & 2 & \cdots \end{array}$$

The 3/2 channel: trifference

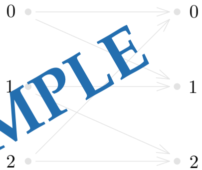


VERY SIMPLE

- Any two input symbols are confusable (share a common output).
  $\Longrightarrow$ Any two codewords are confusable (in any position).
  $\Longrightarrow$ Zero-error capacity 0.
- However, the three inputs have no common output.
  $\Longrightarrow$ We can build triplets of codewords which cannot be confused (i.e., we can narrow down the intended codeword to a list of size $L = 2$).

$$
\begin{array}{ccccccccc}
x & = & 0 & 0 & 0 & 2 & 1 & 2 & \cdots \\
y & = & 0 & 1 & 1 & 1 & 2 & 1 & \cdots \\
z & = & 1 & 1 & 2 & 2 & 1 & 2 & \cdots
\end{array}
$$

The 3/2 channel: trifference



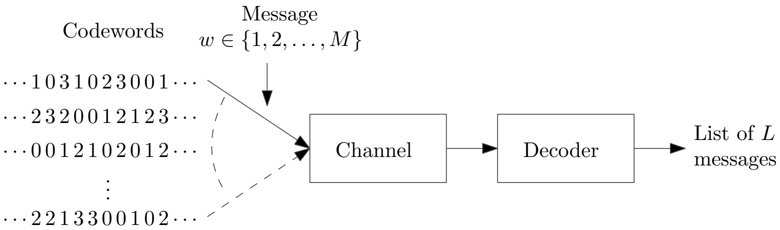
VERY SIMPLE

- Any two input symbols are confusable (share a common output).
  - $\Longrightarrow$ Any two codewords are confusable (in any position).
  - $\Longrightarrow$ Zero-error capacity 0.
- However, the three inputs have no common output.
  - $\Longrightarrow$ We can build triplets of codewords which cannot be confused (i.e., we can narrow down the intended codeword to a list of size $L = 2$).

VERY DIFFICULT

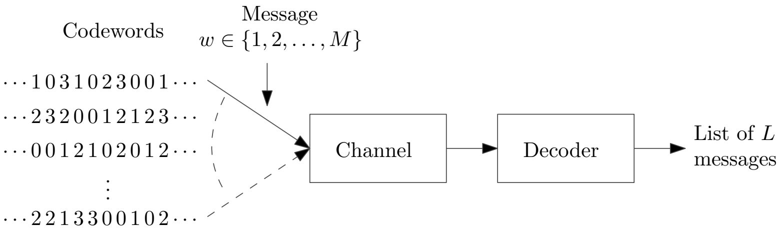
$$
\begin{array}{ccccccccc}
x & = & 0 & 0 & 0 & 2 & 1 & 2 & \cdots \\
y & = & 0 & 1 & 1 & 1 & 2 & 1 & \cdots \\
z & = & 1 & 1 & 2 & 2 & 1 & 2 & \cdots
\end{array}
$$

Zeror-error capacity with list decoding



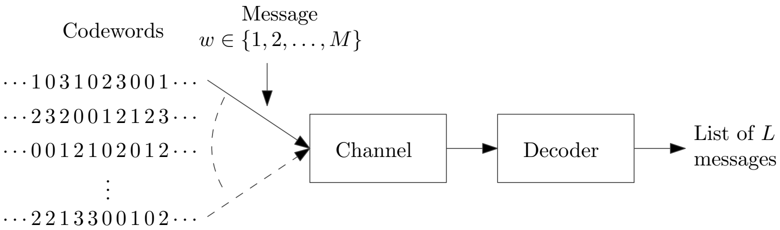
- The decoder outputs a *list* of L messages
- Zero-error code: the correct message is always in the list
  - $\Leftrightarrow$ No $L+1$ codewords are compatible with any output sequence
- Zero-error capacity with list size L: the maximum R such that there is such a zero-error code of size 2^{nR}

Zeror-error capacity with list decoding



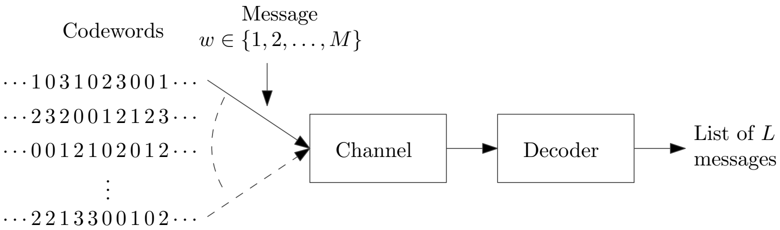
- The decoder outputs a *list* of L messages
- Zero-error code: the correct message is always in the list
  - $\Leftrightarrow$ No $L+1$ codewords are compatible with any output sequence
- Zero-error capacity with list size L: the maximum R such that there is such a zero-error code of size 2^{nR}

Zeror-error capacity with list decoding



- The decoder outputs a *list* of L messages
- Zero-error code: the correct message is always in the list
  - $\Leftrightarrow$ No $L+1$ codewords are compatible with any output sequence
- Zero-error capacity with list size L: the maximum R such that there is such a zero-error code of size 2^{nR}

Zeror-error capacity with list decoding



- The decoder outputs a *list* of L messages
- Zero-error code: the correct message is always in the list
  - $\Leftrightarrow$ No $L + 1$ codewords are compatible with any output sequence
- Zero-error capacity with list size L: the maximum R such that there is such a zero-error code of size 2^{nR}

Ternary encoding

Let $R_3(n)$ be the size of the largest subset $C \subseteq \{0,1,2\}^n$ such that every **three** of them are **trifferent**?

$$\textbf{trifferent} \equiv \forall \text{ (distinct) } x, y, z \in C \quad \exists i : \{x_i, y_i, z_i\} = \{0,1,2\}$$

$$\begin{array}{lcccccccc} x & = & 0 & 0 & 0 & 2 & 1 & 2 & \cdots \\ y & = & 0 & 1 & 1 & 1 & 2 & 1 & \cdots \\ z & = & 1 & 1 & 2 & 2 & 1 & 2 & \cdots \end{array}$$

Answer: $2^{0.212n} \leq R_3(n) \leq 2\left(\frac{3}{2}\right)^n \approx 2^{0.545n}$ (Körner & Marton '88)

Ternary encoding

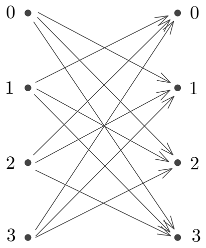
Let $R_3(n)$ be the size of the largest subset $C \subseteq \{0,1,2\}^n$ such that every **three** of them are **trifferent**?

$$\textbf{trifferent} \equiv \forall \text{ (distinct) } x, y, z \in C \quad \exists i : \{x_i, y_i, z_i\} = \{0,1,2\}$$

$$\begin{array}{ccccccccc} x & = & 0 & 0 & 0 & 2 & 1 & 2 & \cdots \\ y & = & 0 & 1 & 1 & 1 & 2 & 1 & \cdots \\ z & = & 1 & 1 & 2 & 2 & 1 & 2 & \cdots \end{array}$$

Answer: $2^{0.212n} \leq R_3(n) \leq 2\left(\frac{3}{2}\right)^n \approx 2^{0.545n}$ (Körner & Marton '88)

Today: the 4/3 channel

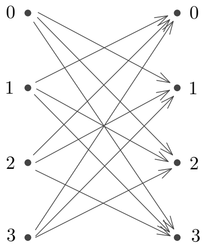


- The four inputs have no common output
  - We can build 4-tuples of codewords which cannot be confused

$$
\begin{array}{llllllll}
w & = & 0 & 2 & 2 & 2 & 3 & 1 \cdots \\
x & = & 2 & 3 & 1 & 0 & 2 & 1 \cdots \\
y & = & 1 & 3 & 3 & 3 & 3 & 0 \cdots \\
z & = & 1 & 0 & 0 & 2 & 1 & 2 \cdots
\end{array}
$$

  - Upper bounds for zero-error capacity with $L = 3$?

Today: the 4/3 channel



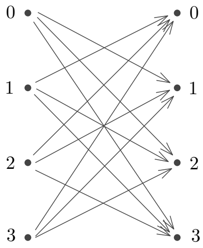
- The four inputs have no common output

$\Longrightarrow$ We can build 4-tuples of codewords which cannot be confused

$$
\begin{array}{cccccccc}
w & = & 0 & 2 & 2 & 2 & 3 & 1 & \cdots \\
x & = & 2 & 3 & 1 & 0 & 2 & 1 & \cdots \\
y & = & 1 & 3 & 3 & 3 & 3 & 0 & \cdots \\
z & = & 1 & 0 & 0 & 2 & 1 & 2 & \cdots
\end{array}
$$

... Upper bounds for zero-error capacity with $L = 3$?

Today: the 4/3 channel



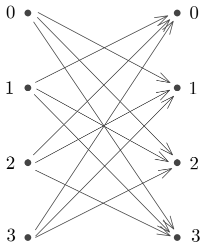
- The four inputs have no common output

$\Longrightarrow$ We can build 4-tuples of codewords which cannot be confused

$$
\begin{array}{llcccccccc}
w & = & 0 & 2 & \color{red}{2} & 2 & 3 & 1 & \cdots \\
x & = & 2 & 3 & \color{red}{1} & 0 & 2 & 1 & \cdots \\
y & = & 1 & 3 & \color{red}{3} & 3 & 3 & 0 & \cdots \\
z & = & 1 & 0 & \color{red}{0} & 2 & 1 & 2 & \cdots
\end{array}
$$

... Upper bounds for zero-error capacity with $L = 3$?

Today: the 4/3 channel



- The four inputs have no common output

$\Longrightarrow$ We can build 4-tuples of codewords which cannot be confused

$$
\begin{array}{lcccccccc}
w & = & 0 & 2 & \color{red}{2} & 2 & 3 & 1 & \cdots \\
x & = & 2 & 3 & \color{red}{1} & 0 & 2 & 1 & \cdots \\
y & = & 1 & 3 & \color{red}{3} & 3 & 3 & 0 & \cdots \\
z & = & 1 & 0 & \color{red}{0} & 2 & 1 & 2 & \cdots
\end{array}
$$

... Upper bounds for zero-error capacity with $L = 3$?

Alphabet size 4

Problem

Let $R_4(n)$ be the size of the largest code $C \subseteq \{0,1,2,3\}^n$ that satisfies

$$\forall \text{ (distinct)} w, x, y, z \in C \quad \exists i : \{w_i, x_i, y_i, z_i\} = \{0,1,2,3\}$$

- Standard random coding: $R_4(n) \geq 2^{0.0473n}$
- Simple upper bound $R_4(n) \leq 3 \cdot (\frac{4}{3})^n \approx 2^{0.415n}$
- Fredman-Komlós '84: $R_4(n) \leq 2^{\frac{3}{8}(1+o(1))n} \approx 2^{0.375n}$
- Arikan '94: $R_4(n) \leq 2^{0.3512(1+o(1))n}$, using Plotkin the bound

  can be improved to about $2^{0.3276n}$, using the linear programming bound

Alphabet size 4

Problem

Let $R_4(n)$ be the size of the largest code $C \subseteq \{0,1,2,3\}^n$ that satisfies

$$\forall \text{ (distinct)} w,x,y,z \in C \quad \exists i : \{w_i, x_i, y_i, z_i\} = \{0,1,2,3\}$$

- Standard random coding: $R_4(n) \geq 2^{0.0473n}$
- Simple upper bound $R_4(n) \leq 3 \cdot (\frac{4}{3})^n \approx 2^{0.415n}$
- Fredman-Komlós '84: $R_4(n) \leq 2^{\frac{3}{8}(1+o(1))n} \approx 2^{0.375n}$
- Arikan '94: $R_4(n) \leq 2^{0.3512(1+o(1))n}$, using Plotkin the bound

  ... can be improved to about $2^{0.3276n}$, using the linear programming bound

Alphabet size 4

Problem

Let $R_4(n)$ be the size of the largest code $C \subseteq \{0,1,2,3\}^n$ that satisfies

$$\forall\ \text{(distinct)} w,x,y,z \in C \quad \exists i : \{w_i, x_i, y_i, z_i\} = \{0,1,2,3\}$$

- Standard random coding: $R_4(n) \geq 2^{0.0473n}$
- Simple upper bound $R_4(n) \leq 3 \cdot (\frac{4}{3})^n \approx 2^{0.415n}$
- Fredman-Komlós '84: $R_4(n) \leq 2^{\frac{3}{8}(1+o(1))n} \approx 2^{0.375n}$
- Arikan '94: $R_4(n) \leq 2^{0.3512(1+o(1))n}$, using Plotkin the bound

  ... can be improved to about $2^{0.3276n}$, using the linear programming bound

Alphabet size 4

Problem

Let $R_4(n)$ be the size of the largest code $C \subseteq \{0,1,2,3\}^n$ that satisfies

$$\forall \text{ (distinct)} w, x, y, z \in C \quad \exists i : \{w_i, x_i, y_i, z_i\} = \{0,1,2,3\}$$

- Standard random coding: $R_4(n) \geq 2^{0.0473n}$
- Simple upper bound $R_4(n) \leq 3 \cdot (\frac{4}{3})^n \approx 2^{0.415n}$
- Fredman-Komlós '84: $R_4(n) \leq 2^{\frac{3}{8}(1+o(1))n} \approx 2^{0.375n}$
- Arikan '94: $R_4(n) \leq 2^{0.3512(1+o(1))n}$, using Plotkin the bound

  ... can be improved to about $2^{0.3276n}$, using the linear programming bound

Alphabet size 4

Problem

Let $R_4(n)$ be the size of the largest code $C \subseteq \{0,1,2,3\}^n$ that satisfies

$$\forall \text{ (distinct)} w,x,y,z \in C \quad \exists i : \{w_i, x_i, y_i, z_i\} = \{0,1,2,3\}$$

- Standard random coding: $R_4(n) \geq 2^{0.0473n}$
- Simple upper bound $R_4(n) \leq 3 \cdot (\frac{4}{3})^n \approx 2^{0.415n}$
- Fredman-Komlós '84: $R_4(n) \leq 2^{\frac{3}{8}(1+o(1))n} \approx 2^{0.375n}$
- Arikan '94: $R_4(n) \leq 2^{0.3512(1+o(1))n}$, using Plotkin the bound

... can be improved to about $2^{0.3276n}$, using the linear programming bound

Alphabet size 4

Problem

Let $R_4(n)$ be the size of the largest code $C \subseteq \{0,1,2,3\}^n$ that satisfies

$$\forall \text{ (distinct)} w, x, y, z \in C \quad \exists i : \{w_i, x_i, y_i, z_i\} = \{0,1,2,3\}$$

- Standard random coding: $R_4(n) \geq 2^{0.0473n}$
- Simple upper bound $R_4(n) \leq 3 \cdot (\frac{4}{3})^n \approx 2^{0.415n}$
- Fredman-Komlós '84: $R_4(n) \leq 2^{\frac{3}{8}(1+o(1))n} \approx 2^{0.375n}$
- Arikan '94: $R_4(n) \leq 2^{0.3512(1+o(1))n}$, using Plotkin the bound
  - ... can be improved to about $2^{0.3276n}$, using the linear programming bound

Today

$$R_4(n) \leq 2^{\frac{6}{19}(1+o(1))n} \approx 2^{0.3158n},$$

that is,

$$\log \frac{R_4(n)}{n} \leq \frac{6}{19}(1+o(1)) \approx 0.3158.$$

(Joint work with Venkat Guruswami and Marco Dalai '17)

The framework (due to Fredman and Komlós)

Let $C \subseteq \{0,1,2,3\}^n$ be a 4-hash code.

Step 1: Fix distinct $x, y \in C$.

Step 2: For $i \in [n]$ and code word $z \in C \setminus \{x, y\}$

$\Longrightarrow$ if $x_i = y_i$, set z_i to $\star$;

$\Longrightarrow$ if $z_i \in \{x_i, y_i\}$, set z_i to $\star$;

$\Longrightarrow$ otherwise, let $\{a_i, b_i\} = \{0,1,2,3\} \setminus \{x_i, y_i\}$;
if $z_i = a_i$ replace it by 0, and if $z_i = b_i$, replace it by 1.

Step 3: Apply Hansel's lemma to the resulting code over $\{0, 1, \star\}$.

The framework (due to Fredman and Komlós)

Let $C \subseteq \{0,1,2,3\}^n$ be a 4-hash code.

Step 1: Fix distinct $x, y \in C$.

Step 2: For $i \in [n]$ and code word $z \in C \setminus \{x, y\}$

$\implies$ if $x_i = y_i$, set z_i to $\star$;

$\implies$ if $z_i \in \{x_i, y_i\}$, set z_i to $\star$;

$\implies$ otherwise, let $\{a_i, b_i\} = \{0,1,2,3\} \setminus \{x_i, y_i\}$;
if $z_i = a_i$ replace it by 0, and if $z_i = b_i$, replace it by 1.

Step 3: Apply Hansel's lemma to the resulting code over $\{0, 1, \star\}$.

The framework (due to Fredman and Komlós)

Let $C \subseteq \{0, 1, 2, 3\}^n$ be a 4-hash code.

Step 1: Fix distinct $x, y \in C$.

Step 2: For $i \in [n]$ and code word $z \in C \setminus \{x, y\}$

$\Longrightarrow$ if $x_i = y_i$, set z_i to $\star$;

$\Longrightarrow$ if $z_i \in \{x_i, y_i\}$, set z_i to $\star$;

$\Longrightarrow$ otherwise, let $\{a_i, b_i\} = \{0, 1, 2, 3\} \setminus \{x_i, y_i\}$;
if $z_i = a_i$ replace it by 0, and if $z_i = b_i$, replace it by 1.

Step 3: Apply Hansel's lemma to the resulting code over $\{0, 1, \star\}$.

The framework (due to Fredman and Komlós)

Let $C \subseteq \{0,1,2,3\}^n$ be a 4-hash code.

Step 1: Fix distinct $x, y \in C$.

Step 2: For $i \in [n]$ and code word $z \in C \setminus \{x, y\}$

$\Longrightarrow$ if $x_i = y_i$, set z_i to $\star$;

$\Longrightarrow$ if $z_i \in \{x_i, y_i\}$, set z_i to $\star$;

$\Longrightarrow$ otherwise, let $\{a_i, b_i\} = \{0,1,2,3\} \setminus \{x_i, y_i\}$;
if $z_i = a_i$ replace it by 0, and if $z_i = b_i$, replace it by 1.

Step 3: Apply Hansel's lemma to the resulting code over $\{0, 1, \star\}$.

The framework (due to Fredman and Komlós)

Let $C \subseteq \{0,1,2,3\}^n$ be a 4-hash code.

Step 1: Fix distinct $x, y \in C$.

Step 2: For $i \in [n]$ and code word $z \in C \setminus \{x, y\}$

$\Longrightarrow$ if $x_i = y_i$, set z_i to $\star$;

$\Longrightarrow$ if $z_i \in \{x_i, y_i\}$, set z_i to $\star$;

$\Longrightarrow$ otherwise, let $\{a_i, b_i\} = \{0,1,2,3\} \setminus \{x_i, y_i\}$;
if $z_i = a_i$ replace it by 0, and if $z_i = b_i$, replace it by 1.

Step 3: Apply Hansel's lemma to the resulting code over $\{0, 1, \star\}$.

The framework (due to Fredman and Komlós)

Let $C \subseteq \{0, 1, 2, 3\}^n$ be a 4-hash code.

Step 1: Fix distinct $x, y \in C$.

Step 2: For $i \in [n]$ and code word $z \in C \setminus \{x, y\}$

$\Longrightarrow$ if $x_i = y_i$, set z_i to $\star$;

$\Longrightarrow$ if $z_i \in \{x_i, y_i\}$, set z_i to $\star$;

$\Longrightarrow$ otherwise, let $\{a_i, b_i\} = \{0, 1, 2, 3\} \setminus \{x_i, y_i\}$;
if $z_i = a_i$ replace it by 0, and if $z_i = b_i$, replace it by 1.

Step 3: Apply Hansel's lemma to the resulting code over $\{0, 1, \star\}$.

The picture

$$\begin{bmatrix} 3 & 1 & 3 & 0 & 0 & 1 & \dots \\ 0 & 1 & 1 & 2 & 3 & 0 & \dots \\ 3 & 2 & 2 & 2 & 0 & 3 & \dots \\ 2 & 3 & 1 & 3 & 2 & 2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ 1 & 3 & 1 & 1 & 0 & 1 & \dots \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & 1 & 3 & 0 & 0 & 1 & \dots \\ 0 & 1 & 1 & 2 & 3 & 0 & \dots \\ \star & \star & 1 & \star & \star & 1 & \dots \\ 1 & \star & \star & 1 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ 0 & \star & \star & 0 & \star & \star & \dots \end{bmatrix}$$

The picture

$$
\begin{bmatrix}
3 & 1 & 3 & 0 & 0 & 1 & \dots \\
0 & 1 & 1 & 2 & 3 & 0 & \dots \\
3 & 2 & 2 & 2 & 0 & 3 & \dots \\
2 & 3 & 1 & 3 & 2 & 2 & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\
1 & 3 & 1 & 1 & 0 & 1 & \dots
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
3 & 1 & 3 & 0 & 0 & 1 & \dots \\
0 & 1 & 1 & 2 & 3 & 0 & \dots \\
\star & \star & 1 & \star & \star & 1 & \dots \\
1 & \star & \star & 1 & 1 & 0 & \dots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\
0 & \star & \star & 0 & \star & \star & \dots
\end{bmatrix}
$$

The picture

$$\begin{bmatrix} 3 & 1 & 3 & 0 & 0 & 1 & \dots \\ 0 & 1 & 1 & 2 & 3 & 0 & \dots \\ 3 & 2 & 2 & 2 & 0 & 3 & \dots \\ 2 & 3 & 1 & 3 & 2 & 2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ 1 & 3 & 1 & 1 & 0 & 1 & \dots \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & 1 & 3 & 0 & 0 & 1 & \dots \\ 0 & 1 & 1 & 2 & 3 & 0 & \dots \\ \star & \star & 1 & \star & \star & 1 & \dots \\ 1 & \star & \star & 1 & 1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots \\ 0 & \star & \star & 0 & \star & \star & \dots \end{bmatrix}$$

The Fredman-Komlós bound

Recall that $C \subseteq \{0,1,2,3\}^n$ be a 4-hash code.

- Pick distinct $x, y \in C$ uniformly at random.
- Let $f[0], f[1], f[2], f[3]$ be the frequencies of the symbols in i-th coordinate. Then,

$$\mathbb{E}[\tau(i)] \approx \sum_{a \neq b} f[a]\, f[b]\, (1 - f[a] - f[b]) \leq \frac{3}{8}.$$

- Use Hansel's lemma to obtain

$$\log_2(|C| - 2) \leq \sum_i \tau(i) \leq \left(\frac{3}{8}\right) n.$$

- Conclusion: $|C| \leq 2^{((3/8)+o(1))n}$.

The Arikan bound

- Pick two distinct $x, y \in C$ that agree on many coordinates.
(Coding theory says: high rate $\implies$ small distance, say Δn.)
- Use an *ad hoc* argument to argue that all four symbols appear in each coordinate with frequency at least f_0 (otherwise, the code must be small anyway).
- Use Hansel's lemma to conclude that

$$\log_2(|C| - 2) \leq \sum_i \tau(i) \leq \Delta(1 - 2f_0)n.$$

- Using bounds on distance in terms of rate gives $|C| \leq 2^{0.3512n}$ (using the Plotkin bound) and $|C| \leq 2^{0.3276n}$ (using the linear programming bound).

Looking deeper into the Plotkin bound

Recall that $C \subseteq \{0,1,2,3\}^n$ is a 4-hash code.

Main idea

Generate x, y more carefully, in a way reminiscent of the proof of the Plotkin bound.

- Fix $s \leq n$. Later we will set $s \approx \left(\frac{1}{2}\right) \log_2 |C|$.
- Pick $x \in C$ uniformly at random; let $w = x[1, \ldots, s]$ be the prefix of x of length s.
- Pick $y \in C$ distinct from x uniformly at random conditioned on $y[1, \ldots, s] = w$.
- This helps us derive a stronger conclusion from Hansel's lemma.

Analysis

- $\tau(i) = 0$ if $i = 1, 2, \ldots, s$.
- For $i = s+1, s+2, \ldots, n$, the quantity $\tau(i)$ is a random variable depending on our choice of x and y. We will show that $\mathbb{E}[\tau(i)] \leq \frac{3}{8}$.
- Use Hansel's lemma to conclude that

$$\log_2(|C| - 2) \leq \left(\frac{3}{8}\right)(n - s).$$

- Set $s \approx (\frac{1}{2}) \log_2 |C|$ and obtain

$$|C| \leq 2^{(6/19)n}.$$

Analysis

- $\tau(i) = 0$ if $i = 1, 2, \ldots, s$.
- For $i = s + 1, s + 2, \ldots, n$, the quantity $\tau(i)$ is a random variable depending on our choice of x and y. We will show that $\mathbb{E}[\tau(i)] \leq \frac{3}{8}$.
- Use Hansel's lemma to conclude that

$$\log_2(|C| - 2) \leq \left(\frac{3}{8}\right)(n - s).$$

- Set $s \approx (\frac{1}{2}) \log_2 |C|$ and obtain

$$|C| \leq 2^{(6/19)n}.$$

Analysis

- $\tau(i) = 0$ if $i = 1, 2, \ldots, s$.
- For $i = s + 1, s + 2, \ldots, n$, the quantity $\tau(i)$ is a random variable depending on our choice of x and y. We will show that $\mathbb{E}[\tau(i)] \leq \frac{3}{8}$.
- Use Hansel's lemma to conclude that

$$\log_2(|C| - 2) \leq \left(\frac{3}{8}\right)(n - s).$$

- Set $s \approx (\frac{1}{2}) \log_2 |C|$ and obtain

$$|C| \leq 2^{(6/19)n}.$$

Analysis

- $\tau(i) = 0$ if $i = 1, 2, \ldots, s$.
- For $i = s+1, s+2, \ldots, n$, the quantity $\tau(i)$ is a random variable depending on our choice of x and y. We will show that $\mathbb{E}[\tau(i)] \leq \frac{3}{8}$.
- Use Hansel's lemma to conclude that

$$\log_2(|C| - 2) \leq \left(\frac{3}{8}\right)(n - s).$$

- Set $s \approx (\frac{1}{2}) \log_2 |C|$ and obtain

$$|C| \leq 2^{(6/19)n}.$$

In expectation

For each prefix $w \in \{0,1,2,3\}^s$, let C_w be the subcode

$$C_w = \{x \in C : x[1,\ldots,s] = w\}.$$

For $i > s$, let $f_{i,w}[a]$ be the frequency of the symbol a in the i-th coordinate in the subcode C_w; let $\tau(i|w) = \sum_{a \neq b} f_{i,w}[a] f_{i,w}[b](1 - f_i[a] - f_i[b])$. Then,

$$\mathbb{E}[f_{i,W}] = f_i \qquad \text{and} \qquad E[\tau(i)] = \mathbb{E}_W[\phi(f_{i,W}, f_i)],$$

where $\phi(f', f) = \sum_{a \neq b} f'[a] f'[b](1 - f[a] - f[b])$.

Claim

Suppose $\mathbb{E}[f_W] = f$ and each component of f is at most $\frac{1}{2}$. Then, $\mathbb{E}_W[\phi(f_W, f)] \leq \phi(f, f) \leq \frac{3}{8}$.

Must ϕ meet our expectation?

Claim

Suppose $\mathbb{E}[f_W] = f$ *and each component of* f *is at most* $\frac{1}{2}$*. Then,*

$$\mathop{\mathbb{E}}_{W}[\phi(f_W, f)] \le \phi(f, f) \le \frac{3}{8}.$$

- Let $\Delta_w = f_w - f$; then, $\mathbb{E}_W[\phi(f_W, f)] = \phi(f, f) - \mathbb{E}_W[\Delta_W^t N \Delta_W]$, where

$$N = -\begin{pmatrix} 0 & f[2]+f[3] & f[1]+f[3] & f[1]+f[2] \\ f[2]+f[3] & 0 & f[0]+f[3] & f[0]+f[2] \\ f[1]+f[3] & f[0]+f[3] & 0 & f[0]+f[1] \\ f[1]+f[2] & f[0]+f[2] & f[0]+f[1] & 0 \end{pmatrix}.$$

- Now, $\mathbf{1} \cdot \Delta_w = 0$. In the subspace orthogonal to $\mathbf{1}$, the matrix N is positive semidefinite (because of our assumption on f).

What about the $q/(q-1)$ channel?

Let $R_{q,L}(n)$ be the size of the largest subset $C \subseteq \{0, 1, 2, \ldots, q-1\}^n$ such that every $L+1$ of them are **qifferent**?

$$\textbf{qifferent} \equiv \forall \text{ (distinct) } x_1, x_2, \ldots, x_{L+1} \in C \quad \exists i \in [L+1] : \\ \{x_1[i], x_2[i], \ldots, x_{L+1}[i]\} = \{0, 1, 2, \ldots, q-1\}.$$

Theorem (with Siddharth Bhandari '18)

For every $\epsilon < 1/6$, for all large q, if $L \leq \varepsilon q \ln q$, then

$$\frac{\log R_{q,L}(n)}{n} = O(\exp(-q^{1-6\epsilon}/8)).$$

Earlier, it was known that the rate of the code is exponentially small if the list size is bounded by $1.58q$ (Chakraborty, Radhakrishnan, Raghunathan and Sasatte '06).

Thank you!

Thank you!

Thank you!