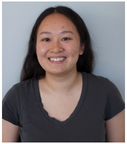
On the Maximum Size of Block Codes Subject to a Distance Criterion

Vincent Y. F. Tan

National University of Singapore (NUS)



Ling-Hua Chang — Yuan Ze Univ.

Po-Ning Chen — NCTU

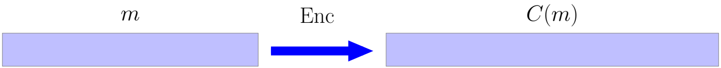
Carol Wang — NUS

Yunghsiang Han — Dongguan Univ. of Tech.

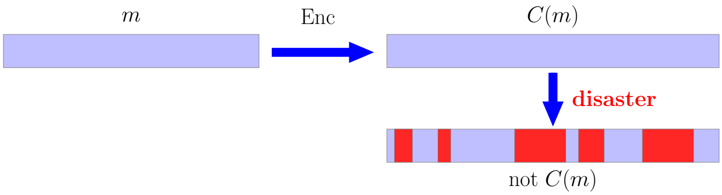
ITCom Workshop (Jan 2019)

Error-correcting codes

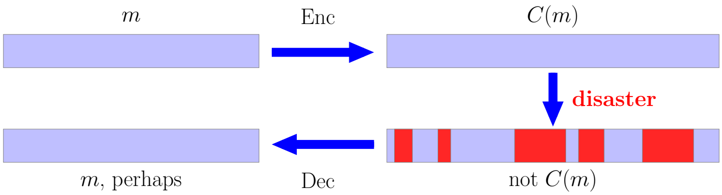
Error-correcting codes



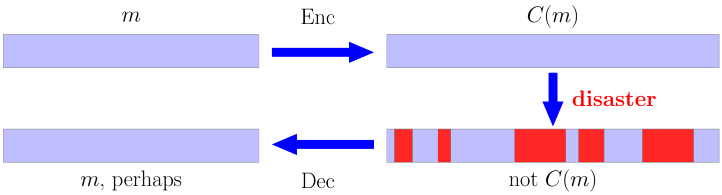
Error-correcting codes



Error-correcting codes



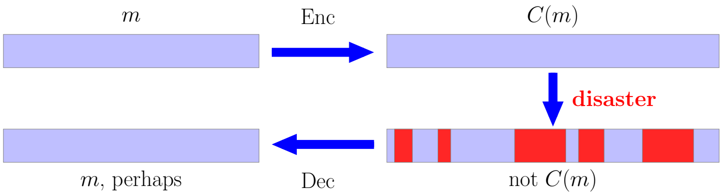
Error-correcting codes



"Message" m (k symbols) maps to "codeword" $C(m)$ ($n > k$ symbols).

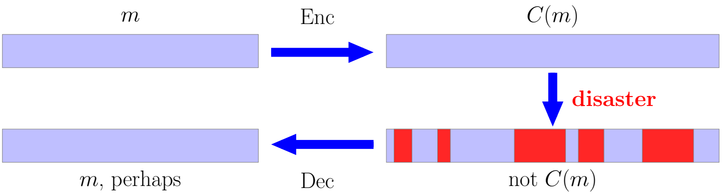
Set of codewords is a code $\mathcal{C}$.

Error-correcting codes



"Message" m (k symbols) maps to "codeword" $C(m)$ ($n > k$ symbols).

Set of codewords is a code $\mathcal{C}$.



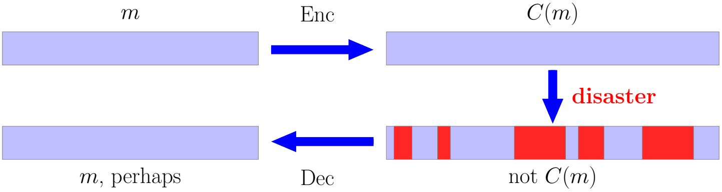
“Message” m (k symbols) maps to “codeword” $C(m)$ ($n > k$ symbols).

Set of codewords is a code $\mathcal{C}$.

Key parameters:

- Rate $\frac{1}{n} \log |\mathcal{C}|$: efficiency

Error-correcting codes



"Message" m (k symbols) maps to "codeword" $C(m)$ ($n > k$ symbols).

Set of codewords is a code $\mathcal{C}$.

Key parameters:

- Rate $\frac{1}{n} \log |\mathcal{C}|$: efficiency
- Distance : error-correction potential

Distance and errors

Distance and errors

Distance: "How many errors do we need to turn $\mathbf{x}$ into $\mathbf{y}$?"

Distance and errors

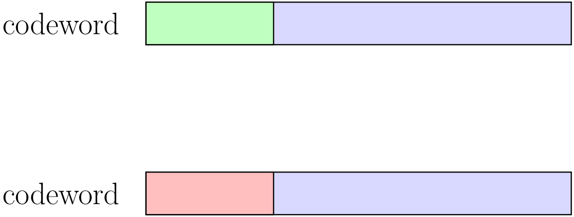
Distance: "How many errors do we need to turn $\mathbf{x}$ into $\mathbf{y}$?"

Can correct as many errors as half the distance:

Distance and errors

Distance: “How many errors do we need to turn $\mathbf{x}$ into $\mathbf{y}$?”

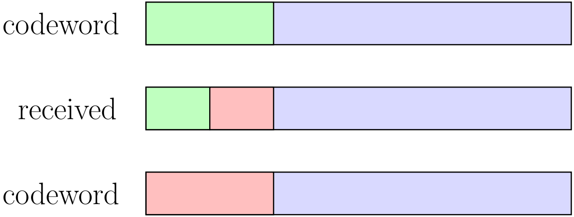
Can correct as many errors as half the distance:



Distance and errors

Distance: "How many errors do we need to turn $\mathbf{x}$ into $\mathbf{y}$?"

Can correct as many errors as half the distance:



Distance

Distance

Different "distances" for different applications.

Distance

Different "distances" for different applications.

$$\mu(\mathbf{x},\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x_i \neq y_i\} \qquad \text{(Hamming distance)}$$

Distance

Different "distances" for different applications.

$$\mu(\mathbf{x},\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x_i \neq y_i\}$$ (Hamming distance)

$$\mu(\mathbf{x},\mathbf{y}) = \begin{cases} 0 & \mathbf{x}=\mathbf{y} \\ 1 & \text{else} \end{cases}$$ (Probability-of-error distortion)

Distance

Different "distances" for different applications.

$$\mu(\mathbf{x},\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x_i \neq y_i\} \qquad \text{(Hamming distance)}$$

$$\mu(\mathbf{x},\mathbf{y}) = \begin{cases} 0 & \mathbf{x} = \mathbf{y} \\ 1 & \text{else} \end{cases} \qquad \text{(Probability-of-error distortion)}$$

$$\mu(\mathbf{x},\mathbf{y}) = \text{pretty much anything!}$$

Distance

Different "distances" for different applications.

$$\mu(\mathbf{x},\mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{x_i \neq y_i\} \qquad \text{(Hamming distance)}$$

$$\mu(\mathbf{x},\mathbf{y}) = \begin{cases} 0 & \mathbf{x}=\mathbf{y} \\ 1 & \text{else} \end{cases} \qquad \text{(Probability-of-error distortion)}$$

$$\mu(\mathbf{x},\mathbf{y}) = \text{pretty much anything!}$$

(deletion distance, rank-metric, etc)

Coding and the distance problem

Coding and the distance problem

Question: What is the optimal rate–distance trade-off?

Coding and the distance problem

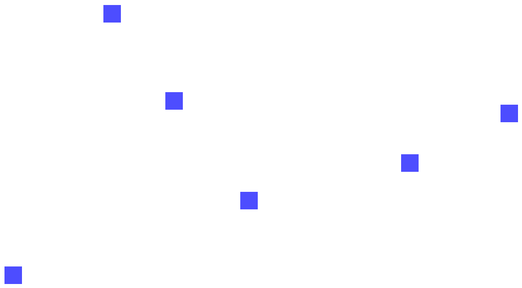
Question: What is the optimal rate–distance trade-off?
In other words, for fixed d, what is the largest size of a distance d code?

Coding and the distance problem

Question: What is the optimal rate–distance trade-off?
In other words, for fixed d, what is the largest size of a distance d code?

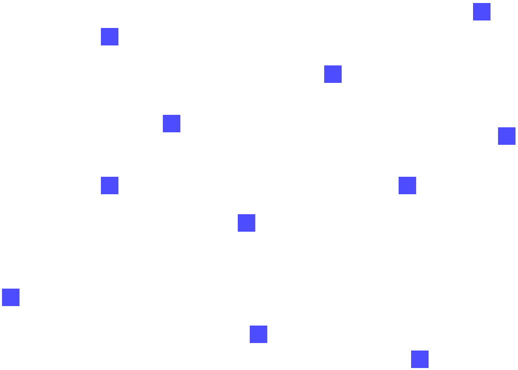
Coding and the distance problem

Question: What is the optimal rate–distance trade-off?
In other words, for fixed d, what is the largest size of a distance d code?



Coding and the distance problem

Question: What is the optimal rate–distance trade-off?
In other words, for fixed d, what is the largest size of a distance d code?



The GV bound and good codes

The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0, 1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy.

The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

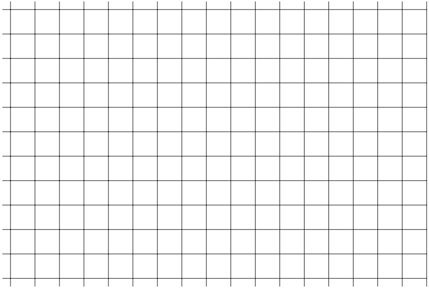
Proof 1: Greedy. Pick codewords at distance d until you can't.

The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

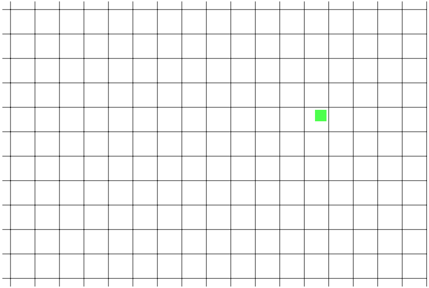


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

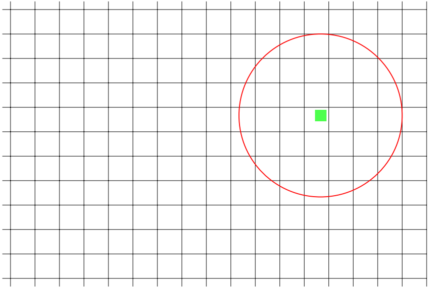


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

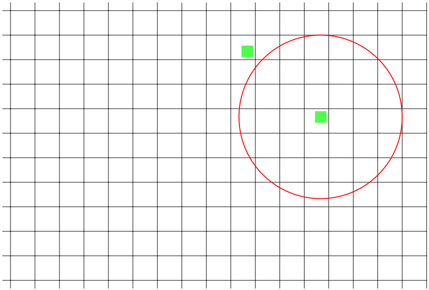


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

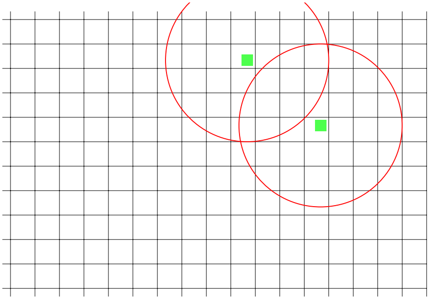


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

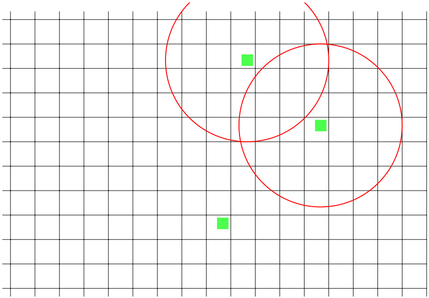


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

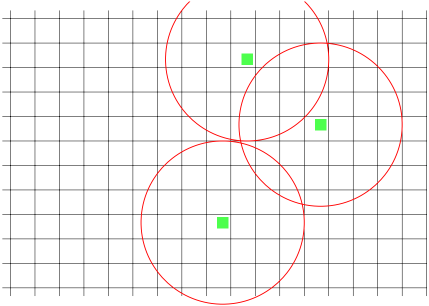


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

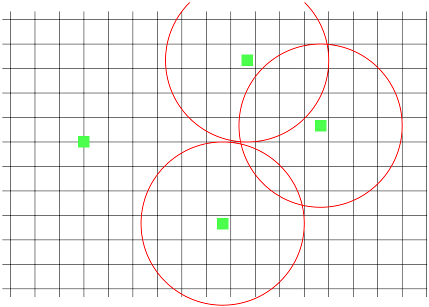


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

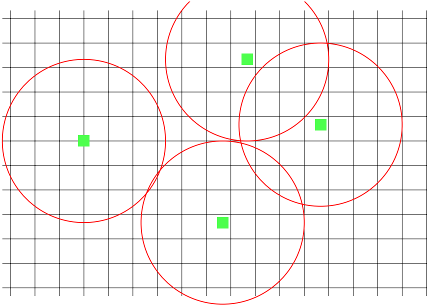


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

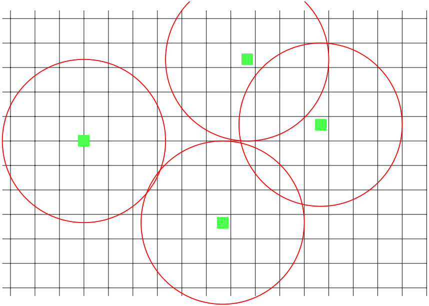


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.

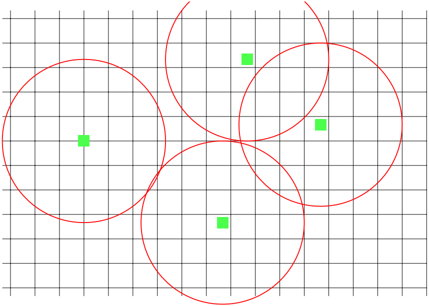


The GV bound and good codes

Theorem (Gilbert-Varshamov bound)

$\exists$ *codes in* $\{0,1\}^n$ *with Hamming distance* $d = \delta n$ *and rate* $\approx 1 - H(\delta)$.

Proof 1: Greedy. Pick codewords at distance d until you can't.



Each circle has $\approx 2^{H(\delta)n}$ vectors, so final code size is $2^n / 2^{H(\delta)n}$.

GV continued

GV continued

Proof 2: Random [Barg and Forney (2002)].

GV continued

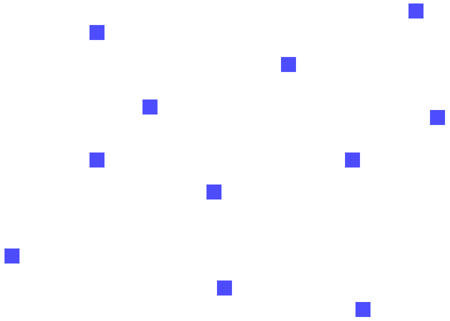
Proof 2: Random [Barg and Forney (2002)].

Pick i.i.d. codewords uniformly from $\{0, 1\}^n$.

GV continued

Proof 2: Random [Barg and Forney (2002)].

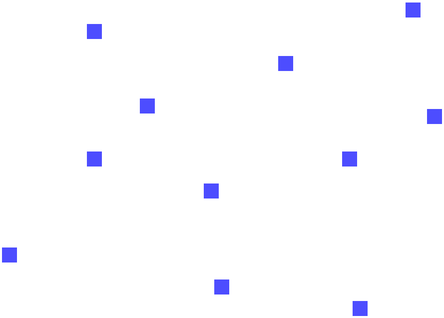
Pick i.i.d. codewords uniformly from $\{0,1\}^n$.



GV continued

Proof 2: Random [Barg and Forney (2002)].

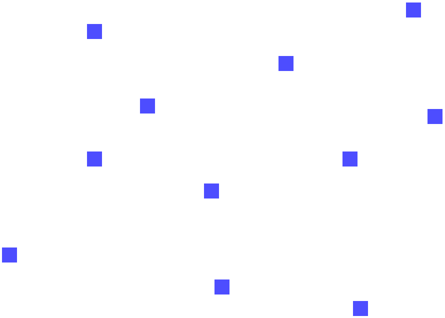
Pick i.i.d. codewords uniformly from $\{0,1\}^n$.



GV continued

Proof 2: Random [Barg and Forney (2002)].

Pick i.i.d. codewords uniformly from $\{0, 1\}^n$.



Works for rate $R \approx 1 - H(\delta)$ (proof on next slide).

GV continued

GV continued

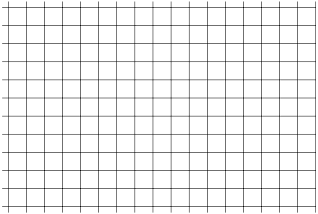
Proof 2: Random.

GV continued

Proof 2: Random. Let $R = 1 - H(\delta) - \epsilon$.

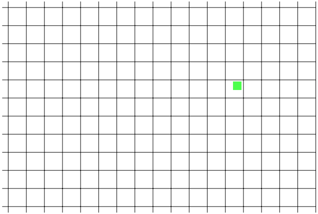
GV continued

Proof 2: Random. Let $R = 1 - H(\delta) - \epsilon$.



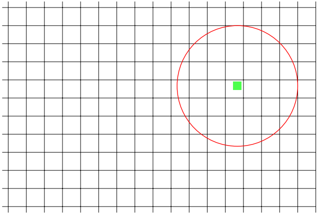
GV continued

Proof 2: Random. Let $R = 1 - H(\delta) - \epsilon$.



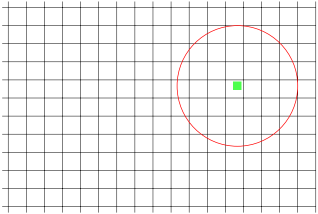
GV continued

Proof 2: Random. Let $R = 1 - H(\delta) - \epsilon$.



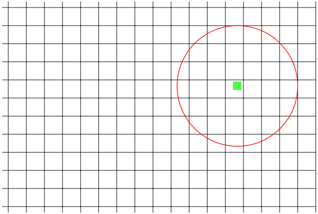
GV continued

Proof 2: Random. Let $R = 1 - H(\delta) - \epsilon$.



GV continued

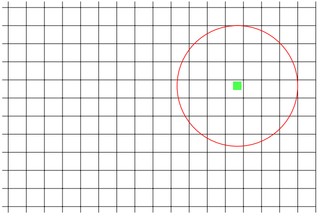
Proof 2: Random. Let $R = 1 - H(\delta) - \epsilon$.



Look at collision probability $\Pr[\mu(\mathbf{X}, \mathbf{Y}) < \delta n] = 2^{H(\delta)n}/2^n$.

GV continued

Proof 2: Random. Let $R = 1 - H(\delta) - \epsilon$.



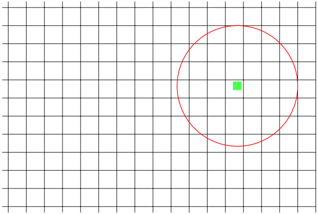
Look at collision probability $\Pr[\mu(\mathbf{X}, \mathbf{Y}) < \delta n] = 2^{H(\delta)n}/2^n$.

Number of "bad" pairs $(\mathbf{x}, \mathbf{y})$ is

$$\approx 2^{2Rn} \cdot \frac{2^{H(\delta)n}}{2^n} = 2^{(R-\epsilon)n}.$$

GV continued

Proof 2: Random. Let $R = 1 - H(\delta) - \epsilon$.



Look at collision probability $\Pr[\mu(\mathbf{X}, \mathbf{Y}) < \delta n] = 2^{H(\delta)n}/2^n$.

Number of "bad" pairs $(\mathbf{x}, \mathbf{y})$ is

$$\approx 2^{2Rn} \cdot \frac{2^{H(\delta)n}}{2^n} = 2^{(R-\epsilon)n}.$$

Remove one element from each bad pair.

Distance is now δ, and rate is still $\approx R$.

Extending GV

Extending GV

Tightness of the GV bound is a major open question!

Extending GV

Tightness of the GV bound is a major open question!

This work: What if we don't use the *uniform* distribution in the random proof?

Extending GV

Tightness of the GV bound is a major open question!

This work: What if we don't use the *uniform* distribution in the random proof?

(Could imagine: supported on structured set, mixing distributions.)

Extending GV

Tightness of the GV bound is a major open question!

This work: What if we don't use the *uniform* distribution in the random proof?

(Could imagine: supported on structured set, mixing distributions.)

To mimic the GV proof, need to understand collision probability.

Extending GV

Tightness of the GV bound is a major open question!

This work: What if we don't use the *uniform* distribution in the random proof?

(Could imagine: supported on structured set, mixing distributions.)

To mimic the GV proof, need to understand collision probability.

When are two random codewords at distance $< d$?

In other words. . .

In other words...

Moral: For various $\mathbf{X}$, want to understand collision probability (distance spectrum):

$$F_{\mathbf{X}}(d) := \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right],$$

where $\hat{\mathbf{X}}$ is an independent copy of $\mathbf{X}$.

In other words...

Moral: For various $\mathbf{X}$, want to understand collision probability (distance spectrum):

$$F_{\mathbf{X}}(d) := \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right],$$

where $\hat{\mathbf{X}}$ is an independent copy of $\mathbf{X}$.

Example. $\mathbf{X}$ uniform over a code $\mathcal{C}$ of distance d.

In other words. . .

Moral: For various $\mathbf{X}$, want to understand collision probability (distance spectrum):

$$F_{\mathbf{X}}(d) := \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right],$$

where $\hat{\mathbf{X}}$ is an independent copy of $\mathbf{X}$.

Example. $\mathbf{X}$ uniform over a code $\mathcal{C}$ of distance d.

$$F_{\mathbf{X}}(d) = \Pr[\mathbf{X} = \hat{\mathbf{X}}]$$

In other words...

Moral: For various $\mathbf{X}$, want to understand collision probability (distance spectrum):

$$F_{\mathbf{X}}(d) := \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right],$$

where $\hat{\mathbf{X}}$ is an independent copy of $\mathbf{X}$.

Example. $\mathbf{X}$ uniform over a code $\mathcal{C}$ of distance d.

$$\begin{aligned} F_{\mathbf{X}}(d) &= \Pr[\mathbf{X} = \hat{\mathbf{X}}] \\ &= \sum_{\mathbf{x} \in \mathcal{C}} \big(P_{\mathbf{X}}(\mathbf{x})\big)^2 \end{aligned}$$

In other words...

Moral: For various $\mathbf{X}$, want to understand collision probability (distance spectrum):

$$F_{\mathbf{X}}(d) := \Pr\big[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\big],$$

where $\hat{\mathbf{X}}$ is an independent copy of $\mathbf{X}$.

Example. $\mathbf{X}$ uniform over a code $\mathcal{C}$ of distance d.

$$\begin{aligned} F_{\mathbf{X}}(d) &= \Pr[\mathbf{X} = \hat{\mathbf{X}}] \\ &= \sum_{\mathbf{x} \in \mathcal{C}} \big(P_{\mathbf{X}}(\mathbf{x})\big)^2 \\ &= \frac{1}{|\mathcal{C}|}. \end{aligned}$$

Exact distance spectrum formula

Exact distance spectrum formula

So, if $\mathbf{X}$ is uniform over $\mathcal{C}$, then

$$|\mathcal{C}| = \frac{1}{F_{\mathbf{X}}(d)}.$$

Exact distance spectrum formula

So, if $\mathbf{X}$ is uniform over $\mathcal{C}$, then

$$|\mathcal{C}| = \frac{1}{F_{\mathbf{X}}(d)}.$$

In fact, this is tight.

Exact distance spectrum formula

So, if $\mathbf{X}$ is uniform over $\mathcal{C}$, then

$$|\mathcal{C}| = \frac{1}{F_{\mathbf{X}}(d)}.$$

In fact, this is tight.

Theorem (Main theorem)

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]}.$$

Exact distance spectrum formula

So, if $\mathbf{X}$ is uniform over $\mathcal{C}$, then

$$|\mathcal{C}| = \frac{1}{F_{\mathbf{X}}(d)}.$$

In fact, this is tight.

Theorem (Main theorem)

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\big[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\big]}.$$

Key points:

Exact distance spectrum formula

So, if $\mathbf{X}$ is uniform over $\mathcal{C}$, then

$$|\mathcal{C}| = \frac{1}{F_{\mathbf{X}}(d)}.$$

In fact, this is tight.

Theorem (Main theorem)

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]}.$$

Key points:

- No asymptotics!

Exact distance spectrum formula

So, if $\mathbf{X}$ is uniform over $\mathcal{C}$, then

$$|\mathcal{C}| = \frac{1}{F_{\mathbf{X}}(d)}.$$

In fact, this is tight.

Theorem (Main theorem)

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]}.$$

Key points:

- No asymptotics!
- Exact formula for basically any distance measure.

Exact distance spectrum formula

So, if $\mathbf{X}$ is uniform over $\mathcal{C}$, then

$$|\mathcal{C}| = \frac{1}{F_{\mathbf{X}}(d)}.$$

In fact, this is tight.

Theorem (Main theorem)

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]}.$$

Key points:

- No asymptotics!
- Exact formula for basically any distance measure.
- Holds for arbitrary (non-discrete) alphabets.

Remarks on the result

Remarks on the result

Theorem

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\big[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\big]}.$$

Remarks on the result

Theorem

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\big[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\big]}.$$

- Turns question about codes into one about distributions.

Remarks on the result

Theorem

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\big[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\big]}.$$

- Turns question about codes into one about distributions.
- Allows us to use optimization techniques for distributions.

Remarks on the result

Theorem

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]}.$$

- Turns question about codes into one about distributions.
- Allows us to use optimization techniques for distributions.
- New bounds on the second-order asymptotics.

Remarks on the result

Theorem

Let $M^*(d)$ be the optimal size of a distance d code. Then

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\big[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\big]}.$$

- Turns question about codes into one about distributions.
- Allows us to use optimization techniques for distributions.
- New bounds on the second-order asymptotics.
- Best distribution is uniform over optimal code, but any distribution gives a lower bound.

Proof for Discrete Case

Proof for Discrete Case

For a fixed random vector $\mathbf{X}$, want to show:

$$F_{\mathbf{X}}(d) = \Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] \geq \frac{1}{M^*(d)}.$$

Proof for Discrete Case

For a fixed random vector $\mathbf{X}$, want to show:

$$F_{\mathbf{X}}(d) = \Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] \geq \frac{1}{M^*(d)}.$$

Two steps:

1. If $|\mathrm{supp}(\mathbf{X})| = M \leq M^*(d)$, then

$$F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}.$$

Proof for Discrete Case

For a fixed random vector $\mathbf{X}$, want to show:

$$F_{\mathbf{X}}(d) = \Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] \geq \frac{1}{M^*(d)}.$$

Two steps:

1. If $|\mathrm{supp}(\mathbf{X})| = M \leq M^*(d)$, then

$$F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}.$$

2. If $M > M^*(d)$, can reduce to first case.

Step 1: small support

Step 1: small support

We have

$$\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right] \geq \sum_{\mathbf{x} \in \mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x})^2.$$

Step 1: small support

We have

$$\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right] \geq \sum_{\mathbf{x} \in \mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x})^2.$$

Assume $|\mathrm{supp}(\mathbf{X})| = M \leq M^*(d)$.

Step 1: small support

We have

$$\Pr\left[\mu(\mathbf{X},\hat{\mathbf{X}}) < d\right] \geq \sum_{\mathbf{x}\in\mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x})^2.$$

Assume $|\mathrm{supp}(\mathbf{X})| = M \leq M^*(d)$. Then

$$\frac{1}{M}\sum_{\mathbf{x}\in\mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{M}.$$

Step 1: small support

We have

$$\Pr\left[\mu(\mathbf{X},\hat{\mathbf{X}}) < d\right] \geq \sum_{\mathbf{x}\in\mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x})^2.$$

Assume $|\mathrm{supp}(\mathbf{X})| = M \leq M^*(d)$. Then

$$\frac{1}{M} \sum_{\mathbf{x}\in\mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{M}.$$

By Cauchy-Schwartz,

$$\sum_{\mathbf{x}\in\mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x})^2 \geq \sum_{\mathbf{x}\in\mathrm{supp}(\mathbf{X})} \frac{1}{M^2} = \frac{1}{M} \geq \frac{1}{M^*(d)}.$$

Step 1: small support

We have

$$\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right] \geq \sum_{\mathbf{x} \in \mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x})^2.$$

Assume $|\mathrm{supp}(\mathbf{X})| = M \leq M^*(d)$. Then

$$\frac{1}{M} \sum_{\mathbf{x} \in \mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{M}.$$

By Cauchy-Schwartz,

$$\sum_{\mathbf{x} \in \mathrm{supp}(\mathbf{X})} P_{\mathbf{X}}(\mathbf{x})^2 \geq \sum_{\mathbf{x} \in \mathrm{supp}(\mathbf{X})} \frac{1}{M^2} = \frac{1}{M} \geq \frac{1}{M^*(d)}.$$

So, for small support, uniform is best.

Step 2: large support

Step 2: large support

Showed that if $|\text{supp}(\mathbf{X})|$ is small, $F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}$.

Step 2: large support

Showed that if $|\mathrm{supp}(\mathbf{X})|$ is small, $F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}$.

Idea: If $|\mathrm{supp}(\mathbf{X})|$ is large, show how to reduce $|\mathrm{supp}(\mathbf{X})|$ without increasing $F_{\mathbf{X}}(d)$.

Step 2: large support

Showed that if $|\mathrm{supp}(\mathbf{X})|$ is small, $F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}$.

Idea: If $|\mathrm{supp}(\mathbf{X})|$ is large, show how to reduce $|\mathrm{supp}(\mathbf{X})|$ without increasing $F_{\mathbf{X}}(d)$.

Specifically, we'll find $\mathbf{X}'$ with support size

$$|\mathrm{supp}(\mathbf{X})| - 1$$

and

$$F_{\mathbf{X}'}(d) \leq F_{\mathbf{X}}(d).$$

Step 2: large support

Showed that if $|\mathrm{supp}(\mathbf{X})|$ is small, $F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}$.

Idea: If $|\mathrm{supp}(\mathbf{X})|$ is large, show how to reduce $|\mathrm{supp}(\mathbf{X})|$ without increasing $F_{\mathbf{X}}(d)$.

Specifically, we'll find $\mathbf{X}'$ with support size

$$|\mathrm{supp}(\mathbf{X})| - 1$$

and

$$F_{\mathbf{X}'}(d) \leq F_{\mathbf{X}}(d).$$

If we iterate this until the support has size $M^*(d)$, then

$$F_{\mathbf{X}}(d) \geq F_{\mathbf{X}'}(d) \geq F_{\mathbf{X}''}(d) \geq \cdots \geq \frac{1}{M^*(d)}.$$

Large support cont.

Large support cont.

Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

Large support cont.

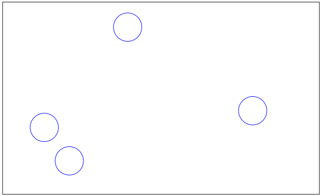
Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

Intuition $\Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] = \sum_{i,j} p_i p_j \mathbf{1}\{\mu(\mathbf{x}_i, \mathbf{x}_j) < d\}$ where $p_i = P_{\mathbf{X}}(\mathbf{x}_i)$

Large support cont.

Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

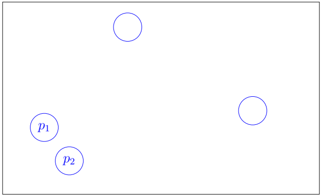
Intuition $\Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] = \sum_{i,j} p_i p_j \mathbf{1}\{\mu(\mathbf{x}_i, \mathbf{x}_j) < d\}$ where $p_i = P_{\mathbf{X}}(\mathbf{x}_i)$



Large support cont.

Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

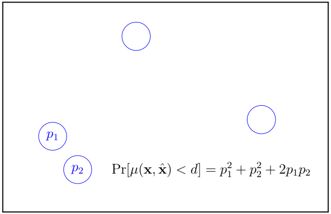
Intuition $\Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] = \sum_{i,j} p_i p_j \mathbf{1}\{\mu(\mathbf{x}_i, \mathbf{x}_j) < d\}$ where $p_i = P_{\mathbf{X}}(\mathbf{x}_i)$



Large support cont.

Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

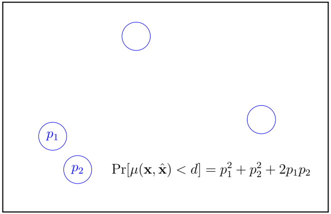
Intuition $\Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] = \sum_{i,j} p_i p_j \mathbf{1}\{\mu(\mathbf{x}_i, \mathbf{x}_j) < d\}$ where $p_i = P_{\mathbf{X}}(\mathbf{x}_i)$



Large support cont.

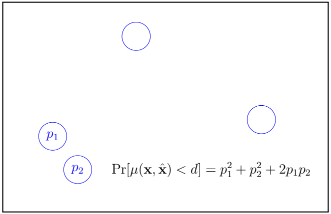
Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

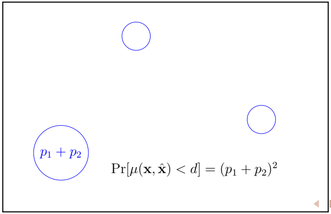
Intuition $\Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] = \sum_{i,j} p_i p_j \mathbf{1}\{\mu(\mathbf{x}_i, \mathbf{x}_j) < d\}$ where $p_i = P_{\mathbf{X}}(\mathbf{x}_i)$



Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

Intuition $\Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] = \sum_{i,j} p_i p_j \mathbf{1}\{\mu(\mathbf{x}_i, \mathbf{x}_j) < d\}$ where $p_i = P_{\mathbf{X}}(\mathbf{x}_i)$





Large support cont.

Large support cont.

Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

Large support cont.

Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

Proof.
If $|\mathrm{supp}(\mathbf{X})| > M^*(d)$, have $\mathbf{x}, \mathbf{y} \in \mathrm{supp}(\mathbf{X})$ at distance $< d$. Want to "combine" $\mathbf{x}, \mathbf{y}$.

Large support cont.

Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

Proof.
If $|\mathrm{supp}(\mathbf{X})| > M^*(d)$, have $\mathbf{x}, \mathbf{y} \in \mathrm{supp}(\mathbf{X})$ at distance $< d$. Want to "combine" $\mathbf{x}, \mathbf{y}$.

Question: Which of $\mathbf{x}, \mathbf{y}$ to keep?

Large support cont.

Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

Proof.
If $|\mathrm{supp}(\mathbf{X})| > M^*(d)$, have $\mathbf{x}, \mathbf{y} \in \mathrm{supp}(\mathbf{X})$ at distance $< d$. Want to "combine" $\mathbf{x}, \mathbf{y}$.

Question: Which of $\mathbf{x}, \mathbf{y}$ to keep?

Answer: "Furthest": Keep $\mathbf{x}$ if

$$\Pr\big[\mu(\mathbf{x}, \mathbf{X}) < d\big] \le \Pr\big[\mu(\mathbf{y}, \mathbf{X}) < d\big].$$

Large support cont.

Support reduction. Starting with distribution $\mathbf{X}$ on large support $M > M^*(d)$, want to construct $\mathbf{X}'$ on smaller support.

Proof.
If $|\mathrm{supp}(\mathbf{X})| > M^*(d)$, have $\mathbf{x}, \mathbf{y} \in \mathrm{supp}(\mathbf{X})$ at distance $< d$. Want to "combine" $\mathbf{x}, \mathbf{y}$.

Question: Which of $\mathbf{x}, \mathbf{y}$ to keep?

Answer: "Furthest": Keep $\mathbf{x}$ if

$$\Pr\left[\mu(\mathbf{x}, \mathbf{X}) < d\right] \leq \Pr\left[\mu(\mathbf{y}, \mathbf{X}) < d\right].$$

Keeps distance spectrum (collision probability) $F_{\mathbf{X}}(d)$ small.

Summary of Proof for Discrete Case

Summary of Proof for Discrete Case

For $\mathbf{X}$ with small support,

$$F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}.$$

Summary of Proof for Discrete Case

For $\mathbf{X}$ with small support,

$$F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}.$$

For other $\mathbf{X}$, can reduce support size.

Summary of Proof for Discrete Case

For $\mathbf{X}$ with small support,

$$F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}.$$

For other $\mathbf{X}$, can reduce support size.

Thus, optimal code size for distance d is

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]}.$$

Summary of Proof for Discrete Case

For $\mathbf{X}$ with small support,

$$F_{\mathbf{X}}(d) \geq \frac{1}{M^*(d)}.$$

For other $\mathbf{X}$, can reduce support size.

Thus, optimal code size for distance d is

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)} = \sup_{\mathbf{X}} \frac{1}{\Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]}.$$

(Upper bound via uniform distribution.)

An Algorithmic Construction

An Algorithmic Construction

"Support reduction" proof is (sort of) constructive.

An Algorithmic Construction

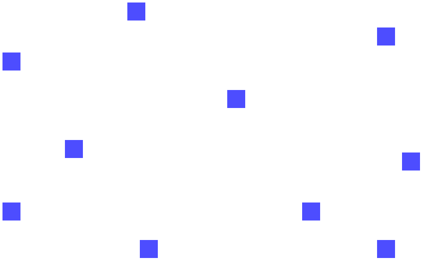
"Support reduction" proof is (sort of) constructive.

Start with any distribution, look at two codewords at distance $< d$, remove the one which is "closer" to the code.

An Algorithmic Construction

"Support reduction" proof is (sort of) constructive.

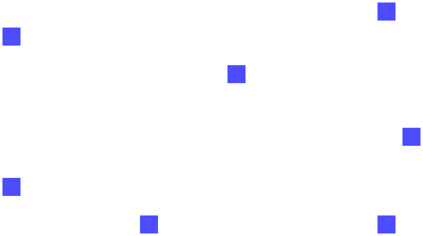
Start with any distribution, look at two codewords at distance $< d$, remove the one which is "closer" to the code.



An Algorithmic Construction

"Support reduction" proof is (sort of) constructive.

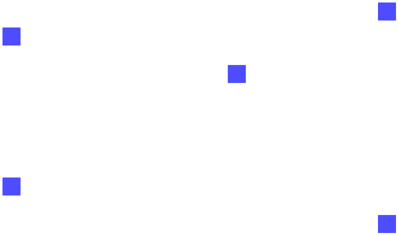
Start with any distribution, look at two codewords at distance $< d$, remove the one which is "closer" to the code.



An Algorithmic Construction

"Support reduction" proof is (sort of) constructive.

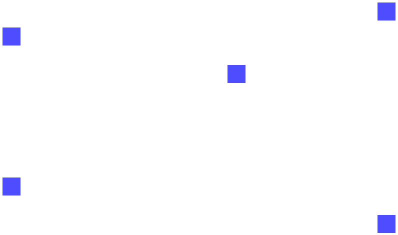
Start with any distribution, look at two codewords at distance $< d$, remove the one which is "closer" to the code.



An Algorithmic Construction

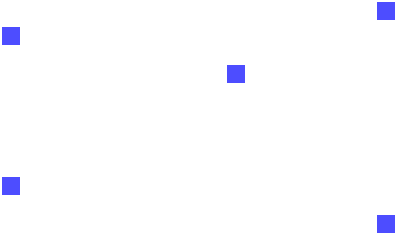
"Support reduction" proof is (sort of) constructive.

Start with any distribution, look at two codewords at distance $< d$, remove the one which is "closer" to the code.



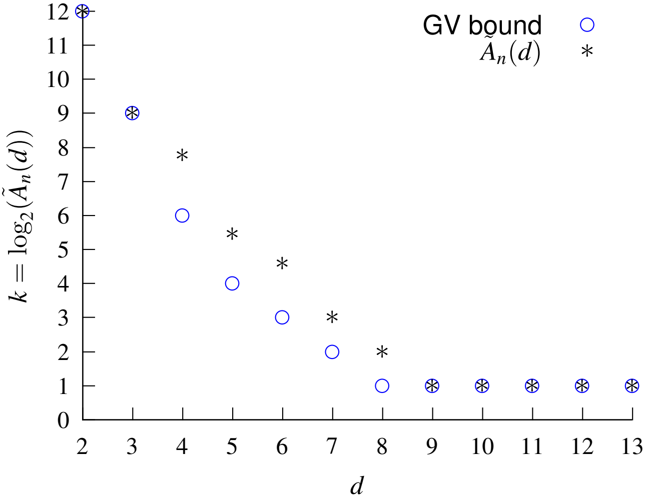
"Support reduction" proof is (sort of) constructive.

Start with any distribution, look at two codewords at distance $< d$, remove the one which is "closer" to the code.



Can be thought of as a different way to implement GV greedy construction. Seems to work well in simulations.

An Algorithmic Construction $(n = 13)$



Generalization to Non-Discrete Alphabets

Generalization to Non-Discrete Alphabets

- Previous achievability proof only works for discrete (finite) alphabets because we used $\mathrm{supp}(\mathbf{X})$.

Generalization to Non-Discrete Alphabets

- Previous achievability proof only works for discrete (finite) alphabets because we used $\mathrm{supp}(\mathbf{X})$.

- Sort of similar to Motzkin-Strass (1965) and Korn (1968)

1. T. S. Motzkin and E. G. Straus, “Maxima for graphs and a new proof of a theorem of Turan,” Canad. J. Math, vol. 17, no. 4, pp. 533–540, 1965.
2. I. Korn, “On the lower bound of zero-error capacity,” IEEE Trans. Inf. Theory, vol. 40, no. 4, pp. 509–510, May 1968.

Generalization to Non-Discrete Alphabets

- Previous achievability proof only works for discrete (finite) alphabets because we used $\mathrm{supp}(\mathbf{X})$.

- Sort of similar to Motzkin-Strass (1965) and Korn (1968)

  1. T. S. Motzkin and E. G. Straus, "Maxima for graphs and a new proof of a theorem of Turan," Canad. J. Math, vol. 17, no. 4, pp. 533–540, 1965.
  2. I. Korn, "On the lower bound of zero-error capacity," IEEE Trans. Inf. Theory, vol. 40, no. 4, pp. 509–510, May 1968.

- We now generalize to the case in which $|\mathcal{X}| = \infty$ (even uncountable)

Generalization to Non-Discrete Alphabets

- Previous achievability proof only works for discrete (finite) alphabets because we used $\mathrm{supp}(\mathbf{X})$.
- Sort of similar to Motzkin-Strass (1965) and Korn (1968)

1. T. S. Motzkin and E. G. Straus, "Maxima for graphs and a new proof of a theorem of Turan," Canad. J. Math, vol. 17, no. 4, pp. 533–540, 1965.
2. I. Korn, "On the lower bound of zero-error capacity," IEEE Trans. Inf. Theory, vol. 40, no. 4, pp. 509–510, May 1968.

- We now generalize to the case in which $|\mathcal{X}| = \infty$ (even uncountable)
- Idea: Greedy selection of codewords $\{\mathbf{u}_i\}_{i=1}^k$ given a fixed random vector/distribution $\mathbf{X} \sim P_{\mathbf{X}}$.

Non-Discrete Code Alphabets: Illustration

Non-Discrete Code Alphabets: Illustration

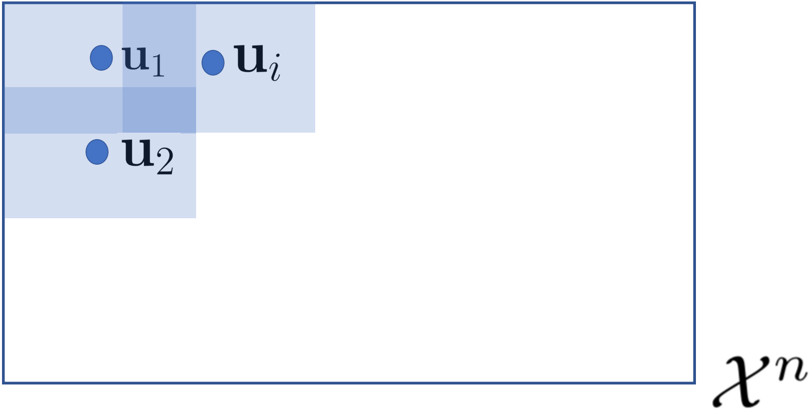
$$\mathbf{u}_1 = \arg\min_{\mathbf{u}_1} \Pr\left[\mathbf{X} \in \mathcal{B}_d(\mathbf{u}_1)\right]$$

$$\mathbf{u}_1 = \arg\min_{\mathbf{u}_1} \Pr\left[\mathbf{X} \in \mathcal{B}_d(\mathbf{u}_1)\right]$$

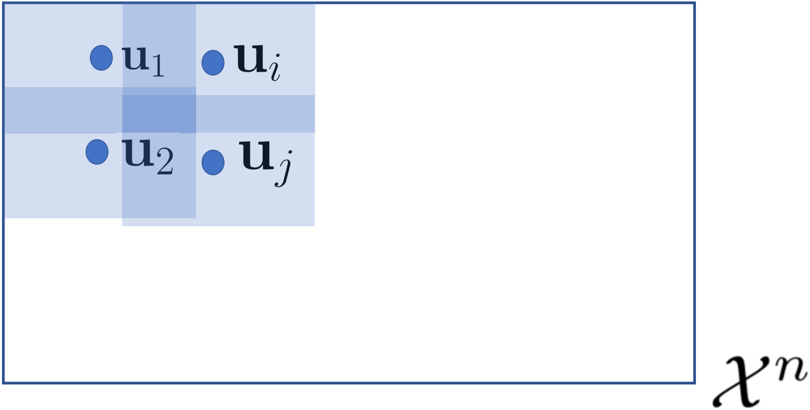
Non-Discrete Code Alphabets: Illustration

$$\mathbf{u}_2 = \arg\min_{\mathbf{u}_2} \Pr\left[\mathbf{X} \in \mathcal{B}_d(\mathbf{u}_2) \setminus \mathcal{B}_d(\mathbf{u}_1)\right]$$

Non-Discrete Code Alphabets: Illustration

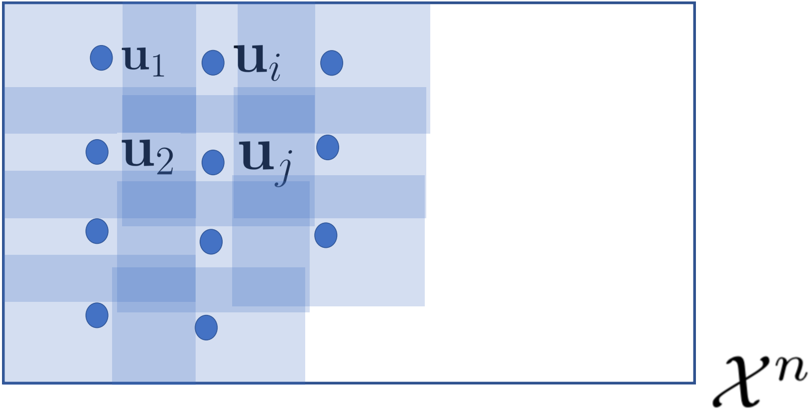


$$\mathbf{u}_i = \arg\min_{\mathbf{u}_i} \Pr\left[\mathbf{X} \in \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)\right]$$



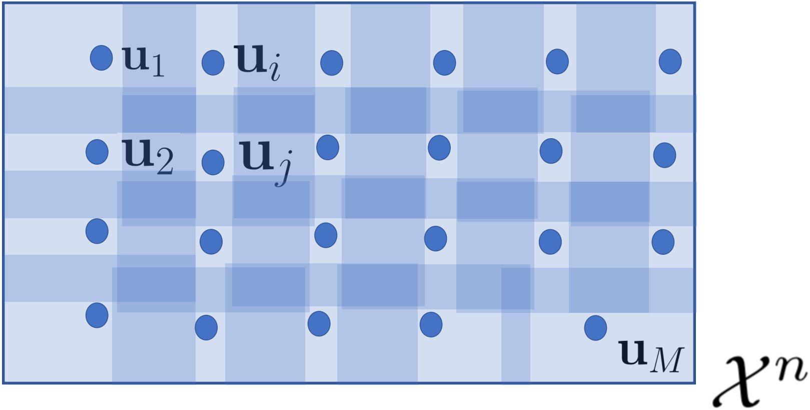
Choose more centers $\mathbf{u}_j$'s not in preceding balls.

Non-Discrete Code Alphabets: Illustration



And more balls...

Non-Discrete Code Alphabets: Illustration



Until you run out of space!

Non-Discrete Code Alphabets: Achievability Proof

Non-Discrete Code Alphabets: Achievability Proof

The code $\mathcal{C} = \{\mathbf{u}_i : i = 1, \ldots, M\}$ formed is a distance-d code and

$$p_j := \Pr\left[\mathbf{X} \in \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)\right], \quad \text{satisfies} \quad \sum_{j=1}^{M} p_j = 1.$$

Non-Discrete Code Alphabets: Achievability Proof

The code $\mathcal{C} = \{\mathbf{u}_i : i = 1, \ldots, M\}$ formed is a distance-d code and

$$p_j := \Pr\left[\mathbf{X} \in \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)\right], \quad \text{satisfies} \quad \sum_{j=1}^{M} p_j = 1.$$

Let $\mathcal{D}_i := \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)$ and note that $\{\mathcal{D}_i\}$ forms a partition of $\mathcal{X}^n$.

Non-Discrete Code Alphabets: Achievability Proof

The code $\mathcal{C} = \{\mathbf{u}_i : i = 1, \ldots, M\}$ formed is a distance-d code and

$$p_j := \Pr\left[\mathbf{X} \in \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)\right], \quad \text{satisfies} \quad \sum_{j=1}^{M} p_j = 1.$$

Let $\mathcal{D}_i := \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)$ and note that $\{\mathcal{D}_i\}$ forms a partition of $\mathcal{X}^n$.

$$\Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] = \sum_{j=1}^{M} \int_{\mathbf{x} \in \mathcal{D}_j} \left(\int_{\hat{\mathbf{x}} \in \mathcal{B}_d(\mathbf{x})} dP_{\mathbf{X}}(\hat{\mathbf{x}}) \right) dP_{\mathbf{X}}(\mathbf{x}) \quad \because \mathbf{X} \perp\!\!\!\perp \hat{\mathbf{X}}$$

Non-Discrete Code Alphabets: Achievability Proof

The code $\mathcal{C} = \{\mathbf{u}_i : i = 1, \ldots, M\}$ formed is a distance-d code and

$$p_j := \Pr\left[\mathbf{X} \in \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)\right], \quad \text{satisfies} \quad \sum_{j=1}^{M} p_j = 1.$$

Let $\mathcal{D}_i := \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)$ and note that $\{\mathcal{D}_i\}$ forms a partition of $\mathcal{X}^n$.

$$\begin{aligned}
\Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] &= \sum_{j=1}^{M} \int_{\mathbf{x} \in \mathcal{D}_j} \left(\int_{\hat{\mathbf{x}} \in \mathcal{B}_d(\mathbf{x})} dP_{\mathbf{X}}(\hat{\mathbf{x}}) \right) dP_{\mathbf{X}}(\mathbf{x}) \quad \because \mathbf{X} \perp\!\!\!\perp \hat{\mathbf{X}} \\
&\geq \sum_{j=1}^{M} \int_{\mathbf{x} \in \mathcal{D}_j} p_j \, dP_{\mathbf{X}}(\mathbf{x}) \qquad \because \min_{\mathbf{x} \in \mathcal{D}_j} P_{\mathbf{X}}\{\mathcal{B}_d(\mathbf{x})\} \geq p_j
\end{aligned}$$

Non-Discrete Code Alphabets: Achievability Proof

The code $\mathcal{C} = \{\mathbf{u}_i : i = 1, \dots, M\}$ formed is a distance-d code and

$$p_j := \Pr\left[\mathbf{X} \in \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)\right], \quad \text{satisfies} \quad \sum_{j=1}^{M} p_j = 1.$$

Let $\mathcal{D}_i := \mathcal{B}_d(\mathbf{u}_i) \setminus \cup_{j=1}^{i-1} \mathcal{B}_d(\mathbf{u}_j)$ and note that $\{\mathcal{D}_i\}$ forms a partition of $\mathcal{X}^n$.

$$\begin{aligned}
\Pr[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d] &= \sum_{j=1}^{M} \int_{\mathbf{x} \in \mathcal{D}_j} \left(\int_{\hat{\mathbf{x}} \in \mathcal{B}_d(\mathbf{x})} dP_{\mathbf{X}}(\hat{\mathbf{x}}) \right) dP_{\mathbf{X}}(\mathbf{x}) \quad \because \mathbf{X} \perp\!\!\!\perp \hat{\mathbf{X}} \\
&\ge \sum_{j=1}^{M} \int_{\mathbf{x} \in \mathcal{D}_j} p_j \, dP_{\mathbf{X}}(\mathbf{x}) \quad \because \min_{\mathbf{x} \in \mathcal{D}_j} P_{\mathbf{X}}\{\mathcal{B}_d(\mathbf{x})\} \ge p_j \\
&\ge \sum_{j=1}^{M} p_j^2 \ge \frac{1}{M} \ge \frac{1}{M^*(d)} \quad \because \text{Cauchy-Schwarz } \& \ M \le M^*(d)
\end{aligned}$$

Summary of Proof for Non-Discrete Alphabets

Summary of Proof for Non-Discrete Alphabets

- Also used a greedy construction (à la Feinstein's lemma in information spectrum analysis)

Summary of Proof for Non-Discrete Alphabets

- Also used a greedy construction (à la Feinstein's lemma in information spectrum analysis)
- But we removed space $\mathcal{B}_d(\mathbf{u}_k) \subset \mathcal{X}^n$ successively instead of codewords successively.

Summary of Proof for Non-Discrete Alphabets

- Also used a greedy construction (à la Feinstein's lemma in information spectrum analysis)
- But we removed space $\mathcal{B}_d(\mathbf{u}_k) \subset \mathcal{X}^n$ successively instead of codewords successively.
- Showed through simple algebraic manipulations that for any $\mathbf{X}$,

$$F_{\mathbf{X}}(d) = \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right] \geq \frac{1}{M^*(d)}$$

Summary of Proof for Non-Discrete Alphabets

- Also used a greedy construction (à la Feinstein's lemma in information spectrum analysis)

- But we removed space $\mathcal{B}_d(\mathbf{u}_k) \subset \mathcal{X}^n$ successively instead of codewords successively.

- Showed through simple algebraic manipulations that for any $\mathbf{X}$,

$$F_{\mathbf{X}}(d) = \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right] \geq \frac{1}{M^*(d)} \quad \Longrightarrow \quad M^*(d) \geq \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)}.$$

Summary of Proof for Non-Discrete Alphabets

- Also used a greedy construction (à la Feinstein's lemma in information spectrum analysis)
- But we removed space $\mathcal{B}_d(\mathbf{u}_k) \subset \mathcal{X}^n$ successively instead of codewords successively.
- Showed through simple algebraic manipulations that for any $\mathbf{X}$,

$$F_{\mathbf{X}}(d) = \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right] \geq \frac{1}{M^*(d)} \quad \Longrightarrow \quad M^*(d) \geq \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)}.$$

- Converse part is the same as for discrete alphabets (hinges on uniform distribution over optimal code $\mathcal{C}^*$)

Summary of Proof for Non-Discrete Alphabets

- Also used a greedy construction (à la Feinstein's lemma in information spectrum analysis)
- But we removed space $\mathcal{B}_d(\mathbf{u}_k) \subset \mathcal{X}^n$ successively instead of codewords successively.
- Showed through simple algebraic manipulations that for any $\mathbf{X}$,

$$F_{\mathbf{X}}(d) = \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right] \ge \frac{1}{M^*(d)} \quad \Longrightarrow \quad M^*(d) \ge \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)}.$$

- Converse part is the same as for discrete alphabets (hinges on uniform distribution over optimal code $\mathcal{C}^*$)
- In summary,

$$M^*(d) = \sup_{\mathbf{X}} \frac{1}{F_{\mathbf{X}}(d)}$$

Refined Asymptotics I

Refined Asymptotics I

Corollary (Refined GV bound)

For the Hamming distance, the optimal code rate for distance δn is

$$R_n^*(\delta) \geq 1 - H(\delta) + \frac{\log n}{2n} + \Theta\left(\frac{1}{n}\right).$$

Refined Asymptotics I

Corollary (Refined GV bound)

For the Hamming distance, the optimal code rate for distance δn is

$$R_n^*(\delta) \geq 1 - H(\delta) + \frac{\log n}{2n} + \Theta\left(\frac{1}{n}\right).$$

Proof.

Let $\mathbf{X}$ be uniform on $\{0,1\}^n$.

Refined Asymptotics I

Corollary (Refined GV bound)

For the Hamming distance, the optimal code rate for distance δn *is*

$$R_n^*(\delta) \geq 1 - H(\delta) + \frac{\log n}{2n} + \Theta\left(\frac{1}{n}\right).$$

Proof.

Let $\mathbf{X}$ be uniform on $\{0,1\}^n$.

$$\Pr[\mu(\mathbf{X},\hat{\mathbf{X}}) < \delta n] = \Pr\left[\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{X_i \neq \hat{X}_i\} < \delta\right] \sim c \cdot \frac{2^{n[1-H(\delta)]}}{\sqrt{n}}.$$

Result follows using exact asymptotics for sums of i.i.d. variables. □

Refined Asymptotics I

Corollary (Refined GV bound)

For the Hamming distance, the optimal code rate for distance δn is

$$R_n^*(\delta) \geq 1 - H(\delta) + \frac{\log n}{2n} + \Theta\left(\frac{1}{n}\right).$$

Proof.

Let $\mathbf{X}$ be uniform on $\{0,1\}^n$.

$$\Pr[\mu(\mathbf{X},\hat{\mathbf{X}}) < \delta n] = \Pr\left[\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{X_i \neq \hat{X}_i\} < \delta\right] \sim c \cdot \frac{2^{n[1-H(\delta)]}}{\sqrt{n}}.$$

Result follows using exact asymptotics for sums of i.i.d. variables. □

Jiang and Vardy (2004) showed that the “second-order term” $\geq \frac{\log n}{n}$.

Refined Asymptotics II

Refined Asymptotics II

Corollary (Upper Bound on Rate)

For any arbitrary bounded distance measure, the optimal code rate for distance δn is

$$R_n^*(\delta) \le I_{X^n}(\delta) + O\left(\frac{1}{\sqrt{n}}\right).$$

where the large-deviations rate function is

$$I_{X^n}(a) := \sup_{\theta} \left\{ a\theta - \varphi_{X^n}(\theta) \right\}, \quad \text{and} \quad \varphi_X(\theta) := \log \mathbb{E}\left[e^{\theta \mu(X, \hat{X})} \right].$$

Refined Asymptotics II

Corollary (Upper Bound on Rate)

For any arbitrary bounded distance measure, the optimal code rate for distance δn is

$$R_n^*(\delta) \le I_{X^n}(\delta) + O\left(\frac{1}{\sqrt{n}}\right).$$

where the large-deviations rate function is

$$I_{X^n}(a) := \sup_{\theta} \{a\theta - \varphi_{X^n}(\theta)\}, \quad \textit{and} \quad \varphi_X(\theta) := \log \mathbb{E}\left[e^{\theta\mu(X,\hat{X})}\right].$$

Proof.

Careful tilting of probability distributions. □

First-Order Asymptotics

First-Order Asymptotics

Corollary (First-Order Asymptotics on Rate)

If the sequence of distance measures satisfies

$$\sup_{n\in\mathbb{N}} \max_{x^n,\hat{x}^n} \frac{1}{n}\mu(x^n,\hat{x}^n) < \infty,$$

then we have

$$\limsup_{n\to\infty} R_n^*(\delta) = \limsup_{n\to\infty} I_{X^n}(\delta), \quad \textit{and}$$

$$\liminf_{n\to\infty} R_n^*(\delta) = \liminf_{n\to\infty} I_{X^n}(\delta)$$

where the large-deviations rate function is

$$I_{X^n}(a) := \sup_{\theta} \{a\theta - \varphi_{X^n}(\theta)\}, \quad \textit{and} \quad \varphi_X(\theta) := \log \mathbb{E}\left[e^{\theta\mu(X,\hat{X})}\right].$$

New derivation of Hamming bound

New derivation of Hamming bound

Corollary (Hamming Bound for Finite $|\mathcal{X}|$)

$$M^*(d) \le \inf_{\epsilon>0} \frac{|\mathcal{X}|^n}{\left|\mathcal{B}_{(d-\epsilon)/2}(\mathbf{0})\right|} \le \frac{|\mathcal{X}|^n}{\left|\mathcal{B}_{\lfloor (d-1)/2 \rfloor}(\mathbf{0})\right|}$$

New derivation of Hamming bound

Corollary (Hamming Bound for Finite $|\mathcal{X}|$)

$$M^*(d) \le \inf_{\epsilon>0} \frac{|\mathcal{X}|^n}{\left|\mathcal{B}_{(d-\epsilon)/2}(\mathbf{0})\right|} \le \frac{|\mathcal{X}|^n}{\left|\mathcal{B}_{\lfloor (d-1)/2 \rfloor}(\mathbf{0})\right|}$$

Proof: (Due to V. Guruswami).

Let $e = (d-\epsilon)/2$. Then

$$|\mathcal{B}_e(\mathbf{0})| F_{\mathbf{X}}(d) = \sum_{\mathbf{x}} \sum_{\mathbf{y} \in \mathcal{B}_e(\mathbf{x})} P_{\mathbf{X}}(\mathbf{y}) \sum_{\mathbf{z}:\mu(\mathbf{x},\mathbf{z})<d} P_{\mathbf{X}}(\mathbf{z})$$

New derivation of Hamming bound

Corollary (Hamming Bound for Finite $|\mathcal{X}|$)

$$M^*(d) \le \inf_{\epsilon>0} \frac{|\mathcal{X}|^n}{\left|\mathcal{B}_{(d-\epsilon)/2}(\mathbf{0})\right|} \le \frac{|\mathcal{X}|^n}{\left|\mathcal{B}_{\lfloor (d-1)/2 \rfloor}(\mathbf{0})\right|}$$

Proof: (Due to V. Guruswami).

Let $e = (d-\epsilon)/2$. Then

$$|\mathcal{B}_e(\mathbf{0})| F_{\mathbf{X}}(d) = \sum_{\mathbf{x}} \sum_{\mathbf{y}\in\mathcal{B}_e(\mathbf{x})} P_{\mathbf{X}}(\mathbf{y}) \sum_{\mathbf{z}:\mu(\mathbf{x},\mathbf{z})<d} P_{\mathbf{X}}(\mathbf{z})$$

$$\ge \sum_{\mathbf{x}} \sum_{\mathbf{y}\in\mathcal{B}_e(\mathbf{x})} \sum_{\mathbf{z}\in\mathcal{B}_e(\mathbf{x})} P_{\mathbf{X}}(\mathbf{y}) P_{\mathbf{X}}(\mathbf{z})$$

New derivation of Hamming bound

Corollary (Hamming Bound for Finite $|\mathcal{X}|$)

$$M^*(d) \le \inf_{\epsilon>0} \frac{|\mathcal{X}|^n}{|\mathcal{B}_{(d-\epsilon)/2}(\mathbf{0})|} \le \frac{|\mathcal{X}|^n}{|\mathcal{B}_{\lfloor (d-1)/2 \rfloor}(\mathbf{0})|}$$

Proof: (Due to V. Guruswami).

Let $e = (d-\epsilon)/2$. Then

$$\begin{aligned}
|\mathcal{B}_e(\mathbf{0})| F_{\mathbf{X}}(d) &= \sum_{\mathbf{x}} \sum_{\mathbf{y}\in\mathcal{B}_e(\mathbf{x})} P_{\mathbf{X}}(\mathbf{y}) \sum_{\mathbf{z}:\mu(\mathbf{x},\mathbf{z})<d} P_{\mathbf{X}}(\mathbf{z}) \\
&\ge \sum_{\mathbf{x}} \sum_{\mathbf{y}\in\mathcal{B}_e(\mathbf{x})} \sum_{\mathbf{z}\in\mathcal{B}_e(\mathbf{x})} P_{\mathbf{X}}(\mathbf{y}) P_{\mathbf{X}}(\mathbf{z}) \\
&\overset{\text{CS}}{\ge} \left(\sum_{\mathbf{x}} \sum_{\mathbf{y}\in\mathcal{B}_e(\mathbf{x})} P_{\mathbf{X}}(\mathbf{y}) \right)^2 = \frac{|\mathcal{B}_e(\mathbf{0})|^2}{|\mathcal{X}|^n}
\end{aligned}$$

Related Work

Related Work

Distance-Spectrum Formulas on the Largest Minimum Distance of Block Codes

Po-Ning Chen, *Member, IEEE*, Tzong-Yow Lee, and Yunghsiang S. Han, *Member, IEEE*

***Abstract*—A general formula for the asymptotic largest minimum distance (in block length) of deterministic block codes under *generalized* distance functions (not necessarily additive, symmetric, and bounded) is presented. As revealed in the formula, the largest minimum distance can be fully determined by the ultimate statistical characteristics of the normalized distance function evaluated**

surable function on the "distance" between two code symbols, determine the asymptotic ratio, the largest minimum distance attainable among M selected codewords divided by the code block length n, as n tends to infinity, subject to a fixed rate $R \triangleq \log(M)/n$.

Related Work

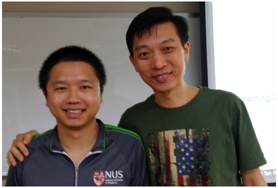
IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 46, NO. 3, MAY 2000 869

Distance-Spectrum Formulas on the Largest Minimum Distance of Block Codes

Po-Ning Chen, *Member, IEEE*, Tzong-Yow Lee, and Yunghsiang S. Han, *Member, IEEE*

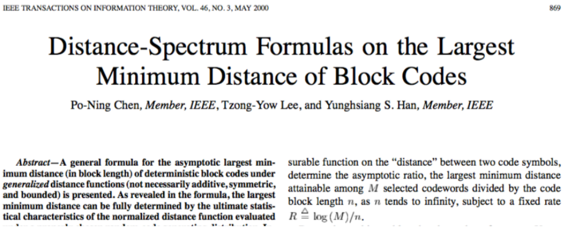
***Abstract*—A general formula for the asymptotic largest minimum distance (in block length) of deterministic block codes under *generalized* distance functions (not necessarily additive, symmetric, and bounded) is presented. As revealed in the formula, the largest minimum distance can be fully determined by the ultimate statistical characteristics of the normalized distance function evaluated**

surable function on the "distance" between two code symbols, determine the asymptotic ratio, the largest minimum distance attainable among M selected codewords divided by the code block length n, as n tends to infinity, subject to a fixed rate $R \triangleq \log(M)/n$.



My visit to NCTU 2015

Related Work

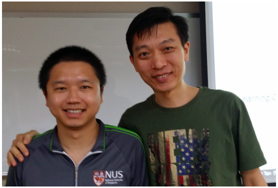

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 46, NO. 3, MAY 2000 869

Distance-Spectrum Formulas on the Largest Minimum Distance of Block Codes

Po-Ning Chen, *Member, IEEE*, Tzong-Yow Lee, and Yunghsiang S. Han, *Member, IEEE*

***Abstract*—A general formula for the asymptotic largest minimum distance (in block length) of deterministic block codes under *generalized* distance functions (not necessarily additive, symmetric, and bounded) is presented. As revealed in the formula, the largest minimum distance can be fully determined by the ultimate statistical characteristics of the normalized distance function evaluated**

surable function on the "distance" between two code symbols, determine the asymptotic ratio, the largest minimum distance attainable among M selected codewords divided by the code block length n, as n tends to infinity, subject to a fixed rate $R \triangleq \log(M)/n$.



My visit to NCTU 2015

- Chen, Lee and Han (2000) proved an elegant information spectrum-style result

Related Work

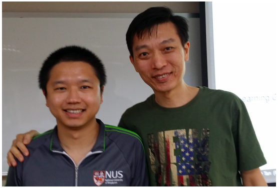
IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 46, NO. 3, MAY 2000 869

Distance-Spectrum Formulas on the Largest Minimum Distance of Block Codes

Po-Ning Chen, *Member, IEEE*, Tzong-Yow Lee, and Yunghsiang S. Han, *Member, IEEE*

***Abstract*—A general formula for the asymptotic largest minimum distance (in block length) of deterministic block codes under *generalized* distance functions (not necessarily additive, symmetric, and bounded) is presented. As revealed in the formula, the largest minimum distance can be fully determined by the ultimate statistical characteristics of the normalized distance function evaluated**

surable function on the "distance" between two code symbols, determine the asymptotic ratio, the largest minimum distance attainable among M selected codewords divided by the code block length n, as n tends to infinity, subject to a fixed rate $R \triangleq \log(M)/n$.

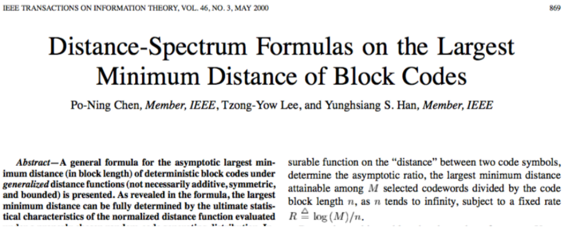


My visit to NCTU 2015

- Chen, Lee and Han (2000) proved an elegant information spectrum-style result

$$\limsup_{n\to\infty} \delta_n^*(2^{nR}) = \sup_{\mathbf{X}=\{X^n\}_{n=1}^{\infty}} \bar{\Lambda}_{\mathbf{X}}(R) \quad \text{(except at countably many points)}$$

Related Work

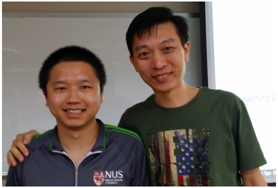

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 46, NO. 3, MAY 2000 869

Distance-Spectrum Formulas on the Largest Minimum Distance of Block Codes

Po-Ning Chen, *Member, IEEE*, Tzong-Yow Lee, and Yunghsiang S. Han, *Member, IEEE*

***Abstract*—A general formula for the asymptotic largest minimum distance (in block length) of deterministic block codes under *generalized* distance functions (not necessarily additive, symmetric, and bounded) is presented. As revealed in the formula, the largest minimum distance can be fully determined by the ultimate statistical characteristics of the normalized distance function evaluated**

surable function on the "distance" between two code symbols, determine the asymptotic ratio, the largest minimum distance attainable among M selected codewords divided by the code block length n, as n tends to infinity, subject to a fixed rate $R \triangleq \log(M)/n$.



My visit to NCTU 2015

- Chen, Lee and Han (2000) proved an elegant information spectrum-style result

$$\limsup_{n\to\infty} \delta_n^*(2^{nR}) = \sup_{\mathbf{X}=\{X^n\}_{n=1}^\infty} \overline{\Lambda}_{\mathbf{X}}(R) \quad \text{(except at countably many points)}$$

$$\overline{\Lambda}_{\mathbf{X}}(R) := \inf\left\{ a \in \mathbb{R} : \lim_{n\to\infty} \Pr\left[\mu(X^n, \hat{X}^n) > a\right]^{2^{nR}} = 0 \right\}.$$

Related Work

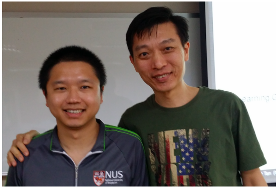
IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 46, NO. 3, MAY 2000 869

Distance-Spectrum Formulas on the Largest Minimum Distance of Block Codes

Po-Ning Chen, *Member, IEEE*, Tzong-Yow Lee, and Yunghsiang S. Han, *Member, IEEE*

***Abstract*—A general formula for the asymptotic largest minimum distance (in block length) of deterministic block codes under *generalized* distance functions (not necessarily additive, symmetric, and bounded) is presented. As revealed in the formula, the largest minimum distance can be fully determined by the ultimate statistical characteristics of the normalized distance function evaluated**

surable function on the "distance" between two code symbols, determine the asymptotic ratio, the largest minimum distance attainable among M selected codewords divided by the code block length n, as n tends to infinity, subject to a fixed rate $R \triangleq \log(M)/n$.



My visit to NCTU 2015

- Chen, Lee and Han (2000) proved an elegant information spectrum-style result

$$\limsup_{n\to\infty} \delta_n^*(2^{nR}) = \sup_{\mathbf{X}=\{X^n\}_{n=1}^{\infty}} \overline{\Lambda}_{\mathbf{X}}(R) \quad \text{(except at countably many points)}$$

$$\overline{\Lambda}_{\mathbf{X}}(R) := \inf\left\{ a \in \mathbb{R} : \lim_{n\to\infty} \Pr\left[\mu(X^n, \hat{X}^n) > a\right]^{2^{nR}} = 0 \right\}.$$

- The present result is a non-asymptotic version of CLH2000.

Conclusion

Conclusion

- Showed how to connect optimal code size/distance tradeoff and distance spectrum

$$F_{\mathbf{X}}(d) = \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]$$

for different random vectors $\mathbf{X}$.

Conclusion

- Showed how to connect optimal code size/distance tradeoff and distance spectrum

$$F_{\mathbf{X}}(d) = \Pr\big[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\big]$$

  for different random vectors $\mathbf{X}$.
- Also got an algorithm for constructing codes.

Conclusion

- Showed how to connect optimal code size/distance tradeoff and distance spectrum

$$F_{\mathbf{X}}(d) = \Pr\big[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\big]$$

  for different random vectors $\mathbf{X}$.
- Also got an algorithm for constructing codes.

Conclusion

- Showed how to connect optimal code size/distance tradeoff and distance spectrum

$$F_{\mathbf{X}}(d) = \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]$$

  for different random vectors $\mathbf{X}$.
- Also got an algorithm for constructing codes.

Some open questions.

- Better algorithm (improved rule for combining codewords)?

Conclusion

- Showed how to connect optimal code size/distance tradeoff and distance spectrum

$$F_{\mathbf{X}}(d) = \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]$$

for different random vectors $\mathbf{X}$.

- Also got an algorithm for constructing codes.

Some open questions.

- Better algorithm (improved rule for combining codewords)?
- Better bounds for the current algorithm?

Conclusion

- Showed how to connect optimal code size/distance tradeoff and distance spectrum

$$F_{\mathbf{X}}(d) = \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]$$

  for different random vectors $\mathbf{X}$.

- Also got an algorithm for constructing codes.

Some open questions.

- Better algorithm (improved rule for combining codewords)?
- Better bounds for the current algorithm?
- Improved codes?

Conclusion

- Showed how to connect optimal code size/distance tradeoff and distance spectrum

$$F_{\mathbf{X}}(d) = \Pr\left[\mu(\mathbf{X}, \hat{\mathbf{X}}) < d\right]$$

  for different random vectors $\mathbf{X}$.
- Also got an algorithm for constructing codes.

Some open questions.

- Better algorithm (improved rule for combining codewords)?
- Better bounds for the current algorithm?
- Improved codes?
- To appear in the IEEE Transactions on Information Theory in 2019.

Thanks!

Thanks!

My collaborators and I at ITW 2017 (Kaohsiung)