CS3236: Solutions to Tutorial 6
(Practical Channel Codes)

1. **[Simple Linear Code]**

   An error-correcting code for the binary symmetric channel with noise $\delta = 0.18$ is given below as a generator matrix $\mathbf{G}$:

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

   (a) How many source bits and parity-check bits are there? Express the parity-check bits as a function of the message bits.

   Solution. *There are 5 source bits and 4 parity-check bits.*

   Denoting the bits by $x_1, \ldots, x_9$, the parity check bits are $x_6 = x_1 \oplus x_2 \oplus x_3$, $x_7 = x_1 \oplus x_3 \oplus x_4$, $x_8 = x_2 \oplus x_3 \oplus x_5$, and $x_9 = x_3 \oplus x_4 \oplus x_5$.

   (b) How many possible valid codewords are there?

   Solution. *With 5 message bits, there are $2^5 = 32$ codewords.*

   (c) Give the parity-check matrix $\mathbf{H}$.

   Solution. *We have*

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

   (d) Suppose that some bits of $\mathbf{x}$ are flipped to produce a noisy version $\mathbf{y}$, i.e., $\mathbf{y} = \mathbf{x} \oplus \mathbf{z}$. Give the decoding algorithm as pairs of (syndrome – procedure) that corrects up to one error.

   Solution. *If only the i-th bit is corrupted during transmission, then the syndrome will equal the i-th row of $\mathbf{H}$. Therefore, the decoding rule can be copied straight from the rows of $\mathbf{H}$:*

Syndrome $\mathbf{S} = \mathbf{yH}$	*Unflip this bit*
0000	*none*
1100	y_1
1010	y_2
1111	y_3
0101	y_4
0011	y_5
1000	y_6
0100	y_7
0010	y_8
0001	y_9
others	*Arbitrary*

(e) What is the rate R of this code ?
Solution. $R = 5/9$.

(f) Write down an expression for the probability that at most 1 bit flip occurs in the transmitted codeword, and compute its value when $\delta = 0.18$. How does this relate to the probability of decoding error?
Solution. *The probability of at most* 1 *bit flip is* $((1-\delta)^9 + 9f(1-\delta)^8) \approx 0.5$.
Since any single bit flip (or no bit flips) gives a unique syndrome (see part (c)), this code can correct one flip. We conclude that the probability of successful decoding is at least ≈ 0.5. *The actual success probability could be higher depending on what is done when a syndrome not in the above table is encountered.*
Note also that while 0.5 *is not a very good success probability, we get something much more reasonable for lower values of* δ *(e.g.,* $\delta = 0.05$ *gives at least a probability* 0.928 *of successful decoding).*

2. **[Two Errors?]**
A Hamming code with $k = 4$ and $n = 7$ can correct up to one bit flip. Explain why a code with $k = 8$ and $n = 14$ could not possibly be guaranteed to correct up to 2 bit flips.
(Hint: Every error sequence $\mathbf{z}$ *of weight at most* 2 *would need to give a unique syndrome.)*
Solution. *If there are* n *transmitted bits, then the number of possible error patterns of weight up to two is*
$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2}$$
For $n = 14$, *this is* $1 + 14 + 91 = 106$ *patterns.*
Now, every distinguishable error pattern must give rise to a distinct syndrome, and the syndrome is $n - k$ *bits, so the maximum possible number of syndromes is* 2^{n-k}. *With* $n = 14$ *and* $k = 8$, *this gives* $2^6 = 64$ *possible syndromes.*
The number of possible error patterns of weight up to two, 106, *is bigger than the number of syndromes,* 64, *so guaranteed correction of up to* 2 *bit flips is impossible.*

3. **[Hamming Code of Order** r**]**
The Hamming code of order r is a linear code with $n = 2^r - 1$ code bits and $k = 2^r - r - 1$ message bits, given by the generator matrix
$$\mathbf{G} = [\mathbf{I}_k \;\; \mathbf{B}],$$
where $\mathbf{B}$ is a $k \times r$ matrix whose rows are all the possible row vectors of length r with entries 0 and 1, and at least two 1's (the number of such vectors is equal to $2^r - r - 1$). The order in which these row vectors appear in $\mathbf{B}$ does not matter.

(a) Write the code matrix for a Hamming code of order 3.

Solution. *Since $k = r$, $n = 7$ and $k = 4$, we just recover the same generator matrix as the lecture:*

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

(b) Show that the Hamming code of order k has minimum weight 3, and can therefore correct one error.

(Hint: Recall that the codewords are linear combinations of the rows of $\mathbf{G}$. Try analyzing three cases separately: (i) single rows, (ii) sums of 3 or more rows, and (iii) sums of exactly 2 rows.)

Solution. *The codewords are the linear combination of rows of generator matrix $\mathbf{G} = [\mathbf{I}_k \quad \mathbf{B}]$. Notice that the minimum weight of each row of $\mathbf{G}$ is at least 3, since by construction they must have one 1 in first k bits and two $1's$ in next r bits.*

Second, a codeword formed by linear combinations of any 3 or more rows in code matrix must have minimum weight of 3 or more, since the formed codeword must have 3 or more $1's$ in first k bits.

Now consider the codewords formed by a linear combination of any 2 rows in code matrix. They must have minimum weight of 3, since they must have two $1's$ in first k bits and they must have at least one 1 in next r bits (since those last r bits are different in every row).

Hence all the non-zero codewords must have a minimum weight of 3, so the Hamming code of order r can correct one error.

4. **[Extended Hamming Code]**

Take the Hamming code $\mathcal{C}$ from the lecture, with generator matrix

$$\mathbf{G}_{\text{Hamming}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

and append an overall parity check to the codewords of that code. That is, each codeword of a Hamming code is extended by 1 bit which is 0 if the codeword contains an even number of 1's and 1 if the codeword contains an odd number of 1's.

For example, the codeword 0000000 becomes 00000000, and the codeword 1110000 becomes 11100001.

(a) Write down the generator matrix of the new code

Solution. *Intuitively, the matrix should be*

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

by taking $\mathbf{G}_{\text{Hamming}}$ and adding a column that makes all of the rows sum to an even number.

To see this more formally, note that the code construction amounts to computing $\mathbf{u}' = \mathbf{u}\mathbf{G}_{\text{Hamming}}$ and then $\mathbf{x} = \mathbf{u}'[\mathbf{I}_7 \quad \mathbf{1}_7]$, where $\mathbf{I}_7$ is the 7×7 identity matrix and $\mathbf{1}_7$ is the 7×1 vector of ones. Combining these equations gives

$$\mathbf{x} = \mathbf{u}\mathbf{G}_{\text{Hamming}}[\mathbf{I}_7 \quad \mathbf{1}_7]$$

which leads to the above 4×8 matrix $\mathbf{G} = \mathbf{G}_{\text{Hamming}}[\mathbf{I}_7 \quad \mathbf{1}_7]$.

(b) Show that the new code has minimum distance 4. *(Hint: You may use the fact that the Hamming code has minimum distance 3. Try looking at different cases of $d_{\mathrm{H}}(\mathbf{y}, \mathbf{y}')$ and whether or not $p = p'$.)*

Solution. *Let $\mathbf{x}$, $\mathbf{x}'$ be two different codewords, and let $\mathbf{y}$, $\mathbf{y}'$ be the vectors with the last bit removed (i.e., the Hamming codeword) and p, p' be the final bits (i.e., the extra parity check).*

Since $\mathbf{x}$, $\mathbf{x}'$ are two different codewords of the new code, $\mathbf{y}$, $\mathbf{y}'$ are also two different codewords of the Hamming code, so $d_{\mathrm{H}}(\mathbf{y}, \mathbf{y}') \geq 3$.

Now if $d_{\mathrm{H}}(\mathbf{y}, \mathbf{y}') = 3$, then $\mathbf{y}$, $\mathbf{y}'$ must have different parity, since if they both had an even (or both odd) number of 1's, they would have differed in an even number of places, and $d_{\mathrm{H}}(\mathbf{y}, \mathbf{y}')$ would have been an even number. Thus, if $d_{\mathrm{H}}(\mathbf{y}, \mathbf{y}') = 3$ then $p \neq p'$ and we have $d_{\mathrm{H}}(\mathbf{x}, \mathbf{x}') = 4$.

The case $d_{\mathrm{H}}(\mathbf{y}, \mathbf{y}') \geq 4$ trivially gives $d_{\mathrm{H}}(\mathbf{x}, \mathbf{x}') \geq 4$ (the extra bit can never decrease distance), so overall the minimum distance must be 4.

5. **[From Non-Systematic to Systematic]**

Recall that a linear code is said to be *systematic* if its generator matrix takes the form

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & \dots & 0 & g_{1,k+1} & \cdots & g_{1,n} \\ 0 & 1 & \dots & 0 & g_{2,k+1} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & g_{k,k+1} & \cdots & g_{k,n} \end{bmatrix}$$

with an identity matrix on the left. In other words, the first k codeword bits are exactly the message bits.

In this question, we will explore how to convert a non-systematic code into an equivalent systematic one (equivalent being in the sense of how many codewords lie at distance 1, distance 2, etc. all the way up to distance n). The idea is to apply the following two operations that do not change distance properties:

- Swapping two rows of $\mathbf{G}$ just amounts to re-labeling which message bit is which.
- Letting $\mathbf{g}_i$ be the i-th row of $\mathbf{G}$, replacing a given row $\mathbf{g}_i$ by $\widetilde{\mathbf{g}}_i = \mathbf{g}_i \oplus \mathbf{g}_j$ for $j \neq i$ does not change the overall set of valid codewords. To see this, consider the linear combinations of the two rows (indexed by i and j) that can be obtained for the two codes:
  - If $u_i = 0$ then we have $(u_i \widetilde{\mathbf{g}}_i) \oplus (u_j \mathbf{g}_j) = u_j \mathbf{g}_j$ and $(u_i \mathbf{g}_i) \oplus (u_j \mathbf{g}_j) = u_j \mathbf{g}_j$, so the two codes produce the same codeword.
  - If $u_i = 1$, we have

$$\begin{aligned} (u_i \widetilde{\mathbf{g}}_i) \oplus (u_j \mathbf{g}_j) &= \widetilde{\mathbf{g}}_i \oplus (u_j \mathbf{g}_j) && \text{(since } u_i = 1\text{)} \\ &= \mathbf{g}_i \oplus \mathbf{g}_j \oplus (u_j \mathbf{g}_j) && \text{(definition of } \tilde{g}_i\text{)} \\ &= \mathbf{g}_i \oplus ((u_j \oplus 1) \mathbf{g}_j) && \text{(by linearity)} \end{aligned}$$

    so whatever was produced by $u_j = 1$ in the old code is produced by $u_j = 0$ in the new code, and vice versa. Therefore, both codes produce the same set of codewords.

(a) Use the above operations to convert the non-systematic code

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

into an equivalent systematic one.

(Hint: Try to get the first row starting with 1*, then the second row starting with* 01*, and the third with* 001*. Then work backwards and try to get the second row starting with* 010*, and the first with* 100*. This is is a form of Gaussian elimination (row reduction), but done with modulo-*2 *arithmetic.)*

Solution. *First, swap the first and second rows to get*

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

The second row already starts with 01*, so we move straight to the third row. Replace (row 3) by (row 3 + row 2) to get*

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

The second row already starts with 010*, so move back to the first row. Replacing (row 1) by (row 1 + row 2 + row 3) (which can be viewed as first doing* $1 \to 1+2$ *and then* $1 \to 1+3$*), we get*

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

(b) List the codewords for the original non-systematic code and the systematic code you found in part (a), and verify that the two codes have just as many codewords of each weight.

Solution. *The codewords are as follows:*

Message Bits	Codeword (Non-Systematic)	Codeword (Systematic)
000	000000 (weight 0)	000000 (weight 0)
001	011001 (weight 3)	001111 (weight 4)
010	111010 (weight 4)	010110 (weight 3)
011	100011 (weight 3)	011001 (weight 3)
100	010110 (weight 3)	100011 (weight 3)
101	001111 (weight 4)	101100 (weight 3)
110	101100 (weight 3)	110101 (weight 4)
111	110101 (weight 4)	111010 (weight 4)

where each codeword is computed via $\mathbf{x} = \mathbf{uG}$ *as usual. In both cases, we have* 1 *codeword of weight* 0*,* 4 *codewords of weight* 3*, and* 3 *codewords of weight* 4*.*

6. **(Optional) [Distance Bounds]**

Here we look at the existence (or non-existence) of binary codes with a given *minimum distance* (i.e., smallest Hamming distance between any two (different) valid codewords).

(a) Show that any binary block code of block length n and minimum distance d has *at most*

$$\frac{2^n}{\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i}}$$

codewords. This is called the *sphere packing* bound.

(Hint: The clue is in the name! Note that with minimum distance d*, the "spheres of radius* $\lfloor (d-1)/2 \rfloor$*" (i.e., the set of all binary codewords within that distance) centered at each codeword must be non-overlapping. How many sequences lie in each sphere?)*

Solution. *As discussed in the lecture, since the code has minimum distance d, it is guaranteed to correct $t = \lfloor (d-1)/2 \rfloor$ errors. In other words, the spheres of radius t around each codeword are disjoint. Since each such sphere contains $\sum_{i=0}^{t} \binom{n}{i}$ sequences (namely, the codeword itself, $\binom{n}{1}$ with one bit flipped, $\binom{n}{2}$ with two bits flipped, etc.), and the total number of binary sequences is 2^n, the bound follows.*

(b) Show that given integers n and d, there exists a code with block length n, minimum distance $d_{\min} \geq d$, and *at least*

$$\frac{2^n}{\sum_{i=0}^{d-1} \binom{n}{i}}$$

codewords. This is known as the *Gilbert-Varshamov Bound*. Notice that the summation goes to $d-1$, rather than $\lfloor (d-1)/2 \rfloor$ in part (a).

(Hint: Ok, no clue in this name. Try to design an iterative procedure that keeps a list of "feasible codewords". Initially everything is feasible, but after a codeword is selected, everything within distance $d-1$ is marked as infeasible. How many iterations can we do before having no more feasible codewords?)

Solution. *We use the procedure in the hint. We did not specify exactly which feasible codeword to pick, but in fact making an arbitrary feasible choice will suffice!*

First note that since selecting a codeword rules out (i.e., marks as infeasible) all of the codewords within distance $d-1$, we do indeed have a minimum distance of at least d, as required. So we only need to show that we are guaranteed to perform $\frac{2^n}{\sum_{i=0}^{d-1} \binom{n}{i}}$ iterations before there are no more feasible codewords to choose.

Such a claim is straightforward: The number of codewords within distance $d-1$ of a given codeword is $\sum_{i=0}^{d-1} \binom{n}{i}$, so each iteration, at most this many codewords are ruled out (and sometimes fewer, if some of those codewords had already been ruled out on an earlier iteration). Since there are 2^n possible codewords in total, it must take at least $\frac{2^n}{\sum_{i=0}^{d-1} \binom{n}{i}}$ iterations before all codewords are ruled out, thereby proving the claim.

7. **(Optional) [Singleton Bound / Maximum Distance Separable Codes]**

While we have focused on binary codes with codewords $\mathbf{x} = (x_1, \ldots, x_n)$ such that $x_i \in \mathcal{X} = \{0, 1\}$, there are also very powerful codes defined on larger alphabets. Here we consider q-ary codes, with $\mathcal{X} = \{0, 1, \ldots, q-1\}$. The Hamming distance is still defined as the number of differing symbols: $d_{\mathrm{H}}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{n} \mathbf{1}\{x_i \neq x_i'\}$.

We consider (possibly non-linear) codes whose number of codewords is $M = q^k$ for some integer $k < n$. We write $n = k + t$, so that t represents how much higher n is than k. (Note that since $M = q^k$, we can view the code as mapping length-k q-ary sequences to length-k q-ary codewords.)

Show that in this setup, any code must have minimum distance at most $d_{\min} \leq t + 1$.

(Hint: The argument is actually very simple. Take the first k symbols of each codeword, and consider two possible cases: (i) There exist two codewords giving the same first k symbols; (ii) All of the $M = q^k$ codewords have a distinct sequence of first k symbols.)

This result is known as the *singleton bound*, and codes achieving $d_{\min} = t + 1$ are said to be *maximum distance separable* (MDS). This is not such a useful concept for binary codes ($q = 2$), because the only MDS codes in that case are trivial. But for higher q, things get more interesting. A famous class of (non-binary linear) codes called *Reed-Solomon codes* are MDS whenever q is a prime power with $q \geq n$.

Solution.

Follow the hint. In case (i), since there are two codewords with the same first k symbols, they can only differ in the remaining t symbols, so the minimum distance is at most t.

In case (ii), no two codewords have the same first k symbols. But each symbol can only take one of q values, and there are only $M = k^q$ codewords by assumption. This means that every possible length-k sequence must occur (otherwise two would be the same). As a result, although no two codewords have the same first k symbols, we can certainly find two codewords agreeing in $k-1$ out of the first k positions. By a similar argument to case (i), this means that the minimum distance is at most $t+1$.