Today

Explicit constructions of asymptotically good codes

- Review Wozencraft ensemble (simplified).
- Codes from other codes:
  - Parity, Puncturing, Restriction, Direct Product.
  - Concatenation.
- Forney codes.
- Interlude: What is explicit?
- Justesen codes.

Wozencraft Ensemble: Special Case + Simplified

- Codes $C_\alpha : \{0,1\}^k \to \{0,1\}^{2k}$.
- Let $\mathbb{F}$ be field of size 2^k. Then $C_\alpha : \mathbb{F} \to \mathbb{F}^2$. One such code for every $\alpha \in \mathbb{F}$: $C_\alpha(x) = (x, \alpha x)$.
- Codes C_α, C_β don't share non-zero codewords ($(x,y) \in C_{x^{-1}y}$).
- $\exists C_\alpha$ with distance $\approx H^{-1}(1/2) \cdot (2k)$.
- Most C_α's have half that distance!
- Ensemble constructible in time $2^{O(k)}$.

Codes from other codes

Many interesting codes obtained from other codes by simple operations. Also useful in bounds.

Parity check Add one bit of parity of code. $(n,k,2t-1)_2 \to (n+1,k,2t)_2$ code.

Puncturing Delete one coordinate of code. $(n,k,d)_q \to (n-1,k,d-1)_q$ code.

Restricting Take subcode corresponding to first coordinate being "most common element" and then delete first coordinate. $(n,k,d)_q \to (n-1,k-1,d)_q$ code.

Codes from other codes (contd.)

Direct Product Messages are matrices. Encode rows with C_1 and then columns with C_2. $(n_1,k_1,d_1)_q \otimes (n_2,k_2,d_2)_q \to (n_1 n_2, k_1 k_2, d_1 d_2)_q$.

Sub-Σ Subcodes Take $\Sigma' \subseteq \Sigma$ and $C \subseteq \Sigma^n$ and let $C' = C \cap (\Sigma')^n$. $(n,k,d)_q \to (n,?,d)_{q'}$ code.

- Not generically useful.
- Gives very nice specific codes. (BCH from RS.)

Most operations weaken codes asymptotically. Only one exception.

Concatenation of codes

- Take code C_1 over large alphabet. Encode message with C_1.
- Then "represent" elements of large alphabets as strings over small alphabet.
- Might as well use small alphabet code C_2 to "represent" elements.
- Specifically:
  - Let $Q = q^k$. Let $C_1 = (N, K, D)_Q$ code. Let $C_2 = (n, k, d)_q$ code.
  - Message comes from $Q^K \cong q^{kK}$.
  - Encoding gives el'ts of Q^N (first stage) and q^{nN} (second stage).
  - Distinct message differ in at least D COORDINATES, and hence in at least dD coordinates.
  - $(N, K, D) \circ (n, k, d) \rightarrow (nN, kK, dD)$ codes.
- Same as direct product? NO! Direct product needs first code to be over q. Concatenation only needs this over Q. Latter easier empirically.
- Idea due to [Forney].

Asymptotically good codes

- Know how to get $(N, K, D)_Q$ codes. (Take Reed-Solomon codes.)
- How to get $(n, k, d)_q$ codes? Forney's idea: Brute force search!
- Why is this ok?
- Example parameters.
  - $K, D = N/2, Q = N = 2^{\log N}, q = 2, k = \log N$.
  - $n = 2k = O(\log N), d = H^{-1}(1/2)n, q = 2$.
- Brute force search (say for random linear code) takes time $2^{O(n^2)} = N^{O(\log N)}$. Asympt. good code in quasi-polynomial time (Rate $1/4$, Rel. Distance $1/2H^{-1/2}$.)
- Search Wozencraft ensemble: $2^{O(n)} = N^{O(1)}$ time. Gives poly-time construction of asymptotically good codes.
- Two-level concatenation:
  - K, D as before.
  - $k = \log N, n = 2k, q = n$, $Q = \log N^{\log N}$.
  - $q_0 = 2, k_0 = \log n = \log\log N, n_0 = 2k_0$, etc.

  Use RS codes at outer and middle level. Brute force search at inner level. Now quasi-polynomial in n and so polynomial in N.

- Bibliographic asides: Forney doesn't mention the codes themselves - only concatenation! The tradeoff (distance to rate) was studied later by Zyablov. So what did Forney do? Gave polytime E and D getting arbitrarily close to Shannon capacity for BSC_p (as also other, more important channels). Will do this part later. Solves biggest problem in Shannon theory; and the Hamming consequences become a footnote.

Explicit constructions

- Are the Forney constructions explicit?
- Standard refrain from the pas: Constructive - yes! Explicit - No! After all we don't know the Forney codes. We have to search for them.
- Debate entirely too subjective.
- Complexity theory can make this objective. To "know" is to be able to compute efficiently.
- Forney codes are explicit .... if explicit is defined as polynomial time constructible.

- Are there other definitions of explicit, that appeal to our intuition?

Shades of Explicitness

Increasingly explicit notions. For simplicity assume linear code, and we wish to construct generator matrix.

- Constructible by finite time procedure!
- Constructible by polynomial time procedure.
- Constructible by logspace procedure: E.g. Forney one-level with brute force is not logspace constructible, but two-level and Wozencraft are logspace constructible.
- Locally polynomial time: Given i, j indices into generator matrix: Can compute G_{ij} in polytime in $|i|, |j|$. Don't have such explicitness yet.

- Locally logspace ...

Justesen's construction of explicit codes

- General question: How can you eliminate the "search" in the Forney-type construction.
- Justesen's insights:
  - Need to find one out of $\{0,1\}^n$ codes (when we know most are good enough) ....
  - ... to use it 2^n times!
  - But who says we must always use the same code?
- Justesen concatenation:
  - Outer code: $(N, K, D)_Q$

  - Inner sequence of codes: $\langle C_i \rangle_{i=1}^N$, with $C_i = (n, k, ?)_q$ codes, with $Q = q^k$ and all but ϵN of the C_i's having distance d.
  - Concatenation: Encode ith symbol of outer encoding by C_i.
  - Yields: $(Nn, Kk, (D - \epsilon N)d)_q$ code!
- Gets about as explicit as we can handle!

Notes on linearity

- Both direct product, and concatenation, can be applied to get linear codes.
- Former case: Just linear algebra.
- Latter case: Make sure elements of $\mathbb{F}_Q$ represented as vectors over $\mathbb{F}_q$ satisfying additivity constraints.
- Details omitted.