Problem Set 4

Information and Coding Theory

March 14, 2021 CLAYTON SEITZ

Problem 0.1. *This is the first problem*

Solution.

$$\begin{aligned}\Delta(C) &= \min_{x_1,x_2\in C} \Delta(x_1,x_2)\\ &= \min_{x_1,x_2\in C} \Delta(0,x_2-x_1)\\ &= \min_{x\in C} \mathbf{wt}(x)\end{aligned}$$

Since the code is linear, $x_2 - x_1 \in C$. Now, we consider the parity check matrix $H \in \mathbb{F}_2^{r\times n}$ where $n = 2^r - 1$. We will find the dimension, block length, and distance for such a code. First, the dimension of the code $\dim(C)$ is $r+1$ since the rank of H is r. The block length is then 2^{r+1} and the distance is 3. Now, consider the Hamming code $C : \mathbb{F}_2^k \to \mathbb{F}_2^n$ which is formally defined as the set of x in the null space in of the parity check matrix:

$$C = \{x \in \mathbb{F}_2^n | Hx = 0\}$$

where $H \sim \mathbb{F}_2^{k\times n}$ is the parity check matrix. We can also define the dual code $C^\perp$ to be the code with generator matrix H^T and parity check matrix G^T.

To see why this is possible, we will use the fact that we have defined our code C to be the vectors x that lie in the null space of the parity matrix H. Now, the definition of our code requires that $H(x) = H(G(w)) = 0$ which means that the generator matrix G is a matrix with columns equal to the basis vectors of the null space of H i.e. $HG = 0$. This is equivalent to saying that the columns of H^T form the basis of the null space of G^T:

$$HG = 0 \iff G^T H^T = 0$$

Therefore H^T can be viewed as the generator matrix and G^T the parity check matrix for the dual code $C^{\perp}$.

■

Problem 0.2. *Good distance codes from linear compression*

We would like to prove that an arbitrary compression algorithm $\text{Com} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ implemented by the matrix $H \sim \mathbb{F}_2^{m \times n}$ yields a good compression scheme for a sequence $Z \sim (\text{Bern}(p))^n$ which means that there exists a decompression algorithm $\text{Decom} : \mathbb{F}_2^m \to \mathbb{F}_2^n$ with the following error bound

$$\underset{Z \sim (\text{Bern}(p))^n}{\mathbf{Pr}} [\text{Decom}(HZ) \neq Z] \leq 2^{-t}$$

Solution.

To prove such a bound, we can first rewrite the LHS of the above. Notice that since $m < n$ whether or not we have a $\text{Decom} : \mathbb{F}_2^m \to \mathbb{F}_2^n$ that comes close to a bijection depends on the distribution over the input: $Z \sim (\text{Bern}(p))^n$.

Assuming that $\text{im}(H) = \mathbb{F}_2^m$, if we consider all possible elements $w \in \mathbb{F}_2^m$, the "lack of bijectivity" is captured by the expected size of a set of inputs that map to a particular output. However, it is simpler to think of the inverse problem where a draw from the input distribution is equal to the decompression of w: $\underset{Z \sim (\text{Bern}(p))^n}{\mathbf{Pr}} [\text{Decom}(w) = Z]$. Therefore,

$$\begin{aligned}\underset{Z \sim (\text{Bern}(p))^n}{\mathbf{Pr}} [\text{Decom}(HZ) \neq Z] &= 1 - \underset{Z \sim (\text{Bern}(p))^n}{\mathbf{Pr}} [\text{Decom}(HZ) = Z] \\ &= 1 - \sum_{w \in \mathbb{F}_q^m} \underset{Z \sim (\text{Bern}(p))^n}{\mathbf{Pr}} [\text{Decom}(w) = Z]\end{aligned}$$

Now, let's say there is a set of inputs that map to the same output $S = \left\{ z \in \mathbb{F}_q^n | Hz = w \right\}$. Since we have said that $p < \frac{1}{2}$, the decompression algorithm that picks the z with minimal weight i.e.

$$\text{Decom}(w) := \underset{z \in S}{\text{argmin}} \{S\}$$

minimizes $\underset{Z \sim (\text{Bern}(p))^n}{\mathbf{Pr}} [\text{Decom}(w) = Z]$. This can be seen if we let $z^* \in S$ be the element with minimal weight

$$\underset{Z \in S}{\mathbf{Pr}}\left[\mathrm{wt}(z) \geq \mathrm{wt}(z^*)\right] = \sum_{t > \mathrm{wt}(z)} \binom{n}{t} \cdot p^t (1-p)^{n-t} \to 0$$

as $p \to 0$.

■

Problem 0.3. *Mixing polynomials*

Solution.

We are given two sequences of values $(b_1, \dots, b_n)$ and $(c_1, \dots, c_n)$ which are the result of evaluating polynomials f_1 and f_2 at points a_i, respectively. Notice that for any particular a_i we have that the sum $f_1(a_i) + f_2(a_i) = b_i + c_i$ and the product $f_1(a_i) \cdot f_2(a_i) = b_i \cdot c_i$ which of course do not change upon swapping b_i and c_i. If y is the sequence of values received, then we can write a bivariate polynomial

$$\begin{aligned} h(x,y) &= y^2 - y(f_1(x) + f_2(x)) + f_1(x) \cdot f_2(x) \\ &= (y - f_1(x))(y - f_2(x)) \end{aligned}$$

If we can perform such a factorization of $h(x,y)$ then we can descramble $f_1(x)$ and $f_2(x)$.

In the second case, we are given a value β_i at each point in the domain a_i but we don't know whether β_i came from $f_1(x)$ or $f_2(x)$. However, we are given the guarantee that the number of points coming from $f_1(x)$ satisfies $\frac{n}{3} \leq n_1 \leq \frac{2n}{3}$ and the points coming from $f_2(x)$ satisfies $\frac{n}{3} \leq n_2 \leq \frac{2n}{3}$ where $n = n_1 + n_2$.

We can recast this problem by thinking of the points that came from one polynomial, say $f_1(x)$, as "errors" and define an error polynomial e that is zero when $y \neq f_1(x)$. Then, we can use the Reed-Solomon decoding scheme to solve for $f_2(x)$. Once we have $f_2(x)$, finding $f_1(x)$ is straightforward: use Lagrange interpolation again on the difference between $f_2(x)$ and y.

Recall that Lagrange interpolation requires that $k \leq n - t$ where n is the total number of points, t the number of errors, and k is the degree of the polynomial to interpolate. We can still apply Lagrange interpolation here since $k \leq \frac{n}{3}$ because $t = \frac{2n}{3}$ is the maximum number of "errors" which must be true since we have already been told $k < \frac{n}{6}$.

■