CAS CS 599 B: Mathematical Methods for TCS

Lecturer: Mark Bun Spring 2022

Lecture Notes 21:

Justesen Codes, Expander Codes, Miscellaneous Coding Topics

Reading.

- Guruswami-Rudra-Sudan §10, 11

Here's a reminder of some definitions:

- An $(n, k, d)_q$ code is a subset $C \subseteq \Sigma^n$ where $|\Sigma| = q$. Here, n is the block length, $k = \log_q |C|$ is the dimension, and $d = \min_{v,w \in C} \Delta(v, w)$ is the distance of the code.
- The rate of a code is $R = k/n$. The relative distance is $\delta = d/n$.
- An $[n, k, d]_q$ linear code is one for which C is a k-dimensional subspace of $\Sigma^n = \mathbb{F}_q^n$.

1 Justesen Codes

Last time, we saw how to get binary, asymptotically good codes by concatenating Reed-Solomon codes with good linear codes of logarithmic dimension. A concatenated code combines an outer code $C_{out} \subseteq \Sigma^N$ with an inner code $C_{in} \subseteq \sigma^n$ where $|C_{in}| \geq |\Sigma|$. The result is a concatenated code $C = C_{out} \circ C_{in}$, for which encoding is performed via

$$\mathrm{Enc}(x) = (\mathrm{Enc}_{in}((\mathrm{Enc}_{out}(x))_1), \ldots, \mathrm{Enc}_{in}((\mathrm{Enc}_{out}(x))_N))$$

If C_{out} is an $(N, K, D)_{q^k}$-code and C_{in} is an $(n, k, d)_q$-code, then the concatenated code C is an (Nn, Kk, Dd)-code. So the rate and relative distance of the concatenated code are the products of the rates and relative distances of the constituent codes.

The construction we saw last time was weakly explicit in the sense that given block length n, one can construct the generator matrix of the code in time $\mathrm{poly}(n)$. We can also ask for strongly explicit (linear) codes which are $[n, k]$ codes for which, given any pair $(i, j) \in [k] \times [n]$, the bit $G_{i,j}$ of the generator matrix for C can be computed in time $\mathrm{poly}(\log n)$.

Justesen codes are strongly explicit asymptotically good codes. The idea is that nobody said the inner codes C_{in} in a concatenated code have to all be the same. We can take them to be different, with the property that most of them (even if we don't know which ones) meet the Gilbert-Varshamov bound.

Theorem 1. *For each $\alpha \in \mathbb{F}_{2^k}, \alpha \neq 0$, define $C_{in}^{\alpha} \subseteq \mathbb{F}_2^{2k}$ by $\mathrm{Enc}_{in}^{\alpha}(x) = (x, \alpha x)$ for $x \in \mathbb{F}_2^k$. Then $(C_{in}^{\alpha})_{\alpha \in \mathbb{F}_{2^k}, \alpha \neq 0}$ is a family of strongly explicit codes with rate $1/2$ such that a $1 - \varepsilon$ fraction of these codes have relative distance at least $H_2^{-1}(1/2 - \varepsilon)$.*

Hopefully not-too-confusingly, αx denotes the vector in $\mathbb{F}_2^k$ that corresponds to the field multiplication of α and x regarded as elements of $\mathbb{F}_{2^k}$. This collection of codes is called the Wozencraft ensemble. You can see the proof of this statement in Section 10.3.1 of GRS, but I encourage you to prove it yourself as an exercise.

The Justensen code is obtained by concatenating the Reed-Solomon code with the Wozencraft ensemble:

- Take the outer code C_{out} to be a Reed-Solomon code over $\mathbb{F}_{2^k}$ with rate R. The relative distance is $\delta_{out} = 1 - R$ and the block length is $N = 2^k - 1$.
- The inner codes are $(C_{in}^{\alpha})_{\alpha \in \mathbb{F}_{2^k}, \alpha \neq 0}$.

The concatenated code has rate $R/2$. Its distance is given by

Theorem 2. *The Justensen code C above has relative distance at least $(1 - R - \varepsilon) \cdot H_2^{-1}(1/2 - \varepsilon)$ for $\varepsilon > 0$.*

Proof. Let $x \neq x' \in \mathbb{F}_{2^k}^K$ be distinct messages. Let $S = \{i \in [N] \mid C_{out}(x)_i \neq C_{out}(x')_i\}$. By the distance of the outer code, $|S| \geq (1 - R)N$. Now call the i'th inner code "good" if it has distance at least $d = H_2^{-1}(1/2 - \varepsilon) \cdot 2k$. The number of good codes is at least $(1 - \varepsilon)N$. Hence, there are at least $(1 - R - \varepsilon)N$ codes in S that are also good. This implies that the distance between the encodings of x and x' is at least

$$(1 - R - \varepsilon)N \cdot d \geq (1 - R - \varepsilon) \cdot H_2^{-1}(1/2 - \varepsilon) \cdot (N \cdot 2k).$$

□

Decoding. The natural decoder for a Justensen code is to first decode each inner code using maximum likelihood decoding, and then decode the outer code. Each inner decoding step takes time exponential in the inner block length, which is polynomial in the concatenated block length. Decoding the outer code can be done in polynomial time, e.g., using the Berlekamp-Welch algorithm for decoding Reed-Solomon codes. The natural decoder can correct up to a $\delta/4$ fraction of errors for a relative distance δ code.

This can be improved to $\delta/2$ using "Generalized Minimum Distance" decoding described in GRS §13.3. The idea is to obtain more information about the outer codeword based on the outcome of the inner decoder. If the inner decoder decodes a symbol at large Hamming distance from the received inner codeword, this should be treated as an erasure, which is easier for the outer decoder to handle. Outer errors, then, result from the inner decoded symbol being different from an inner codeword, which can't happen too often.

2 Expander Codes

Justesen codes were the first explicit asymptotically good codes over a binary alphabet. Moreover, encoding and decoding can both be performed in polynomial time. We now know of many other families of codes with these properties. Another important such class is codes based on explicit constructions of expanders.

The connection is based on a correspondence between linear codes and bipartite graphs. Let C be an $[n, n - m]_2$ linear with $m \times n$ parity check matrix H. Then H is the adjacency matrix of the following bipartite graph:

- The left set of vertices is $L = [n]$, corresponding to the indices of bits in a codeword.
- The right set of vertices $R = [m]$ corresponds to the constraints (rows) of H.

- There is an edge from $i \in L$ to $j \in R$ iff codeword bit i participates in parity check constraint j.

The resulting graph G is called the factor graph of C and satisfies

$$C = \{c \in \mathbb{F}_2^n \mid Hc = \mathbf{0}\} = \{c \in \mathbb{F}_2^n \mid \forall j \in [m] : \bigoplus_{i \sim j} c_i = 0\}.$$

Linear codes with sparse factor graphs are called *low-density parity check (LDPC) codes*. Such codes with good rate and distance can generically be constructed from expanders. The expanders we need here are slightly different form the ones we've studied so far.

Definition 3. Let $G = (L = [n], R = [m], E)$ be a D-left regular bipartite graph. Then G is an (α, β)-vertex expander if for every set $S \subseteq L$ with $|S| \leq \alpha n$, we have $|N(S)| \geq \beta D|S|$.

That is, every small (captured by α) set of vertices on the left has a large (captured by β) number of neighbors on the right.

Building on the zig-zag graph product construction, Capalbo, Reingold, Vadhan, and Wigderson gave an explicit construction of such expanders with β arbitrarily close to 1, constant D, and m a constant multiple of n.

A key property of such expanders is that every small set S of left vertices has many *unique* neighbors. A right vertex j is a unique neighbor of S if it has only one neighbor in S.

Claim 4. *For $S \subseteq [n]$, define $U(S) = \{j \in R \mid |S \cap N(j)| = 1\}$. If G is an (α, β)-vertex expander, then for every $|S| \leq \alpha n$, we have $|U(S)| \geq (2\beta - 1)D|S|$.*

Proof. Let $u = |U(S)|$. Then the total number of edges out of S is

$$D|S| \geq u + 2(|N(S)| - u).$$

Rearranging gives

$$u \geq 2|N(S)| - D|S| \geq (2\beta - 1)D|S|$$

by expansion. □

Theorem 5. *If $G = (L = [n], R = [m], E)$ is a D-left regular (α, β)-vertex expander for $\beta > 1/2$, then the corresponding code C is an $[n, n - m, \alpha n + 1]_2$ linear code.*

Proof. The block length and linearity claims are immediate from the definition.

As for the distance, assume for the sake of contradiction that C has distance at most αn. Then there exists a codeword c of Hamming weight at most αn. Let S be the set of nonzero coordinates of c. By Claim 4, we have $|U(S)| \geq (2\beta - 1)D|S| > 0$, so there exists a vertex $j \in R$ with exactly one neighbor $i^* \in S$. Thus, $\oplus_{i \sim j} c_i = c_i = 1$, which contradicts the parity check constraint corresponding to j.

Finally, the dimension bound follows from the fact that the code has distance at least 1. □

A more careful counting argument can be used to show that the distance of the code in Theorem 5 can be taken to be $2\alpha\beta n$, though still only for $\beta > 1/2$.

The magic of LDPC codes is that they have ridiculously simple (linear time and efficiently parallelizable) decoding algorithms. Here's the whole algorithm: If a codeword bit $\hat{c}_i$ looks wrong in the sense that the majority of the constraints it participates in are violated, then flip it. Keep doing this until there are no more candidates to flip.

Theorem 6. *Let $\beta > 3/4$, $c \in C$ be a codeword, and $\hat{c} \in \mathbb{F}_2^n$ such that $\Delta(\hat{c}, c) \leq \alpha n/4$. Then the above decoding algorithm recovers c.*

This works because of the following sequence of claims:

Claim 7. *If $\hat{c} \in \mathbb{F}_2^n$ is not a codeword, there is a candidate bit to flip.*

Proof. Let S be the set of coordinates on which c and $\hat{c}$ disagree. Then by Claim 4, we have $|U(S)| \geq (2\beta - 1)D|S| > D|S|/2$. Therefore, some $i \in S$ has more than $D/2$ neighbors in $U(S)$, so i is a candidate to flip. □

Claim 8. *The number of satisfied parity checks always increases.*

Since the total number of parity checks is m, the above two claims show that the algorithm terminates at some codeword after this many rounds. The final claim ensures that this is the correct codeword.

Claim 9. *The algorithm can never reach a candidate codeword with $\alpha n/2$ errors.*

Proof. The starting codeword $\hat{c}$ had at most $\alpha n/4$ errors, so the starting number of unsatisfied parity checks is at most $D\alpha n/4$. If the algorithm reaches a candidate codeword with $\alpha n/2$ errors, then Claim 4 implies that the number of incorrect parity checks is greater than $D\alpha n/4$, which contradicts Claim 8. □