Solutions to Exercise Sheet 6

NOTE: There were errors in the paper sheet distributed in class. These solutions concern the corrected version, which can be found online.

Exercise 1. First note that the codewords of a linear code C form a closed linear subspace, hence the sum (modulo 2) of two codewords of a binary linear code is also a codeword of the code:

$$\forall i, j, \exists l : \mathbf{w}_i + \mathbf{w}_j = \mathbf{w}_l \ \in C. \tag{1}$$

The minimum Hamming distance is by definition

$$d = \min_{i,j,i\neq j} d(\mathbf{w}_i, \mathbf{w}_j), \tag{2}$$

where $d(\mathbf{a}, \mathbf{b})$ is the Hamming distance between $\mathbf{a}$ and $\mathbf{b}$. By using the linearity (1) and the definition of the Hamming distance, we can rewrite (2) as

$$d = \min_{i,j,i\neq j} d(\mathbf{w}_i, \mathbf{w}_j) = \min_{i,j,i\neq j} d(\mathbf{w}_i - \mathbf{w}_j, 0) = \min_{k,\mathbf{w}_k\neq 0} d(\mathbf{w}_k, 0). \tag{3}$$

Hence the minimum Hamming distance of the code is equal to the minimum Hamming weight of all nonzero codewords (also called the *minimum weight* of the code). This completes the proof. □

Exercise 2. Suppose that $\mathbf{w}$ is a codeword. Then, from the definition of the parity check matrix H, we have

$$H\mathbf{w} = 0. \tag{4}$$

We can rewrite the matrix product (4) as

$$\sum_i \mathbf{H}_i w_i = 0, \tag{5}$$

where $\mathbf{H}_i$ denotes the i-th column of H and w_i is the i-th entry of $\mathbf{w}$. Assume now that m of the columns of H are linearly dependent (i.e., the sum of these columns equals the zero vector). Then, there must exist a codeword $\mathbf{w}$ whose only nonzero entries correspond to these linearly dependent columns (since such a vector $\mathbf{w}$ would satisfy (4)). This means that if d is the minimum weight of the code (or equivalently, the minimum Hamming distance), the minimum number of linearly dependent columns must be d.
The code can correct $e-1$ errors if and only if $d \geq 2(e-1)+1 = 2e-1$. Since it is given that $e-1$ is the maximal number of correctable errors, d can only take values $2e-1$ or $2e$ (if it could take value $2e+1$, the code would be able to correct e errors). But it is also given that the code can detect (without correcting) $2e-1$ errors. The latter means that $d \geq 2e$, which implies that the distance of the code must be $2e$. As shown above, this is equivalent to the condition that the minimal number of linearly dependent columns of the parity-check matrix is $2e$. □

Exercise 3.

(a) The size of the matrix is given by $n = 6$ and $m = 4$. n corresponds to the size of the codewords. The rank of H (number of linearly independent rows or columns) is equal to $m = 4$ and corresponds to the number of parity bits. Consider a codeword vector $\mathbf{w}$ of components w_i, $i = 1, 2, ..., n$. The condition $H\mathbf{w} = 0$ can be written in terms of the system of equations:

$$\begin{cases} w_1 + w_5 + w_6 = 0 \\ w_1 + w_2 + w_6 = 0 \\ w_2 + w_3 + w_6 = 0 \\ w_1 + w_4 + w_6 = 0. \end{cases}$$

Expressing w_1 from the first equation and inserting it into the second and the fourth ones, we obtain

$$\begin{cases} w_1 = w_5 + w_6 \\ w_5 = w_2 \\ w_2 + w_3 + w_6 = 0 \\ w_5 = w_4. \end{cases}$$

Expressing w_2 from second and first equations and inserting it into the third one, obtain

$$\begin{cases} w_1 = w_5 + w_6 \\ w_5 = w_2 \\ w_1 = w_3 \\ w_5 = w_4. \end{cases}$$

We have 4 equations and 6 unknown variables. Therefore, we can choose arbitrarily the values of two of the variables and obtain a solution. By choosing all four possible combinations for the pair of $\{w_1, w_2\}$ and inserting them into the last system of equations, we find the following solutions:

$$\mathbf{w} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

It is easy to see that the minimum distance between the codewords is 3. Therefore, this code can correct one error.

(b) To correct 1 error and detect 3, the minimum Hamming distance has to be $d = 4$. This means that the minimum number of linearly dependent columns of H is 4 (see Exercise 2) and all sets of 3 columns must be linearly independent. In particular, the last column of H can neither be equal to another column of H nor to a linear combination of any two columns. There are 5 other columns and the number of possible different linear combinations of two of them is $\frac{5\cdot(5-1)}{2} = 10$. Remember that the column with all zeros is also forbidden, so this gives us $5 + 10 + 1 = 16$ different forbidden columns. Because the entries $h_{i,6}$ are bits, only $2^4 = 16$ different combinations are possible, and they are all excluded by the argument above. Therefore the Hamming distance d can not be 4.

Exercise 4.

(a) The parity check matrix of an (n, k) code is a $(n-k) \times n$ matrix, where n is the length of the code, k is the number of information (or logical) bits, and $n-k$ is the number of parity bits. In this case, the parity check matrix will have 2 rows and 5 columns. By definition, H is the matrix whose kernel (or null space) is the code space, i.e., $H\mathbf{w} = 0$ for all vectors $\mathbf{w}$ from the code space and only for vectors from the code space. In principle, if the generator matrix is given in the standard form $G = [I_k|P]$, where I_k is the identity matrix on the k information bits and P is some matrix, then the parity check matrix can be taken to be $H = [-P^T|I_{n-k}]$, where P^T is the transpose of P. In this example, the generator matrix G_1 can be seen to be already in the standard form if we regard the first, third, and fourth bits as the information bits (indeed, the first, third, and fourth columns form the submatrix I_3). With this relabeling of the bits in mind, the above standard construction yields

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Alternatively, we could treat the first three bits as the information bits (since the first three columns of G_1 are linearly independent and hence the rank of the submatrix they form is 3). We can then try to find a parity check matrix of the standard form:

$$H = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & 1 & 0 \\ h_{2,1} & h_{2,2} & h_{2,3} & 0 & 1 \end{pmatrix}$$

Multiplying each row in H by the thee codewords from G_1, we get three equations for the three coefficients of each row of H, which allow us to find:

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

This is an alternative (but equivalent) form of the parity check matrix for this code.

By calculating all combinations of the three codes words from G_1 we get $\mathbf{w}_4 = \mathbf{w}_1 + \mathbf{w}_2$, $\mathbf{w}_5 = \mathbf{w}_1 + \mathbf{w}_3$, $\mathbf{w}_6 = \mathbf{w}_2 + \mathbf{w}_3$, $\mathbf{w}_7 = \mathbf{w}_1 + \mathbf{w}_2 + \mathbf{w}_3$ and by adding $\mathbf{w}_8 = (0,0,0,0,0)$ we obtain all the codewords (note that here indexes enumerate the code words and not the elements of a codeword). We can check that all codewords have at least two bits and therefore, the minimum distance of the code is $d = 2$. Hence, this code detects single errors without correcting them.

(b) In this case, $n = 4$ and $k = 1$. The number of codewords is $2^k = 2$ and the number of parity bits is $m = n - k = 3$. Therefore we can write 3 (last) columns of H as the identity matrix while keeping the first column unknown. By inserting the codeword $\mathbf{w} = (1,1,1,1)$ into the equation $H\mathbf{w} = 0$, we easily find

$$H = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix},$$

which could also be obtained directly via the standard construction described above. The code has only two codewords:

$$\mathbf{w} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

The Hamming distance is $d = 4$. This is a repetition code which corrects single errors and detects two errors.

Remark: The transmission rate is given by $R = k/n$. We observe that $R_{G_1} = 3/5$, but the code cannot correct any error (it can only detect single errors). In contrast, $R_{G_2} = 1/4$, but the code can correct single errors and detect double errors. There is a trade-off between the transmission rate and the ability of the code to correct errors.