Introduction to Coding Theory **Técnico–ULisboa, 2026/2027**

Part 7: Concatenated codes

Lecturer: João Ribeiro

Recommended reading: Essential Coding Theory, Chapters 10 and 14.

1 Introduction

A family of binary codes is *asymptotically good* if it has positive asymptotic rate and positive relative distance. Do asymptotically good codes exist? Can we construct them efficiently, and get efficient encoding and decoding algorithms?

The GV bound shows that asymptotically good binary codes exist. In fact, we even know that asymptotically good *linear* codes exist, because random linear codes achieve the GV bound. However, given our current state of knowledge, constructing a code that lies on the GV bound is extremely inefficient, because the underlying proof is non-constructive in nature. Brute-forcing over all generator matrices of binary linear codes of block length n would take time $2^{\Theta(n^2)}$. Ideally, we would like to:

1. Design a family of asymptotically good binary linear codes whose generator matrices can be constructed in time polynomial in the block length n.
2. Devise an efficient unique decoding algorithm for this code.

Note that Reed-Solomon codes satisfy both items above. However, they require alphabet size $q \geq n$. Can we somehow transfer the nice properties of Reed-Solomon codes to $\mathbb{F}_2$?

2 Code concatenation

We will tackle the problems outlined above using *code concatenation*. Fix $q = 2^r$ and let $\mathcal{C}_{\text{out}}$ be a q-ary code of block length n_{out}, size $q^{k_{\text{out}}}$, and distance d_{out}. We may interpret the codewords of $\mathcal{C}_{\text{out}}$ as vectors in $\mathbb{F}_q^{n_{\text{out}}}$. Furthermore, since $q = 2^r$, we can also interpret the elements of $\mathbb{F}_q$ as vectors in $\mathbb{F}_2^r$. Writing codewords of $\mathcal{C}_{\text{out}}$ as vectors in $(\mathbb{F}_2^r)^{n_{\text{out}}}$, we get a binary code of block length $n = r \cdot n_{\text{out}}$, size $q^{k_{\text{out}}} = 2^{rk_{\text{out}}}$, and, pessimistically, distance d_{out}, since if two codewords, interpreted as vectors in $\mathbb{F}_q^{n_{\text{out}}}$, differ in one coordinate, then their binary representations may differ in only 1 bit of the corresponding binary representations. Applying this to Reed-Solomon codes with $q = n_{\text{out}}$, we get a binary code with block length $n = n_{\text{out}} \log n_{\text{out}}$, dimension $k_{\text{out}} \log n_{\text{out}}$, and distance $d_{\text{out}} = n_{\text{out}} - k_{\text{out}} + 1$. Therefore, although we can get positive rate if $k_{\text{out}} = \Theta(n_{\text{out}})$, the relative distance guarantee is smaller than $1/\log n_{\text{out}}$, and so this is not an asymptotically good code. Can we do better?

To get better distance guarantees, instead of simply replacing each symbol in $\mathbb{F}_q$ by its r-bit binary representation, we encode these binary representations using a binary *inner code* $\mathcal{C}_{\text{in}}$ of block length n_{in}, size 2^r, and distance d_{in}. Then, letting Enc_{in} denote the encoding procedure corresponding to $\mathcal{C}_{\text{in}}$, we get the concatenated code

$$\mathcal{C}_{\text{in}} \circ \mathcal{C}_{\text{out}} = \{(\text{Enc}_{\text{in}}(c_1), \text{Enc}_{\text{in}}(c_2), \ldots, \text{Enc}_{\text{in}}(c_{n_{\text{out}}})) : (c_1, \ldots, c_{n_{\text{out}}}) \in \mathcal{C}_{\text{out}}\}.$$

The following theorem captures the main properties of this code.

Theorem 1 *$\mathcal{C}_{\text{in}} \circ \mathcal{C}_{\text{out}}$ is a binary code of block length $n = n_{\text{in}} \cdot n_{\text{out}}$, size $2^{r \cdot k_{\text{out}}}$, and distance $d \geq d_{\text{in}} \cdot d_{\text{out}}$. In particular, if $\mathcal{C}_{\text{out}}$ and $\mathcal{C}_{\text{in}}$ have rate and relative distance R_{out}, δ_{out} and R_{in}, δ_{in}, respectively, then $\mathcal{C}_{\text{in}} \circ \mathcal{C}_{\text{out}}$ has rate $R = R_{\text{in}} \cdot R_{\text{out}}$ and relative distance $\delta \geq \delta_{\text{in}} \cdot \delta_{\text{out}}$.*

Furthermore:

1. *If $\mathcal{C}_{\text{out}}$ and $\mathcal{C}_{\text{in}}$ are linear codes, then so is $\mathcal{C}_{\text{in}} \circ \mathcal{C}_{\text{out}}$, and;*
2. *If we can construct generator matrices for $\mathcal{C}_{\text{out}}$ and $\mathcal{C}_{\text{in}}$ in time polynomial in n, then we can construct a generator matrix for $\mathcal{C}_{\text{in}} \circ \mathcal{C}_{\text{out}}$ in time polynomial in n.*

Therefore, if we can construct the generator matrix of an asymptotically good binary linear inner code $\mathcal{C}_{\text{in}}$ running in time polynomial in $n = n_{\text{in}} \cdot n_{\text{out}}$, then we are done. Crucially, note that we have made the problem easier, since the block length of the inner code is $n_{\text{in}} = O(r) = O(\log n_{\text{out}})$. Indeed, recalling that there is a binary linear code on the GV bound, we can construct the generator matrix of an asymptotically good inner code by naively brute-forcing over all $n_{\text{in}} \times r$ generator matrices in time $2^{O(n_{\text{in}} \cdot r)} = 2^{O(r^2)} = 2^{O(\log^2 n_{\text{out}})} = n^{O(\log n)}$. This is not quite polynomial in the final block length $n = n_{\text{in}} \cdot n_{\text{out}}$. But, in fact, we can focus on brute-forcing only over all *Toeplitz* matrices, which only takes time $2^{O(r)} = 2^{O(\log n_{\text{out}})} = n^{O(1)}$, as desired. You will show this in the problem set.

The discussion in the paragraph above combined with instantiating the outer code $\mathcal{C}_{\text{out}}$ as a Reed-Solomon code with dimension $k_{\text{out}} = \Omega(n_{\text{out}})$ already answers Item 1 affirmatively. But how far can we push this? More precisely, given a target relative distance δ, how large can we make the rate R via concatenated coding with a Reed-Solomon outer code and an inner code on the GV bound?

A Reed-Solomon outer code with rate R_{out} has asymptotic relative distance $\delta_{\text{out}} = 1 - R_{\text{out}}$. Since the inner code can achieve any rate up to the GV bound, we can choose any $0 \leq R_{\text{in}} \leq 1 - h(\delta_{\text{in}})$. Flipping things around, for any R_{in} we can achieve distance $\delta_{\text{in}} = h^{-1}(1 - R_{\text{in}})$, where $h^{-1} : [0, 1] \to [0, 1/2]$ is the inverse of the binary entropy function restricted to $[0, 1/2]$ (where h is increasing).

The resulting relative distance of the concatenated code is at least

$$\delta = \delta_{\text{in}}\delta_{\text{out}} = h^{-1}(1 - R_{\text{in}})(1 - R_{\text{out}}),$$

which we can rewrite as $R_{\text{out}} = 1 - \frac{\delta}{h^{-1}(1-R_{\text{in}})}$. Therefore, we can achieve any rate

$$R = R_{\text{in}}R_{\text{out}} = R_{\text{in}}\left(1 - \frac{\delta}{h^{-1}(1 - R_{\text{in}})}\right)$$

with $0 \leq R_{\text{in}} \leq 1 - h(\delta_{\text{in}})$. Since $\delta \leq \delta_{\text{in}}$, we get the *Zyablov bound*, stated next.

Theorem 2 (Zyablov bound) *For any relative distance $\delta < 1/2$ there exists a family of binary linear codes of asymptotic rate*

$$R \geq \max_{0<R_{\text{in}}<1-h(\delta)} R_{\text{in}} \left(1 - \frac{\delta}{h^{-1}(1-R_{\text{in}})}\right)$$

with relative distance at least δ and whose generator matrices can be constructed in time polynomial in the block length.

Figure 1 compares the Zyablov and GV bounds as a function of δ.

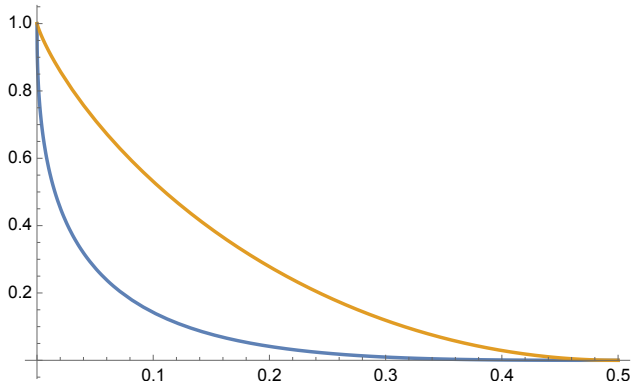


Figure 1: Comparison between the Zyablov bound (in blue) and the GV bound (in orange) as a function of the relative distance δ.

3 Strongly explicit asymptotically good codes

In the previous section we gave, in particular, an asymptotically good binary linear code whose generator matrix G can be constructed in time polynomial in the block length n. But what if we just want to compute a specific entry of G? In other words, what if we only want an algorithm that given indices i and j outputs $G_{i,j}$? Note that the description length of the input to this algorithm is $O(\log n)$. Therefore, it makes sense to aim for an algorithm that runs in time polynomial in the input description length, i.e., polynomial in $\log n$. Codes with this property are said to be *strongly explicit*, and this is an important property for algorithmic applications of error-correcting codes.

Reed-Solomon codes are strongly explicit (can you see why?). To get a strongly explicit asymptotically good binary linear code, it suffices to concatenate a Reed-Solomon outer code with a strongly explicit asymptotically good binary linear inner code (again, can you see why?). The previous strategy of brute-forcing the generator matrix of a linear code on the GV bound will not work, because that is not strongly explicit. Instead, we follow a recipe for code concatenation due to Justesen [Jus72].

The key observations behind Justesen's approach is that to get an asymptotically good binary code via code concatenation (1) we can apply possibly distinct inner codes to distinct coordinates of the outer code, and (2) it suffices that *most* of these inner codes have good distance. Contrast this with

the previous approach, where we used the same inner code for every outer coordinate and required that this single inner code have good distance properties.

Consider now n_{out} inner codes $\mathcal{C}_1, \ldots, \mathcal{C}_{n_{\text{out}}}$, each with block length n_{in}, rate R_{in}, and encoder Enc_i. Then, we define the concatenated code

$$(\mathcal{C}_1, \ldots, \mathcal{C}_{n_{\text{out}}}) \circ \mathcal{C}_{\text{out}} = \{(\mathsf{Enc}_1(c_1), \mathsf{Enc}_2(c_2), \ldots, \mathsf{Enc}_{n_{\text{out}}}(c_{n_{\text{out}}})) : (c_1, \ldots, c_{n_{\text{out}}}) \in \mathcal{C}_{\text{out}}\}.$$

The following theorem states that this concatenated code has good properties whenever most of the inner codes have good properties.

Theorem 3 *Suppose that at least $(1-\varepsilon)n_{\text{out}}$ of the inner codes $\mathcal{C}_1, \ldots, \mathcal{C}_{n_{\text{out}}}$ have relative distance at least δ_{in}. Then, $(\mathcal{C}_1, \ldots, \mathcal{C}_{n_{\text{out}}}) \circ \mathcal{C}_{\text{out}}$ has block length $n = n_{\text{in}} n_{\text{out}}$, rate $R = R_{\text{in}} R_{\text{out}}$, and relative distance at least $\delta = \delta_{\text{in}}(\delta_{\text{out}} - \varepsilon)$.*

Proof: We argue briefly about the distance guarantee. Call a coordinate i *good* if $\mathcal{C}_i$ has relative distance at least δ_{in}, and *bad* otherwise. We know that there are at most $\varepsilon n_{\text{out}}$ bad coordinates. Two distinct outer codewords $c, c' \in \mathcal{C}_{\text{out}}$ differ in at least $\delta_{\text{out}} n_{\text{out}}$ coordinates. Therefore, they differ in at least $(\delta_{\text{out}} - \varepsilon)n_{\text{out}}$ good coordinates. Then, the inner encodings of these good cordinates differ in at least $\delta_{\text{in}} n_{\text{in}}$ coordinates each, leading to overall relative distance at least $\delta_{\text{in}}(\delta_{\text{out}} - \varepsilon)$. ■

It suffices now to present an ensemble of strongly explicit inner codes such that most of these codes have good distance. There is an elegant construction, first described by Massey in 1963 [Mas63] and attributed to John Wozencraft. In fact, this *Wozencraft ensemble* can be interpreted as a small collection of codes *most of which are on the GV bound*!

Lemma 1 (Wozencraft ensemble) *Fix $\varepsilon \in (0, 1/2)$. Then, for all large enough r, there exists an ensemble of binary linear codes $\mathcal{C}_1, \ldots, \mathcal{C}_{2^r-1}$, all of block length $2r$ and rate $1/2$, such that for at least $(1-\varepsilon)(2^r - 1)$ of indices i it holds that $\mathcal{C}_i$ has relative distance at least $h^{-1}(1/2 - \varepsilon)$.*

Proof: Consider the code ensemble $(\mathcal{C}_\alpha)_{\alpha \in \mathbb{F}_q^*}$ such that

$$\mathcal{C}_\alpha = \{\text{bin}(x, \alpha x) : x \in \mathbb{F}_q\},$$

where $\text{bin}(v)$ denotes the representation of $v \in \mathbb{F}_q^\ell$ as an element of $\mathbb{F}_2^{r\ell}$. There are $|\mathbb{F}_q^*| = q - 1$ codes in this ensemble.

The claims about the block length, rate, and linearity are easy, so we focus on the distance properties of this ensemble. We first show that each nonzero $y \in \mathbb{F}_2^{2r}$ belongs to at most one $\mathcal{C}_\alpha$ by case analysis. We interpret y as $y = (y_1, y_2)$ with $y_1, y_2 \in \mathbb{F}_q$.

- If $y_1 \neq 0$ and $y_2 \neq 0$, then $y \in \mathcal{C}_\alpha$ only for $\alpha = y_2 y_1^{-1}$.
- If $y_1 = 0$ and $y_2 \neq 0$, then $y \notin \mathcal{C}_\alpha$ for any α, as $\alpha y_1 = 0$ always.
- If $y_1 \neq 0$ and $y_2 = 0$, then again $y \notin \mathcal{C}_\alpha$ for any α, as $\alpha y_1 \neq 0$ always.

Therefore, the number of $\mathcal{C}_\alpha$'s with relative distance below $h^{-1}(1/2-\varepsilon)$ is at most the number of nonzero vectors y of Hamming weight less than $h^{-1}(1/2-\varepsilon)\cdot 2r$, which is at most

$$\mathrm{Vol}_2(2r, h^{-1}(1/2-\varepsilon)\cdot 2r) \leq 2^{2rh(h^{-1}(1/2-\varepsilon))} = 2^{2r(1/2-\varepsilon)} = 2^{r(1-2\varepsilon)}.$$

To conclude the proof, note that $2^{r(1-2\varepsilon)} < \varepsilon(2^r-1)$ for all sufficiently large r as a function of ε only. ∎

We can now put everything together. Take $\mathcal{C}_{\mathrm{out}}$ to be the Reed-Solomon code over $q=2^r$ with block length n_{out}, set of evaluation points $\mathbb{F}_q^*$, and dimension $k = R_{\mathrm{out}} n_{\mathrm{out}}$. Note that this outer code has relative distance $\delta_{\mathrm{out}} \geq 1 - R_{\mathrm{out}}$. Index the coordinates of this outer code by $\alpha \in \mathbb{F}_q^*$, and take the Wozencraft ensemble from Lemma 1 as the inner codes, for some arbitrarily small but fixed constant $\varepsilon > 0$. Then, the resulting code $(\mathcal{C}_\alpha)_{\alpha\in\mathbb{F}_q^*} \circ \mathcal{C}_{\mathrm{out}}$ is strongly explicit, has block length $n = 2rn_{\mathrm{out}}$, rate $R = R_{\mathrm{out}}/2$, and relative distance

$$\delta \geq h^{-1}(1/2-\varepsilon)(1-R_{\mathrm{out}}-\varepsilon).$$

We can ensure that both $R > 0$ and $\delta > 0$ as $n \to \infty$ by taking R_{out} to be some constant in $(0,1)$ and ε to be a small enough constant depending on R_{out}.

4 Efficient decoding of concatenated codes: The Generalized Minimum Distance decoder

Above, we showed that there exist strongly explicit asymptotically good binary linear codes via code concatenation, more than fulfilling the first goal listed in the introduction. We now turn to the second goal: devising efficient encoding and decoding algorithms for these codes. Here, "efficient" means "time polynomial in the block length".

Recall the construction above, where we take a linear outer code $\mathcal{C}_{\mathrm{out}}$ over $\mathbb{F}_q$ with $q=2^r$ and concatenate it with a binary linear inner code $\mathcal{C}_{\mathrm{in}}$. The fact that $\mathcal{C}_{\mathrm{in}} \circ \mathcal{C}_{\mathrm{out}}$ admits efficient encoding is clear, because linear codes come with efficient encoders by default. Therefore, we focus now on efficient decoding.

First, observe that since the inner code $\mathcal{C}_{\mathrm{in}}$ has block length $n_{\mathrm{in}} = O(r) = O(\log n)$, the naive decoder that given $z \in \mathbb{F}_2^{n_{\mathrm{in}}}$ brute forces over all $2^r = n^{O(1)}$ codewords of $\mathcal{C}_{\mathrm{in}}$ and outputs the one closest to z in Hamming distance (with ties broken arbitrarily) runs in time $n^{O(1)}$ and corrects any $t < d_{\mathrm{in}}/2$ errors. We will denote the output of this decoder on input z by $\mathrm{MLD}_{\mathcal{C}_{\mathrm{in}}}(z)$, where MLD stands for "Maximum Likelihood Decoding".

We start by considering a natural approach. Let $c = (c_1, \ldots, c_{n_{\mathrm{out}}}) \in \mathbb{F}_2^{n_{\mathrm{in}} n_{\mathrm{out}}}$ be a codeword of $\mathcal{C}_{\mathrm{in}} \circ \mathcal{C}_{\mathrm{out}}$, where each $c_i \in \mathbb{F}_2^{n_{\mathrm{in}}}$ represents a codeword of $\mathcal{C}_{\mathrm{in}}$. Suppose that c is corrupted by t errors, yielding $(y_1, \ldots, y_{n_{\mathrm{out}}})$. Then, we can do the following:

1. Apply the inner code decoder to each y_i, obtaining guesses $\tilde{y}_i = \mathrm{MLD}(y_i)$ for each $i \in \{1, \ldots, n_{\mathrm{out}}\}$.

2. Denote the decoder for the outer code $\mathcal{C}_{\text{out}}$ by Dec_{out}. Let z_i denote the element of $\mathbb{F}_2^r$ that is mapped to $\tilde{y}_i$ by $\mathcal{C}_{\text{in}}$. Then, output $\text{Dec}_{\text{out}}(z_1, \ldots, z_{n_{\text{out}}})$.

Theorem 4 *The algorithm above succeeds whenever there are $t < \frac{d_{\text{in}} d_{\text{out}}}{4}$ errors, and runs in time polynomial in $n = n_{\text{in}} n_{\text{out}}$ whenever Dec_{out} runs in time polynomial in n.*

Proof: Let t_i denote the number of errors introduced in y_i (and so $\sum_{i=1}^{n_{\text{out}}} t_i \leq t$). Since $\mathcal{C}_{\text{in}}$ has distance d_{in}, we have $\tilde{y}_i = c_i$ whenever $t_i < d_{\text{in}}/2$. The outer decoder fails only if there are at least $d_{\text{out}}/2$ errors in $z = (z_1, \ldots, z_{n_{\text{out}}})$. But, since $t < d_{\text{in}} d_{\text{out}}/4$, there are fewer than $d_{\text{out}}/2$ coordinates i for which $t_i \geq d_{\text{in}}/2$. ■

This is a good start, but we can hope for better. Recall that $\mathcal{C}_{\text{in}} \circ \mathcal{C}_{\text{out}}$ has distance $d_{\text{in}} d_{\text{out}}$, and so we can hope to efficiently correct any $t < \frac{d_{\text{in}} d_{\text{out}}}{2}$ errors. At a very high level, we will accomplish this by incorporating additional information revealed by maximum likelihood decoding into the decoding process to turn likely errors into erasures, leading to the *Generalized Minimum Distance* (GMD) decoder of Forney [For66].

The high-level structure of GMD decoding is the following. As above, suppose that a codeword $(c_1, \ldots, c_{n_{\text{out}}})$ of $\mathcal{C}_{\text{in}} \circ \mathcal{C}_{\text{out}}$ is corrupted by up to $t < \frac{d_{\text{in}} d_{\text{out}}}{2}$ errors into $(y_1, \ldots, y_{n_{\text{out}}})$. Suppose also that when applying the inner decoder $\text{MLD}_{\mathcal{C}_{\text{in}}}(y_i)$ we generate not only a guess $\tilde{y}_i$ for c_i, but also an *unreliability score* $u_i \in [0, 1]$, to be defined. Intuitively, $u_i = 0$ means that we are pretty sure about $\tilde{y}_i = c_i$, and $u_i = 1$ means that we are not sure at all. Then, GMD decoding roughly works as follows:

1. Choose a threshold $\theta \in [0, 1]$.
2. For every i such that $u_i \geq \theta$, set $z_i =?$ (i.e., treat the i-th coordinate as an erasure). For all other i's, set z_i to be such that $\text{Enc}_{\text{in}}(z_i) = \tilde{y}_i$.
3. Let Dec_{out} be a decoder for $\mathcal{C}_{\text{out}}$ that correctly decodes c from up to t_{error} errors and t_{erasure} erasures whenever $2t_{\text{error}} + t_{\text{erasure}} < d_{\text{out}}$. Output $\text{Dec}_{\text{out}}(z_1, \ldots, z_{n_{\text{out}}})$.

Besides the fact that we have not yet specified how to compute the unreliability score, it is also not clear what threshold θ we should choose. Let's try choosing θ uniformly at random from $[0, 1]$. In this case, the i-th coordinate is erased with probability $\Pr[u_i > \theta] = u_i$.

Under this random choice of θ, let X_i denote the contribution of the i-th coordinate to the count $2t_{\text{error}} + t_{\text{erasure}}$. If $\tilde{y}_i = c_i$, then $X_i = 0$ if the i-th coordinate is not erased, and $X_i = 1$ otherwise. Else, if $\tilde{y}_i \neq c_i$, then $X_i = 2$ if the i-th coordinate is not erased, and $X_i = 1$ otherwise. Therefore,

$$\text{E}_\theta \left[\sum_{i=1}^{n_{\text{out}}} X_i \right] = \sum_{i : \tilde{y}_i = c_i} (0 \cdot (1 - u_i) + 1 \cdot u_i) + \sum_{i : \tilde{y}_i \neq c_i} (2(1 - u_i) + 1 \cdot u_i) = \sum_{i : \tilde{y}_i = c_i} u_i + \sum_{i : \tilde{y}_i \neq c_i} (2 - u_i).$$

Now, we would like to set up our unreliability scores u_i so that

$$\text{E}_\theta \left[\sum_{i=1}^{n_{\text{out}}} X_i \right] < d_{\text{out}}.$$

If this holds, then we know that there exists a choice of the threshold θ that yields $z = (z_1, \ldots, z_{n_{\text{out}}})$ that can be obtained from c via at most some t_{error} errors and t_{erasure} erasures with $2t_{\text{error}} + t_{\text{erasure}} < d_{\text{out}}$, and so $\text{Dec}_{\text{out}}(z) = c$. We can then try various choices for the threshold θ until we hit a good one (we will discuss this more carefully later).

Our unreliability score u_i will be a suitable normalization of the Hamming distance $d(\tilde{y}_i, y_i)$ between our "best" guess $\tilde{y}_i$ and the corrupted inner codeword y_i. When $d(\tilde{y}_i, y_i) = 0$ we should be very confident that $\tilde{y}_i = c_i$, since for this *not* to happen while having $d(\tilde{y}_i, y_i) = 0$ we would have to introduce at least d_{in} errors in c_i, eating up a sizable chunk of the adversary's error budget $t < \frac{d_{\text{in}} d_{\text{out}}}{2}$. On the other hand, if $d(\tilde{y}_i, y_i) \geq d_{\text{in}}/2$, then it is not clear at all whether $\tilde{y}_i = c_i$, since there may exist strings that are at distance $d_{\text{in}}/2$ from several codewords of $\mathcal{C}_{\text{in}}$. Since it is likely that $\tilde{y}_i \neq c_i$ in this case, we treat it as an erasure. We then interpolate linearly between these extremes by setting

$$u_i = \min(1, 2d(\tilde{y}_i, y_i)/d_{\text{in}}).$$

The next key lemma shows that this choice gives us what we want.

Lemma 2 *Under the choice of u_i above we have*

$$\mathrm{E}_\theta \left[\sum_{i=1}^{n_{\text{out}}} X_i \right] = \sum_{i:\tilde{y}_i = c_i} u_i + \sum_{i:\tilde{y}_i \neq c_i} (2 - u_i) \leq \sum_{i=1}^{n_{\text{out}}} \frac{2t_i}{d_{\text{in}}} < d_{\text{out}}.$$

Proof: First, observe that $d(\tilde{y}_i, y_i) \leq t_i$, since $d(c_i, y_i) = t_i$ and $c_i \in \mathcal{C}_{\text{in}}$.

Consider an arbitrary i such that $\tilde{y}_i = c_i$. Then,

$$u_i \leq \frac{2d(\tilde{y}_i, y_i)}{d_{\text{in}}} \leq \frac{2d(c_i, y_i)}{d_{\text{in}}} = \frac{2t_i}{d_{\text{in}}}.$$

On the other hand, consider an arbitrary i such that $\tilde{y}_i \neq c_i$. Note that

$$2 - u_i = 2 - \min(1, 2d(\tilde{y}_i, y_i)/d_{\text{in}}) = \max(1, 2(1 - d(\tilde{y}_i, y_i)/d_{\text{in}})) = \max\left(1, \frac{2}{d_{\text{in}}}(d_{\text{in}} - d(\tilde{y}_i, y_i))\right).$$

The triangle inequality gives

$$d(\tilde{y}_i, y_i) + t_i = d(\tilde{y}_i, y_i) + d(y_i, c_i) \geq d(\tilde{y}_i, c_i) \geq d_{\text{in}},$$

since $c_i, \tilde{y}_i \in \mathcal{C}_{\text{in}}$ are distinct, and so $t_i \geq d_{\text{in}} - d(\tilde{y}_i, y_i)$. Thus,

$$2 - u_i \leq \max\left(1, \frac{2t_i}{d_{\text{in}}}\right) = \frac{2t_i}{d_{\text{in}}},$$

recalling that $t_i \geq d_{\text{in}}/2$ in this case. ∎

Lemma 2 implies that there is a choice of threshold θ, possibly depending on the input corrupted codeword $y = (y_1, \ldots, y_{n_{\text{out}}})$, that makes the GMD decoder work. We now briefly discuss how we can find this good threshold θ. First, note that, for any given choice of θ, the GMD decoder produces some guess $\tilde{c}$ for c, and we can check whether $\tilde{c} = c$ by verifying whether $d(\tilde{c}, y) < \frac{d_{\text{in}} d_{\text{out}}}{2}$

since we are promised at most $t < \frac{d_{\text{in}} d_{\text{out}}}{2}$ errors. Second, while iterating over $[0, 1]$ to find the good θ is certainly not feasible, it holds that the unreliability scores u_i can only take values in the finite subset

$$\left\{0, \frac{2}{d_{\text{in}}}, \frac{4}{d_{\text{in}}}, \ldots, \frac{2\lfloor d_{\text{in}}/2 \rfloor}{d_{\text{in}}}, 1\right\}.$$

Choosing any θ in $(\frac{2j}{d_{\text{in}}}, \frac{2(j+1)}{d_{\text{in}}}]$ leads to the same behavior of the GMD decoder, and so it suffices to run GMD decoding for each choice of θ over this finite subset, which has size $O(n_{\text{in}}) = O(\log n)$, until we find c. This leads to the following theorem.

Theorem 5 *Suppose that $\mathcal{C}_{\text{out}}$ has an efficient decoding algorithm correcting up to t_{error} errors and t_{erasure} erasures for any t_{error} and t_{erasure} such that $2t_{\text{error}} + t_{\text{erasure}} < d_{\text{out}}$. Then, $\mathcal{C}_{\text{in}} \circ \mathcal{C}_{\text{out}}$ can be efficiently decoded from any $t < \frac{d_{\text{in}} d_{\text{out}}}{2}$ errors.*

Finally, it is not hard to see that a Reed-Solomon outer code satisfies the hypothesis of Theorem 5.

References

[For66] George Forney. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, 12(2):125–131, 1966.

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972.

[Mas63] James Massey. Threshold decoding, 1963. MIT Press, https://mitpress.mit.edu/9780262130141/threshold-decoding/.