Basic Channels

Channels: the construct through which information is passed from a sender to a receiver.

Binary symmetric channel: each symbol sent is a bit, that is flipped with probability $p$.

Erasure channel: each symbol sent is a bit; instead of a bit, a “?” arrives with probability $p$. 
Channel Capacity

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

Intuition: I am going to choose a probability distribution \( p(x) \) to send symbols from \( X \). The entropy of the initial input is \( H(X) \); the entropy at the receiver side, where \( Y \) is received, is \( H(X) - H(X | Y) \). This will be shown to be equivalent to average number of useful bits received per bit sent.

Example: binary symmetric channel. Capacity is \( \max_{p(x)} H(Y) - H(Y | X) = 1 - H(p) \).

Example: erasure channel. Capacity is \( \max_{p(x)} H(X) - H(X | Y) = 1 - p \).
Shannon’s Theorem

Shannon’s theorem: By using a long enough block, you can achieve transmission rates as close as you like to capacity \( C - \epsilon \) with arbitrarily small \( \epsilon \) probability of error.

Proof (for a binary symmetric channel): for large blocks \( n \), choose about \( 2^{Cn} \) codewords. We expect about \( pn \) errors. Take the received block; if just one codeword differs from the received block in about \( pn \) places, output that. Most of the time this will be correct.
Mathematical preliminaries

Stirling’s approximation:

\[ n! = \sqrt{2\pi n} \left( \frac{n}{e} \right)^n \left( 1 + \frac{1}{12n} + O \left( \frac{1}{n^2} \right) \right) \]

Chernoff bounds: Let \( Y = X_1 + X_2 \ldots + X_n \) be the sum of \( n \) independent 0/1 random variables with \( E[Y] = \mu \). Then for \( \epsilon < 1 \),

\[ Pr(|Y - \mu| > \epsilon \mu) < 2e^{-\epsilon^2 \mu/3}. \]

Corollary: For a block of \( n \) bits, each in error with probability \( p \), let \( Y \) be the number of errors.

\[ Pr(|Y - np| > \epsilon n) < 2e^{-\epsilon^2 n/3p^2}. \]
Proof of Shannon’s Theorem: Binary Symmetric Channel

We demonstrate for $R = C - \gamma$ that there exists a code with rate $R$ and maximum error probability $\gamma$. Consider blocks of $n$ bits.

Choose $2^{nR}$ codewords randomly. If decoding is successful, receiver gets $R$ bits of information for every $n$ sent. Decoder will search through a list of codewords, until finding one with between $(p - \gamma)n$ and $(p + \gamma)n$ errors. Decoder could fail if there are too many or few errors, or there is more than once codeword “near” the received block.
Problem 1: Too many/few errors
By Chernoff’s bound

\[ Pr(|Y - np| > \gamma n) < 2e^{-\gamma^2 n/3p^2}. \]

Problem 2: Two possible codewords
Consider a codeword \( x_1 \) to be sent and the transmission \( y \) received. What is the probability that something that differs from \( y \) in \((p - \gamma)n\) to \((p + \gamma)n\) places is also a codeword?

For another codeword \( x_2 \), probability is at most

\[
\sum_{k=[n(p-\gamma)]}^{[n(p+\gamma)]} \binom{n}{k}/2^n
\]
Approximate by one term, and Stirling’s formula

\[
\binom{n}{pn}/2^n \approx \left(\frac{n}{e}\right)^n / \left(\frac{np}{e}\right)^n \left(\frac{n(1-p)}{e}\right)^{n(1-p)}
\]

\[
= p^{-np} (1-p)^{-n(1-p)}
\]

\[
= 2^{-np \log_2 p} 2^{-n(1-p) \log_2 (1-p)}
\]

\[
= 2^{-n(1-H(p))} = 2^{-nC'.}
\]

That was probability for one codeword \(x_2\) being decoded from \(y\). For all codewords, the probability is at most

\[
2^{-nC} 2^{nR} = 2^{-n(C-R)} = 2^{-n\gamma}.
\]

On average
Conclusions

This says for a random code, can send at rate $R = C - \gamma$, and average error rate is exponentially small.

By picking the “best” code, it must be the case that we can achieve rate $R$ with exponentially small average error rate.

By “shrinking” the code slightly, we can guarantee the error rate is exponentially small in the worst case (regardless of which code word is sent). Simply get rid of the worst half of the original codewords.

Also, can change proof so that the “nearest” codeword (i.e., minimizing the number of changes) is the right codeword for a transmission.
Weaknesses of Shannon’s Theorem

Decoding is exponential time.

It says a random code is good... but it doesn’t give a simple construction of how to find a good one!

Shannon says random codes are good; the last 50+ years have simply been a matter of finding a good code you can write down and use.

So the question is: how do we find an explicit, good code, with reasonable encoding and decoding time bounds.
Linear codes

Any sum (modulo 2, for our purposes) of a codeword is a codeword. That is, codewords form a linear subspace. An \((n, k)\) code has \(2^k\) codewords, \(n\) bits.

Encoding can be described by

\[ v = u \cdot G, \]

where \(u\) is a \(k\)-bit message vector, and \(G\) is a \((k \text{ by } n)\) generator matrix. The rows of \(G\) span the space of codewords. A code is systematic if the original bits are part of the message; i.e., \(G\) contains the identity matrix.

Codewords can also be described by

\[ H \cdot v = 0, \]

where \(H\) is an \((n - k \text{ by } n)\) parity check matrix.
Example

\[ G = \begin{bmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}, \quad H = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1
\end{bmatrix} \]

\[ G = [P I], \quad H = [I P^T]. \]
Linear codes

Hamming distance: The Hamming distance between (0/1) vectors \( x \) and \( y \) is the number of coordinates where they differ.

The minimum distance of a code is the minimum Hamming distance between codewords. This equals the minimum number of 1’s in a non-zero codeword, and equals the minimum number of columns in the parity check matrix \( H \) that sum to 0.

If minimum distance is \( d \), a code can detect all error of \( d - 1 \) or fewer bits, and can correct all error patterns of \( \lfloor (d - 1)/2 \rfloor \) or fewer bits.

Errors and erasures?
Linear codes, decoding

Send $v = u \cdot G$, receive $r = u + e$ at the other end.

Compute syndrome $s = Hr$, and obtain

$$s = Hr = H(u + e) = Hu + He = He.$$ 

Use look-up table to compute most likely $e$ from $He$, and thereby compute $u$. 
Hamming codes

For all $m \geq 3$,

- Code length $n = 2^m - 1$.
- Symbols $k = 2^m - m - 1$.
- Redundant symbols $n - k = m$.
- Minimum distance 3.

Columns of parity matrix $H$ consists of all non-zero vectors of length $m$. 
Example

\[ G = \begin{bmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix},
\]

\[ H = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

\[ G = [PI],
\]

\[ H = [IP^T].\]