The Golden Ratio in Communication

Blackwell’s Trapdoor Channel

Task Assignment

Paul Cuff
(Cover, Permuter, Weissman, Van Roy)

Stanford University

November 24, 2008
Golden Ratio

\[1, 1, 2, 3, 5, 8, 13, \ldots\]
The Trapdoor Channel

Introduced by David Blackwell in 1961. [Ash 65], [Ahlswede & Kaspi 87], [Ahlswede 98], [Kobayashi 02 & 03].

A “simple two-state channel.” - Blackwell
The Trapdoor Channel

\[
s_t = s_{t-1} + x_t - y_t
\]

\[s_0 = 0\]
The Trapdoor Channel

Input

\[
\begin{array}{cc}
1 & 0 \\
\end{array}
\]

Channel

\[
\begin{array}{c}
0 \\
1 \\
\end{array}
\]

Output

\[
s_t = s_{t-1} + x_t - y_t
\]

\[
s_0 = 0
\]

\[
x_1 = 1,
\]
The Trapdoor Channel

\[ s_t = s_{t-1} + x_t - y_t \]

\[ s_0 = 0 \]
\[ x_1 = 1, \quad s_1 = 1, \quad y_1 = 0, \]
The Trapdoor Channel

\[ s_t = s_{t-1} + x_t - y_t \]

\[ s_0 = 0 \]
\[ x_1 = 1, \ s_1 = 1, \ y_1 = 0, \]
\[ x_2 = 0, \]
The Trapdoor Channel

\[ s_t = s_{t-1} + x_t - y_t \]

\[ s_0 = 0 \]
\[ x_1 = 1, \ s_1 = 1, \ y_1 = 0, \]
\[ x_2 = 0, \ s_2 = 1, \ y_2 = 0, \]
The Trapdoor Channel

\[
s_t = s_{t-1} + x_t - y_t
\]

\[
\begin{align*}
 s_0 &= 0 \\
 x_1 &= 1, \quad s_1 = 1, \quad y_1 = 0, \\
 x_2 &= 0, \quad s_2 = 1, \quad y_2 = 0, \\
 x_3 &= 1,
\end{align*}
\]
The Trapdoor Channel

\[ s_t = s_{t-1} + x_t - y_t \]

\[
\begin{align*}
  s_0 &= 0, \\
  x_1 &= 1, \ s_1 = 1, \ y_1 = 0, \\
  x_2 &= 0, \ s_2 = 1, \ y_2 = 0, \\
  x_3 &= 1, \ s_3 = 1, \ y_3 = 1.
\end{align*}
\]
Communication without Feedback

- Repeat each bit three times: $R = 1/3$ bit.
Communication without Feedback

- Repeat each bit three time: $R = 1/3$ bit.
- Repeat each bit twice: $R = 1/2$ bit. [Ahlswede & Kaspi 87]
Communication without Feedback

- Repeat each bit three time: $R = 1/3$ bit.
- Repeat each bit twice: $R = 1/2$ bit. [Ahlswede & Kaspi 87]
- $C \approx 0.572$ bits per channel use. [Kobayashi & Morita 03]

\[
C = \lim_{n \to \infty} \max_{p(x_1, \ldots, x_n)} \frac{1}{n} I(X^n; Y^n).
\]

![Graph showing the relation between $\frac{1}{n} I(X^n; Y^n)$ and $n$.]
Communication Setting (with feedback)

Encoder: $x_t(m, y^{t-1})$

Chemical Channel: $p(y_t | x^t, y^{t-1})$

Decoder: $\hat{m}(y^N)$

Feedback: $y_{t-1}$

Unit Delay: $y_t$

Message: $m$

Estimated message: $\hat{m}$

Figure: Communication with feedback
Feedback Capacity of Unifilar, Strongly Connected, FSC

- Capacity of the Trapdoor Channel

\[ C = \lim_{n \to \infty} \frac{1}{n} \max_{p(x_1, \ldots, x_n)} I(X^n; Y^n) \]

- Feedback capacity of the Trapdoor Channel

\[ C_{FB} = \lim_{n \to \infty} \frac{1}{n} \max_{\{p(x_i|x^{i-1}, y^{i-1})\}_{i=1}^n} I(X^n \to Y^n) \]

[Permuter, Weissman & Goldsmith ISIT06]
Directed Information

Mutual Information

\[
I(X^n; Y^n) = \sum_{i=1}^{n} I(X^n; Y_i | Y^{i-1})
\]

Directed Information was defined by Massey in 1990.

\[
I(X^n \rightarrow Y^n) \triangleq \sum_{i=1}^{n} I(X^i; Y_i | Y^{i-1})
\]
Directed Information

Mutual Information

\[ I(X^n; Y^n) = \sum_{i=1}^{n} I(X^n; Y_i | Y^{i-1} ) \]

Directed Information was defined by Massey in 1990.

\[ I(X^n \rightarrow Y^n) \triangleq \sum_{i=1}^{n} I(X^i; Y_i | Y^{i-1} ) \]

Intuition [Massey 05]:

\[ I(X^n; Y^n) = I(X^n \rightarrow Y^n) + I(Y^{n-1} \rightarrow X^n) \]
Feedback Capacity of Unifilar, Strongly Connected, FSC

$$C_{FB} = \lim_{N \to \infty} \frac{1}{N} \max_{t=1} \{p(x_t|s_{t-1},y_{t-1})\} I(X^N \to Y^N)$$

$$= \lim_{N \to \infty} \frac{1}{N} \max_{t=1} \sum_{t=1}^{N} I(X^t; Y_t|Y^{t-1})$$

$$= \lim_{N \to \infty} \frac{1}{N} \max_{t=1} \sum_{t=1}^{N} I(X_t, S_{t-1}; Y_t|Y^{t-1})$$

$$= \sup \lim_{N \to \infty} \inf \frac{1}{N} \sum_{t=1}^{N} I(X_t, S_{t-1}; Y_t|Y^{t-1})$$
Dynamic Programming (infinite horizon, average reward)

Variable Assignments

State:  \( \beta_t = p(s_t|y^t) \)

Action:  \( u_t = p(x_t|s_{t-1}) \)

Disturbance:  \( w_t = y_{t-1} \)

Dynamic Programming Requirements

State evolution:

\[ \beta_t = F(\beta_{t-1}, u_t, w_t) \]

Reward function per unit time:

\[ g(\beta_{t-1}, u_t) = I(X_t, S_{t-1}; Y_t|\beta_{t-1}) \]

Similar dynamic programming approaches:

- Yang, Kavčić & Tatikonda - 2005
- Chen & Berger - 2005
- Mitter & Tatikonda - 2006
Dynamic Programming (infinite horizon, average reward)

**Dynamic Programming Operator $T$**

The dynamic programming operator $T$ is given by

$$T \circ J(\beta) = \sup_{u \in \mathcal{U}} \left( g(\beta, u) + \int P_w(dw|\beta, u) J(F(\beta, u, w)) \right).$$

**Bellman Equation**

If there exist a function $J(\beta)$ and constant $\rho$ that satisfy

$$J(\beta) = T \circ J(\beta) - \rho$$

then $\rho$ is the optimal infinite horizon average reward.
Trapdoor Channel—20th Value iteration

\[ J_{20} \]

\[ \delta, \text{ iteration}=20 \]

\[ \gamma, \text{ iteration}=20 \]

\[ \text{histogram of beta} \]
Calculation and Coincidence

Trapdoor channel: $C_{FB} \approx 0.694$ bits
Calculation and Coincidence

Trapdoor channel: \( C_{FB} \approx 0.694 \) bits

Homework Question

*Entropy rate.* Find the maximum entropy rate of the following two-state Markov chain:

![Diagram of a two-state Markov chain with transition probabilities p and 1-p. State A transitions to State B with probability 1-p, and State B transitions to State A with probability 1-p.]
Calculation and Coincidence

Trapdoor channel: \( C_{FB} \approx 0.694 \) bits

Homework Question

*Entropy rate.* Find the maximum entropy rate of the following two-state Markov chain:

\[
\begin{align*}
\mathcal{X} &\rightarrow A \xrightarrow{p} B \xrightarrow{1-p} \mathcal{X} \\
\mathcal{X} &\rightarrow B \xrightarrow{1} A
\end{align*}
\]

*Solution:* (Golden Ratio: \( \phi = \frac{\sqrt{5}+1}{2} \))

\[
p^* = \phi - 1 = \frac{1}{\phi}
\]

\[
H(\mathcal{X}) = \log \phi = 0.6942\ldots \text{ bits}
\]
Calculation and Coincidence

Trapdoor channel: \( C_{FB} \approx 0.694 \) bits

Homework Question

Entropy rate. Find the maximum entropy rate of the following two-state Markov chain:

\[
\begin{array}{ccc}
A & \xrightarrow{p} & B \\
B & \xrightarrow{1-p} & A \\
\end{array}
\]

Solution: (Golden Ratio: \( \phi = \frac{\sqrt{5}+1}{2} \))

\[
p^* = \phi - 1 = \frac{1}{\phi}
\]

\[
H(X) = \log \phi = 0.6942... \text{ bits}
\]
Equivalent Channel Formulation

Rename channel input: $\tilde{x}_t = x_t \oplus s_{t-1}$.
Rename channel output: $\tilde{y}_t = y_t \oplus y_{t-1}$. 
Equivalent Channel Formulation

Rename channel input: \( \tilde{x}_t = x_t \oplus s_{t-1} \).
Rename channel output: \( \tilde{y}_t = y_t \oplus y_{t-1} \).

Case 1: \( \tilde{x}_t = 0 \)

\[
\tilde{x}_t = 0 \quad \Rightarrow \quad \tilde{y}_t = \tilde{x}_{t-1}.
\]
Equivalent Channel Formulation

Rename channel input: \( \tilde{x}_t = x_t \oplus s_{t-1} \).
Rename channel output: \( \tilde{y}_t = y_t \oplus y_{t-1} \).

Case 1: \( \tilde{x}_t = 0 \)

<table>
<thead>
<tr>
<th>Proof:</th>
<th>( x_t = s_{t-1} = y_t = s_t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tilde{x}_t = 0 ) ( \Rightarrow ) ( \tilde{y}<em>t = \tilde{x}</em>{t-1} ).</td>
<td></td>
</tr>
</tbody>
</table>
Equivalent Channel Formulation

Rename channel input: \( \tilde{x}_t = x_t \oplus s_{t-1} \).
Rename channel output: \( \tilde{y}_t = y_t \oplus y_{t-1} \).

Case 1: \( \tilde{x}_t = 0 \)

\[
\begin{align*}
\tilde{x}_t &= 0 & \Rightarrow & & \tilde{y}_t &= \tilde{x}_{t-1}.
\end{align*}
\]

Proof:

\[
\begin{align*}
x_t &= s_{t-1} = y_t = s_t \\
\end{align*}
\]

\[
\begin{align*}
x_{t-1} \oplus s_{t-2} &= y_{t-1} \oplus s_{t-1}
\end{align*}
\]
Equivalent Channel Formulation

Rename channel input: \( \tilde{x}_t = x_t \oplus s_{t-1} \).
Rename channel output: \( \tilde{y}_t = y_t \oplus y_{t-1} \).

**Case 1: \( \tilde{x}_t = 0 \)**

\[
\begin{align*}
\tilde{x}_t = 0 & \quad \Rightarrow \quad \tilde{y}_t = \tilde{x}_{t-1}.
\end{align*}
\]

**Proof:**
\[
\begin{align*}
x_t = s_{t-1} = y_t = s_t \\
 x_{t-1} \oplus s_{t-2} & = y_{t-1} \oplus s_{t-1} \\
x_{t-1} \oplus s_{t-2} & = y_{t-1} \oplus y_t
\end{align*}
\]
Equivalent Channel Formulation

Rename channel input: \( \tilde{x}_t = x_t \oplus s_{t-1} \).
Rename channel output: \( \tilde{y}_t = y_t \oplus y_{t-1} \).

Case 1: \( \tilde{x}_t = 0 \)

\[
\begin{align*}
\tilde{x}_t = 0 & \quad \Rightarrow \quad \tilde{y}_t = \tilde{x}_{t-1}. \\
\text{Proof:} & \quad x_t = s_{t-1} = y_t = s_t \\
& \quad x_{t-1} \oplus s_{t-2} = y_{t-1} \oplus s_{t-1} \\
& \quad x_{t-1} \oplus s_{t-2} = y_{t-1} \oplus y_t
\end{align*}
\]

Case 2: \( \tilde{x}_t = 1 \)

\[
\begin{align*}
\tilde{x}_t = 1 & \quad \Rightarrow \quad \tilde{y}_t \sim \text{Bern}(1/2), \text{ independent of the past.}
\end{align*}
\]
Equivalent Channel

- Two states specify whether the channel is all noise or noise-free.

\[
\begin{align*}
\tilde{X}_{t-1} & \quad \tilde{X}_t = 0 & \quad 0 \\
\tilde{Y}_t & \quad 0 \\
1 & \quad 1 \\
\end{align*}
\]

\[
\begin{align*}
\tilde{X}_{t-1} & \quad \tilde{X}_t = 1 & \quad 0 \\
\tilde{Y}_t & \quad 1/2 \\
1 & \quad 1 \\
\end{align*}
\]
Equivalent Channel

- Two states specify whether the channel is all noise or noise-free.

\[
\begin{align*}
\tilde{X}_t = 0 & \quad \tilde{X}_t = 1 \\
\tilde{X}_{t-1} & \quad \tilde{Y}_t \\
0 & \quad 0 \\
1 & \quad 1 \\
1/2 & \quad 1/2
\end{align*}
\]

- Use Markov input

\[
\begin{align*}
p & \quad 1 - p \\
0 & \quad 1 \\
1 & \quad 1
\end{align*}
\]
Decoding Example

- Transmitter promises to never use $\tilde{x} = 1$ twice in a row.
- The number of such sequences of length $n$ is the Fibonacci sequence.

**Decoding rules**

\[
\begin{align*}
\tilde{x}_t &= 0 & \Rightarrow & \tilde{x}_{t-1} &= \tilde{y}_t \\
\tilde{x}_t &= 1 & \Rightarrow & \tilde{x}_{t-1} &= 0
\end{align*}
\]

$\tilde{x}^n$: 0

$\tilde{y}^n$: 1 1 1 1 0 0 1
Decoding Example

- Transmitter promises to never use $\tilde{x} = 1$ twice in a row.
- The number of such sequences of length $n$ is the Fibonacci sequence.

Decoding rules

\[ \begin{align*}
\tilde{x}_t &= 0 & \Rightarrow & & \tilde{x}_{t-1} = \tilde{y}_t \\
\tilde{x}_t &= 1 & \Rightarrow & & \tilde{x}_{t-1} = 0
\end{align*} \]

$\tilde{x}^n$: 0 0

$\tilde{y}^n$: 1 1 1 1 0 0 1
Decoding Example

- Transmitter promises to never use $\tilde{x} = 1$ twice in a row.
- The number of such sequences of length $n$ is the Fibonacci sequence.

Decoding rules

$$\tilde{x}_t = 0 \quad \Rightarrow \quad \tilde{x}_{t-1} = \tilde{y}_t$$
$$\tilde{x}_t = 1 \quad \Rightarrow \quad \tilde{x}_{t-1} = 0$$

$\tilde{x}^n$: 1 0 0

$\tilde{y}^n$: 1 1 1 1 0 0 1
Decoding Example

- Transmitter promises to never use $\tilde{x} = 1$ twice in a row.
- The number of such sequences of length $n$ is the Fibonacci sequence.

**Decoding rules**

\[
\begin{align*}
\tilde{x}_t &= 0 \quad \Rightarrow \quad \tilde{x}_{t-1} = \tilde{y}_t \\
\tilde{x}_t &= 1 \quad \Rightarrow \quad \tilde{x}_{t-1} = 0
\end{align*}
\]

$\tilde{x}^n$: 0 1 0 0  
$\tilde{y}^n$: 1 1 1 1 0 0 1
Decoding Example

- Transmitter promises to never use $\tilde{x} = 1$ twice in a row.
- The number of such sequences of length $n$ is the Fibonacci sequence.

Decoding rules

\[
\begin{align*}
\tilde{x}_t &= 0 \quad \Rightarrow \quad \tilde{x}_{t-1} = \tilde{y}_t \\
\tilde{x}_t &= 1 \quad \Rightarrow \quad \tilde{x}_{t-1} = 0
\end{align*}
\]

$\tilde{x}^n$: 1 0 1 0 0
$\tilde{y}^n$: 1 1 1 0 0 1
Dynamic Programming 20th Value iteration

Bellman Equation: $J(\beta) = T \circ J(\beta) - \rho$. 
Conjectured Solution to the Bellman Equation

Bellman Equation: $J(\beta) = T \circ J(\beta) - \rho$. 
Feedback Capacity and Zero-Error Capacity

Bellman Equation is satisfied.

Trapdoor channel feedback capacity:

\[ C_{FB} = \log \phi = 0.6942\ldots \text{ bits.} \]

\[ C \approx .572 \text{ bits.} \]
Feedback Capacity and Zero-Error Capacity

Bellman Equation is satisfied.

Trapdoor channel feedback capacity:

\[ C_{FB} = \log \phi = 0.6942 \ldots \text{bits.} \]
\[ C \approx 0.572 \text{ bits.} \]

Trapdoor channel zero-error capacity:

\[ C_{FB} = \log \phi. \]
\[ C = 0.5 \text{ bits.} \]

[Ahlswede & Kaspi 87]
Biochemical interpretation of the trapdoor channel [Berger 71]
Biological Cells

Biochemical interpretation of the trapdoor channel [Berger 71]
Biological Cells

Biochemical interpretation of the trapdoor channel [Berger 71]
Biochemical interpretation of the trapdoor channel [Berger 71]
Coordinated Action in a Network

\[ X_1 \xrightarrow{\ M_1 \} X_2 \xrightarrow{\ M_2 \} \]
\[ \downarrow \]
\[ X_3 \xrightarrow{\ M_3 \} X_4 \]
\[ \downarrow \]
\[ X_5 \xrightarrow{\ M_4 \} \]
Task Assignment

Computation Tasks: $T = \{0, 1, 2\}$
Processors: $X$, $Y$, and $Z$

$X^n \sim iid, \ Unif(\{0, 1, 2\})$

What are the required rates $(R_1, R_2)$?
Task Assignment

Computation Tasks: $T = \{0, 1, 2\}$
Processors: $X$, $Y$, and $Z$

$X^n \sim iid, \ Unif(\{0, 1, 2\})$

$M_1 \in \{1, \ldots, 2^{nR_1}\}$

What are the required rates $(R_1, R_2)$?
Idea 1—Describe $X^n$:

$$R_1 = R_2 = H(X) = \log(3).$$

A protocol tells $Y$ and $Z$ how to orient around $X$. 
Task Assignment: Achievable Rates

Idea 2—Minimize $R_1$:

$$R_1 \geq \min_{p(y|x):X \neq Y} I(X;Y)$$

$$= H(X) - \max_{p(y|x):X \neq Y} H(Y|X)$$

$$= \log 3 - \log 2.$$

Describe $Z^n$ at full rate: $R_2 = \log 3$. 
Task Assignment: Achievable Rates

Idea 3—Restrict the support of $Y$ and $Z$:

$$Y \in \{0, 1\}$$
$$Z \in \{0, 2\}.$$ 

By default $Y = 1$ and $Z = 2$. The encoder tells them when to move out of the way.

$$R_1 \geq H(Y) = H(1/3) = \log 3 - 2/3 \text{ bits},$$
$$R_2 \geq H(Z) = H(1/3) = \log 3 - 2/3 \text{ bits}.$$
Achievable rates for general joint distribution $p(x, y, z)$:

\[
\begin{align*}
R_1 & \geq I(X; Y, U), \\
R_2 & \geq I(X; Z, U), \\
R_1 + R_2 & \geq I(X; Y, U) + I(X; Z, U) + I(Y; Z|X, U).
\end{align*}
\]

for some $U$.

[Cover, Permuter 07], [Zhang, Berger 95]
Idea 4—Restrict the support of $Y$ and $Z$ around $\hat{X}$:

First send $\hat{X}$ to both $Y$ and $Z$, where $\hat{X} = X$ with probability $\frac{\alpha}{\alpha + 2}$.

$$p(\hat{x}|x = 1) = \begin{cases} \frac{1}{\alpha + 2} & \text{if } x = 0 \\ \frac{\alpha}{\alpha + 2} & \text{if } x = 1 \\ \frac{1}{\alpha + 2} & \text{if } x = 2 \end{cases}$$
Task Assignment: Achievable Rates

Idea 4—Restrict the support of $Y$ and $Z$ around $\hat{X}$:

$$
Y \in \{\hat{X}, \hat{X} + 1\}
$$

$$
Z \in \{\hat{X}, \hat{X} + 2\}.
$$

The encoder tells them when to move out of the way (onto $\hat{X}$).

$$
R_1 \geq I(X; \hat{X}) + H(Y|\hat{X}),
$$

$$
R_2 \geq I(X; \hat{X}) + H(Z|\hat{X}).
$$
Idea 4—Restrict the support of $Y$ and $Z$ around $\hat{X}$:

Optimize $\alpha$ (compression rate of $\hat{X}$):

$$\alpha^* = \phi = \frac{\sqrt{5} + 1}{2}.$$ 

Resulting rates:

$$R_1 = R_2 = \log 3 - \log \phi.$$
Summary

1. Blackwell’s Trapdoor Channel
   - $C_{FB} = \log \phi$.
   - Transform to equivalent channel
   - Simple zero-error communication scheme
   - Guess and check solution to Bellman equation

2. Three-node task assignment
   - Achievable symmetric rate: $H(X) - \log \phi$
   - Two phases of communication