Class 4 Exercises

CS250/EE387, Winter 2025

1. Fix $\alpha_0, \alpha_1, \ldots, \alpha_r \in \mathbb{F}_q$. Fix $y_0, y_1, \ldots, y_r \in \mathbb{F}_q$. Let

$$f(X) = \sum_{i=0}^{r} y_i \frac{L_i(X)}{L_i(\alpha_i)},$$

where

$$L_i(X) = \frac{\prod_{j=0}^{r}(X - \alpha_j)}{X - \alpha_i} = \prod_{j \neq i}(X - \alpha_j).$$

(Note that L_i depends on the definition of the α_j's).

(a) Show that $f(\alpha_\ell) = y_\ell$ for all $\ell = 0, \ldots, r$. (If you haven't seen this before, this is called *Lagrange Interpolation.*)

(b) Explain what part (a) has to do with the fact (which we saw in the lecture videos/notes) that a Reed-Solomon code is MDS (Maximum Distance Separable).

(c) In the lecture videos/notes, we defined a natural encoding map for a Reed-Solomon code $RS(\vec{\alpha}, n, k)$ by

$$(f_0, \ldots, f_{k-1}) \mapsto (f(\alpha_0), \ldots, f(\alpha_{n-1}))$$

for evaluation points $\alpha_0, \ldots, \alpha_{n-1}$. Use part (a) to give a *systematic* encoding map for $RS(\vec{\alpha}, n, k)$: that is, an encoding map of the form

$$(x_0, \ldots, x_{k-1}) \mapsto (x_0, \ldots, x_{k-1}, z_k, z_{k+1}, \ldots, z_{n-1})$$

where the message symbols appear as the first k symbols of the codeword.

(If you finish this one early, jump to Question 3 and then 4, and we'll come back together before embarking on Question 2).

Solution

(a) We can just plug in:

$$f(\alpha_j) = \sum_{i=1}^{r} y_i \frac{L_i(\alpha_j)}{L_i(\alpha_i)} = y_j,$$

because $L_i(\alpha_j) = 0$ if $j \neq i$.

(b) Part (a) tells us how to do polynomial interpolation: given any r pairs (α_i, y_i), we can come up with a degree-r polynomial that goes through those evaluation points. One way to define an MDS code of dimension k is that any k points completely determine a codeword. Applying part (a), with $r \leftarrow k-1$, we see that any k points completely determines a degree $< k$ polynomial, and hence a Reed-Solomon codeword (which are the evaluations of that polynomial).

(c) We interpolate $f(X)$ as in part (a) so that $f(\alpha_i) = x_i$ for $i = 0, \ldots, k-1$. Then f has degree at most $k-1$. Then we set $z_j = f(\alpha_j)$ for $j \geq k$.

2. Fix $\vec{\lambda} = (\lambda_1, \ldots, \lambda_n) \in \mathbb{F}^n$ and $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}^n$ so that the λ_j's are all nonzero and the α_j's are all distinct. The **generalized** Reed-Solomon code $GRS(\vec{\lambda}; \vec{\alpha}, n, k)$ of dimension k is given by

$$GRS(\vec{\lambda}; \vec{\alpha}, n, k) = \{(\lambda_1 f(\alpha_1), \lambda_2 f(\alpha_2), \ldots, \lambda_n f(\alpha_n)) \,:\, f \in \mathbb{F}[X], \deg(f) < k\}.$$

(a) What is the generator matrix for $GRS(\vec{\lambda}; \vec{\alpha}, n, k)$? Convince yourself that generalized RS codes are MDS codes.

Solution

The generator matrix is given by $D_{\vec{\lambda}} \cdot G$, where $D_{\vec{\lambda}}$ is the diagonal matrix with $\vec{\lambda}$ on the diagonal, and G is a Vandermonde matrix (the generator matrix for $RS(\vec{\alpha}, n, k)$). GRS codes are MDS since any $k \times k$ submatrix of this generator matrix is full rank; that's true because we saw in the lecture videos/notes that it was true for G, and multiplying by $D_{\vec{\lambda}}$ won't change that.

(b) Forget about generalized RS codes for a moment. Fix distinct $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$. Show that, for any polynomial $h(X)$ with $\deg(h) < n-1$, we have

$$\sum_{i=1}^{n} \frac{h(\alpha_i)}{L_i(\alpha_i)} = 0,$$

where

$$L_i(X) = \prod_{j \neq i} (X - \alpha_j)$$

as in the previous problem.
Hint: Write out $h(X)$ using Lagrange interpolation with all n points $\alpha_1, \alpha_2, \ldots, \alpha_n$. What is the coefficient on X^{n-1} when you write it out this way?

Solution

Following the hint, we can write

$$h(X) = \sum_i h(\alpha_i) \frac{L_i(X)}{L_i(\alpha_i)}.$$

In this view, the coefficient on X^{n-1} is

$$\sum_i \frac{h(\alpha_i)}{L_i(\alpha_i)} \cdot (\text{Coeff on } X^{n-1} \text{ in } L_i(X)).$$

We observe that $L_i(X) = \prod_{j \neq i}(X - \alpha_j)$ has degree exactly $n-1$, and that the leading coefficient is 1. (That's because the only way to get X^{n-1} in this product is to take the "X" from each term $(X - \alpha_j)$). Thus, the coefficient on X^{n-1} in $h(X)$ is

$$\sum_i \frac{h(\alpha_i)}{L_i(\alpha_i)}.$$

On the other hand, the degree of $h(X)$ is at most $n-2$ by assumption. So the coefficient on X^{n-1} is zero. This is what we wanted to show.

(c) Back to GRS codes.

i. Show that $RS(\vec{\alpha}, n, k)^{\perp} = GRS(\vec{\lambda}; \vec{\alpha}, n, n-k)$ for some vector $\vec{\lambda}$. What is $\vec{\lambda}$, in terms of $\vec{\alpha}$?

Solution

We can use part (b). Let $\lambda_i = \frac{1}{L_i(\alpha_i)}$. Let $f(X)$ be a degree $< k$ polynomial corresponding to a codeword c of the RS code, and let $g(X)$ be a degree $< n-k$ polynomial corresponding to a codeword c' of the GRS code with weights λ_i. Then the degree of $h(X) = f(X)g(X)$ is at most $k-1+n-k-1 = n-2 < n-1$, so we apply part (b) to $h(X)$. We get that

$$\sum_{i=1}^{n} c_i c'_i = \sum_{i=1}^{n} f(\alpha_i)\lambda_i g(\alpha_i) = \sum_i \frac{f(\alpha_i)g(\alpha_i)}{L_i(\alpha_i)} = 0.$$

ii. Find a parity-check matrix for $RS(\vec{\alpha}, n, k)$, in terms of $\vec{\alpha}$?

Solution

A parity-check matrix for $RS(\vec{\alpha}, n, k)$ is the transpose of a generator matrix for its dual, which by part i is $GRS(\vec{\lambda}; \vec{\alpha}, n, n-k)$, where $\lambda_i = 1/L_i(\alpha_i)$. So it is $V^T \cdot D$, where $V \in \mathbb{F}_q^{n-k\times n}$ is Vandermonde with evaluation points $\alpha_1, \ldots, \alpha_n$, and $D \in \mathbb{F}_q^{n\times n}$ is diagonal with $1/L_i(\alpha_i)$ in the i'th diagonal position.
Notice that this generalizes the example we saw in the lecture videos where $\vec{\alpha} = (\gamma^0, \gamma^1, \ldots, \gamma^{q-2})$. In that case, $L_i(\alpha_i) = 1$ for all i.

iii. **(Bonus)** More generally, show that $GRS(\vec{\lambda}; \vec{\alpha}, n, k)^{\perp} = GRS(\vec{\sigma}; \vec{\alpha}, n, n-k)$ for some $\vec{\sigma}$. What is $\vec{\sigma}$, in terms of $\vec{\lambda}$ and $\vec{\alpha}$? What is the parity-check matrix of $GRS(\vec{\lambda}; \vec{\alpha}, n, k)$?

Solution

The same proof as above works, but we should take $\sigma_i = \frac{1}{\lambda_i L_i(\alpha_i)}$. (And the parity-check matrix is the same as above, just with the σ_i instead of the λ_i).

3. **(Bonus, if time)** Let $n = q-1$ and suppose that $f : \mathbb{F}_q \to \mathbb{F}_q$ given by $f(X) = \sum_{i=0}^{n-1} f_i X^i$ and $g : \mathbb{F}_q \to \mathbb{F}_q$ given by $g(X) = \sum_{i=0}^{n-1} g_i X^i$ are polynomials that both vanish on $\gamma, \gamma^2, ..., \gamma^{n-k}$, for a primitive element γ of $\mathbb{F}_q$. Prove that the polynomial $h(X)$ given by

$$h(X) = \sum_{i=0}^{n-1} f_i g_i X^i$$

vanishes on $\gamma, \gamma^2, \ldots, \gamma^{n-2k+1}$.

Hint: There is a short proof using something from the lecture videos/notes...

Solution

By the dual view of RS codes, the coefficients f_i of $f(X) = \sum_{i=0}^{n-1} f_i X^i$ are evaluations of a polynomial $\tilde{f}$ of degree at most $k-1$: that is, $f_i = \tilde{f}(\alpha^i)$ for $i = 1, \ldots, n$. The same is true for g and a degree-$\leq k-1$ polynomial $\tilde{g}$. Thus, the coefficients $h_i = f_i \cdot g_i$ are given by

$$h_i = \tilde{g}(\alpha^i) \cdot \tilde{f}(\alpha^i).$$

Now, the polynomial $\tilde{h}(X) = \tilde{f}(X) \cdot \tilde{g}(X)$ has degree at most $2k-2$, and so by the duality of RS codes again, the polynomial $h(X)$ vanishes on $\alpha, \alpha^2, \ldots, \alpha^{n-2k+1}$, as desired.

4. **(Extra Bonus, if even more time)**

(a) Over finite fields, we can define something called a *Hasse* derivative, as follows. Let $f \in \mathbb{F}_q[X]$ be a polynomial over $\mathbb{F}_q$. Then the k'th Hasse derivative of f is denoted $f^{(\ell)} \in \mathbb{F}_q[X]$, and is defined to satisfy the Taylor-like expansion:

$$f(X) = \sum_{\ell=0}^{\deg(f)} f^{(\ell)}(a)(X-a)^\ell \qquad\qquad \forall a \in \mathbb{F}_q.$$

(Note: you can define $f^{(\ell)}(X)$ directly by defining the ℓ'th Hasse derivative of X^r as $\binom{r}{\ell}X^{r-\ell}$ and extending linearly, but this isn't important for this problem).

Let $\alpha_1, \alpha_2, \ldots, \alpha_n \subseteq \mathbb{F}_q$ be n distinct evaluation points. Define a new code C (called a *derivative code* or *univariate multiplicity code*) as follows. Let $C \subseteq (\mathbb{F}_q^m)^n$ be defined by taking all codewords of the form

$$\left(\begin{bmatrix} f(\alpha_1) \\ f^{(1)}(\alpha_1) \\ \vdots \\ f^{(m-1)}(\alpha_1) \end{bmatrix}, \begin{bmatrix} f(\alpha_2) \\ f^{(1)}(\alpha_2) \\ \vdots \\ f^{(m-1)}(\alpha_2) \end{bmatrix}, \cdots, \begin{bmatrix} f(\alpha_n) \\ f^{(1)}(\alpha_n) \\ \vdots \\ f^{(m-1)}(\alpha_n) \end{bmatrix} \right) \in (\mathbb{F}_q^m)^n$$

for all polynomials f of degree at most $k-1$.

i. What is the rate of this code?

ii. What is the distance?

Hint: You may use the fact that a degree d polynomial has at most d roots *counting multiplicities*. For example, $g(X) = (X-1)^2$ has a root of multiplicity 2 at $X = 1$, which means that it can't have any more roots. Formally, the *multiplicity* of a root α is $\ell + 1$ for the largest ℓ so that $f(\alpha), f^{(1)}(\alpha), \ldots, f^{(\ell)}(\alpha)$ all vanish. For example, $g^{(1)}(X) = 2X - 2$, so $g(1) = g^{(1)}(1) = 0$; and $g^{(2)}(X) = 2$ which doesn't vanish at $X = 1$, so the multiplicity of the root at 1 is 2.

iii. What happens if $k > n$? Does it still make sense?

(b) Notice that we can write $f(\alpha)$ as "$f(X) \mod (X - \alpha)$". Thus, if we like, we can write an RS codeword as:

$$(f(X) \mod (X-\alpha), f(X) \mod (X-\alpha_2), \ldots, f(X) \mod (X-\alpha_n)).$$

For $f \in \mathbb{F}_q[X]$ of degree at most $k-1$, consider the corresponding codeword

$$(f(X) \mod (X-\alpha_1)^m, f(X) \mod (X-\alpha_2)^m, \ldots, f(X) \mod (X-\alpha_n)^m).$$

Here, we can think of this as a polynomial of degree at most $m-1$ (since we can view it as the remainder when we divide by a polynomial of degree m). Thus, we can also think about it as a vector in $\mathbb{F}_q^m$ (eg, the coefficients of that degree at most $m-1$ polynomial).

Consider the code that we get when taking all such codewords (for all $f \in \mathbb{F}_q[X]$ of degree at most $k-1$). This code lies in $(\mathbb{F}_q^m)^n$. Show that it is equivalent to the derivative code in the previous part (possibly up to taking a fixed linear transformation $L_i : \mathbb{F}_q^m \to \mathbb{F}_q^m$ in each coordinate i).

(c) (Super open-ended) Try to develop a theory of all codes that you can get by starting with a polynomial and modding out by a polynomial $E_i(X)$ in the i'th position. What do you think their rate and distance will be? What assumptions do you need on the polynomials that you mod out by? Can you think of any other examples that have a standalone interpretation?

Solution

(a) i. The rate is k/mn. That's because we want to encode k symbols over $\mathbb{F}_q$ and end up encoding nm of them. (Formally, the rate is $\frac{\log_{q^m}|C|}{n} = \frac{\log(q^k)}{\log(q^m)n} = \frac{k}{mn}$).

ii. The distance is $n - k/m$ (assuming m divides k; otherwise it's this up to some floors/ceilings and/or ±1).

First, we show that the distance is at least this. Let f be a polynomial of degree at most $k-1$, and consider the codeword corresponding to f. If the i'th symbol vanishes, then $f^{(\ell)}(\alpha_i) = 0$ for $\ell = 0, 1, \ldots, m-1$. This means that f has a root of order m at α_i. Since there are at most k roots with multiplicity, this means that there are at most $\lfloor k/m \rfloor$ places that this codeword can vanish. So the distance is at least $n - \lfloor k/m \rfloor$. (Here we use the fact that this code is still linear over $\mathbb{F}_q$, so it suffices to look at the lowest-weight codeword.)

To see that this is tight, consider the polynomial $(X-\alpha_1)^m (X-\alpha_2)^m \cdots, (X-\alpha_{k/m})^m$. The corresponding codeword vanishes on k/m places.

iii. It does make sense! The reason that we couldn't do this with Reed-Solomon codes is that X^q is the same as X over $\mathbb{F}_q$, so if $k \geq q$, then we'd have two polynomials, e.g. X^q and X, that would map to the same codeword. More generally, X^n is the same as ($X^n \mod \prod_{i=1}^{n}(X-\alpha_i)$) if we only look at evaluations on $\alpha_1, \ldots, \alpha_n$, which is why we can't take $k > n$. If we did, our encoding map wouldn't be injective.

But for multiplicity codes, we can have $k > n$! To get some intuition for this, let's suppose that $n = q$ and consider $f(X) = X^q$ and $g(X) = X$. Now $f^{(1)}(X) = qX^{q-1} = 0$ in $\mathbb{F}_q$, while $g^{(1)}(X) = 1$ in $\mathbb{F}_q$. So even though $f(\alpha) = g(\alpha)$ for all $\alpha \in \mathbb{F}_q$, if we look at $(f(\alpha), f^{(1)}(\alpha))$ and $(g(\alpha), g^{(1)}(\alpha))$, these will actually always be different.

Formally, the way to see this is to look at our distance proof above. It's enough to show that no polynomial of degree at most $k-1$ will have all zeros, and the distance proof above shows that this will happen as long as $k/m < n$, meaning that $k < nm$. So we can actually take k much bigger than n!

(b) Consider $f(X) \mod (X-\alpha)^m$. By the Taylor expansion formula above, we can write

$$f(X) = \sum_{\ell=0}^{k-1} f^{(\ell)}(\alpha)(X-\alpha)^\ell.$$

Thus, mod $(X-\alpha)^m$, all of the terms with $\ell \geq m$ vanish, and we are left with

$$f(X) \mod (X-\alpha)^m = \sum_{\ell=1}^{m-1} f^{(\ell)}(\alpha)(X-\alpha)^\ell.$$

The coefficients of this polynomial are exactly the vector

$$\begin{bmatrix} f^{(0)}(\alpha) \\ f^{(1)}(\alpha) \\ \vdots \\ f^{(m-1)}(\alpha) \end{bmatrix},$$

aka the α'th symbol of the encoding with the derivative code.

(Note that this isn't quite the same as the description above of how we represent $f(X) \mod (X-\alpha)^m$, since there we'd take the coefficients when we write it out in the $(1, X, X^2, \ldots, X^{m-1})$ basis, rather than the $(1, X-\alpha, (X-\alpha)^2, \ldots, (X-\alpha)^{m-1})$ basis. But you can convince yourself that changing from one basis to another is just a linear transformation, which we can capture as the L_i noted in the problem statement.)