Today

- More on Shannon's theory
  - Proof of converse.
  - Few words on generality.
  - Contrast with Hamming theory.
- Back to error-correcting codes: Goals.
- Tools:
  - Probability theory:
  - Algebra: Finite fields, Linear spaces.

Proof of Converse Coding Theorem

- Intuition: For message m, let $S_m \subseteq \{0,1\}^n$ be the set of received words that decode to m. ($S_m = D^{-1}(m)$).
- Average size of $D(m) = 2^{n-k}$.
- Volume of disc of radius pn around $E(m)$ is $2^{H(p)n}$.
- Intuition: If volume $\gg 2^{n-k}$ can't have this ball decoding to m — but we need to!
- Formalize?

Proof of Converse Coding Theorem (contd.)

Let $I_{m,\eta}$ be the indicator variable that is 1 iff $D((E(m)+\eta)) = m$.

Let $p' < p$ be such that $R > 1 - H(p')$.

Prob. [correct decoding]

$$\begin{aligned}
&= \sum \eta \in \{0,1\}^n \sum_{m \in \{0,1\}^k} \Pr[m \text{ sent}, \eta \text{ error a} \\
&\leq \sum_{\eta \in B(p'n,n)} \Pr[\eta \text{ error}] + \sum_{\eta \notin B(p'n,n)} \sum_{m} 2^{-k} \cdot \frac{}{2} \\
&\leq \exp(-n) + 2^{-k-H(p')n} \cdot \sum_{m,\eta} I_{m,\eta} \\
&= \exp(-n) + 2^{-k-H(p')n} \cdot 2^n \\
&\leq \exp(-n)
\end{aligned}$$

Generalizations of Shannon's theorem

- Channels more general
  - Input symbols Σ, Output symbols Γ, where both may be infinite (reals/complexes).
  - Channel given by its probability transition matrix $P = P_{\sigma,\gamma}$.
  - Channel need not be independent - could be Markovian (remembers finite amount of state in determining next error bit).
- In almost all cases: random coding + mld works.
- Always non-constructive.

Some of the main contributions

- Rigorous Definition of elusive concepts: Information, Randomness.
- Mathematical tools: Entropy, Mutual information, Relative entropy.
- Theorems: Coding theorem, converse.
- Emphasis on the "feasible" as opposed to "done".

Contrast between Hamming and Shannon

- Works intertwined in time.
- Hamming's work focusses on distance, and image of E.
- Shannon's work focusses on probabilities only (no mention of distance) and E, D but not properties of image of E.
- Hamming's results more constructive, definitions less so.
- Shannon's results not constructive, though definitions beg constructivitty.

- Most important difference: modelling of error — adversarial vs. probabilistic. Accounts for the huge difference in our ability to analyze one while having gaps in the other.
- Nevertheless good to build Hamming like codes, even when trying to solve the Shannon problem.

Our focus

- Codes, and associated encoding and decoding functions.
- Distance is not the only measure, but we will say what we can about it.
- Code parameters: n, k, d, q;
- typical goal: given three optimize fourth.
- Coarser goal: consider only $R = k/n$, $\delta = d/n$ and q and given two, optimize the third.
- In particular, can we get $R, \delta > 0$ for constant q?

- Will combine with analysis of encoding complexity and decoding complexity.

Tools

- Probability tools:
  - Linearity of expections, Union bound.
  - Expectation of product of independent r.v.s
  - Tail inqualities: Markov, Chebychev, Chernoff.
- Algebra
  - Finite fields.
  - Vector spaces over finite fields.
- Elementary combinatorics and algorithmics.

Finite fields and linear error-correcting codes

- Field: algebraic structure with addition, multiplication, both commutative and associative with inverses, and multiplication] distributive over addition.
- Finite field: Number of elements finite. Well known fact: field exists iff size is a prime power. See lecture notes on algebra for further details. Denote field of size q by $\mathbb{F}_q$.
- Vector spaces: V defined over a field $\mathbb{F}$. Addition of vectors, multiplication of vector with "scalar" (i.e., field element) is defined,

and finally an inner product (product of two vectors yielding a scalar is defined).

- If alphabet is a field, then ambient space Σ^n becomes a vector space $\mathbb{F}_q^n$.
- If a code forms a vector space within $\mathbb{F}_q^n$ then it is a linear code. Denoted $[n, k, d]_q$ code.

Why study this category?

- Linear codes are the most common.
- Seem to be as strong as general ones.
- Have succinct specification, efficient encoding and efficient error-detecting algorithms. Why? (Generator matrix and Parity check matrix.)
- Linear algebra provides other useful tools: Duals of codes provide interesting constructions.
- Dual of linear code is code generated by transpose of parity check matrix.

Example: Dual of Hamming codes

- Message $\mathbf{m} = \langle m_1, \ldots, m_\ell \rangle$.
- Encoding given by $\langle \langle \mathbf{m}, \mathbf{x} \rangle \rangle_{\mathbf{x} \in \mathbb{F}_2^\ell - \mathbf{0}}$.
- Fact: (will prove later): $\mathbf{m} \neq 0$ implies $\Pr_{\mathbf{x}}[\langle \langle \mathbf{m}, \mathbf{x} \rangle = 0] = \frac{1}{2}$
- Implies dual of $[2^\ell - 1, 2^\ell - \ell - 1, 3]_2$ Hamming code is a $[2^\ell - 1, \ell, 2^{\ell-1}]$ code.
- Often called the simplex code or the Hadamard code. (If we add a coordinate that is zero to all coordinates, and write 0s as -1s, then the matrix whose rows are all the codewords form a $+1/-1$ matrix whose product with its transpose is a multiple of

the identity matrix. Such matrices are called Hadamard matrices, and hence the code is called a Hadamard code.)

- Moral of the story: Duals of good codes end up being good. No proven reason.

Next few lectures

- Towards asymptotically good codes:
  - Some good codes that are not asymptotically good.
  - Some compositions that lead to good codes.