Table of contents

Week 1 (9/14/2009) ........................................ 2
Course Overview ........................................ 2

Week 2 (9/21/2009) ......................................... 6
Jensen’s Inequality ........................................ 12

Week 3 (9/28/2009) ......................................... 13
Data Processing Inequality ................................ 17

Week 4 (10/5/2009) ......................................... 20
Asymptotic Equipartition Property ....................... 21

Week 5 (10/19/2009) ...................................... 25
Entropy Rates for Stochastic Processes ................... 25
Data Compression ........................................ 28

Week 6 (10/26/2009) ...................................... 29
Optimal Codes ........................................... 30
Huffman Codes ........................................... 32

Week 7 (11/9/2009) ...................................... 35
Other Codes with Special Properties ..................... 35
Channel Capacity ......................................... 36

Week 8 (11/16/2009) .................................... 41
Channel Coding Theorem ................................ 41

Week 9 (11/23/2009) .................................... 44
Differential Entropy ...................................... 47

Week 10 (11/30/2009) ................................... 49
Gaussian Channels ...................................... 50
Parallel Gaussian Channels ............................. 52
Non-white Gaussian noise channels ..................... 53

Week 11 (12/7/2009) ................................... 53
Rate Distortion Theory .................................. 53
Binary Source / Hamming Distribution ................ 55
Gaussian Source with Squared-Error Distortion .... 56
Independent Gaussian Random Variables with Single-Letter Distortion 57

Note: This set of notes follows the lecture, which references the text Elements of Information Theory by Cover, Thomas, Second Edition.
Course Overview

Communications-centric Motivation: Communication over noisy channels.

![Diagram of communication process]

Examples: Email over internet, Voice over phone network, Data over USB storage.

In these examples, the source would be the voice, email text, and data, the channel would be the (wired or wireless) phone network, internet, and storage device.

Purpose: Reproduce source data at the destination (with presence of noise in channel)

- Channel can introduce noise (static), interference (multiple copies at destination arriving at different times from radio waves bouncing on buildings, etc), or distortion (amplitude, phase, modulation)

- Sources have redundancy, e.g. the english language, can reconstruct words from parts: “th_t” → “that”, and images, where individual pixel differences are not noticed by the human eye. Some sources can tolerate distortion (images).

A typical example of an encoder is broken into three components:

From Source → Source Encoder → Channel Encoder → Modulator → Channel

- The **source encoder** outputs a “source codeword”, an efficient representation of the source, removing redundancies. The output is taken to be a sequence of bits (binary)

- The **channel encoder** outputs a “channel codeword”, introducing “controlled” redundancy to tolerate errors that may be introduced by the channel. The controlled redundancy is taken into account while decoding. Note the distinction from redundancies present in the source, where the redundancies may be arbitrary (many pixels in an image). The codeword is designed by examining channel properties. The simplest channel encoder involves just repeating the source codeword.

- The **modulator** converts the bits of the channel codeword into waveforms for transmission over the channel. They may be modulated in amplitude or frequency, e.g. radio broadcasts.

The corresponding decoder can also be broken into three components that invert the operations of the source encoder above:

From Channel → Demodulator → Channel Decoder → Source Decoder → Destination

We call the output of the demodulator the “received word” and the output of the channel decoder the “estimated source codeword”.

**Remark.** Later we will see that separating the source and channel encoder/decoder as above does not cause a loss in performance under certain conditions. This allows us to concentrate on each component independently of the other components. Called the **Joint Source Channel Coding Theorem**.
Information Theory addresses the following questions:

- Given a source, how much can I compress the data? Are there any limits?
- Given a channel, how noisy can the channel be, or how much redundancy is necessary to minimize error in decoding? What is the maximum rate of communication?

Information Theory interacts with many other fields as well, probability/statistics, computer science, economics, quantum theory, etc. See book for details.

There are two fundamental theorems by Shannon that address these questions. First we define a few quantities.

Typically, the source data is unknown, and we can treat the input data as a random variable, for instance, flipping coins and sending that as input across the channel to test the communication scheme.

We define the **entropy** of a random variable $X$, taking values in the alphabet $\mathcal{X}$ as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x)$$

The base 2 logarithm measures the entropy in *bits*. The intuition is that entropy describes the “compressibility” of the source.

**Example 1.** Let $X \sim \text{unif}\{1, \ldots, 16\}$. Note we need 4 bits to represent the values of $X$ intuitively. The entropy is

$$H(X) = -\sum_{x=1}^{16} \frac{1}{16} \log_2 \frac{1}{16} = 4 \text{ bits}$$

**Example 2.** 8 horses in a race with winning probabilities $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64})$. Note

$$H(X) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{8} \log_2 \frac{1}{8} - \frac{1}{16} \log_2 \frac{1}{16} - \frac{4}{64} \log_2 \frac{1}{64} = 2 \text{ bits}$$

The naive encoding for outputting the winner of the race uses 8 bits, labelling the horses from 000 to 111 in binary. However, since some horses win more frequently, it makes sense to use shorter encoding words for more frequent winners. For instance, label the horses (0,10,110,11100,11101,111100,111101,111111). This particular encoding has the property that it is “prefix-free”, no codeword is a prefix of another codeword. This makes it easy to determine how many bits to read until we can determine which horse is being referenced. Now note some codewords are longer than 3 bits, but on the average, we see that the average description length is

$$\frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{8}(3) + \frac{1}{16}(4) + \frac{4}{64}(6) = 2 \text{ bits}$$

which is exactly the entropy.

Now we are ready to state (roughly) Shannon’s First Theorem:

**Theorem 3. (Source Code Theorem)** Roughly speaking, $H(X)$ is the minimum rate at which we can compress a random variable $X$ and recover it fully.

Intuition can be drawn from the previous two examples. We will visit this again in more detail later.
Now we define the **mutual information** between two random variables \((X, Y) \sim p(x, y)\) defined as

\[
I(X; Y) = \sum_{x,y} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}
\]

Later we will show that \(I(X; Y) = H(X) - H(X | Y)\), where \(H(X | Y)\) is to be defined later. Here \(H(X)\) is to be interpreted as the compressibility (as stated earlier), or the amount of uncertainty in \(X\). \(H(X | Y)\) is the amount of uncertainty in \(X\), knowing \(Y\), and finally \(I(X; Y)\) is interpreted as the reduction in uncertainty of \(X\) due to knowledge of \(Y\), or a measure of dependency between \(X\) and \(Y\). For instance, if \(X\) and \(Y\) are completely dependent (knowing \(Y\) gives all information about \(X\)), then \(H(X | Y) = 0\) (no uncertainty given \(Y\)), and so \(I(X; Y) = H(X)\) (a complete reduction of uncertainty). On the other hand, if \(X\) and \(Y\) are independent, then \(H(X | Y) = H(X)\) (still uncertain) and \(I(X; Y) = 0\) (no reduction of uncertainty).

Now we will characterize a communications channel as one for which the channel output \(Y\) depends probabilistic on the input \(X\). That is, the channel is characterized by the conditional probability \(P(Y | X)\). For a simple example, suppose the input is \(\{0, 1\}\), and the channel flips the input bit with probability \(\varepsilon\), then \(P(Y = 1 | X = 0) = \varepsilon, P(Y = 0 | X = 0) = 1 - \varepsilon\).

Define the **capacity** of the channel to be

\[
C = \max_{p(x)} I(X; Y)
\]

noting here that \(I(X; Y)\) depends on the joint \(p(x, y) = p(y | x)p(x)\), and thus depends on the input distribution as well as the channel. The maximization is taken over all possible input distributions.

We can now roughly state Shannon’s Second Theorem:

**Theorem 4. (Channel Coding Theorem)** *The maximum rate of information over a channel with arbitrarily low probability of error is given by its capacity \(C\).*

Here “rate” is “number of bits per transmission”. This means that as long as the transmission rate is below the capacity, the probability of error can be made arbitrarily small.

We can motivate the theorem above with a few examples:

**Example 5.** Consider a noiseless binary channel, where input is \(\{0, 1\}\) and the output is equal to the input. We would then expect to be able to send 1 bit of data in each transmission through the channel with no probability of error. Indeed if we compute

\[
I(X; Y) = H(X) - H(X | Y) = H(X)
\]

since knowing \(Y\) gives complete information about \(X\), and thus

\[
C = \max_{p(x)} H(X) = \max_{p} \left[ -p \log_2 p - (1 - p) \log_2 (1 - p) \right]
\]

taking derivative and setting to zero gives us

\[- \log p - 1 + \log(1 - p) + 1 = 0\]

and so \(p = 1 - p\) and \(p = \frac{1}{2}\). Then

\[
C = - \log_2 \frac{1}{2} = 1 \text{ bit}
\]
as expected.

**Example 6.** Consider a noisy 4 symbol channel, where the input and output \( X, Y \) take values in \( \{0, 1, 2, 3\} \) and

\[
P(Y = j | X = j) = P(Y = j + 1 | X = j) = \frac{1}{2}
\]

where \( j + 1 \) is taken modulo 4. That is, a given input is either preserved or incremented upon output with equal probability. To make use of this noisy channel, we note that we do not have to use all the inputs when sending information. If we restrict the input to only 0 or 2, then we note that 0 is sent to 0 or 1, and 2 is sent to 2 or 3, and we can easily deduce from the output whether we sent a 0 or a 2. This allows us to send one bit of information through the channel with zero probability of error. It turns out that this is the best strategy. Given an input distribution \( p_X(x) \), we compute the joint distribution:

\[
p(x, y) = \begin{cases} \frac{1}{2} p_X(x) & y \in \{x, x+1\} \\ 0 & \text{else} \end{cases}
\]

Note that the marginal distribution of the output is

\[
p_Y(y) = \frac{1}{2}(p_X(y) + p_X(y - 1))
\]

Applying the definition of conditional entropy, we have

\[
H(Y | X) = \sum_x p(x) H\left(\frac{1}{2}\right) = 1
\]

Then the mutual information is

\[
I(X; Y) = H(Y) - 1
\]

The entropy of \( Y \) is

\[
H(Y) = -\sum_{y=0}^{3} p_Y(y) \log_2 p_Y(y)
\]

Since given \( p_Y(y) \), we can find \( p_X \) corresponding to \( p_Y \), it suffices to maximize \( I(X; Y) \) over possibilities for \( p_Y \). Letting \( p_Y(0) = p, p_Y(1) = q, p_Y(2) = r, p_Y(3) = 1 - p - q - r \), we have the maximization problem

\[
\max_{p,q,r} -1 - p \log p - q \log q - r \log r - (1 - p - q - r) \log(1 - p - q - r)
\]

Taking the gradient and setting it to zero, we have the equations

\[
- \log p - 1 + \log(1 - p - q - r) + 1 = 0 \\
- \log q - 1 + \log(1 - p - q - r) + 1 = 0 \\
- \log r - 1 + \log(1 - p - q - r) + 1 = 0
\]

which reduces to

\[
p = 1 - p - q - r \\
q = 1 - p - q - r \\
r = 1 - q - q - r
\]
so that \( p = q = r = \frac{1}{4} \). Plugging this into the above gives

\[
C = -1 - 4 \left[ \frac{1}{4} \log_{\frac{1}{4}} \frac{1}{4} \right] = 1 \text{ bit}
\]
as expected.

**Remark.** Above, to illustrate how to use the noisy 4-symbol channel, suppose the input string is 0101110. If the 4-symbol channel has no noise, then we could send 2 bits of information per transmission by using the following scheme:

Block the inputs two bits at a time: 01, 01, 11, 10, and send the messages 1, 1, 3, 2, across the channel.

Then the channel decoder inverts back to 01, 01, 11, 10 and reproduces the input string.

However, since the channel is noisy, we cannot do this without incurring some probability of error. Thus instead, we decide to send one bit at a time, and so we send 0, 2, 0, ..., and the output of the channel is 0 or 1, 0 or 2, which we can easily invert to recover the input.

Furthermore, by the theorem, 1 bit per transmission is the capacity of the channel, so this is the best scheme for this channel. An additional bonus is that we can even operate at capacity with zero probability of error!

In general, we can compute the capacity of a channel, but it is hard to find a nice scheme. For instance, consider the binary symmetric channel, with inputs \{0, 1\} where the output is flipped with probability \( \varepsilon \). Note that sending one bit through this channel has a probability of error \( \varepsilon \). One possibility for using the channel involves sending the same bit repeatedly through the channel and taking the majority in the channel decoding. For instance, if we send the same bit three times, the probability of error is

\[
P(2 \text{ or } 3 \text{ flips}) = \varepsilon^3 + 3\varepsilon^2(1 - \varepsilon) = 3\varepsilon^2 - 2\varepsilon^3 = O(\varepsilon^2)
\]

whereas the data rate becomes \( \frac{1}{3} \) bits per transmission (we needed to send 3 bits to send 1 bit of information). To lower the probability of error we would then need to send more bits. If we send \( n \) bits, then the data rate is \( \frac{1}{n} \) and the probability of error is

\[
P(n/2 \text{ flips or more}) = \sum_{j=n/2}^{n} \binom{n}{j} \varepsilon^j (1 - \varepsilon)^{n-j}
\]

We can compute the capacity of the channel directly in this case as well. Note that

\[
H(Y \mid X) = \sum_{x=0}^{1} p_X(x) H(\varepsilon) = H(\varepsilon)
\]

and \( I(X; Y) = H(X) - H(\varepsilon) = -H(\varepsilon) - p \log p - (1 - p) \log (1 - p) \), which is maximized at \( p = \frac{1}{2} \), corresponding to a channel capacity of \( 1 - H(\varepsilon) \). Note that the proposed repetition codes above do not achieve this capacity for arbitrarily low probability of error. Later we will show how to find such codes.

**Week 2**

(9/21/2009)

Today: Following Chapter 2, defining and examining properties of basic information-theoretic quantities.

Entropy is for a discrete random variable \( X \), taking values in a countable alphabet \( \mathcal{X} \) with probability mass function \( p(x) = P(X = x), x \in \mathcal{X} \). Later we will discuss a notion of entropy for the continuous case.
**Definition.** The **entropy** of \( X \sim p(x) \) is defined to be

\[
H(X) = - \sum_{x \in X} p(x) \log p(x) = E_{p(x)}(-\log p(x))
\]

If we use the base 2 logarithm, then the measure is in bits. If we use the natural logarithm, the measure is in “nats”. Also, by convention (continuity), we take \( 0 \log 0 = 0 \).

Note that \( H(X) \) is “independent on labeling.” It only depends on the masses in the pmf \( p(x) \).

Also, \( H(X) \geq 0 \), which follows from the fact that \(-\log(p(x)) \geq 0 \) for \( 0 \leq p(x) \leq 1 \).

**Example 7.** The entropy of a Bernoulli\( (p) \) random variable, 

\[
X = \begin{cases} 
0 & \text{w.p. } p \\
1 & \text{w.p. } 1-p 
\end{cases}
\]

is \( H(X) = -p \log p - (1-p) \log(1-p) \).

We will often use the notation 

\[
H(p) := -p \log p - (1-p) \log(1-p)
\]

(binary entropy function) for convenience. Note the graph of \( H(p) \):

In particular, we note the following observations:

- \( H(p) \) is concave. Later we will show that entropy in general is a concave function of the underlying probability distribution.

- \( H(p) = 0 \) for \( p = 0, 1 \). This reinforces the intuition that entropy is a measure of randomness, since in the case of no randomness \( p = 0, 1 \), there is no entropy.

- \( H(p) \) is maximized when \( p = 1/2 \). Likewise, this is the situation where a binary distribution is the most random.
• $H(p) = H(1 - p)$.

**Example 8.** Let

$$X = \begin{cases} 
1 \text{ w.p. } \frac{1}{4} \\
2 \text{ w.p. } \frac{1}{4} \\
3 \text{ w.p. } \frac{1}{2}
\end{cases}$$

In this case $H(X) = -\frac{1}{2}\log\frac{1}{2} - 2 \cdot \frac{1}{4}\log\frac{1}{4} = \frac{1}{2} + 1 = \frac{3}{2}$ bits. Last time we viewed entropy as the average length of a binary code needed to encode $X$. We can interpret such binary codes as a sequence of yes/no questions that are needed to determine the value of $X$. The entropy is then the average number of questions needed using the optimal strategy for determining $X$. In this case, we can use the following simple strategy, taking advantage of the fact that $X = 1$ occurs more often than the other cases:

• Question 1: Is $X = 1$?

• Question 2: Is $X = 2$?

In this case, the number of questions needed is 1 if $X = 1$, which happens with probability $\frac{1}{2}$ and 2 otherwise, with probability $\frac{1}{2}$, so the expected number of questions needed is $1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{2} = \frac{3}{2}$, which coincides with the entropy.

**Example 9.** Let $X_1, ..., X_n$ be i.i.d Bernoulli($p$) distributed ($0$ with probability $p$, $1$ with probability $1 - p$). Then the joint distribution is given by

$$p(x_1, ..., x_n) = p^{\# \text{ of } 0's}(1 - p)^{\# \text{ of } 1's} = (1 - p)^{\sum_i X_i} p^n - \sum_i X_i$$

Observe that if $n$ is large, then by the law of large numbers $\sum_{i=1}^n X_i \approx n(1 - p)$ with high probability, so that

$$p(x_1, ..., x_n) \approx (1 - p)^n (1 - p) p^n = 2^{-nH(p)}$$

The interpretation is that for $n$ large, the probability of the joint is concentrated in a set of size $2^n H(p)$ on which it is approximately uniformly distributed. In this case we can exploit this compression and use $n H(p)$ bits to distinguish sequences in this set (per random variable we are using $H(p)$ bits. For this reason $2^n H(p)$ is the “effective alphabet size” of $(X_1, ..., X_n)$.

**Definition.** The **joint entropy** of $(X, Y) \sim p(x, y)$ is given by

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log p(x, y) = E_{p(x, y)}(-\log p(x, y))$$

**Definition.** The **conditional entropy** of $Y$ given $X$ is

$$H(Y \mid X) = \sum_{x \in X} p(x) H(Y \mid X = x)$$

$$= -\sum_{x \in X} \sum_{y \in Y} p(x) p(y \mid x) \log p(y \mid x)$$

$$= -\sum_{x, y} p(x, y) \log p(y \mid x)$$

$$= E_{p(x, y)}(-\log p(y \mid x))$$

We expect that we can write the joint entropy in terms of the individual entropies, and this is indeed the case, in a result we call the **Chain Rule for Entropy:**

"
Proposition 10. (Chain Rule for Entropy) For two random variables $X, Y$,

$$H(X, Y) = H(X) + H(Y | X) = H(Y) + H(X | Y)$$

For multiple random variables $X_1, ..., X_n$, we have that

$$H(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i | X_{i-1})$$

Proof. We just perform a simple computation

$$H(X, Y) = -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x, y) \log p(x, y)$$

$$= -\sum_{x\in\mathcal{X}} \log p(x) \sum_{y\in\mathcal{Y}} p(x, y) - \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x, y) \log p(y | x)$$

$$= H(X) + H(Y | X)$$

by symmetry (switching the roles of $X$ and $Y$) we get the other equality. For $n$ random variables $X_1, ..., X_n$ we proceed by induction. We have just proved the result for two random variables. Assume it is true for $n-1$ random variables. Then

$$H(X_1, ..., X_n) = H(X_n | X_1, ..., X_{n-1}) + \sum_{i=1}^{n-1} H(X_i | X_{i-1})$$

where we have used the usual chain rule and the induction hypothesis.

We also immediately get the chain rule for conditional entropy:

Corollary 11. (Chain Rule for Conditional Entropy) For three random variables $X, Y, Z$, we have

$$H(X, Y | Z) = H(Y | Z) + H(X | Y, Z) = H(X | Z) + H(Y | X, Z)$$

and for $n$ random variables $X_1, ..., X_n$ and a random variable $Z$, we have

$$H(X_1, ..., X_n | Z) = \sum_{i=1}^{n} H(X_i | Z, X_1, ..., X_{i-1})$$

Proof. This follows from the chain rule for entropy given $Z = z$, and averaging over $z$ gives the result.

Example 12. Consider the following joint distribution for $X, Y$ in tabular form with the marginals:

<table>
<thead>
<tr>
<th>$y \backslash x$</th>
<th>1</th>
<th>2</th>
<th>$p_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3/4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1/4</td>
</tr>
<tr>
<td>$p_x$</td>
<td>1/2</td>
<td>1/2</td>
<td></td>
</tr>
</tbody>
</table>

$$p(x, y) = \frac{1}{2}$$
Note that \( H(X) = 1 \) bit and \( H(Y) = H\left(\frac{1}{4}\right) = -\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4} = 2 - \frac{3}{4}\log 3 \) bits. The conditional entropies are computed as follows:

\[
H(X) = H\left(\frac{1}{2}\right) = 1 \text{ bit}
\]
\[
H(Y) = H\left(\frac{1}{4}\right) = -\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4} = 2 - \frac{3}{4}\log 3 \text{ bits}
\]
\[
H(Y | X) = P(X = 1) \cdot H(Y | X = 1) + P(X = 2) \cdot H(Y | X = 2) = \frac{1}{2} \cdot H\left(\frac{1}{2}\right) = \frac{1}{2} \text{ bits}
\]
\[
H(X | Y) = P(Y = 1) \cdot H(X | Y = 1) + P(Y = 2) \cdot H(X | Y = 2) = \frac{3}{4} \cdot H\left(\frac{1}{3}\right) = \frac{3}{4} \left[ -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3} \right] = \frac{3}{4}\log 3 - \frac{1}{2} \text{ bits}
\]
\[
H(X, Y) = -2 \cdot \frac{1}{4}\log\frac{1}{4} - \frac{1}{2} \log\frac{1}{2} = \frac{3}{2} \text{ bits}
\]

Above we note that \( H(Y | X = 2) = H(X | Y = 2) = H(0) = 0 \) since in each case the distribution becomes deterministic. Note that in general \( H(Y | X) \neq H(X | Y) \) and in this example \( H(X | Y) \leq H(X) \), \( H(Y | X) \leq H(Y) \). We will show this in general later. Also, \( H(Y | X = x) \) has no relation to \( H(Y) \), can be larger or smaller.

The conditional entropy is related to a more general quantity called relative entropy, which we now define.

**Definition.** For two pmf’s \( p(x) \), \( q(x) \) on \( X \) the **relative entropy** (Kullback-Leibler distance) is defined by

\[
D(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}
\]

using the convention that \( 0 \log \frac{0}{q} = 0 \log \frac{0}{0} = 0 \) and \( p \log \frac{p}{0} = \infty \). In particular, the distance is allowed to be infinite. Note that the “distance” is not symmetric, so \( D(p \parallel q) \neq D(q \parallel p) \) in general, and so it is not a metric. However it does satisfy positivity. We will show that \( D(p \parallel q) \geq 0 \) with equality when \( p = q \). Roughly speaking this describes the loss from using \( q \) instead of \( p \) for compression.

**Definition.** The **mutual information** between two random variables \( X, Y \sim p(x, y) \) is

\[
I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} = d(p(x, y) \parallel p(x)p(y))
\]
Relationship between Entropy and Mutual Information

\[
I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\
= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)} - \sum_y \left( \log p(y) \sum_x p(x,y) \right) \\
= -H(Y|X) + H(Y) \\
= H(Y) - H(Y|X)
\]

and thus we conclude by symmetry that

\[
I(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)
\]

**Remark.** As we said before, this is the reduction in entropy of \(X\) given knowledge of \(Y\), and we will show later that \(I(X;Y) \geq 0\) with equality if and only if \(X, Y\) are independent.

Also, note that \(I(X;X) = H(X) - H(X|X) = H(X)\), and sometimes \(H(X)\) is referred to as “self-information”.

The relationships between the entropies and the mutual information can be represented by a Venn Diagram. \(H(X), H(Y)\) represent the two circles, the intersection is \(I(X;Y)\), the union is \(H(X,Y)\), and the differences \(X - Y\) and \(Y - X\) represent \(H(X|Y)\) and \(H(Y|X)\), respectively.

We will also establish chain rules for mutual information and relative entropy. First we define the mutual information conditioned on a random variable \(Z\):

**Definition.** The conditional mutual entropy of \(X, Y\) given \(Z\) for \((X, Y, Z) \sim p(x, y, z)\) is

\[
I(X;Y|Z) = H(X|Z) - H(X|Y, Z) = E_{p(x,y,z)} \left( \frac{\log p(x,y|z)}{p(x|z)p(y|z)} \right)
\]

noting that this is also \(I(X;Y|Z = z)\) averaged over \(z\).

The chain rule for mutual information is then

\[
I(X_1, ..., X_n; Y) = \sum_{i=1}^{n} I(X_i; Y|X_1, ..., X_{i-1})
\]

The proof just uses the chain rules for entropy and conditional entropy:

\[
I(X_1, ..., X_n; Y) = H(X_1, ..., X_n) - H(X_1, ..., X_n|Y) \\
= \sum_{i=1}^{n} H(X_i | X_1, ..., X_{i-1}) - H(X_i | Y, X_1, ..., X_{i-1}) \\
= \sum_{i=1}^{n} I(X_i; Y|X_1, ..., X_{i-1})
\]

Now we define the relative entropy conditioned on a random variable \(Z\), and prove the corresponding chain rule:
**Definition.** Let $X$ be a random variable with distribution given by pmf $p(x)$. Let $p(y|x), q(y|x)$ be two conditional pmfs. The **conditional relative entropy** of $p(y|x)$ and $q(y|x)$ is then defined to be

$$D(p(y|x) \| q(y|x)) = \sum_x p(x) D(p(y|X=x) \| q(y|X=x))$$

$$= \sum_{x,y} p(x)p(y|x) \log \frac{p(y|x)}{q(y|x)}$$

$$= E_{p(x,y)} \left( \log \frac{p(Y|X)}{q(Y|X)} \right)$$

Then the chain rule for relative entropy is

$$D(p(x,y) \| q(x,y)) = E_{p(x,y)} \left( \log \frac{p(X,Y)}{q(X,Y)} \right)$$

$$= E_{p(x,y)} \left( \log \frac{p(Y|X)}{q(Y|X)} + \log \frac{p(X)}{q(X)} \right)$$

$$= E_{p(x,y)} \left( \log \frac{p(Y|X)}{q(Y|X)} \right) + E_{p(x)} \left( \log \frac{p(X)}{q(X)} \right)$$

$$= D(p(y|x) \| q(y|x)) + D(p(x) \| q(x))$$

The various chain rules will help simplify calculations later in the course.

**Jensen’s Inequality**

Convexity and Jensen’s Inequality will be very useful for us later. First a definition:

**Definition.** A function $f$ is said to be **convex** on $[a, b]$ if for all $[a', b'] \subset [a, b]$ and $0 \leq \lambda \leq 1$ we have

$$f(\lambda a' + (1-\lambda)b') \leq \lambda f(a') + (1-\lambda)f(b')$$

If the inequality is strict, then we say $f$ is **strictly convex**.

Note that if we have a binary random variable $X \sim \text{Bernoulli}(\lambda)$, then we can restate convexity as

$$f(E_{p(x)}(X)) \leq E_{p(x)}(f(X))$$

We can generalize this to more complicated random variables, which leads to Jensen’s inequality:

**Theorem 13. (Jensen’s Inequality)** If $f$ is convex, and $X$ is a random variable, then

$$f(E(X)) \leq E(f(X))$$

and if $f$ is strictly convex, then equality above implies $X = E(X)$ with probability 1, i.e. $X$ is constant.

**Proof.** There are many proofs, but for finite random variables, we can show this by induction and the fact that it is already true by definition for binary random variables. To extend the proof to the general case, can use the idea of supporting lines: we note that

$$f(x) + m(y-x) \leq f(y)$$
for all $x$ and some $m$ dependent on $x$. Set $x = E(X)$ to get

$$f(E(X)) + m(y - E(X)) \leq f(y)$$

and integrating in $y$ gives the result

$$f(E(X)) \leq E(f(X))$$

If we have equality, then

$$f(y) - f(E(X)) + m(y - E(X)) = 0$$

or

$$f(y) = f(E(X)) + m(y - E(X))$$

for a.e. $y$. However, by strict convexity this can occur only if $y = E(X)$ a.e. □

**Week 3**

*(9/28/2009)*

We continue with establishing information theoretic quantities and inequalities. We start with applications of Jensen’s inequality.

**Theorem 14. (Information Inequality)** For any $p(x), q(x)$ pmfs, we have that

$$D(p \parallel q) \geq 0$$

with equality if and only if $p = q$.

**Proof.** First let $A = \{x: p(x) > 0\}$, and compute

$$D(p \parallel q) = \sum_{x \in A} p(x) \log \frac{p(x)}{q(x)}$$

$$= \sum_{x \in A} p(x) \left( - \log \frac{q(x)}{p(x)} \right)$$

$$\geq - \log \left( \sum_{x \in A} p(x) \cdot \frac{q(x)}{p(x)} \right)$$

$$\geq - \log \left( \sum_{x \in \mathcal{X}} q(x) \right)$$

$$= 0$$

Examining the proof above, equality occurs if and only if

$$\sum_{x \in A} q(x) = \sum_{x \in \mathcal{X}} q(x)$$

and equality is achieved in the application of Jensen’s inequality. The first condition leads to $q(x) = 0$ whenever $p(x) = 0$. 

13
Equality in the application of Jensen’s inequality with strict convexity of $-\log(\cdot)$ implies that on $A,$

$$
\frac{q(x)}{p(x)} = \sum_{x \in A} q(x) = 1
$$

and therefore $q = p.$

This has quite a few implications:

**Corollary 15.** $I(X;Y) \geq 0$ if and only if $X,Y$ are independent.

**Proof.** Noting $I(X;Y) = D(p(x,y)\|p(x)p(y)),$ we have that $I(X;Y) \geq 0$ with equality if and only if $p(x, y) = p(x)p(y),$ i.e. $X,Y$ are independent.

**Corollary 16.** If $|\mathcal{X}| < \infty,$ then we have the following useful upper bound:

$$
H(X) \leq \log |\mathcal{X}|
$$

**Proof.** Let $u(x) \sim \text{unif}(\mathcal{X}),$ i.e. $u(x) = \frac{1}{|\mathcal{X}|}$ for $x \in \mathcal{X}.$ Let $X \sim p(x)$ be any random variable taking values in $\mathcal{X}.$ Then

$$
0 \leq D(p\|u) = \sum_x p(x) \log \frac{p(x)}{u(x)} = -H(X) + \sum_x p(x) \log |\mathcal{X}|
$$

from which the result follows. Equality holds if and only if $p = u.$

**Theorem 17.** (Conditioning Reduces Entropy)

$$
H(X|Y) \leq H(X)
$$

with equality if and only if $X,Y$ are independent.

**Proof.** Note $I(X;Y) = H(X) - H(X|Y) \geq 0,$ where equality holds if and only if $X,Y$ are independent.

**Remark.** Do not forget that $H(X|Y = y)$ is not necessarily $\leq H(X)$.

The conditional entropy inequality allows us to show the concavity of the entropy function:

**Corollary 18.** Suppose $X \sim p(x)$ is a r.v. Then

$$
H(p) := H(X)
$$

is a concave function of $p(x),$ i.e.

$$
H(\lambda p_1 + (1-\lambda) p_2) \geq \lambda H(p_1) + (1-\lambda) H(p_2)
$$
Note that $p_i$ are pmf’s, so $H(p_i)$ is not the binary entropy function.

**Proof.** Let $X_i \sim p_i(x)$ for $i = 1, 2$ defined on the same alphabet $\mathcal{X}$. Let

$$\theta = \begin{cases} 1 & \text{w.p. } \lambda \\ 2 & \text{w.p. } (1 - \lambda) \end{cases}$$

and define

$$Z = X_\theta = \begin{cases} X_1 & \text{w.p. } \lambda \\ X_2 & \text{w.p. } (1 - \lambda) \end{cases}$$

Now note that $H(Z|\theta) = \lambda H(p_1) + (1 - \lambda)H(p_2)$, and by the conditional entropy inequality,

$$\lambda H(p_1) + (1 - \lambda)H(p_2) = H(Z|\theta) \leq H(Z) = H(\lambda p_1 + (1 - \lambda)p_2)$$

Note that $P(Z = a) = \lambda p_1(a) + (1 - \lambda)p_2(a)$. \hfill \Box

**Corollary 19.** $H(X_1, \ldots, X_n) \leq \sum_{i=1}^n H(X_i)$

**Proof.** By the chain rule,

$$H(X_1, \ldots, X_n) = \sum_{i=1}^n H(X_i | X_1, \ldots, X_{i-1}) \leq \sum_{i=1}^n H(X_i)$$

using the conditional entropy inequality. Equality holds if and only if $X_1, \ldots, X_n$ are mutually independent, following from $X_i$ being independent from the joint $(X_1, \ldots, X_{i-1})$. \hfill \Box

Another application of Jensen’s inequality will give us another set of useful inequalities.

**Theorem 20. (Log Sum Inequality)** Let $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ be nonnegative numbers. Then

$$\sum_{i=1}^n a_i \log \frac{a_i}{b_i} \geq \left( \sum_{i=1}^n a_i \right) \left( \log \sum_{i=1}^n \frac{a_i}{b_i} \right)$$

**Proof.** This follows from the convexity of $t \log t$ (with second derivative $\frac{1}{t}$). Then defining

$$X = \frac{a_i}{b_i} \text{ w.p. } b_i \sum_i b_i$$

we have that

$$\frac{1}{b_i} \sum_{i=1}^n a_i \log \frac{a_i}{b_i} = E(X \log X) \geq E(X) \log E(X) = \sum_{i=1}^n \frac{a_i}{b_i} \log \left( \sum_{i=1}^n \frac{a_i}{b_i} \right)$$

and multiplying by $\sum b_i$ gives the result. Note that equality holds if and only if $\frac{a_n}{b_n} = \sum \frac{a_i}{b_i}$, or $a_i = C b_i$. \hfill \Box
Then we have the following corollaries:

**Corollary 21.** $D(p\|q)$ is convex in the pair $(p,q)$, i.e.

$$D(\lambda p_1 + (1-\lambda)p_2 \| \lambda q_1 + (1-\lambda)q_2) \leq \lambda D(p_1 \| q_1) + (1-\lambda)D(p_2 \| q_2)$$

**Proof.** Expanding the inequality above, we want to show that

$$\sum_x (\lambda p_1(x) + (1-\lambda)p_2(x)) \log \frac{\lambda p_1(x) + (1-\lambda)p_2(x)}{\lambda q_1(x) + (1-\lambda)q_2(x)} \leq \lambda \sum_x p_1(x) \log \frac{p_1(x)}{q_1(x)} + (1-\lambda) \sum_x p_2(x) \log \frac{p_2(x)}{q_2(x)}$$

Then for a fixed $x$, it suffices to show that

$$\sum_{i=1,2} \lambda_i p_i \log \frac{\lambda_i p_i}{\lambda_i q_i} \geq \left( \sum_{i=1,2} \lambda_i p_i \right) \left( \log \sum_{i=1,2} \frac{\lambda_i p_i}{\lambda_i q_i} \right)$$

noting $\lambda_1 = \lambda, \lambda_2 = (1-\lambda)$, which is exactly the log sum inequality. Summing over $x$ gives the result.  

**Theorem 22.** Let $(X,Y) \sim p(x,y) = p(x)p(y|x)$. Then

1. $I(X;Y)$ is a concave function of $p(x)$ for a fixed conditional distribution $p(y|x)$.
2. $I(X;Y)$ is a convex function of $p(y|x)$ for a fixed marginal distribution $p(x)$.

**Remark.** This will be useful in studying the source-channel coding theorems, since $p(y|x)$ captures information about the channel. Concavity implies that local maxima are also global maxima, and when computing the capacity of a channel (maximizing mutual information between input and output of channel) will allow us to look for local maxima. Likewise, convexity implies that local minima are global minima, and when we study lossy compression we will want to compute given a fixed input distribution, what is the worst case scenario, which involves minimizing the mutual information.

**Proof.** For (1), note that $I(X;Y) = H(Y) - H(Y \mid X)$. Since $p(y) = \sum_x p(x)p(y \mid x)$ is a linear function of $p(x)$ for a fixed $p(y \mid x)$, and $H(Y)$ is a concave function of $p(y)$, we see that $H(Y)$ is a concave function of $p(x)$, since a composition of a concave function $\phi$ with a linear function $T$ is concave:

$$\phi(T(\lambda x_1 + (1-\lambda)x_2)) = \phi(\lambda T x_1 + (1-\lambda)Tx_2) \geq \lambda \phi(Tx_1) + (1-\lambda)\phi(Tx_2)$$

As for the second term, note that

$$H(Y \mid X) = \sum_x p(x) H(Y \mid X = x)$$

where $H(Y \mid X = x) = -\sum_y p(y \mid x) \log p(y \mid x)$ which is fixed for a fixed $p(y \mid x)$, and thus $H(Y \mid X)$ is a linear function of $p(x)$, which in particular is concave. Thus $I(X;Y)$ is a sum of two concave functions which is also concave:

$$(\phi_1 + \phi_2)(\lambda x + (1-\lambda)y) \leq \sum_i (\lambda \phi_i(x) + (1-\lambda)\phi_i(y)) = \lambda (\phi_1 + \phi_2)(x) + (1-\lambda)(\phi_1 + \phi_2)(y)$$

This proves (1).
To prove (2), consider $p_1(y|x)$, $p_2(y|x)$, two conditional distributions given $p(x)$, with corresponding joints $p_1(x,y), p_2(x,y)$ and output marginals $p_1(y), p_2(y)$. Then also consider

$$p_\lambda(y|x) = \lambda p_1(y|x) + (1-\lambda) p_2(y|x)$$

with corresponding joint $p_\lambda(x,y) = \lambda p_1(x,y) + (1-\lambda) p_2(x,y)$ and marginals $p_\lambda(y) = \lambda p_1(y) + (1-\lambda) p_2(y)$. Then the mutual information $I_\lambda(X;Y)$ using $p_\lambda$ is

$$I_\lambda(X:Y) = D(p_\lambda(x,y) \| p(x)p_\lambda(y))$$

$$\leq \lambda D(p_1(x,y) \| p(x)p_1(y)) + (1-\lambda) D(p_2(x,y) \| p(x)p_2(y))$$

$$= \lambda I_1(X:Y) + (1-\lambda) I_2(X:Y)$$

which shows that $I_\lambda$ is a convex function of $p(y|x)$ for a fixed $p(x)$. \qed

**Data Processing Inequality**

We now turn to proving a result about Markov chains, which we define first:

**Definition 23.** The random variables $X, Y, Z$ form a Markov chain, denoted $X \rightarrow Y \rightarrow Z$ if the joint factors in the following way:

$$p(x, y, z) = p(x)p(y|x)p(z|y)$$

for all $x, y, z$.

**Properties.**

1. $X \rightarrow Y \rightarrow Z$ if and only if $X, Z$ are conditionally independent given $Y$, i.e.

$$p(x,z|y) = p(x|y)p(z|y)$$

**Proof.** First assume $X \rightarrow Y \rightarrow Z$. Then

$$p(x,z|y) = \frac{p(x,y,z)}{p(y)} = \frac{p(x,y)p(z|y)}{p(y)} = p(x|y)p(z|y)$$

Conversely, suppose $p(x,z|y) = p(x|y)p(z|y)$. Then

$$p(x,y,z) = p(y)p(x,z|y) = p(y)p(x|y)p(z|y) = p(x)p(y|x)p(z|y)$$

as desired. \qed

2. $X \rightarrow Y \rightarrow Z \implies Z \rightarrow Y \rightarrow X$, which follows from the previous property.

3. If $Z = f(Y)$, then $X \rightarrow Y \rightarrow Z$. This also follows from the first property, since given $Y, Z$ is independent of $X$:

$$p(z|x,y) = p(z|y) = \begin{cases} 1 & z = f(y) \\ 0 & \text{else} \end{cases}$$

Now we can state the theorem:

**Theorem 24. (Data Processing Inequality)** Suppose $X \rightarrow Y \rightarrow Z$. Then $I(X;Y) \geq I(X;Z)$. 

**Remark.** This means that along the chain, the dependency decreases. Alternatively, if we interpret $I(X; Y)$ as the amount of knowledge $Y$ gives about $X$, then any processing of $Y$ to get $Z$ may decrease the information that we can get about $X$.

**Proof.** Expand $I(X; Y, Z)$ by chain rule in two ways:

\[
I(X; Y, Z) = I(X; Y) + I(X; Z | Y)
\]

\[
I(X; Y, Z) = I(X; Z) + I(X; Y | Z)
\]

Note that $I(X; Z | Y) = 0$ since $X, Z$ are independent given $Y$. Also $I(X; Y | Z) \geq 0$, and thus

\[
I(X; Y) = I(X; Y, Z) = I(X; Z) + I(X; Y | Z) \geq I(X; Z)
\]

as desired. □

Now we illustrate some examples.

**Example 25. (Communications Example)** Modeling the process of sending a (random) message $W$ to some destination:

\[
W \rightarrow \text{Encoder} \rightarrow X \rightarrow \text{Channel} \rightarrow Y \rightarrow \text{Decoder} \rightarrow Z
\]

Note that $W \rightarrow X \rightarrow Y \rightarrow Z$ is a Markov chain, each only depends directly on the previous. The data processing inequality tells us that

\[
I(X; Y) \geq I(X; Z)
\]

Thus, if we are not careful, the decoder may decrease mutual information (and of course cannot add any information about $X$). Similarly,

\[
I(X; Y) \geq I(W; Z)
\]

**Example 26. (Statistics Examples)** In statistics, frequently we want to estimate some unknown distribution $X$ by making measurements (observations) and then coming up with some estimator:

\[
X \rightarrow \text{Measurement} \rightarrow Y \rightarrow \text{Estimator} \rightarrow Z
\]

For instance, $X$ could be the distribution of heights of a population (gaussian with unknown mean and standard deviation), then $Y$ would be some samples, and from samples we can come up with an estimator (say the sample mean / standard deviation). $X \rightarrow Y \rightarrow Z$, and by the data processing inequality we then note that the estimator may decrease information. An estimator that does not decrease information about $X$ is called a “sufficient statistic”.

Here is a probabilistic setup. Suppose we have a family of probability distributions \( \{p_\theta(x)\} \), and $\Theta$ is the parameter to be estimated. $X$ represents samples from some distribution $p_\theta$ in the family. Let $T(X)$ be any function of $X$ (for instance, the sample mean). Then $\Theta \rightarrow X \rightarrow T(X)$ form a Markov Chain. By the data processing inequality, $I(\Theta; T(X)) \leq I(\Theta; X)$.

**Definition 27.** $T(X)$ is a sufficient statistic for $\Theta$ if $\Theta \rightarrow T(X) \rightarrow X$ also forms a Markov Chain. Note that this implies $I(\Theta; T(X)) = I(\Theta, X)$. 

18
As an example, consider \( p_\theta(x) \sim \text{Bernoulli}(\theta) \). And consider \( X = (X_1, \ldots, X_n) \sim \text{i.i.d.} p_\theta(x) \). Then \( T(X) = \sum_{i=1}^n X_i \) is a sufficient statistic for \( \Theta \), i.e. given \( T(X) \), the distribution of \( X \) does not depend on \( \Theta \). Here \( T(X) \) counts the number of 1’s. Then the joint

\[
P(X = (x_1, \ldots, x_n) \mid T(X) = k, \Theta = \theta) = \begin{cases} \frac{1}{k} \sum_i x_i = k \\ 0 \text{ else} \end{cases}
\]

where we note that \( P(X = (x_1, \ldots, x_n) \land T(X) = k \mid \Theta = \theta) = \theta^k (1 - \theta)^{n-k} \) (if \( \sum x_i = k \)) and \( P(T(X) = k \mid \Theta = \theta) = \binom{n}{k} \theta^k (1 - \theta)^{n-k} \).

As a second example, consider \( f_\theta(x) \sim N(\theta, 1) \) and consider \( X = (X_1, \ldots, X_n) \sim \text{i.i.d.} f_\theta(x) \). Then \( T(X) = \frac{1}{n} \sum_{i=1}^n X_i \) is a sufficient statistic for \( \Theta \). Note that \( \sum_{i=1}^n X_i \sim N(n\theta, n) \) so that \( T(X) \sim N(\theta, 1/n) \). Consider the joint of \( (X_1, \ldots, X_n, T(X)) \), which is

\[
f(x_1, \ldots, x_n, \bar{x}) = f_\theta(x_1, \ldots, x_n) P(T(X) = \bar{x} \mid x_1, \ldots, x_n) = \begin{cases} \exp\left(\frac{1}{2} \sum_{i=1}^n |x_i - \theta|^2\right) \bar{x} = \frac{1}{n} \sum x_i \\ 0 \text{ else} \end{cases}
\]

Note the \( \theta \)-dependent part is \(- \sum_{i=1}^n x_i \theta \). Also,

\[
f(\bar{x}) = \exp\left(\frac{n}{2} |\bar{x} - \theta|^2\right)
\]

and if \( \bar{x} = \frac{1}{n} \sum x_i \), then the \( \theta \)-dependent part of \( f(\bar{x}) \) is \(- n \bar{x} \theta = - \sum_{i=1}^n x_i \theta \). Thus the \( \theta \)-dependent parts cancel in \( f(x_1, \ldots, x_n \mid \bar{x}, \Theta) \) so that given \( T(X) \), we have that \( X \) and \( \Theta \) are independent.

As a final example, if \( g_\theta \sim \text{Unif}(\theta, \theta + 1) \), and \( X = (X_1, \ldots, X_n) \sim \text{i.i.d.} g_\theta \), then a sufficient statistic for \( \theta \) is

\[
T(X_1, \ldots, X_n) = \left( \max_i X_i, \min_i X_i \right)
\]

In this case we note that

\[
P(x_1 \in I_1, \ldots, x_n \in I_n \mid \max_i X_i = M, \min_i X_i = m) = \prod_{i=1}^n |I_i \cap [m, M]|
\]

which does not depend on the parameter \( \theta \). Given the max and min, we know that the \( X_i \) are in between, so we do not need to know \( \theta \) to compute the conditional distribution.

Now consider the problem of estimating an unknown r.v. \( X \sim p(x) \) by observing \( Y \) with conditional distribution \( p(y \mid x) \). Use \( g(Y) = \hat{X} \) as the estimate for \( X \), and consider the error \( P_e = \Pr(\hat{X} \neq X) \). Fano’s inequality relates this error to the conditional entropy \( H(X \mid Y) \). This is intuitive, since the lower the entropy, the lower we expect the probability of error to be.

**Theorem 28. (Fano’s Inequality)** Suppose we have a setup as above with \( X \) taking values in the finite alphabet \( \mathcal{X} \). Then

\[
H(P_e) + P_e \log(|\mathcal{X}| - 1) \geq H(X \mid Y)
\]

where \( H(P_e) \) is the binary entropy function \( P_e \log P_e + (1 - P_e) \log (1 - P_e) \). Note that a weaker form of the inequality is (after using \( H(P_e) \leq 1 \) and \( \log(|\mathcal{X}| - 1) \leq \log(|\mathcal{X}|) \)):

\[
P_e \geq \frac{H(X \mid Y) - 1}{\log(|\mathcal{X}|)}
\]
Proof. Let $E = \begin{cases} 1 & X \neq \hat{X} \\ 0 & X = \hat{X} \end{cases}$ be the indicator of error, and note $E \sim \text{Bernoulli}(P_e)$. Examine $H(E, X | Y)$ using the chain rule: $H(E | Y) + H(X | E, Y) = H(E, X | Y) = H(X | Y) + H(E | X, Y)$. Note that $H(E | Y) \leq H(E) = H(P_e)$. Also,

$$H(X | E, Y) = P(E = 0)H(X | E = 0, Y) + P(E = 1)H(X | E = 1, Y)$$

where in the first component $H(X | E = 0, Y) = 0$ because knowing $Y$ and that there was no error gives us information that $X = g(Y)$, so it becomes deterministic. Also, given $E = 1, Y$, we know that $X$ takes values in $\mathcal{X} \setminus \{\hat{X}\}$, an alphabet of size $|\mathcal{X}| - 1$. Then using the estimate $H(X) \leq \log |\mathcal{X}|$, we have that

$$H(X | E, Y) \leq P_e \log(|\mathcal{X}| - 1)$$

and finally since $H(E | X, Y) = 0$ since knowing $X, Y$ allows us to determine whether there is an error,

$$H(X | Y) \leq P_e \log(|\mathcal{X}| - 1) + H(P_e)$$

which is the desired inequality. □

Remark. The theorem is also true with $g(Y)$ random. See 2.10.1 in book. In fact, we just make a slight adjustment. In the proof above we have actually showed the result with $H(X | \hat{X})$ instead of $H(X | Y)$. Then the rest is a consequence of the data processing inequality, since

$$I(X, \hat{X}) \leq I(X, Y) \implies H(X) - H(X | \hat{X}) \leq H(X) - H(X | Y) \implies H(X | Y) \leq H(X | \hat{X})$$

Also, the inequality may be sharp, there exist distributions for which equality is satisfied. See book.

**Week 4**

(10/5/2009)

Suppose we have $X, X'$ i.i.d with entropy $H(X)$. Note $P(X = X') = \sum_{x \in \mathcal{X}} p(x)^2$. We can bound this probability below:

**Lemma 29.** $P(X = X') \geq 2^{-H(X)}$, with equality if and only if $X$ is uniformly distributed.

**Proof.** Since $2^x$ is convex, and $-H(X) = E_{p(x)} \log p(x)$, we have that

$$2^{-H(X)} = 2^{E_{p(x)} \log p(x)} \leq E_{p(x)} p(x) = \sum_{x \in \mathcal{X}} p(x)^2 = P(X = X')$$

With the same proof, we also get

**Lemma 30.** Let $X \sim p(x), X' \sim r(x)$. Then

$$P(X = X') \geq 2^{-H(p) - D(p \| r)}$$

or

$$P(X = X') \geq 2^{-H(r) - D(r \| p)}$$
Proof.

\[ 2^{-H(p) - D(p \| r)} = 2^{E_{p(x)} \log p(x) - \log \frac{p(x)}{r(x)}} \leq E_{p(x)} r(x) = P(X = X') \]

and swapping the roles of \( p(x), r(x) \), the other inequality holds. \( \square \)

These match intuition, since the higher the entropy (uncertainty), the less likely the two random variables will agree.

**Asymptotic Equipartition Property**

Now we will shift gears and discuss an important result that allows for compression of random variables. First recall the different notions of convergence of random variables:

- **Usual convergence** (of real numbers, or deterministic r.v.):
  
  \[ a_n \to a \iff \forall \varepsilon, \exists N \text{ s.t. for } n > N, |a_n - a| \leq \varepsilon \]

- **Convergence a.e. (with probability 1):**
  \[ X_n \to X \text{ a.e.} \iff P(\omega : X_n(\omega) \to X(\omega)) = 1 \]

- **Convergence in probability** (weaker than above)
  \[ X_n \overset{p}{\to} X \iff \lim_{n} P(|X_n(\omega) - X(\omega)| > \varepsilon) = 0 \forall \varepsilon \]

- **Convergence in law/distribution:**
  \[ X_n \overset{d}{\to} X \iff P(X_n \in [a, b]) \to P(X \in [a, b]) \text{ for all } [a, b] \text{ with } P(X = a) = P(X = b) = 0 \]

  (or cdf’s converge at pts of continuity, or characteristic functions converge pt-wise, etc...)

Now we work towards a result known as the Law of Large Numbers for i.i.d r.v. (sample means tend to the expectation in the limit). First,

**Lemma 31. (Chebyshev’s Inequality)** Let \( X \) be a r.v. with mean \( \mu \) and variance \( \sigma^2 \). Then

\[ P(|X - \mu| > \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2} \]

**Proof.**

\[
\sigma^2 = \int |X - \mu|^2 dP \geq \int_{|X - \mu|^2 > \varepsilon} |X - \mu|^2 dP \\
\quad \geq \varepsilon^2 P(|X - \mu|^2 > \varepsilon)
\]

from which we conclude the result. \( \square \)
Theorem 32. (Weak Law of Large Numbers (WLLN)) Let \( X_1, \ldots, X_n \) be i.i.d \( \sim X \) with mean \( \mu \) and variance \( \sigma^2 \). Then

\[
\frac{1}{n} \sum_{i=1}^{n} X_i \longrightarrow X
\]

in probability, i.e.

\[
\lim_{n \to \infty} P \left( \left| \frac{1}{n} \sum_{i=1}^{n} X_i - \mu \right| > \varepsilon \right) = 0
\]

Proof. Note that letting \( A_n = \frac{1}{n} \sum_{i=1}^{n} X_i \), the mean of \( A_n \) is \( \mu \) and the variance is \( \frac{\sigma^2}{n} \). Then by Chebyshev’s inequality,

\[
P \left( \left| \frac{1}{n} \sum_{i=1}^{n} X_i - \mu \right| > \varepsilon \right) \leq \frac{\sigma^2}{n\varepsilon^2} \longrightarrow 0
\]

so that we have convergence in probability.

With a straightforward application of the law of large numbers, we arrive at the following:

Theorem 33. (Asymptotic Equipartition Property (AEP)) If \( X_1, \ldots, X_n \) are i.i.d \( \sim p(x) \), then

\[
-\frac{1}{n} \log p(x_1, \ldots, x_n) \longrightarrow H(X)
\]

in probability.

Proof. Since \( p(x_1, \ldots, x_n) = p(x_1) \cdots p(x_n) \), we have that

\[
-\frac{1}{n} \log p(x_1, \ldots, x_n) = -\frac{1}{n} \sum_{i=1}^{n} \log p(x_i) \longrightarrow -E(\log p(X)) = H(X)
\]

a.e. using the Law of Large Numbers on the random variable \( \log p(X) \), which are still i.i.d, since \( (X, Y) \) independent implies \( (f(X), g(Y)) \) independent.

Remark. Note that this shows that \( p(x_1, \ldots, x_n) \approx 2^{-nH(X)} \), so that the joint is approximately uniform on a smaller space. This is where we get compression of the r.v., and motivates the following definition.

Definition 34. The **typical set** \( A^{(n)}_{\varepsilon} \) with respect to \( p(x) \) is the set of sequences \((x_1, \ldots, x_n)\) such that

\[
2^{-n(H(X)+\varepsilon)} \leq p(x_1, \ldots, x_n) \leq 2^{-n(H(X)-\varepsilon)}
\]

Properties of \( A^{(n)}_{\varepsilon} \):

1. Directly from the definition, we have that

\[
(X_1, \ldots, X_n) \in A^{(n)}_{\varepsilon} \iff H(X) - \varepsilon \leq -\frac{1}{n} \log p(x_1, \ldots, x_n) \leq H(X) + \varepsilon
\]

\[
\iff \left| -\frac{1}{n} \log p(x_1, \ldots, x_n) - H(X) \right| < \varepsilon
\]
2. \( P\left(A^{(n)}_{\varepsilon}\right) \geq 1 - \varepsilon \) for large \( n \).

Proof. The AEP shows that for some

\[
P\left(\left| \frac{1}{n} \log p(X_1, \ldots, X_n) - H(X) \right| > \varepsilon \right) < \varepsilon
\]

and comparing with the first property, this gives a bound on the measure of the complement of the typical set, and this gives the result. \(\square\)

3. \((1-\varepsilon)2^{n(H(X)-\varepsilon)} \leq \left| A^{(n)}_{\varepsilon}\right| \leq 2^{n(H(X)+\varepsilon)}\)

Proof.

\[
1 = \sum_{(x_1, \ldots, x_n)} p(x_1, \ldots, x_n) \geq \sum_{A^{(n)}_{\varepsilon}} p(x_1, \ldots, x_n) \geq \left| A^{(n)}_{\varepsilon}\right| 2^{-n(H(X)+\varepsilon)}
\]

gives the RHS inequality, and

\[
1 - \varepsilon < P\left(A^{(n)}_{\varepsilon}\right) = \sum_{A^{(n)}_{\varepsilon}} p(x_1, \ldots, x_n) \leq \left| A^{(n)}_{\varepsilon}\right| 2^{-n(H(X)-\varepsilon)}
\]

gives the LHS inequality (only true for large enough \( n \)). \(\square\)

Given these properties, we can come up with a scheme for compressing data. Given \( X_1, \ldots, X_n \) i.i.d \( \sim p(x) \) over the alphabet \( \mathcal{X} \), note that sequences from the typical set occur with very high probability \( > (1 - \varepsilon) \). Thus we expect to be able to use shorter codes for typical sequences. The scheme is very simple. Since there are at most \( 2^{n(H(X)+\varepsilon)} \) sequences in the typical set, we need at most \( n(H(X) + \varepsilon) \) bits, and rounding up this is \( n(H(X) + \varepsilon) + 1 \) bits. To indicate in the code that we are describing a typical sequence, we will append a 0 to any code word of a typical sequence. Thus to describe a typical sequence we use at most \( n(H(X) + \varepsilon) + 2 \) bits. For non-typical sequences, since they occur rarely we can afford to use the lousy description with at most \( n \log |\mathcal{X}| + 1 \) bits, and appending a 1 at the beginning to indicate that it is not typical, we describe non-typical sequences with \( n \log |\mathcal{X}| + 2 \) bits. Note that this code is 1-1: given the code word we will be able to decode by looking at the first digit to see whether it is a typical sequence, and then looking in the appropriate table to recover the sequence.

Now we compute the average length of the code. To fix notation, let \( x^n = (x_1, \ldots, x_n) \) and let \( l(x^n) \) denote the length of the corresponding codeword. Then the expected length of code is

\[
E(l(x^n)) = \sum_{x^n} p(x^n) l(x^n)
\leq \sum_{A^{(n)}_{\varepsilon}} p(x^n) [n(H(X) + \varepsilon) + 2] + \sum_{\left(A^{(n)}_{\varepsilon}\right)^c} p(x^n) [n \log |\mathcal{X}| + 2]
\leq P(A^{(n)}_{\varepsilon}) [n(H(X) + \varepsilon)] + (1 - P(A^{(n)}_{\varepsilon})) [n \log |\mathcal{X}| + 2]
\leq n(H(X) + \varepsilon) + \varepsilon n \log |\mathcal{X}| + 2
\leq n(H(X) + \varepsilon')
\]

with \( \varepsilon' = \varepsilon + \varepsilon \log |\mathcal{X}| + 2/n \). Note that \( \varepsilon' \) can be made arbitrarily small, since \( \varepsilon \to 0 \Rightarrow n \to \infty \Rightarrow \varepsilon' \to 0 \).

Thus

\[
E\left[ \frac{1}{n} l(x^n) \right] \leq H(X) + \varepsilon'
\]
for large $n$, and on the average we have represented $X$ using $\approx nH(X)$ bits.

**Question:** Can one do better than this? It turns out that the answer is no, but we will only be able to answer this much later.

**Remark.** This scheme is also not very practical:

- For one, we cannot compress individual r.v., we must take $n$ large and compress them together.
- The table to look up will be large and inefficient.
- This requires knowledge of the underlying distribution $p(x)$.

We will later see schemes that can estimate the underlying distribution as the input comes, a sort of "universal encoding scheme".

**Next time:** We will discuss entropy in the context of stochastic processes.

**Definition 35.** A **stochastic process** is an indexed set of r.v.’s with arbitrary dependence. To describe a stochastic process we specify the joint $p(x_1,\ldots,x_n)$ for all $n$.

And we will be focusing on **stationary** stochastic processes, i.e. ones for which

$$p(x_1,\ldots,x_n) = p(x_{t+1},\ldots,x_{t+n})$$

for all $n,t$, i.e. independent of time shifts.

In the special case of i.i.d. random variables, we had that $H(X_1,\ldots,X_n) = \sum_{i=1}^n H(X) = nH(X)$, and we know how the entropy behaves for large $n$. We will ask how $H(X_1,\ldots,X_n)$ grows for a general stationary process.

**Week 5**

Notes: Midterm 11/2, Next class starts at 2pm.

**Entropy Rates for Stochastic Processes**

Continuing on, we can now define an analogous quantity for entropy:

**Definition 36.** The **entropy rate** of a stochastic process (arbitrary) $\{X_i\}$ is

$$H(X) = \lim_{n \to \infty} \frac{1}{n} H(X_1,\ldots,X_n)$$

when the limit exists. Also define a related quantity,

$$H'(X) = \lim_{n \to \infty} H(X_n | X_1,\ldots,X_{n-1})$$

when the limit exists.

Note that $H(X)$ is the Cesaro limit of $H(X_n | X_1,\ldots,X_{n-1})$ (using the chain rule), whereas $H'(X)$ is the direct limit.
Also, for i.i.d random variables, this agrees with the usual definition of entropy, since

\[ \frac{1}{n} H(X_1, \ldots, X_n) \to H(X) \]

in that case.

**Theorem 37.** For a stationary stochastic process, the limits \( H(X) \) and \( H'(X) \) exist and are equal.

**Proof.** This follows from the fact that \( H(X_n | X_1, \ldots, X_{n-1}) \) is monotone decreasing for a stationary process:

\[
H(X_{n+1} | X_1, \ldots, X_n) \geq H(X_{n+1} | X_2, \ldots, X_n) = H(X_n | X_1, \ldots, X_{n-1})
\]

where the inequality follows since conditioning reduces entropy and the equality follows since the process is stationary. Then the sequence is monotone decreasing and also bounded below by 0 (entropy is nonnegative) and thus the limit and the Cesaro limit exists and are equal. □

Here is an important result that we will not prove (maybe later), by Shannon, McMillan, and Breiman:

**Theorem 38. (Generalized AEP)** For any stationary ergodic process,

\[ -\frac{1}{n} \log p(X_1, \ldots, X_n) \to H(X) \text{ a.e.} \]

Thus, the results about compression and typical sets generalize to stationary ergodic process.

Now we introduce the following notation for Markov Chains (MC) and also some review:

1. We will take \( \mathcal{X} \) to be a discrete, at most countable alphabet, called the **state space**. For instance, \( \mathcal{X} \) could be \( \{0, 1, 2, \ldots\} \).

2. A stochastic process is a Markov Chain (MC) if

\[
P(X_n = j | X_1 = i, X_2 = i_2, \ldots, X_{n-1} = i_{n-1}) = P(X_n = j | X_{n-1} = i)
\]

i.e. conditioning only depends on the previous time. We call this probability which we denote \( P_{ij} \) the **one-step transition probability**. If this is independent of \( n \), then we call the process a time-invariant MC.

Thus, we can write \( X_0 \to X_1 \to \ldots \) and \( X_{n+1} \to X_n \to \ldots \to X_0 \) (since the joint on any finite segment factorizes with dependencies only on the previous random variable.

3. For a time-invariant MC, we express \( P_{ij} \) in a compact **probability transition matrix** \( P \) (may be “infinite dimensional”), where \( P_{ij} \) is the entry of the \( i \)-th row, \( j \)-th column. Note that \( \sum_j P_{ij} = 1 \) for each \( i \).

Given a distribution \( \mu \) on \( \mathcal{X} \), which we can write

\[
\mu = [p(0), p(1), \ldots]
\]
if at time 0 the marginal distribution is $\mu$, then at time 1 the probabilities can be found by the formula

$$P(X_1 = j) = \sum_i P(X_0 = i) P_{ij} = [\mu P]_j$$

where we use the notation row vector $\mu$ times matrix $P$. It turns out that a necessary and sufficient condition for stationarity is that the marginal distributions all equal some distribution $\mu$ for every $n$, where $\mu = \mu P$. For a time-invariant MC, the joint depends only on the initial distribution $P(X_0 = i)$ and the time invariant transition probability (which is fixed), and thus if $P(X_0 = i) = \mu_i$ where $\mu = \mu P$, stationarity follows.

4. Under some conditions on a general (not necessarily time-invariant) Markov chain (ergodicity, for instance), no matter what initial distribution $\mu_0$ we start from $\mu_n \to \mu$ (convergence a.e. to a unique stationary distribution $\mu$).

Perron-Frobenius has such a result about conditions on $P$.

An example of a “not so nice” Markov chain is a Markov chain with two isolated components. Then the limits exist but depend on the initial distribution.

Now we turn to results about Entropy Rates. For stationary MC’s, note that

$$H(X) = H'(X) = \lim_{n \to \infty} H(X_n | X_{n-1}) = \lim_{n \to \infty} H(X_n | X_{n-1}) = H(X_2 | X_1) = -\sum_{i,j} \mu_i P_{ij} \log P_{ij}$$

**Example 39.** For a 2 state Markov chain with

$$P = \begin{pmatrix} 1 - \alpha & \alpha \\ \beta & 1 - \beta \end{pmatrix}$$

we can spot a stationary distribution to be $\mu_1 = \frac{\beta}{\alpha + \beta}$, $\mu_2 = \frac{\alpha}{\alpha + \beta}$. Then we can compute the entropy rate accordingly

$$H(X) = H(X_2 | X_1) = \frac{\beta}{\alpha + \beta} H(\alpha) + \frac{\alpha}{\alpha + \beta} H(\beta)$$

**Example 40.** (Random Walk on a Weighted Graph)

For a graph with $m$ nodes $X = \{1, ..., m\}$, assign weights $w_{ij} \geq 0$ with $w_{ij} = w_{ji}$ and $w_{ii} = 0$ to edges of the graph. Let $w_i = \sum_j w_{ij}$, the total weight of edges leaving $i$, and $W = \sum_{i<j} w_{ij}$, the sum of all the weights. Then define transition probabilities $p_{ij} = \frac{w_{ij}}{w_i}$ and examine the resulting stationary distribution. (An example random walk fixes a starting point, say 1, which corresponds to initial distribution $\mu_0 = [1, 0, ...]$). The stationary distribution (turns out to be unique) can be guessed: $\mu_i = \frac{w_i}{2W}$. To verify:

$$[\mu P]_j = \sum_i \mu_i p_{ij} = \sum_i \frac{w_i}{2W} \cdot \frac{w_{ij}}{w_i} = \frac{1}{2W} \sum_i w_{ij} = \frac{w_j}{2W}$$
which shows that $\mu P = \mu$. Now we can compute the entropy rate:

$$H(\mathcal{X}) = H(X_2 | X_1)$$

$$= - \sum_{i,j} \mu_i p_{ij} \log p_{ij}$$

$$= - \sum_{i,j} \frac{w_i}{2W} w_{ij} \log \frac{w_{ij}}{w_i}$$

$$= - \sum_{i,j} \frac{w_{ij}}{2W} \log \frac{w_{ij}}{w_i} + \sum_{i,j} \frac{w_{ij}}{2W} \log \frac{2W}{w_i}$$

$$= H\left(\frac{w_{ij}}{2W}\right) + H\left(\frac{w_i}{2W}\right)$$

which we can interpret as the entropy of the edge distribution plus the entropy of all the node distribution, so to speak.

In the special case where all the weights are equal, let $E_i$ be the number of edges from node $i$, and let $E$ be the total number of edges. Then $\mu_i = \frac{E_i}{2E}$ and

$$H(\mathcal{X}) = \log(2E) - H\left(\frac{E_1}{2E}, \ldots, \frac{E_m}{2E}\right)$$

Example 41. (Random Walk on a Chessboard) In the book, but is just a special case of the above.

Example 42. (Hidden Markov Model) This concerns a Markov Chain $\{X_n\}$ which we cannot observe (is hidden), and functions (observations) $Y_n(X_n)$ which we can observe. We can bound the entropy rate of $Y_n$. $Y_n$ does not necessarily form a Markov Chain.

Data Compression

We now work towards the goal of arguing that entropy is the limit of data compression. First let’s consider properties of codes. For instance, a simple example is the Morse code, where each letter is encoded by a sequence of dots and dashes (binary alphabet), where the idea is to put frequently used letters with shorter descriptions, and the descriptions have arbitrary length.

We define a code $C$ for some source (r.v.) $X$ to be a mapping from $\mathcal{X} \rightarrow D^*$, finite length strings from a $D$-ary alphabet. We denote $C(x)$ to be the code word of $x$ and $l(x)$ to be the length of $C(x)$. The expected codeword length is then $L(C) = E(l(X))$.

Reasonable Properties of Codes.

We say a code is non-singular if it is one to one, i.e. each $C(x)$ is easily decodable to recover $x$.

Now consider the extension $C^*$ of a code $C$ to be defined by $C(x_1, \ldots, x_n) = C(x_1) \cdots C(x_n)$ (concatenation of codes), for encoding many realizations of $\mathcal{X}$ at once (for instance, to send across a channel). It would be good for the extension then to be non-singular as well. We call a code uniquely decodable (UD) if the extension is nonsingular.

Even more specific, a code is prefix-free (or prefix) if no codeword is a prefix of another code word. In this case the moment a code word is transmitted we can decode it immediately (without having to receive the next letter).

Example binary codes:

<table>
<thead>
<tr>
<th>$x$</th>
<th>singular</th>
<th>nonsingular but UD</th>
<th>UD but not prefix</th>
<th>prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>01</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>00</td>
<td>011</td>
<td>110</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>11</td>
<td>0111</td>
<td>111</td>
</tr>
</tbody>
</table>

27
The goal is to minimize $L(C)$, and to this end we note a useful inequality.

**Theorem 43. (Kraft Inequality)** For any prefix code over an alphabet of size $D$, the codeword lengths $l_1, \ldots, l_m$ must satisfy

$$\sum_{i=1}^{m} D^{-l_i} \leq 1$$

Conversely, for any set of lengths satisfying this inequality, there exists a prefix code with the specified lengths.

**Proof.** The idea is to examine a $D$-ary tree, where from the root, each branch represents a letter from the alphabet. The prefix condition implies that the codewords correspond to the leaves of the $D$-ary tree. The argument is then to extend the tree to a full $D$-ary tree (add nodes until the depth is $l_{\text{max}}$, the max length). The number of leaves at $l_{\text{max}}$ for a full tree is $D^{l_{\text{max}}}$, and if a codeword has length $l_i$, then the number of leaves at the bottom level that correspond to this word is $D^{l_{\text{max}}-l_i}$. Thus, the codewords of a prefix encoding satisfy

$$\sum_{i} D^{l_{\text{max}}-l_i} \leq D^{l_{\text{max}}}$$

and the result follows. The converse is easy to see by reverseing the construction. Associate each codeword to $D^{l_{\text{max}}-l_i}$ consecutive leaves (which have a common ancestor). 

Next time we will show that the Kraft inequality needs to hold also for UD codes, in which case there is no loss or benefit from focusing on prefix codes.

**Week 6** (10/26/2009)

First we note that we can extend the Kraft inequality discussed last week to countably many lengths as well.

**Theorem 44. (Extended Kraft Inequality)** For any prefix code over an alphabet of size $D$, the codeword lengths $l_1, l_2, \ldots$ must satisfy

$$\sum_{i=1}^{\infty} D^{-l_i} \leq 1$$

Conversely, for any set of lengths satisfying this inequality, there exists a prefix code with the specified lengths.

**Proof.** Given a prefix code, we can again consider the “infinite” tree with the codewords at the leaves. We still extend the tree so that the tree is “full” in a different sense, adding nodes until every node either has $D$ children or no children. Such a tree can then be associated to a probability space on the (countably-infinite) leaves by the formula $P(X = l) = D^{-\text{depth}(l)}$, and the sum of all the probabilities is 1. Since we have added dummy nodes, we have only increased the sum of $D^{l_i}$, and thus

$$\sum_{i=1}^{\infty} D^{-l_i} \leq 1$$
Conversely, as before we can reverse the procedure above to construct a $D$-ary tree with leaves at depth $l_i$. 

It turns out that UD codes also need to satisfy Kraft’s inequality.

**Theorem 45. (McMillan)** The codeword lengths $l_1, l_2, \ldots, l_m$ of a UD code satisfy

$$\sum_{i=1}^{m} D^{-l_i} \leq 1$$

and conversely, for any set of lengths satisfying this inequality, there exists a UD code with the specified lengths.

**Proof.** The converse is immediate from the previous Theorem 44, since we can find a prefix code satisfying the specified lengths, and prefix codes are a subset of UD codes.

Now given a UD code, examine $C^k$, the extension of the code, where $C(x_1, \ldots, x_k) = C(x_1) \cdots C(x_k)$, and $l(x_1, \ldots, x_k) = \sum_{m=1}^{k} l(x_m)$. Now look at

$$\left( \sum_{x \in \mathcal{X}} D^{-l(x)} \right)^k = \sum_{x_1, \ldots, x_k} D^{-l(x_1, \ldots, x_k)} = \sum_{m=1}^{kl_{\text{max}}} a(m)D^{-m}$$

which is a polynomial in $D^{-1}$, where $a(m)$ is the number of sequences with code length $m$, and since $C^k$ is nonsingular, $a(m) \leq D^m$. Thus

$$\left( \sum_{x \in \mathcal{X}} D^{-l(x)} \right)^k \leq \sum_{m=1}^{kl_{\text{max}}} 1 = kl_{\text{max}}$$

Taking $k$-th roots, and taking the limit as $k \to \infty$, we then have that

$$\sum_{x \in \mathcal{X}} D^{-l(x)} \leq (kl_{\text{max}})^{1/k} \to 1$$

which gives the result. 

**Remark.** Thus, there is no loss in considering prefix codes as opposed to UD codes, since for any UD code there is a prefix code with the same lengths.

**Optimal Codes**

Given a source with probability distribution $(p_1, \ldots, p_m)$, we would want to know how to find a prefix code with the minimum expected length. The problem is then a minimization problem:

$$\min L = \sum_{i=1}^{m} p_i l_i$$

s.t. $\sum_{i=1}^{m} D^{-l_i} \leq 1$

$l_i \in \mathbb{Z}^+$
where $D$ is a fixed positive integer (the base). This is a linear program with integer constraints. First, we ignore the integer constraint and assume equality in Kraft’s inequality. Then we have a Lagrange multiplier problem. The derivative of the Lagrangian gives the equations

\[
p_i = \lambda D^{-l_i} \log D
\]

\[
D^{-l_i} = \frac{p_i}{\lambda \log D}
\]

\[
\sum_{i=1}^{m} \frac{p_i}{\lambda \log D} = 1
\]

\[
\frac{1}{\lambda \log D} = 1
\]

\[
\lambda = \frac{1}{\log D}
\]

\[
D^{-l_i} = p_i
\]

Thus the optimal codeword lengths are $l_i^* = -\log_D p_i$ which yields the expected code length

\[
L^* = \sum_i p_i l_i^* = -\sum_i p_i \log_D p_i = H(X)
\]

the entropy being in base $D$. This gives a lower bound for the minimization problem.

**Theorem 46.** For any prefix code for a random variable $X$, the expected codeword length satisfies

\[
L \geq H(X)
\]

with equality if and only if $D^{-l_i} = p_i$.

**Proof.** This has already been proven partially via Lagrange multipliers (with equality constraint in Kraft), but we now turn to an Information Theoretic approach. Let $r_i = \frac{D^{-l_i}}{c}$ where $c = \sum_i D^{-l_i}$. Then

\[
L - H(X) = \sum_i p_i l_i + \sum_i p_i \log p_i
\]

\[
= -\sum_i p_i \log D^{-l_i} + \sum_i p_i \log p_i
\]

\[
= \sum_i p_i \log \frac{p_i}{r_i} - \sum_i p_i \log c
\]

\[
\geq D(p \parallel r)
\]

\[
\geq 0
\]

and thus $L \geq H(X)$ as desired. Equality holds if and only if $\log c = 0$ and $p = r$, in which case we have equality in Kraft’s inequality ($c = 1$) and $D^{-l_i} = p_i$, which finishes the proof. \qed

Now if $p_i \neq D^{-l_i}$, we would like to know how close to the entropy lower bound we can get. A good suboptimal code is the Shannon-Fano code, which involves a simple rounding procedure.

**Theorem 47.** For optimal codeword lengths $l_1^*, \ldots, l_m^*$ for a random variable $X$, $L^* = \sum_i p_i l_i^*$ satisfies

\[
H(X) \leq L^* \leq H(X) + 1
\]
Proof. Recall from Theorem 46 that lengths of $\log \frac{1}{p_i}$ achieve the lower bound if they are integers. We now suppose we used $\lceil \log \frac{1}{p_i} \rceil$ instead. This corresponds to the Shannon code, but first we note that the lengths still satisfy Kraft’s inequality:

$$\sum_i D^{-\lceil \log \frac{1}{p_i} \rceil} \leq \sum_i D^{-\log \frac{1}{p_i}} \leq 1$$

and also that this gives the desired upper bound:

$$\sum_i p_i \lceil \log \frac{1}{p_i} \rceil \leq \sum_i p_i \left( \log \frac{1}{p_i} + 1 \right) = H(X) + 1$$

Since the optimal code is better than this Shannon code, this gives the desired upper bound $H(X) + 1$. The lower bound was already proved in the previous theorem 46. □

Remark 48. The overhead of 1 in the previous theorem can be made insignificant if we encode blocks $(X_1, \ldots, X_B)$ to increase $H(X_1, \ldots, X_B)$, effectively spreading the overhead across a block. By the previous theorem, we can find a code for $(X_1, \ldots, X_B)$ with lengths $l(x_1, \ldots, x_n)$ such that

$$H(X_1, \ldots, X_n) \leq E[l(x_1, \ldots, x_n)] \leq H(X_1, \ldots, X_n) + 1$$

Then per outcome, we have that

$$\frac{H(X_1, \ldots, X_n)}{n} \leq \frac{E[l(x_1, \ldots, x_n)]}{n} \leq \frac{H(X_1, \ldots, X_n)}{n} + \frac{1}{n}$$

and letting $L_n = \frac{E[l(x_1, \ldots, x_n)]}{n}$, we see that $L_n \to H(X)$ if $X_1, \ldots, X_n$ is stationary. Note the connection to the Shannon-McMillan-Breiman Theorem (or AEP in the case of i.i.d $X_i$). There we saw that we can design codes using typical sequences to get a compression rate arbitrarily close to the entropy. This remark uses a slightly different approach using Shannon codes.

Huffman Codes

Now we consider a particular prefix code for a probability distribution, which will turn out to be optimal. The idea is essentially to construct a prefix code tree (as in the proofs of Kraft’s inequality) from the bottom up. We first look at the case $D = 2$ (binary codes)

Huffman Procedure:

1. Order the symbols in decreasing order of probabilities, so $x_1, \ldots, x_m$ satisfies $p_1 \geq p_2 \geq \ldots \geq p_m$.

2. Merge the two symbols of lowest probability (last 2 symbols) into a single tree associated with probability $p_m + p_{m-1}$.

Denote this symbol $x_{m-1}^r$. 31
3. Repeat this procedure until all symbols are merged into a giant tree.

4. Assign \{0, 1\} to the branches of the tree (each node should have one branch labeled 0 and one branch labeled 1).

If \(D > 2\), then there is an obvious generalization, except that every merge step reduces the number of symbols by \(D - 1\), so that the procedure works without a hitch when the number of symbols is \(1 \mod D - 1\). The appropriate way to deal with this is to add dummy symbols associated with probability 0 until the number of symbols is \(1 \mod D - 1\), and perform the procedure.

**Example 49.** Consider four symbols \(a, b, c, d\) with probabilities 0.5, 0.3, 0.1, 0.1 and \(D = 3\). Then we add a fourth dummy variable and perform the procedure. There are two steps, first merging \(c, d, \text{dummy}\), and then merging the rest:

```
     0
    / \
   1   2
  / \ / \ \
 a  b c  d dummy
```

so that \(C(a) = 0, C(b) = 1, C(c) = 20, C(d) = 21\). The expected length is 1.2 ternary digits, and the entropy is 1.06 ternary digits.

**Remark 50.** (Huffman vs Shannon codes) Note that there are examples where Shannon codes give longer or shorter codewords than Huffman codes for particular codes, but in expected length, Huffman always performs better, as we will soon prove. In the examples below, we will also see that Huffman’s procedure does not give a unique tree in cases involving ties, and how ties are broken can produce codes with different sets of lengths.

Consider four symbols \(a, b, c, d\) with probabilities \(1/3, 1/3, 1/4, 1/12\). There are two possible binary code trees:

```
     a
    / \     /
   b   c   d

     a
    / \
   b   c
```

corresponding to different length codes. To \(c\), corresponding to probability \(1/4\), Shannon’s code gives a code of length 2, which is shorter than the code for \(c\) in the first code tree.

Also, an easy example shows that the Shannon code can be worse than optimal for a particular outcome. Consider a binary random variable \(X\) distributed Bernoulli(1/10000). Huffman code uses just 1 bit, whereas the Shannon code uses \([\log 10000]\) = 14 bits for one, and 1 bit for the other. But of course on the average it is off by at most 1 bit, as proven previously.

Now we prove the optimality of the Huffman Code. First assume that \(p_1 \geq p_2 \geq \ldots \geq p_m\).
Lemma 51. For any distribution, any optimal prefix code satisfies

1. If \( p_j > p_k \) then \( l_j \leq l_k \)
2. The two largest codewords have the same length

and there exists an optimal prefix code satisfying

3. The largest codewords differ only in the last bit and correspond to the least likely symbols.

Proof. First we prove (1). Given an optimal prefix code \( C_m \), and let \( p_j > p_k \) corresponding to \( l_j, l_k \), and let \( C_m' \) be the code resulting from swapping \( x_j \) and \( x_k \). Then

\[
L(C_m') - L(C_m) = p_j(l_k - l_j) + p_k(l_j - l_k) = (p_j - p_k)(l_k - l_j)
\]

and since \( L(C_m) \) is optimal, \( L(C_m') - L(C_m) \geq 0 \), and since \( p_j > p_k \), \( p_j - p_k > 0 \) and therefore \( l_k - l_j \geq 0 \) as well, proving the result.

To prove (2), suppose that the two largest codewords in an optimal code are not the same length. Then we can delete the last bit of the longest codeword, preserving the prefix property but shortening the expected length, contradicting optimality.

To prove (3), we suppose that the two largest codewords \( x, y \) differ in more than the last bit. This means that the two are leaves corresponding to different parents. Note that both must have at least one sibling, or else we can trim the code while preserving prefix property and shortening the expected length, which contradicts optimality. Then we can swap one sibling of \( x \) with \( y \) and preserve the structure of the tree, preserving the prefix property and the expected length (since \( x, y \) are at the same depth). This gives a new code with the same expected length but which satisfies (3).

Now we are in a position to prove optimality.

Theorem 52. The Huffman code is optimal, i.e. any other code has larger expected length.

Proof. We will proceed by induction on the number of symbols. If there are only 2 symbols, then trivially the only code has length 1 and agrees with the Huffman procedure. Now assume that the Huffman procedure gives the optimal expected length for \( m - 1 \) symbols, which we denote \( L^*_{m-1} \). Now suppose that we have a code for \( m \) symbols \( C_m \) with lengths \( l_i \) that is not generated from the Huffman procedure. By the lemma we can rearrange the code so that the three properties are satisfied (in particular the two largest code words differ only by the last bit). Now as in the Huffman procedure, we combine the two largest code words of lengths \( l_m \) to a single code word of length \( l_m' = l_m - 1 \) (truncating the last bit) corresponding to probability \( p_{m-1} + p_m \). This gives a code for \( m - 1 \) symbols, which we denote \( C_{m-1} \). By the induction hypothesis \( L^*_{m-1} \leq L(C_{m-1}) \). Also we have that

\[
L(C_m) = \sum_{i=1}^{m} p_i l_i = \sum_{i=1}^{m-2} p_i l_i + p_{m-1}(l_{m-1} + 1) + p_m(l_m' + 1) = L(C_{m-1}) + p_{m-1} + p_m
\]

and thus \( L(C_m) \geq L^*_{m-1} + p_{m-1} + p_m = L^*_m \), where \( L^*_m \) is the expected length of the Huffman code on \( m \) symbols. This completes the proof. \( \square \)
Other Codes with Special Properties

Shannon-Fano-Elias Coding

Here we describe an explicit code with lengths \(\lceil \log \frac{1}{p_i} \rceil + 1\) (slightly worse than lengths in Shannon code). Let \(\mathcal{X} = \{1, \ldots, m\}\) and assume \(p(x) > 0\) for all \(x\). Let \(F(x) = \sum_{a < x} p(a)\) be the cdf (step function in this case), and let \(\tilde{F}(x)\) be the same as \(F(x)\) except that \(\tilde{F}(x) = F(x) + \frac{1}{2} p(x)\). However instead of using \(F(x) + \frac{1}{2} p(x)\), we round off this number for \(l(x)\) bits of precision, so that \(\tilde{F}(x)\) and its truncation differs by at most \(\frac{1}{2^{l(x)}}\). Then if we take \(l(x) = \lceil \log \frac{1}{p_x} \rceil + 1\), then we have that \(\frac{1}{2^{l(x)}} \leq \frac{p(x)}{2}\), so that the truncation of \(\tilde{F}(x)\) does not drop below \(F(x)\). Denote the truncated version of \(\tilde{F}(x)\) by \(\tilde{F}(x)\).

The claim is that the \(l(x)\) bits of \(\tilde{F}(x)\) form a prefix code. Note \(\tilde{F}(x) = 0.z_1\ldots z_l(x)\), and the prefix condition tells us that no other code word is of the form \(0.z_1\ldots z_l(x) \ast \ldots\), or in other words, no other code word is in the interval \([\tilde{F}(x), \tilde{F}(x) + 2^{-l(x)}]\). So in this way we can identify each codeword with an interval, and the prefix condition is satisfied if and only if the intervals do not overlap. By construction, this is satisfied, since the interval corresponding to \(\tilde{F}(x)\) is between \(F(x) = \tilde{F}(x) - p(x)/2\) and \(\tilde{F}(x) + p(x)/2 = F(x + 1)\), and thus the intervals are disjoint. See diagram:

Thus we have a prefix code, with expected length

\[
L = \sum_x p(x) \left( \left\lceil \log \frac{1}{p(x)} \right\rceil + 1 \right) \leq H(X) + 2
\]

So the overhead is slightly worse, and in fact there is a way to modify the procedure so that the overhead is reduced to 1. The main advantage to this coding is that we do not need to rearrange the probabilities. The second advantage is that this code can be slightly generalized to an “arithmetic coding”, a coding that is easily adapted to encoding \(n\) realizations of the r.v. \(X^n\) (which is how we increase the entropy to make the overhead insignificant). More specifically, it is already easy to see that \(F(x^n)\) can be computed from \(p(x^n)\), and more importantly, \(F(x^{n+1})\) can be computed from \(F(x^n)\). See Section 13.3 of text.

Lempel-Ziv Algorithm

This is from Section 13.4, 13.5. Can we compress information without knowing \(p(x)\)? It turns out there is an algorithm which can compress information adaptively, as the information is received, in such a way to achieve rates of compression close to the entropic lower bound. The algorithm is as follows:

Given a block in the source sequence, e.g. 010011011001:

1. Parse the block with commas, so that every term in the sequence is unique. i.e. place next comma after the shortest string not already in the list.
For the given example,

\[ 0, 1, 00, 11, 10, 110, 001 \]

2. Letting \( n \) be the block length, suppose we end up with \( c(n) \) strings. We encode each string by an index describing the prefix of the string (excluding the last bit), and the last bit. Possible prefixes in this case is the null prefix \( \emptyset \) and the first \( c(n) - 1 \) strings (the last string cannot be a prefix of the previous strings by construction). Thus there are \( c(n) \) prefixes, so we need \( \lceil \log c(n) \rceil \) bits to describe the prefix.

For the example, we have \( n = 14 \) and \( c(n) = 7 \) and \( \lceil \log c(n) \rceil = 3 \). Thus, we index the prefixes:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>\emptyset</th>
<th>0</th>
<th>1</th>
<th>00</th>
<th>11</th>
<th>10</th>
<th>110</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>000</td>
<td>001</td>
<td>010</td>
<td>011</td>
<td>100</td>
<td>101</td>
<td>110</td>
</tr>
</tbody>
</table>

and then we encode each string:

<table>
<thead>
<tr>
<th>String</th>
<th>0</th>
<th>1</th>
<th>00</th>
<th>11</th>
<th>10</th>
<th>110</th>
<th>001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>000,0</td>
<td>000,1</td>
<td>001,0</td>
<td>010,1</td>
<td>010,0</td>
<td>100,0</td>
<td>011,1</td>
</tr>
</tbody>
</table>

Then the block 0100110110001 is encoded roughly as

\[(000,0)(000,1)(001,0)(010,1)(010,0)(100,0)(011,1)\]

(need some scheme to either indicate block sizes at the beginning or represent parentheses and commas) Note that knowing the algorithm used to construct this sequence, we can decode easily: First two are 0, 1, the third says 10, etc. We can construct a table of prefixes as we decode.

Note that in this case the “compressed” sequence is actually longer! The original block is 14 bits and the compressed sequence is around 28 bits. However, for a very long block we can start to see savings. In general, for a block of size \( n \), we achieve a rate of compression of

\[
\frac{c(n)(\log(\lceil c(n) \rceil) + 1)}{n} \rightarrow H(X)
\]

where the convergence can be shown for a stationary ergodic source sequence. Such codes are called universal codes, since we do not need to know \( p(x) \), yet the average lengths per outcome still become close to the entropy \( H(X) \).

**Lossy Compression**

We can then go further and allow for losses according to some distortion function. Given a source outcome, we reproduce as \( \hat{X} \), so that \( d(X, \hat{X}) \) is minimized, and for instance \( d(X, \hat{X}) = (X - \hat{X})^2 \). Letting \( D \) be the maximum average distortion, and \( R \) the rate of compression, Rate distortion theory is about possible \((R, D)\) points. We may return to this later. This is very common, for instance in JPEG compression, where we allow for a slight blurring of the image to achieve a compressed representation.

**Channel Capacity**

We now turn to communication over a channel, working towards Shannon’s Second Theorem, (Theorem 4) discussed in the overview at the beginning of the notes. Recall a channel takes input \( X \) and outputs \( Y \) according to the transition probability \( p(y|x) \).

\[ X \rightarrow \text{Channel} \rightarrow Y \]
We want to find the best way to make use of the channel. As in compression, it turns out to be useful to look at \( X^n \) (sending \( n \) pieces of data at a time through the channel in some form). The picture looks like

For each point \( X^n \in X^n \), the picture above describes possibilities for what the channel can output. The images may certainly overlap, and the goal is the maximize the number of inputs such that the output sets do not intersect. Then every output can be uniquely decoded without any error. We may also allow for a small probability of error as well. In any case, the scheme relies on redundancy to achieve a robust communication code suitable for the channel.

First, we establish a few definitions:

A **discrete channel** processes a sequence of r.v. \( X_k \) and outputs \( Y_k \) according to some probability distribution, where \( X_k \) and \( Y_k \) come from discrete alphabets. A channel is **memoryless** if the output only depends on input at a particular time, i.e.

\[
p(y_k | x_k, y_{k-1}, \ldots) = p(y_k | x_k)
\]

and

\[
p(y^n | x^n) = \prod_{i=1}^{n} p(y_i | x_i)
\]

Recall that \( y^n \) is notation for \((y_1, \ldots, y_n)\).

We will mainly be dealing with discrete memoryless channels (DMC), which is completely described by a probability transition matrix \( p(y | x) \).

The **“Information” Channel Capacity** of a DMC \( p(y | x) \) is defined as

\[
C = \max_{p(x)} I(X; Y)
\]

noting that \( I(X; Y) \) depends entirely on the joint \( p(x, y) = p(x)p(y | x) \) and \( p(y | x) \) is fixed by the channel.

We will show that this quantity describes the max communication rate (in bits per channel use) of information we can send through the channel with arbitrarily low probability of error. For now we first compute \( C \) for a few easy examples.

**Example 53.** *Noiseless binary channel.* This channel is entirely deterministic, and \( p(y | x) = \begin{cases} 1 & y = x \\ 0 & y \neq x \end{cases} \), so that \( Y = X \) (input always equals output).

\[
\begin{array}{c}
0 \\
X
\end{array} \quad \begin{array}{c}
0 \\
\to
\end{array} \quad \begin{array}{c}
1 \\
Y
\end{array}
\]
We then expect to be able to send 1 bit perfectly across the channel at every use, and so we expect the channel capacity $C$ to be 1 bit. Then we note

$$I(X;Y) = H(Y) - H(Y | X) = H(Y) \leq 1$$

noting that $H(Y | X) = 0$ since $Y = X$ ($Y$ is a function of $X$), and since $Y$ is binary the entropy is upper bounded by 1, with equality if and only if $Y \sim \text{Bern}(1/2)$. Then equality can easily be achieved if we choose $X \sim \text{Bern}(1/2)$, and this shows that $C = 1$ bit per transmission.

**Example 54. Noisy channel with nonoverlapping outputs.** Here we have

![Diagram](image)

so that if $X = 0$ then $Y = \begin{cases} 1 \text{ w.p. } 1/2 \\ 2 \text{ w.p. } 1/2 \end{cases}$, and similarly for $X = 1$. Since given $Y$ we can determine completely $X$, we expect to be able to communicate 1 bit of information per transmission. This time we expand $I(X;Y)$ a little differently when computing $C$:

$$I(X;Y) = H(X) - H(X | Y) = H(X) \leq 1$$

Here we note that $X$ is a function of $Y$, so $H(X | Y) = 0$, and again the upper bound can be achieved with $X \sim \text{Bern}(1/2)$. Thus $C = 1$ bit per transmission. Note that the transition probabilities above can be arbitrary and the same result holds.

**Example 55. Noisy typewriter.**

![Diagram](image)

Consider the diagram above, where each branch is labeled with probability 1/2. Note that if we restrict ourselves to using only 0 or 2 (or 1 and 3), then this situation looks exactly like the previous example, which allows us to communicate at 1 bit per transmission through this channel. It turns out that this is the best we can do.

$$I(X;Y) = H(Y) - H(Y | X) \leq 1$$
noting that given $X, Y \sim \text{Bern}(1/2)$ so that $H(Y \mid X = x) = 1$ and therefore $H(Y \mid X) = 1$, and $H(Y) \leq \log 4 = 2$. Note that equality is achieved when $Y \sim \text{unif}\{0, 1, 2, 3\}$, which is easily achieved if we take $X$ also to be $\text{unif}\{0, 1, 2, 3\}$. Note that the $X$ chosen does not quite match our intuition for how best to use the channel, but it allows us to compute $C$ easily.

**Example 56. Binary symmetric channel.** This is perhaps the most common channel model used, and models a channel that is a little noisy, where a bit is flipped with some probability.

![Binary symmetric channel diagram](https://via.placeholder.com/150)

Thus $X$ is preserved with probability $1 - \alpha$ and flipped with probability $\alpha$. Computing the capacity, we have that

$$I(X; Y) = H(Y) - H(Y \mid X)$$
$$= H(Y) - \sum_{0}^{1} p(x) H(Y \mid X = x)$$
$$= H(Y) - H(\epsilon)$$
$$\leq 1 - H(\epsilon)$$

noting that given $X, Y$ is $\text{Bern}(\epsilon)$ distributed. Equality is achieved if $Y \sim \text{Bern}(1/2)$. Again, we note that $X \sim \text{Bern}(1/2)$ works, and the capacity is $1 - H(\epsilon)$. It is not clear how to use this channel to operate slightly below capacity with arbitrarily low probability of error.

Since we will use this channel often, we will often abbreviate with $\text{BSC}(\alpha)$.

**Example 57. Binary erasure channel.** This is perhaps the third most common, after the Gaussian channel (covered later). Here the input is preserved, and with some probability is erased, denoted by $Y = e$.

![Binary erasure channel diagram](https://via.placeholder.com/150)

Computing the capacity, we have

$$I(X; Y) = H(Y) - H(Y \mid X)$$
$$= H(Y) - H(\alpha)$$

It is tempting to use $H(Y) \leq \log 3$, with equality when $Y \sim \text{unif}\{0, e, 1\}$, however there is no distribution on $X$ that can achieve this. Instead, we compute more directly. Note that

$$Y = \begin{cases} e & w.p. \alpha \\ X & w.p. 1 - \alpha \end{cases}$$
which is a disjoint mixture, and therefore

\[ H(Y) = (1 - \alpha)H(X) + H(\alpha) \]

Plugging in, we then have

\[ I(X; Y) = (1 - \alpha)H(X) \leq 1 - \alpha \]

with equality when \( X \sim \text{Bern}(1/2) \). Thus the capacity is \( 1 - \alpha \) bits per transmission.

Here we have a little bit of intuition. Noting that the channel drops \( \alpha \) proportion of the input, the best we can do is communicate at a rate of \( 1 - \alpha \), though it is not clear how to do so.

We can generalize the procedure used in a few of the examples above to a more general class of channels. Suppose that our DMC has finite alphabet, and thus our transition probability can be written in matrix form \( P_{ij} = P(Y = j | X = i) \). We say that a DMC (finite) is symmetric if the rows of \( P_{ij} \) are permutations of each other, and the same for the columns. For example,

\[
P_{ij} = \begin{pmatrix}
0.1 & 0.4 & 0.1 & 0.4 \\
0.4 & 0.1 & 0.4 & 0.1
\end{pmatrix}
\]

is such an example. Another example is the BSC(\( \alpha \)), with transition matrix

\[
\begin{pmatrix}
1 - \alpha & \alpha \\
\alpha & 1 - \alpha
\end{pmatrix}
\]

We then can state a result for symmetric channels:

**Theorem 58.** For symmetric channels with input alphabet \( \mathcal{X} \) of size \( n \) and output alphabet \( \mathcal{Y} \) of size \( m \), the capacity is

\[ C = \log |\mathcal{Y}| - H(P_{ij}, j = 1, \ldots, m) \]

i.e. \( \log |\mathcal{Y}| \) minus the entropy of a row of \( P \).

**Proof.** As usual, we compute

\[
I(X; Y) = H(Y) - H(Y | X) 
\leq \log |\mathcal{Y}| - H(P_{ij}, j = 1, \ldots, m)
\]

where we note that given \( X \), the probabilities in each row are permutations, and thus correspond to the same entropy. Equality holds if \( Y \sim \text{unif}(\mathcal{Y}) \). The most sensible candidate for \( X \) is then also \( X \sim \text{unif}(\mathcal{X}) \), which will turn out to work due to symmetry in columns:

\[
p(y) = \sum_{x \in \mathcal{X}} p(x)p(y | x) \\
= \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} p(y | x) \\
= \frac{C}{|\mathcal{X}|}
\]
where the last equality follows from symmetry (all column sums are the same). Thus \( Y \) is uniform, and we have achieved the upper bound established earlier. Thus the capacity is \( \log |Y| - H(P_{ij}, j = 1, \ldots, m) \) as desired.

Note that in the proof above we only needed the columns of \( P_{ij} \) to sum to the same value, which is slightly weaker than the symmetry condition. Thus if we then define a weakly symmetric channel to be one where the rows are permutations and the columns have the same sum, the above result holds for this case as well. A simple example of such a channel that is not symmetric is

\[
P_{ij} = \begin{pmatrix} 1/3 & 1/6 & 1/2 \\ 1/3 & 1/2 & 1/6 \end{pmatrix}
\]

Next time we will establish the connection between the channel capacity and the maximum rate of communication across the channel.

**Week 8**

The information channel capacity \( C = \max_{p(x)} I(X; Y) \) satisfies the following properties:

- \( C \geq 0 \), (since \( I(X; Y) \geq 0 \))
- \( C \leq \log |\mathcal{X}| \) and \( C \leq \log |\mathcal{Y}| \) (since \( I(X; Y) \leq H(X) \leq \log |\mathcal{X}| \) and similarly for \( Y \))
- \( I(X; Y) \) is a continuous, concave function of \( p(x) \), and thus a local maximum of \( I(X; Y) \) is the global maximum.

**Channel Coding Theorem**

First, we provide some intuition for why \( I(X; Y) \) is the relevant quantity to study for channels. As we did with results about compression, we will make use of the typical sequences for \( \mathcal{X}^n \) and \( \mathcal{Y}^n \) in \( n \) uses of the channel.

Suppose we fix the distribution of \( X \). Recall that to make the best use of the channel, we want to use as many elements of \( \mathcal{X}^n \) whose images do not overlap (in the best case scenario). Once we have chosen such elements the situation looks like the noisy nonoverlapping outputs channel. Recall that the size of the typical set of \( \mathcal{Y}^n \) is around \( 2^{nH(Y)} \) and the size of the typical set of \( \mathcal{Y}^n \) given \( \mathcal{X}^n \) is \( 2^{nH(Y|X)} \). We can then upper bound the maximal number of non-overlapping images to be simply

\[
\frac{2^n H(Y)}{2^{nH(Y|X)}} = 2^n I(X; Y)
\]
These correspond to $2^{nI(X;Y)}$ sequences, and using these we can communicate at a rate of

$$\frac{1}{n} \log 2^{nI(X;Y)} = I(X;Y)$$

bits per transmission. Thus we want to choose $X$ that maximizes this rate.

**Preliminaries for Channel Coding Theorem**

Recall that when communicating across a channel, we have

$$W \xrightarrow{\text{Encoder}} X^n \xrightarrow{\text{Channel}} Y^n \xrightarrow{\text{Decoder}} \hat{W} \text{ (estimate of message)}$$

$W$ is drawn from the set $\{1, \ldots, M\}$, the encoder forms $X^n(W)$ and decoder receives $Y^n(W)$ with distribution $p(y^n | x^n(W))$. If $\hat{W} \neq W$, then we say there is an error.

**Question:** What is the maximum number of messages we can send with arbitrarily low probability of error?

Denote a discrete channel by $(\mathcal{X}, p(y|x), \mathcal{Y})$. The $n$-th extension of a DMC is given by $(\mathcal{X}^n, p(y^n | x^n), \mathcal{Y}^n)$ where

$$p(y_k | x_k, y_{k-1}) = p(y_k | x_k), k = 1, \ldots, n$$

We also impose a no feedback condition, i.e. $p(x_k | x_{k-1}, y_{k-1}) = p(x_k | x_{k-1})$ (cannot use previous output to improve next input). Also, $p(y^n | x^n) = \prod_{i=1}^n p(y_i | x_i)$.

**Definition:** An $(M, n)$-code for the channel $(\mathcal{X}, p(y|x), \mathcal{Y})$ consists of

1. An index set $\{1, \ldots, M\}$

2. An *encoding function* for $X^n$: $\{1, \ldots, M\} \to \mathcal{X}^n$, yielding codewords $X^n(1), \ldots, X^n(M)$. We call a codebook the collection of all codewords.

3. A decoding function $g: \mathcal{Y}^n \to \{1, \ldots, M\}$

The *probability of error* for the error code for input $i$ is given by

$$\lambda_i = \Pr\{g(Y^n) \neq i | X^n = X^n(i)\} = \sum_{g(y^n) \neq i} p(y^n | X^n(i))$$

and the maximal probability of error, denoted $\lambda^{(n)}$, is defined to be $\max_{1 \leq i \leq M} \lambda_i$.

The average probability of error, denoted $p_e^{(n)}$ is defined to be $\frac{1}{M} \sum_{i=1}^M \lambda_i$. In particular, we note that $p_e^{(n)} \leq \lambda^{(n)}$. This is the probability of error when $I \sim \text{unif}\{1, \ldots, M\}$.

The *rate* of an $(M, n)$-code is defined to be

$$R = \frac{\log M}{n} \text{ bits/transmission}$$

A rate $R$ is *achievable* if there exists a sequence of $(\lceil 2^{nR} \rceil, n)$-code for which $\lambda^{(n)} \to 0$. Then as $n \to \infty$, the (operational) *capacity* is the supremum of all achievable rates.
The channel coding theorem roughly follows the intuition described earlier, but we need a result about which sequences in the joint \((X^n, Y^n)\) are typical.

**Definition:** The set \(A^{(n)}_\varepsilon\) of *jointly typical sequences* \(\{(X^n, Y^n)\}\) is given by:

\[
A^{(n)}_\varepsilon = \{(x^n, y^n) \in A^{(n)}_{(X,Y),\varepsilon}; x^n \in A^{(n)}_{X,\varepsilon}, y^n \in A^{(n)}_{Y,\varepsilon}\}
\]

(same as usual definition of typical, except we require that the components are also typical). In other words, \((x^n, y^n)\) is typical if:

1. \(|-\frac{1}{n}\log p(x^n) - H(X)| < \varepsilon\)
2. \(|-\frac{1}{n}\log p(y^n) - H(Y)| < \varepsilon\)
3. \(|-\frac{1}{n}\log p(x^n, y^n) - H(X,Y)| < \varepsilon\)

We have the following theorem:

**Theorem 59. (Joint AEP)** Let \((X_i, Y_i)\) be i.i.d. \(\sim p(x, y)\) for \(i = 1, 2, \ldots\) Consider

\[
(X^n, Y^n) \sim p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i)
\]

Then

1. \(\Pr((X^n, Y^n) \in A^{(n)}_\varepsilon) \to 1\) as \(n \to \infty\)
2. \(|A^{(n)}_\varepsilon| \leq 2^{n(H(X,Y)+\varepsilon)}\)
3. Suppose \(\tilde{X}^n, \tilde{Y}^n\) are independent, but have the same marginals as \(X^n, Y^n\), respectively. Then the joint \((\tilde{X}^n, \tilde{Y}^n) \sim p(x^n)p(y^n)\), and

\[
P((\tilde{X}^n, \tilde{Y}^n) \in A^{(n)}_\varepsilon) \leq 2^{-n[I(X;Y) - 3\varepsilon]}
\]

and for large \(n\),

\[
P((\tilde{X}^n, \tilde{Y}^n) \in A^{(n)}_\varepsilon) \geq (1-\varepsilon)2^{-n[I(X;Y) + 3\varepsilon]}
\]

**Proof.** The first two properties follow immediately from the usual AEP. The third is slightly different, and tells us that taking a pair consisting of a typical \(X^n\) and a typical \(Y^n\) does not give a jointly typical sequence \((X^n, Y^n)\). The proof is a direct computation. Note that for elements \((x^n, y^n)\) in the typical set \(A^{(n)}_\varepsilon\), we have that \(p(x^n) \leq 2^{-n[H(X) - \varepsilon]}\) and \(p(y^n) \leq 2^{-n[H(Y) - \varepsilon]}\), and thus

\[
P((\tilde{X}^n, \tilde{Y}^n) \in A^{(n)}_\varepsilon) = \sum_{(x^n, y^n) \in A^{(n)}_\varepsilon} p(x^n)p(y^n)
\]

\[
\leq \sum_{(x^n, y^n) \in A^{(n)}_\varepsilon} 2^{-n[H(X) - \varepsilon]}2^{-n[H(Y) - \varepsilon]}
\]

\[
= 2^{-n[I(X;Y) - 3\varepsilon]}
\]
The lower bound is proved in the same way, using the lower bounds in the properties of the typical set, found below Theorem 33. In particular, \(|A_{\varepsilon}^{(n)}| \geq (1-\varepsilon)2^{n(H(X;Y) - \varepsilon)}\) and \(p(x^n) \geq 2^{n[H(X) + \varepsilon]}\), and thus

\[
P((X^n, Y^n) \in A_{\varepsilon}^{(n)}) \geq (1-\varepsilon)2^{-n[I(X;Y) + 3\varepsilon]}
\]

Now we are ready to present the Channel Coding Theorem (Shannon’s Second Theorem, referenced in Theorem 4).

**Theorem 60. (Channel Coding Theorem)** Let \(C\) be the information capacity \(C = \max_{p(x)} I(X; Y)\). Then all rates below \(C\) are achievable, i.e. if \(R < C\), then there exists a sequence of \([2^n, n]\)-codes for which \(\lambda^{(n)} \to 0\) (the maximum probability of error decays)

Conversely, if \(R > C\), then \(R\) is not achievable, i.e. for any sequence of \([2^n, n]\)-codes for which \(\lambda^{(n)} \to 0\), it must be the case that \(R < C\).

**Ideas (due to Shannon):**

- Allow for arbitrarily small probability of error
- Use the channel many times, exploiting AEP
- Construct a codebook at random, and compute the probability of error, and then show existence of a codebook (nonconstructive proof)

**Week 9 (11/23/2009)**

**Proof. (of Theorem 60)** First we prove *achievability* of any \(R < C\). First we fix \(p(x)\), and generate an \([2^n, n]\)-code as follows: For each \(i \in \{1, \ldots, [2^nR]\}\), pick \(X^n(i)\) independently from the distribution \(p(x^n) = \prod_i p(x_i)\). We then get the codebook

\[
C = \begin{pmatrix}
X^n(1) \\
\vdots \\
X^n([2^nR])
\end{pmatrix}
\]

i.e. a random codebook. We will use this code and compute the probability of error of the following process:

1. Draw \(W \sim \text{unif}\{1, \ldots, [2^nR]\}\), and send \(X^n(W)\) to channel
2. Channel outputs \(Y^n\) from distribution \(p(y^n | x^n = X^n(W)) = \prod_i p(y_i | x_i(W))\) (assuming DMC and no feedback, as stated earlier).
3. Decoder guesses \(\hat{W}\) if \((X^n(\hat{W}), Y^n)\) are jointly typical, and no other codeword is jointly typical with \(Y^n\). If no such codeword exists, report “error”

Then if \(W \neq \hat{W}\), this is also an error. Then we compute the expected probability of error (averaged over possible codebooks):
If $\varepsilon = \{\hat{W} \neq W\}$ then

$$
\Pr(\varepsilon) = \sum_c \Pr(C) P_{\varepsilon}^{(n)}(C) = \sum_c \Pr(C) \frac{1}{2^n R} \sum_{w=1}^{2^n R} \lambda_w(C) = \sum_c \Pr(C) \lambda_i(C) = \hat{P}(\varepsilon | W = 1)
$$

where the third line follows since $\lambda_w(C)$ is the same for all $w$ by symmetry.

Now let $E_i = \{(X^n(i), Y^n) \in A_{\varepsilon}^{(n)}\}$ for $i = 1, \ldots, 2^n R$, the event that the $i$-th codeword is typical with the received sequence. Then the event that there was an error given that the first codeword was sent is simply

$$
\varepsilon = \bigcup_{i=1}^{2^n R} E_i
$$

which has probability upper bounded by the union bound:

$$
\Pr(\varepsilon) \leq \Pr(E_1^c) + \sum_{i=2}^{2^n R} \Pr(E_i)
$$

Given $W = 1$, the input to the channel is $X^n(1)$ and the output is $Y^n$. Note that $(X^n(1), Y^n)$ has distribution given by $\prod p(x_i)p(y_i | x_i)$. By Joint AEP, we know that for sufficiently large $n$, most sequences with this distribution are jointly typical. More precisely, $\Pr\{(X^n(1), Y^n) \in A_{\varepsilon}^{(n)}\} \geq 1 - \varepsilon$ for large $n$, which means that $\Pr(E_1^c) < \varepsilon$ for large $n$.

As for other $E_i$, $i > 1$, we note that $X^n(i)$ and $X^n(1)$ are independent by construction, and thus $X^n(i)$ and $Y^n$ are independent. However, $X^n(i)$ and $Y^n$ have the same marginals as $X^n(i)$ and $Y^n$, $p(x)$ and $p(y)$, and thus by Joint AEP again, we have that

$$
\Pr(E_i) = \Pr\{(X^n(i), Y^n) \in A_{\varepsilon}^{(n)}\} \leq 2^{-n(I(X;Y) - 3\varepsilon)}
$$

Now we have that

$$
\Pr(\varepsilon | W = 1) \leq \Pr(E_1^c) + \sum_{i=2}^{2^n R} \Pr(E_i) \leq \varepsilon + 2^{-n(I(X;Y) - R - 3\varepsilon)}
$$

and if $R < I(X;Y)$, then the second term can be made smaller than $\varepsilon$, for $3\varepsilon < I(X;Y) - R$. Now we can choose $p(x)$ that maximizes $I(X;Y)$, so that this is true for $R < C$. Thus we now have that

$$
\Pr(\varepsilon) \leq 2\varepsilon
$$

for sufficiently large $n$ (and sufficiently small $\varepsilon$). Since this is an average probability over all possible codebooks, there exists a codebook $C^*$ such that $\Pr(\varepsilon | C^*) \leq 2\varepsilon$ (otherwise $\Pr(\varepsilon) > 2\varepsilon$).

What remains is to show that we can make the maximum probability of error small using $C^*$. Although not all the codewords will have a small probability of error, a fraction of them will. In fact, at least half of the codewords must satisfy $\lambda_i(C^*) \leq 4\varepsilon$. To show this, let $r$ be the proportion of codewords satisfying $\lambda_i(C^*) \leq 4\varepsilon$. Then we note that

$$
\Pr(\varepsilon | C^*) = \frac{1}{2^n R} \left( \sum_{\lambda_i \leq 4\varepsilon} \lambda_i(C^*) + \sum_{\lambda_i > 4\varepsilon} \lambda_i(C^*) \right) \geq (1-r)4\varepsilon \geq 2\varepsilon
$$
and thus \( r \geq 1/2 \).

Now we simply throw away the codewords for which \( \lambda_i(C^*) > 4 \varepsilon \). We now have \( 2^{n(R - 1/n)} \) codewords for which \( \lambda^{(n)}(C^*) \leq 4 \varepsilon \) (the max probability is bounded by \( 4 \varepsilon \) for the remaining codewords). Using this set of codewords reduces the rate from \( R \) to \( R - 1/n \), but taking \( n \) sufficiently large this reduction in rate can be made arbitrarily small. Thus, any rate \( R < C \) is achievable.

Conversely, suppose we have a sequence of \( (2^n R, n) \)-codes where \( \lambda^{(n)} \to 0 \), then we show that \( R \leq C \).

First consider a special case. Suppose that \( P_e(n) = 0 \), i.e. \( \lambda_i = 0 \) for all codewords. Then \( g(Y^n) = W \) with probability 1, and \( H(W | Y^n) = 0 \). Assuming \( W \sim \text{unif}\{1, \ldots, 2^n R\} \), we have that

\[
    n R = H(W) = H(W | Y^n) + I(W; Y^n) = I(X^n; Y^n) \leq \sum_{i=1}^n I(X_i; Y_i) \leq nC
\]

The third line follows since \( g(Y^n) = W \) is a one-to-one function, and thus the mutual information does not change. The fourth line follows by the fact that the channel is DMC:

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

and the last line follows from the fact that the mutual information is bounded above by the capacity. This shows that \( R \leq C \).

The general case follows from the fact that \( H(W | Y^n) \) is close to 0 by Fano’s inequality: Above we upper bound \( H(W | Y^n) \) by \( 1 + P_e(n) nR \), which is \( o(n) \), \( (P_e(n) \to 0 \text{ as } n \to \infty) \).

This gives the existence of a code, and establishing good codes that achieve rates close to the capacity is quite difficult. Read 7.11 in the texts for an explicit code, called Hamming Codes. In 7.12 the text discusses what happens if we allow feedback (using the output of the channel of the previous codeword when deciding how to encode the next input). It turns out that the capacity is exactly the same!

In summary, we have established two results. The Source Coding Theorem (Theorem 46) tells us that the rate of compression is lower bounded \( R > H \) by the entropy. The Channel Coding Theorem (Theorem 60) tells us that the rate of communication across the channel is upper bounded \( R < C \) by the channel capacity. In general, to communicate a source across a channel, we can first compress the source with some compression code, and then encode it further with a channel code and send it across the channel. This is a two step process, that allows us to communicate the source across the channel at any rate \( R \) with \( H < R < C \). The question is whether there is another scheme that takes into consideration the source and the channel together, and allows us to achieve better rates of communication. The answer is no, given by the Joint Source-Channel Coding Theorem.
**Theorem 61. (Joint Source-Channel Coding Theorem)** If \( V_1, \ldots, V_n \) is a finite-alphabet stochastic process satisfying AEP, then there exists a joint-source channel code for which \( P_e(n) \to 0 \) if \( H(V) \leq C \). Conversely, if \( H(V) > C \), then the probability of error cannot be made arbitrarily small.

**Proof.** For achievability, we simply use the typical set. Since the source satisfies AEP, there is a typical set \( A^{(n)}_\epsilon \) of size bounded by \( 2^{n(H(V) + \epsilon)} \) which contains most of the probability. We can use \( n(H(V) + \epsilon) \) bits to describe the elements of the typical set, and simply ignore non-typical elements (generate an error if we see a non-typical element). As long as \( H(V) + \epsilon < C \), we can use the channel code to transmit the source.

The probability of error in this case is the probability that either the source sequence is not typical, or the channel decoder makes an error. Each is bounded by \( \epsilon \), and thus the error is bounded by \( 2\epsilon \). Thus, if \( H(V) < C \) then \( P_e(n) \to 0 \).

Conversely, suppose that \( P_e(n) \) is small for a given code, where \( X^{(n)}(V^n) \) is the input to the channel and the output is some \( \hat{V}^n \). Then Fano’s inequality gives us that \( H(V^n | \hat{V}^n) \leq 1 + n P_e(n) \log |V| \). Then using a similar proof to converse of the previous theorem, we have that

\[
H(V) \leq \frac{H(V^n)}{n} \leq \frac{H(V^n | \hat{V}^n) + I(V^n; \hat{V}^n)}{n} \leq \frac{1}{n} + P_e(n) \log |V| + \frac{I(V^n; Y^n)}{n} \to C
\]

and thus \( H(V) \leq C \). \( \square \)

So the separation of source and channel codes in theory is good. But in practice, we may not want codes that use block lengths that are so large. Also, the channel may vary with time. In these cases it may then be important to consider the source and channel jointly.

**Differential Entropy**

Now we move onto continuous time random variables, to discuss the Gaussian channel (second most commonly used channel).

First a few definitions. Let \( F(x) = \Pr(X \leq x) \) be the cdf or a random variable \( X \). If \( F \) is continuous, then we say that \( X \) is a continuous random variable. Then \( f(x) = F'(x) \) is the probability density function (pdf) for \( X \), and \( \int f = 1 \). Let \( S_X = \{ x : f(x) > 0 \} \), the support set of \( X \).

Then we define the **differential entropy** of a continuous r.v. \( X \) with pdf \( f(x) \) to be

\[
h(X) = -\int_S f(x) \log f(x) dx = -\int f(x) \log^+ f(x) dx
\]

when the limit exists. Note this only depends on the density.

**Example 62.** For the uniform distribution \( f(x) = \frac{1}{a} 1_{[0,a]} \), we have that

\[
h(X) = -\int_0^a \frac{1}{a} \log \frac{1}{a} dx = \log a
\]

In particular, if \( a < 1 \), then \( h(X) < 0 \), so that differential entropy may be negative.
Example 63. For the normal distribution $X \sim N(0, \sigma^2)$, with $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$, we have that

$$h(X) = -\int_{-\infty}^{\infty} f(x) \left[ \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{x^2}{2\sigma^2} \right]$$

$$= \frac{1}{2} \ln (2\pi\sigma^2) + \frac{1}{2\sigma^2} \mathbb{E}(X^2)$$

$$= \frac{1}{2} \ln (2\pi\sigma^2) + \frac{1}{2}$$

$$= \frac{1}{2} \ln (2\pi e\sigma^2)$$

in terms of nats. If we used the base 2 logarithm instead, we would divide the entire expression by $\ln 2$ and this gives us $\frac{1}{2} \log_2 (2\pi e\sigma^2)$ bits.

Now many of the properties of entropy for discrete random variables carry over. For instance, AEP still follows from the law of large numbers:

Theorem 64. Let $X_1, ..., X_n$ be i.i.d $\sim f(x)$. Then

$$-\frac{1}{n} \log f(x_1, ..., x_n) \to \mathbb{E}(-\log f(X)) = h(X)$$

Thus for large $n$, $f(x_1, ..., x_n) \approx 2^{-n h(X)}$, i.e. almost constant.

Then as before we define the typical set $A_\varepsilon^{(n)}$ with respect to $f(x)$ to be

$$A_\varepsilon^{(n)} = \{ (x_1, ..., x_n) : \left| -\frac{1}{n} \log f(x_1, ..., x_n) - h(X) \right| \leq \varepsilon \}$$

To quantify the size of the typical set, we talk about the volume (Lebesgue measure) of the set instead of the cardinality. Then we have the following properties of the typical set:

Proposition 65. The typical set $A_\varepsilon^{(n)}$ has the following properties:

1. $\Pr(A_\varepsilon^{(n)}) > 1 - \varepsilon$ for large $n$

2. $\text{Vol}(A_\varepsilon^{(n)}) \leq 2^{n(h(X)+\varepsilon)}$ for all $n$

3. $\text{Vol}(A_\varepsilon^{(n)}) \geq (1 - \varepsilon)2^{n(h(X)-\varepsilon)}$ for large $n$

The proofs are exactly the same as before (now dealing with the integral instead of the sum). So the volume of the typical set is roughly $2^{n h(X)}$ for large $n$, and per dimension this gives an interpretation for $2^{h(X)}$ as an “effective sidelength” or “radius”.

The joint differential entropy of $(X, Y) \sim f(x, y)$ is as expected, $h(X, Y) = \mathbb{E}(-\log(f(X, Y)))$, and follows the chain rule $h(X, Y) = h(X | Y) + h(Y) = h(Y | X) + h(X)$.

Example 66. We compute the differential entropy of a multivariate Gaussian $(X_1, ..., X_n) \sim N(\mu, K)$ where $K$ is the covariance matrix, and the joint is

$$f(x_1, ..., x_n) = f(x) = \frac{1}{\sqrt{(2\pi)^n|K|}} e^{-\frac{1}{2}(x-\mu, K^{-1}(x-\mu))}$$

47
Then

$$h(X_1, \ldots, X_n) = \frac{1}{2} \log[(2\pi e)^n |K|] \text{ bits}$$

The computation is very straightforward:

$$h(X_1, \ldots, X_n) = -\int f(x_1, \ldots, x_n) \left[ \ln \frac{1}{\sqrt{(2\pi)^n |K|}} - \frac{1}{2} \langle x - \mu, K^{-1}(x - \mu) \rangle \right] dx$$

$$= \frac{1}{2} \ln[(2\pi)^n |K|] + \frac{1}{2} \sum_{ij} E[(x_i - \mu_i)(x_j - \mu_j)] K_{ij}^{-1}$$

$$= \frac{1}{2} \ln[(2\pi)^n |K|] + \frac{1}{2} \sum_{ij} K_{ji} K_{ij}^{-1}$$

$$= \frac{1}{2} \ln[(2\pi)^n |K|] + \frac{1}{2} \sum_j I_{jj}$$

$$= \frac{1}{2} \ln[(2\pi e)^n |K|]$$

The relative entropy, or the Kullback-Liebler distance between two pdfs $f(x), g(x)$ is

$$D(f \parallel g) = \int_{\text{supp } f} f(x) \log \frac{f(x)}{g(x)} dx$$

Note that if the set where $g(x) = 0$ and $f(x) > 0$ has positive measure, then the distance is $\infty$. As with the discrete case, the relative entropy still satisfies $D(f \parallel g) \geq 0$ with $D(f \parallel g) = 0$ if and only if $f = g$ almost everywhere.

The mutual information is as before,

$$I(X; Y) = D(f(x, y) \parallel f(x)g(y)) = \int f(x, y) \log \frac{f(x, y)}{f(x)g(y)} dx dy$$

and satisfies $I(X; Y) = h(Y) - h(Y \mid X) = h(X) - h(X \mid Y)$. Also $I(X; Y) \geq 0$ with equality if and only if $X, Y$ are independent. Furthermore, $h(Y \mid X) \leq h(Y)$, and so conditioning reduces differential entropy, and combined with the chain rule we have that

$$h(X_1, \ldots, X_n) = \sum_{i=1}^n h(X_i \mid X^{i-1}) \leq \sum_{i=1}^n h(X_i)$$

**Week 10**

First we start with more properties about differential entropy:

**Proposition 67.**

1. $h(X + c) = h(X)$, i.e. entropy is translation invariant.

2. $h(aX) = h(X) + \log |a|$ for any constant $a$, and similarly, for vector valued r.v. $X$ and invertible matrices $A$, we have $h(AX) = h(X) + \log |A|$.
Proof. The first statement follows from the fact that the density of \(X + c\) is the same as \(X\), except shifted by \(c\). Then a simple change of variables verifies the first statement. For the second statement, we simply use the fact that the density of \(aX\) is just \(\frac{1}{|a|} f(x/a)\) where \(f\) is the density of \(X\). Then the result follows from change of variables. The same proof holds for vector valued matrices \(A\). \(\square\)

The following proposition is useful for maximization problems that occur in computing capacity. First, we have Hadamard’s inequality:

**Theorem 68. (Hadamard’s Inequality)** Let \(K\) be a symmetric, positive definite matrix. Then

\[
|K| \leq \prod_{i=1}^{n} K_{ii}
\]

**Proof.** Let \(X \sim N(0, K)\) (which in particular means that \(X_i \sim N(0, K_{ii})\)). Then we use the fact that \(h(X) \leq \sum_{i=1}^{n} h(X_i)\) to get that

\[
\frac{1}{2} \log((2\pi e)^n |K|) \leq \sum_{i=1}^{n} \frac{1}{2} \log(2\pi e K_{ii}) = \frac{1}{2} \log \left( (2\pi e)^n \prod_{i=1}^{n} K_{ii} \right)
\]

and thus \(|K| \leq \prod_{i=1}^{n} K_{ii}\). \(\square\)

The next result shows that among all vector valued random variables with covariance matrix \(K\), the random variable that maximizes entropy is the Gaussian \(N(0, K)\).

**Theorem 69.** Suppose \(X \in \mathbb{R}^n\) is a vector valued r.v. with 0 mean and covariance matrix \(K = \mathbb{E}(X X^T)\). Then

\[
h(X) \leq \frac{1}{2} \log((2\pi e)^n |K|)
\]

with equality if and only if \(X \sim N(0, K)\)

**Proof.** The result follows from computing the relative entropy of the random variable compared with the Gaussian random variable, resulting in

\[
0 \leq D(g \| f) = -h(X) + \frac{1}{2} \ln((2\pi e)^n |K|)
\]

The computation is similar to the computation of the entropy of a joint Gaussian. \(\square\)

**Gaussian Channels**

First, let’s consider a noiseless continuous channel \(Y = X\), where \(X \in \mathcal{X} \subset \mathbb{R}\). If \(\mathcal{X}\) is not discrete then the capacity is \(\infty\) (we can use arbitrarily many symbols, so the channel looks like a noiseless channel with \(n\) inputs, with capacity \(\log n\)).

Now suppose \(Y = X + Z\) where \(Z\) is Gaussian noise, distributed \(N(0, \sigma^2_X)\), where \(X \in \mathbb{R}\). Note that the capacity with no restrictions on \(X\) is still infinite. If we let \(\mathcal{X}\) be discrete, evenly spaced on \(\mathbb{R}\), then we have a channel with infinitely many inputs and whose probability of error decays as the spacing increases. However, this is not a realistic model, since for instance we would require arbitrarily large voltages. A more realistic constraint is some sort of power constraint, for instance a **peak power constraint** \(X_i^2 \leq P\) for \(i = 1, \ldots, n\) (\(n\) uses of the channel), or a more relaxed **average power constraint** \(\frac{1}{n} \sum X_i^2 \leq P\).
Example 70. Consider such a constraint for 1 use of the channel, and suppose further that we are sending 1 bit, with the constraint that \(X^2 \leq P\). Then we use the symbols \(\{ \pm \sqrt{P} \}\), and given the output \(Y = X + Z\), we decode to the nearest symbol; that is, if \(Y < 0\), then decoder outputs \(\hat{X} = -\sqrt{P}\), otherwise \(\hat{X} = \sqrt{P}\). The probability of error if \(X = -\sqrt{P}\) is the probability that \(Z > \sqrt{P}\), and by symmetry the same probability of error holds for \(X = \sqrt{P}\). Thus

\[
P(\text{error}) = Q\left(\frac{\sqrt{P}}{\sigma_N}\right)
\]

where \(Q(z) = \int_{x}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt\). Letting \(p_e\) be the probability of error, we have reduced the channel to a Binary symmetric channel with crossover probability \(p_e\). The capacity is \(1 - H(p_e) \leq 1\).

Consider now \(n\) uses of the channel. The output \(Y_i = X_i + Z_i\) where \(Z_i\) are i.i.d distributed \(N(0, \sigma_N^2)\) and \(X_i, Z_i\) are independent. We impose an average power constraint \(P\) on \(X\), so \(\frac{1}{n} \sum_{i=1}^{n} X_i^2 \leq P\), and we want to know what the capacity is. We define the information capacity of a Gaussian channel with average power constraint \(P\) to be

\[
C = \max_{X} \frac{1}{n} \sum_{i=1}^{n} I(X; Y)
\]

and we will show the analogue to the Channel-Coding Theorem for Gaussian channels. First we compute what the information capacity is for the Gaussian channel:

\[
I(X; Y) = h(Y) - h(Y | X)
\]

\[
= h(Y) - h(X + Z | X)
\]

\[
= h(Y) - h(Z | X) \quad \text{(invariance of entropy)}
\]

\[
= h(Y) - h(Z) \quad \text{(X, Z independent)}
\]

\[
= h(Y) - \frac{1}{2} \log(2\pi e \sigma_N^2)
\]

Now we note that since \(E(X^2) \leq P\), then

\[
E(Y^2) = E[(X + Z)^2]
\]

\[
= E(X^2) + E(Z^2) + 2E(X)E(Z) \quad \text{(X, Z independent)}
\]

\[
\leq P + \sigma_N^2
\]

and thus we have that \(h(Y) \leq \frac{1}{2} \log(2\pi e (P + \sigma_N^2))\) by Theorem 69, and so

\[
I(X; Y) \leq \frac{1}{2} \log \left(1 + \frac{P}{\sigma_N^2}\right)
\]

with equality if and only if \(Y \sim N(0, P + \sigma_N^2)\), which is the case if we choose \(X \sim N(0, P)\). Note that \(E(X^2) \leq P\) so we have achieved the upper bound. Thus \(C = \frac{1}{2} \log \left(1 + \frac{P}{\sigma_N^2}\right)\), and \(\frac{P}{\sigma_N^2}\) is the SNR (signal to noise ratio).

Read Theorem 9.1.1 for the full proof of the channel capacity theorem, but the definitions of codes and achievable rates remain the same, and the proof idea is also the same.

We will instead sketch an intuitive reason for why the information capacity gives the operational capacity, and even this is the same as before. If we send input \(x^n\), then by AEP \(y^n\) will lie in a set of volume \(2^{n \log(Y | X)}\) with high probability and for large \(n\). We want to choose inputs \(x^n\) so that the typical set for \(y^n | x^n\) do not overlap. Since the typical set of \(y^n\) lies in a set of volume \(2^{n I(X; Y)}\), so the maximum number of inputs with no overlap is bounded above by

\[
2^{n H(Y)} / 2^{n H(Y | X)} = 2^n I(X; Y)
\]
which is maximized by the information capacity.

**Parallel Gaussian Channels**

Now we consider the problem of using Parallel Gaussian Channels, where we can send input $X_1, ..., X_n$ simultaneously through $n$ Gaussian channels, and get output $Y_i = X_i + Z_i$ where $Z_j = N(0, \sigma_{N_j}^2)$ and independent, and $X_i, Z_i$ are independent. Denote $\sigma_N^2 = \sum_j \sigma_{N_j}^2$. We impose a common power constraint $\sum_j E(X_j^2) \leq P$. Then the information capacity is

$$C = \max_{X_i, \sum_j E(X_j^2) \leq P} I(X_1, ..., X_n; Y_1, ..., Y_n)$$

Now we have

$$I(X_1, ..., X_n; Y_1, ..., Y_n) = h(Y_1, ..., Y_n) - h(Y_1, ..., Y_n | X_1, ..., X_n)$$

$$= h(Y_1, ..., Y_n) - h(Z_1, ..., Z_n)$$

$$= h(Y_1, ..., Y_n) - \sum_{j=1}^n h(Z_j)$$

$$\leq \sum_{j=1}^n [h(Y_j) - h(Z_j)]$$

$$\leq \sum_{j=1}^n \frac{1}{2} \log \left(1 + \frac{P_j}{\sigma_{N_j}^2}\right)$$

where the second to last inequality is equality if $Y_j$ are independent, and the last line equality if $Y_j$ are Gaussian $N(0, P_j + \sigma_{N_j}^2)$, which is the case if we choose $X_j \sim N(0, P_j)$. The $P_j$ are to be determined. Thus the problem reduces to maximizing $\sum_j \frac{1}{2} \log \left(1 + \frac{P_j}{\sigma_{N_j}^2}\right)$ such that $\sum_j P_j \leq P$, $P_j \geq 0$. Since we have a concave function that increases with $P_j$, we can reduce the constraint further to $\sum_j P_j = P$. Now Lagrange multipliers shows that the optimal $P_j$ satisfy

$$\sigma_{N_j}^2 + P_j = \text{const}$$

Since $P_j = \text{const} - \sigma_{N_j}^2$ may not be positive, we need a slight adjustment. It turns out that $P_j = (\text{const} - \sigma_{N_j}^2)^+$ satisfies the Karush-Kuhn-Tucker conditions for optimality in the presence of the additional inequality constraint $P_j \geq 0$ (see wikipedia; in particular, “stationarity” constraint yields $\sigma_{N_j}^2 + P_j = \text{const}$, with $\mu = 0$, and this satisfies “primal and dual feasibility”, etc). We then choose the constant so that $\sum_j P_j = P$, and pictorially we can view this as a “water-filling procedure”:

![Water-filling diagram](image)

Note that in the picture $P_2 = 0$. We keep raising the constant until the area of the shaded regions gives $P$, hence the name.
Non-white Gaussian noise channels

In this scenario, we again have $n$ parallel Gaussian channels, but this time the noise is correlated, $(Z_1, ..., Z_n) \sim N(0, K_N)$, though we still have $Z_j, X_j$ independent. We impose the usual power constraint $\sum_{j=1}^n E(X_j^2) \leq P$, which we can now write as $\text{tr}(K_X) \leq P$. Computing the capacity, we again have

$$I(X_1, ..., X_n; Y_1, ..., Y_n) = h(Y_1, ..., Y_n) - h(Z_1, ..., Z_n) = h(Y_1, ..., Y_n) - \frac{1}{2} \log(2\pi e)^n |K|$$

Now we note that the covariance of $Y$ is $K_Y = \mathbb{E}(YY^T) = K_X + K_N$ using $Y = X + Z$ and that $X, Z$ are independent. The entropy of $Y$ is thus maximized when $Y \sim N(0, K_X + K_N)$, which is achieved if we choose $X \sim N(0, K_X)$. Then the problem reduces to maximizing $h(Y) = \frac{1}{2} \log(2\pi e)^n |K_X + K_N|$ such that $\text{tr}(K_X) \leq P$. Now we diagonalize $K_N = QAQ^T$ with unitary $Q$, then

$$|K_X + K_N| = |Q^TK_XQ + A|$$

and noting that $\text{tr}(Q^TK_XQ) = \text{tr}(Q^TK_X) = \text{tr}(K_X)$ the problem reduces to maximizing $|A + \Lambda|$ such that $\text{tr}(A) \leq P$. By Hadamard’s Theorem,

$$|A + \Lambda| \leq \prod_{j=1}^n (A_{jj} + \Lambda_{jj})$$

with equality if and only if $A + \Lambda$ is diagonal. Thus we choose $A$ to be diagonal, and the problem finally reduces to maximizing $\prod_{k=1}^n (A_{kk} + \Lambda_{kk})$ such that $\sum_j A_{jj} \leq P$, which is the same maximization problem as in the parallel white Gaussian noise channel. Thus the maximum is achieved when $A_{jj} = (\text{const} - \Lambda_{jj})^+$ where the constant is chosen so that $\sum_j A_{jj} = P$.

**Side Note for Continuous Time Channels**: Consider a channel $Y(t) = X(t) + Z(t)$ where $Z$ is now white Gaussian noise with power spectral density $N_0/2$, and let $W$ be the channel bandwidth, and $P$ be the input power constraint. Then the capacity of the channel is

$$C = W \log\left(1 + \frac{P}{N_0W}\right)$$

bits per second. It is interesting to note that we still have a “water-filling” procedure to choose the power spectral density of the input.

**Week 11**

(12/7/2009)

**Rate Distortion Theory**

We revisit source coding theorems, where we had the earlier assumption of **lossless** compression, where we wanted $P(X \neq \hat{X})$ to be small, i.e. we wanted exact recovery with small probability of error. In this case we know that the lower bound for compression rates is the entropy $H(X)$, and is achieved by Shannon-type codes or AEP-based codes with arbitrarily small probability of error (zero error for Shannon/Huffman codes).

Instead, we consider **lossy** compression, where we allow $X$ and $\hat{X}$ to differ by a little. We use a distortion function $d$ to determine how close two words are. Thus we now find a compression scheme so that $P(d(X, \hat{X}) < \varepsilon)$ is small for $\varepsilon$ small. This is a natural assumption for analog sources, for instance representing real numbers, which we may only represent with finitely many digits, and also for representing images, where smearing a few pixels does not affect the overall image by too much.
**Example 71.** Suppose $X \sim N(0, \sigma^2)$, and we consider the distortion function $d(x, \hat{x}) = (x - \hat{x})^2$. First we consider representing $X$ with a single bit (compression rate $R = 1$ bit). This means that once we observe $X$, we encode as one of two values $\{a, b\}$, which we denote $\hat{X}$. We want to minimize $E(d(X, \hat{X}))$. By symmetry considerations, we should choose $b = -a$, and choose

$$
\hat{X}(x) = \begin{cases} 
a & x > 0 \\
-a & x < 0 
\end{cases}
$$

In this case,

$$
E(|X - \hat{X}(X)|^2) = \int_0^\infty (x-a)^2 f(x) dx + \int_{-\infty}^0 (x+a)^2 f(x) dx
$$

and the optimal choice of $a$ is

$$
a = E(X \mid X > 0) = \sigma \sqrt{\frac{2}{\pi}}
$$

(The conditional expectation minimizes mean square error. Linking this to the abstract probability theory and conditional expectation, we wish to find another estimator random variable $\hat{X}$, which is constant on $\{X < 0\}$ and $\{X > 0\}$, such that $E(|X - \hat{X}|^2)$ is minimized. In other words, $\hat{X}$ is measurable with respect to the $\sigma$-field $\Sigma = \{\emptyset, \{X < 0\}, \{X > 0\}, \mathbb{R}\}$, and the minimizer is found by conditional expectation $\hat{X} = E(X \mid \Sigma)$. In this case, this is exactly

$$
\hat{X} = \begin{cases} 
E(X \mid X > 0) = \sigma \sqrt{\frac{2}{\pi}} & X > 0 \\
E(X \mid X < 0) = -\sigma \sqrt{\frac{2}{\pi}} & X < 0 
\end{cases}
$$

Now if we consider two bits instead, we consider the following procedure. Pick four points arbitrarily $\{a, b, c, d\}$. This corresponds to four regions, determined by which of the four points is closest, $A, B, C, D$, where all points in $A$ are closest to $a$, and etc. However, given the four regions $A, B, C, D$, the four points $a, b, c, d$ may not be the best encoding points that minimize $E(|X - \hat{X}|^2)$. We know from above that this is given by the conditional means over each region ($E(X \mid \Sigma)$, where $\Sigma = \sigma(A, B, C, D)$, the $\sigma$-algebra generated by the four sets). The conditional means gives us four new points $a', b', c', d'$, which correspond to new regions $A', B', C', D'$. We can then iterate this procedure to obtain the best four points. This is known as the **Lloyd Algorithm**. We do not discuss convergence here.

We now turn to the general setting, and introduce a few definitions. In the above example we dealt with encoding a single random variable. In general we will work as in the lossless case to encode $n$ i.i.d copies of a random variable, or a stationary stochastic process. To simplify discussion, we consider the i.i.d case. Given $X_1, \ldots, X_n$ i.i.d $\sim p(x)$, with $x \in \mathcal{X}$, we have

$$
X^n \longrightarrow \text{Encoder} \longrightarrow f_n(X^n) \in \{1, \ldots, 2^n R\} \longrightarrow \text{Decoder} \longrightarrow g_n(f_n(X^n)) = \hat{X}^n
$$

The **distortion function** is a mapping $d: \mathcal{X} \times \hat{\mathcal{X}} \rightarrow \mathbb{R}^+$ where $d(x, \hat{x})$ is the cost of representing the source outcome $x$ as the reproduction outcome $\hat{x}$. In most cases $\mathcal{X} = \hat{\mathcal{X}}$. Common distortion functions:

- **Hamming Distortion**

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

53
and in this case $\mathbb{E}(d(X, \hat{X})) = P(X \neq \hat{X})$

- **Squared Error Distortion**

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

A distortion function is allowed to be $\infty$, in which case compression must avoid the $\infty$ cases. A **bounded distortion** function is one which does not take $\infty$. We can extend distortion functions to look at the joint as well (or blocks of random variables). The **single-letter distortion function** is

\[ d(X^n, \hat{X}^n) = \frac{1}{n} \sum_{i=1}^{n} d(X_i, \hat{X}_i) \]

which is essentially the “average per-letter distortion”.

Given the distortion function, a $(2^nR, n)$ rate distortion code consists of an encoding function $f_n$ mapping $X^n \rightarrow \{1, \ldots, 2^{nR} \}$ and a decoding function $g_n$ mapping $\{1, \ldots, 2^{nR} \} \rightarrow \hat{X}^n$. The distortion of the code is

\[ D = \mathbb{E}(d(X^n, g_n(f_n(X^n)))) = \sum_{X^n} p(x^n) d(x^n, g_n(f_n(x^n))) \]

we represent $g_n(i)$ as $\hat{X}^n(i), i = 1, \ldots, 2^{nR}$.

**Definition 72.** A rate-distortion pair $(R, D)$ is **achievable** if there exists a sequence of $(2^{nR}, n)$ rate distortion codes with $\lim_{n \rightarrow \infty} \mathbb{E}(d(X^n, g_n(f_n(X^n)))) \leq D$. The **rate distortion region** is the closure of the set of achievable $(R, D)$ pairs. The **rate distortion function** $R(D)$ is the infimum of $R$ such that $(R, D)$ is achievable and the **distortion rate function** $D(R)$ is the infimum of $D$ such that $(R, D)$ is achievable.

It is always the case that $R(D)$ is convex, though we will not prove this here. As before, it turns out that $R(D)$ will coincide with the **information rate distortion function**

\[ R(D) = \min_{I(X, \hat{X})} \mathbb{E}_{d(X, \hat{X}) \leq D} I(X, \hat{X}) \]

This is the rate-distortion theorem, which we also do not prove, but the idea is similar to the channel coding theorem proof with a random codebook. Instead we focus on how to compute the information rate distortion function. Now we are fixing the source $p(x)$ and minimizing the mutual information over all possible choices of the conditional distribution $p(\hat{x} \mid x)$. The key is in the observation that $I(X; \hat{X})$ is a convex function of $p(\hat{x} \mid x)$, so again we only need to look for critical points to find a global minimum.

**Binary Source / Hamming Distribution**

Let $X \sim \text{Bern}(p)$, where $X = 1$ with probability $p$, and let $p < 1/2$. We compute the mutual information

\[
I(X; \hat{X}) = H(X) - H(X \mid \hat{X})
= H(p) - H(X \oplus \hat{X} \mid \hat{X})
\geq H(p) - H(X \oplus \hat{X})
\]

using $Z = X \oplus \hat{X}$ since $Z = 1$ if and only if $X \neq \hat{X}$ (relates to distortion function), and we note that

\[
P(Z = 1) = P(X \neq \hat{X}) = \mathbb{E}(d(X, \hat{X})) \leq D
\]
so that \( I(X; \hat{X}) \geq H(p) - H(D) \) if \( D \leq 1/2 \) (recall \( H(p) \) is concave with maximum at \( 1/2 \)). To meet the lower bound, we need to find \( p(x) \) such that \( H(X | \hat{X}) = H(D) \). In order to talk about \( H(X | \hat{X}) \), we consider a “test” channel, with input \( \hat{X} \) and output \( X \). Note that Bayes’ rule allows us to invert \( p(\hat{x} | x) \) to get \( p(x | \hat{x}) \), which corresponds to the test channel and is relevant for \( H(X | \hat{X}) \). Note that \( H(X | \hat{X}) = H(D) \) in the case of BSC(D) channel. Then if we set \( p(x | \hat{x}) = \begin{cases} D & \hat{x} \neq x \\ 1-D & \hat{x} = x \end{cases} \), as in BSD, if the corresponding \( p(\hat{x}) \) is a valid probability distribution, then we have achieved the minimum. We have the following picture:

\[
\begin{array}{ccc}
\hat{X} & (1-a) & 0 \\
\downarrow & 1-D & \rightarrow 0 \\
\uparrow & a & \rightarrow 1 \\
X & 0 & (1-p) \\
\end{array}
\]

so that \( a = P(\hat{X} = 1) \). Then we compute that \( a(1-D) + (1-a)D = p \), or \( a = \frac{p-D}{1-2D} \), which is valid if \( D \leq p < \frac{1}{2} \). In this case we have achieved the minimum mutual information, so the rate distortion function is \( R(D) = H(p) - H(D) \). In the case \( D > p \), we note that simply setting \( \hat{X} = 0 \) always (i.e. \( p(\hat{x} | x) = 1 \) if \( \hat{x} = 0 \)). In this case \( P(X \neq \hat{X}) = p < D \) already, and we have achieved transmission at a rate of 0, which is the minimum possible value for \( I(X; \hat{X}) \). Thus \( R(D) = 0 \) for \( D > p \). In summary

\[
R(D) = \begin{cases} 
H(p) - H(D) & 0 \leq D \leq \min \{ p, 1-p \} \\
0 & \text{else}
\end{cases}
\]

**Gaussian Source with Squared-Error Distortion**

Let \( X \sim N(0, \sigma^2) \) as in the previous example. Computing the information rate distortion function, we have

\[
I(X; \hat{X}) = h(X) - h(X | \hat{X}) \\
= \frac{1}{2} \log(2\pi e \sigma^2) - h(X - \hat{X} | \hat{X}) \\
\geq \frac{1}{2} \log(2\pi e \sigma^2) - h(X - \hat{X})
\]

considering \( Z = X - \hat{X} \), we note \( \mathbb{E}(Z^2) \leq D \) from the constraint. We then maximize \( h(Z) \) subject to the constraint that \( \mathbb{E}(Z^2) \leq D \). Then the Gaussian \( Z \sim N(0, D) \) will maximize \( h(Z) \), and so

\[
I(X; \hat{X}) \geq \frac{1}{2} \log(2\pi e \sigma^2) - \frac{1}{2} \log(2\pi e D) = \frac{1}{2} \log \left( \frac{\sigma^2}{D} \right)
\]

Studying the bounds above, we note that equality is achieved if \( Z = X - \hat{X} \) is independent of \( \hat{X} \) and \( Z \sim N(0, D) \). We are given that \( X \sim N(0, \sigma^2) \). This shows that \( \hat{X} \sim N(0, \sigma^2 - D) \) since \( \hat{X} = Z - X \). \( \hat{X} \) is a valid distribution so long as \( \sigma^2 \geq D \), and in this case we can achieve the minimum. Now if \( \sigma^2 < D \), as in the previous case if we simply set \( \hat{X} = 0 \) the distortion is \( \sigma^2 < D \) already with a rate of 0. Thus

\[
R(D) = \begin{cases} 
\frac{1}{2} \log \left( \frac{\sigma^2}{D} \right) & D \leq \sigma^2 \\
0 & D > \sigma^2
\end{cases}
\]

55
Note that \( D(R) \), computed by inverting \( R(D) \) is \( D(R) = \sigma^2 2^{-2R} \). This can be interpreted by saying that for "every additional bit we want to send, the distortion lowers by a factor of \( 1/4 \)." For \( R = 1 \) we have \( D = \sigma^2/4 \), and in the earlier example where we used 1-shot, 1-bit quantization, \( D \approx 0.36\sigma^2 \) (but this tells us that we can lower the distortion using some other method).

### Independent Gaussian Random Variables with Single-Letter Distortion

Now consider \( X_1, ..., X_m \) independent with \( X_i \sim N(0, \sigma_i^2) \). The distortion function is

\[
d(X^m, \hat{X}^m) = \sum_{i=1}^{m} (X_i - \hat{X}_i)^2
\]

We can compute the information rate distortion function similarly to the channel capacity for independent Gaussian noise. We have that

\[
I(X^m, \hat{X}^m) = h(X^m) - h(X^m | \hat{X}^m) = \sum_{i=1}^{m} h(X_i) - \sum_{i=1}^{m} h(X_i | X_i^{i-1}, \hat{X}^m) \\
\geq \sum_{i=1}^{m} h(X_i) - \sum_{i=1}^{m} h(X_i | \hat{X}_i) = \sum_{i=1}^{m} I(X_i; \hat{X}_i)
\]

where equality is achieved if given \( \hat{X}_i \), \( X_i \) is independent of \( X_i^{i-1} \) and other \( \hat{X}_j \neq i \). We can lower bound further by

\[
\sum_{i=1}^{m} I(X_i; \hat{X}_i) \geq \sum_{i=1}^{m} R_i(D_i) = \sum_{i=1}^{m} \frac{1}{2} \log \left( \frac{\sigma_i^2}{D_i} \right)
\]

where equality is achieved if we use independent Gaussians \( X_i - \hat{X}_i \sim N(0, D_i) \), and the \( D_i \) are to be chosen. Since \( D_i = \mathbb{E}[(X_i - \hat{X}_i)^2] \), we have the constraint \( \sum_i D_i \leq D \).

The problem therefore reduces to minimizing \( \sum_{i=1}^{m} \frac{1}{2} \log \left( \frac{\sigma_i^2}{D_i} \right) \) subject to the constraint \( \sum_i D_i \leq D \).

Ignoring the positive part by \( \log^+ \) with \( \log \), the Lagrangian yields \( D_i = \lambda \). Recall that \( \hat{X}^m \sim N(0, \sigma_i^2 - D_i) \), and so we need that \( \lambda \leq \sigma_i^2 \). We then verify KKT for the solution \( D_i = \begin{cases} \sigma_i^2 & \lambda \geq \sigma_i^2 \\ \lambda & \lambda < \sigma_i^2 \end{cases} \), and choose \( \lambda \) such that \( \sum_i D_i = D \). This can be interpreted as a “reversed” water-filling procedure:

![Water-filling diagram](image)

So we have \( R_i^* = \frac{1}{2} \log^+ \frac{\sigma_i^2}{D_i} \) and \( R(D) = \sum_i R_i^* \).