Outstanding Reading

- Read all chapters assigned from IT-I (EE514, Winter 2012).
- Read chapter 8 in the book.
- Read chapter 9 in the book.
- Read chapter 10 in the book.
Please do use our discussion board (https://catalyst.uw.edu/gopost/board/bilmes/27386/) for all questions, comments, so that all will benefit from them being answered.
Water Filling for Parallel Gaussian Channels

\[ \sum_i \left( \frac{1}{\nu^*} - N_i \right)^+ = P \]  

- Think of this as gradually increasing \(1/\nu^*\) until Equation 2 is satisfied.
Water Filling for Parallel Gaussian Channels

\[ \sum_i \left( \frac{1}{\nu^*} - N_i \right)^+ = P \] (2)

- We can think of this as filling up a bin with uniformly dispersed rain water and stopping the rain until we achieve equality in Eq. 2
Water Filling for Parallel Gaussian Channels

\[ \sum_i \left( \frac{1}{\nu^*} - N_i \right)^+ = P \]  

- We can think of this as filling up a bin with uniformly dispersed rain water and stopping the rain until we achieve equality in Eq. 2

- the red area indicated above is equal to \( P \).
So the final capacity is then

\[ C_n = \frac{1}{2} \sum_{j=1}^{k} \log(1 + P_i / N_i) \]  

(3)

\[ = \frac{1}{2} \sum_{j=1}^{k} \log \left( 1 + \frac{(1/\nu^* - N_i)^+}{N_i} \right) \text{ bits per parallel channel use} \]  

(4)
Capacity of Parallel Channels

- So the final capacity is then

\[
C_n = \frac{1}{2} \sum_{j=1}^{k} \log(1 + \frac{P_i}{N_i})
\]

\[
= \frac{1}{2} \sum_{j=1}^{k} \log \left( 1 + \frac{(1/\nu^* - N_i)^+}{N_i} \right) \text{ bits per parallel channel use}
\]

- In units of bits per transmission (bits per single channel transmission, take the average):

\[
C_n = \frac{1}{2n} \sum_{j=1}^{k} \log \left( 1 + \frac{(\nu^* - N_i)^+}{N_i} \right) \text{ bits per transmission}
\]
Colored Noise - KKT

Like before, we have KKT conditions to get:

\[ A_{ii} = (\nu - \lambda_i)^+ \]  \hspace{1cm} (6)

and

\[ \sum_i (\nu - \lambda_i)^+ = nP \]  \hspace{1cm} (7)

So this is water filling again, but on the eigenvalues of \( K \). I.e., we are filling a diagonalized version of the problem. Once we get \( A_{ii} \), we can "re-correlate" using \( KX = QAQ^\top \) to get final constraint on \( X \).
Like before, we have KKT conditions to get:

\[ A_{ii} = (\nu - \lambda_i)^+ \]  

(6)

and

\[ \sum_i (\nu - \lambda_i)^+ = nP \]  

(7)

And the total capacity becomes

\[ C_n = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \frac{(\nu - \lambda_i)^+}{\lambda_i}) \text{ bits per transmission} \]  

(8)
Like before, we have KKT conditions to get:

\[ A_{ii} = (\nu - \lambda_i)^+ \]  

and

\[ \sum_i (\nu - \lambda_i)^+ = nP \]  

And the total capacity becomes

\[ C_n = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \frac{(\nu - \lambda_i)^+}{\lambda_i}) \text{ bits per transmission} \]

So this is water filling again, but on the eigenvalues of \( K_Z \).
Like before, we have KKT conditions to get:

\[ A_{ii} = (\nu - \lambda_i)^+ \]  

and

\[ \sum_i (\nu - \lambda_i)^+ = nP \]  

And the total capacity becomes

\[
C_n = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \frac{(\nu - \lambda_i)^+}{\lambda_i}) \quad \text{bits per transmission}
\]  

So this is water filling again, but on the eigenvalues of \( K_Z \).

I.e., we are filling a diagonalized version of the problem. Once we get \( A_{ii} \), we can “re-correlate” using \( K_X = Q A Q^T \) to get final constraint on \( X \).
Stationary Stochastic (Gaussian) Processes

- It can be shown that we have water filling in this case as well, but this time in the frequency domain.
Stationary Stochastic (Gaussian) Processes

- It can be shown that we have water filling in this case as well, but this time in the frequency domain.
- Let $R_{ZZ}(e^{j\omega})$ be the power spectrum of the noise and signal discrete FT $X(e^{j\omega})$. 

\[
C = \int_{-\pi}^{\pi} \frac{1}{2} \log\left(1 + (\nu - R_{ZZ}(e^{j\omega})) + R_{ZZ}(e^{j\omega})\right) d\omega
\]
It can be shown that we have water filling in this case as well, but this time in the frequency domain.

Let $R_{ZZ}(e^{j\omega})$ be the power spectrum of the noise and signal discrete FT $X(e^{j\omega})$.

Then the capacity in this case becomes:

$$C = \int_{-\pi}^{\pi} \frac{1}{2} \log \left( 1 + \frac{(\nu - R_{ZZ}(e^{j\omega}))^+}{R_{ZZ}(e^{j\omega})} \right) d\omega$$

(9)

where $\nu$ is chosen to make

$$\int_{-\pi}^{\pi} (\nu - R_{ZZ}(e^{j\omega}))^+ d\omega = P$$

(10)
Spectral Water filling

Pictured:

So put energy first where the noise is low.
Feedback

- $C_n$ was the capacity without feedback, and $C_{n,\text{FB}}$ is the capacity with, and our goal here is to compare the two.
Feedback

- $C_n$ was the capacity without feedback, and $C_{n,FB}$ is the capacity with, and our goal here is to compare the two.

- For discrete **memoryless** channel, feedback does not increase capacity (since in finding $C$, you are already exploiting the randomness to the extent possible).
Feedback

- \( C_n \) was the capacity without feedback, and \( C_{n,FB} \) is the capacity with, and our goal here is to compare the two.

- For discrete memoryless channel, feedback does not increase capacity (since in finding \( C \), you are already exploiting the randomness to the extent possible).

- For discrete non-memoryless channels, feedback can help (but not by too much as we will see)
Feedback

- $C_n$ was the capacity without feedback, and $C_{n,FB}$ is the capacity with, and our goal here is to compare the two.

- For discrete memoryless channel, feedback does not increase capacity (since in finding $C$, you are already exploiting the randomness to the extent possible).

- For discrete non-memoryless channels, feedback can help (but not by too much as we will see)

Here, the analogy is that the noise is correlated, i.e.,

$$Z_{1:n}^\top \sim \mathcal{N}(0, K_Z^{(n)})$$  \hspace{1cm} (11)

where again $K_Z^{(n)}$ is not a diagonal matrix.
Feedback

- $C_n$ was the capacity without feedback, and $C_{n, FB}$ is the capacity with, and our goal here is to compare the two.

- For discrete [underline]memoryless[/underline] channel, feedback does not increase capacity (since in finding $C$, you are already exploiting the randomness to the extent possible).

- For discrete [underline]non-memoryless[/underline] channels, feedback can help (but not by too much as we will see)

- Here, the analogy is that the noise is correlated, i.e.,

$$Z_{1:n}^\top \sim \mathcal{N}(0, K_Z^{(n)})$$

(11)

where again $K_Z^{(n)}$ is not a diagonal matrix.

- Also, we have $Y_{1:n} = X_{1:n} + Z_{1:n}$ but **not** $X_{1:n} \perp \perp Z_{1:n}$ due to feedback (i.e., independence doesn’t hold in this case).
Feedback

- $C_n$ was the capacity without feedback, and $C_{n,FB}$ is the capacity with, and our goal here is to compare the two.

- For discrete memoryless channel, feedback does not increase capacity (since in finding $C$, you are already exploiting the randomness to the extent possible).

- For discrete non-memoryless channels, feedback can help (but not by too much as we will see)

- Here, the analogy is that the noise is correlated, i.e.,

$$Z_{1:n}^T \sim \mathcal{N}(0, K_Z^{(n)})$$  \hspace{1cm} (11)

where again $K_Z^{(n)}$ is not a diagonal matrix.

- Also, we have $Y_{1:n} = X_{1:n} + Z_{1:n}$ but not $X_{1:n} \perp \perp Z_{1:n}$ due to feedback (i.e., independence doesn’t hold in this case).

- Since noise is correlated, can use previous received values to help predict the noise.
Definition 3.1

We define a \((2^{nR}, n)\) code as a set of mappings of the form

\[
x_i(w, Y_{1:i-1}) : \{1, 2, \ldots, 2^{nR}\} \times \mathbb{R}^{(i-1)} \rightarrow \mathbb{R}
\]  

where \(w \in \{1, 2, \ldots, 2^{nR}\}\) is the input message and \(Y_{1:i-1}\) is the sequence of past output values (feedback). We think of \(x_i(w, \cdot)\) this as a “code function” rather than a mere codeword.

- We also have a power constraint of the form

\[
E \left[ \frac{1}{n} \sum_{i=1}^{n} x_i^2(w, Y_{1:i-1}) \right] \leq P \text{ for all } w \in \{1, 2, \ldots, 2^{nR}\}
\]
Feedback

We already saw that

$$C_n = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \frac{(\nu - \lambda_i)^+}{\lambda_i})$$

bits per transmission \hfill (14)

is the capacity for colored noise channel w/o feedback.
Feedback

- We already saw that

\[ C_n = \frac{1}{2} \sum_{i=1}^{n} \log \left(1 + \frac{(\nu - \lambda_i)^+}{\lambda_i} \right) \text{ bits per transmission} \quad (14) \]

is the capacity for colored noise channel w/o feedback.

- We want to see, in various different ways, how much better \( C_{n,\text{FB}} \) (the capacity with colored noise and feedback) is going to be.
Feedback

- We already saw that
  
  \[ C_n = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \frac{(\nu - \lambda_i)^+}{\lambda_i}) \]  
  
  bits per transmission \hspace{1cm} (14)

  is the capacity for colored noise channel w/o feedback.

- We want to see, in various different ways, how much better \( C_{n,\text{FB}} \) (the capacity with colored noise and feedback) is going to be.

- Feedback is useful if we have channel asymmetric communication costs or abilities. E.g.,
We already saw that

\[ C_n = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \frac{(\nu - \lambda_i)^+}{\lambda_i}) \]  \text{ bits per transmission (14)}

is the capacity for colored noise channel w/o feedback.

We want to see, in various different ways, how much better \( C_{n,FB} \) (the capacity with colored noise and feedback) is going to be.

Feedback is useful if we have channel asymmetric communication costs or abilities. E.g.,

- Cheap to communicate in one direction, expensive to communicate in the other direction (e.g., Mars rover, some sensor networks, internet service providers).
Feedback

We already saw that

\[ C_n = \frac{1}{2} \sum_{i=1}^{n} \log(1 + \frac{(\nu - \lambda_i)^+}{\lambda_i}) \]  

bits per transmission  \hspace{1cm} (14)

is the capacity for colored noise channel w/o feedback.

We want to see, in various different ways, how much better \( C_{n, FB} \) (the capacity with colored noise and feedback) is going to be.

Feedback is useful if we have channel asymmetric communication costs or abilities. E.g.,

- Cheap to communicate in one direction, expensive to communicate in the other direction (e.g., Mars rover, some sensor networks, internet service providers).
- Feasible to consider case when feedback is partly noisy (but asymmetries exist in the noise in different directions).
Bayesian Network View

Top: without feedback and uncorrelated noise

Bottom: without feedback and correlated noise
Bayesian Network View

With feedback and correlated noise:

Note, \( Z_i = Y_i - X_i \) which means that

\[
I(X_{1:n};Y_{1:n}) = h(Y_{1:n}) - h(Y_{1:n}|X_{1:n})
\]

\[
= h(X_{1:n} + Z_{1:n}) - h(Z_{1:n})
\]

max when Gaussian, Gaussian

\[ (15) \]

\[ (16) \]
Bayesian Network View

With feedback and correlated noise:

Note, $Z_i = Y_i - X_i$ which means that

$$I(X_{1:n}; Y_{1:n}) = h(Y_{1:n}) - h(Y_{1:n} | X_{1:n})$$

$$= h(X_{1:n} + Z_{1:n}) - h(Z_{1:n})$$

(15)

max when Gaussian

(16)

Gaussian

So, max is achieved when $X_{1:n}$ is also Gaussian, and in fact can be jointly Gaussian (exercise: show this).
feedback and correlated noise

- Note that $B$ controls the feedback — if $B = 0$ then no feedback.
feedback and correlated noise

- Note that $B$ controls the feedback — if $B = 0$ then no feedback.
- This is a general form of conditional Gaussian, where $V$ is also Gaussian and independent of $Z$. 

Note that $B$ controls the feedback — if $B = 0$ then no feedback. This is a general form of conditional Gaussian, where $V$ is also Gaussian and independent of $Z$. 

So we can write this capacity as the following optimization:

$$C_{n,FB} = \max_{1 \leq P} \frac{1}{n} \text{tr}(KXX) \leq P \frac{1}{2} \log \frac{|KX + Z|}{|KZ|} \tag{17}$$

where $KX + Z = \mathbb{E}[(X + Z)(X + Z)^\top] = \max_{K \succeq 0} \text{tr}(BKZ) = \text{pred}(B) = \text{tr}(BKZB^\top + KV) \leq nP$. Justification on next slide.
feedback and correlated noise

- Note that $B$ controls the feedback — if $B = 0$ then no feedback.
- This is a general form of conditional Gaussian, where $V$ is also Gaussian and independent of $Z$.
- So we can write this capacity as the following optimization:

$$C_{n, FB} = \max_{\frac{1}{n} \text{tr}(K_X) \leq P} 2^n \log \frac{|K_{X+Z}|}{|K_Z|}$$  

where $K_{X+Z} = E[(X + Z)(X + Z)^\top]$ 

$$\max_{K_V \succeq 0} \quad \frac{1}{2n} \log \frac{|(B + I)K_Z(B + I)^\top + K_V|}{|K_Z|}$$

$$\quad \text{pred}(B) = \text{true}$$

where $\text{pred}(B) = \text{tr}(BK_ZB^\top + K_V) \leq nP$. Justification on next slide.
Theorem 3.2

For a Gaussian channel with feedback, the rate $R_n$ for any sequence of $(2^{nR}, n)$ codes with $P_e^{(n)} \to 0$ satisfies $R_n \leq C_{n,FB} + \epsilon_n$ when $\epsilon_n \to 0$ as $n \to \infty$.

Next, we are going to prove a number of very useful matrix inequality theorems that are important in their own right, but that are also going to be useful to compare the capacity with and without feedback and we’ll find that feedback doesn’t help that much.
Theorem 3.2

For a Gaussian channel with feedback, the rate $R_n$ for any sequence of $(2^{nR}, n)$ codes with $P_e^{(n)} \to 0$ satisfies $R_n \leq C_{n,FB} + \epsilon_n$ when $\epsilon_n \to 0$ as $n \to \infty$.

- Next, we are going to prove a number of very useful matrix inequality theorems that are important in their own right, but that are also going to be useful to compare the capacity with and without feedback and we’ll find that feedback doesn’t help that much.

- See also chapter 17 in the book (2nd edition).
Matrix Inequalities

Theorem 4.1

∀ \( X, Z \) \( n \)-D random variables (not necessarily Gaussian), we have

\[
K_{X+Z} + K_{X-Z} = 2K_X + 2K_Z
\]  

(19)

Proof.

\[
K_{X+Z} = E(X + Z)(X + Z)^\top = EXX^\top + EXZ^\top + EZZ^\top
\]

\[
= K_X + K_{XZ} + K_{ZX} + K_Z
\]

(20)

\[
K_{X-Z} = E(X - Z)(X - Z)^\top = EXX^\top - EXZ^\top - EZZ^\top
\]

\[
= K_X - K_{XZ} - K_{ZX} + K_Z
\]

(21)

sum up both to get the result.

Prof. Jeff Bilmes  
page 17
Matrix Inequalities

Theorem 4.2

If $A \succeq 0$, $B \succeq 0$, and $A - B \succeq 0$, then $|A| \geq |B|$

Proof.

Let $C = A - B$, $X_1 \sim \mathcal{N}(0, B)$, $X_2 \sim \mathcal{N}(0, C)$, $X_1 \perp \perp X_2$. 

Then we get

$$h(Y) \geq h(Y | X_2) = h(X_1 | X_2) = h(X_1)$$

Then, using the entropy of Gaussian formula,

$$\frac{1}{2} \log((2\pi e)^n |A|) \geq \frac{1}{2} \log((2\pi e)^n |B|)$$

and the result follows.
Matrix Inequalities

Theorem 4.2

If $A \succeq 0$, $B \succeq 0$, and $A - B \succeq 0$, then $|A| \geq |B|$.

Proof.

- Let $C = A - B$, $X_1 \sim \mathcal{N}(0, B)$, $X_2 \sim \mathcal{N}(0, C)$, $X_1 \perp \perp X_2$.
- If $Y = X_1 + X_2$ then $Y \sim \mathcal{N}(0, B + C) = \mathcal{N}(0, A)$.
Matrix Inequalities

Theorem 4.2

If $A \succeq 0$, $B \succeq 0$, and $A - B \succeq 0$, then $|A| \geq |B|$

Proof.

1. Let $C = A - B$, $X_1 \sim \mathcal{N}(0, B)$, $X_2 \sim \mathcal{N}(0, C)$, $X_1 \perp \perp X_2$.
2. If $Y = X_1 + X_2$ then $Y \sim \mathcal{N}(0, B + C) = \mathcal{N}(0, A)$.
3. Then we get

$$h(Y) \geq h(Y|X_2) = h(X_1|X_2) = h(X_1)$$ (22)

Then, using the entropy of Gaussian formula,

$$\frac{1}{2} \log((2\pi e)^n |A|) \geq \frac{1}{2} \log((2\pi e)^n |B|)$$ (23)

and the result follows.
Matrix Inequalities

Theorem 4.2

If $A \succeq 0$, $B \succeq 0$, and $A - B \succeq 0$, then $|A| \geq |B|$

Proof.

- Let $C = A - B$, $X_1 \sim \mathcal{N}(0, B)$, $X_2 \sim \mathcal{N}(0, C)$, $X_1 \perp \perp X_2$.
- If $Y = X_1 + X_2$ then $Y \sim \mathcal{N}(0, B + C) = \mathcal{N}(0, A)$.
- Then we get
  \[ h(Y) \geq h(Y|X_2) = h(X_1|X_2) = h(X_1) \] (22)

- Then, using the entropy of Gaussian formula,
  \[
  \frac{1}{2} \log((2\pi e)^n |A|) \geq \frac{1}{2} \log((2\pi e)^n |B|)
  \] (23)

and the result follows
Matrix Inequalities

Theorem 4.3

Let $X, Z$ be arbitrary $n$-D random variables. Then

$$|K_{X+Z}| \leq 2^n|K_X + K_Z|$$  \hfill (24)

Proof.

$$2(K_X + K_Z) = K_{X+Z} + K_{X-Z}$$  \hfill (25)

$$|K_{X+Z}| \leq |2(K_X + K_Z)| = 2^n|K_{X+Z}|$$  \hfill (28)
Theorem 4.3

Let $X, Z$ be arbitrary $n$-D random variables. Then

$$|K_{X+Z}| \leq 2^n |K_X + K_Z| \quad (24)$$

Proof.

$$2(K_X + K_Z) = K_{X+Z} + K_{X-Z} \quad (25)$$

$$\Rightarrow 2(K_X + K_Z) - K_{X+Z} = K_{X-Z} \geq 0 \quad (26)$$

$$\Rightarrow |K_{X+Z}| \leq |2(K_X + K_Z)| = 2^n |K_X + K_Z| \quad (28)$$
Matrix Inequalities

Theorem 4.3

Let $X, Z$ be arbitrary $n$-D random variables. Then

$$|K_{X+Z}| \leq 2^n |K_X + K_Z|$$ (24)

Proof.

$$2(K_X + K_Z) = K_{X+Z} + K_{X-Z}$$ (25)

$$\Rightarrow 2(K_X + K_Z) - K_{X+Z} = K_{X-Z} \geq 0$$ (26)

$$\Rightarrow K_{X+Z} \leq 2(K_X + K_Z)$$ (27)

$$\Rightarrow |K_{X+Z}| \leq 2^n |K_X + K_Z|$$ (28)
Matrix Inequalities

Theorem 4.3

Let $X, Z$ be arbitrary $n$-D random variables. Then

$$|K_{X+Z}| \leq 2^n |K_X + K_Z|$$

(24)

Proof.

$$2(K_X + K_Z) = K_{X+Z} + K_{X-Z}$$

(25)

$$\Rightarrow 2(K_X + K_Z) - K_{X+Z} = K_{X-Z} \geq 0$$

(26)

$$\Rightarrow K_{X+Z} \leq 2(K_X + K_Z)$$

(27)

$$\Rightarrow |K_{X+Z}| \leq |2(K_X + K_Z)| = 2^n |K_X + K_Z|$$

(28)
Matrix Inequalities

Theorem 4.4

Let $A \succeq 0$, $B \succeq 0$, and $0 \leq \lambda \leq 1$, then

$$|\lambda A + (1 - \lambda)B| \geq |A|^\lambda |B|^{1-\lambda}$$

or in other words, $\log(|\lambda A + (1 - \lambda)B|) \geq \lambda \log|A| + (1 - \lambda) \log|B|$ or log-determinant is concave for non-negative definite matrices.

Proof.

1. Let $X \sim \mathcal{N}(0, A)$, $Y \sim \mathcal{N}(0, B)$
Matrix Inequalities

Theorem 4.4

Let $A \succeq 0$, $B \succeq 0$, and $0 \leq \lambda \leq 1$, then

$$|\lambda A + (1 - \lambda)B| \geq |A|^\lambda |B|^{1-\lambda} \tag{29}$$

or in other words, $\log(|\lambda A + (1 - \lambda)B|) \geq \lambda \log |A| + (1 - \lambda) \log |B|$ or log-determinant is concave for non-negative definite matrices.

Proof.

- Let $X \sim \mathcal{N}(0, A)$, $Y \sim \mathcal{N}(0, B)$
- Let $Z = X1(\theta = 1) + Y1(\theta = 0)$ with $X \perp \perp Y$ and where $P(\theta = 1) = 1 - P(\theta = 0) = \lambda$.

...
Matrix Inequalities

**Theorem 4.4**

Let $A \succeq 0$, $B \succeq 0$, and $0 \leq \lambda \leq 1$, then

$$|\lambda A + (1 - \lambda)B| \geq |A|^\lambda |B|^{1-\lambda}$$

(29)

or in other words, $\log(|\lambda A + (1 - \lambda)B|) \geq \lambda \log |A| + (1 - \lambda) \log |B|$ or log-determinant is concave for non-negative definite matrices.

**Proof.**

- Let $X \sim \mathcal{N}(0, A)$, $Y \sim \mathcal{N}(0, B)$
- Let $Z = X1(\theta = 1) + Y1(\theta = 0)$ with $X \perp \perp Y$ and where $P(\theta = 1) = 1 - P(\theta = 0) = \lambda$. Note $Z$ is not necessarily Gaussian.

...
Matrix Inequalities

**Theorem 4.4**

Let $A \succeq 0$, $B \succeq 0$, and $0 \leq \lambda \leq 1$, then

$$|\lambda A + (1 - \lambda)B| \geq |A|^\lambda |B|^{1-\lambda}$$

or in other words, $\log(|\lambda A + (1 - \lambda)B|) \geq \lambda \log |A| + (1 - \lambda) \log |B|$ or log-determinant is concave for non-negative definite matrices.

**Proof.**

- Let $X \sim \mathcal{N}(0, A)$, $Y \sim \mathcal{N}(0, B)$
- Let $Z = X1(\theta = 1) + Y1(\theta = 0)$ with $X \perp \perp Y$ and where $P(\theta = 1) = 1 - P(\theta = 0) = \lambda$. Note $Z$ is not necessarily Gaussian.

foo
Matrix Inequalities

\[ \begin{align*}
\text{... proof continued.} \\
\bullet \quad \text{Then} \\
\end{align*} \]

Thus, we get

\[ \begin{align*}
(32) \\
(33) \\
(34) \\
(35)
\end{align*} \]
... proof continued.

Then

\[ K_Z = EZZ^\top = E[X_1(\theta = 1) + Y_1(\theta = 0)]^2 \tag{30} \]

\[ = E[XX_1^\top(\theta = 1)1(\theta = 1)] + E[XY_1^\top(\theta = 1)1(\theta = 0)] + E[YX_1^\top(\theta = 0)1(\theta = 1)] + E[YY_1^\top(\theta = 1)1(\theta = 1)] \tag{31} \]

\[ = \lambda A + (1 - \lambda) B \tag{32} \]

Thus, we get

\[ \frac{1}{2} \log \left( \frac{2\pi e n |\lambda A + (1 - \lambda) B|}{\lambda |X| + (1 - \lambda) |Y|} \right) \geq h(Z) \geq h(Z|\theta) \tag{33} \]

\[ = \lambda h(X) + (1 - \lambda) h(Y) = \frac{1}{2} \log \left( \frac{2\pi e n |A| |\lambda| |B| (1 - \lambda)}{1 - \lambda} \right) \tag{35} \]
... proof continued.

Then

\[ K_Z = EZZ^\top \]  

Thus, we get

\[ \frac{1}{2} \log \left( \frac{2\pi e}{|\lambda A + (1 - \lambda)B|n} \right) \geq h(Z) \geq h(Z | \theta) = \frac{1}{2} \log \left( \frac{2\pi e}{|A|\lambda + |B|1 - \lambda} \right) \]  

(32)
... proof continued.

Then

$$K_Z = EZZ^\top = E[(X1(\theta = 1) + Y1(\theta = 0))^{\text{''2''}}]$$  \hfill (30)

Thus, we get

$$\frac{1}{2} \log((2\pi e)^n | \lambda A + (1 - \lambda) B |) \geq h(Z) \geq h(Z | \theta) \hfill (33)$$

$$= \lambda h(X) + (1 - \lambda) h(Y) = \frac{1}{2} \log((2\pi e)^n | A | \lambda | B | 1 - \lambda)$$  \hfill (35)
... proof continued.

Then

$$K_{Z} = EZZ^\top = E[(X\mathbf{1}(\theta = 1) + Y\mathbf{1}(\theta = 0))^2]$$

$$= E[XX^\top\mathbf{1}(\theta = 1)^2] + E[XY^\top\mathbf{1}(\theta = 1)\mathbf{1}(\theta = 0)]$$

$$+ E[YY^\top\mathbf{1}(\theta = 0)^2]$$

Thus, we get

$$= \lambda A + (1 - \lambda) B$$

(30)

(31)

(32)

(35)
Then

\[ K_Z = EZZ^\top = E[(X1(\theta = 1) + Y1(\theta = 0))^2] \]
\[ = E[XX^\top 1(\theta = 1)1(\theta = 1)] + E[XY^\top 1(\theta = 1)1(\theta = 0)] \]
\[ = \lambda A + (1 - \lambda)B \]  

Thus, we get

\[ \geq h(Z) \]
\[ = \frac{1}{2} \log((2\pi e)^n |\lambda A + (1 - \lambda)B|) \]
Matrix Inequalities

... proof continued.

Then

\[ K_Z = EZZ^\top = E[(X1(\theta = 1) + Y1(\theta = 0))^{\text{2}}] \]
\[ = E[XX^\top 1(\theta = 1)1(\theta = 1)] + E[XY^\top 1(\theta = 1)1(\theta = 0)] \]
\[ = \lambda A + (1 - \lambda)B \] (30)

Thus, we get

\[ \frac{1}{2} \log((2\pi e)^n|\lambda A + (1 - \lambda)B|) \] (33)

(35)
\[ K_Z = EZZ^\top = E[(X \mathbf{1}(\theta = 1) + Y \mathbf{1}(\theta = 0))^{"2"}] \]
\[ = E[XX^\top \mathbf{1}(\theta = 1) \mathbf{1}(\theta = 1)] + E[XY^\top \mathbf{1}(\theta = 1) \mathbf{1}(\theta = 0)] \]
\[ + E[XY^\top \mathbf{1}(\theta = 0) \mathbf{1}(\theta = 1)] + E[YY^\top \mathbf{1}(\theta = 1) \mathbf{1}(\theta = 1)] \]
\[ = \lambda A + (1 - \lambda)B \] (30)

Thus, we get

\[ \frac{1}{2} \log((2\pi e)^n |\lambda A + (1 - \lambda)B|) \geq h(Z) \] (33)

\[ \geq \lambda h(X) + (1 - \lambda)h(Y) \] (35)
Then

\[ K_Z = EZZ^\top = E[(X1(\theta = 1) + Y1(\theta = 0))^{\prime 2}] \]
\[ = E[XX^\top 1(\theta = 1)1(\theta = 1)] + E[XY^\top 1(\theta = 1)1(\theta = 0)] \]
\[ = \lambda A + (1 - \lambda)B \]  

Thus, we get

\[ \frac{1}{2} \log((2\pi e)^n|\lambda A + (1 - \lambda)B|) \]
\[ \geq h(Z) \geq h(Z|\theta) \]

(30)

(31)

(32)

(33)

(34)

(35)
Matrix Inequalities

... proof continued.

Then

\[
K_Z = EZZ^\top = E[(X1(\theta = 1) + Y1(\theta = 0))^{\prime2}] \\
= E[XX^\top 1(\theta = 1)1(\theta = 1)] + E[XY^\top 1(\theta = 1)1(\theta = 0)] \\
+ E[YY^\top 1(\theta = 0)1(\theta = 1)] + E[YY^\top 1(\theta = 1)1(\theta = 1)] \\
= \lambda A + (1 - \lambda)B
\]  

(30)

(31)

(32)

Thus, we get

\[
\frac{1}{2} \log((2\pi e)^n|\lambda A + (1 - \lambda)B|) \\
\geq h(Z) \geq h(Z|\theta) \\
\geq \lambda h(X) + (1 - \lambda)h(Y)
\]

(33)

(34)

(35)
Then
\[ K_Z = EZZ^\top = E[(X1(\theta = 1) + Y1(\theta = 0))^2] \]
\[ = E[XX^\top 1(\theta = 1)1(\theta = 1)] + E[XY^\top 1(\theta = 1)1(\theta = 0)] \]
\[ + E[YX^\top 1(\theta = 0)1(\theta = 1)] + E[YY^\top 1(\theta = 1)1(\theta = 1)] \]
\[ = \lambda A + (1 - \lambda)B \] (32)

Thus, we get
\[ \frac{1}{2} \log((2\pi e)^n|\lambda A + (1 - \lambda)B|) \]
\[ \geq h(Z) \geq h(Z|\theta) \] (34)
\[ = \lambda h(X) + (1 - \lambda)h(Y) = \frac{1}{2} \log((2\pi e)^n|A|^{\lambda}|B|^{1-\lambda}) \] (35)
We define a distribution that **factorizes generatively** if:

\[
 f(x_{1:n}, z_{1:n}) = f(x_{1:n}) \prod_{i} f(x_i | x_{1:i-1}, z_{1:i-1}) \tag{36}
\]

Note this is called “causally related” in the book, but it need not have anything to do with causality, which can be something entirely different (see the work of J. Pearl for discussions on and models of causality).
We define a distribution that factorizes generatively if:

\[ f(x_{1:n}, z_{1:n}) = f(x_{1:n}) \prod_i f(x_i | x_{1:i-1}, z_{1:i-1}) \] (36)

- Note this is called “causally related” in the book, but it need not have anything to do with causality, which can be something entirely different (see the work of J. Pearl for discussions on and models of causality).
- We can view this pictorially:
Definition 4.6

We define a distribution that factorizes generatively if:

\[ f(x_{1:n}, z_{1:n}) = f(x_{1:n}) \prod_i f(x_i | x_{1:i-1}, z_{1:i-1}) \]  \hspace{1cm} (37)

called “causally related” in the book, but it need not have anything to do with causality, which can be something entirely different (see the work of J. Pearl for discussions on and models of causality).
**Definition 4.6**

We define a distribution that \textbf{factorizes generatively} if:

\[
f(x_{1:n}, z_{1:n}) = f(x_{1:n}) \prod_{i} f(x_i | x_{1:i-1}, z_{1:i-1})
\]  

(37)

- called “causally related” in the book, but it need not have anything to do with causality, which can be something entirely different (see the work of J. Pearl for discussions on and models of causality).

- Pictorially:
Since feedback codes at capacity are of the form \( X_i = f(Z_{1:i-1}) = f(Z_{1:i-1}, X_{1:i-1}) \) the above generative factorization model applies in this case as well.

**Theorem 4.7**

if \( x_{1:n} \) and \( z_{1:n} \) factorize generatively (not nec. Gaussian), then

\[
h(x_{1:n} - z_{1:n}) \geq h(z_{1:n})
\]

and

\[
|K_{X-Z}| \geq |K_Z|
\]
Proof.

\[ h(X_{1:n} - Z_{1:n}) = \sum_i h(X_i - Z_i | X_{1:i-1} - Z_{1:i-1}) \quad \text{(chain rule)} \]  

(40)
Proof.

\[ h(X_{1:n} - Z_{1:n}) = \sum_i h(X_i - Z_i | X_{1:i-1} - Z_{1:i-1}) \quad (\text{chain rule}) \]  

\[ \geq \sum_i h(X_i - Z_i | X_{1:i-1}, Z_{1:i-1}, X_i) \quad (\text{conditioning}) \]  

\[ = \sum_i h(Z_i | X_{1:i-1}, Z_{1:i-1}, X_i) \quad (\text{factorization}) \]  

\[ = h(Z_{1:n}) \quad (\text{chain rule}) \]
Proof.

\[ h(X_{1:n} - Z_{1:n}) = \sum_i h(X_i - Z_i | X_{1:i-1} - Z_{1:i-1}) \]  

\[ \geq \sum_i h(X_i - Z_i | X_{1:i-1}, Z_{1:i-1}, X_i) \]  

\[ = \sum_i h(Z_i | X_{1:i-1}, Z_{1:i-1}, X_i) \]  

\[ = \sum_i h(Z_i | X_{1:i-1}) \]  

\[ (\text{chain rule}) \quad (40) \]

\[ (\text{conditioning}) \quad (41) \]

\[ (\text{factorization}) \quad (42) \]

\[ (44) \]
Proof.

\[ h(X_{1:n} - Z_{1:n}) = \sum_i h(X_i - Z_i | X_{1:i-1} - Z_{1:i-1}) \quad (\text{chain rule}) \quad (40) \]

\[ \geq \sum_i h(X_i - Z_i | X_{1:i-1}, Z_{1:i-1}, X_i) \quad (\text{conditioning}) \quad (41) \]

\[ = \sum_i h(Z_i | X_{1:i-1}, Z_{1:i-1}, X_i) \quad (h(X_i | X_i) = 0) \quad (42) \]

\[ = \sum_i h(Z_i | Z_{1:i-1}) \quad (\text{factorization}) \quad (43) \]

\[ \ldots \quad (44) \]
Proof.

\[ h(X_{1:n} - Z_{1:n}) = \sum_i h(X_i - Z_i | X_{1:i-1} - Z_{1:i-1}) \]  \hspace{1cm} (chain rule) \hspace{1cm} (40)

\[ \geq \sum_i h(X_i - Z_i | X_{1:i-1}, Z_{1:i-1}, X_i) \]  \hspace{1cm} (conditioning) \hspace{1cm} (41)

\[ = \sum_i h(Z_i | X_{1:i-1}, Z_{1:i-1}, X_i) \]  \hspace{1cm} (h(X_i | X_i) = 0) \hspace{1cm} (42)

\[ = \sum_i h(Z_i | Z_{1:i-1}) \]  \hspace{1cm} (factorization) \hspace{1cm} (43)

\[ = h(Z_{1:n}) \]  \hspace{1cm} (chain rule) \hspace{1cm} (44)

\[ \ldots \]
Next, let $\tilde{X}_{1:n}, \tilde{Z}_{1:n}$ be independent Gaussian r.v.’s with covariance $K_X, K_Z$ respectively and let factorization assumption hold. Then

$$\frac{1}{n} \log \left( \frac{1}{n} \right) = h(\tilde{X} - \tilde{Z})$$

(45)

and the result follows due to monotonicity of the log

Now we are ready to start comparing $C_n$ (capacity without feedback) and $C_n, FB$ (capacity with feedback). We’ll do this both with an additive bound and a multiplicative bound.
Next, let $\tilde{X}_{1:n}, \tilde{Z}_{1:n}$ be independent Gaussian r.v.’s with covariance $K_X, K_Z$ respectively and let factorization assumption hold. Then

$$\frac{1}{n} \log[(2\pi e)^n |K_{X-Z}|]$$

(46)
Next, let $\tilde{X}_{1:n}, \tilde{Z}_{1:n}$ be independent Gaussian r.v.’s with covariance $K_X, K_Z$ respectively and let factorization assumption hold. Then

$$\frac{1}{n} \log [(2\pi e)^n |K_{X-Z}|] = h(\tilde{X} - \tilde{Z})$$  

(45)

$$= \frac{1}{2} \log((2\pi e)^n |K_Z|)$$  

(46)

Now we are ready to start comparing $C_n$ (capacity without feedback) and $C_{n, FB}$ (capacity with feedback). We’ll do this both with an additive bound and a multiplicative bound.
Next, let $X_{1:n}^{\tilde{}}$, $Z_{1:n}^{\tilde{}}$ be independent Gaussian r.v.’s with covariance $K_X$, $K_Z$ respectively and let factorization assumption hold. Then

$$\frac{1}{n} \log[(2\pi e)^n |K_{X-Z}|] = h(\tilde{X} - \tilde{Z})$$

(45)

$$\geq h(\tilde{Z})$$

(46)
Next, let $\tilde{X}_{1:n}, \tilde{Z}_{1:n}$ be independent Gaussian r.v.'s with covariance $K_X, K_Z$ respectively and let factorization assumption hold. Then

$$\frac{1}{n} \log[(2\pi e)^n |K_{X-Z}|] = h(\tilde{X} - \tilde{Z})$$  \hspace{1cm} (45)$$

$$\geq h(\tilde{Z}) = \frac{1}{2} \log((2\pi e)^n |K_Z|)$$  \hspace{1cm} (46)$$

Now we are ready to start comparing $C_n$ (capacity without feedback) and $C_{n,FB}$ (capacity with feedback). We'll do this both with an additive bound and a multiplicative bound.
Next, let $X_{\tilde{1}:n}, Z_{\tilde{1}:n}$ be independent Gaussian r.v.’s with covariance $K_X, K_Z$ respectively and let factorization assumption hold. Then

$$\frac{1}{n} \log[(2\pi e)^n|K_{X-Z}|] = h(\tilde{X} - \tilde{Z})$$ (45)

$$\geq h(\tilde{Z}) = \frac{1}{2} \log((2\pi e)^n|K_Z|)$$ (46)

and the result follows due to monotonicity of the log.

... proof continued.
Next, let $X_{1:n}, Z_{1:n}$ be independent Gaussian r.v.’s with covariance $K_X, K_Z$ respectively and let factorization assumption hold. Then

$$\frac{1}{n} \log[(2\pi e)^n |K_{X-Z}|] = h(\tilde{X} - \tilde{Z})$$

$$\geq h(\tilde{Z}) = \frac{1}{2} \log((2\pi e)^n |K_{Z}|)$$

and the result follows due to monotonicity of the log.

Now we are ready to start comparing $C_n$ (capacity without feedback) and $C_{n,FB}$ (capacity with feedback).
factorization

... proof continued.

Next, let $\tilde{X}_{1:n}, \tilde{Z}_{1:n}$ be independent Gaussian r.v.’s with covariance $K_X, K_Z$ respectively and let factorization assumption hold. Then

$$\frac{1}{n} \log[(2\pi e)^n|K_{X-Z}|] = h(\tilde{X} - \tilde{Z})$$

$$\geq h(\tilde{Z}) = \frac{1}{2} \log((2\pi e)^n|K_Z|)$$

and the result follows due to monotonicity of the log

- Now we are ready to start comparing $C_n$ (capacity without feedback) and $C_{n,FB}$ (capacity with feedback).
- We’ll do this both with an additive bound and a multiplicative bound.
Additive Bound

Theorem 4.8

\[ C_{n, FB} \leq C_n + \frac{1}{2} \]  \hspace{1cm} (47)

So at most 1/2 bit per channel use of gain!

Proof.

\[ C_{n, FB} \]
Theorem 4.8

\[ C_{n,FB} \leq C_n + \frac{1}{2} \] (47)

So at most 1/2 bit per channel use of gain!

Proof.

\[ C_{n,FB} \leq \frac{1}{n} \max \left\{ \frac{1}{2} \log \left| \frac{K_Y}{K_Z} \right| \right\} \] (48)

\[ \leq C_n + \frac{1}{2} \] (50)
Theorem 4.8

\[ C_{n,FB} \leq C_n + 1/2 \]  

So at most 1/2 bit per channel use of gain!

Proof.

\[ C_{n,FB} \leq \max_{\frac{1}{n} \text{tr}(K_X) \leq P} \frac{1}{2n} \log \frac{|K_Y|}{|K_Z|} \]  

\[ \leq \max_{\frac{1}{n} \text{tr}(K_X) \leq P} \frac{1}{2n} \log \frac{2^n |K_X+Z|}{|K_Z|} \]  

\[ \leq \max_{\frac{1}{n} \text{tr}(K_X) \leq P} \frac{1}{2n} \log \frac{2^n |K_X|}{|K_Z|} + \frac{1}{2} \]
Additive Bound

Theorem 4.8

\[ C_{n,FB} \leq C_n + 1/2 \]  

(47)

So at most 1/2 bit per channel use of gain!

Proof.

\[
C_{n,FB} \leq \max_{\frac{1}{n} \text{tr}(K_X) \leq P} \frac{1}{2n} \log \frac{|K_Y|}{|K_Z|} \\
\leq \max_{\frac{1}{n} \text{tr}(K_X) \leq P} \frac{1}{2n} \log \frac{2^n|K_{X+Z}|}{|K_Z|} \\
\leq \max_{\frac{1}{n} \text{tr}(K_X) \leq P} \frac{1}{2n} \log \frac{|K_{X+Z}|}{|K_Z|} + \frac{1}{2} 
\]

(48)

(49)

(50)

(51)
Theorem 4.8

\[ C_{n,FB} \leq C_n + 1/2 \]  

So at most 1/2 bit per channel use of gain!

Proof.

\[
C_{n,FB} \leq \max_{\frac{1}{n} \text{tr}(K_X) \leq P} \frac{1}{2n} \log \frac{|K_Y|}{|K_Z|} 
\leq \max_{\frac{1}{n} \text{tr}(K_X) \leq P} \frac{1}{2n} \log \frac{2^n |K_{X+Z}|}{|K_Z|} 
\leq C_n + \frac{1}{2}
\]
Theorem 4.9

\[ C_{n,FB} \leq 2C_n \] (52)

or equivalently

\[
\frac{1}{2n} \log \frac{|K_X + Z|}{|K_Z|} \leq 2 \frac{1}{2n} \log \frac{|K_X + K_Z|}{|K_Z|}
\] (53)
Multiplicative Bound

Proof.

\[
\frac{1}{2} \cdot \frac{1}{2n} \log \left| \frac{KX + Z}{KZ} \right|
\]

(59)
Proof.

\[
\frac{1}{2} \frac{1}{2n} \log \frac{|K_{X+Z}|}{|K_Z|} = \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2}}{|K_Z|^{1/2}}
\]  

(54)
Proof.

\[ \frac{1}{2} \frac{1}{2n} \log \frac{|K_{X+Z}|}{|K_Z|} = \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2}}{|K_Z|^{1/2}} \]  

(54)

\[ = \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2}|K_Z|^{1/2}}{|K_Z|} \]  

(55)

\[ \leq \frac{1}{2} \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2} |K_Z|^{1/2}}{|K_{X-Z}|^{1/2}} \]  

by Thm 4.7  

(56)

\[ \leq \frac{1}{2} \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2} |K_Z|^{1/2}}{|K_X|^{1/2} |K_{X-Z}|^{1/2}} \]  

by Thm 4.4  

(57)

\[ = \frac{1}{2} \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2}|K_Z|^{1/2}}{|K_Z|} \]  

by Thm 4.1  

(58)
Proof.

\[
\frac{1}{2} \frac{1}{2n} \log \frac{|K_{X+Z}|}{|K_Z|} = \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2}}{|K_Z|^{1/2}}
\]

(54)

\[
= \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2} |K_Z|^{1/2}}{|K_Z|}
\]

(55)

\[
\leq \frac{1}{2n} \log \frac{|K_{X-Z}|^{1/2} |K_Z|^{1/2}}{|K_Z|}
\]

by Thm 4.7

(56)

\[
= \frac{1}{2} \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2} + |K_{X-Z}|^{1/2}}{|K_Z|}
\]

(58)
Multiplicative Bound

Proof.

\[
\frac{1}{2} \frac{1}{2n} \log \frac{|K_{X+Z}|}{|K_Z|} = \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2}|K_Z|^{1/2}}{|K_Z|} = \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2}|K_{X-Z}|^{1/2}}{|K_Z|} \leq \frac{1}{2n} \log \frac{\frac{1}{2}K_{X+Z} + \frac{1}{2}K_{X-Z}}{|K_Z|} \tag{54}
\]

\[
\leq \frac{1}{2n} \log \frac{|\frac{1}{2}K_{X+Z} + \frac{1}{2}K_{X-Z}|}{|K_Z|} \tag{55}
\]

\[
= \frac{1}{2n} \log \frac{|\frac{1}{2}K_{X+Z} + \frac{1}{2}K_{X-Z}|}{|K_Z|} \leq \frac{1}{2n} \log \frac{|\frac{1}{2}K_{X+Z} + \frac{1}{2}K_{X-Z}|}{|K_Z|} \tag{56}
\]

by Thm 4.7

\[
\leq \frac{1}{2n} \log \frac{|\frac{1}{2}K_{X+Z} + \frac{1}{2}K_{X-Z}|}{|K_Z|} \tag{57}
\]

by Thm 4.4

\[
\leq \frac{1}{2n} \log \frac{|\frac{1}{2}K_{X+Z} + \frac{1}{2}K_{X-Z}|}{|K_Z|} \tag{58}
\]

by Thm 4.1

\[
\frac{1}{2n} \log \frac{|\frac{1}{2}K_{X+Z} + \frac{1}{2}K_{X-Z}|}{|K_Z|} = \frac{1}{2n} \log \frac{|\frac{1}{2}K_{X+Z} + \frac{1}{2}K_{X-Z}|}{|K_Z|} \tag{59}
\]
Multiplicative Bound

Proof.

\[ \frac{1}{2} \frac{1}{2n} \log \left| \frac{K_{X+Z}}{K_Z} \right| = \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2}}{|K_Z|^{1/2}} \]

\[ = \frac{1}{2n} \log \frac{|K_{X+Z}|^{1/2}|K_{X-Z}|^{1/2}}{|K_Z|} \]

\[ \leq \frac{1}{2n} \log \frac{|\frac{1}{2}K_{X+Z} + \frac{1}{2}K_{X-Z}|}{|K_Z|} \]

\[ = \frac{1}{2n} \log \frac{|K_X + K_Z|}{|K_Z|} \]

by Thm 4.7

by Thm 4.4

by Thm 4.1
Corollary 4.10

\[ C_{n,FB} \leq \min \{2C_n, C_n + 1/2\} \] (60)

So unfortunately, feedback in this model is not as useful as we might think it would be.
We know that the source compresses down to the entropy $H$, but no further.
We know that the source compresses down to the entropy $H$, but no further.

We also know that the signal may be sent through the channel at a rate no more than $C$. 

$$
\log P_e \quad C \quad R \rightarrow
$$
Coding/Compression with distortion

- What if we want to compress $R < H$ or transmit $R > C$?
What if we want to compress $R < H$ or transmit $R > C$? ⇒ Error.
What if we want to compress $R < H$ or transmit $R > C$? ⇒ Error.

Similarly, what if we allow for errors, but rather than measure error or no error, measure average distortion.
Coding/Compression with distortion

- What if we want to compress $R < H$ or transmit $R > C$? $\Rightarrow$ Error.
- Similarly, what if we allow for errors, but rather than measure error or no error, measure average distortion.
- But are all errors created equality? Are all errors as bad as others?
What if we want to compress $R < H$ or transmit $R > C$? ⇒ Error.

Similarly, what if we allow for errors, but rather than measure error or no error, measure average distortion.

But are all errors created equality? Are all errors as bad as others?

We can measure errors with a distortion function, and we have generalization of the previously stated results.
What if we want to compress $R < H$ or transmit $R > C$? ⇒ Error.

Similarly, what if we allow for errors, but rather than measure error or no error, measure average distortion.

But are all errors created equality? Are all errors as bad as others?

We can measure errors with a distortion function, and we have generalization of the previously stated results.

Rate-distortion curves with achievable region
Vector Quantization

- We have symbols $X \in \mathcal{X}$ which could be a continuous or a (say big) discrete domain.
Vector Quantization

- We have symbols $X \in \mathcal{X}$ which could be a continuous or a (say big) discrete domain.
- We quantize this region to $\hat{\mathcal{X}}$ where $\hat{\mathcal{X}}$ is discrete and not too big (if $\hat{\mathcal{X}}$ is discrete, then $|\hat{\mathcal{X}}| < |\mathcal{X}|$. 

In above, $\hat{\mathcal{X}} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_7\}$, $|\hat{\mathcal{X}}| = 7 = M$. There are a set of regions $R_i, i = 1, \ldots, M$, disjoint so that $R_i \cap R_j = \emptyset$ for $i \neq j$, and with $\hat{x}_i \in R_i$ for all $i$. 

Prof. Jeff Bilmes
Vector Quantization

- We have symbols $X \in \mathcal{X}$ which could be a continuous or a (say big) discrete domain.
- We quantize this region to $\hat{\mathcal{X}}$ where $\hat{\mathcal{X}}$ is discrete and not too big (if $\hat{\mathcal{X}}$ is discrete, then $|\hat{\mathcal{X}}| < |\mathcal{X}|$).

In above, $\hat{\mathcal{X}} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_7\}$, $|\hat{\mathcal{X}}| = 7 = M$.

There are a set of regions $\mathcal{R}_i$, $i = 1, ..., M$, disjoint so that $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ for $i \neq j$, and with $\hat{x}_i \in \mathcal{R}_i$ for all $i$. 
Vector Quantization

- We have symbols $X \in \mathcal{X}$ which could be a continuous or a (say big) discrete domain.
- We quantize this region to $\hat{\mathcal{X}}$ where $\hat{\mathcal{X}}$ is discrete and not too big (if $\hat{\mathcal{X}}$ is discrete, then $|\hat{\mathcal{X}}| < |\mathcal{X}|$).

$\mathcal{X} \Rightarrow \hat{\mathcal{X}}$

- In above, $\hat{\mathcal{X}} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_7\}$, $|\hat{\mathcal{X}}| = 7 = M$
We have symbols $X \in \mathcal{X}$ which could be a continuous or a (say big) discrete domain.

We quantize this region to $\hat{\mathcal{X}}$ where $\hat{\mathcal{X}}$ is discrete and not too big (if $\hat{\mathcal{X}}$ is discrete, then $|\hat{\mathcal{X}}| < |\mathcal{X}|$).

In above, $\hat{\mathcal{X}} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_7\}$, $|\hat{\mathcal{X}}| = 7 = M$

There are a set of regions $\mathcal{R}_i$, $i = 1, \ldots, M$, disjoint so that $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ for $i \neq j$, and with $\hat{x}_i \in \mathcal{R}_i$ for all $i$. 
Vector Quantization

- The regions cover the entire $\mathcal{X}$ space (i.e., $\bigcup_i \mathcal{R}_i = \mathcal{X}$).
Vector Quantization

- The regions cover the entire $\mathcal{X}$ space (i.e., $\bigcup_i \mathcal{R}_i = \mathcal{X}$).
- In general, there are $M$ regions.
Vector Quantization

- The regions cover the entire $\mathcal{X}$ space (i.e., $\bigcup_i \mathcal{R}_i = \mathcal{X}$).
- In general, there are $M$ regions.
- Within each region lies a representative, or exemplar, of that region. That is, for any $x \in \mathcal{R}_i$, then $x$ is represented by, say, $\hat{x}_i \in \mathcal{R}_i$.
The regions cover the entire $\mathcal{X}$ space (i.e., $\bigcup_i \mathcal{R}_i = \mathcal{X}$).

In general, there are $M$ regions.

Within each region lies a representative, or exemplar, of that region. That is, for any $x \in \mathcal{R}_i$, then $x$ is represented by, say, $\hat{x}_i \in \mathcal{R}_i$.

For all $x \in \mathcal{X}$, define function

$$f : \mathcal{X} \rightarrow \{1, 2, \ldots, M\}, \text{ with } f(x) = i \text{ where } x \in \mathcal{R}_i \quad (61)$$
Vector Quantization

- The regions cover the entire $\mathcal{X}$ space (i.e., $\bigcup_i \mathcal{R}_i = \mathcal{X}$).
- In general, there are $M$ regions.
- Within each region lies a representative, or exemplar, of that region. That is, for any $x \in \mathcal{R}_i$, then $x$ is represented by, say, $\hat{x}_i \in \mathcal{R}_i$.
- For all $x \in \mathcal{X}$, define function

$$f: \mathcal{X} \rightarrow \{1, 2, \ldots, M\}, \text{ with } f(x) = i \text{ where } x \in \mathcal{R}_i$$  \hspace{1cm} (61)

- For all $x \in \mathcal{X}$, define function

$$g: \{1, 2, \ldots, M\} \rightarrow \mathcal{X}, \text{ with } f(i) = \hat{x}_i$$  \hspace{1cm} (62)
Vector Quantization

- The regions cover the entire $\mathcal{X}$ space (i.e., $\bigcup_i \mathcal{R}_i = \mathcal{X}$).
- In general, there are $M$ regions.
- Within each region lies a representative, or exemplar, of that region. That is, for any $x \in \mathcal{R}_i$, then $x$ is represented by, say, $\hat{x}_i \in \mathcal{R}_i$.
- For all $x \in \mathcal{X}$, define function
  \[ f : \mathcal{X} \rightarrow \{1, 2, \ldots, M\}, \text{ with } f(x) = i \text{ where } x \in \mathcal{R}_i \] (61)
- For all $x \in \mathcal{X}$, define function
  \[ g : \{1, 2, \ldots, M\} \rightarrow \mathcal{X}, \text{ with } f(i) = \hat{x}_i \] (62)
- Thus, $g(f(x)) = \hat{x}_i$ if $x \in \mathcal{R}_i$. I.e., $x$’s representative is $g(f(x))$. 

This is a quantization of $\mathcal{X}$ with codebook the set $\{\hat{x}_i\}_{i=1}^M$. 

Vector Quantization

- The regions cover the entire $\mathcal{X}$ space (i.e., $\cup_i \mathcal{R}_i = \mathcal{X}$).
- In general, there are $M$ regions.
- Within each region lies a representative, or exemplar, of that region. That is, for any $x \in \mathcal{R}_i$, then $x$ is represented by, say, $\hat{x}_i \in \mathcal{R}_i$.
- For all $x \in \mathcal{X}$, define function

  $$f : \mathcal{X} \to \{1, 2, \ldots, M\}, \text{ with } f(x) = i \text{ where } x \in \mathcal{R}_i \quad (61)$$

- For all $x \in \mathcal{X}$, define function

  $$g : \{1, 2, \ldots, M\} \to \mathcal{X}, \text{ with } f(i) = \hat{x}_i \quad (62)$$

- Thus, $g(f(x)) = \hat{x}_i$ if $x \in \mathcal{R}_i$. I.e., $x$’s representative is $g(f(x))$.
- This is a quantization of $\mathcal{X}$ with codebook the set $\{\hat{x}_i\}_{i=1}^M$. 

Prof. Jeff Bilmes  
page 34
The “quality” of the codebook depends on

The values \(\{\hat{x}_i\}_{i=1}^{M}\) How often each of the values within \(\{\hat{x}_i\}_{i=1}^{M}\) are used, or more accurately, a probability distribution \(p(x)\) over \(X\).

A measure of how bad it is to represent \(x \in X\) by \(g(f(x))\), or a distortion \(d(x, g(f(x)))\).

Expected distortion \(D = E_{p(x)} d(X, g(f(X)))\).
Vector Quantization

The “quality” of the codebook depends on

1. The values $\{\hat{x}_i\}_{i=1}^M$
The “quality” of the codebook depends on

1. The values $\{\hat{x}_i\}_{i=1}^M$

2. How often each of the values within $\{\hat{x}_i\}_{i=1}^M$ are used, or more accurately, a probability distribution $p(x)$ over $\mathcal{X}$. 

Expected distortion $D = \mathbb{E}_{p(x)} d(\mathcal{X}, g(f(\mathcal{X})))$. 

Prof. Jeff Bilmes

The “quality” of the codebook depends on

1. The values \( \{\hat{x}_i\}_{i=1}^M \)
2. How often each of the values within \( \{\hat{x}_i\}_{i=1}^M \) are used, or more accurately, a probability distribution \( p(x) \) over \( X \).
3. A measure of how bad it is to represent \( x \in X \) by \( g(f(x)) \), or a distortion \( d(x, g(f(x))) \).
The “quality” of the codebook depends on

1. The values \( \{\hat{x}_i\}_{i=1}^{M} \)

2. How often each of the values within \( \{\hat{x}_i\}_{i=1}^{M} \) are used, or more accurately, a probability distribution \( p(x) \) over \( \mathcal{X} \).

3. A measure of how bad it is to represent \( x \in \mathcal{X} \) by \( g(f(x)) \), or a distortion \( d(x, g(f(x))) \).

Expected distortion \( D = E_{p(x)}d(X, g(f(X))) \).
The “Vector” in Vector Quantization

Now, let's say that we want to “code” a vector $x \in \mathcal{X}^n$ (e.g., say an $n$-D vector $x$ where each element $x_i \in \mathcal{X}$.

- Suppose $M = 2^{nR}$ for some $R > 0$.

  - The $R$ can be viewed as a rate in terms of number of bits per source symbol. $R = \frac{\log M}{n} = \frac{\log 2^{nR}}{n} = \frac{nR}{n} = R$ bits per source symbol (63).

Intuitively, as rate $\uparrow$ it should be possible for the distortion $D \downarrow$. Let's now be a little more formal.
The “Vector” in Vector Quantization

- Now, let’s say that we want to “code” a vector $x \in \mathcal{X}^n$ (e.g., say an $n$-D vector $x$ where each element $x_i \in \mathcal{X}$.

- We quantize the entire space $\mathcal{X}^n$ into $M$ disjoint regions $\{\mathcal{R}_i\}_i$ that cover $\mathcal{X}^n$ (so picture before really showed case for $\mathcal{X}^2$, or $n = 2$).
The “Vector” in Vector Quantization

- Now, let’s say that we want to “code” a vector \( x \in \mathcal{X}^n \) (e.g., say an \( n \)-D vector \( x \) where each element \( x_i \in \mathcal{X} \).
- We quantize the entire space \( \mathcal{X}^n \) into \( M \) disjoint regions \( \{R_i\}_i \) that cover \( \mathcal{X}^n \) (so picture before really showed case for \( \mathcal{X}^2 \), or \( n = 2 \)).
- Each vector \( x \in \mathcal{X}^n \) consists of \( n \) source symbols (each source symbol lives in \( \mathcal{X} \)).
The “Vector” in Vector Quantization

Now, let's say that we want to “code” a vector $x \in \mathcal{X}^n$ (e.g., say an $n$-D vector $x$ where each element $x_i \in \mathcal{X}$.

We quantize the entire space $\mathcal{X}^n$ into $M$ disjoint regions $\{R_i\}_i$ that cover $\mathcal{X}^n$ (so picture before really showed case for $\mathcal{X}^2$, or $n = 2$).

Each vector $x \in \mathcal{X}^n$ consists of $n$ source symbols (each source symbol lives in $\mathcal{X}$).

Suppose $M = 2^{nR}$ for some $R > 0$. 
The “Vector” in Vector Quantization

- Now, let’s say that we want to “code” a vector \( x \in \mathcal{X}^n \) (e.g., say an \( n \)-D vector \( x \) where each element \( x_i \in \mathcal{X} \).

- We quantize the entire space \( \mathcal{X}^n \) into \( M \) disjoint regions \( \{ \mathcal{R}_i \} \) that cover \( \mathcal{X}^n \) (so picture before really showed case for \( \mathcal{X}^2 \), or \( n = 2 \)).

- Each vector \( x \in \mathcal{X}^n \) consists of \( n \) source symbols (each source symbol lives in \( \mathcal{X} \)).

- Suppose \( M = 2^{nR} \) for some \( R > 0 \).

- The \( R \) can be viewed as a rate in terms of number of bits per source symbol

\[
\text{rate} = \frac{\log M}{n} = \frac{\log 2^{nR}}{n} = \frac{nR}{n} = R \text{ bits per source symbol} \quad (63)
\]
The “Vector” in Vector Quantization

- Now, let's say that we want to “code” a vector $x \in \mathcal{X}^n$ (e.g., say an $n$-D vector $x$ where each element $x_i \in \mathcal{X}$).
- We quantize the entire space $\mathcal{X}^n$ into $M$ disjoint regions $\{R_i\}_i$ that cover $\mathcal{X}^n$ (so picture before really showed case for $\mathcal{X}^2$, or $n = 2$).
- Each vector $x \in \mathcal{X}^n$ consists of $n$ source symbols (each source symbol lives in $\mathcal{X}$).
- Suppose $M = 2^{nR}$ for some $R > 0$.
- The $R$ can be viewed as a rate in terms of number of bits per source symbol

$$\text{rate} = \frac{\log M}{n} = \frac{\log 2^{nR}}{n} = \frac{nR}{n} = R \text{ bits per source symbol} \quad (63)$$

- Intuitively, as rate ↑ it should be possible for the distortion $D \downarrow$
The “Vector” in Vector Quantization

- Now, let's say that we want to “code” a vector $x \in \mathcal{X}^n$ (e.g., say an $n$-D vector $x$ where each element $x_i \in \mathcal{X}$).
- We quantize the entire space $\mathcal{X}^n$ into $M$ disjoint regions $\{R_i\}_i$ that cover $\mathcal{X}^n$ (so picture before really showed case for $\mathcal{X}^2$, or $n = 2$).
- Each vector $x \in \mathcal{X}^n$ consists of $n$ source symbols (each source symbol lives in $\mathcal{X}$).
- Suppose $M = 2^{nR}$ for some $R > 0$.
- The $R$ can be viewed as a rate in terms of number of bits per source symbol
  \[
  \text{rate} = \frac{\log M}{n} = \frac{\log 2^{nR}}{n} = \frac{nR}{n} = R \text{ bits per source symbol (63)}
  \]
- Intuitively, as rate ↑ it should be possible for the distortion $D$ ↓
- Let's now be a little more formal
Set up

- A source produces $x_1, x_2, \ldots$ $p(x)$ based on source distribution $p(x)$ with $x_i \in \mathcal{X}$ for all $i$. 
Set up

- A **source** produces \( x_1, x_2, \ldots \) \( p(x) \) based on source distribution \( p(x) \) with \( x_i \in \mathcal{X} \) for all \( i \).
- An **encoder** \( f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\} \) takes a sequence of source symbols \( x_{1:n} \) and maps them to an integer.
Set up

- A **source** produces $x_1, x_2, \ldots, p(x)$ based on source distribution $p(x)$ with $x_i \in \mathcal{X}$ for all $i$.

- An **encoder** $f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\}$ takes a sequence of source symbols $x_1:n$ and maps them to an integer:

- A **decoder** $g_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \hat{\mathcal{X}}^n$ takes an integer and maps to quantized vector.

Ex: Hamming (probability of error) distortion.

$$d(x, \hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{otherwise} \end{cases}$$
Set up

- A source produces $x_1, x_2, \ldots, p(x)$ based on source distribution $p(x)$ with $x_i \in \mathcal{X}$ for all $i$.
- An encoder $f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\}$ takes a sequence of source symbols $x_1^n$ and maps them to an integer.
- A decoder $g_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \hat{\mathcal{X}}^n$ takes an integer and maps to quantized vector.
- A distortion function $d : \mathcal{X} \times \hat{\mathcal{X}} \rightarrow \mathbb{R}^+$ measures how bad the mapping is. I.e., $d(x, \hat{x})$ measures the “cost” of representing $x \in \mathcal{X}$ by $\hat{x} \in \hat{\mathcal{X}}$. 

Distortion is bounded (sometimes needed) if $\exists d_{\text{max}}$ such that $d_{\text{max}} \equiv \max_{x, \hat{x}} d(x, \hat{x}) < \infty$.

Ex: Hamming (probability of error) distortion. $d(x, \hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{otherwise} \end{cases}$

Then $E_d(\mathcal{X}, \hat{\mathcal{X}}) = \Pr(\mathcal{X} \neq \hat{\mathcal{X}})$
A source produces $x_1, x_2, \ldots, p(x)$ based on source distribution $p(x)$ with $x_i \in X$ for all $i$.

An encoder $f_n : X^n \to \{1, 2, \ldots, 2^{nR}\}$ takes a sequence of source symbols $x_{1:n}$ and maps them to an integer:

A decoder $g_n : \{1, 2, \ldots, 2^{nR}\} \to \hat{X}^n$ takes an integer and maps to quantized vector.

A distortion function $d : X \times \hat{X} \to \mathbb{R}^+$ measures how bad the mapping is. I.e., $d(x, \hat{x})$ measures the “cost” of representing $x \in X$ by $\hat{x} \in \hat{X}$.

Distortion is bounded (sometimes needed) if $\exists d_{\text{max}}$ such that $d_{\text{max}} \triangleq \max_{x, \hat{x}} d(x, \hat{x}) < \infty$. 

Ex: Hamming (probability of error) distortion.

$d(x, \hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{otherwise} \end{cases}$ 

Then $Ed(X, \hat{X}) = \Pr(X \neq \hat{X})$.
Set up

A source produces \( x_1, x_2, \ldots, p(x) \) based on source distribution \( p(x) \) with \( x_i \in \mathcal{X} \) for all \( i \).

An encoder \( f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\} \) takes a sequence of source symbols \( x_{1:n} \) and maps them to an integer.

A decoder \( g_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \hat{\mathcal{X}}^n \) takes an integer and maps to quantized vector.

A distortion function \( d : \mathcal{X} \times \hat{\mathcal{X}} \rightarrow \mathbb{R}^+ \) measures how bad the mapping is. I.e., \( d(x, \hat{x}) \) measures the “cost” of representing \( x \in \mathcal{X} \) by \( \hat{x} \in \hat{\mathcal{X}} \).

Distortion is bounded (sometimes needed) if \( \exists d_{\text{max}} \) such that

\[
d_{\text{max}} \triangleq \max_{x, \hat{x}} d(x, \hat{x}) < \infty.
\]

Ex: Hamming (probability of error) distortion.

\[
d(x, \hat{x}) = \begin{cases} 0 & \text{if } x = \hat{x} \\ 1 & \text{otherwise} \end{cases}
\]
**Set up**

- A source produces $x_1, x_2, \ldots, p(x)$ based on source distribution $p(x)$ with $x_i \in \mathcal{X}$ for all $i$.

- An encoder $f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\}$ takes a sequence of source symbols $x_{1:n}$ and maps them to an integer.

- A decoder $g_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \hat{\mathcal{X}}^n$ takes an integer and maps to quantized vector.

- A **distortion** function $d : \mathcal{X} \times \hat{\mathcal{X}} \rightarrow \mathbb{R}^+$ measures how bad the mapping is. I.e., $d(x, \hat{x})$ measures the “cost” of representing $x \in \mathcal{X}$ by $\hat{x} \in \hat{\mathcal{X}}$.

- Distortion is **bounded** (sometimes needed) if $\exists d_{\text{max}}$ such that $d_{\text{max}} \triangleq \max_{x, \hat{x}} d(x, \hat{x}) < \infty$.

- Ex: Hamming (probability of error) distortion.

\[
d(x, \hat{x}) = \begin{cases} 
0 & \text{if } x = \hat{x} \\
1 & \text{otherwise}
\end{cases}
\]  

Then $\text{Ed}(X, \hat{X}) = \Pr(X \neq \hat{X})$
Another possible distortion for $x \in \mathbb{R}^n$ might be squared error

$$d(x, \hat{x}) = (x - \hat{x})^2,$$

or 2-norm $d(x, \hat{x}) = \|x - \hat{x}\|^2$ or $p$-norm

$$d(x, \hat{x}) = \|x - \hat{x}\|^p$$
Another possible distortion for $x \in \mathbb{R}^n$ might be squared error $d(x, \hat{x}) = (x - \hat{x})^2$, or 2-norm $d(x, \hat{x}) = \| x - \hat{x} \|^2$ or $p$-norm $d(x, \hat{x}) = \| x - \hat{x} \|^p$.

We can form the extended distortion as follows

$$d(X_{1:n}, \hat{X}_{1:n}) \triangleq \frac{1}{n} \sum_{i=1}^{n} d(x_i, \hat{x}_i)$$ (65)
Definition 5.1

A \((2^{nR}, n)\) rate distortion code consists of an encoding function

\[ f_n : \mathcal{X}^n \rightarrow \{1, 2, \ldots, 2^{nR}\} \tag{66} \]

and a decoding function

\[ g_n : \{1, 2, \ldots, 2^{nR}\} \rightarrow \hat{\mathcal{X}}^n \tag{67} \]

(Note, \(H(\hat{\mathcal{X}}^n) \leq nR\) since only \(2^{nR}\) different codewords.)

The distortion of this code is

\[ D = Ed(X_{1:n}, g_n(f_n(X_{1:n}))) = \sum_{x_{1:n} \in \mathcal{X}^n} p(x_{1:n})d(x_{1:n}, g_n(f_n(x_{1:n}))) \tag{68} \]
Logistics
Review
Feedback
Matrix Inequalities by IT
Rate Distortion
Scratch

Comments

- So very much like compression
So very much like compression

rate of the code $R$ is in bits per source symbol, and bigger rates are worse (from compression point of view).
So very much like compression

rate of the code $R$ is in bits per source symbol, and bigger rates are worse (from compression point of view).

Ideally, we want to compress as much as possible while keeping the distortion minimal.
So very much like compression
rate of the code $R$ is in bits per source symbol, and bigger rates are worse (from compression point of view).

Ideally, we want to compress as much as possible while keeping the distortion minimal.

We have a codebook which consists of $2^{nR}$ codewords.

$$(g_n(1), g_n(2), \ldots, g_n(2^{nR})) = (\hat{X}^n(1), \hat{X}^n(2), \ldots, \hat{X}^n(2^{nR}))$$ (69)
So very much like compression

rate of the code $R$ is in bits per source symbol, and bigger rates are worse (from compression point of view).

Ideally, we want to compress as much as possible while keeping the distortion minimal.

We have a codebook which consists of $2^{nR}$ codewords.

$$ (g_n(1), g_n(2), \ldots, g_n(2^{nR})) = (\hat{X}^n(1), \hat{X}^n(2), \ldots, \hat{X}^n(2^{nR})) \quad (69) $$

Each codeword is a vector, i.e., $g_n(i) \in \mathcal{X}^n$. 

Prof. Jeff Bilmes
So very much like compression
rate of the code $R$ is in bits per source symbol, and bigger rates are worse (from compression point of view).

Ideally, we want to compress as much as possible while keeping the distortion minimal.

We have a codebook which consists of $2^{nR}$ codewords.

\[(g_n(1), g_n(2), \ldots, g_n(2^{nR})) = (\hat{X}^n(1), \hat{X}^n(2), \ldots, \hat{X}^n(2^{nR}))\] (69)

Each codeword is a vector, i.e., $g_n(i) \in \mathcal{X}^n$.

The assignment regions are now refereed to using $f_n^{-1}(i)$ for the $i$'th assignment region. Same as $\mathcal{R}_i$ from before.
So very much like compression

rate of the code $R$ is in bits per source symbol, and bigger rates are worse (from compression point of view).

Ideally, we want to compress as much as possible while keeping the distortion minimal.

We have a codebook which consists of $2^{nR}$ codewords.

\[(g_n(1), g_n(2), \ldots, g_n(2^{nR})) = (\hat{X}^n(1), \hat{X}^n(2), \ldots, \hat{X}^n(2^{nR})) \quad (69)\]

Each codeword is a vector, i.e., $g_n(i) \in X^n$.

The assignment regions are now refereed to using $f_n^{-1}(i)$ for the $i$'th assignment region. Same as $R_i$ from before.

Due to distortion, quite likely we will have more than one source string map to same target string. I.e., exists $x, x' \in X$ with $x \neq x'$ such that $g_n(f_n(x)) = g_n(f_n(x'))$. 
A rate-distortion pair \((R, D)\) is said to be achievable if there exists a sequence of \((2^n R, n)\) codes \((f_n, g_n)\) with

\[
\lim_{n \to \infty} Ed(X_{1:n}, g_n(f_n(X_{1:n}))) \leq D
\]

So \(D\) is the max allowable distortion.
Achievability and rate-distortion pairs

Definition 5.2

A rate-distortion pair \((R, D)\) is said to be achievable if there exists a sequence of \((2^nR, n)\) codes \((f_n, g_n)\) with

\[
\lim_{n \to \infty} Ed(X_{1:n}, g_n(f_n(X_{1:n}))) \leq D
\]  

(70)

- So \(D\) is the max allowable distortion.
- We can make errors, but not too many (bounded average distortion).
A rate-distortion pair \((R, D)\) is said to be achievable if \(\exists\) a sequence of \((2^nR, n)\) codes \((f_n, g_n)\) with

\[
\lim_{n \to \infty} Ed(X_{1:n}, g_n(f_n(X_{1:n}))) \leq D
\]

So \(D\) is the max allowable distortion.
We can make errors, but not too many (bounded average distortion).
The type of errors we can make is entirely dependent on the distortion function.
Definition 5.2

A rate-distortion pair \((R, D)\) is said to be achievable if \(\exists\) a sequence of \((2^{nR}, n)\) codes \((f_n, g_n)\) with

\[
\lim_{n \to \infty} Ed(X_{1:n}, g_n(f_n(X_{1:n}))) \leq D
\]  

(70)

- So \(D\) is the max allowable distortion.
- We can make errors, but not too many (bounded average distortion).
- The type of errors we can make is entirely dependent on the distortion function.
- Def: A rate distortion region for a source is the closure of achievable rate distortion pairs \((R, D)\)
Achievability and rate-distortion pairs

Definition 5.2

A rate-distortion pair \((R, D)\) is said to be **achievable** if \(\exists\) a sequence of \((2^n R, n)\) codes \((f_n, g_n)\) with

\[
\lim_{n \to \infty} Ed(X_{1:n}, g_n(f_n(X_{1:n}))) \leq D
\]  

(70)

- So \(D\) is the max allowable distortion.
- We can make errors, but not too many (bounded average distortion).
- The type of errors we can make is entirely dependent on the distortion function.
- Def: A **rate distortion region** for a source is the closure of achievable rate distortion pairs \((R, D)\)
- Def: A **rate distortion function** \(R(D)\) is the infimum of rates \(R\) such that \((R, D)\) is in rate distortion region. I.e.,

\[
R(D) = \inf \{ R : (R, D) \text{ is achievable} \}
\]

(71)
Achievability and rate-distortion pairs

- Def: A distortion rate function \( D(R) \) is the infimum of distortions \( D \) such that \((R, D)\) is in rate distortion region. I.e.,

\[
D(R) = \inf \{ D : (R, D) \text{ is achievable} \} \quad (72)
\]
Achievability and rate-distortion pairs

- **Def:** A distortion rate function $D(R)$ is the infimum of distortions $D$ such that $(R, D)$ is in rate distortion region. I.e.,

$$D(R) = \inf \{ D : (R, D) \text{ is achievable} \}$$  \hspace{1cm} (72)

- The next definition is very important
Achievability and rate-distortion pairs

- Def: A distortion rate function $D(R)$ is the infimum of distortions $D$ such that $(R, D)$ is in rate distortion region. I.e.,

$$D(R) = \inf \{ D : (R, D) \text{ is achievable} \} \quad (72)$$

- The next definition is very important

**Definition 5.3**

The “information” rate distortion function $R^{(I)}(D)$ for source $X$ and distortion $d(x, \hat{x})$ is defined as

$$R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_{x, \hat{x}} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \quad (73)$$
Achievability and rate-distortion pairs

- **Def:** A *distortion rate function* $D(R)$ is the infimum of distortions $D$ such that $(R, D)$ is in rate distortion region. I.e.,

$$D(R) = \inf \{ D : (R, D) \text{ is achievable} \} \quad (72)$$

- The next definition is very important

**Definition 5.3**

The “information” rate distortion function $R^{(I)}(D)$ for source $X$ and distortion $d(x, \hat{x})$ is defined as

$$R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_x \sum_{\hat{x}} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D} I(X; \hat{X}) \quad (73)$$

- Lets now spend a bit of time getting some intuition on this function.
\[ R^{(I)}(D) = \min_{p(\hat{x}|x)} \sum_{x, \hat{x}} p(x) p(\hat{x}|x) d(x, \hat{x}) \leq D \]

For fixed \( p(x) \), \( I(X; \hat{X}) \) is convex in \( p(\hat{x}|x) \). \( \Rightarrow \) convex optimization.
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x} | x)} \sum_{x, \hat{x}} p(x)p(\hat{x} | x) d(x, \hat{x}) \leq D \]

For fixed \( p(x) \), \( I(X; \hat{X}) \) is convex in \( p(\hat{x} | x) \). \( \Rightarrow \) convex optimization.

We will see in fact how we will use certain methods in convex optimization later (alternating minimization) for computing \( R^{(I)}(D) \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_x p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

(75)

- Related to lossy entropy compression.
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_{x, \hat{x}} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D} I(X; \hat{X}) \]  (75)

- Related to lossy entropy compression.
- Suppose \( D = Ed(X, \hat{X}) = 0 \) (no distortion) and recall \( d(x, \hat{x}) \geq 0 \) by definition.
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_{x,\hat{x}} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

(75)

- Related to lossy entropy compression.
- Suppose \( D = Ed(X, \hat{X}) = 0 \) (no distortion) and recall \( d(x, \hat{x}) \geq 0 \) by definition.
- Thus, we must at least have that \( \forall x, \hat{x} : p(x, \hat{x}) > 0, d(x, \hat{x}) = 0 \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum x, \hat{x} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

(75)

- Related to lossy entropy compression.
- Suppose \( D = Ed(X, \hat{X}) = 0 \) (no distortion) and recall \( d(x, \hat{x}) \geq 0 \) by definition.
- Thus, we must at least have that \( \forall x, \hat{x} : p(x, \hat{x}) > 0, d(x, \hat{x}) = 0 \).
- Consider distortions of the form: \( d(x, \hat{x}) = 0 \Rightarrow \{ x = \hat{x} \} \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_x, \hat{x} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \] (75)

- Related to lossy entropy compression.
- Suppose \( D = Ed(X, \hat{X}) = 0 \) (no distortion) and recall \( d(x, \hat{x}) \geq 0 \) by definition.
- Thus, we must at least have that \( \forall x, \hat{x} : p(x, \hat{x}) > 0, d(x, \hat{x}) = 0 \).
- Consider distortions of the form: \( d(x, \hat{x}) = 0 \Rightarrow \{x = \hat{x}\} \). For example, \( d(x, \hat{x}) = 1_{\{x \neq \hat{x}\}} \), or alternatively \( d(x, \hat{x}) = (x - \hat{x})^2 \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x) : \sum_{x,\hat{x}} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \] (75)

- Related to lossy entropy compression.
- Suppose \( D = Ed(X, \hat{X}) = 0 \) (no distortion) and recall \( d(x, \hat{x}) \geq 0 \) by definition.
- Thus, we must at least have that \( \forall x, \hat{x} : p(x, \hat{x}) > 0, d(x, \hat{x}) = 0 \).
- Consider distortions of the form: \( d(x, \hat{x}) = 0 \Rightarrow \{x = \hat{x}\} \). For example, \( d(x, \hat{x}) = 1_{\{x \neq \hat{x}\}} \), or alternatively \( d(x, \hat{x}) = (x - \hat{x})^2 \).
- Thus, \( \forall x, \hat{x}, \) if \( p(x, \hat{x}) > 0 \) then \( x = \hat{x} \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_x, \hat{x} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

(75)

- Related to lossy entropy compression.
- Suppose \( D = Ed(X, \hat{X}) = 0 \) (no distortion) and recall \( d(x, \hat{x}) \geq 0 \) by definition.
- Thus, we must at least have that \( \forall x, \hat{x} : p(x, \hat{x}) > 0, d(x, \hat{x}) = 0 \).
- Consider distortions of the form: \( d(x, \hat{x}) = 0 \Rightarrow \{ x = \hat{x} \} \). For example, \( d(x, \hat{x}) = 1_{\{ x \neq \hat{x} \}} \), or alternatively \( d(x, \hat{x}) = (x - \hat{x})^2 \).
- Thus, \( \forall x, \hat{x} \), if \( p(x, \hat{x}) > 0 \) then \( x = \hat{x} \). Or, the random variables are such that \( X = \hat{X} \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x) : \sum_{x, \hat{x}} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D} I(X; \hat{X}) \]  

(75)

- Related to lossy entropy compression.
- Suppose \( D = Ed(X, \hat{X}) = 0 \) (no distortion) and recall \( d(x, \hat{x}) \geq 0 \) by definition.
- Thus, we must at least have that \( \forall x, \hat{x} : p(x, \hat{x}) > 0, d(x, \hat{x}) = 0 \).
- Consider distortions of the form: \( d(x, \hat{x}) = 0 \Rightarrow \{x = \hat{x}\} \). For example, \( d(x, \hat{x}) = 1_{\{x \neq \hat{x}\}} \), or alternatively \( d(x, \hat{x}) = (x - \hat{x})^2 \).
- Thus, \( \forall x, \hat{x} \), if \( p(x, \hat{x}) > 0 \) then \( x = \hat{x} \). Or, the random variables are such that \( X = \hat{X} \)
- Hence,

\[ I(X; \hat{X}) = I(X; X) = H(X) \]  

(76)
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_x, \hat{x} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

(75)

- Related to lossy entropy compression.
- Suppose \( D = Ed(X, \hat{X}) = 0 \) (no distortion) and recall \( d(x, \hat{x}) \geq 0 \) by definition.
- Thus, we must at least have that \( \forall x, \hat{x} : p(x, \hat{x}) > 0, d(x, \hat{x}) = 0 \).
- Consider distortions of the form: \( d(x, \hat{x}) = 0 \Rightarrow \{ x = \hat{x} \} \). For example, \( d(x, \hat{x}) = 1 \{ x \neq \hat{x} \} \), or alternatively \( d(x, \hat{x}) = (x - \hat{x})^2 \).
- Thus, \( \forall x, \hat{x} \), if \( p(x, \hat{x}) > 0 \) then \( x = \hat{x} \). Or, the random variables are such that \( X = \hat{X} \).
- Hence,

\[ I(X; \hat{X}) = I(X; X) = H(X) \]  

(76)

- And if \( p(x) \) is uniform, then

\[ R^{(I)}(D) = H(X) \]  

(77)
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_x \sum_{\hat{x}} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

(78)

- So, in this case we see that \( D = 0 \) implies \( P_e = 0 \) (zero probability of error).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_{x, \hat{x}} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D} I(X; \hat{X}) \]  

(78)

- So, in this case we see that \( D = 0 \) implies \( P_e = 0 \) (zero probability of error).
- In general, does \( \{ P_e = 0 \} \Rightarrow \{ D = 0 \} \)?

Yes, as long as \( d(x, \hat{x}) = 0 \).

In general, does \( \{ D = 0 \} \Rightarrow \{ P_e = 0 \} \)?

No. Consider \( d(x, \hat{x}) = \begin{cases} (x - \hat{x})^2 - \tau & \text{if } (x - \hat{x})^2 > \tau \\ 0 & \text{else} \end{cases} \)
\[ R^{(I)}(D) = \min_{p(\hat{x}|x) : \sum_{x, \hat{x}} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D} I(X; \hat{X}) \]  

So, in this case we see that \( D = 0 \) implies \( P_e = 0 \) (zero probability of error).

In general, does \( \{P_e = 0\} \Rightarrow \{D = 0\} \)? Yes, as long as \( d(x, x) = 0 \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum x, \hat{x} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

- So, in this case we see that \( D = 0 \) implies \( P_e = 0 \) (zero probability of error).
- In general, does \( \{P_e = 0\} \Rightarrow \{D = 0\} \)? Yes, as long as \( d(x, \hat{x}) = 0 \).
- In general, does \( \{D = 0\} \Rightarrow \{P_e = 0\} \)? No. Consider \( d(x, \hat{x}) = \begin{cases} (x - \hat{x})^2 - \tau & \text{if } (x - \hat{x})^2 > \tau \\ 0 & \text{else} \end{cases} \) 

Prof. Jeff Bilmes
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x) : \sum x, \hat{x} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

(78)

- So, in this case we see that \( D = 0 \) implies \( P_e = 0 \) (zero probability of error).
- In general, does \( \{P_e = 0\} \Rightarrow \{D = 0\}? \) Yes, as long as \( d(x, x) = 0 \).
- In general, does \( \{D = 0\} \Rightarrow \{P_e = 0\}? \) No. Consider

\[ d(x, \hat{x}) = \begin{cases} 
(x - \hat{x})^2 - \tau & \text{if } (x - \hat{x})^2 > \tau \\
0 & \text{else}
\end{cases} \]  

(79)
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x} | x): \sum_{x, \hat{x}} p(x)p(\hat{x} | x)d(x, \hat{x}) \leq D} I(X; \hat{X}) \]

Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:

1. This is a process of compression at a particular rate and distortion.
2. \( I(X; \hat{X}) \) measures the rate of transmission from source symbols to compressed form.
3. To have a high compression ratio, this rate should be low.

Consider channel \( X \rightarrow \hat{X} \) used to represent \( X \) (e.g., mp3 file, mp4 video file, etc.)

Ultimate (lossy) compression would lose all information about \( X \), meaning we’d communicate at a rate of \( R = 0 \leq I(X; \hat{X}) \).

To compress well, want the communication rate from source \( X \) to its representation \( \hat{X} \) to be small.

Without distortion constraint, we’d hit \( R = 0 \). With distortion constraint \( Ed(X, \hat{X}) \leq D \), this keeps us from communicating at a rate of \( R = 0 \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_x, \hat{x} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:

1. This is a process of compression at a particular rate and distortion.

To have a high compression ratio, this rate should be low.

Consider channel \( X \rightarrow \hat{X} \) used to represent \( X \) (e.g., mp3 file, mp4 video file, etc.)

Ultimate (lossy) compression would lose all information about \( X \), meaning we’d communicate at a rate of \( R = 0 \leq I(X; \hat{X}) \).

To compress well, want the communication rate from source \( X \) to its representation \( \hat{X} \) to be small.

Without distortion constraint, we’d hit \( R = 0 \) With distortion constraint \( Ed(x,\hat{x}) \leq D \), this keeps us from communicating at a rate of \( R = 0 \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_{x, \hat{x}} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D} I(X; \hat{X}) \]  

(80)

Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:

1. This is a process of compression at a particular rate and distortion.
2. \( I(X; \hat{X}) \) measures the rate of transmission from source symbols to compressed form.
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x)} \sum_{x, \hat{x}} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D \]

Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:

1. This is a process of compression at a particular rate and distortion.
2. \( I(X; \hat{X}) \) measures the rate of transmission from source symbols to compressed form.
3. To have a high compression ratio, this rate should be low.
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_{x,\hat{x}} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:

1. This is a process of compression at a particular rate and distortion.
2. \( I(X; \hat{X}) \) measures the rate of transmission from source symbols to compressed form.
3. To have a high compression ratio, this rate should be low.

Consider channel \( X \rightarrow \text{channel} \rightarrow \hat{X} \)
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x} | x) : \sum_{x, \hat{x}} p(x)p(\hat{x} | x)d(x, \hat{x}) \leq D} I(X; \hat{X}) \]  

Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:

1. This is a process of compression at a particular rate and distortion.
2. \( I(X; \hat{X}) \) measures the rate of transmission from source symbols to compressed form.
3. To have a high compression ratio, this rate should be low.

Consider channel \( X \rightarrow \text{channel} \rightarrow \hat{X} \)

\( \hat{X} \) used to represent \( X \) (e.g., mp3 file, mp4 video file, etc.)
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x)} \sum x, \hat{x} p(x)p(\hat{x}|x)d(x, \hat{x}) \leq D \]

(80)

Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:

1. This is a process of compression at a particular rate and distortion.
2. \( I(X; \hat{X}) \) measures the rate of transmission from source symbols to compressed form.
3. To have a high compression ratio, this rate should be low.

Consider channel \( X \rightarrow \text{channel} \rightarrow \hat{X} \)

\( \hat{X} \) used to represent \( X \) (e.g., mp3 file, mp4 video file, etc.)

Ultimate (lossy) compression would loose all information about \( X \), meaning we’d communicate at a rate of \( R = 0 \leq I(X; \hat{X}) \).
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x) : \sum_x, \hat{x} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  \quad (80)

- Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:
  1. This is a process of compression at a particular rate and distortion.
  2. \( I(X; \hat{X}) \) measures the rate of transmission from source symbols to compressed form.
  3. To have a high compression ratio, this rate should be low.

- Consider channel \( X \rightarrow \) [channel] \( \rightarrow \hat{X} \)
- \( \hat{X} \) used to represent \( X \) (e.g., mp3 file, mp4 video file, etc.)
- Ultimate (lossy) compression would lose all information about \( X \), meaning we’d communicate at a rate of \( R = 0 \leq I(X; \hat{X}) \).
- To compress well, want the communication rate from source \( X \) to its representation \( \hat{X} \) to be small.
\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_x p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \] (80)

Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:

1. This is a process of compression at a particular rate and distortion.
2. \( I(X; \hat{X}) \) measures the rate of transmission from source symbols to compressed form.
3. To have a high compression ratio, this rate should be low.

Consider channel \( X \rightarrow \text{channel} \rightarrow \hat{X} \)

\( \hat{X} \) used to represent \( X \) (e.g., mp3 file, mp4 video file, etc.)

Ultimate (lossy) compression would lose all information about \( X \), meaning we’d communicate at a rate of \( R = 0 \leq I(X; \hat{X}) \).

To compress well, want the communication rate from source \( X \) to its representation \( \hat{X} \) to be small.

Without distortion constraint, we’d hit \( R = 0 \)
Intuition: Information Rate Distortion Function

\[ R^{(I)}(D) = \min_{p(\hat{x}|x): \sum_x,\hat{x} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D} I(X; \hat{X}) \]  

(80)

- Intuition, why we want to minimize \( I(X; \hat{X}) \) is that:
  1. This is a process of compression at a particular rate and distortion.
  2. \( I(X; \hat{X}) \) measures the rate of transmission from source symbols to compressed form.
  3. To have a high compression ratio, this rate should be low.

- Consider channel \( X \rightarrow \text{channel} \rightarrow \hat{X} \)

- \( \hat{X} \) used to represent \( X \) (e.g., mp3 file, mp4 video file, etc.)

- Ultimate (lossy) compression would loose all information about \( X \), meaning we’d communicate at a rate of \( R = 0 \leq I(X; \hat{X}) \).

- To compress well, want the communication rate from source \( X \) to its representation \( \hat{X} \) to be small.

- Without distortion constraint, we’d hit \( R = 0 \)

- With distortion constraint \( Ed(X, \hat{X}) \leq D \), this keeps us from communicating at a rate of \( R = 0 \).
The rate-distortion function $R(D)$ for Bernoulli($p$) with $d(x, \hat{x}) = 1_{\{x \neq \hat{x}\}}$ (Hamming distortion) has the following form:

$$R(D) = \begin{cases} 
H(p) - H(D) & \text{if } 0 \leq D \leq \min \{ p, 1 - p \} \\
0 & \text{if } D > \min \{ p, 1 - p \}
\end{cases}$$

(81)
Binary Source $R(D)$, $X \sim \text{Bernoulli}(p)$ r.v.

- **Rate Distortion Region**
  \[ R(D) = \inf \{ R : (R, D) \text{ is achievable} \} \]

- When $D = 0$, minimum rate is the entropy, and can’t compress below the entropy with zero distortion.

- As $D \uparrow$, we can “compress” more, below the entropy, but we suffer some distortion, and the cyan curve (as we will soon see) gives the limits of achievability.

- If we have $D > p$, then random noise will have that distortion, so we can just decode noise and achieve a rate of zero.
Binary Source $R(D)$, $X \sim \text{Bernoulli}(p)$ r.v.

- When $D = 0$, minimum rate is the entropy, and can’t compress below the entropy with zero distortion.
- As $D \uparrow$, we can “compress” more, below the entropy, but we suffer some distortion, and the cyan curve (as we will soon see) gives the limits of achievability.
Binary Source $R(D)$, $X \sim \text{Bernoulli}(p)$ r.v.

- When $D = 0$, minimum rate is the entropy, and can’t compress below the entropy with zero distortion.
- As $D \uparrow$, we can “compress” more, below the entropy, but we suffer some distortion, and the cyan curve (as we will soon see) gives the limits of achievability.
- If we have $D > p$, then random noise will have that distortion, so we can just decode noise and achieve a rate of zero.