Information and Coding Theory. Homework 2

Andrei Dzis

March 2019

1 Problem

1.1 Solution

1. Let's find parameters of code with generation matrix G:

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$n = 9$ is obviously, due to the fact that number of columns of generation matrix equals to 9. For $k = 3$, because rank of generation matrix and the dimension of our linear code as a vector subspace equals to 3. The distance d of the linear code is the minimum weight of its nonzero codewords, or equivalently, the minimum distance between distinct codewords. So we need to find codewords, count its weights and after that find the minimum weight among non-zero codewords. Using that fact, that we have binary alphabet, number of codewords equals to $2^3 = 8$ and they are:

$$\{000000000\}, \{001010110\}, \{110001000\}, \{000110001\},$$

$$\{111011110\}, \{001100111\}, \{110111001\}, \{111101111\}$$

With weigths $w(\bar{c})$ (amount of non-zero bits):

$$w(\{000000000\}) = 0$$

$$w(\{001010110\}) = 4$$

$$w(\{110001000\}) = 3$$

$$w(\{000110001\}) = 3$$

$$w(\{111011110\}) = 7$$

$$w(\{001100111\}) = 5$$

$$w(\{110111001\}) = 6$$

$$w(\{111101111\}) = 8$$

So for distance d of linear code:

$$d = \min_{c \neq \{000000000\}} w(\bar{c}) = w(\{110001000\}) = w(\{110001000\}) = 3 \quad (1)$$

2. Let's represent the code in a systematic form. For this we need to represent G in form:

$$G = \left[I_k | P\right],$$

where I_k is the $k \times k$ identity matrix and P, a $k \times (n-k)$ matrix. This can be done with using adding rows to each other and columns permutations:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \sim$$

$$\sim \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

So, finally:

$$G_{\text{systematic}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

3. Now let's fing parity check matrix H for our linear code with given G generation matrix:

$$H = \left[-P^\top | I_{n-k}\right]$$

In our case:

$$H_{\text{systematic}} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For this matrix $G_{\text{systematic}} H_{\text{systematic}}^\top = [0]$, $[0]$ - zero matrix size $k \times (n-k)$ can be easily proved. Systematic presentation of our code preserves the correcting properties of the original code, but has different codewords. Thus, I am to write down the parity check matrix for original generator matrix:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

4. Dual code for code with generation matrix G is a code, for which generation matrix is H. For this code $n = 9$, $k = 6$. Using same procedure as for original one, dual code's distance equals $d = 2$.

2 Problem

2.1 Solution

It's known that parity check codes have even weights of their codewords. So for product code's (based on two parity check codes) codebook we assume, that we have even weights for every row and column in our codeword matrix (matrix, where rows are codewords for one parity check code, columns are codewords for another one).

- **Case for w = 4**

  Our goals is to find maximal number of codewords of weight 4 to satisfy properties of even rows and columns. Let's start with properties of even weights for rows. To make rows weights even, we can put in one row only 2 ones or 4 ones, situation with 1 or 3 ones in one row are not suitable (because of single parity check code properties). Let's go into details for this situations:

  1. We put 4 ones in one row. Thus, we received codeword with weight $w = 4$, but for columns we didn't achieve properties for weights, because in every column only 1 one is put. The solution is to put additional 4 ones to corresponding columns, which is not suitable for our problem statement.
  2. We put only 2 ones in a single row. So we have 2 rows with 2 ones in each of them. The point is to satisfy the even property for columns. This can be done easily: we need to choose 2 columns and for row put ones in positions, corresponding to this columns. To explain this solution, let's chose one row and put 2 ones in random places. For this points we have coordinate representation:

     $$1_1 = (i_1, j_1), 1_2 = (i_1, j_2)$$

     where i_1 denotes our row, and we fix columns coordinates as j_1 and j_2. So to make our columns properties satisfied, we need to put another 2 ones in positions:

     $$1_3 = (i_2, j_1), 1_4 = (i_2, j_2)$$

  So the number of all possible codewords with weight $w = 4$ is the amount of possible combinations of choosing 2 columns and 2 rows, on which intersections to put ones. This probabilities are equal to $C^2_{n_1}$, $C^2_{n_2}$, so the final number of this codewords:

  $$N = C^2_{n_1} C^2_{n_2} \tag{2}$$

- **Case for w = 6**

  As the previous one, choosing to satisfy row even property, we can choose such options:

  1. **6 ones in one row.**
     This is not suitable, because we don't satisfy column properties and we have to add additional ones to satisfy for columns.
  2. **4 ones in one row, 2 in another.**
     This is also not suitable, because we are to add additional ones to satisfy column even properties.
  3. **2 ones in 3 rows** This is the only opportunity to satisfy both columns and rows even properties. But the point is that we need to do quite same technique as for case $w = 4$, but with one addition: we should place 2 ones in 3 rows such, that that for corresponding columns there were only 2 ones inserted. Using notations from the previous case, we should put ones in the following positions:

     $$1_1 = (i_1, j_1), 1_2 = (i_1, j_2), 1_3 = (i_2, j_2),$$

     $$1_4 = (i_2, j_3), 1_5 = (i_3, j_3), 1_6 = (i_3, j_1)$$

     This is one of opportunities. So for counting all possible codewords, we should select first of all 3 rows and 3 columns and place 6 ones in positions of intersections to satisfy the represented in scheme above (2 ones in rows and 2 in columns). All possible positioning can be counted easily and equals to 6.

  So the final answer is:

  $$N = 6C_{n_1}^3 C_{n_2}^3$$

3 Problem

3.1 Solution

As the extended Hamming code we assume Hamming code [7,4] extended to an [8,4] by adding an extra parity bit on top of the [7,4] encoded word. For extended code the minimum distance has increased from 3, in the [7,4] code, to 4 in the [8,4] code (wiki fact). Now we need to find generator polynomial $g(x)$. For $g(x)$ for cyclic code (n, k) we have, that it must be polynomial divider of polynomial $x^n - 1$ of degree $r = n - k = 4$, due to that fact that we work with cyclic code in $GF(2^3)$, we can assume:

$$x^8 - 1 = |\text{in } GF(2^3)| = x^8 + 1 = (x + 1)^8$$

From that fact, we can easily denote, that:

$$g(x) = (x + 1)^4 = |\text{in } GF(2^3)| = x^4 + 1$$

Let's code the following information (in right column polynomial representation $u(x)$):

$$\{1000\} \to 1$$
$$\{0100\} \to x$$
$$\{0010\} \to x^2$$
$$\{0001\} \to x^3$$

So for codewords:

$$c = u(x)g(x)$$

Thus:

$$c_1 = x^4 + 1$$
$$c_2 = x^5 + x$$
$$c_3 = x^6 + x^2$$
$$c_4 = x^7 + x^3$$

For example $c_1 = x^4 + 1 = \{10010000\} \to w(c_1) = 2$. But as we said, for extended Hamming code minimal distance $d = 4$, that means that non-zero codeword weight can't be less that d, but we received codeword with weight 2, so extended Hamming code can't be represented as cyclic code.

4 Problem

4.1 Solution

First of all, codeword $\{111\ldots111\}$ corresponds to $c = x^{n-1}+x^{n-2}+\cdots+x+1$. For generator polynomial $g(x)$ for cyclic code (n, k) we have, that it must be polynomial divider of polynomial $x^n - 1$. That means, that $x^n - 1$ polynomial can be represented as a product of two polynomials: $g(x)$ and the remaining one, let's denote as $f(x)$. So we have:

$$x^n - 1 = g(x)f(x)$$

Now let's deal with polynomial:

$$x^n-1 = (x-1)(x^{n-1}+x^{n-2}+\cdots+x+1) = |\text{in } GF(2^m)| = (x+1)(x^{n-1}+x^{n-2}+\cdots+x+1)$$

We receive:

$$x^n - 1 = g(x)f(x) = (x + 1)(x^{n-1} + x^{n-2} + \cdots + x + 1)$$

If we divide both equations on $x + 1$,we receive:

$$\frac{g(x)f(x)}{x+1} = x^{n-1} + x^{n-2} + \cdots + x + 1$$

Using that fact that $x^n - 1/(x+1)$, but $(x+1) \nmid g(x) \to f(x)/(x+1)$. Let's denote $\frac{f(x)}{(x+1)} = u(x)$. Now we have:

$$u(x)g(x) = x^{n-1} + x^{n-2} + \cdots + x + 1$$

As we can see we encoded information, defined by polynomial $u(x)$ and received codeword with all ones without any restrictions on polynomials $g(x)$ and $f(x)$, only by given one for $g(x)$. So we proved problem statement.

5 Problem

5.1 Solution

- Let's construct a finite field $GF(2^3)$ modulo a polynomial $\varphi(x) = x^3+x+1$ For root of polynomial (denoted in task as α) we receive:

  $$\alpha^3 = \alpha + 1$$

  Let's start constructing a field:

  $$\alpha^0 = 1$$

  $$\alpha^1 = \alpha$$

  $$\alpha^2 = \alpha^2$$

  $$\alpha^3 = \alpha + 1$$

  $$\alpha^4 = \alpha^2 + \alpha$$

  $$\alpha^5 = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1$$

  $$\alpha^6 = \alpha^3 + \alpha^2 + \alpha = \alpha^2 + \alpha + \alpha + 1 = \alpha^2 + 1$$

- Let's construct a [7,5] Reed-Solomon code over $GF(2^3)$ with roots:

  $$\alpha_1 = \alpha, \alpha_2 = \alpha^2$$

  Let's start with generator polynomial:

  $$g(x) = (x - \alpha_1)(x - \alpha_2) = (x - \alpha)(x - \alpha^2) = x^2 - \alpha^2 x - \alpha x + \alpha^3 =$$

  $$= x^2 + x(-\alpha - \alpha^2) + \alpha^3 = x^2 + \alpha^4 x + \alpha^3 = x^2 + \alpha^4 x + \alpha^3$$

  For Reed-Solomon code's distance:

  $$d = n - k + 1 = 7 - 5 + 1 = 3$$

  We finally have RS code with generator polynomial $g(x)$ with parameters $[7, 5, 3]$.
  For parity check matrix for our code we have:

  $$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^8 & \alpha^{10} & \alpha^{12} \end{bmatrix} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \end{bmatrix}$$

- Let's decode vector:

$$y = \begin{bmatrix} \alpha & 0 & \alpha & 1 & \alpha^6 & \alpha^3 & \alpha^2 \end{bmatrix}$$

We need to calculate syndromes S_1 and S_2, which are calculated as dot product of y and each row of H:

$$S_1 = \alpha * 1 + 0 * \alpha + \alpha * \alpha^2 + 1 * \alpha^3 + \alpha^6 * \alpha^4 + \alpha^3 * \alpha^5 + \alpha^2 * \alpha^6 =$$

$$= \alpha + \alpha^3 + \alpha^3 + \alpha^{10} + \alpha^8 + \alpha^8 = \alpha + \alpha^{10} = \alpha + \alpha^3 = \alpha + \alpha + 1 = 1$$

$$S_2 = \alpha * 1 + 0 * \alpha^2 + \alpha * \alpha^4 + 1 * \alpha^6 + \alpha^6 * \alpha + \alpha^3 * \alpha^3 + \alpha^2 * \alpha^5 =$$

$$= \alpha + \alpha^5 + \alpha^6 + \alpha^7 + \alpha^6 + \alpha^7 = \alpha + \alpha^5 = \alpha + \alpha^2 + \alpha + 1 = \alpha^2 + 1 = \alpha^6$$

So from this facts, we can easily find coefficients for error locator polynomial $\sigma(x) = \sigma_1 x + \sigma_0$, where σ_0, σ_1 can be easily found from equations:

$$S_1 + \sigma_0 = 0 \rightarrow \sigma_0 = 1$$

$$S_2 + S_1\sigma_1 = 0 \rightarrow \sigma_1 = \alpha^6$$

So error locator polynomial:

$$\sigma(x) = \alpha^6 x + 1$$

It's easy to see, that α is the root of error locator polynomial, because $\sigma(\alpha) = \alpha^7 + 1 = 0$. To find error location, we need to find inverse symbol for $x = \alpha$. It's easy to see, that inverse element is α^6, because:

$$\alpha\alpha^{-1} = 1, \alpha\alpha^6 = 1 \rightarrow \alpha^{-1} = \alpha^6 = x_0$$

That means, that error locator x_0 points at 7-th bit in y. Now we need to calculate error value, as follows (denote error as ϵ):

$$S_1 = \epsilon x_0 \rightarrow \epsilon = S_1(x_0)^{-1} = 1 * (\alpha^6)^{-1} = \alpha$$

Now corrected 7th bit looks like:

$$y_7 = \alpha^2 + \alpha = \alpha^4$$

Now the corrected y looks like:

$$y = \begin{bmatrix} \alpha & 0 & \alpha & 1 & \alpha^6 & \alpha^3 & \alpha^4 \end{bmatrix}$$

To find original encoded data, we need to divide corrected y in polynomial representation $y(x)$ by $g(x)$. The result is:

$$u(x) =\rightarrow \begin{bmatrix} 1 & \alpha^6 & \alpha^6 & \alpha^4 & \alpha^6 \end{bmatrix}$$

6 Problem

6.1 Solution

For dual code, generator matrix is the parity check matrix for original code, in another words:

$$G_{\text{dual}} = H$$

Let's have a closer look at G_{dual}: If we choose submatrix of $m-1$ rows and columns, starting rows from the 2-nd row, we receive identity matrix of the correspondent size. From that fact comes, that:

$$Rg(G_{\text{dual}}) \geq m-1. \tag{3}$$

What is more, because of the fact, that all possible positions of 2 ones in a column represented in matrix means that full sum of all rows equals to zero-vector, that means in case of rank that:

$$Rg(G_{\text{dual}}) < m. \tag{4}$$

From 4 and 3 comes the fact that:

$$Rg(G_{\text{dual}}) = m-1.$$

To find weight enumerator we need to find the number of codewords of weight w in code X, n is the length of code. We need to count amount of codewords of fixed weight, let's start with the information vector of the fixed weight (let this variable be denoted as w).
Imagine we have information vector with w ones. The point is to realize, when ones occur in codeword. Due to the fact, that we have codeword's element as the dot product of information vector and column of generator matrix, we can count this number. Imagine, we are talking about j-th element of a codeword (that means we are to talk about dot product of information vector and and j-th column of generator matrix, with k and m positions of ones in it), in this position one can occur only if we have the one in position of k and zero in position m of the information vector. So according to this, there are $\binom{m}{w}$ information vectors with w ones to be encoded in codewords with weights $w(m-w)$.
Important notice, that two information vectors can be encoded in same codeword if they are inverse to each other, i.e information vector of all ones and all zeros. That means, that for $\binom{m}{w}$ information vectors there are $\frac{1}{2}\binom{m}{w}$ codewords with weighs calculated above. From this comes the fact, that there are in total $\frac{1}{2}2^m$ codewords, so:

$$|C_{\text{dual}}|^{-1} = \frac{1}{2^{m-1}}$$

So, finally for enumerator:

$$W_{\text{dual}}(x,y) = \sum_{w=0}^{m} \frac{1}{2}\binom{m}{w} x^{w(m-w)} y^{\binom{m}{2}-w(m-w)}$$

And for original, using MacWilliams identity:

$$W(x,y) = |C_{\text{dual}}|^{-1} W_{\text{dual}}(y-x, y+x) =$$

$$= \frac{1}{2^{m-1}} \sum_{w=0}^{m} \frac{1}{2} \binom{m}{w} (y-x)^{w(m-w)} (y+x)^{\binom{m}{2} - w(m-w)} =$$

$$= \frac{1}{2^{m}} \sum_{w=0}^{m} \binom{m}{w} (y-x)^{w(m-w)} (y+x)^{\binom{m}{2} - w(m-w)}$$

7 Problem

7.1 Solution

There are a lot of papers of trellis representation of linear block code, among them:

- "On the BCJR Trellis for Linear Block Codes" by Robert J. McEliece.
- "Decoding of convolutional codes using a syndrome trellis" by V. Sidorenko and V.Zyablov.

Now I will explain the method: So code $\mathcal{C}$ consists of all vectors, for which:

$$c^{(j)} = (c_1^{(j)}, \ldots, c_n^{(j)}) : H(c^{(j)})^T = c_1^{(j)} h_1 + \ldots c_n^{(j)} h_n = 0,$$

where $H(h_1, \ldots, h_n)$ - is a parity matrix for code $\mathcal{C}$.
(**Here is answer for a)**) Since vertex set of trellis consists of 2^{n-k} vertices at depth $i = 0, \ldots, n$ and each vertex is identified by binary vector of length 2^{n-k} called state of vertex.
From that fact comes, that there are $2^{n-k}(n+1)$ vertices (or states) in trellis.
Edges of trellis are defined by codewords of the given code $\mathcal{C}$. Let's denote the j-th codeword as $c^{(j)} = (c_1^{(j)}, \ldots, c_n^{(j)})$), for which there exist n corresponding labeled edges e in the trellis forming a path of length n defined as

$$init(e^{(j)}) = \sum_{i=1}^{j-1} C_i h_i,$$

$$fin(e^{(j)}) = \sum_{i=1}^{j} C_i h_i,$$

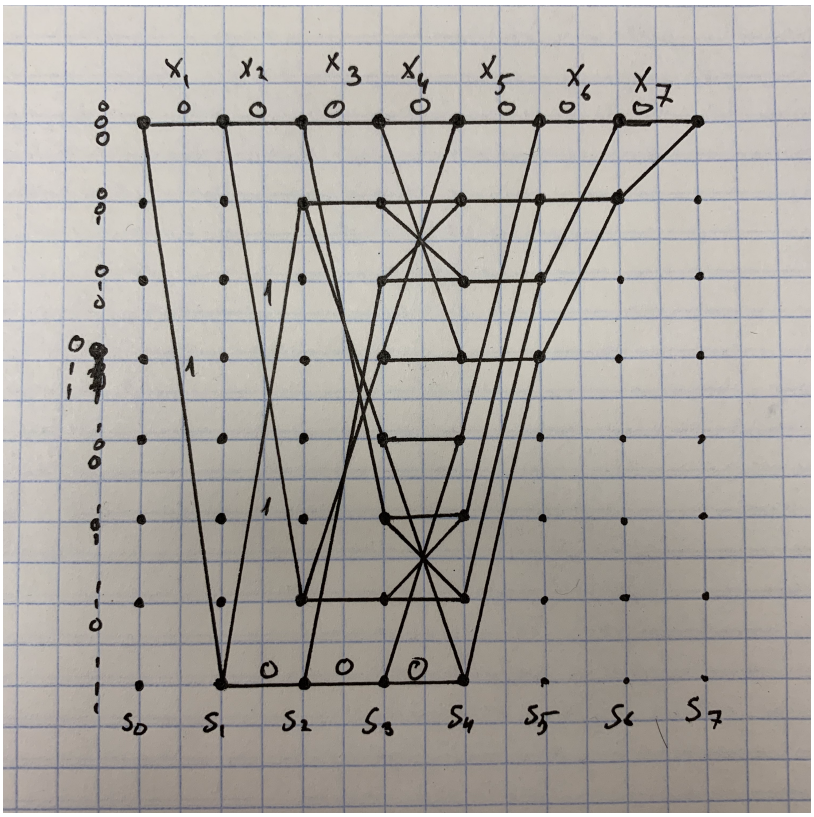
$$\lambda(e_k) = C_k$$

and

$$init(e) = 0 \ \forall e \in E_{0,1}$$

$$fin(e) = 0 \ \forall e \in E_{n-1,n}$$

where $E_{i,i+1}$ denote the set of edges connecting vertices at depth i to vertices at depth $i+1$.
c) Trellis for [7,4] Hamming code, represented in figure below:



8 Problem

8.1 Solution

Represented in **problem8.ipynb**