Converse of the Noisy Coding Theorem (NOT in Mackay)

"If \( \tilde{R} > C(\Omega) \): \( \exists \delta > 0 \) \( \forall N, N \geq N_0 \): \# code with \( \frac{K}{N} \geq \tilde{R} \) \& \( P_B \leq \delta \)

Tools: ① Data Processing Inequality (DPI) for \( A \to B \to C \) Markov chain:
\[
I(B: C) \geq I(A: C) \quad \& \quad H(A|B) \leq H(A|C)
\]
② If \( X^N \) arbitrary and \( Y^N \) channel output:
\[
I(X^N: Y^N) = \sum_{i=1}^{N} I(X_i; Y_i) \leq N \cdot C(\Omega)
\]
③ Fano's inequality for \( S \to T \to \hat{S} \) Markov chain, \( p = \Pr(C(S \neq \hat{S})) \)
\[
H(\hat{S}|p, 1-p) + p \cdot \log \#AS \geq H(C(S | \hat{S})) \geq H(S|T)
\]

Proof of the converse: Consider \((N, K)\)-code with \( \frac{K}{N} \geq \tilde{R} > C(\Omega) \).

Let \( S \in \{1, \ldots, 2^K\} \) uniform. Recall: \( S \rightarrow X^N \rightarrow Y^N \rightarrow \hat{S} \).

Then:
* \( H(S|Y^N) = H(S) - I(S; Y^N) \geq H(S) - I(X^N; Y^N) \geq K - N \cdot C(\Omega) \)

By DPI ①
\( S \rightarrow X^N \rightarrow Y^N \) Markov chain

* \( H(S|Y^N) \leq 1 + \Pr(C(S \neq \hat{S})) \cdot \log \#AS = 1 + P_B \cdot K \)

By Fano's inequality ②
\( S \rightarrow Y^N \rightarrow \hat{S} \) Markov chain

\[ K - N \cdot C(\Omega) \leq 1 + P_B \cdot K \]
\[ \Rightarrow P_B \geq \frac{1}{K} (K - N \cdot C(\Omega) - 1) = 1 - \frac{N \cdot C(\Omega)}{K} - \frac{1}{K} \geq 1 - \frac{C(\Omega)}{N} - \frac{1}{N} \]

Can never go below this for large enough \( N \)

Are we happy? What questions does Shannon's theorem leave unaddressed? Algorithmics, large \( N \), ... how to even compute \( C(\Omega) \)?
Shannon’s Theorem vs. Practice

Need large block size \( N \) for joint typicality vs. fixed packet size

Codebook \( X^N (U_1) \ldots X^N (2^K) \) exponentially large in \( N \) (if \( R > 0 \))

Random codes vs. predictable performance

A family of codes is "very good" if \( \frac{K}{N} \to 0 \) & \( p_c \to 0 \)

"good" if \( \frac{K}{N} \geq R \) & \( p_c \to 0 \) for some \( \tilde{R} > 0 \)

"bad" otherwise

...and "practical" if efficient encoder + decoder

In practice:

* most codes are linear (\( x^N \) linear function of \( S^K \) )

* "easy" to come up with "plausible" encoders — but optimal decoding is in general (NP) hard! — unlike for compression!

\[
\sigma_{\text{opt}} (y^N) = \arg\max_S P(S|y^N)
\]

Why? If \( P(s) \) arbitrary, want to choose \( \sigma \) to maximize \( P(s|y^N) \)

\[
= \sum \frac{P(s|o(y^N), t^N = y^N)}{y^N} \]

Choose \( s = o(y^N) \) that maximizes \( P(s|y^N) \)

For erasure channel:

\( S_1 \oplus S_2 \oplus S_3 \oplus t_1 = 0 \)
\( S_2 \oplus \ldots \oplus S_{K-2} \oplus t_2 = 0 \)

... & \( \frac{4}{3} \) bits per parity constraint, each bit in 3 parity constraints

* types of decoders: "algebraic" vs. "iterative"

Types of codes:

* block codes: e.g. Hamming, Reed-Solomon, LDPC codes, WiFi, DVB, ..

* convolutional: e.g. turbo codes

\( 36/46/\text{LTE} \), \( \text{Sat comm.} \)

\( \text{linear streaming codes} \)