1 Introduction

In previous lectures, we saw that code concatenation [For65] helps us obtain binary linear codes that achieve Zyablov bound [Zya71], and we even saw an explicit construction for the same through Justesen codes [Jus72]. In this lecture, we focus on how to decode concatenated codes *efficiently*.

In more detail, we want to achieve the following goal:

Goal: Given an outer code C_{out} and an inner code C_{in} with relative distances δ_{out} and δ_{in}, respectively, correct $e < \frac{\delta_{\mathsf{out}} \cdot \delta_{\mathsf{in}}}{2}$ fraction of errors in the concatenated code $C = C_{\mathsf{out}} \diamond C_{\mathsf{in}}$ *efficiently*, i.e,. in time polynomial in block length of C.

Note that a straightforward corollary of achieving this goal would be an *explicit* and *asymptotically good* binary linear code family that can correct up to $\tau(R) = \frac{\delta_{\mathsf{Zyablov}}(R)}{2}$ fraction of errors, where R is the rate of the family.

Before we get into the solutions, we first setup notation for the next section:

- $C_{\mathsf{out}} : \Sigma^k \to \Sigma^n$ is the outer code with distance D.
- $C_{\mathsf{in}} : \Sigma \to \{0,1\}^{n'}$ is the inner code with distance d.
- $\langle c_1, \ldots, c_n \rangle = C_{\mathsf{out}}(m)$ denote codewords in C_{out}, where $m \in \Sigma^k$ and $c_i \in \Sigma$ for all $i \in [n]$.
- $\langle z_1, \ldots, z_n \rangle$ denote the bit-string we want to correct, where $z_i \in \{0,1\}^{n'}$ for all $i \in [n]$.

The notation as well as code concatenation are summarized in Figure 7.1.

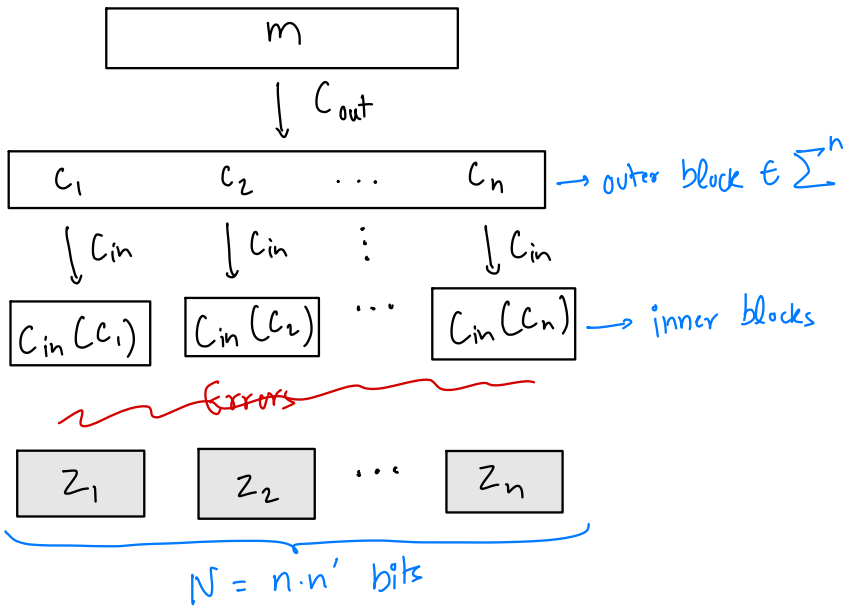

Figure 7.1: Corrupted string $\langle z_1, \ldots, z_n \rangle$ obtained from errors introduced to codeword in $C_{\mathsf{out}} \diamond C_{\mathsf{in}}$.

2 Decoding Concatenated Codes

2.1 Naïve Solution

The naïve solution is to first decode the inner code blocks, and then decode the outer code. In more detail, we have the following two-step process:

1. Inner Decoding: decode each z_i to $a_i \in \Sigma$ such that $\Delta(z_i, C_{\mathsf{in}}(a_i))$ is minimized (breaking ties arbitrarily). Note that even though nearest codeword problem (NCP) is hard, this step can be brute-forced as Σ is small.
2. Outer Decoding: decode $\langle a_1, \ldots, a_n \rangle$ using outer decoder up to decoding radius $\frac{D}{2}$.

Claim 7.1. *If the outer decoder is efficient, so is this solution.*

It is easy to see that the runtime of this solution is $T(\text{outer decoder}) + n \cdot \mathsf{poly}(|\Sigma|, n')$, which is polynomial in block length $N = n \cdot n'$ of C if $T(\text{outer decoder})$ is polynomial in n.

Claim 7.2. *This solution can correct $< \frac{D \cdot d}{4}$ errors.*

Proof. Note that the decoding only fails if $\#\{i \mid a_i \neq c_i\} \geq \frac{D}{2}$, and $a_i \neq c_i$ implies $\Delta(z_i, C_{\mathsf{in}}(c_i)) \geq \frac{d}{2}$. Thus, a failed decoding necessarily implies $\sum_{i=1}^{n} \Delta(z_i, C_{\mathsf{in}}(c_i)) \geq \frac{D \cdot d}{4}$ errors. □

Hence, this simple solution can correct up to $\tau = \frac{\delta_{\mathsf{out}} \cdot \delta_{\mathsf{in}}}{2}$ fraction of errors, but we lost a factor of 2. Next, we see how to achieve the goal we set out at the start of this lecture.

2.2 Generalized Minimum Distance Decoding

The high level idea behind Generalized Minimum Distance (GMD) Decoding [For66] is that in addition to providing a_i's obtained from correcting z_i's like in first solution, the inner decoder also provides more guidance to outer decoder through "soft information". Specifically, we have the following algorithm:

1. Inner Decoding: decode each z_i to $a_i \in \Sigma$ such that $u_i = \Delta(z_i, C_{\mathsf{in}}(a_i))$ is minimized (breaking ties arbitrarily). Compute $w_i = \min(\frac{d}{2}, u_i)$ and also pass this "soft-info" to the outer decoder.
2. Outer Decoding: for each a_i, set it as an *erasure* with probability $\frac{2w_i}{d}$ (let θ denote the random coins), and then run errors and erasure decoding on $\langle a_1, \ldots, a_n \rangle$ using outer decoder.

Lemma 7.3. *Let $e_i = \Delta(z_i, C_{\mathsf{in}}(c_i))$. If $\sum_{i=1}^{n} e_i < \frac{D \cdot d}{2}$, then $\mathop{\mathbf{E}}_{\theta}\left[2\mathcal{Z}^{\mathsf{errors}} + \mathcal{Z}^{\mathsf{erasures}}\right] < D$, where $\mathcal{Z}^{\mathsf{errors}}$ and $\mathcal{Z}^{\mathsf{erasures}}$ are random variables that count the number of errors and erasures, respectively.*

Proof. Let $\mathcal{Z}_i^{\mathsf{errors}}$ denote the random variable that measures the probability of a_i being input as an error to the outer decoder, and similarly define $\mathcal{Z}_i^{\mathsf{erasures}}$. It is easy to see that $\mathcal{Z}^{\mathsf{errors}} = \sum_{i=1}^{n} \mathcal{Z}_i^{\mathsf{errors}}$ and $\mathcal{Z}^{\mathsf{erasures}} = \sum_{i=1}^{n} \mathcal{Z}_i^{\mathsf{erasures}}$. Now, we prove that for each $i \in [n]$, $\mathbf{E}\left[2\mathcal{Z}_i^{\mathsf{errors}} + \mathcal{Z}_i^{\mathsf{erasures}}\right] \leq \frac{2e_i}{d}$, which will prove $\mathbf{E}\left[2\mathcal{Z}^{\mathsf{errors}} + \mathcal{Z}^{\mathsf{erasures}}\right] < D$ as $\sum_{i=1}^{n} e_i < \frac{D \cdot d}{2}$ and complete the proof.

Case I ($a_i = c_i$): since $\mathbf{E}\left[\mathcal{Z}_i^{\mathsf{errors}}\right] = 0$ and $\mathbf{E}\left[\mathcal{Z}_i^{\mathsf{erasures}}\right] = \frac{2w_i}{d} = \frac{2e_i}{d}$, $\mathbf{E}\left[2\mathcal{Z}_i^{\mathsf{errors}} + \mathcal{Z}_i^{\mathsf{erasures}}\right] = \frac{2e_i}{d}$.

Case II ($a_i \neq c_i$): here $\mathbf{E}\left[\mathcal{Z}_i^{\mathsf{errors}}\right] = 1 - \frac{2w_i}{d}$ and $\mathbf{E}\left[\mathcal{Z}_i^{\mathsf{erasures}}\right] = \frac{2w_i}{d}$. Since $w_i + e_i = \Delta(z_i, C_{\mathsf{in}}(a_i)) + \Delta(z_i, C_{\mathsf{in}}(c_i)) \geq \Delta(C_{\mathsf{in}}(a_i), C_{\mathsf{in}}(c_i)) \geq d$, we have $\mathbf{E}\left[2\mathcal{Z}_i^{\mathsf{errors}} + \mathcal{Z}_i^{\mathsf{erasures}}\right] = 2 - \frac{2w_i}{d} \leq \frac{2e_i}{d}$. □

It follows from Lemma 7.3 that there exists a choice of θ for which the outer decoder will be successful. Thus, we can simply brute-force search for such a θ if we can sufficiently reduce its search space. Now, we discuss how this can be done.

Concretely, the outer decoding procedure can sample θ as follows: pick a threshold $\theta \in [0,1]$ uniformly at random and set a_i as an erasure if $\theta \leq \frac{2w_i}{d}$. It is easy to see that instead of checking all $\theta \in [0,1]$, we can simply check $\theta \in \Theta = \{0, \frac{2w_1}{d}, \frac{2w_2}{d}, \ldots, \frac{2w_n}{d}, 1\}$ to learn all possible outcomes. Since $w_i \in \{0, \ldots, \frac{d}{2} - 1\}$, size of $\Theta \in O(d)$, and we can *efficiently* bruth-force search for θ. Thus, we have proved the following theorem:

Theorem 7.4. *The GMD decoding algorithm runs in* $\mathsf{poly}(N)$ *time and can correct up to* $\frac{D \cdot d}{2}$ *errors.*

3 Parallel Concatenation

So far, we've only seen serial concatenation where inner blocks are concatenated in a serial fashion, i.e., without any interaction (see Figure 7.2).

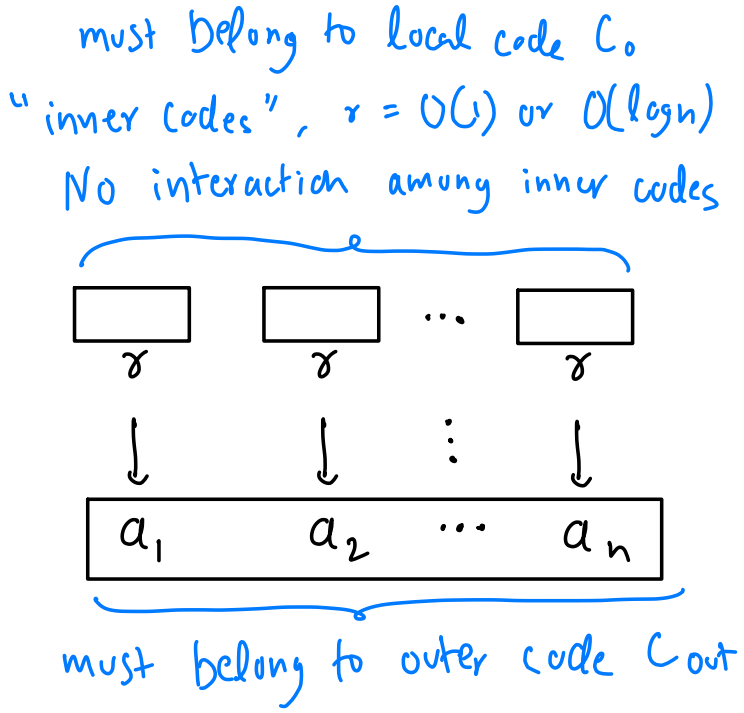

Figure 7.2: Serial Concatenation

Now, we look at parallel concatenation where we stipulate that inner blocks overlap and different overlapping parts all belong to a local code $C_0 \subseteq \{0,1\}^r$ (see Figure 7.3). Specifically, given a set $\mathcal{S} \subseteq \binom{[n]}{r}$, a codeword c belongs to a parallelly concatenated code $C_1 \subseteq \{0,1\}^n$ if for all $s \in \mathcal{S}$, we have $[c]_s \in C_0$, where $[c]_s$ is codeword c projected onto coordinates in s.

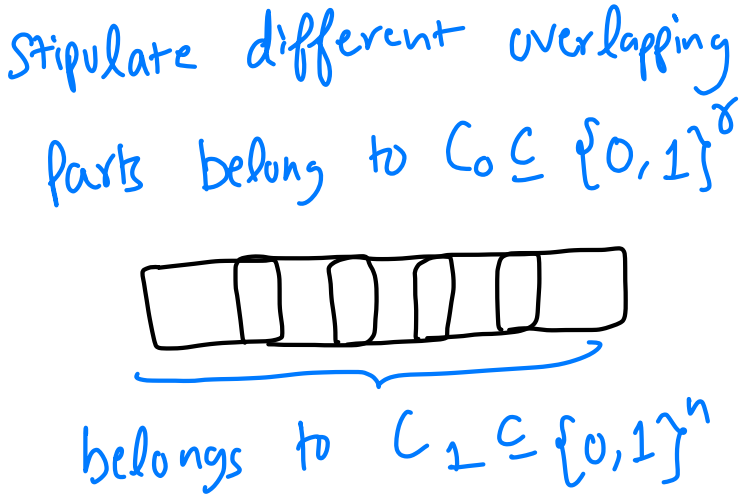

Figure 7.3: Parallel Concatenation

Now, we look at a concrete construction of a parallelly concatenated code.

3.1 Tensor Product Code

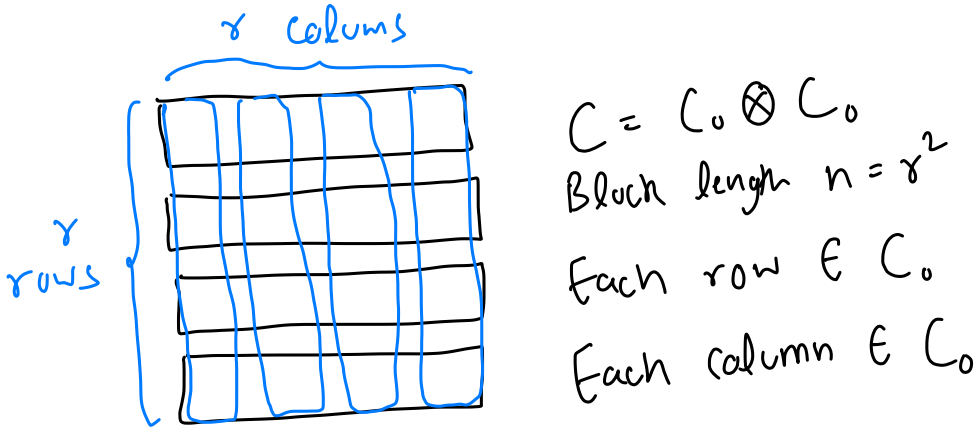

Figure 7.4: Tensor Product Code

Claim 7.5. *Distance of $C = C_0 \otimes C_0$ is at least d^2.*

Proof. Assume that the first column is non-zero. Since it belongs to C_0, it must have 1's in at least d locations. Note that the rows corresponding to these locations are also non-zero and must have at least d 1's as they also belong to C_0. Thus, every codeword in C must have weight at least d^2, which implies that the distance is at least d^2 as C is linear code. □

Claim 7.6. *Dimension of $C = C_0 \otimes C_0$ is s^2.*

The proof for this claim is left as an exercise.

The two claims combined with the fact that block length of $C_0 \otimes C_0$ is r^2 give us the following theorem:

Theorem 7.7. *If C_0 is a $[r, s, d]_{\mathbb{F}_q}$ code, then the product code $C = C_0 \otimes C_0$ is a $[r^2, s^2, d^2]_{\mathbb{F}_q}$ code.*

References

[For65] G. David Forney. Decoding concatenated codes. 1965. 1

[For66] G. David Forney. Generalized minimum distance decoding. *IEEE Trans. Inf. Theory*, 12(2):125–131, 1966. 2.2

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Trans. Inf. Theory*, 18(5):652–656, 1972. 1

[Zya71] V. V. Zyablov. An estimate of the complexity of constructing binary linear cascade codes. *Problemy Peredachi Informatsii*, 7:5–13, 1971. 1