CS 860 Topics in Coding Theory Oct. 7, 2025

Lecture 10

Lecturer: Elena Grigorescu *Scribe: Jacob Skitsko*

Last time:

- *Explicit* (i.e. efficiently computable) binary codes achieving the Zyablov bound:

$$R \geq \sup_{0 \leq r \leq 1-H(\delta)-\varepsilon} r \cdot \left(1 - \frac{\delta}{H^{-1}(1-r-\varepsilon)}\right).$$

  - Accomplished with code concatenation, with a Reed-Solomon code as C_{out} and a small binary code at the Gilbert-Varshamov bound as C_{in}.

Today:

- Decoding concatenated codes.
- *Explicit* (i.e. with an explicit description) binary codes with good parameters. (Wozencraft ensembles, and Justesen codes).

1 Decoding Concatenated Codes

First let us review code concatenation, before moving on to some different methods of decoding concatenated codes. Recall the following pictorial definition:

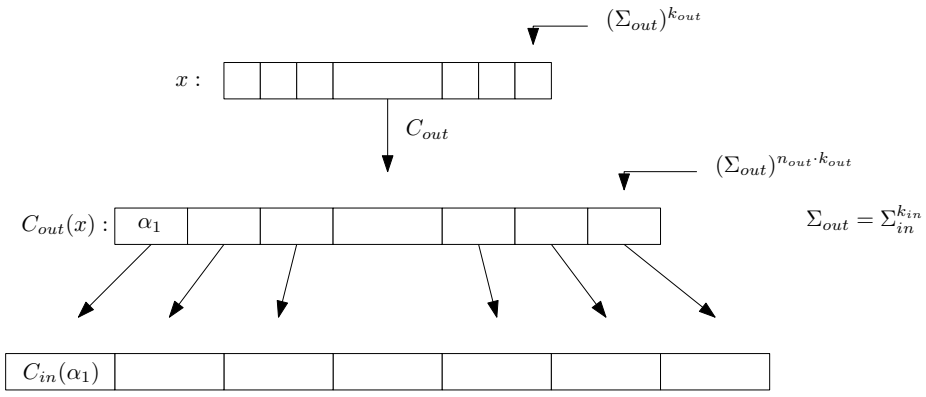

Figure 1: Code Concatenation Image

where we have an outer and inner code

$$C_{\text{out}} : \Sigma_{\text{out}}^{k_{\text{out}}} \to \Sigma_{\text{out}}^{n_{\text{out}}} \quad \text{and} \quad C_{\text{in}} : \Sigma_{\text{in}}^{k_{\text{in}}} \to \Sigma_{\text{in}}^{n_{\text{in}}}.$$

Note the rate of the concatenated code $C_{\text{out}} \circ C_{\text{in}}$ is $r_{\text{out}} \cdot r_{\text{in}}$ and the distance is $\delta_{\text{out}} \cdot \delta_{\text{in}}$.

Now, say we have received some $(y_1, \ldots, y_n) \in \Sigma_{\text{in}}^{n_{\text{in}} \cdot n_{\text{out}}}$. How should we decode this?

We will assume we have efficient decoders for the inner and outer codes $C_{\text{in}}, C_{\text{out}}$ with respect to $N := n_{\text{in}} n_{\text{out}}$. Note in the construction from the last class we could use a brute force decoder for the small inner blocks (since they are only of size $O(\log n)$), and then use the Berlekamp-Welch decoder for the outer Reed-Solomon code, so this assumption is reasonable.

1.1 A Naïve Decoder

One natural idea is to simply decode the blocks of the inner code individually into some $(\alpha_1, \ldots, \alpha_n)$ and then decode $(\alpha_1, \ldots, \alpha_n)$ using the outer decoder.

How much error can this natural decoder withstand? Our goal, as always, is to decode up to half of the minimum distance of the concatenated code, i.e. up to $d_{\text{in}} d_{\text{out}}/2$ errors.

Observe that the decoder for C_{in} on block i is correct if there are fewer than $d_{\text{in}}/2$ errors, and otherwise the decoder may fail. This means if there are $d_{\text{in}} d_{\text{out}}/2$ errors, then the number of blocks the decoder fails on is at most

$$\frac{d_{\text{in}} d_{\text{out}}}{2} \cdot \frac{2}{d_{\text{in}}} = d_{\text{out}} \,.$$

But that's every block! This naïve decoder fails to decode at $d_{\text{in}} d_{\text{out}}/2$ errors. However, with minor adjustments, the above argument does show that the naïve decoder succeeds at $d_{\text{in}} d_{\text{out}}/4$ errors.

One may still hope to decode at half the minimum distance. Can we beat the naïve decoder, if only for the specific code concatenation we talked about last time?

1.2 A Better Decoder

We will discuss an idea called *Generalized Minimum Distance Decoding*, from Forney 1966.

Idea: the inner code is small enough to brute force compute the nearest codeword to any given y (this problem is NP-Hard in general, but the inner code has size $O(\log N)$). Can we use this "soft information" to give us an advantage?

Issues:

- It could be that the nearest codeword to $C_{\text{in}}(\alpha)$, for some α, is at distance $> d_{\text{in}}/2$ and no other $C_{\text{in}}(\beta)$ is within distance $d_{\text{in}}/2$. This suggests to us that there were many errors, and so we should ignore this block. So, we can treat this as an *erasure*!
- It could be that some block has so much error added to it so that the nearest codeword to y is some other $C_{\text{in}}(\beta)$, rather than the transmitted $C_{\text{in}}(\alpha)$. So, we will need to somehow account for this possibility.

Can we make use of the above information in our outer decoder (for a Reed-Solomon code)? It turns out yes!

Claim 1 *A decoder for $RS(n, k)$ can efficiently and uniquely decode from e errors and s erasures if*

$$2e + s < n - k + 1 \,.$$

Remark In the case $s = 0$, the above claim exactly states that a decoder for $RS_q(n, k)$ can efficiently and uniquely decode from up to half the minimum distance.

Proof Idea First disregard the erasures. This yields a new Reed-Solomon code, $RS_q(n - s, k)$ of distance $n - s - k + 1$. Note $(n - s - k + 1)/2 > e$, so we can do unique decoding. Then we can do Gaussian elimination afterwards to recover erasures. ■

Now we can try decoding $C_{\text{out}} \circ C_{\text{in}}$ with the additional "soft information" of the distance of block i to C_{in}. Let

$$u_i := \Delta(y_i, C_{\text{in}}(\alpha_i)) \,,$$

where α_i is the nearest codeword. Recall again that $n_{\text{in}} = O(\log N)$, so we can brute force calculate each u_i. Then consider the "soft information"

$$w_i := \min(d_{\text{in}}/2, u_i) \,.$$

We now try to solve the above issues with the following idea. For the outer code:

- if $u_i > d/2$ then we should treat block i as an erasure,
- if $u_i = 0$ then we should leave block i alone,
- otherwise, if $u_i \leq d/2$ we should treat block i as an erasure with probability $w_i/(d/2)$.

Call the resulting blocks a_i, and then run the error and erasure decoding on the blocks $(a_i)_i$. We want to show $\mathbb{E}[2e + s] < n - k + 1$, to show this process works in expectation. Afterwards, we can derandomize this using threshold rounding.

Lemma 2 *Let Z^{errs}, Z^{eras} be random variables counting the number of errors and erasures respectively, after we pass $(a_i)_i$ to the outer decoder. Let $e_i := \Delta(y_i, C_{in}(c_i))$ where c_i is the actual encoded codeword. If*

$$\sum_i e_i < \frac{d_{in} d_{out}}{2}$$

then

$$\mathbb{E}[2Z^{errs} + Z^{eras} < d_{out} \,.$$

Note the expectation is over the random choices in our process to make $(a_i)_i$.

Proof We'll show the statement coordinate-wise. Let $Z_i^{\text{errs}}, Z_i^{\text{eras}}$ be 0/1 indicators for the error or erasure state of block i. Note

$$Z^{\text{errs}} = \sum_i Z_i^{\text{errs}} \,, \quad Z^{\text{eras}} = \sum_i Z_i^{\text{eras}} \,.$$

Claim 3

$$\mathbb{E}[2Z_i^{errs} + Z_i^{eras}] \leq \frac{2e_i}{d_{in}} \,.$$

Proof

case 1: Suppose $c_i = a_i$, so $w_i = e_i$. Then $\mathbb{Z}_i^{\text{errs}} = 0$ and $\mathbb{E}[Z_i^{\text{eras}}] = 2w_i/d_{\text{in}} = 2e_i/d_{\text{in}}$. So,

$$\mathbb{E}[2Z_i^{\text{errs}} + Z_i^{\text{eras}}] = 2e_i/d_{\text{in}}] \,.$$

case 2: Suppose $c_i \neq a_i$, so $\mathbb{E}[Z_i^{\text{eras}}] = 2w_i/d_{\text{in}}$ and $\mathbb{E}[Z_i^{\text{errs}}] = 1 - 2w_i/d_{\text{in}}$. So,

$$\mathbb{E}[2Z_i^{\text{errs}} + Z_i^{\text{eras}}] = 2 - 2w_i/d_{\text{in}}] \,.$$

Let us compare $2-2w_i/d_{\text{in}}$ and $2e_i/d_{\text{in}}$, or in other words let us compare d and e_i+w_i. Recall $e_i = \Delta(y_i, C_{\text{in}}(c_i))$ and $w_i = \min(d_{\text{in}}/2, u_i)$. Note that $\Delta(y_i, C_{\text{in}}(\alpha_i)) \geq d_{\text{in}}/2$, and that $C_{\text{in}}(\alpha_i)$ and $C_{\text{in}}(c_i)$ have at least distance d_{in}. Then we can use the triangle inequality, and rearrange, to show $d_{\text{in}} \leq e_i + w_i$. This yields the desired claim in this

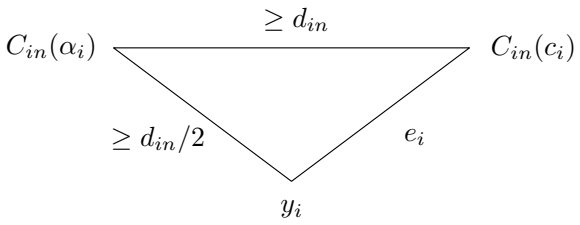

Figure 2: Triangle inequality for errors.

case:

$$\mathbb{E}[2Z_i^{\text{errs}} + Z_i^{\text{eras}}] \leq \frac{2e_i}{d_{\text{in}}}$$

Since the inequality holds coordinate wise for each i, we have that it holds overall also. ■

The above randomized algorithm works in expectation. We can repeat it many times, and obtain an algorithm that works with high probability. However, we can do better. We can derandomize our random algorithm for choosing the $(a_i)_i$ values with the following idea. It is straightforward to check the following random process will give the same number of expected errors and erasures.

- Pick some threshold θ $(0, 1)$ uniformly randomly.
- Set a_i as an erasure if $\theta \leq 2w_i/d_{\text{in}}$.

Observe that this algorithm only behaves differently at the threshold points $0, 2w_1/d_{\text{in}}, 2w_2/d_{\text{in}}, \ldots, 2w_n/d_{\text{in}}$ where $w_i \in \{0, 1, \ldots, d_{\text{in}}/2\}$. So it suffices to check each of these possible θ values explicitly (and deterministically!).

■

2 Explicit Good Binary Codes

We will start by giving the following ensemble of codes.

2.1 Wozencraft Ensemble

For $\alpha \in \mathbb{F}_{2^k}$ define

$$C_{\text{in}}^{\alpha} : x \to (x, \alpha.x)\,,$$

where $x \in \mathbb{F}_2^k$ is treated as an element of $\mathbb{F}_{2^k}$. Note $(x, a.x)$ has $2k$ bits. It is clear to see such a C_{in}^{α} has rate $1/2$.

Theorem 4 *Let $\varepsilon > 0$, and fix k. Consider the ensemble of binary codes*

$$C_{in}^{\alpha_1}, \ldots, C_{in}^{\alpha_N} \subseteq \mathbb{F}_2^{2k}\,,$$

where $N \geq 2^k - 1$ and $\alpha_i \in \mathbb{F}_{2^k} \setminus \{0\}$ for each $i \in [N]$. Then for at least $(1-\varepsilon)N$ values α_i, $C_{in}^{\alpha_i}$ has distance $H_2^{-1}(1/2 - \varepsilon)$ (i.e. the GV bound at rate 1/2).

The proof of this is omitted, but it is perhaps not too surprising in retrospect with the knowledge that random codes are likely at the GV bound.

2.2 Justesen Code

Let $k > 0$ be the dimension of the inner code in the Wozencraft ensemble. Recall Reed-Solomon codes took in evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{2^k}^*$ and was defined as $\langle f(\alpha_1), \ldots, f(\alpha_n)\rangle$. Instead, we can define the following Justensen code

$$\Big(f(\alpha_1), \alpha_1 f(\alpha_1), \ldots, f(\alpha_n), \alpha_n f(\alpha_n\Big)\,,$$

where we remark that each interior block $f(\alpha_1), \alpha_1 f(\alpha_1)$ looks like a code from the Wozencraft ensemble $C_{\text{in}}^{\alpha_i}(f(\alpha_i))$. So, we are really just concatenating with a different code!

More formally:

$$C_{\text{out}} = RS_{2^k}(F_{2^k}^*, 2^k - 1, R_{\text{out}}(2^k - 1)$$

and

$$C = \{\langle C_{\text{in}}^{\alpha}(f(\alpha))\rangle_{\alpha \in \mathbb{F}_{2^k}^*}\}$$

for $f \in \mathbb{F}_{2^k}[x]$, $\deg f < R_{\text{out}}(2^k - 1)$.

We remark the Justesen code has rate $R_{\text{out}}/2$, since each inner block has rate $1/2$.

Proposition 5 *If the outer code has distance δ_{out}, and $1 - R_{out} \geq 2\varepsilon$ then the Justesen code has distance at least $(1 - R_{out} - \varepsilon) \cdot H_2^{-1}(1/2 - \varepsilon)$.*

This follows from having an ε fraction of non-zero blocks of weight $H_2^{-1}(1/2 - \varepsilon)$. Observe that $\delta > \varepsilon H_2^{-1}(1/2 - \varepsilon) = \Theta(1)$. So, this code is asymptotically good.