What Hockey Teams and Foraging Animals Can Teach Us About Feedback Communication

Part I: A Tutorial on Feedback

Aaron Wagner
Cornell University
The “Coat of Arms”

The Information Theorist’s Coat of Arms

Shannon’s canonical block diagram of the one-way communication system. (Reproduced with permission from “A Mathematical Theory of Communication,” C. E. Shannon, Bell System Technical Journal, October 1948.)

[source: Key Papers in the Development of Information Theory]
Communication *Without* Feedback is the Exception
Communication *Without* Feedback is the Exception
Communication *Without* Feedback is the Exception
A Doctored Coat of Arms

The Information Theorist’s Coat of Arms


[source: *Key Papers in the Development of Information Theory*]
A Doctored Coat of Arms

The Information Theorist’s Coat of Arms

Shannon’s canonical block diagram of the one-way communication system. (Reproduced with permission from “A Mathematical Theory of Communication,” C. E. Shannon, Bell System Technical Journal, October 1948.)

[source: Key Papers in the Development of Information Theory]

[We only consider ideal feedback in this talk]
Mechanisms
Mechanisms

‣ How can one use feedback to improve block coding performance in point-to-point channels?
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
- [Other contexts: networks, control over noisy channels, streaming codes, complexity-constrained coding....]
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.

- [Other contexts: networks, control over noisy channels, streaming codes, complexity-constrained coding....]
Given:

- input alphabet: $\mathcal{X}$ (finite)
- output alphabet: $\mathcal{Y}$ (finite)
- channel matrix: $W(y|x)$ (indep. over time)
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)
Discrete Memoryless Channels without Feedback

Given:
- input alphabet: $\mathcal{X}$ (finite)
- output alphabet: $\mathcal{Y}$ (finite)
- channel matrix: $W(y|x)$ (indep. over time)

Diagram:
- Encoder
- Decoder
- $U_1, \ldots, U_k$ (finite set, $U_i \in \{0, 1\}$)
- $W(\cdot|\cdot)$

Given:
- input alphabet: $\mathcal{X}$ (finite)
- output alphabet: $\mathcal{Y}$ (finite)
- channel matrix: $W(y|x)$ (indep. over time)
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)

![Diagram of a discrete memoryless channel with feedback](image)

- $U_1, \ldots, U_k$ (input sequence)
- $\hat{U}_1, \ldots, \hat{U}_k$ (feedback sequence)
- $X_1, \ldots, X_n$ (encoded symbols)
- $Y_1, \ldots, Y_n$ (received symbols)
- $W(\cdot|\cdot)$ (channel matrix)
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: \( \mathcal{X} \) (finite)
  - output alphabet: \( \mathcal{Y} \) (finite)
  - channel matrix: \( W(y|x) \) (indep. over time)
Discrete Memoryless Channels without Feedback

- **Given:**
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)
Discrete Memoryless Channels without Feedback

Given:
- input alphabet: $\mathcal{X}$ (finite)
- output alphabet: $\mathcal{Y}$ (finite)
- channel matrix: $W(y|x)$ (indep. over time)
Discrete Memoryless Channels without Feedback

Given:
- input alphabet: $\mathcal{X}$ (finite)
- output alphabet: $\mathcal{Y}$ (finite)
- channel matrix: $W(y|x)$ (indep. over time)
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: \( \mathcal{X} \) (finite)
  - output alphabet: \( \mathcal{Y} \) (finite)
  - channel matrix: \( W(y|x) \) (indep. over time)
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: \( \mathcal{X} \) (finite)
  - output alphabet: \( \mathcal{Y} \) (finite)
  - channel matrix: \( W(y|x) \) (indep. over time)
Discrete Memoryless Channels without Feedback

- Given:
  - input alphabet: $\mathcal{X}$ (finite)
  - output alphabet: $\mathcal{Y}$ (finite)
  - channel matrix: $W(y|x)$ (indep. over time)
Given:
- input alphabet: $\mathcal{X}$ (finite)
- output alphabet: $\mathcal{Y}$ (finite)
- channel matrix: $W(y|x)$ (indep. over time)
Discrete Memoryless Channels without Feedback

Given:
- input alphabet: $\mathcal{X}$ (finite)
- output alphabet: $\mathcal{Y}$ (finite)
- channel matrix: $W(y|x)$ (indep. over time)
Figures of Merit

\[ U_1, \ldots, U_k \rightarrow \text{Encoder} \rightarrow X_1, \ldots, X_n \]

\[ \hat{U}_1, \ldots, \hat{U}_k \leftarrow \text{Decoder} \leftarrow Y_1, \ldots, Y_n \]

\[ \text{W}(\cdot|\cdot) \]
Figures of Merit

- Number of bits sent: \( k \)
Figures of Merit

- Number of bits sent: $k$
- Transmission time: $n$
Figures of Merit

- Number of bits sent: $k$
- Transmission time: $n$
- Rate $R = k/n$
Figures of Merit

- Number of bits sent: $k$
- Transmission time: $n$
- Rate $R = k/n$
- Error probability: $P_e = P(U^k \neq \hat{U}^k)$
Asymptotic Metrics
Asymptotic Metrics

- Capacity
Asymptotic Metrics

- Capacity
- Error exponents
Asymptotic Metrics

- Capacity
- Error exponents
- Second-order coding rate (normal approximation)
Asymptotic Metrics

- Capacity
- Error exponents
- Second-order coding rate (normal approximation)
- Moderate deviations performance
I.I.D. Sums
I.I.D. Sums

- Let \( \{X_i\} \) be an i.i.d. sequence with zero mean, unit variance.
I.I.D. Sums

- Let \( \{X_i\} \) be an i.i.d. sequence with zero mean, unit variance
- (Weak) Law of large numbers:

\[
\lim_{n \to \infty} \Pr \left( \sum_{i=1}^{n} X_i > \epsilon n \right) = 0 \quad \epsilon > 0
\]
I.I.D. Sums

- Let \( \{X_i\} \) be an i.i.d. sequence with zero mean, unit variance
- (Weak) Law of large numbers:
  \[
  \lim_{n \to \infty} \Pr \left( \sum_{i=1}^{n} X_i > \epsilon n \right) = 0 \quad \epsilon > 0
  \]
- Large deviations*:
  \[
  \lim_{n \to \infty} \frac{1}{n} \log \Pr \left( \sum_{i=1}^{n} X_i > \epsilon n \right) = \Lambda^*(\epsilon) > 0 \quad \epsilon > 0
  \]
I.I.D. Sums

- Let \( \{X_i\} \) be an i.i.d. sequence with zero mean, unit variance.

- (Weak) Law of large numbers:

  \[
  \lim_{n \to \infty} \mathbb{P}(\sum_{i=1}^{n} X_i > \epsilon n) = 0 \quad \epsilon > 0
  \]

- Large deviations*:

  \[
  \lim_{n \to \infty} -\frac{1}{n} \log \mathbb{P}(\sum_{i=1}^{n} X_i > \epsilon n) = \Lambda^*(\epsilon) > 0 \quad \epsilon > 0
  \]

- Central Limit Theorem (CLT):

  \[
  \lim_{n \to \infty} \mathbb{P}(\sum_{i=1}^{n} X_i > \epsilon \sqrt{n}) = Q(\epsilon)
  \]
I.I.D. Sums

- Moderate deviations*: if $\beta$ is in $(1/2, 1)$:

$$\lim_{n \to \infty} - \frac{1}{n^{2\beta-1}} \log \Pr \left( \sum_{i=1}^{n} X_i > \epsilon n^\beta \right) = \Lambda_N^*(\epsilon) \quad \epsilon > 0$$
Lemma (Shannon ’57): For a DMC without feedback, for any input dist. $P$ and any $\theta > 0$, there exists a code with rate $R$, block length $n$, and error prob.

\[
P_e \leq \Pr \left( \sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta \right) + 2^{-n\theta}
\]
Lemma (Shannon ’57): For a DMC without feedback, for any input dist. $P$ and any $\theta > 0$, there exists a code with rate $R$, block length $n$, and error prob.

$$P_e \leq \Pr \left( \sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta \right) + 2^{-n\theta}$$

i.i.d. $P \circ W$
Lemma (Shannon ’57): For a DMC without feedback, for any input dist. \( P \) and any \( \theta > 0 \), there exists a code with rate \( R \), block length \( n \), and error prob.

\[
P_e \leq \Pr \left( \sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta \right) + 2^{-n\theta}
\]

“information density”
Connection to Channel Coding

- **Lemma (Shannon ’57):** For a DMC without feedback, for any input dist. $P$ and any $\theta > 0$, there exists a code with rate $R$, block length $n$, and error prob.

\[ P_e \leq \Pr \left( \sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta \right) + 2^{-n\theta} \]

“information density”

- For the information density,
Lemma (Shannon ’57); For a DMC without feedback, for any input dist. $P$ and any $\theta > 0$, there exists a code with rate $R$, block length $n$, and error prob.

$$P_e \leq \Pr \left( \sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta \right) + 2^{-n\theta}$$

“information density”

For the information density,
- Law of large numbers $\rightarrow$ capacity
Lemma (Shannon ’57): For a DMC without feedback, for any input dist. \( P \) and any \( \theta > 0 \), there exists a code with rate \( R \), block length \( n \), and error prob.

\[
P_e \leq \Pr \left( \sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta \right) + 2^{-n\theta}
\]

“information density”

For the information density,
- Law of large numbers \( \rightarrow \) capacity
- Large deviations \( \rightarrow \) error exponents
Connection to Channel Coding

- **Lemma** (Shannon ’57): For a DMC without feedback, for any input dist. \( P \) and any \( \theta > 0 \), there exists a code with rate \( R \), block length \( n \), and error prob.

\[
P_e \leq \Pr \left( \sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta \right) + 2^{-n\theta}
\]

“information density”

- For the information density,
  - Law of large numbers \( \rightarrow \) capacity
  - Large deviations \( \rightarrow \) error exponents
  - Central limit theorem \( \rightarrow \) second-order coding rate
Lemma (Shannon ’57); For a DMC without feedback, for any input dist. $P$ and any $\theta > 0$, there exists a code with rate $R$, block length $n$, and error prob.

$$P_e \leq \Pr \left( \sum_{i=1}^{n} \log \frac{W(Y_i|X_i)}{PW(Y_i)} \leq nR + n\theta \right) + 2^{-n\theta}$$

"information density"

For the information density,

- Law of large numbers $\rightarrow$ capacity
- Large deviations $\rightarrow$ error exponents
- Central limit theorem $\rightarrow$ second-order coding rate
- Moderate deviations $\rightarrow$ moderate deviations
Def:

\[ P_e(n, R) = \min \{ P_e : \exists \text{ an } (n, k, P_e) \text{ code with } k/n \geq R \} \]
Def: The reliability function or error exponent at rate $R$ is

$$E(R) = \lim_{n \to \infty} - \frac{1}{n} \log P_e(n, R)$$
**Error Exponents**

- **Def:** The *reliability function* or *error exponent* at rate $R$ is
  \[ E(R) = \lim_{n \to \infty} -\frac{1}{n} \log P_e(n, R) \]

- Characterized w/o feedback for a range of rates close to capacity and at very low rates [Shannon, Gallager, Berlekamp ('67)].
Second-Order Coding Rate

- **Def:**

\[ R(n, \epsilon) = \max \left\{ \frac{k}{n} : \exists \text{ an } (n, k, P_e) \text{ code with } P_e \leq \epsilon \right\} \]
Def:

\[ R(n, \epsilon) = \max \left\{ \frac{k}{n} : \exists \text{ an } (n, k, P_e) \text{ code with } P_e \leq \epsilon \right\} \]

\[ \text{Think: } R(n, \epsilon) \approx C + \frac{\beta(\epsilon)}{\sqrt{n}} + \cdots \]
Def: Second-Order Coding Rate (SOCR):

\[ R(n, \epsilon) = \max \left\{ \frac{k}{n} : \exists \text{ an } (n, k, P_e) \text{ code with } P_e \leq \epsilon \right\} \]

Think: \( R(n, \epsilon) \approx C + \frac{\beta(\epsilon)}{\sqrt{n}} + \cdots \)

Def: Second-Order Coding Rate (SOCR):

\[ \beta(\epsilon) = \lim_{n \to \infty} \frac{(R(n, \epsilon) - C)\sqrt{n}}{\sqrt{n}} \]
Second-Order Coding Rate

▶ Def:

\[ R(n, \varepsilon) = \max \left\{ \frac{k}{n} : \exists \text{ an } (n, k, P_e) \text{ code with } P_e \leq \varepsilon \right\} \]

\[
\left[ \text{Think: } R(n, \varepsilon) \approx C + \frac{\beta(\varepsilon)}{\sqrt{n}} + \cdots \right]
\]

▶ Def: Second-Order Coding Rate (SOCR):

\[ \beta(\varepsilon) = \lim_{n \to \infty} (R(n, \varepsilon) - C) \sqrt{n} \]

- Characterized w/o feedback by Strassen (’62).
**Theorem** (Altuğ-Wagner ’14):

Consider a DMC without feedback. Let \( R_n = C - \epsilon_n \) be s.t.

\[
\lim_{n \to \infty} \epsilon_n = 0 \quad \lim_{n \to \infty} \epsilon_n \sqrt{n} = \infty
\]

Then

\[
\lim_{n \to \infty} \frac{-\log P_e(n, R_n)}{\epsilon_n^2 \cdot n} = \frac{1}{2V_{\min}}
\]
Theorem (Altuğ-Wagner ’14):

Consider a DMC without feedback. Let \( R_n = C - \epsilon_n \) be s.t.

\[
\lim_{n \to \infty} \epsilon_n = 0 \quad \quad \lim_{n \to \infty} \epsilon_n \sqrt{n} = \infty
\]

Then

\[
\lim_{n \to \infty} \frac{- \log P_e(n, R_n)}{\epsilon_n^2 \cdot n} = \frac{1}{2V_{\text{min}}}
\]

constant depending on the channel
Moderate Deviations

\[ P_e(n, R) \]

1

\[ O(1) \]

\[ 2^{-nE(R)} \]

Normal approximation

Moderate deviations

Error exponents
A Non-Example

\( U^k \in \{0, 1\}^k \)

Encoder

\( X^n \in \{0, 1\}^n \)

Decoder

\( \hat{U}^n \in \{0, 1\}^k \)

\( y^n \in \{0, 1, e\}^n \)
A Non-Example

$U^k \in \{0, 1\}^k$

Encoder

$X^n \in \{0, 1\}^n$

Decoder

$\hat{U}^n \in \{0, 1\}^k$

$y^n \in \{0, 1, e\}^n$
A Non-Example

- Scheme: repeatedly transmit each bit until it gets through
A Non-Example

- Scheme: repeatedly transmit each bit until it gets through

\[ P_e = \sum_{\ell=0}^{k-1} P\left( \sum_{i=1}^{n} Z_i = \ell \right) \cdot \left( 1 - \frac{1}{2^{k-\ell}} \right) \leq P\left( \sum_{i=1}^{n} Z_i < k \right) \]
A Non-Example

- Scheme: repeatedly transmit each bit until it gets through

\[ P_e = \sum_{\ell=0}^{k-1} P \left( \sum_{i=1}^{n} Z_i = \ell \right) \cdot \left( 1 - \frac{1}{2^{k-\ell}} \right) \]

\[ \leq P \left( \sum_{i=1}^{n} Z_i < k \right) \]
A Non-Example

Scheme: repeatedly transmit each bit until it gets through

\[ P_e = \sum_{\ell=0}^{k-1} P\left( \sum_{i=1}^{n} Z_i = \ell \right) \cdot \left( 1 - \frac{1}{2^{k-\ell}} \right) \]

\[ \leq P\left( \sum_{i=1}^{n} Z_i < k \right) \]
A Non-Example

Scheme: repeatedly transmit each bit until it gets through

\[ P_e = \sum_{\ell=0}^{k-1} P\left( \sum_{i=1}^{n} Z_i = \ell \right) \cdot \left( 1 - \frac{1}{2^{k-\ell}} \right) \]

\[ \leq P\left( \sum_{i=1}^{n} Z_i < k \right) \to 0 \]

if \( n, k \to \infty \) as \( k = nR \) with \( R < 1 - p \)
A Non-Example

$U^k \in \{0, 1\}^k$

Encoder

$X^n \in \{0, 1\}^n$

Decoder

$Y^n \in \{0, 1, e\}^n$

$\hat{U}^n \in \{0, 1\}^k$
A Non-Example

\[ U^k \in \{0, 1\}^k \]

\[ X^n \in \{0, 1\}^n \]

\[ Y^n \in \{0, 1, e\}^n \]

\[ \hat{U}^n \in \{0, 1\}^k \]

\[
\begin{bmatrix}
0 \\
1 \\
1 \\
0 \\
\vdots \\
1
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 \\
1 & 1 & \ldots & 1 \\
1 & 0 & \ldots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 1 & \ldots & 1
\end{bmatrix}
\cdot 
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]
A Non-Example

\( U^k \in \{0, 1\}^k \quad X^n \in \{0, 1\}^n \quad Y^n \in \{0, 1, e\}^n \quad \hat{U}^n \in \{0, 1\}^k \)

\[
\begin{bmatrix}
0 \\
1 \\
1 \\
0 \\
\vdots \\
1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
1 & 1 & \cdots & 1 \\
1 & 0 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 1 & \cdots & 1
\end{bmatrix} \cdot \begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

\[ G \in \{0, 1\}^{n \times k} \text{ [uniform]} \]
A Non-Example

Encoder

Decoder

\[ U^k \in \{0, 1\}^k \]

\[ X^n \in \{0, 1\}^n \]

\[ Y^n \in \{0, 1, e\}^n \]

\[ \hat{U}^n \in \{0, 1\}^k \]

\[
\begin{bmatrix}
0 \\
1 \\
1 \\
0 \\
\vdots \\
1
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
1 & 1 & \cdots & 1 \\
1 & 0 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 1 & \cdots & 1
\end{bmatrix}
\cdot
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]
A Non-Example

\[ U^k \in \{0, 1\}^k \]

Encoder

\[ X^n \in \{0, 1\}^n \]

Decoder

\[ \hat{U}^n \in \{0, 1\}^k \]

\[
\begin{bmatrix}
0 \\
1 \\
1 \\
0 \\
\vdots \\
1
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
1 & 1 & \cdots & 1 \\
1 & 0 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 1 & \cdots & 1
\end{bmatrix}
\cdot
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

\[ U^k \in \{0, 1\}^k \]
A Non-Example

\[ U^k \in \{0, 1\}^k \]

\[ X^n \in \{0, 1\}^n \]

\[ y^n \in \{0, 1, e\}^n \]

\[ \hat{U}^n \in \{0, 1\}^k \]

\[
\begin{bmatrix}
0 \\
1 \\
1 \\
0 \\
\vdots \\
1
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
1 & 1 & \cdots & 1 \\
1 & 0 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 1 & \cdots & 1
\end{bmatrix}
\cdot
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]
A Non-Example

\[ U^k \in \{0, 1\}^k \]

\[ X^n \in \{0, 1\}^n \]

\[ Y^n \in \{0, 1, e\}^n \]

\[ \hat{U}^n \in \{0, 1\}^k \]

\[
\begin{bmatrix}
0 \\
1 \\
1 \\
0
\end{bmatrix} = 
\begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 0 & \cdots & 0 \\
1 & 1 & \cdots & 1 \\
1 & 0 & \cdots & 1
\end{bmatrix} \cdot 
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]
A Non-Example

$U^k \in \{0, 1\}^k$

Encoder

$X^n \in \{0, 1\}^n$

Decoder

$\hat{U}^n \in \{0, 1\}^k$

$y^n \in \{0, 1, e\}^n$

$Y^n \in \{0, 1, e\}^n$

\[
\begin{bmatrix}
0 \\
1 \\
1 \\
0 \\
\vdots \\
1
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 \\
1 & 1 & \ldots & 1 \\
1 & 0 & \ldots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 1 & \ldots & 1
\end{bmatrix}
\cdot
\begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]
A Non-Example

\[ U^k \in \{0, 1\}^k \]

\[ X^n \in \{0, 1\}^n \]

\[ Y^n \in \{0, 1, e\}^n \]

\[ P_e \leq \sum_{\ell=0}^{n} \left[ P \left( \sum_{i=1}^{n} Z_i = \ell \right) \cdot P(\ell \times k \text{ sub-matrix of } G \text{ not full column-rank}) \right] \]

\[ \leq \sum_{\ell=0}^{n} \left[ P \left( \sum_{i=1}^{n} Z_i = \ell \right) \cdot \max(2^{k-\ell}, 1) \right] \]
A Non-Example

$U^n \in \{0, 1\}^k$

$X^n \in \{0, 1\}^n$

$y^n \in \{0, 1, e\}^n$

$P_e \leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot P(\ell \times k \text{ sub-matrix of } G \text{ not full column-rank})$

$\leq \sum_{\ell=0}^{n} P\left(\sum_{i=1}^{n} Z_i = \ell\right) \cdot \max\left(2^{k-\ell}, 1\right) \to 0$
A Non-Example

\[ P_e \leq \sum_{\ell=0}^{n} P\left( \sum_{i=1}^{n} Z_i = \ell \right) \cdot P(\ell \times k \text{ sub-matrix of } G \text{ not full column-rank}) \]

\[ \leq \sum_{\ell=0}^{n} P\left( \sum_{i=1}^{n} Z_i = \ell \right) \cdot \max(2^{k-\ell}, 1) \to 0 \]

if \( n, k \to \infty \) as \( k = nR \) with \( R < 1 - p \)
A Non-Example

\[ U^k \in \{0, 1\}^k \]

Encoder

\[ X^n \in \{0, 1\}^n \]

Decoder

\[ \hat{U}^n \in \{0, 1\}^k \]

\[ P_e \leq \sum_{\ell=0}^n P \left( \sum_{i=1}^n Z_i = \ell \right) \cdot P(\ell \times k \text{ sub-matrix of } G \text{ not full column-rank}) \]

\[ \leq \sum_{\ell=0}^n P \left( \sum_{i=1}^n Z_i = \ell \right) \cdot \max(2^{k-\ell}, 1) \rightarrow 0 \]

if \( n, k \to \infty \) as \( k = nR \) with \( R < 1 - p \)

Also no improvement in (high rate) error exponents, SOCR, or moderate deviations.
More Generally...

- **Def:** A channel (stochastic matrix) $W$ is *symmetric* if its columns (outputs) can be partitioned so that, within each partition, the columns are permutations of each other, as are the rows.
More Generally...

Def: A channel (stochastic matrix) $W$ is *symmetric* if its columns (outputs) can be partitioned so that, within each partition, the columns are permutations of each other, as are the rows.

$$\begin{bmatrix} 1-p & 0 & p \\ 0 & 1-p & p \end{bmatrix}$$

Symmetric
More Generally...

- **Def:** A channel (stochastic matrix) $W$ is *symmetric* if its columns (outputs) can be partitioned so that, within each partition, the columns are permutations of each other, as are the rows.

\[
\begin{bmatrix}
1 - p & 0 & p \\
0 & 1 - p & p \\
\end{bmatrix}
\]

Symmetric

\[
\begin{bmatrix}
3/4 & 1/4 \\
1/3 & 2/3 \\
\end{bmatrix}
\]

Not symmetric
More Generally…
More Generally...

- For symmetric channels, feedback does not improve:
More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
More Generally…

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the third-order coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the third-order coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
  - the moderate deviations performance [Altuğ-Poor-Verdú (’15)]
More Generally...

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the third-order coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
  - the moderate deviations performance [Altuğ-Poor-Verdú (’15)]

- For asymmetric channels,
More Generally…

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the third-order coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
  - the moderate deviations performance [Altuğ-Poor-Verdú (’15)]

- For asymmetric channels,
  - The high-rate error exponent is not improved by feedback [Nakiboğlu ’19, Augustin ’78]
More Generally…

- For symmetric channels, feedback does not improve:
  - the error exponent (for large rates) [Haroutunian ‘77; Dobrushin ‘62]
  - the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ‘21]
  - the second-order coding rate [Polyanskiy et al. ’11]
  - the third-order coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
  - the moderate deviations performance [Altuğ-Poor-Verdú (’15)]

- For asymmetric channels,
  - The high-rate error exponent is not improved by feedback [Nakiboğlu ’19, Augustin ’78]
  - The second-order coding rate can be improved by feedback [Part II]
More Generally...

For symmetric channels, feedback does not improve:
- the error exponent (for large rates) [Haroutunian ’77; Dobrushin ’62]
- the order of the polynomial pre-factor in the error exponent (for large rates) [Altuğ-Wagner ’21]
- the second-order coding rate [Polyanskiy et al. ’11]
- the third-order coding rate [Polyanskiy et al. ’11, Altuğ-Wagner ’21]
- the moderate deviations performance [Altuğ-Poor-Verdú (’15)]

For asymmetric channels,
- The high-rate error exponent is not improved by feedback [Nakiboğlu ’19, Augustin ’78]
- The second-order coding rate can be improved by feedback [Part II]
- Moderate deviations?
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
First attempt at an example: consider the binary symmetric channel (BSC):

\[ \mathcal{X} = \mathcal{Y} = \{0, 1\} \]

\[ Y^n = X^n \oplus Z^n \]

where \( \{Z_n\} \) is an arbitrary stationary and ergodic process.
First attempt at an example: consider the binary symmetric channel (BSC):

\[
X = Y = \{0, 1\}
\]

\[
Y^n = X^n \oplus Z^n
\]

where \{Z_n\} is an arbitrary stationary and ergodic process.

Then \( C = 1 - H(\{Z_n\}) \)
With feedback: \( I(U^k; Y^n) = H(Y^n) - H(Y^n|U^k) \)

\[
= H(Y^n) - \sum_{i=1}^{n} H(Y_i|U^k, Y^{i-1})
\]

\[
= H(Y^n) - \sum_{i=1}^{n} H(X_i \oplus Z_i|U^k, Y^{i-1})
\]

\[
= H(Y^n) - \sum_{i=1}^{n} H(Z_i|U^k, Y^{i-1}, Z^{i-1})
\]

\[
= H(Y^n) - \sum_{i=1}^{n} H(Z_i|Z^{i-1})
\]

\[
= H(Y^n) - H(Z^n)
\]

\[
\leq n - H(Z^n)
\]
Channels with Memory

With feedback:  
\[ I(U^k; Y^n) = H(Y^n) - H(Y^n|U^k) \]

\[ = H(Y^n) - \sum_{i=1}^{n} H(Y_i|U^k, Y^{i-1}) \]

\[ = H(Y^n) - \sum_{i=1}^{n} H(X_i \oplus Z_i|U^k, Y^{i-1}) \]

\[ = H(Y^n) - \sum_{i=1}^{n} H(Z_i|U^k, Y^{i-1}, Z^{i-1}) \]

\[ = H(Y^n) - \sum_{i=1}^{n} H(Z_i|Z^{i-1}) \]

\[ = H(Y^n) - H(Z^n) \]

\[ \leq n - H(Z^n) \]

\[ C_{FB} = 1 - H(\{Z_n\}) \]
With feedback: \( I(U^k; Y^n) = H(Y^n) - H(Y^n|U^k) \)

\[
= H(Y^n) - \sum_{i=1}^{n} H(Y_i|U^k, Y^{i-1}) \\
= H(Y^n) - \sum_{i=1}^{n} H(X_i \oplus Z_i|U^k, Y^{i-1}) \\
= H(Y^n) - \sum_{i=1}^{n} H(Z_i|U^k, Y^{i-1}, Z^{i-1}) \\
= H(Y^n) - \sum_{i=1}^{n} H(Z_i|Z^{i-1}) \\
= H(Y^n) - H(Z^n) \\
\leq n - H(Z^n) \\
C_{FB} = 1 - H(\{Z_n\})
\]

Feedback does not increase the capacity of discrete additive-noise channels [Alajaji ('95)]
Channels with Memory
Channels with Memory

- Consider a channel with ternary channel with three “states”
Channels with Memory

- Consider a channel with ternary channel with three “states”

- The channel starts in a random state and then deterministically cycles $1 \rightarrow 2$, $2 \rightarrow 3$, $3 \rightarrow 1$. 
Consider a channel with ternary channel with three “states”

- The channel starts in a random state and then deterministically cycles 1 → 2, 2 → 3, 3 → 1.
- Each constituent channel has $C = 1$ bit.
Consider a channel with ternary channel with three “states”

The channel starts in a random state and then deterministically cycles $1 \rightarrow 2$, $2 \rightarrow 3$, $3 \rightarrow 1$.

Each constituent channel has $C = 1$ bit.

With feedback, encoder can learn the phase: $C_{FB} = 1$ bit.
Channels with Memory

Consider a channel with ternary channel with three “states”

1. The channel starts in a random state and then deterministically cycles $1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 1$.
2. Each constituent channel has $C = 1$ bit.
3. With feedback, encoder can learn the phase: $C_{FB} = 1$ bit
4. Without feedback, encoder uses each input equally:
   
   $$C = H(B(1/3)) < 1 \text{ bit}$$
Channels with Memory
Channels with Memory

- ... are closely related to unknown channels.
Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?
Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

\[
I(U^k; Y^n) = h(Y^n) - h(Y^n | U^k)
= h(Y^n) - h(Z^n)
\leq \frac{1}{2} \log((2\pi e)^n |K_{Y^n}|) - \frac{1}{2} \log((2\pi e)^n |K_{Z^n}|)
\]
Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

\[
I(U^k; Y^n) = h(Y^n) - h(Y^n | U^k) \\
= h(Y^n) - h(Z^n) \\
\leq \frac{1}{2} \log((2\pi e)^n |K_{Y^n}|) - \frac{1}{2} \log((2\pi e)^n |K_{Z^n}|)
\]

achieved with Gaussian inputs
Channels with Memory

- are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

\[
I(U^k; Y^n) = h(Y^n) - h(Y^n|U^k) \\
= h(Y^n) - h(Z^n) \\
\leq \frac{1}{2} \log((2\pi e)^n |K_{Y^n}|) - \frac{1}{2} \log((2\pi e)^n |K_{Z^n}|)
\]

achieved with Gaussian inputs

independent of the input
Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

\[
I(U^k; Y^n) = h(Y^n) - h(Y^n|U^k) \\
= h(Y^n) - h(Z^n) \\
\leq \frac{1}{2} \log(1 + \frac{nK_{Y^n}}{2\pi e}) - \frac{1}{2} \log(1 + \frac{nK_{Z^n}}{2\pi e})
\]

- achieved with Gaussian inputs
- can be better whitened with feedback
- independent of the input
Channels with Memory

- ... are closely related to unknown channels.
- Why does feedback increase the capacity of Gaussian additive noise channels but not discrete ones?

\[
I(U^k; Y^n) = h(Y^n) - h(Y^n|U^k) \\
= h(Y^n) - h(Z^n) \\
\leq \frac{1}{2} \log((2\pi e)^n|K_{Y^n}|) - \frac{1}{2} \log((2\pi e)^n|K_{Z^n}|)
\]

- ARMA\((k)\) Gaussian feedback capacity found by Kim (’10)
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
Opportunistic Use of the Channel
Opportunistic Use of the Channel

- Up to now, # of channel uses has been fixed.
Opportunistic Use of the Channel

- Up to now, # of channel uses has been fixed.
- For some transmissions, we might wish we had more. For others, we could do with fewer.
Opportunistic Use of the Channel

- Up to now, # of channel uses has been fixed.
- For some transmissions, we might wish we had more. For others, we could do with fewer.
- Suppose the transmission ends at a random (stopping) time $N$. 
Opportunistic Use of the Channel

- Up to now, # of channel uses has been fixed.
- For some transmissions, we might wish we had more. For others, we could do with fewer.
- Suppose the transmission ends at a random (stopping) time $N$.
- Define the effective rate $k/E[N]$.
Opportunistic Use of the Channel
Opportunistic Use of the Channel

- Consider the BEC:

\[
U^k \in \{0, 1\}^k
\]

\[
X^n \in \{0, 1\}^n
\]

\[
y^n \in \{0, 1, e\}^n
\]

\[
\hat{U}^n \in \{0, 1\}^k
\]
Opportunistic Use of the Channel

- Consider the BEC:

- Suppose we transmit each bit until it passes through.
Opportunistic Use of the Channel

- Consider the BEC:

  Suppose we transmit each bit until it passes through.
  Let $N$ be the # channel uses required for all bits to pass through. Then $E[N] = k/(1 - p)$. So

  \[
  \frac{k}{E[N]} = 1 - p = C = C_{FB}
  \]
Consider the BEC:

Suppose we transmit each bit until it passes through.

Let \( N \) be the \# channel uses required for all bits to pass through. Then \( E[N] = k/(1-p) \). So

\[
\frac{k}{E[N]} = 1 - p = C = C_{FB}
\]

\( P_e = 0 \)
Opportunistic Use of the Channel

- Consider the BEC:

- Suppose we transmit each bit until it passes through.

- Let $N$ be the # channel uses required for all bits to pass through. Then $E[N] = k / (1 - p)$. So

$$\frac{k}{E[N]} = 1 - p = C = C_{FB}$$

$$P_e = 0$$

- A little opportunism goes a long way:

$$\lim_{n \to \infty} \Pr(N \geq (1 + \varepsilon)E[N]) = 0 \text{ for any } \varepsilon > 0.$$
Following Burnashev (’76), reflecting later refinements:
As a general scheme, following Burnashev (’76), reflecting later refinements:

- Send message using non-feedback code

0 \quad E[N]
A General Scheme

- Following Burnashev (’76), reflecting later refinements:

  ![Diagram showing the process of sending a message using non-feedback code and sending ACK/NACK](chart)

  - Send message using non-feedback code
  - Send ACK/NACK
A General Scheme

- Following Burnashev (’76), reflecting later refinements:

  0 \quad E[N]

  Send message using non-feedback code
  Send ACK/NACK
  Send message using non-feedback code
  Send ACK/NACK
A General Scheme

Following Burnashev (’76), reflecting later refinements:

- Send message using non-feedback code
- Send ACK/NACK
- Send message using non-feedback code
- Send ACK/NACK
- Send message using non-feedback code
- Send ACK/NACK

0 \quad E[N]
A General Scheme

- Following Burnashev ('76), reflecting later refinements:

- Error exponent determined by Burnashev ('76)
A General Scheme

- Following Burnashev (’76), reflecting later refinements:

- Error exponent determined by Burnashev (’76)
  - Typically beats non-feedback error exponent at all rates
A General Scheme

- Error exponent determined by Burnashev (‘76)
  - Typically beats non-feedback error exponent at all rates
- Feedback provides an order improvement in
A General Scheme

- Following Burnashev (’76), reflecting later refinements:
  - Error exponent determined by Burnashev (’76)
    - Typically beats non-feedback error exponent at all rates
  - Feedback provides an order improvement in
    - the moderate deviations regime [Truong and Tan (’19)]
A General Scheme

- Following Burnashev (’76), reflecting later refinements:
  - Error exponent determined by Burnashev (’76)
    - Typically beats non-feedback error exponent at all rates
  - Feedback provides an order improvement in
    - the moderate deviations regime [Truong and Tan (’19)]
    - the second-order coding rate regime [Polyanskiy et al. (’11)]
Mechanisms

‣ How can one use feedback to improve block coding performance in point-to-point channels?

‣ If the channel has memory, we can predict the future noise realization.

‣ If the channel is unknown, we can learn its law.

‣ If the decoding time is not fixed, we can decode early or late opportunistically.

‣ If there is an average cost (e.g., power) constraint, we can use resources opportunistically.

‣ If the rate is low, we can increase the effective minimum distance of the code.
Mechanisms

‣ How can one use feedback to improve block coding performance in point-to-point channels?
  ▸ If the channel has memory, we can predict the future noise realization.
  ▸ If the channel is unknown, we can learn its law.
  ▸ If the decoding time is not fixed, we can decode early or late opportunistically.
  ▸ If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  ▸ If the rate is low, we can increase the effective minimum distance of the code.
Opportunistic Use of Power

- Consider the AWGN

\[ Y^n = X^n + Z^n \]

- Power constraint:

\[ E \left[ \frac{1}{n} \sum_{i=1}^{n} X_i^2(u^k, Y^{i-1}) \right] \leq P \text{ for all messages } u^k \]

\[ Z^n \text{ i.i.d. } \mathcal{N}(0, 1) \]
Schalkwijk-Kailath (’66) Scheme
Schalkwijk-Kailath ('66) Scheme

- Partition $[-\sqrt{P}, \sqrt{P}]$ into $2^k$ equal-sized intervals.
Schalkwijk-Kailath (’66) Scheme

- Partition $[-\sqrt{P}, \sqrt{P}]$ into $2^k$ equal-sized intervals.
- Assign each message string to one of the intervals
Schalkwijk-Kailath ('66) Scheme

- Partition \([-\sqrt{P}, \sqrt{P}]\) into \(2^k\) equal-sized intervals.
- Assign each message string to one of the intervals.
- Let \(\theta(u^k)\) be the midpoint of the interval for string \(u^k\).
Schalkwijk-Kailath (’66) Scheme

- Partition $[-\sqrt{P}, \sqrt{P}]$ into $2^k$ equal-sized intervals.
- Assign each message string to one of the intervals
- Let $\theta(u^k)$ be the midpoint of the interval for string $u^k$
- Encoding:
Schalkwijk-Kailath (’66) Scheme

- Partition $[-\sqrt{P}, \sqrt{P}]$ into $2^k$ equal-sized intervals.
- Assign each message string to one of the intervals
- Let $\theta(u^k)$ be the midpoint of the interval for string $u^k$
- Encoding:
  - Time 1: $\theta(U^k)$
Schalkwijk-Kailath ('66) Scheme

- Partition $\left[-\sqrt{P}, \sqrt{P}\right]$ into $2^k$ equal-sized intervals.
- Assign each message string to one of the intervals.
- Let $\theta(u^k)$ be the midpoint of the interval for string $u^k$.
- Encoding:
  - Time 1: $\theta(U^k)$
  - Time $j$: Send $\gamma_j \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)$, where $\gamma_j$ is chosen so that

\[
E \left[ \gamma_j^2 \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)^2 \bigg| U^k \right] \leq P \quad \text{a.s.}
\]
Schalkwijk-Kailath (’66) Scheme

- Partition $[-\sqrt{P}, \sqrt{P}]$ into $2^k$ equal-sized intervals.
- Assign each message string to one of the intervals.
- Let $\theta(u^k)$ be the midpoint of the interval for string $u^k$.
- Encoding:
  - Time 1: $\theta(U^k)$
  - Time $j$: Send $\gamma_j \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)$, where $\gamma_j$ is chosen so that
    \[
    E \left[ \gamma_j^2 \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)^2 \bigg| U^k \right] \leq P \quad \text{a.s.}
    \]
- Decoding: output string whose interval contains $E[\theta(U^k)|Y^n]$. 

Schalkwijk-Kailath ('66) Scheme

- Partition $\left[ -\sqrt{P}, \sqrt{P} \right]$ into $2^k$ equal-sized intervals.
- Assign each message string to one of the intervals.
- Let $\theta(u^k)$ be the midpoint of the interval for string $u^k$.

**Encoding:**
- Time 1: $\theta(U^k)$
- Time $j$: Send $\gamma_j \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)$, where $\gamma_j$ is chosen so that

$$E \left[ \gamma_j^2 \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)^2 \right] \leq P \quad \text{a.s.}$$

**Decoding:** Output string whose interval contains $E[\theta(U^k)|Y^n]$.

**Performance:**
$$P_e \leq \sqrt{\frac{2}{\pi}} e^{-\frac{2n(C-R) \sqrt{P}}{2}}$$
Schalkwijk-Kailath (’66) Scheme

- Partition \([-\sqrt{P}, \sqrt{P}]\) into \(2^k\) equal-sized intervals.
- Assign each message string to one of the intervals.
- Let \(\theta(u^k)\) be the midpoint of the interval for string \(u^k\).
- Encoding:
  - Time 1: \(\theta(U^k)\)
  - Time \(j\): Send \(\gamma_j \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)\), where \(\gamma_j\) is chosen so that
    \[
    E \left[ \gamma_j^2 \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)^2 \bigg| U^k \right] \leq P \quad \text{a.s.}
    \]
- Decoding: output string whose interval contains \(E[\theta(U^k)|Y^n]\).
- Performance:
    \[
    P_e \leq \sqrt{\frac{2}{\pi}} e^{-\frac{2n(C-R)\sqrt{P}}{2}} \quad [!!]
    \]
Notes on the SK Scheme

- The Schalkwijk-Kailath scheme uses (a lot) more power when decoding errors are imminent:

\[
E \left[ \gamma_j^2 \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)^2 \bigg| U^k \right] \leq P \quad \text{a.s.}
\]
Notes on the SK Scheme

- The Schalkwijk-Kailath scheme uses (a lot) more power when decoding errors are imminent:

\[
E \left[ \gamma_j^2 \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)^2 | U^k \right] \leq P \quad \text{a.s.}
\]

- Performance is much degraded if the power constraint is imposed a.s. [Pinkser ('68), Shepp et al. ('69), Altuğ-Poor-Verdú ('15)]
The Schalkwijk-Kailath scheme uses (a lot) more power when decoding errors are imminent:

\[ E \left[ \gamma_j^2 \left( \theta(U^k) - E[\theta(U^k)|Y^{j-1}] \right)^2 | U^k \right] \leq P \quad \text{a.s.} \]

Performance is much degraded if the power constraint is imposed a.s. [Pinkser (’68), Shepp et al. (’69), Altuğ-Poor-Verdú (’15)]

Error exponent of fixed-length coding for DMCs with a cost constraint?
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.

- If the rate is low, we can increase the effective minimum distance of the code.
Min. Distance Example
Consider the binary symmetric channel, w/o feedback,

\[ Y^n = X^n \oplus Z^n \quad Z^n \text{ i.i.d. } B(p) \]

and at low rate, \( k = \epsilon n, \epsilon \approx 0. \) Then \( P_e \) is exp. small.
Consider the binary symmetric channel, w/o feedback,

\[ Y^n = X^n \oplus Z^n \quad Z^n \text{ i.i.d. } B(p) \]

and at low rate, \( k = \epsilon n, \epsilon \approx 0. \) Then \( P_e \) is exp. small.

Suppose the codewords are

\[ x_1^n, x_2^n, \ldots, x_{2^k}^n \]
Consider the binary symmetric channel, w/o feedback, 

\[ Y^n = X^n \oplus Z^n \quad Z^n \text{ i.i.d. } B(p) \]

and at low rate, \( k = \epsilon n, \epsilon \approx 0 \). Then \( P_e \) is exp. small.

Suppose the codewords are 

\[ x_1^n, x_2^n, \ldots, x_{2^k}^n \]

ML decoding rule

\[ \text{argmin}_i \ d_H(x_i^n, Y^n) \]
Consider the binary symmetric channel, w/o feedback, and at low rate, $k = \epsilon n$, $\epsilon \approx 0$. Then $P_e$ is exp. small.

Suppose the codewords are

$$Y^n = X^n \oplus Z^n \quad Z^n \text{ i.i.d. } B(p)$$

and at low rate, $k = \epsilon n$, $\epsilon \approx 0$. Then $P_e$ is exp. small.

Suppose the codewords are

$$x_1^n, x_2^n, \ldots, x_{2^k}^n$$

ML decoding rule

$$\arg\min_i d_H(x_i^n, Y^n)$$

Hamming distance
Min. Distance Example

\[ P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr(\text{error} \mid x_m^n) \]
Min. Distance Example

\[ P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr(\text{error} \mid x_m^n) \]

\[ \leq 2^{-k} \sum_{m=1}^{2^k} \sum_{l=1, l \neq m}^{2^k} \Pr \left( d_H(x_l^n, Y^n) \leq d_H(x_m^n, Y^n) \mid x_m^n \right) \]
Min. Distance Example

\[
P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr \left( \text{error} \mid x^n_m \right)
\]

\[
\leq 2^{-k} \sum_{m=1}^{2^k} \sum_{l=1, l \neq m}^{2^k} \Pr \left( d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m \right)
\]

\[
\leq 2^k \cdot \max_{l \neq m} \Pr \left( d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m \right)
\]
Min. Distance Example

\[ P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr(\text{error} \mid x^n_m) \]

\[ \leq 2^{-k} \sum_{m=1}^{2^k} \sum_{l=1}^{2^k} \Pr(d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m) \]

\[ \leq 2^k \cdot \max_{l \neq m} \Pr(d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m) \]

\[ P_e \geq 2^{-k} \cdot \max_{l \neq m} \Pr(d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m) \]
Min. Distance Example

\[ P_e = 2^{-k} \sum_{m=1}^{2^k} \Pr(\text{error} \mid x^n_m) \]

\[ \leq 2^{-k} \sum_{m=1}^{2^k} \sum_{l=1, l \neq m}^{2^k} \Pr( d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m) \]

\[ \leq 2^k \cdot \max_{l \neq m} \Pr( d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m) \]

\[ P_e \geq 2^{-k} \cdot \max_{l \neq m} \Pr( d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m) \]

\[ \frac{1}{n} \log P_e \approx \max_{l \neq m} \frac{1}{n} \log \Pr( d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m) \]
Min. Distance Example

\[
\frac{1}{n} \log P_e \approx \max_{l \neq m} \frac{1}{n} \log \Pr \left( d_H(x_l^n, Y^n) \leq d_H(x_m^n, Y^n) \middle| x_m^n \right)
\]
Min. Distance Example

\[
\frac{1}{n} \log P_e \approx \max_{l \neq m} \frac{1}{n} \log \Pr \left( d_H(x_l^n, Y^n) \leq d_H(x_m^n, Y^n) \mid x_m^n \right)
\]

\[
\approx \exp \left( -d_H(x_l^n, x_m^n) \cdot D \left( \frac{1}{2} \mid \mid p \right) \right)
\]
### Min. Distance Example

\[
\frac{1}{n} \log P_e \approx \max_{l \neq m} \frac{1}{n} \log \Pr \left( d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \bigg| x^n_m \right)
\]

\[
\approx \exp \left( -d_H(x^n_l, x^n_m) \cdot D \left( \frac{1}{2} \left\| p \right\| \right) \right)
\]

So

\[
- \frac{1}{n} \log P_e \approx \min_{l \neq m} \frac{d_H(x^n_l, x^n_m)}{n} \cdot D \left( \frac{1}{2} \left\| p \right\| \right)
\]
Min. Distance Example

\[ \frac{1}{n} \log P_e \approx \max_{l \neq m} \frac{1}{n} \log \Pr \left( d_H(x^n_l, Y^n) \leq d_H(x^n_m, Y^n) \mid x^n_m \right) \]

\[ \approx \exp \left( -d_H(x^n_l, x^n_m) \cdot D \left( \frac{1}{2} \| p \right) \right) \]

So

\[ \frac{1}{n} \log P_e \approx \min_{l \neq m} \frac{d_H(x^n_l, x^n_m)}{n} \cdot D \left( \frac{1}{2} \| p \right) \]

min. distance of the code
\[
\frac{1}{n} \log P_e \approx \max_{l \neq m} \frac{1}{n} \log \Pr \left( d_H(x_l^n, Y^n) \leq d_H(x_m^n, Y^n) \mid x_m^n \right)
\]

\[
\approx \exp \left( -d_H(x_l^n, x_m^n) \cdot D \left( \frac{1}{2} \parallel p \right) \right)
\]

So

\[
- \frac{1}{n} \log P_e \approx \min_{l \neq m} \frac{d_H(x_l^n, x_m^n)}{n} \cdot D \left( \frac{1}{2} \parallel p \right)
\]

min. distance of the code

Q: How large can the minimum distance be?
Min. Distance Example
Min. Distance Example

- How large can the minimum distance be?
How large can the minimum distance be?
- If \( k = 1 \), min. distance is \( n \).
How large can the minimum distance be?
- If $k = 1$, min. distance is $n$.
  - 000000000000 vs. 111111111111
How large can the minimum distance be?
- If $k = 1$, min. distance is $n$.
  - 000000000000 vs. 111111111111
- If $k/n = \epsilon$, where $\epsilon$ is small, then min. distance $\approx n/2$
How large can the minimum distance be?
- If $k = 1$, min. distance is $n$.
  - 000000000000 vs. 111111111111
- If $k/n = \epsilon$, where $\epsilon$ is small, then min. distance $\approx n/2$

Suppose near the end of transmission, a genie ruled out all but one of the incorrect codewords.
Min. Distance Example

- How large can the minimum distance be?
  - If $k = 1$, min. distance is $n$.
    - 000000000000 vs. 111111111111
  - If $k/n = \epsilon$, where $\epsilon$ is small, then min. distance $\approx n/2$

- Suppose near the end of transmission, a genie ruled out all but one of the incorrect codewords.
  - Remaining transmission can be 0000... vs. 1111.....
How large can the minimum distance be?

- If \( k = 1 \), min. distance is \( n \).
  - 000000000000 vs. 111111111111
- If \( k/n = \epsilon \), where \( \epsilon \) is small, then min. distance \( \approx n/2 \)

Suppose near the end of transmission, a genie ruled out all but one of the incorrect codewords.

- Remaining transmission can be 0000... vs. 1111.....
- Would yield an effective min. distance increase.
How large can the minimum distance be?
- If $k = 1$, min. distance is $n$.
  - $000000000000$ vs. $111111111111$
- If $k/n = \epsilon$, where $\epsilon$ is small, then min. distance $\approx n/2$

Suppose near the end of transmission, a genie ruled out all but one of the incorrect codewords.
- Remaining transmission can be $0000...$ vs. $1111.....$
- Would yield an *effective* min. distance increase.
- We can achieve a similar effect with feedback ....
Min. Distance Example
Min. Distance Example

- Following Zigangirov (’70),
Following Zigangirov (’70),
- At time $i$, compute posterior prob. of messages given $Y^{i-1}$. 
Following Zigangirov (’70),

- At time $i$, compute posterior prob. of messages given $Y_{i-1}$.
- Greedily partition messages into two groups to minimize the difference of their sum-probabilities:
Following Zigangirov (’70),

- At time $i$, compute posterior prob. of messages given $Y_{i-1}$.
- Greedily partition messages into two groups to minimize the difference of their sum-probabilities:

```
message: 1 2 3 4 5 6
```
Following Zigangirov (’70),

- At time $i$, compute posterior prob. of messages given $Y_{i-1}$.
- Greedily partition messages into two groups to minimize the difference of their sum-probabilities:
Following Zigangirov (’70),

- At time $i$, compute posterior prob. of messages given $Y_{i-1}$.
- Greedily partition messages into two groups to minimize the difference of their sum-probabilities:

![Diagram showing message partitioning with messages 1, 2, 3, 4 grouped together and messages 5, 6 separated.]
Following Zigangirov (’70),
- At time $i$, compute posterior prob. of messages given $Y_{i-1}$.
- Greedily partition messages into two groups to minimize the difference of their sum-probabilities:

![Message Distribution Chart]
Following Zigangirov (’70),

- At time $i$, compute posterior prob. of messages given $Y_{i-1}$.
- Greedily partition messages into two groups to minimize the difference of their sum-probabilities:
Following Zigangirov (’70),
- At time $i$, compute posterior prob. of messages given $Y_{i-1}$.
- Greedily partition messages into two groups to minimize the difference of their sum-probabilities:

![Bar chart example]

message: 1 2 3 4 5 6
Following Zigangirov (’70),

- At time $i$, compute posterior prob. of messages given $Y_{i-1}$.
- Greedily partition messages into two groups to minimize the difference of their sum-probabilities:

![Min. Distance Example](image)

message: 1 2 3 4 5 6

send ‘0’

send ‘1’
- Following Zigangirov (’70),
  - At time $i$, compute posterior prob. of messages given $Y_{i-1}$.
  - Greedily partition messages into two groups to minimize the difference of their sum-probabilities:
  - Improves low-rate error exponent over non-feedback case.
Following Zigangirov (’70),

- At time $i$, compute posterior prob. of messages given $Y^{i-1}$.
- Greedily partition messages into two groups to minimize the difference of their sum-probabilities:

- Symmetric channel: no high-rate error exponent, moderate deviations, or second-order coding rate improvement.
Mechanisms
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
Mechanisms

† How can one use feedback to improve block coding performance in point-to-point channels?
  † If the channel has memory, we can predict the future noise realization.
  † If the channel is unknown, we can learn its law.
  † If the decoding time is not fixed, we can decode early or late opportunistically.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
Mechanisms

- How can one use feedback to improve block coding performance in point-to-point channels?
  - If the channel has memory, we can predict the future noise realization.
  - If the channel is unknown, we can learn its law.
  - If the decoding time is not fixed, we can decode early or late opportunistically.
  - If there is an average cost (e.g., power) constraint, we can use resources opportunistically.
  - If the rate is low, we can increase the effective minimum distance of the code.
- [See Part II]