In today's lecture, we will begin by reviewing Reed-Solomon codes, and discussing their relation to Bose–Chaudhuri–Hocquenghem (BCH) codes. RS codes are the first class of codes we will see which match the Singleton/Pigeonhole bound, at the cost of a large alphabet size. After which, the rest of today's lecture will be dedicated to code concatenation, a technique to decrease alphabet size while almost preserving rate and distance.

1 Reed-Solomon Codes

Given an alphabet (field) $\mathbb{F}$, a set $S \subset \mathbb{F}$ of $|S| = n$ distinct evaluation points in the field, and an integer degree parameter $0 < k < n$, the degree $k-1$ RS code over S is defined by

Definition 5.1 ([RS60])**.** If $S \subset \mathbb{F}$, $|S| = n, S = \{a_1, \cdots, a_n\}$,

$$RS_{\mathbb{F}}(S,k) = \left\{ (f(a_1), \cdots, f(a_n)) : (f_0, \cdots, f_{k-1}) \in \mathbb{F}^k, f(X) = \sum_i^{k-1} f_i X^i \right\} \tag{5.1}$$

Each codeword of the degree $k-1$ RS code corresponds a distinct degree $k-1$ polynomial over $\mathbb{F}$. Linearity of the $RS_{\mathbb{F}}(S,k)$ code follows from the linearity over the polynomial coefficients. We note that via Lagrange interpolation, given any $\geq k$ elements of a codeword (say, $(c_0, \cdots c_{k-1}) = (f(a_1) \cdots f(a_k))$), one can decode the unique degree $k-1$ polynomial $(f_0, \cdots f_{k-1})$ which passes through those points.

Theorem 5.2. *The $RS_{\mathbb{F}}(S,k)$ code with $|S| = n < |\mathbb{F}|$ is a $[n, k, n-k+1]_{|\mathbb{F}|}$ error-correcting code.*

To achieve error resilience, the RS code 'oversamples' the number of points to decode. Any two degree $k-1$ polynomials can agree on at most $k-1$ points, since their difference can have at most $k-1$ roots. In this manner, Reed-Solomon codes match the Singleton bound, and are referred to as Maximum Distance Separable (MDS) codes.

Their encoding is remarkably simple, as its $n \times k$ generator matrix $G_{S,k}$ is the well-known Vandermonde matrix over elements $S = \{a_1, \cdots, a_n\}$:

$$G_{S,k} = \begin{bmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{k-1} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{k-1} \\ 1 & \vdots & \ddots & \cdots & a_3^{k-1} \\ \vdots & & & & \\ 1 & a_n & a_n^2 & \cdots & a_n^{k-1} \end{bmatrix} \tag{5.2}$$

Thus $(G_{S,k})_{ij} = a_i^j$.

Exercise 5.3. A 'systematic' encoding of an error correcting code places the k symbol message as the first k symbols of the codeword. Give a systematic encoding of Reed-Solomon codes.

1.1 The Dual RS Code and Parity Check Matrix

Informally, the dual of the RS code is also a RS code. While technically not exactly accurate, by introducing a slight generalization to the RS code definition, we will be able to recover this intuition. Beforehand, let us first introduce a converse view of the RS code, in terms of its parity-check matrix. Given a field $\mathbb{F}$ with primitive element α, let $\mathbb{F}^*$ be its non-zero elements.

Theorem 5.4 (The Parity-Check View of Reed-Solomon Codes)**.**

$$RS_{\mathbb{F}}(\mathbb{F}^*, k) = \left\{ (c_0, \ldots, c_{n-1}) \in \mathbb{F}^n : c(\alpha^i) = 0 \text{ for all } i \in [n-k], \text{ where } c(X) \equiv \sum_{i=0}^{n-1} c_i X^i \right\} \quad (5.3)$$

That is, the 'parity'-checks are defined by treating the codeword as coefficients of a degree $n-1$ polynomial over $\mathbb{F}$. Naturally, by 'parity,' we refer to a q-ary parity, as the polynomial evaluations are over $\mathbb{F}$. The proof of the theorem above follows from an insightful fact over $\mathbb{F}$.

Fact 5.5. *For any $\gamma \in \mathbb{F}$ with $\gamma \neq 0, 1$, we have the identity $\sum_{j=0}^{q-2} \gamma^j = \frac{1-\gamma^{q-1}}{1-\gamma} = 0$ where $q = |\mathbb{F}|$.*

From the parity-check description above, we can extract the dual RS code:

Corollary 5.6 (Dual RS Code)**.**

$$(RS_{\mathbb{F}}(\mathbb{F}^*, k))^{\perp} = \left\{ (g(1), \alpha g(\alpha), \alpha^2 g(\alpha^2), \cdots, \alpha^{n-1} g(\alpha^{n-1})) \in \mathbb{F}^n : deg(g) < n-k \right\} \quad (5.4)$$

That is, the dual RS code is a $[n, n-k, k+1]_{\mathbb{F}}$ code, which is again MDS. Its codewords are evaluations of degree $n-k-1$ polynomials over $\mathbb{F}$, multiplied by certain scalars at each symbol. It is in this sense that RS codes are only 'informally' self-dual. However, by introducing multipliers inside the RS definition, we are able to make this formally correct.

Definition 5.7. Pick symbols $V = \{v_1, \cdots v_n\} \subset \mathbb{F}$, distinct symbols $S \subset \{a_1, \cdots a_n\}$, and an integer $0 < k < n$. Then the Generalized Reed-Solomon code is defined by

$$GRS_{\mathbb{F}}(V, S, k) \equiv \left\{ (v_1 f(a_1), v_2 f(a_2) \cdots v_n f(a_n)) \in \mathbb{F}^n : deg(f) < k \right\}. \quad (5.5)$$

We can now recover the clean self-dual intuition:

Theorem 5.8. *In the context of the GRS definition above,*

$$(GRS_{\mathbb{F}}(V, S, k))^{\perp} = GRS_{\mathbb{F}}(U, S, n-k), \quad (5.6)$$

where $U = \{u_1, \ldots, u_n\} \subset \mathbb{F}$ and $u_i^{-1} = v_i \prod_{j \neq i} (a_j - a_i)$.

Exercise 5.9. Prove Theorem 5.8 above.

2 Relation to the BCH Code

Let us recall the definition of the BCH code:

Definition 5.10 (Binary BCH codes [BRC60])**.** Fix two positive integers m, D and a primitive element $\alpha \in \mathbb{F}_{2^m}^*$, the Binary BCH code is defined by

$$BCH[n, D] \equiv \left\{ (c_0, \ldots, c_{n-1}) \in \mathbb{F}_2^n : c(\alpha^i) = 0 \text{ for all } i \in [D-1], \text{ where } c(X) \equiv \sum_{i=0}^{n-1} c_i X^i \right\}, \quad (5.7)$$

where the block length is $n = 2^m - 1$.

Note the remarkable similarity with the definition in Theorem 5.4. The key distinction is that here, we constrain the symbols in the code to be binary, as opposed to any element in the extension field. Each constraint $c(\alpha^i) = 0$ is a linear equation over $\mathbb{F}_{2^m}$, but it can equivalently be viewed as m linear equations over $\mathbb{F}_2$. In this sense, in fact

Claim 5.11. *The Binary BCH code is a 'subfield subcode' of a RS code, that is:*

$$BCH[2^m - 1, D] = RS_{\mathbb{F}_{2^m}}(\mathbb{F}_{2^m}^*, 2^m - D) \cap \mathbb{F}_2^{2^m - 1} \quad (5.8)$$

The proof of which follows by careful inspection of Theorem 5.4 and Definition 5.10. Essentially, the claim states that we can take the codewords of a RS code, whose symbols happen to evaluate over $\mathbb{F}_2$, and they will compose a well-defined binary linear code.

The astute reader may note a slight redundancy in the parity check definition of the BCH codes in Definition 5.10. Indeed, since each codeword is defined over a binary alphabet, it would suffice to constrain $c(\alpha^i) = 0$ for odd powers i. We formalize this discussion in the following theorem

Theorem 5.12. *$BCH[n, D]$ is a $[n, k, D]_2$ code of dimension $k \geq n - \frac{D-1}{2} \log(n+1)$.*

Proof. As defined in Definition 5.10, the codewords would seem to be constrained by $(D-1) \cdot m = (D-1) \cdot \log(n+1)$ linear equations, and thus the dimension of $BCH[n, D]$ is at least $n - (D-1)\log(n+1)$. To shave off the factor of 2, we note that there is in fact a linear dependence between certain parity checks. For any even power we note $c(\alpha^{2p}) = (c(\alpha^p))^2 = 0$, since $c_i \in \mathbb{F}_2$ and $(a+b)^2 = a^2 + b^2$ for $a, b \in \mathbb{F}_{2^m}$. In this manner, Definition 5.10 introduces at most $\frac{(D-1)}{2} \cdot \log(n+1)$ linearly independent constraints. □

Note 5.13. BCH codes effectively meet the Hamming bound, but aren't asymptotically good since their distances reach at most $O(n/\log n)$.

3 Code Concatenation

Can we explicitly define asymptotically good codes over small alphabets? Recall that RS codes aren't just asymptotically good; they are in fact optimal, since they hit the Singleton bound. If we are fine with slightly sacrificing the optimality of RS codes, can we transform them into a smaller alphabet size?

A simple transformation to obtain binary linear codes from RS codes is to define a linear bijection $\phi : \mathbb{F}_{2^m} \to \mathbb{F}_2^m$, such that one can map any RS codeword $(f(a_1) \cdots f(a_n)) \in \mathbb{F}_{2^m}^n \to (\phi(f(a_1)) \cdots \phi(f(a_1))) \in \mathbb{F}_2^{n \cdot m}$ onto a binary alphabet. This mapping preserves the code rate, since intuitively there is no added redundancy, and both the block length and the code dimension are scaled up by a factor of $m = \log(n+1)$: $n' = n \cdot m$ and $k' = k \cdot m$.

Unfortunately, the distance of the resulting code remains $n - k + 1$. That's because if any two polynomial evaluations differ, say $f(a_1) \neq g(a_1)$, their images under ϕ may only differ by 1 bit, instead of the total m bits in $\phi \circ f$. In this manner, the resulting composed code is a $[n \cdot \log(n+1), k \cdot \log(n+1), \geq n - k + 1]_2$ code, which is even worse than BCH codes!

Remark 5.14. While a single linear map ϕ doesn't suffice, one can show that there exists a sequence $\phi_1 \cdots \phi_n$ of linear maps, one for each symbol in the codeword, which achieves the GV bound. However, these ϕ_i aren't explicit, as they are randomly sampled.

What if instead of encoding each symbol in the RS code using a bijective map, we added some redundancy to the final encoding?

Definition 5.15 (Concatenated Codes, [For65])**.** Given a $[n,k]_{\mathbb{F}_{q^m}}$ outer code C_{out} and a $[n',m]_{\mathbb{F}_q}$ inner code C_{in}, the concatenated code $C \equiv C_{out} \diamond C_{in}$ is defined by the composition of C_{in} and a bijective linear map $\phi : \mathbb{F}_{q^m} \to \mathbb{F}_q^m$ on each symbol of C_{out}.

Image 5.1 below illustrates the code concatenation.

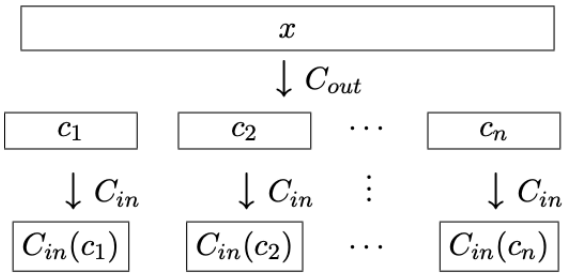

Figure 5.1: Code Concatenation, from [BG10]

We easily read off the dimension and block length of the concatenated code.

Claim 5.16. *The concatenated code C is a $[n \cdot n', k \cdot m]_q$ linear code.*

In this manner, the rate of the resulting code is the product of rates of C_{out}, C_{in}.

Claim 5.17. *If C_{out}, C_{in} have distances d_{out}, d_{in} respectively, then the distance of the concatenated code $C_{out} \diamond C_{in}$ is $\geq d_{out} \times d_{in}$.*

Proof. Consider two codewords c, c' of C_{out}. By definition, they differ in at least d_{out} locations. Consider one such location i, with distinct symbols $c_i, c_i' \in \mathbb{F}_{q^m}$. Their image under composition in C_{in} is a n'-dimensional vector over $\mathbb{F}_q$. Since $c_i \neq c_i'$ by assumption, $C_{in}(c_i), C_{in}(c_i')$ must differ in at least d_{in} locations. Thus $d(C_{out} \diamond C_{in}) \geq d_{out} \times d_{in}$. □

3.1 Good Concatenated Codes and the Zyablov Bound

From the previously derived properties of concatenated codes, we conclude that to obtain asymptotically good binary codes from RS codes, we require an $[m/r, m, \delta_{in} \cdot m]_2$ inner code with $r, \delta_{in} = \Omega(1)$. We can construct such a code through a greedy method in $2^{O(m)}$ time, with $r = 1/2$, $\delta_{in} \approx h^{-1}(1/2) \approx 0.11$ matching the GV bound. Here, we denote $h^{-1}(x)$ as the inverse entropy function, with domain $[0,1]$ and image $[0,1/2]$. Since in the RS code construction $m = \log(n+1)$, this construction can be instantiated in polynomial time in the block length. More generally,

Theorem 5.18 (The Zyablov Bound, [Zya71])**.** *One can construct in $n^{O(1)}$ time a family of binary linear codes of rate R and distance $\delta_{Zyablov}(R) = \max_{r:R \leq r \leq 1}(1 - \frac{R}{r})h^{-1}(1-r)$.*

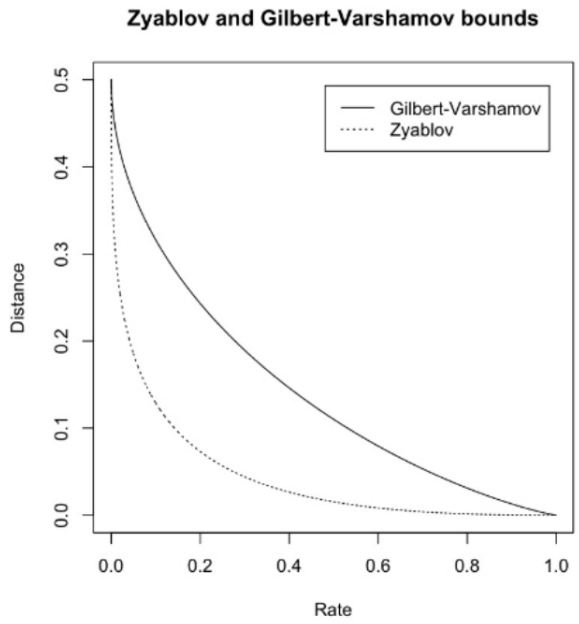

Figure 5.2: The Zyablov and GV bounds, from [BG10]

In Figure 5.2, we compare the GV and Zyablov bounds in terms of the target rate R. Note that for all non-vanishing R, $\delta_{Zyablov}(R) > 0$, and thus there exist explicit asymptotically good binary codes for all intended target rates.

Remark 5.19. It is instructive to consider the limits of the GV and Zylabov bound as the relative distance δ approaches 0 and $1/2 - \epsilon$. We observe $R_{Zylabov}(\delta = 1/2 - \epsilon) = \Omega(\epsilon^3)$ while $R_{GV}(\delta = 1/2 - \epsilon) = \Omega(\epsilon^2)$; whereas when $\delta \to 0$, we have $R_{Zyablov}(\delta) = 1 - O(\sqrt{\delta} \log 1/\delta)$, while $R_{GV}(\delta) = 1 - O(\delta \log 1/\delta)$.

Remark 5.20. In 2017, Ta-Shma constructed a family of explicit, almost optimal, 'epsilon-balanced codes' (of relative distance around $1/2 - \epsilon$) of rate $\epsilon^{2+o(1)}$, close to the GV bound [TS17].

3.2 Justesen Codes

The codes defined by the Zyablov bound, while efficiently constructible, relied on a brute force search over inner codes which matched the intended rate/distance guarantees. They left open the question of whether such constructions could be made fully explicit.

Justesen [Jus72] defined the first family of fully explicit binary codes which matched the Zyablov bound, at least over a large range of rates R. Justesen's key insight was that the choice of inner code C_{in} doesn't have to be unique, similarly to the idea in Remark 5.14. Moreover, it actually suffices that most of these inner codes have 'good' distance properties - instead of a strong guarantee on every inner code. By combining these insights with a Reed-Solomon code as an outer code, Justesen constructed a family of 'expanded' RS codes, using a particularly simple choice of inner codes:

Definition 5.21 (Binary Justesen Codes [Jus72])**.** Fixed positive integers $k, n = 2^m - 1$, a subset $S \subset \mathbb{F}_{2^m}$ of n distinct points $S = \{a_1, \cdots, a_n\}$, and a bijective linear map $\phi : \mathbb{F}_{2^m} \to \mathbb{F}_2^m$, the Justesen code is defined by

$$JRS_m(S,k) \equiv \left\{ (\phi \circ f(a_1), \phi \circ (a_1 f(a_1)), \cdots, \phi \circ f(a_n), \phi \circ (a_n f(a_n))) \in \mathbb{F}_2^{2nm} : deg(f) < k \right\} \quad (5.9)$$

That is, the Justesen code corresponds to the evaluation of polynomials $f(X)$ and $Xf(X)$ on the evaluation set S, and viewed as elements of $\mathbb{F}_2^{2m}$ through the action of ϕ. Analogous to the definition of RS codes, the $k \cdot m$ bit message defines the degree $k-1$ polynomial f over $\mathbb{F}_{2^m}$.

Theorem 5.22 below formalizes the rate/distance tradeoff of these codes, as well as a more general tradeoff result based on a generalization of the codes in Definition 5.21.

Theorem 5.22 (The Justesen Tradeoff [Jus72])**.** *The Binary Justesen Codes of Definition 5.21 define a family of explicit binary linear codes of rate $R < 1/2$ and relative distance $\delta(R) = (1 - 2R)h^{-1}(1/2) - o(1)$. More generally, for any $R \in (0,1)$, there are explicit binary linear codes of rate at least R and relative distance at least*

$$\delta_{Justesen}(R) = \max_{r \geq \max(1/2, R)} \left(1 - \frac{R}{r}\right) h^{-1}(1-r) - o(1) \tag{5.10}$$

Note 5.23. $\delta_{Justesen}(R) = \delta_{Zyablov}(R)$ for $R \geq 0.31$.

Note 5.24. The key reason the Justesen code doesn't achieve the Zyablov bound at lower rates is that we don't have small inner code ensembles of rate $r < 1/2$ where most of the codes meet the GV bound.

While we won't fully cover the analysis of these codes in today's lecture, the intuition behind the correctness of Justesen's scheme lies in the following exercise:

Exercise 5.25 (The Wozencraft ensemble)**.** Fix a bijective linear map $\phi : \mathbb{F}_{2^m} \to \mathbb{F}_2^m$. Consider the family of $[2m, m]_2$ codes $C = \{C_\alpha : \alpha \in \mathbb{F}_{2^m}, \alpha \neq 0\}$, where $C_\alpha : \mathbb{F}_2^m \to \mathbb{F}_2^{2m}$ is defined by

$$C_\alpha(x) = (x, \phi(\alpha \cdot \phi^{-1}(x))) \tag{5.11}$$

Prove that all but an $1 - o(1)$ fraction of the codes in the family C have relative distance $h^{-1}(1/2) - o(1)$, matching the GV bound.

References

[BG10] Eric Blais and Venkat Guruswami. Notes 6: Reed-solomon, bch, reed-muller, and concatenated codes. *Introduction to Coding Theory, CMU*, 2010. 5.1, 5.2

[BRC60] R. C. Bose and Dwijendra K. Ray-Chaudhuri. On a class of error correcting binary group codes. *Inf. Control.*, 3:68–79, 1960. 5.10

[For65] G. David Forney. Concatenated codes. 1965. 5.15

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Trans. Inf. Theory*, 18:652–656, 1972. 3.2, 5.21, 5.22

[RS60] Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of The Society for Industrial and Applied Mathematics*, 8:300–304, 1960. 5.1

[TS17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 2017. 5.20

[Zya71] Victor Vasilievich Zyablov. An estimate of the complexity of constructing binary linear cascade codes. *Problemy Peredachi Informatsii*, 7(1):5–13, 1971. 5.18