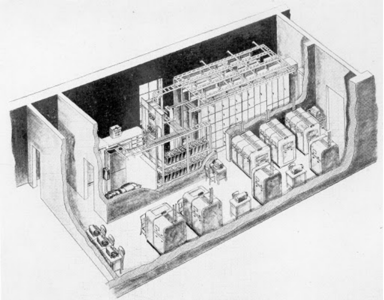
The Fundamental Limits of Communication and How to Achieve Them

Hsin-Po WANG

Department of Mathematics, University of Illinois at Urbana–Champaign

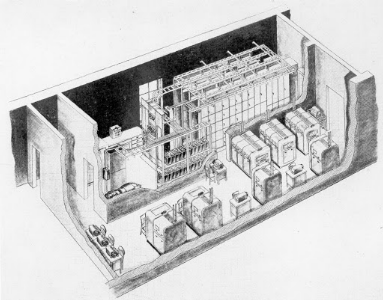
2021-01-04 NTU Special Math Seminar

1940s; Bell labs; Model V computer



Should the machine detect an error, it pauses.
A person will come to fix it.
That person does not work on weekends.

1940s; Bell labs; Model V computer



Should the machine detect an error, it pauses.
A person will come to fix it.
That person does not work on weekends.

1940s; Bell labs; Model V computer

Richard W. Hamming (UIUC PhD 1942) usually ordered the machine to do complicated computation over the weekend, only to find on Monday that it was interrupted on Friday midnight.

Quote from an interview: [ISBN:0883850370 p.17]

Damn it, if the machine can detect an error, why can't it locate the position of the error and correct it?

1940s; Bell labs; Model V computer

Richard W. Hamming (UIUC PhD 1942) usually ordered the machine
to do complicated computation over the weekend,
only to find on Monday that it was interrupted on Friday midnight.

Quote from an interview:

[ISBN:0883850370 p.17]

Damn it, if the machine can detect an error,
why can't it locate the position of the error and correct it?

Hamming code

For every 7 bits, instead of all 128 vectors, allow only the null space of

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Let $\bar{x} \in \mathbb{F}_2^7$ be a valid vector (codeword), that is, $H\bar{x} = 0$.
If $\bar{y} = \bar{x} + \bar{e}$ is the corrupted word, where $\bar{e}$ contains one 1 and six 0's, then $H\bar{y} = H(\bar{x} + \bar{e}) = H\bar{e}$ is the error position; flip that bit back.

Hamming code

For every 7 bits, instead of all 128 vectors, allow only the null space of

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Let $\bar{x} \in \mathbb{F}_2^7$ be a valid vector (codeword), that is, $H\bar{x} = 0$.
If $\bar{y} = \bar{x} + \bar{e}$ is the corrupted word, where $\bar{e}$ contains one 1 and six 0's, then $H\bar{y} = H(\bar{x} + \bar{e}) = H\bar{e}$ is the error position; flip that bit back.

Performance of Hamming code

Block length $N = 7$ (how many bits are grouped together).

Code rate $R = 4/7$ (how efficiently the information is recorded).

What could go wrong? When $\bar{e}$ contains more than one 1.
How often will things go wrong?

Performance of Hamming code

Block length $N = 7$ (how many bits are grouped together).

Code rate $R = 4/7$ (how efficiently the information is recorded).

What could go wrong? When $\bar{e}$ contains more than one 1.
How often will things go wrong?

Performance of Hamming code

Block length $N = 7$ (how many bits are grouped together).

Code rate $R = 4/7$ (how efficiently the information is recorded).

What could go wrong? When $\bar{e}$ contains more than one 1.
How often will things go wrong?

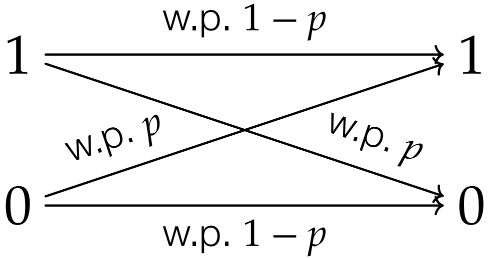
Probabilistic model

Input is a random variable $X \in \mathbb{F}_2$; output is another $Y \in \mathbb{F}_2$.
Y is usually X; But Y could be $1 - X$ (flipped) with probability p.

Binary symmetric channel (BSC) with crossover probability p.

Probabilistic model

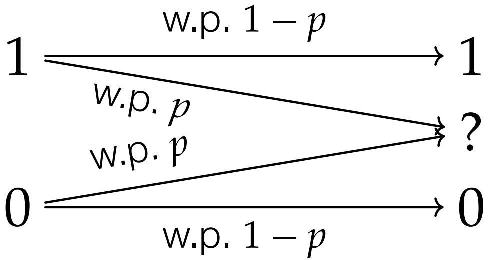
Input is a random variable $X \in \mathbb{F}_2$; output is another $Y \in \mathbb{F}_2$.
Y is usually X; But Y could be $1 - X$ (flipped) with probability p.



Binary symmetric channel (BSC) with crossover probability p.

Another probabilistic model

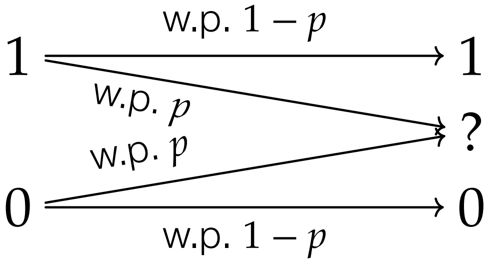
Binary erasure channel (BEC) with erasure probability p.
Y is usually X; But Y could be "?" (erased) with probability p.



In classical coding theory, BSC and BEC are the mental models.

Another probabilistic model

Binary erasure channel (BEC) with erasure probability p.
Y is usually X; But Y could be "?" (erased) with probability p.



In classical coding theory, BSC and BEC are the mental models.

Coding = encoding + decoding

Choose a codebook $\mathscr{B} \subseteq \mathbb{F}_2^N$, the set of valid codewords.
The inputs are limited to codewords in the codebook $\bar{X}_1^N \in \mathscr{B}$.
$\bar{X}_1^N$ is shorthand for a vector $X_1 X_2 \cdots X_N \in \mathbb{F}_2^N$.

Example: $\mathscr{B} := \{0000, 0111, 1100\} \subseteq \mathbb{F}_2^3$ over BEC.
See 0?00 $\Longrightarrow$ it must be 0000 with one erasure.
See 1?0? $\Longrightarrow$ it must be 1100 with two erasures.
See ??00 $\Longrightarrow$ No idea; could be 0000 or 1100.

Coding = encoding + decoding

Choose a codebook $\mathcal{B} \subseteq \mathbb{F}_2^N$, the set of valid codewords.
The inputs are limited to codewords in the codebook $\bar{X}_1^N \in \mathcal{B}$.
$\bar{X}_1^N$ is shorthand for a vector $X_1 X_2 \cdots X_N \in \mathbb{F}_2^N$.

Example: $\mathcal{B} := \{0000, 0111, 1100\} \subseteq \mathbb{F}_2^3$ over BEC.
See $0?00 \implies$ it must be 0000 with one erasure.
See $1?0? \implies$ it must be 1100 with two erasures.
See $??00 \implies$ No idea; could be 0000 or 1100.

Coding = encoding + decoding

Choose a codebook $\mathscr{B} \subseteq \mathbb{F}_2^N$, the set of valid codewords.
The inputs are limited to codewords in the codebook $\bar{X}_1^N \in \mathscr{B}$.
$\bar{X}_1^N$ is shorthand for a vector $X_1 X_2 \cdots X_N \in \mathbb{F}_2^N$.

Example: $\mathscr{B} := \{0000, 0111, 1100\} \subseteq \mathbb{F}_2^3$ over BEC.
See $0?00 \implies$ it must be 0000 with one erasure.
See $1?0? \implies$ it must be 1100 with two erasures.
See $??00 \implies$ No idea; could be 0000 or 1100.

Coding = encoding + decoding

Choose a codebook $\mathscr{B} \subseteq \mathbb{F}_2^N$, the set of valid codewords.
The inputs are limited to codewords in the codebook $\bar{X}_1^N \in \mathscr{B}$.
$\bar{X}_1^N$ is shorthand for a vector $X_1 X_2 \cdots X_N \in \mathbb{F}_2^N$.

Example: $\mathscr{B} := \{0000, 0111, 1100\} \subseteq \mathbb{F}_2^3$ over BEC.
See $0?00 \implies$ it must be 0000 with one erasure.
See $1?0? \implies$ it must be 1100 with two erasures.
See $??00 \implies$ No idea; could be 0000 or 1100.

More ambiguity

Example: $\mathcal{B} := \{0000, 0111, 1100\} \subseteq \mathbb{F}_2^3$ over BSC.
See 0001 $\Longrightarrow$ it is most likely 0000, likelihood $p(1-p)^3$;
$\Longrightarrow$ but could be 0111 with two flips, likelihood $p^2(1-p)^2$;
$\Longrightarrow$ could also be 1100 with three flips, likelihood $p^3(1-p)$.

See $Y_1Y_2Y_3Y_4 \Longrightarrow$ compute the likelihood for $x_1x_2x_3x_4 \in \mathcal{B}$ by
$\mathbb{P}(Y_1Y_2Y_3Y_4 \mid x_1x_2x_3x_4) = \mathbb{P}(Y_1 \mid x_1)\mathbb{P}(Y_2 \mid x_2)\mathbb{P}(Y_3 \mid x_3)\mathbb{P}(Y_4 \mid x_4)$,
where $\mathbb{P}(Y_i \mid x_i)$ equals $1-p$ if $Y_1 = x_1$, or p otherwise.

Find the $x_1x_2x_3x_4 \in \mathcal{B}$ that maximizes the likelihood.

More ambiguity

Example: $\mathcal{B} := \{0000, 0111, 1100\} \subseteq \mathbb{F}_2^3$ over BSC.
See $0001 \Longrightarrow$ it is most likely 0000, likelihood $p(1-p)^3$;
$\Longrightarrow$ but could be 0111 with two flips, likelihood $p^2(1-p)^2$;
$\Longrightarrow$ could also be 1100 with three flips, likelihood $p^3(1-p)$.

See $Y_1Y_2Y_3Y_4 \Longrightarrow$ compute the likelihood for $x_1x_2x_3x_4 \in \mathcal{B}$ by
$\mathbb{P}(Y_1Y_2Y_3Y_4 \mid x_1x_2x_3x_4) = \mathbb{P}(Y_1 \mid x_1)\mathbb{P}(Y_2 \mid x_2)\mathbb{P}(Y_3 \mid x_3)\mathbb{P}(Y_4 \mid x_4)$,
where $\mathbb{P}(Y_i \mid x_i)$ equals $1-p$ if $Y_1 = x_1$, or p otherwise.

Find the $x_1x_2x_3x_4 \in \mathcal{B}$ that maximizes the likelihood.

More ambiguity

Example: $\mathcal{B} := \{0000, 0111, 1100\} \subseteq \mathbb{F}_2^3$ over BSC.
See $0001 \implies$ it is most likely 0000, likelihood $p(1-p)^3$;
$\implies$ but could be 0111 with two flips, likelihood $p^2(1-p)^2$;
$\implies$ could also be 1100 with three flips, likelihood $p^3(1-p)$.

See $Y_1Y_2Y_3Y_4 \implies$ compute the likelihood for $x_1x_2x_3x_4 \in \mathcal{B}$ by
$\mathbb{P}(Y_1Y_2Y_3Y_4 \mid x_1x_2x_3x_4) = \mathbb{P}(Y_1 \mid x_1)\mathbb{P}(Y_2 \mid x_2)\mathbb{P}(Y_3 \mid x_3)\mathbb{P}(Y_4 \mid x_4)$,
where $\mathbb{P}(Y_i \mid x_i)$ equals $1-p$ if $Y_1 = x_1$, or p otherwise.

Find the $x_1x_2x_3x_4 \in \mathcal{B}$ that maximizes the likelihood.

More ambiguity

Example: $\mathcal{B} := \{0000, 0111, 1100\} \subseteq \mathbb{F}_2^3$ over BSC.
See 0001 $\implies$ it is most likely 0000, likelihood $p(1-p)^3$;
$\implies$ but could be 0111 with two flips, likelihood $p^2(1-p)^2$;
$\implies$ could also be 1100 with three flips, likelihood $p^3(1-p)$.

See $Y_1Y_2Y_3Y_4 \implies$ compute the likelihood for $x_1x_2x_3x_4 \in \mathcal{B}$ by
$\mathbb{P}(Y_1Y_2Y_3Y_4 \mid x_1x_2x_3x_4) = \mathbb{P}(Y_1 \mid x_1)\mathbb{P}(Y_2 \mid x_2)\mathbb{P}(Y_3 \mid x_3)\mathbb{P}(Y_4 \mid x_4)$,
where $\mathbb{P}(Y_i \mid x_i)$ equals $1-p$ if $Y_1 = x_1$, or p otherwise.

Find the $x_1x_2x_3x_4 \in \mathcal{B}$ that maximizes the likelihood.

More ambiguity

Example: $\mathcal{B} := \{0000, 0111, 1100\} \subseteq \mathbb{F}_2^3$ over BSC.
See $0001 \implies$ it is most likely 0000, likelihood $p(1-p)^3$;
$\implies$ but could be 0111 with two flips, likelihood $p^2(1-p)^2$;
$\implies$ could also be 1100 with three flips, likelihood $p^3(1-p)$.

See $Y_1Y_2Y_3Y_4 \implies$ compute the likelihood for $x_1x_2x_3x_4 \in \mathcal{B}$ by
$\mathbb{P}(Y_1Y_2Y_3Y_4 \mid x_1x_2x_3x_4) = \mathbb{P}(Y_1 \mid x_1)\mathbb{P}(Y_2 \mid x_2)\mathbb{P}(Y_3 \mid x_3)\mathbb{P}(Y_4 \mid x_4)$,
where $\mathbb{P}(Y_i \mid x_i)$ equals $1-p$ if $Y_1 = x_1$, or p otherwise.

Find the $x_1x_2x_3x_4 \in \mathcal{B}$ that maximizes the likelihood.

Goal of coding

A good codebook $\mathcal{B}$ should possess the following properties.

Small block length $N \implies$ recall $\mathcal{B} \subseteq \mathbb{F}_2^N$; shorter vectors, easier life.

Big $|\mathcal{B}| \implies$ more codewords mean sending more information at once.
In fact, it is $R := \dfrac{\log_2 |\mathcal{B}|}{N}$ that should be large; this is the code rate.

Low error probability $P \implies$ efforts are pointless if information is lost.

Goal of coding

A good codebook $\mathscr{B}$ should possess the following properties.

Small block length $N \implies$ recall $\mathscr{B} \subseteq \mathbb{F}_2^N$; shorter vectors, easier life.

Big $|\mathscr{B}| \implies$ more codewords mean sending more information at once.
In fact, it is $R := \dfrac{\log_2 |\mathscr{B}|}{N}$ that should be large; this is the code rate.

Low error probability $P \implies$ efforts are pointless if information is lost.

Goal of coding

A good codebook $\mathscr{B}$ should possess the following properties.

Small block length $N \implies$ recall $\mathscr{B} \subseteq \mathbb{F}_2^N$; shorter vectors, easier life.

Big $|\mathscr{B}| \implies$ more codewords mean sending more information at once.
In fact, it is $R := \dfrac{\log_2 |\mathscr{B}|}{N}$ that should be large; this is the code rate.

Low error probability $P \implies$ efforts are pointless if information is lost.

Goal of coding

A good codebook $\mathscr{B}$ should possess the following properties.

Small block length $N \implies$ recall $\mathscr{B} \subseteq \mathbb{F}_2^N$; shorter vectors, easier life.

Big $|\mathscr{B}| \implies$ more codewords mean sending more information at once.
In fact, it is $R := \dfrac{\log_2 |\mathscr{B}|}{N}$ that should be large; this is the code rate.

Low error probability $P \implies$ efforts are pointless if information is lost.

Fundamental limit of coding

Consider the parameter triple (N, R, P)
(block length, code rate, error probability).

Shannon48: You can find a series of block codes $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \ldots$
such that $N(\mathcal{B}_n) \to \infty$ and $R(\mathcal{B}_n) \to C$ and $P(\mathcal{B}_n) \to 0$ as $n \to \infty$,
where C is a magic number that you cannot beat, provably.

C is called the Shannon capacity. It is the fundamental limit of coding.

Fundamental limit of coding

Consider the parameter triple (N, R, P)
(block length, code rate, error probability).

Shannon48: You can find a series of block codes $\mathscr{B}_1, \mathscr{B}_2, \mathscr{B}_3, \ldots$
such that $N(\mathscr{B}_n) \to \infty$ and $R(\mathscr{B}_n) \to C$ and $P(\mathscr{B}_n) \to 0$ as $n \to \infty$,
where C is a magic number that you cannot beat, provably.

C is called the Shannon capacity. It is the fundamental limit of coding.

Fundamental limit of coding

Consider the parameter triple (N, R, P)
(block length, code rate, error probability).

Shannon48: You can find a series of block codes $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \ldots$
such that $N(\mathcal{B}_n) \to \infty$ and $R(\mathcal{B}_n) \to C$ and $P(\mathcal{B}_n) \to 0$ as $n \to \infty$,
where C is a magic number that you cannot beat, provably.

C is called the Shannon capacity. It is the fundamental limit of coding.

How to construct codebook

Let N be really large. Let $R := C - \varepsilon$. We preset $|\mathscr{B}| = 2^{RN}$.

Choose a random subset $\mathscr{B} \subseteq \mathbb{F}_2^N$ of that size. (It's just that easy!)

With high probability, any two distinct codewords in $\mathscr{B}$ are far away enough such that you cannot (well~) confuse them.

How to construct codebook

Let N be really large. Let $R := C - \varepsilon$. We preset $|\mathcal{B}| = 2^{RN}$.

Choose a random subset $\mathcal{B} \subseteq \mathbb{F}_2^N$ of that size. (It's just that easy!)

With high probability, any two distinct codewords in $\mathcal{B}$ are far away enough such that you cannot (well~) confuse them.

How to construct codebook

Let N be really large. Let $R := C - \varepsilon$. We preset $|\mathscr{B}| = 2^{RN}$.

Choose a random subset $\mathscr{B} \subseteq \mathbb{F}_2^N$ of that size. (It's just that easy!)

With high probability, any two distinct codewords in $\mathscr{B}$ are far away enough such that you cannot (well~) confuse them.

What Shannon doesn't tell you

Of course we can make P (or R) arbitrarily close to 0 (or C).
But at what cost? If $N \geq 10^{100}$ is needed, the whole theory is useless.

Next step: understand the relation between N and P and $C - R$.

What Shannon doesn't tell you

Of course we can make P (or R) arbitrarily close to 0 (or C).
But at what cost? If $N \geq 10^{100}$ is needed, the whole theory is useless.

Next step: understand the relation between N and P and $C - R$.

The trend of P

Fix a code rate $R < C$, error probability P decays to 0 really fast:

$$P \approx e^{-\Theta(N)}.$$

To get a feeling: $P = 10^{-3}$ (once per hour) at $N = 100$,
$P = 10^{-6}$ (once per day) at $N = 200$.
$P = 10^{-9}$ (once per year) at $N = 300$.
$P = 10^{-12}$ (once per lifetime) at $N = 400$.

The trend of P

Fix a code rate $R < C$, error probability P decays to 0 really fast:

$$P \approx e^{-\Theta(N)}.$$

To get a feeling: $P = 10^{-3}$ (once per hour) at $N = 100$,
$P = 10^{-6}$ (once per day) at $N = 200$.
$P = 10^{-9}$ (once per year) at $N = 300$.
$P = 10^{-12}$ (once per lifetime) at $N = 400$.

The trend of P

Fix a code rate $R < C$, error probability P decays to 0 really fast:

$$P \approx e^{-\Theta(N)}.$$

To get a feeling: $P = 10^{-3}$ (once per hour) at $N = 100$,
$P = 10^{-6}$ (once per day) at $N = 200$.
$P = 10^{-9}$ (once per year) at $N = 300$.
$P = 10^{-12}$ (once per lifetime) at $N = 400$.

The trend of P

Fix a code rate $R < C$, error probability P decays to 0 really fast:

$$P \approx e^{-\Theta(N)}.$$

To get a feeling: $P = 10^{-3}$ (once per hour) at $N = 100$,
$P = 10^{-6}$ (once per day) at $N = 200$.
$P = 10^{-9}$ (once per year) at $N = 300$.
$P = 10^{-12}$ (once per lifetime) at $N = 400$.

The trend of P

Fix a code rate $R < C$, error probability P decays to 0 really fast:

$$P \approx e^{-\Theta(N)}.$$

To get a feeling: $P = 10^{-3}$ (once per hour) at $N = 100$,
$P = 10^{-6}$ (once per day) at $N = 200$.
$P = 10^{-9}$ (once per year) at $N = 300$.
$P = 10^{-12}$ (once per lifetime) at $N = 400$.

The trend of R

Fix an error probability P, code rate R approaches C moderately fast:

$$R \approx C - \frac{1}{\sqrt{N}}.$$

$R = C - 1/10$ at $N = 100$ (CPU word).
$R = C - 1/100$ at $N = 10^4$ (internet package).
$R = C - 1/1000$ at $N = 10^6$ (least file size of linux).

The trend of R

Fix an error probability P, code rate R approaches C moderately fast:

$$R \approx C - \frac{1}{\sqrt{N}}.$$

$R = C - 1/10$ at $N = 100$ (CPU word).
$R = C - 1/100$ at $N = 10^4$ (internet package).
$R = C - 1/1000$ at $N = 10^6$ (least file size of linux).

The trend of R

Fix an error probability P, code rate R approaches C moderately fast:

$$R \approx C - \frac{1}{\sqrt{N}}.$$

$R = C - 1/10$ at $N = 100$ (CPU word).
$R = C - 1/100$ at $N = 10^4$ (internet package).
$R = C - 1/1000$ at $N = 10^6$ (least file size of linux).

The trend of R

Fix an error probability P, code rate R approaches C moderately fast:

$$R \approx C - \frac{1}{\sqrt{N}}.$$

$R = C - 1/10$ at $N = 100$ (CPU word).
$R = C - 1/100$ at $N = 10^4$ (internet package).
$R = C - 1/1000$ at $N = 10^6$ (least file size of linux).

The joint trend of P and R

The trend of P says $P \approx e^{-\Theta(N)}$; or equivalently $-\log P \approx N$.
The trend of R says $C - R \approx 1/\sqrt{N}$; or equivalently $1/(C - R)^2 \approx N$.

A (natural) generalization

$$\frac{-\log P}{(C - R)^2} \approx N.$$

The joint trend of P and R

The trend of P says $P \approx e^{-\Theta(N)}$; or equivalently $-\log P \approx N$.
The trend of R says $C - R \approx 1/\sqrt{N}$; or equivalently $1/(C-R)^2 \approx N$.

A (natural) generalization

$$\frac{-\log P}{(C-R)^2} \approx N.$$

Analog in probability theory

Probability theory concerns sum of i.i.d.—Z mean μ variance σ.

Large deviations principle:
Fix a z, then $\mathbb{P}\{Z_1 + Z_2 + \cdots + Z_N > N(\mu + z)\} = e^{-\Theta(N)}$.

Central limit theory:
$Z_1 + Z_2 + \cdots + Z_N \approx$ Normal$(\mu, \sigma\sqrt{N})$; or equivalently $\frac{\sum Z}{N} \approx \mu \pm \frac{\Theta(1)}{\sqrt{N}}$.

Generalization: moderate deviations principle:
$\frac{-\log \mathbb{P}\{Z_1+Z_2+\cdots+Z_N>N(\mu+\varepsilon_n z)\}}{\varepsilon_n^2} \approx N$.

Analog in probability theory

Probability theory concerns sum of i.i.d.—Z mean μ variance σ.

Large deviations principle:
Fix a z, then $\mathbb{P}\left\{Z_1 + Z_2 + \cdots + Z_N > N(\mu + z)\right\} = e^{-\Theta(N)}$.

Central limit theory:
$Z_1 + Z_2 + \cdots + Z_N \approx \text{Normal}(\mu, \sigma\sqrt{N})$; or equivalently $\frac{\sum Z}{N} \approx \mu \pm \frac{\Theta(1)}{\sqrt{N}}$.

Generalization: moderate deviations principle:
$$\frac{-\log \mathbb{P}\left\{Z_1+Z_2+\cdots+Z_N>N(\mu+\varepsilon_n z)\right\}}{\varepsilon_n^2} \approx N.$$

Analog in probability theory

Probability theory concerns sum of i.i.d.—Z mean μ variance σ.

Large deviations principle:
Fix a z, then $\mathbb{P}\Big\{Z_1 + Z_2 + \cdots + Z_N > N(\mu + z)\Big\} = e^{-\Theta(N)}$.

Central limit theory:
$Z_1 + Z_2 + \cdots + Z_N \approx \text{Normal}(\mu, \sigma\sqrt{N})$; or equivalently $\frac{\sum Z}{N} \approx \mu \pm \frac{\Theta(1)}{\sqrt{N}}$.

Generalization: moderate deviations principle:
$\frac{-\log \mathbb{P}\{Z_1+Z_2+\cdots+Z_N>N(\mu+\varepsilon_n z)\}}{\varepsilon_n^2} \approx N$.

Analog in probability theory

Probability theory concerns sum of i.i.d.—Z mean μ variance σ.

Large deviations principle:
Fix a z, then $\mathbb{P}\left\{Z_1 + Z_2 + \cdots + Z_N > N(\mu + z)\right\} = e^{-\Theta(N)}$.

Central limit theory:
$Z_1 + Z_2 + \cdots + Z_N \approx \text{Normal}(\mu, \sigma\sqrt{N})$; or equivalently $\frac{\sum Z}{N} \approx \mu \pm \frac{\Theta(1)}{\sqrt{N}}$.

Generalization: moderate deviations principle:
$\frac{-\log \mathbb{P}\left\{Z_1+Z_2+\cdots+Z_N>N(\mu+\varepsilon_n z)\right\}}{\varepsilon_n^2} \approx N$.

Comparison

	Probability theory	Coding theory
LLN	$\bar{Z} \to \mu$	$(P, R) \to (0, C)$
LDP	$\mathbb{P}\{\bar{Z} - \mu > z\} \approx e^{-NI(z)}$	$P \approx e^{-E_{\mathrm{r}}(R)N}$
CLT	$\bar{Z} - \mu \sim \text{Normal}(0, \frac{\sigma}{\sqrt{N}})$	$C - R \approx \frac{Q^{-1}(P)}{\sqrt{VN}}$
MDP	$\frac{-\log \mathbb{P}\{\bar{Z}-\mu>\varepsilon_N z\}}{\varepsilon_N^2} \approx NI(z)$	$\frac{-\log P}{(C-R)^2} \approx \frac{N}{2V}$

What second-order theory doesn't tell you

Of course we can make P (or R) close to 0 (or C) at the pace of $\frac{-\log P}{(C-R)^2} \approx \frac{N}{2V}$, but at what cost?

Recall that we have to compute likelihoods (assuming i.i.d. channels)
$\mathbb{P}(Y_1 Y_2 \cdots Y_N \mid x_1 x_2 \cdots x_N) = \mathbb{P}(Y_1 \mid x_1)\mathbb{P}(Y_2 \mid x_2) \cdots \mathbb{P}(Y_N \mid x_N)$
and choose the maximizing $x_1 x_2 x_3 x_4 \in \mathcal{B}$.

Next step: how to maximize likelihood, easily?

What second-order theory doesn't tell you

Of course we can make P (or R) close to 0 (or C) at the pace of $\frac{-\log P}{(C-R)^2} \approx \frac{N}{2V}$, but at what cost?

Recall that we have to compute likelihoods (assuming i.i.d. channels)
$\mathbb{P}(Y_1 Y_2 \cdots Y_N \mid x_1 x_2 \cdots x_N) = \mathbb{P}(Y_1 \mid x_1)\mathbb{P}(Y_2 \mid x_2) \cdots \mathbb{P}(Y_N \mid x_N)$
and choose the maximizing $x_1 x_2 x_3 x_4 \in \mathscr{B}$.

Next step: how to maximize likelihood, easily?

What second-order theory doesn't tell you

Of course we can make P (or R) close to 0 (or C) at the pace of $\frac{-\log P}{(C-R)^2} \approx \frac{N}{2V}$, but at what cost?

Recall that we have to compute likelihoods (assuming i.i.d. channels)
$\mathbb{P}(Y_1 Y_2 \cdots Y_N \mid x_1 x_2 \cdots x_N) = \mathbb{P}(Y_1 \mid x_1)\mathbb{P}(Y_2 \mid x_2) \cdots \mathbb{P}(Y_N \mid x_N)$
and choose the maximizing $x_1 x_2 x_3 x_4 \in \mathscr{B}$.

Next step: how to maximize likelihood, easily?

Low complexity codes

Engineers develop “easy codes” regardless of Shannon theory.

Reed–Muller (1954), convolutional (1955), Bose–Chaudhuri–Hocquenghem (1959), Reed–Solomon (1960), trellis modulation (1970s), turbo (1990s), low-density parity-check (1963 and 1996), repeat-accumulate (1998), fountain (1998), and polar (2009).

Only polar codes and LDPC codes achieve first-order limit.

Low complexity codes

Engineers develop “easy codes” regardless of Shannon theory.

Reed–Muller (1954), convolutional (1955), Bose–Chaudhuri–Hocquenghem (1959), Reed–Solomon (1960), trellis modulation (1970s), turbo (1990s), low-density parity-check (1963 and 1996), repeat-accumulate (1998), fountain (1998), and polar (2009).

Only polar codes and LDPC codes achieve first-order limit.

Low complexity codes

Engineers develop “easy codes” regardless of Shannon theory.

Reed–Muller (1954), convolutional (1955), Bose–Chaudhuri–Hocquenghem (1959), Reed–Solomon (1960), trellis modulation (1970s), turbo (1990s), low-density parity-check (1963 and 1996), repeat-accumulate (1998), fountain (1998), and polar (2009).

Only polar codes and LDPC codes achieve first-order limit.

Who achieves second-order limits?

	Coding theory	Years
LLN	$(P,R) \to (0,C)$	random 1948, polar 2009, LDPC 2014
LDP	$P \approx e^{-E_{\mathrm{r}}(R)N}$	random 1961, polar 2009–14
CLT	$C - R \approx \frac{Q^{-1}(P)}{\sqrt{VN}}$	random 1950, polar 2010–20
MDP	$\frac{-\log P}{(C-R)^2} \approx \frac{N}{2V}$	random 2014, polar 2016–20

Polar codes key ideas

Consider BEC with erasure probability $0 < p < 1$.

Polar transformation $p \mapsto (p^2, 2p - p^2)$, recursively.
$p^2 \mapsto (p^4, 2p^2 - p^4)$ and $2p - p^2 \mapsto ((2p - p^2)^2, 2(2p - p^2) - (2p - p^2)^2)$.

Arıkan's martingale $Z_{n+1} = Z_n^2$ or $2Z_n - Z_n^2$ with equal probability.

Doob's martingale convergence theorem $Z_n \to Z_\infty \in \{0, 1\}$.
But how fast? (If it takes forever to converge the theory is useless.)

Polar codes key ideas

Consider BEC with erasure probability $0 < p < 1$.

Polar transformation $p \mapsto (p^2, 2p - p^2)$, recursively.
$p^2 \mapsto (p^4, 2p^2 - p^4)$ and $2p - p^2 \mapsto ((2p - p^2)^2, 2(2p - p^2) - (2p - p^2)^2)$.

Arıkan's martingale $Z_{n+1} = Z_n^2$ or $2Z_n - Z_n^2$ with equal probability.

Doob's martingale convergence theorem $Z_n \to Z_\infty \in \{0, 1\}$.
But how fast? (If it takes forever to converge the theory is useless.)

Polar codes key ideas

Consider BEC with erasure probability $0 < p < 1$.

Polar transformation $p \mapsto (p^2, 2p - p^2)$, recursively.
$p^2 \mapsto (p^4, 2p^2 - p^4)$ and $2p - p^2 \mapsto ((2p-p^2)^2, 2(2p-p^2) - (2p-p^2)^2)$.

Arıkan's martingale $Z_{n+1} = Z_n^2$ or $2Z_n - Z_n^2$ with equal probability.

Doob's martingale convergence theorem $Z_n \to Z_\infty \in \{0, 1\}$.
But how fast? (If it takes forever to converge the theory is useless.)

Polar codes key ideas

Consider BEC with erasure probability $0 < p < 1$.

Polar transformation $p \mapsto (p^2, 2p - p^2)$, recursively.
$p^2 \mapsto (p^4, 2p^2 - p^4)$ and $2p - p^2 \mapsto ((2p - p^2)^2, 2(2p - p^2) - (2p - p^2)^2)$.

Arıkan's martingale $Z_{n+1} = Z_n^2$ or $2Z_n - Z_n^2$ with equal probability.

Doob's martingale convergence theorem $Z_n \to Z_\infty \in \{0, 1\}$.
But how fast? (If it takes forever to converge the theory is useless.)

Polar counterpart of LDP

When p is really really small, $p \mapsto (p^2, p)$. That is, $\log p \mapsto (2\log p, \log p)$, or $\log_2(-\log p) \mapsto (1 + \log_2(-\log p), \log_2(-\log p))$.

With 1/2 probability, $\log_2(-\log Z_n)$ increases by 1, otherwise nothing. For those Z_n that are small, $\log_2(-\log Z_n) \approx n/2$; or $Z_n \approx e^{-2^{n/2}}$.

With more advanced tricks, $Z_n \approx e^{-\ell^{0.99n}}$; this means $P \approx e^{-N^{0.99}}$.

Polar counterpart of LDP

When p is really really small, $p \mapsto (p^2, p)$. That is, $\log p \mapsto (2\log p, \log p)$, or $\log_2(-\log p) \mapsto (1 + \log_2(-\log p), \log_2(-\log p))$.

With $1/2$ probability, $\log_2(-\log Z_n)$ increases by 1, otherwise nothing. For those Z_n that are small, $\log_2(-\log Z_n) \approx n/2$; or $Z_n \approx e^{-2^{n/2}}$.

With more advanced tricks, $Z_n \approx e^{-\ell^{0.99n}}$; this means $P \approx e^{-N^{0.99}}$.

Polar counterpart of LDP

When p is really really small, $p \mapsto (p^2, p)$. That is, $\log p \mapsto (2\log p, \log p)$, or $\log_2(-\log p) \mapsto (1 + \log_2(-\log p), \log_2(-\log p))$.

With $1/2$ probability, $\log_2(-\log Z_n)$ increases by 1, otherwise nothing. For those Z_n that are small, $\log_2(-\log Z_n) \approx n/2$; or $Z_n \approx e^{-2^{n/2}}$.

With more advanced tricks, $Z_n \approx e^{-\ell^{0.99n}}$; this means $P \approx e^{-N^{0.99}}$.

Polar counterpart of CLT

$\sqrt{Z_n(1-Z_n)}$ is a supermartingale. In fact, $\mathbb{E}[\sqrt{Z_n(1-Z_n)}] \approx 2^{-\rho n}$.

This means that most of the times $\sqrt{Z_n(1-Z_n)}$ is really small, wherein either Z_n is small or $1-Z_n$ is small.
Those that are not small are of measure $2^{-\rho n}$.

With more advanced tricks, $\mathbb{P}\{Z_n(1-Z_n)$ not small$\} \approx \ell^{-0.49n}$; this means $C-R \approx N^{-0.49}$.

Polar counterpart of CLT

$\sqrt{Z_n(1-Z_n)}$ is a supermartingale. In fact, $\mathbb{E}[\sqrt{Z_n(1-Z_n)}] \approx 2^{-\rho n}$.

This means that most of the times $\sqrt{Z_n(1-Z_n)}$ is really small, wherein either Z_n is small or $1-Z_n$ is small.
Those that are not small are of measure $2^{-\rho n}$.

With more advanced tricks, $\mathbb{P}\{Z_n(1-Z_n) \text{ not small}\} \approx \ell^{-0.49n}$; this means $C-R \approx N^{-0.49}$.

Polar counterpart of CLT

$\sqrt{Z_n(1-Z_n)}$ is a supermartingale. In fact, $\mathbb{E}[\sqrt{Z_n(1-Z_n)}] \approx 2^{-\rho n}$.

This means that most of the times $\sqrt{Z_n(1-Z_n)}$ is really small, wherein either Z_n is small or $1-Z_n$ is small.
Those that are not small are of measure $2^{-\rho n}$.

With more advanced tricks, $\mathbb{P}\{Z_n(1-Z_n) \text{ not small}\} \approx \ell^{-0.49n}$;
this means $C - R \approx N^{-0.49}$.

Polar counterpart of MDP

When $Z_n(1 - Z_n)$ is small, martingale is controlled by the LDP theory.
When $Z_n(1 - Z_n)$ is large, martingale is controlled by the CLT theory.

With positive measure, Z_n goes back and forth between LDP and CLT.

With more advanced tricks, $\frac{-\log P}{(C-R)^2} \approx N^{0.99}$.

Polar counterpart of MDP

When $Z_n(1 - Z_n)$ is small, martingale is controlled by the LDP theory.
When $Z_n(1 - Z_n)$ is large, martingale is controlled by the CLT theory.

With positive measure, Z_n goes back and forth between LDP and CLT.

With more advanced tricks, $\frac{-\log P}{(C-R)^2} \approx N^{0.99}$.

Polar counterpart of MDP

When $Z_n(1-Z_n)$ is small, martingale is controlled by the LDP theory.
When $Z_n(1-Z_n)$ is large, martingale is controlled by the CLT theory.

With positive measure, Z_n goes back and forth between LDP and CLT.

With more advanced tricks, $\frac{-\log P}{(C-R)^2} \approx N^{0.99}$.

Future works

Complexity is $O(N \log N)$. With more advanced tricks $O(N \log \log N)$.
$\Longrightarrow$ Can it be better?

Polar codes apply to many-to-one communication,
one-to-many communication, and many others.
$\Longrightarrow$ Generalize polar code to even more scenarios.

In some scenarios, random codes' best performance is still unclear.
$\Longrightarrow$ Use polar to push random.

Future works

Complexity is $O(N \log N)$. With more advanced tricks $O(N \log \log N)$.
$\Longrightarrow$ Can it be better?

Polar codes apply to many-to-one communication,
one-to-many communication, and many others.
$\Longrightarrow$ Generalize polar code to even more scenarios.

In some scenarios, random codes' best performance is still unclear.
$\Longrightarrow$ Use polar to push random.

Future works

Complexity is $O(N \log N)$. With more advanced tricks $O(N \log \log N)$.
$\Longrightarrow$ Can it be better?

Polar codes apply to many-to-one communication, one-to-many communication, and many others.
$\Longrightarrow$ Generalize polar code to even more scenarios.

In some scenarios, random codes' best performance is still unclear.
$\Longrightarrow$ Use polar to push random.

Future works

Complexity is $O(N \log N)$. With more advanced tricks $O(N \log \log N)$.
$\Longrightarrow$ Can it be better?

Polar codes apply to many-to-one communication,
one-to-many communication, and many others.
$\Longrightarrow$ Generalize polar code to even more scenarios.

In some scenarios, random codes' best performance is still unclear.
$\Longrightarrow$ Use polar to push random.

Future works

Complexity is $O(N \log N)$. With more advanced tricks $O(N \log \log N)$.
$\Longrightarrow$ Can it be better?

Polar codes apply to many-to-one communication,
one-to-many communication, and many others.
$\Longrightarrow$ Generalize polar code to even more scenarios.

In some scenarios, random codes' best performance is still unclear.
$\Longrightarrow$ Use polar to push random.

Future works

Complexity is $O(N \log N)$. With more advanced tricks $O(N \log \log N)$.
$\Longrightarrow$ Can it be better?

Polar codes apply to many-to-one communication, one-to-many communication, and many others.
$\Longrightarrow$ Generalize polar code to even more scenarios.

In some scenarios, random codes' best performance is still unclear.
$\Longrightarrow$ Use polar to push random.

Comment and questions

?

	Symmetric			Asymmetric	
	binary	prime-ary	finite	binary	finite
LLN	Arikan09	STA09i	STA09i	SRDR12	
LDP*	AT09	MT14	Sasoglu11	HY13	
CLT*	KMTU10,MHU16	BGNRS18			
MDP*	GX15,MHU16	BGS18			
LDP	KSU10,HMTU13				
CLT	FHMV18,GRY20				
MDP					