How to write good code(s)

Adam Lott
23 April 2020
Outline

1. What is information?

2. Data compression

3. Data transmission
What is information?
What is information?

- One case contains a prize
- How to find with the fewest yes/no questions?
What is information?

- One case contains a prize
- How to find with the fewest yes/no questions?
  - "Is it in number $\leq 4"$?
What is information?

• One case contains a prize

• How to find with the fewest yes/no questions?

  - "Is it in number $\leq 4"$?  No
What is information?

• One case contains a prize

• How to find with the fewest yes/no questions?

  - "Is it in number \( \leq 4\)"? No
  - "Is it in number \( \leq 6\)"?
What is information?

• One case contains a prize

• How to find with the fewest yes/no questions?

  - "Is it in number ≤ 4"?  No
  - "Is it in number ≤ 6"?  Yes
What is information?

- One case contains a prize
- How to find with the fewest yes/no questions?
  - "Is it in number \( \leq 4\)?"? No
  - "Is it in number \( \leq 6\)?"? Yes
  - "Is it in number \( \leq 5\)?"?
What is information?

- One case contains a prize
- How to find with the fewest yes/no questions?
  - "Is it in number ≤ 4"? No
  - "Is it in number ≤ 6"? Yes
  - "Is it in number ≤ 5"? Yes
What is information?

• One case contains a prize

• How to find with the fewest yes/no questions?

  - "Is it in number ≤ 4"? No
  - "Is it in number ≤ 6"? Yes
  - "Is it in number ≤ 5"? Yes

• 3 questions ↔ 3 units of "information"?
What is information?
What is information?

- 2 questions is always sufficient, but how many questions on average?
What is information?

• 2 questions is always sufficient, but how many questions on average?

• 1 question 1/3 of the time, 2 questions 2/3 of the time

• \( \frac{1}{3}(1) + \frac{2}{3}(2) = \frac{5}{3} \) questions "on average"
What is information?

• Better strategy: do $N$ trials simultaneously
What is information?

• Better strategy: do $N$ trials simultaneously

• Possible configurations = $\{1,2,3\}^N$
What is information?

• Better strategy: do $N$ trials simultaneously

• Possible configurations = $\{1,2,3\}^N$

• Use bisection strategy to find correct configuration in $\lceil \log_2(3^N) \rceil = N \log 3 + O(1)$ many questions

• $\log_2 3 \approx 1.58$ questions "on average"
What is information?

**Definition, attempt #1:** The amount of information contained in an experiment is the minimum number of yes/no questions required (on average) to determine the outcome.

"Theorem": We gain $\log_2 k$ bits of information when we observe one of $k$ equally likely outcomes.
What is information?

• What if each outcome is not equally likely?

\[
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8}
\end{array}
\]
What is information?

What if each outcome is not equally likely?

We can do better than the naive bisection strategy:
What is information?

- What if each outcome is not equally likely?
- We can do better than the naive bisection strategy:
  - "Is it in number $\leq 1"?\)
What is information?

- What if each outcome is not equally likely?
- We can do better than the naive bisection strategy:
  - "Is it in number $\leq 1$"? No
What is information?

- What if each outcome is not equally likely?

- We can do better than the naive bisection strategy:
  - "Is it in number \( \leq 1 \)? No
  - "Is it in number \( \leq 2 \)?
What is information?

- What if each outcome is not equally likely?
- We can do better than the naive bisection strategy:
  - "Is it in number $\leq 1$"? No
  - "Is it in number $\leq 2$"? No
What is information?

- What if each outcome is not equally likely?
- We can do better than the naive bisection strategy:
  - "Is it in number ≤ 1"? No
  - "Is it in number ≤ 2"? No
  - "Is it in number ≤ 3"?
What is information?

• What if each outcome is not equally likely?

• We can do better than the naive bisection strategy:
  - "Is it in number $\leq 1$"?  No
  - "Is it in number $\leq 2$"?  No
  - "Is it in number $\leq 3$"?  Yes
What is information?

• What if each outcome is not equally likely?

• We can do better than the naive bisection strategy:
  - "Is it in number $\leq 1$"? No
  - "Is it in number $\leq 2$"? No
  - "Is it in number $\leq 3$"? Yes

• # questions on average = $(1/2)(1) + (1/4)(2) + (1/4)(3) = 1.75$
  • Worst case scenario is worse, but better on average!
What is information?

- Let's apply the $\mathcal{N}$ simultaneous trials strategy:
What is information?

• Let's apply the "$N$ simultaneous trials" strategy:

• Possible configurations = strings in $\{1, 2, 3, 4\}^N$ with the correct distribution of values
  - Viable configuration must have $1/2$ 1s, $1/4$ 2s, $1/8$ 3s, $1/8$ 4s
What is information?

・Let's apply the "$N$ simultaneous trials" strategy:

・Possible configurations = strings in $\{1,2,3,4\}^N$ with the correct distribution of values

  • Viable configuration must have $1/2$ 1s, $1/4$ 2s, $1/8$ 3s, $1/8$ 4s

Average # of questions $= \frac{1}{N} \log_2(\# \text{ viable configurations}) = \frac{1}{N} \log_2 \left( \frac{N!}{(N/2)!(N/4)!(N/8)!(N/8)!} \right)$
What is information?

• Let's apply the \(N\) simultaneous trials strategy:

• Possible configurations = strings in \(\{1,2,3,4\}^N\) with the correct distribution of values

  • Viable configuration must have 1/2 1s, 1/4 2s, 1/8 3s, 1/8 4s

Average # of questions = \(\frac{1}{N} \log_2(\text{# viable configurations}) = \frac{1}{N} \log_2 \frac{N!}{(N/2)!(N/4)!(N/8)!(N/8)!}\)

(Stirling's formula) \(\approx - \frac{1}{2} \log_2 \left(\frac{1}{2}\right) - \frac{1}{4} \log_2 \left(\frac{1}{4}\right) - \frac{1}{8} \log_2 \left(\frac{1}{8}\right) - \frac{1}{8} \log_2 \left(\frac{1}{8}\right)\)

\(\approx 1.75\)
What is information?

1 \quad 2 \quad \ldots \quad k

p_1 \quad p_2 \quad \ldots \quad p_k
What is information?

\[ 1 \quad 2 \quad \ldots \quad k \]

\[ p_1 \quad p_2 \quad \ldots \quad p_k \]

• Now apply this strategy in the most general case
What is information?

• Now apply this strategy in the most general case

• Possible configurations = strings in $\{1, 2, \ldots, k\}^N$ with the correct distribution -- 1 appears $p_1$ of the time, 2 appears $p_2$ of the time, etc.
What is information?

\[ \begin{array}{cccc}
1 & 2 & \ldots & k \\
p_1 & p_2 & \ldots & p_k
\end{array} \]

- Now apply this strategy in the most general case
- Possible configurations = strings in \( \{1,2,\ldots,k\}^N \) with the correct distribution -- 1 appears \( p_1 \) of the time, 2 appears \( p_2 \) of the time, etc.

Average # of questions \( \approx \frac{1}{N} \log_2 \frac{N!}{(Np_1)!\cdots(Np_k)!} \approx \sum_{i=1}^{k} -p_i \log_2(p_i) \)
What is information?

1 2 \ldots k

\begin{align*}
p_1 & \quad p_2 & \quad \ldots & \quad p_k
\end{align*}

- Now apply this strategy in the most general case
- Possible configurations = strings in \( \{1, 2, \ldots, k\}^N \) with the correct distribution -- 1 appears \( p_1 \) of the time, 2 appears \( p_2 \) of the time, etc.

\[
\text{Average \# of questions} = \frac{1}{N} \log_2 \frac{N!}{(Np_1)! \cdots (Np_k)!} \approx \sum_{i=1}^{k} -p_i \log_2(p_i)
\]

\[H(p) = \textbf{Shannon entropy} \text{ of probability distribution } p\]
What is information?
What is information?

What's going on?
What is information?

What's going on?

• To maximize efficiency: with each question, we don't need to reduce the number of possibilities by 1/2, but rather we need to distinguish between two equally probable outcomes.
What is information?

What's going on?

- To maximize efficiency: with each question, we don't need to reduce the number of possibilities by 1/2, but rather we need to distinguish between two equally probable outcomes.

Definition, attempt #2: We gain one bit of information each time we distinguish between two equally likely events.
What is information?

What's going on?

- To maximize efficiency: with each question, we don't need to reduce the number of possibilities by 1/2, but rather we need to distinguish between two equally probable outcomes.

Definition, attempt #2: We gain one bit of information each time we distinguish between two equally likely events.

- The amount of information contained in an experiment is the number of "probability bisections" required (on average) to determine the outcome.

- $H(p) =$ amount of information contained in an experiment with outcome probabilities $p_1, \ldots, p_k$
What is information?
What is information?

Alternate perspective:
What is information?

Alternate perspective:

- Say the "information gained" from observing an event of probability $a$ is $f(a) := -\log_2 a$
What is information?

Alternate perspective:

• Say the "information gained" from observing an event of probability \( a \) is \( f(a) := -\log_2 a \)

\[
H(p) = \sum_{i=1}^{k} p_i(-\log_2 p_i)
\]

• \( H(p) = \sum_{i=1}^{k} p_i(-\log_2 p_i) \) is the average (expected) information gained from observing an experiment with outcome probabilities \( p_1, \ldots, p_k \)
What is information?
What is information?

Alternate perspective #2 (axiomatic approach):
What is information?

Alternate perspective #2 (axiomatic approach):

- Information function $f$ should have some desirable properties:
What is information?

Alternate perspective #2 (axiomatic approach):

- Information function $f$ should have some desirable properties:
  - Independent events add information: $f(xy) = f(x) + f(y)$
  - Rarer events give more information: $f$ is decreasing
  - Normalization: $f(1/2) = 1$
What is information?

Alternate perspective #2 (axiomatic approach):

- Information function $f$ should have some desirable properties:
  - Independent events add information: $f(xy) = f(x) + f(y)$
  - Rarer events give more information: $f$ is decreasing
  - Normalization: $f(1/2) = 1$

$$f(x) = - \log_2(x) \, \text{ is the only such function!}$$
Outline

1. What is information?

2. Data compression

3. Data transmission
Data compression

- Also known as source coding

- Encode data into 0s and 1s in an injective way (lossless compression)

- Goal: minimize number of bits needed to encode
Data compression, formally

• **alphabet** that you want to encode (e.g. $A = \{a, b, c, \ldots, z\}$)

• **Encoder** = injective map $f : A \rightarrow \{0,1\}^* = \bigcup_{n=1}^{\infty} \{0,1\}^n = \text{all finite strings of 0s and 1s}$
Example

• Fixed-length code
  • $A = \{a, b, c, d\}$
  • $f(a) = 00$, $f(b) = 01$, $f(c) = 10$, $f(d) = 11$

• Not very efficient, in fact no compression at all
More data compression
More data compression

- **Idea:** gain efficiency by considering relative frequencies of letters in $A$
More data compression

- **Idea:** gain efficiency by considering relative frequencies of letters in $A$

  - Equip $A$ with probability distribution $p = (p(a))_{a \in A}$ that indicates relative frequencies
More data compression

- **Idea:** gain efficiency by considering relative frequencies of letters in $A$
  
  - Equip $A$ with probability distribution $p = (p(a))_{a \in A}$ that indicates relative frequencies

- **Goal:** define $f$ to minimize $\mathbb{E}_p |f(a)| = \sum_{a \in A} p(a) |f(a)|$
More data compression

• **Idea:** gain efficiency by considering relative frequencies of letters in $A$

  • Equip $A$ with probability distribution $p = (p(a))_{a \in A}$ that indicates relative frequencies

• **Goal:** define $f$ to minimize $\mathbb{E}_p |f(a)| = \sum_{a \in A} p(a) |f(a)|$

• We can save time on average by reserving shorter code words for more common letters
Prefix codes
Prefix codes

• One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$.
Prefix codes

- One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

- Allows decoding in real time
Prefix codes

• One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

• Allows decoding in real time

0
Prefix codes

- One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

- Allows decoding in real time
Prefix codes

- One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

- Allows decoding in real time

01
b
Prefix codes

- One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

- Allows decoding in real time

011

b
Prefix codes

• One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

• Allows decoding in real time

0111

b
Prefix codes

• One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

• Allows decoding in real time

0111
b d
Prefix codes

- One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

- Allows decoding in real time

```
01110
b d
```
Prefix codes

- One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$
- Allows decoding in real time

011100
b d
Prefix codes

- One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$
- Allows decoding in real time

011100

b d a
Prefix codes

• One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

• Allows decoding in real time

0111001
b d a
Prefix codes

- One more desirable property for encoder \( f \): require that for all distinct \( a, b \in A \), \( f(a) \) is not a prefix of \( f(b) \)
- Allows decoding in real time

\[ 01110011 \]
\[ b \quad d \quad a \]
Prefix codes

- One more desirable property for encoder $f$: require that for all distinct $a, b \in A$, $f(a)$ is not a prefix of $f(b)$

- Allows decoding in real time

01110011
b d a d
Example
Example

• Variable-length code
  • $A = \{a, b, c, d\}$
  • $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = p(d) = 1/8$
  • $f(a) = 0$, $f(b) = 10$, $f(c) = 110$, $f(d) = 111$
Example

• Variable-length code
  • \( A = \{a, b, c, d\} \)
  • \( p(a) = 1/2, \ p(b) = 1/4, \ p(c) = p(d) = 1/8 \)
  • \( f(a) = 0, \ f(b) = 10, \ f(c) = 110, \ f(d) = 111 \)
  
• \( \mathbb{E}_p |f(a)| = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1.75 \)
  
• More efficient (on average) than fixed-length code (2 bits/letter)
Example

- Variable-length code
  - $A = \{a, b, c, d\}$
  - $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = p(d) = 1/8$
  - $f(a) = 0$, $f(b) = 10$, $f(c) = 110$, $f(d) = 111$
  - $\mathbb{E}_p |f(a)| = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1.75$
  - More efficient (on average) than fixed-length code (2 bits/letter)

- Does this look familiar?
Example

- Variable-length code
  - \( A = \{a, b, c, d\} \)
  - \( p(a) = 1/2, \ p(b) = 1/4, \ p(c) = p(d) = 1/8 \)
  - \( f(a) = 0, \ f(b) = 10, \ f(c) = 110, \ f(d) = 111 \)
  - \( \mathbb{E}_p |f(a)| = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 1.75 \)
    - More efficient (on average) than fixed-length code (2 bits/letter)

- Does this look familiar?
  \[
  \begin{array}{cccc}
  1 & 2 & 3 & 4 \\
  \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8}
  \end{array}
  \]
The return of entropy
The return of entropy

• It seems that the entropy of the frequency distribution is related to the efficiency of prefix codes
The return of entropy

- It seems that the entropy of the frequency distribution is related to the efficiency of prefix codes.

- **(a version of) Shannon's source coding theorem:** Let $A$ be an alphabet equipped with a probability (frequency) distribution $p = (p(a))_{a \in A}$. Then any prefix code $f : A \to \{0,1\}^*$ satisfies $\mathbb{E}_p |f(a)| \geq H(p)$. Moreover, there always exists a code $f$ with $\mathbb{E}_p |f(a)| \approx H(p)$. 
Proof of source coding theorem

- A prefix code is like a section of a binary tree
Proof of source coding theorem

- A prefix code is like a section of a binary tree
Proof of source coding theorem

- A prefix code is like a section of a binary tree
Proof of source coding theorem

• A prefix code is like a section of a binary tree
Proof of source coding theorem

- A prefix code is like a section of a binary tree
Proof of source coding theorem

• A prefix code is like a section of a binary tree

• This controls the length profile of the code:
  \[
  \sum_{a \in A} 2^{-|f(a)|} \leq 1
  \]
Proof of source coding theorem
Proof of source coding theorem

- Compare $\mathbb{E}_p |f(a)|$ to $H(p) = \mathbb{E}_p (-\log_2 p(a))$
Proof of source coding theorem

• Compare $\mathbb{E}_p |f(a)|$ to $H(p) = \mathbb{E}_p (-\log_2 p(a))$

• $\mathbb{E}_p |f(a)| - \mathbb{E}_p (-\log_2 p(a)) = \mathbb{E}_p \left[ -\log_2 \left( \frac{2^{-|f(a)|} / p(a)}{2} \right) \right]$

  $\geq - \log_2 \left( \mathbb{E}_p \left[ 2^{-|f(a)|} / p(a) \right] \right)$ \hspace{1cm} (Jensen)

  $= - \log_2 \left( \sum_{a \in A} 2^{-|f(a)|} \right)$ \hspace{1cm} (previous slide)

  $\geq 0$
Proof of source coding theorem

• Compare $\mathbb{E}_p |f(a)|$ to $H(p) = \mathbb{E}_p (-\log_2 p(a))$

• $\mathbb{E}_p |f(a)| - \mathbb{E}_p (-\log_2 p(a)) = \mathbb{E}_p \left[ -\log_2 \left( 2^{-|f(a)|} / p(a) \right) \right]$

  $\geq - \log_2 \left( \mathbb{E}_p \left[ 2^{-|f(a)|} / p(a) \right] \right)$  \hspace{1cm} (Jensen)

  $= - \log_2 \left( \sum_{a \in A} 2^{-|f(a)|} \right)$  \hspace{1cm} (previous slide)

  $\geq 0$

• Equality in Jensen when $2^{-|f(a)|} / p(a)$ is constant in $a$, i.e. $|f(a)| = - \log_2 p(a)$
Proof of source coding theorem

1. Compare $\mathbb{E}_p |f(a)|$ to $H(p) = \mathbb{E}_p (-\log_2 p(a))$

2. $\mathbb{E}_p |f(a)| - \mathbb{E}_p (-\log_2 p(a)) = \mathbb{E}_p \left[ -\log_2 \left( 2^{-|f(a)|}/p(a) \right) \right]$

   $\geq -\log_2 \left( \mathbb{E}_p \left[ 2^{-|f(a)|}/p(a) \right] \right)$ \hspace{1cm} \text{(Jensen)}

   $= -\log_2 \left( \sum_{a \in A} 2^{-|f(a)|} \right)$ \hspace{1cm} \text{(previous slide)}

   $\geq 0$

3. Equality in Jensen when $2^{-|f(a)|}/p(a)$ is constant in $a$, i.e.

   $|f(a)| = -\log_2 p(a)$

   a. This length profile satisfies $\sum_{a \in A} 2^{-|f(a)|} = 1$, so by a greedy algorithm one can define a corresponding prefix code
Interpretation
Interpretation

• The maximum efficiency prefix code has length profile

\[ |f(a)| = -\log_2 p(a), \quad a \in A \]
Interpretation

• The maximum efficiency prefix code has length profile

\[ |f(a)| = - \log_2 p(a), \quad a \in A \]

• The moral of the story: to achieve maximum efficiency, each letter gets coded with exactly the number of bits of information that its occurrence conveys
Interpretation

• The maximum efficiency prefix code has length profile

\[ |f(a)| = -\log_2 p(a), \quad a \in A \]

• The moral of the story: to achieve maximum efficiency, each letter gets coded with exactly the number of bits of information that its occurrence conveys

• Afterthought: \( \approx \) appears when the \( p(a) \) aren't perfect powers of \( 1/2 \)
Outline

1. What is information?

2. Data compression

3. Data transmission
Data transmission

- Also known as **channel coding**
- Send data through a noisy channel, some distortion happens
- Goal: find a way to transmit to maximize accuracy
Formalism (attempt #1)
Formalism (attempt #1)

- $B = \text{alphabet of transmitted message}, \ B' = \text{alphabet of received message}$
  - Different alphabets allow for possibility of corruption
Formalism (attempt #1)

- $B =$ alphabet of transmitted message, $B'$ = alphabet of received message
  - Different alphabets allow for possibility of corruption
- $\theta = \left( \theta_{bb'} \right)_{b \in B, b' \in B'} = \left( \theta(b' | b) \right)_{b \in B, b' \in B'}$ is a channel (or probability kernel or stochastic matrix):
Formalism (attempt #1)

- $B = \text{alphabet of transmitted message, } B' = \text{alphabet of received message}$
  - Different alphabets allow for possibility of corruption

- $\theta = (\theta_{bb'})_{b \in B, b' \in B'} = (\theta(b' \mid b))_{b \in B, b' \in B'}$ is a **channel** (or probability kernel or stochastic matrix):

$$\theta(b' \mid b) = \text{probability that } b' \text{ is received, given that } b \text{ is sent}$$
Formalism (attempt #1)

- $B$ = alphabet of transmitted message, $B'$ = alphabet of received message

- Different alphabets allow for possibility of corruption

- $\theta = \left( \theta_{bb'} \right)_{b \in B, b' \in B'} = \left( \theta(b' | b) \right)_{b \in B, b' \in B'}$ is a **channel** (or **probability kernel** or **stochastic matrix**):

  $\theta(b' | b) = \text{probability that } b' \text{ is received, given that } b \text{ is sent}$

- Example: $B = \{0,1\}$, $B' = \{0,1,e\}$, $\theta = \begin{pmatrix} .95 & .01 & .04 \\ .01 & .95 & .04 \end{pmatrix}$
Formalism (attempt #1)
Formalism (attempt #1)

- A **decoder** is a function $g : B' \rightarrow B$
Formalism (attempt #1)

- A **decoder** is a function $g : B' \rightarrow B$

- Goal: make the **worst-case probability of error**
  \[ p_e = \max_{b \in B} \theta \left( \left\{ b' : g(b') \neq b \right\} \mid b \right) \] as small as possible
Formalism (attempt #1)

• A decoder is a function $g : B' \to B$

• Goal: make the worst-case probability of error
  
  \[ p_e = \max_{b \in B} \theta(\{b' : g(b') \neq b\} | b) \text{ as small as possible} \]

• Naive strategy: Send each letter 3 times ($B' = B^3$) and define $g$ by majority rule
A decoder is a function \( g : B' \rightarrow B \)

Goal: make the worst-case probability of error
\[
p_e = \max_{b \in B} \theta(\{b' : g(b') \neq b\} | b)
\]
as small as possible

Naive strategy: Send each letter 3 times \((B' = B^3)\) and define \( g \) by majority rule

Slightly better strategy: set
\[
g(b') = \arg\max_{b \in B} \mathbb{P}(b \text{ sent} | b' \text{ received})
\]
(Bayesian maximum likelihood estimator)
Formalism (attempt #2)
Formalism (attempt #2)

- New idea: transmit letters from $B$ in blocks of length $N \gg 1$

\[(B \mapsto B^N, B' \mapsto (B')^N, \theta \mapsto \theta^N)\]
Formalism (attempt #2)

- New idea: transmit letters from $B$ in blocks of length $N \gg 1$
  $\left( B \mapsto B^N, B' \mapsto (B')^N, \theta \mapsto \theta^N \right)$

- Idea: try to pick a special subset $\mathcal{A}_N \subseteq B^N$ of "acceptable words" that are very unlikely to be confused with each other, and only transmit those
Formalism (attempt #2)

• New idea: transmit letters from $B$ in blocks of length $N \gg 1$
  
  
  
  \[
  \begin{align*}
  B &\mapsto B^N, \\
  B' &\mapsto (B')^N, \\
  \theta &\mapsto \theta^N
  \end{align*}
  \]

• Idea: try to pick a special subset $\mathcal{A}_N \subseteq B^N$ of "acceptable words" that are very unlikely to be confused with each other, and only transmit those

• New goal: make $|\mathcal{A}_N|$ as large as possible while keeping $p_e$ as small as possible
Formalism (attempt #2)

• New idea: transmit letters from $B$ in blocks of length $N \gg 1$
  
  \[
  (B \mapsto B^N, B' \mapsto (B')^N, \theta \mapsto \theta^N)
  \]

• Idea: try to pick a special subset $\mathcal{A}_N \subseteq B^N$ of "acceptable words" that are very unlikely to be confused with each other, and only transmit those

• New goal: make $|\mathcal{A}_N|$ as large as possible while keeping $p_e$ as small as possible

• Toy example: $B = B' = \{0,1\}, \theta = \begin{pmatrix}
  .99 & .01 \\
  .01 & .99
\end{pmatrix}$

• Let $\mathcal{A}_N$ be a subset of $B^N$ with the property that any two strings in $\mathcal{A}_N$ differ in at least $.03N$ letters. When $N$ is huge it is virtually impossible for any of these to get confused for any other.
Channel capacity
Channel capacity

- Say a number $R$ is an **achievable rate** if it is possible to choose acceptable words $\mathcal{A}_N$ and decoder $g$ such that $|\mathcal{A}_N| \geq 2^{RN}$ and $p_e$ is arbitrarily small.

- Maximum possible rate = $\log_2 |B|$
Channel capacity

• Say a number $R$ is an **achievable rate** if it is possible to choose acceptable words $\mathcal{A}_N$ and decoder $g$ such that $|\mathcal{A}_N| \geq 2^{RN}$ and $p_e$ is arbitrarily small.

  • Maximum possible rate = $\log_2 |B|$.

• The **channel capacity** of a channel $\theta$ is $C(\theta) := \sup$ of all achievable rates.

  • Most information that can be transmitted per unit time, subject to the constraint of high accuracy.
Mutual information
Mutual information

- Given an input frequency distribution \( q \) on \( B \), the channel \( \theta \) induces an output frequency distribution \( q'_\theta \) on \( B' \) and a joint input-output distribution \( q \triangleleft \theta \) on \( B \times B' \)

- \((q \triangleleft \theta)(b, b') = q(b)\theta(b' \mid b) = \) prob. that \( b \) is sent and \( b' \) is received

- \( q'_\theta(b') = \sum_{b \in B} q(b)\theta(b' \mid b) = \) total prob. that \( b' \) is received
Mutual information

- Given an input frequency distribution \( q \) on \( B \), the channel \( \theta \) induces an output frequency distribution \( q'_\theta \) on \( B' \) and a joint input-output distribution \( q \bowtie \theta \) on \( B \times B' \)

- \( (q \bowtie \theta)(b, b') = q(b)\theta(b'|b) = \) prob. that \( b \) is sent and \( b' \) is received

- \( q'_\theta(b') = \sum_{b \in B} q(b)\theta(b'|b) = \) total prob. that \( b' \) is received

- The mutual information between \( q \) and \( \theta \) is
  \[ I(q, \theta) := H(q) + H(q'_\theta) - H(q \bowtie \theta) \]

- Measures how much information is faithfully transmitted by \( \theta \)

- \( \theta = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \) (perfect transmission) \( \longrightarrow I(q, \theta) = H(q) \)

- \( \theta = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \) (total corruption) \( \longrightarrow I(q, \theta) = 0 \)
Channel coding theorem
Channel coding theorem

- Shannon's channel coding theorem:

\[ C(\theta) = \sup_{q \in \text{Prob}(B)} I(q, \theta) \]
Channel coding theorem

- Shannon's channel coding theorem:

  \[
  C(\theta) = \sup_{q \in \text{Prob}(B)} I(q, \theta)
  \]

- Maximum rate of information that can be passed accurately through \( \theta \) is determined by how much entropy \( \theta \) can transport from \( B \) to \( B' \)
Examples

0 \rightarrow^1 0

1 \rightarrow^1 1
Examples

• Noiseless channel

\[\begin{align*}
0 & \quad \xrightarrow{1} \quad 0 \\
1 & \quad \xrightarrow{1} \quad 1
\end{align*}\]
Examples

• Noiseless channel

\[ \begin{align*}
0 & \xrightarrow{1} 0 \\
1 & \xrightarrow{1} 1
\end{align*} \]

• \( I(q, \theta) = H(q) \) for any input distribution \( q \), maximized when \( q = (1/2, 1/2) \)

• \( C(\theta) = 1 \)
Examples
Examples

- Binary erasure channel

```
0 → 0 with probability 0.99
0.01
0.10
1 → e with probability 0.10
0.90
1 → 1
```
Examples

• Binary erasure channel

![Diagram]

• Might expect to maximize efficiency by being biased towards 0
Examples

• Binary erasure channel

0 \rightarrow 1
0.99
0.01

1 \rightarrow 0
0.10
0.90

e

• Might expect to maximize efficiency by being biased towards 0

• But recall: goal is to maximize accurate decodability, not error-free transmission
Examples

- Binary erasure channel

- Might expect to maximize efficiency by being biased towards 0

- But recall: goal is to maximize accurate decodability, not error-free transmission

  - If e is received, it probably came from 1

  - Actually more efficient to bias a bit towards 1: \( C(\theta) = 0.976 \), achieved by \( \mathbb{P}(0) = 0.496 \)
Examples
Examples

• Noisy typewriter
Examples

• Noisy typewriter

• Notice that A, C, E, G, ... can't be confused for each other
Examples

- Noisy typewriter

  - Notice that A, C, E, G, ... can't be confused for each other

  - It turns out that the best input distribution is the choose uniformly from the uniquely decodable subset

  - $C(\theta) = \log_2 13$