Coding Theory and Applications

Solved Exercises and Problems of Linear Codes

Enes Pasalic
University of Primorska

Koper, 2013

Contents

1 Preface

This is a collection of solved exercises and problems of linear codes for students who have a working knowledge of coding theory. Its aim is to achieve a balance among the computational skills, theory, and applications of cyclic codes, while keeping the level suitable for beginning students. The contents are arranged to permit enough flexibility to allow the presentation of a traditional introduction to the subject, or to allow a more applied course.

Enes Pasalic
enes.pasalic@upr.si

2 Problems

In these exercises we consider some basic concepts of coding theory, that is we introduce the redundancy in the information bits and consider the possible improvements in terms of improved error probability of correct decoding.

1. (Error probability): Consider a code of length six $n = 6$ defined as,

$$(a_1, a_2, a_3, a_2 + a_3, a_1 + a_3, a_1 + a_2)$$

where $a_i \in \{0, 1\}$. Here a_1, a_2, a_3 are information bits and the remaining bits are redundancy (parity) bits. Compute the probability that the decoder makes an incorrect decision if the bit error probability is $p = 0.001$. The decoder computes the following entities

$$\begin{aligned} b_1 + b_3 + b_4 &= s_1 \\ b_1 + b_3 + b_5 &== s_2 \\ b_1 + b_2 + b_6 &= s_3 \end{aligned}$$

where $\mathbf{b} = (b_1, b_2, \ldots, b_6)$ is a received vector.

We represent the error vector as $\mathbf{e} = (e_1, e_2, \ldots, e_6)$ and clearly if a vector $(a_1, a_2, a_3, a_2 + a_3, a_1 + a_3, a_1 + a_2)$ was transmitted then $\mathbf{b} = \mathbf{a} + \mathbf{e}$, where '+' denotes bitwise mod 2 addition.
The reason the decoder choose the above equations is the following,

$$\begin{aligned} b_2 + b_3 + b_4 &= a_2 + e_2 + a_3 + e_3 + a_2 + a_3 + e_4 = e_2 + e_3 + e_4 = s_1 \\ b_1 + b_3 + b_5 &== \ldots = e_1 + e_3 + e_5 = s_2 \\ b_1 + b_2 + b_6 &= e_1 + e_2 + e_6 = s_3 \end{aligned}$$

Therefore, based on the knowledge of **b** the receiver computes the linear combinations of error bits. Given s_1, s_2, s_3 the decoder must choose the most likely error pattern **e** which satisfies the three equations. Notice that there is a unique solution for **e** for any $(s_1, s_2, s_3) \neq (1, 1, 1)$.
E.g. assume that $s = (0, 0, 1)$ then obviously $(e_1, \ldots, e_6) = (0, 0, \ldots, 1)$ satisfies the above equations. Can there be some other **e** of weight one to get $s = (0, 0, 1)$?
Let us try $(e_1, \ldots, e_6) = (1, 0, \ldots, 0)$. Then $s_2 = 1$ etc. Therefore, an error pattern of weight 1 is decoded correctly.

Now if $(s_1, s_2, s_3) = (1, 1, 1)$ the decoder must choose one of the possibilities,

$$\begin{gathered} (1, 0, 0, 1, 0, 0) \\ (0, 1, 0, 0, 1, 0) \\ (0, 0, 1, 0, 0, 1) \end{gathered}$$

The decoder need to select one of these patterns (usually in advance).

Remark: We will later see (when we introduce standard array and syndrome decoding) that only if the coset leader that correspond to the syndrome appears to be the error pattern then we make a correct decision. Among all other error patterns there is one with two errors that is decoded correctly. Do not forget that we only consider the solutions of minimum weight for $\mathbf{e}$.

The probability of correct decoding is therefore,

$$P_C = (1-p)^6 + 6(1-p)^5 p + (1-p)^4 p^2 = 0.999986.$$

That is the decoder can correct if no error has occured in transmission (first term), can correct all single error patterns (second term), and a single error pattern of weight 2 (second term). Note that there are $\binom{6}{2}$ error patterns of weight 2 but only one can be corrected.

2. We consider the same example but we want to find out what is the probability that the information bits a_1, a_2, a_3 are correctly decoded. Above, we considered the probability that a received word was decoded correctly. The probability that a symbol in the information sequence is decoded incorrectly is called symbol error probability (this was p without coding).

   This computation is quite demanding (even for a small code) and we only give a sketch of the procedure.
   There are 64 error patterns for a code of length 6 (binary). We know that 8 of these lead to all 3 correct information symbols after decoding. The patterns are,

$$\begin{aligned} (0,0,0,0,0,0) \\ wt(\mathbf{e}) = 1 \\ \text{one pattern of weight } 2 \end{aligned}$$

   What about the rest ? Firstly, one can find out there are only 4 essentially different 3-tuples (s_1, s_2, s_3) (all of which would lead to a similar computation) . Nevertheless, one may examine all the 7 nonzero syndromes and compute the probability that at least one information symbol is incorrect.

   As an example of calculation let us consider $(s_1, s_2, s_3) = (1, 1, 0)$. The decoder equations gives the solution space for $\mathbf{e}$ to be:

$$\begin{aligned} (101011), (011101) \\ (110000), (010011) \\ (100101), (000110) \\ (111110), (\mathbf{001000}) \end{aligned}$$

The bold face entry is the most likely pattern and the receiver takes the decision that $e_3 = 1$ and $e_i = 0$ for $i \neq 3$. But what if there was not one error during the transmission. Then,

$$Pb(\text{two correct information symbols } = p^2(1-p)^4 + 2p^4(1-p)^2.$$

This corresponds to the vectors (000110) and $(101011), (011101)$ respectively. The first one occurs with probability $p^2(1-p)^4$. If the actual error was (000110) and we used **(001000)** in the decoding then the third information bit a_3 will be incorrect. The same applies for $(101011), (011101)$ where the correction agrees with actual errors in the third position so the first and third information bit is incorrect respectively.

We can also obtain two bit errors if the actual error vectors are (111110), (010011), (100101). This occurs with probability :

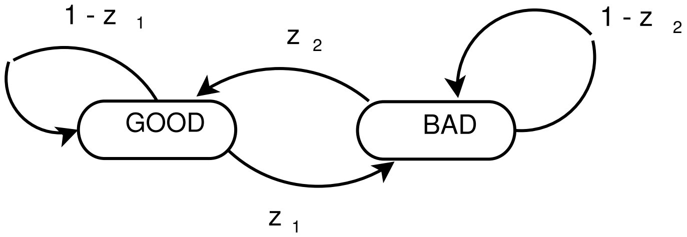
$$Pb(\text{one correct information symbols } = p^5(1-p) + 2p^3(1-p)^3.$$

Finally, all the information symbols might be incorrect which occurs if we correct **r** using **e**=(001000) and the actual error vector was (110000). This occurs with probability $P_b = p^3(1-p)^3$. One can proceed in this way by examining the other possibilities for (s_1, s_2, s_3) and calculating corresponding probabilities. After some tedious counting one arrive at something like

$$P_S = \frac{1}{3}(22p^2(1-p)^4 + 36p^3(1-p)^3 + 24p^4(1-p)^2 + 12p^5(1-p) + 2p^6).$$

For $p = 0.001$ this gives $P_s = 0.000007$!

3. (Gilbert-Eliot channel model) This exercise introduces a useful channel model in the presence of burst errors, i.e. we assume that the errors comes in a consecutive manner and the channel remains some time in that state. The channel later returns with some probability to a "good state". This is presented in the following figure.



Suppose that the transition probabilities are $z_1 = 10^{-6}$ and $z_2 = 10^{-1}$.

a) Find the average length of an error burst.

The average length of an error burst is the same as the average time of staying in the bad state. The probability of an error burst of length k is the same as the probability of staying in bad state for an interval of length k, which is equal to,

$$P(\text{error burst of length } k) = (1-z_2)^{k-1} z_2.$$

The error burst length has a geometric probability mass function, so average error burst length

$$
\begin{aligned}
&= \sum_{k\geq 1} kP(\text{error burst of length } k)\\
&= \sum_{k\geq 1} k(1-z_2)^{k-1}z_2\\
&= z_2\left[\sum_{k\geq 1}(1-z_2)^{k-1} + \sum_{k\geq 2}(1-z_2)^{k-1} + \sum_{k\geq 3}(1-z_2)^{k-1} + \cdots\right]\\
&= z_2\left[(1/z_2) + (1-z_2)/z_2 + (1-z)^2/z_2 + \cdots\right]\\
&= = 1 + (1-z_2) + (1-z_2)^2 + \cdots\\
&= \frac{1}{z_2} = 10.
\end{aligned}
$$

b)Find the average length of an error-free sequence of bits.

Similar to the last part, the average length of an error-free sequence of bits is the average time of staying in good state, which means that average error-free sequence length is

$$
\begin{aligned}
&= \sum_{k\geq 1} kP(\text{error-free sequence of length } k)\\
&= \sum_{k\geq 1} k(1-z_1)^{k-1}z_1\\
&= \frac{1}{z_1} = 10^6.
\end{aligned}
$$

c) Find the bit error rate for this Gilbert-Elliot channel.

Using the results of the last two parts, long sequences of bits have on average error bursts of length 10 and error free sequences of length 10^6 . This means that the fraction of bits in error is,

$$\frac{10}{10+10^6} = \frac{1}{100001} \approx 10^{-5}.$$

4. A (6, 3) linear block code C over GF(2) is defined by the following parity check matrix,

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

(a) Find the generator matrix of C .

(b) The parity check matrix H does not allow the presence of the codewords of weight < 3 (apart from the all zero codeword). Explain why ?

(c) Suppose that the code is used for error detection only over a binary symmetric channel with error rate $p = 10^{-3}$. Find the probability of undetected error.

Hint: W.l.o.g. assume that all zero codeword was transmitted. Be careful with the interpretation of undetected error.

5. (Code design) Suppose we have a binary channel with bit error rate 10^{-5}. Packets (codewords) transmitted over this channel will consist of 1000 bits including check bits. We want to specify codes that satisfy the reliability requirements of several scenarios. Assume that the number of check bits needed to correct t errors in the received packet is $10t$ for $t \leq 10$.

a) Suppose that no coding is used. What is the probability of receiving a packet with k bit errors ?

The number of bit errors has a binomial probability distribution. For this channel,

$$\begin{aligned} P(k \text{ errors in 1000 bits }) &= \binom{1000}{k} (10^{-5})^k (1 - 10^{-5})^{1000-k} \\ &= \frac{1000 \cdot 999 \cdots (1000 - k + 1)}{k!} \times \\ &\quad 10^{-5k} (1 - 10^{-5})^{1000-k}. \end{aligned}$$

For small k, use $(1-10^{-5})^{1000-k} \approx 0.99 \approx 1$ and $1000 \cdot 999 \cdots (1000-k+1) \approx 1000^k$. Therefore we can approximate the above probability by $(10^{-2})^k / k!$.

b) Suppose that the bit error rate after packet decoding is required to be less than 10^{-9}. In your code design, how many bit errors must be correctable in each packet in order to satisfy this requirement?

The bit error rate can be expressed as the average number of incorrect bits per packet after decoding, divided by the number of bits per packet. If a code can correct up to t errors, then when $t + 1$ errors occur the worst that can happen is that the decoder flips t bits to arrive at an incorrect codeword. Thus a conservative assumption is that $t+1$ errors in a packet result in $2t+1$ bit errors after decoding, and the contribution to the bit error rate is,

$$P(t + 1 \text{ errors in packet }) \cdot \frac{2t + 1}{1000}.$$

Using the above results and the result in a) we have,

t	Probability of t+1 errors	Bit error after decoding
0	$(10^{-2}/1! = 10^{-2}$	$10^{-2} \cdot 1/1000 = 10^{-5}$
1	$(10^{-2})^2/2! = (1/2) \cdot 10^{-4}$	$(1/2) \cdot 10^{-4} \cdot 3/1000 = 1.5 \cdot 10^{-7}$
2	$(10^{-2})^3/3! = (1/6) \cdot 10^{-6}$	$(1/6) \cdot 10^{-6} \cdot 5/1000 = 8.3 \cdot 10^{-10}$

Since the probability of ≥ 3 errors is negligible, it is sufficient to be able to correct up to two errors.

6. (Error probability): Let C be a ternary repetition code of length 4 over the alphabet $\{0,1,2\}$.

   (a) List the vectors which will be uniquely decoded as 1111 using nearest neighbour decoding.

   (b) If the probability of each symbol being wrongly received is t and each symbol is equally likely, find the word error probability; that is, the probability P_e of a word being incorrectly decoded.

   (c) What is P_e when $t = 0.05$?

   Solution: The code $C = \{0000, 1111, 2222\}$. Hence C corrects $(4-1)/2 = 1$ error. However,some words at distance 2 from a codeword can also be corrected.

   (a) The received words decoded as 1111 are as follows:
   1111;
   0111, 2111, 1011, 1211, 1101, 1121, 1110, 1112;
   0211, 2011, 0121, 2101, 0112, 2110, 1021, 1201, 1012, 1210, 1102, 1120.

   (b) First, note that
   $$\text{P(1 being received)} = 1 - t$$
   and
   $$\text{P(0 or 2 being received)} = t$$
   thus
   $$\text{P(0 being received)} = \text{P(2 being received)} = t/2.$$
   Hence, the probability of correct decoding of the word 1111 is,
   $$\begin{aligned} P_c &= (1-t)^4 + 8(t/2)(1-t)^3 + 12(t/2)^2(1-t)^2 \\ &= (1-t)^2\{(1-t)^2 + 4t(1-t) + 3t^2\} \\ &= (1-t)^2(1+2t). \end{aligned}$$
   Hence the probability of a word being incorrectly decoded is,
   $$P_e = 1 - (1-t)^2(1+2t) = t^2(3-2t).$$

   (c) When $t = 0.05$ then $P_e \approx 3t^2 = 0.0075$.

7. (Error probability II): Let C be the binary repetition code of length 5. List the vectors which will be uniquely decoded as 11111 using nearest neighbour decoding. Do parts (b) and (c) of the previous question for this code.

   Here $C = \{00000, 11111\}$. So the words decoded as 11111 and their probabilities are,
   $$\begin{array}{lll} 11111 & & (1-t)^5; \\ 01111, & (5\ \textit{like this}) & 5t(1-t)^4; \\ 00111, & (10\ \textit{like this}) & 10t^2(1-t)^3. \end{array}$$
   Therefore,
   $$\begin{aligned} P_c &= (1-t)^5 + 5t(1-t)^4 + 10t^2(1-t)^3 \\ &= \cdots \\ &= (1-t)^3(1+3t+6t^2). \end{aligned}$$

Then, $P_e = 1 - (1-t)^3(1+3t+6t^2)$ which for $t = 0.05$ gives $P_e = 0.00116$.
Moral: We get a better error probability at the price for slower rate (5 symbols sent). On the other hand, a larger alphabet is used in the first case.

8. (Undetected error probability): Consider the following two binary codes with blocklength 8.
The (8, 4) extended Hamming code with minimum distance 4.
The (8, 7) simple even parity-check code.
The weight enumerator for the extended Hamming code is $A(x) = 1+14x^4+x^8$. This states that for every codeword there are 14 codewords at distance 4 and one codeword at distance 8.

   (a) Find the weight enumerator for the (8, 7) simple even parity-check code.

   (b) Derive the undetected error probability P_{ue} of these two codes as a function of the bit error rate ϵ for $\epsilon \leq 1/2$. Assume a binary symmetric channel with independent errors. Explain the behaviour of P_{ue} for the two codes as $\epsilon \to 1/2$.

   (a) The simple parity-check code consists of all 8-tuples of even weight. The number of 8-tuples of weight w is given by the binomial coefficient,

$$\binom{8}{w} = \frac{8!}{w!(8-w)!}$$

   The weight enumerator is

$$\binom{8}{0} + \binom{8}{2}x^2 + \binom{8}{4}x^4 + \binom{8}{6}x^6 + \binom{8}{8}x^8 = 1 + 28x^2 + 70x^4 + 28x^6 + x^8.$$

   For the (8, 4) expanded Hamming code,

$$P_{ue} = 14\epsilon^4(1-\epsilon)^4 + \epsilon^8.$$

   For the (8, 7) simple parity-check code,

$$P_{ue} = 28\epsilon^2(1-\epsilon)^6 + 70\epsilon^4(1-\epsilon)^4 + 28\epsilon(1-\epsilon)^2 + \epsilon^8.$$

   (b) For the (8,4) expanded Hamming code, P_{ue} increases monotonically to the value 0.0586 as $\epsilon \to 1/2$. For small ϵ, the probability of 8 bit errors is small, so P_{ue} is dominated by the probability of flipping bits so that the received vector is one of the 14 codewords at distance 4. Thus, as $\epsilon \to 1/2$, the behavior of P_{ue} follows that of the polynomial $14\epsilon^4(1-\epsilon)^4$.

   For the (8,7) simple even parity-check code, P_{ue} is just under 1/2 as $\epsilon \to 1/2$, namely, $P_{ue} = 1/2 - 2^{-8}$.

9. (Decoded bit error rate): A (15,11) binary Hamming code is a single-error correcting code with $d = 3$. It is used on a binary symmetric channel with raw error rate $p = 10^{-4}$. Find the cooked error rate - the probability that a decoded message bit is incorrect.
Decoding Fact: When two bit errors occur, the decoder incorrectly changes a third codeword bit. You may assume that the probability of ≥ 3 errors is negligible.

Solution: For raw bit error rate $p = 10^{-4}$, the probability of two errors in a codeword is given by the binomial distribution:

$$P(2 \text{ errors }) = \binom{15}{2}(10^{-4})^2(1-10^{-4})^13 = 1.05 \times 10^{-6}$$

One can compute the probability of three or more errors which is 4.6×10^{-10}, which is negligible.

When two errors occur in a received codeword of length 15, the decoder miscorrects by changing one more bit to arrive at a codeword that differs from the transmitted codeword in three bits. The probability that any particular bit within a codeword (information bit or check bit) is incorrect is the average number of wrong bits per codeword divided by the number of bits per codeword. Therefore the cooked bit error rate is,

$$\frac{3}{15} \times P(2 \text{ errors })2.1 \times 10^{-7}.$$

For raw error rate $p < 2 \times 10^{-3}$, an accurate approximation to the cooked error rate is,

$$\frac{3}{15}\binom{15}{2}p^2 = 21p^2.$$

10. (Linear independence): Is it true that if x, y, and z are linearly independent vectors over $GF(q)$, then so are $x + y, y + z$, and $z + x$?

    Solution: The vectors are not linearly independent over fields of characteristic 2, since over $GF(2^m)$

    $$(x + y) + (y + z) + (z + x) = (x + x) + (y + y) + (z + z) = 0 + 0 + 0 = 0.$$

    On the other hand, when q is odd, x, y, and z can be expressed as linear combinations of the three vectors, which therefore span a subspace of dimension 3. For example,

    $$x = 2^{-1}((x + y) + (z + x) - (y + z)).$$

11. (Subspaces): Given that S and T are distinct two- dimensional subspaces of a three-dimensional vector space, show that the intersection of S and T is a one-dimensional subspace.

    Solution: There are two reasonable approaches. Both start from the fact that the intersection of S and T is a linear subspace whose dimension is at most 2. We must show that the dimension is not 0 or 2, and is therefore exactly 1.

    Proof 1 (Direct): $S \cap T$ cannot have dimension 2, because in this case any basis for $S \cap T$ would also span S and T, contradicting the hypothesis that S and T are distinct.

    $S \cap T$ cannot have dimension 0, because in this case the set of four vectors consisting of a basis for S and a basis for T would be linearly independent, which contradicts the hypothesis that S and T are subspaces of a 3-dimensional vector space.

Proof 2: (Using orthogonal complements) The orthogonal complements of S and T have dimensions $3-2=1$. The orthogonal complements are distinct because S and T are distinct. The linear subspace generated by the orthogonal complements has dimension exactly 2; the dimension is greater than 1 because the orthogonal complements are distinct and is at most 2 because their dimensions are each 1. But the linear combinations of the orthogonal complements of S and T form the orthogonal complement of $S \cap T$. So the dimension of $S \cap T$ is $3-2=1$.

12. (Standard form): Let C be a binary $(5,3)$ code with generator matrix,

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

(a) Reduce G to standard form.

(b) Find a parity-check matrix for C.

(c) Write out the elements of the dual code $C^{\perp}$.

(a) Reduce G to standard form.

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

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

(b) Find a parity-check matrix for C.

$$H = [A^T I_k] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

(c) Write out the elements of the dual code $C^{\perp}$. The elements of $C^{\perp}$ are linear combinations of the rows of H,

$$= C^{\perp} = \{00000, 10010, 11101, 01111\}.$$

13. Modified block codes. Some of the following operations on rows or columns of the generator matrix G or the parity-check matrix H may decrease the minimum Hamming weight of a linear block code? Which of the operations below can cause a reduction in the minimum weight.

(a) Exchanging two rows of G.

(b) Exchanging two rows of H.

(c) Exchanging two columns of G.

(d) Exchanging two columns of H.

(e) Deleting a row of G.

(f) Deleting a row of H.

(g) Deleting a column of G and the corresponding column of H.

(h) Adding a column to G and a corresponding column to H.

(i) Multiplying a column of H by a nonzero element of the channel alphabet.

(j) Adding one column of H to another column of H.

Solution

(a) Exchanging two rows of G - Same basis vectors same code. NO CHANGE.

(b) Exchanging two rows of H.- Same check equations same code. NO CHANGE

(c) Exchanging two columns of G.- Equivalent code has codewords of same weight. NO CHANGE

(d) Exchanging two columns of H.- Equivalent code has codewords of same weight.

(e) Deleting a row of G. - Reduces code dimension, **may increase** minimum weight.

(f) Deleting a row of H. - Increases code dimension, may decrease minimum weight.

(g) Deleting a column of G and the corresponding column of H. - May delete a check symbol and decrease minimum weight.

(h) Adding a column to G and a corresponding column to H.-New codewords have an additional symbol hence weight is at least large.

(i) Multiplying a column of H by a nonzero element of the channel alphabet.- Subspaces generated by columns do not change when one column is multiplied by a nonzero scalar.

(j) Adding one column of H to another column of H.- One column might be negative of another column, may decrease the weight of the code.

14. (Standard form): Let C be a binary $(5, 3)$ code with generator matrix,

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

(a) Reduce G to standard form.

(b) Find a parity-check matrix for C.

(c) Write out the elements of the dual code $C^{\perp}$.

(a) Reduce G to standard form.

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

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

(b) Find a parity-check matrix for C.

$$H = [A^T I_k] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

(c) Write out the elements of the dual code $C^\perp$. The elements of $C^\perp$ are linear combinations of the rows of H,

$$= C^\perp = \{00000, 10010, 11101, 01111\}.$$

15. (Counting codewords): Prove that it is not possible to find 32 binary words, each of length 8 bits, such that each word differs from every other word in at least 3 places.

Solution: The volume of the space of 8-bit binary words is the number of points, $2^8 = 256$. If no two codewords are within distance 3, then the spheres of radius 1.5 about any two codewords must be disjoint. The volume of a sphere is the number of points in the sphere. Within distance 1.5 of any 8-bit word are 9 words - the word itself and those words that differ from it in exactly one bit. Since $256/9 = 28.444$, there can be at most 28 disjoint spheres of radius 1.5 (same as spheres of radius 1), and so there can be at most 28 codewords all of which are distance 3 from each other.

16. (Codeword weight): The purpose of this exercise is to demonstrate how we can get the information about the codewords only based on the parity check matrix (without finding G).
Let C be the binary linear block code whose parity-check matrix H have all the parity check columns of weight 3:

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

(a) Find a codeword of minimum weight.

Solution: Note here that any codeword in the code satisfies $H\mathbf{c}^T = 0$. The rightmost 6 columns of H are the complements of the leftmost six columns. So the sum of the first and last columns is the all-ones vector, as is the sum of the second and next to last column. The sum of the first two and last two columns is zero, which corresponds to the following weight 4 codeword:

$$(1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1).$$

Another weight 4 codeword, which corresponds to a linear dependence of the first six columns of H, is

$$(0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0).$$

Easy to see there can not be the codewords of less weight (apart from $\mathbf{c}$=$\mathbf{0}$).

(b) Prove that all codewords have even parity.

Solution: All columns of H have odd parity, so the sum (exclusive-or) of an odd number of columns has odd parity and is therefore nonzero. Every codeword selects a set of columns of H whose sum is zero, so codewords must have an even number of nonzero components.

(c) Does C have a systematic parity-check matrix of the form $[I| - P^T]$?

Solution: No. A systematic parity-check matrix of the form $[I| - P^T]$ would be obtained by elementary row operations on H. But the first six columns of H have rank ≤ 5 since the last row of that submatrix is zero. Therefore it is not possible to transform H to the required form.

(d) What are the parameters of the code ?

Solution: Clearly, dimension $n = 12$, and $d = 4$. The dimension is then $n - k = 6$ which gives $k = 6$.

17. (Code rate): a) Let C_1 be the binary code of blocklength 14 consisting of all sequences in which there are at least three 0s between any two 1s. Find the rate of C_1.

Solution: The rate of a block code with alphabet size Q, blocklength n, and M codewords is $\frac{1}{n} \log_Q M$.
Let M_n be the number of binary sequences of length n that have at least three 0s between any two 1s. The first four values of M_n are given in the following table together with the codewords of length n.

n	M_n	code of blocklength n
1	2	$\{0, 1\}$
2	3	$\{00, 01, 10\}$
3	4	$\{000, 001, 010, 100\}$
4	5	$\{0000, 0001, 0010, 0100, 1000\}$

For $n \geq 5$ any valid sequence begins with either 0 or 1. If the first bit is 0, then the remaining $n - 1$ bits can be any valid sequence of length $n - 1$. If the first bit is 1, then the next three bits must be 0, followed by any valid sequence of length $n - 4$. Therefore $\{M_n\}$ satisfies the following recurrence:

$$M_n = \begin{cases} n+1 & n = 1, 2, 3, 4 \\ M_{n-1} + M_{n-4} & n \geq 5 \end{cases}$$

Using this recurrence, we can generate M_n for $n = 5, \ldots, 14$:

n	5	6	7	8	9	10	11	12	13	14
M_n	7	10	14	19	26	36	50	60	95	131

Therefore the rate of C_1 is $\frac{1}{14} \log_2 131 = 0.5024$.

18. (Systematic codes): A code C is called *systematic* on k positions (and the symbols on these positions are called information symbols) if $|C| = q^k$ and there is exactly one codeword for

every possible choice of coordinates in these k positions.
Let C be a (n, k) code over $\mathbb{F}_q$ which is systematic on any set of k positions. Show that C has minimum distance $d = n - k + 1$.

We shot that weight of any nonzero codeword in C is $> n - k$. On contrary, assume $wt(c) \leq n - k$. Then there exists a subset of k coordinates for which $c_{i_1} = c_{i_2} = \ldots = c_{i_k}$. Since C is systematic in these positions as well we must have that $\mathbf{c}$ =0. Simply note that we may write $\mathbf{c}$ as,

$$\mathbf{c} = \alpha \mathbf{c}_1 + \beta \mathbf{c}_2 + \gamma \mathbf{c}_3; \ \alpha, \beta, \gamma \in \mathbb{F}_q.$$

The fact that $\mathbf{c}$ is zero on k systematic positions implies that $\alpha = \beta = \gamma = 0$. Hence, $d > n-k$. On the other hand $d \leq n-k+1$ since given any k positions there will be codewords of weight 1 on these k positions, implying $d \leq n - k + 1$. Thus, C is a $(n, k, n - k + 1)$ code, also known as MDS (Maximum Distance Separable) code.

19. (Codes over $\mathbb{Z}_3$) The idea is to examine the minimum distance through the parity check matrix over the alphabet $A = \{0, 1, 2\}$. Show that the following parity check matrix,

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & 2 & 2 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & 0 \end{bmatrix}$$

generates a code with minimum distance $d = 4$.

Solution Using the result (Theorem 3.5) in the textbook we have to show that any 3 columns of H are linearly independent. Permuting the columns of H we get an equivalent code with the same minimum distance,

$$H' = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 2 & 2 \\ 1 & 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

Then it is clear that you cannot find 3 or less linearly dependent columns of H. The code is (6,2,4) over GF(3) and we essentially have no improvement over binary alphabet, that is there is a (6,2,4) code over GF(2).
Can we find (6,3,4) code over GF(3) ? No, try to remove one row of H and to preserve independency of 4 vectors for any entries of H. Not possible.

20. (Generator matrices and error correction): Let C be a binary (6,3) code with the parity check matrix,

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

(a) Find coset leaders and their syndromes.

(b) Use syndrome decoding to decode the following received vectors: (i) 110000; (ii) 000011.

(a) The code C has 8 elements and so 8 cosets in $V(6, 2)$.
Coset leaders are :

000000 100000 010000 001000 000100 000010 000001 100001

The corresponding syndromes are:

000 101 110 010 111 011 001 100

These are computed as $\mathbf{s} = Hl^T$ where l is a coset leader. E.g. the first coset leader $l = (100000)$ selects the first column of H in the multiplication Hl^T, i.e. $s = 101$

(b) Use syndrome decoding to decode the following received vectors:
(i) $\mathbf{r} = 110000 \rightarrow \mathbf{s} = 011 \rightarrow \mathbf{l} = 000010 \rightarrow \mathbf{c} = 110010$;
(ii) $\mathbf{r} = 000011 \rightarrow \mathbf{s} = 010 \rightarrow \mathbf{l} = 001000 \rightarrow \mathbf{c} = 001011$

21. (Binary Hamming code & bound):

(a) What is the rate of the Hamming code of block length $2^\ell - 1$?

By definition of the Hamming code the message length of the code is $2^\ell - \ell - 1$ and so the rate is $1 - \frac{\ell}{2^\ell - 1}$.

(b) Show that if C is a t-error-correcting code in $\{0, 1\}^n$, then $|C| \leq 2^n / \operatorname{Vol}(n, t)$, where $\operatorname{Vol}(n, t) = \sum_{i=0}^{t} \binom{n}{i}$.

By definition, the Hamming balls of radius t around codewords are non-intersecting in a t-error correcting code. Since each such Hamming ball is a subset of $\{0, 1\}^n$ we get that the sum of their volumes is at most 2^n. Each has volume equal to $\operatorname{Vol}(n, t)$ and this gives the bound.

(c) Conclude that the Hamming codes of Part (a) are optimal in their performance.

Hamming codes are 1-error correcting codes. The bound of Part (b) implies such codes may have at most $2^n/(n + 1)$ codewords. Part (a) shows that Hamming codes do achieve this bound exactly when $n = 2^\ell - 1$. (The number of codewords is $2^{2^\ell - \ell - 1} = 2^{n-\ell} = 2^n/2^\ell = 2^n/(n + 1)$.)

22. (Perfect codes I): Show that the binary repetition code of length n, with n odd, is perfect. How many errors does it correct?

Let $C = \{a_0 = 000\ldots0, a_1 = 111\ldots1\}$ be of length n. Any vector $x \in \mathbb{F}_2^n$ has t coordinates 1, and $n - t$ coordinates 0. So $d(x, a_0) = t$ and $d(x, a_1) = n - t$. Hence, if $t < n/2$, then x is uniquely decoded as a_0, whereas, if $t > n/2$, then x is uniquely decoded as a_1. So C is perfect and corrects $\lfloor n/2 \rfloor = (n - 1)/2$ errors. This can also be done using the Sphere Packing Bound.

23. Let C and D be linear codes over $\mathbb{F}_q$ of the same length. Define:

$$C + D = \{c + d | c \in C, d \in D\}.$$

Show that $C + D$ is a linear code and that $(C + D)^\perp = C^\perp \cap D^\perp$.

First we need to prove that $C + D$ is a linear code. The proof is elementary. Since $\mathbf{0} \in C, D$ then clearly $\mathbf{0} \in C + D$. Furthermore if $c_1 + d_1$ and $c_2 + d_2$ are in $C + D$ we have to show that

$$(c_1 + d_1) + (c_2 + d_2) \in C + D.$$

But

$$(c_1 + d_1) + (c_2 + d_2) = (c_1 + c_2) + (d_1 + d_2) = c + d$$

for some $c \in C$ and $d \in D$ as C and D are linear.
We now prove that

$$(C + D)^\perp = C^\perp \cap D^\perp.$$

Claim $(C + D)^\perp \subseteq C^\perp \cap D^\perp$: Let $x \in (C + D)^\perp$. Then for all $v \in C + D$ it holds that $x \cdot v = 0$.

We show that $x \in C^\perp$ and $x \in D^\perp$. Let $c \in C$. Now,

$$c = c + \mathbf{0} \in C + D$$

and therefore $x \cdot c = 0$ and we conclude that $x \in C^\perp$. Similarly, for all $d \in D$,

$$d = \mathbf{0} + d \in C + D$$

and therefore $x \cdot d = 0$ and $x \in D^\perp$.

Claim $C^\perp \cap D^\perp \subseteq (C + D)^\perp$: For all

$$v \in C^\perp \; D^\perp : v \cdot c = 0$$

and $v \cdot d = 0$, for all ($c \in C$ and $d \in D$). Therefore $v \cdot (c + d)$ equals zero as well, and $v \in (C + D)^\perp$.

24. (Counting the codes): Determine the number of binary linear codes with parameters $(n, n - 1, 2)$.

The number of binary linear codes with parameters $(n, n - 1, 2)$ is 1 !

To prove this we begin by showing that there exists at most one binary linear code with the given parameters and conclude by showing the existence of such a code.

Assume that there exist two binary linear codes C_1, C_2 with parameters $(n, n - 1, 2)$. Note that for every code with these parameters, if we delete the last coordinate, we obtain all strings of length $n - 1$ (since all codewords differ in at least 2 coordinates, deleting the last coordinate results in a set of different strings of the same size as the original code).

Now, if there exist two different codes C_1, C_2 with parameters $(n, n-1, 2)$, then there exists some $v \in \{0,1\}^{n-1}$ such that (w.lo.g.) $v||0 \in C_1$ and $v||1 \in C_2$. If $v = 0^{n-1}$, then C_2 is not a linear code since it does not contain the all-zero vector (recall that all codewords must differ in at least 2 coordinates). Let i be the first index in v such that $v_i = 1$ (where v_i denotes the i-th bit in v). Let $v' \in \{0,1\}^{n-1}$ be the binary string that is identical to v in all coordinates except for the i-th coordinate $v'_i = 0$. It must hold that $v'||1 \in C_1$ and $v'||0 \in C_2$ (why?). Let i' be the first index in v' such that $v'_{i'} = 1$. Let $v'' \in \{0,1\}^{n-1}$ be the binary string that is identical to v' in all coordinates except for the i'-th coordinate which is equal to 0. It must hold that $v''||0 \in C_1$ and $v''||1 \in C_2$.

We continue with this procedure until we obtain the all-zero string of length $n-1$. It must hold that either C_1 contains $0^{n-1}||1$ or C_2 contains $0^{n-1}||1$ and therefore either C_1 is not a linear code or C_2 is not a linear code.

We now show the set of binary strings with even weight is a linear $(n, n-1, 2)$ code. It is easy to verify that this set constitutes a linear code with minimum distance 2.

Finally we prove that exactly 2^{n-1} strings from the set $\mathbb{F}_2^n$ have an even weight. This is true because the set of strings in $\mathbb{F}_2^n$ with even weight can be obtained by adding a parity bit to each string of length $n-1$ and there are exactly 2^{n-1} such strings.

25. (Parity of codewords): Show that in a binary linear code, either all codewords have even Hamming weight or exactly half of the codewords have even Hamming weight.

Observation 1 Let x and y be two vectors in $\mathbb{F}_2^n$. The Hamming weight of $x + y$ is even if and only if the Hamming weight of both vectors is even or the Hamming weight of both vectors id odd.

Let C be a binary linear code and let $v_1, \ldots, v_k$ be a basis for C (that is, every codeword in C is a linear combination of $v_1, \ldots, v_k$). Now, if the Hamming weight of all $v_1, \ldots, v_k$ is even, then by the observation, all codewords in C have even Hamming weight. Assume there exist $t \geq 1$ vectors in the basis with odd Hamming weight. W.l.o.g , we assume that $v_1, \ldots, v_t$ are the vectors with odd Hamming weight and $v_{t+1}, \ldots, v_k$ are the vectors with even Hamming weight. Every codeword in C is a linear combination of $v_1, \ldots, v_k$ and therefore can be viewed as a binary string of length k (where we have 1 in the i-th coordinate if and only if v_i appears in the linear combination). Now, each codeword has Even Hamming weight if and only if the number of the vectors from $v_1, \ldots, v_t$ in the combination is even (why?). That is, a codeword c has even Hamming weight if and only if the number of 1's in coordinates 1 to t in the binary string that represents the combination is even. Therefore in order to find the number of codewords with even Hamming weight, we count the number of binary strings of length k where the number of 1's in coordinates 1 to t is even. This equals the number of binary strings of length t with even number of 1's × number of binary strings of length

$k-t$ (recall that we may have every string in the coordinates $t+1$ to k). This equals

$$\frac{2^t}{2} \times 2^{k-t} = 2^{k-t+t-1} = 2^{k-1}.$$

26. (McWilliams Identity): Given is a generator matrix of a (4,2) code C with $d=2$

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

Find the weight distribution of the dual code $C^{\perp}$.

The weight distribution of C is obviously given by $(1,0,1,2,0)$. We need to compute

$$W_C(x+y, x-y) = \sum_{i=0}^{4} A_i (x+y)^{n-i}(x-y)^i$$

This gives that $W_C(x+y, x-y)$ equals to

$$\begin{aligned}
&= (x+y)^4 + (x+y)^2(x-y)^2 + 2(x+y)(x-y)^3 = \\
&= (x+y)^2[(x+y)^2 + (x-y)^2] + 2(x+y)(x-y)^3 = \\
&= (x+y)^2[2x^2+2y^2] + 2(x+y)[x^3 - 3x^2y + 3xy^2 - y^3] = \\
&= 2x^4 + 2x^2y^2 + 4x^3y + 4xy^3 + 2x^2y^2 + 2y^4 + \\
&+ [2x^4 - 6x^3y + 6x^2y^2 - 2xy^3 + 2yx^3 - 6x^2y^2 + 6xy^3 - 2y^4] = \\
&= 4x^4 + 4x^2y^2 + 8xy^3
\end{aligned}$$

Thus the weight distribution of the dual code is,

$$W_{C^{\perp}} = \frac{1}{4} W_C(x+y, x-y) = x^4 + x^2y^2 + 2xy^3.$$

The distribution is the same. Check that (0111) and (1110) are one basis of H.

27. (Concatenation of codes):

28. Let C_1 and C_2 be two linear codes over $\mathbb{F}_q$. Show that $C = \{(c_1||c_2) : c_1 \in C_1, c_2 \in C_2\}$ (where $||$ stands for concatenations) is a linear code with $d = \min\{d_1, d_2\}$.

We first show that C is a linear code. Let $x, y \in C$ and $\alpha, \beta \in \mathbb{F}_q$. We show that

$$\alpha x + \beta y \in C.$$

From the definition of C: $x = x_1||x_2$ and $y = y_1||y_2$ where $x_1, y_1 \in C_1$ and $x_2, y_2 \in C_2$. Now,

$$\alpha x + \beta y = \alpha(x_1||x_2) + \beta(y_1||y_2) = \alpha x_1 + \beta y_1 || \alpha x_2 + \beta y_2$$

, where by the linearity of C_1 and C_2,

$$\alpha x_1 + \beta y_1 \in C_1$$

and

$$\alpha x_2 + \beta y_2 \in C_2$$

and therefore $\alpha x + \beta y \in C$.

We now show that $d = \min\{d_1, d_2\}$. Since every codeword in C is a concatenation of a codeword form C_1 and a codeword from C_2, it is clear that $d \geq \min\{d_1, d_2\}$. Now, assume w.lo.g, that $d_1 \leq d_2$. The codeword$(c_1||\mathbf{0})$ where $c_1 \in C_1$ such that $wt(c_1) = d_1$ and $\mathbf{0}$ is the all-zero codeword in C_2, has weight d_1 and therefore $d = \min\{d_1, d_2\}$.

29. (Combinatorial bounds I): Prove that $A_q(n, d) \leq qA_q(n-1; d)$, where $A_q(n, d)$ is the maximal number of codewords of length n with distance d over an alphabet of q symbols.

Let C be a code with $M = A_q(n, d)$. We divide the codewords in C into disjoint sets by the value of the first location in the codeword (that is, all codewords in the same subset have the same value in their first location).

Now, note that the number of codewords in each subset is at most

$$A_q(n-1, d).$$

(If we remove the first location of all codewords in each subset, we obtain a code with parameters $n-1$ and d.) Now, there are at most q subsets and therefore

$$M = A_q(n, d) \leq qA_q(n-1, d).$$

30. (Combinatorial bounds II): Show that the minimum distance of a perfect code must be odd.

Assume C is a perfect code with even weight d. Recall that a perfect code is defined as an e-error correcting $[n, M]$ code over alphabet A for which every n-tuple over A is in the sphere of radius e about some codeword.

We first prove that there exists a word $x \in \mathbb{F}_q^n$ such that

$$d(x, c) \geq d/2 \forall c \in C$$

Let c_1, c_2 be 2 codewords with $d(c_1, c_2) = d$. Let x be the vector that agrees with c_1 and c_2 in all their identical entries and for the d different locations, it has the same symbol as c_1 for the first $d/2$ and the same as c_2 for the last $d/2$ locations. It is easy to see that

$$d(x, c_1) = d(x, c_2) = d/2.$$

Now, for every $c \in C$ such that $c \neq c_1, c_2$ it holds that $d(c, c_1) \leq d(c, x) + d(x, c_1)$ and therefore

$$d(c, x) \geq d(c, c_1) - d(x, c_1) \geq d - d/2 = d/2.$$

We conclude that $d(x,c) \geq d/2$ for all $c \in C$. However, this implies that the spheres of radius

$$\lfloor \frac{d-1}{2} \rfloor$$

around codewords of C do not contain x and therefore

$$MV_q(n, \lfloor \frac{d-1}{2} \rfloor) < q^n$$

and the code is not perfect. Here,

$$V_q(n,r) = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i$$

is the volume of the sphere of radius r.

31. (Perfect codes I): Show that the binary repetition code of length n, with n odd, is perfect. How many errors does it correct?

    Let

    $$C = \{a_0 = 000\ldots0, a_1 = 111\ldots1\}$$

    be of length n. Any vector $x \in \mathbb{F}_2^n$ has t coordinates 1, and $n-t$ coordinates 0. So $d(x,a_0) = t$ and $d(x,a_1) = n-t$.

    Hence, if $t < n/2$, then x is uniquely decoded as a_0, whereas, if $t > n/2$, then x is uniquely decoded as a_1. So C is perfect and corrects

    $$\lfloor n/2 \rfloor = (n-1)/2$$

    errors.
    This can also be done using the Sphere Packing Bound.

32. (Direct product codes): The product of a (7,4) Hamming code with itself is a (49,16) binary code with minimum distance 9 and therefore error-correcting ability 4 (check slides for general properties). Direct product code is defined as a Kronecker product of codes in this case $V = C \otimes C$. This means that if for instance $\mathbf{c}_1 = (1001100)$ and $\mathbf{c}_2 = (0100110)$ are two codeword of C then the associated codeword in V is given by,

$$\mathbf{c}_1^T \mathbf{c}_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} (0100110) = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \in V$$

Associated codeword $\mathbf{c}^V$ is the consecutive rows of the above matrix

$$\mathbf{c}^V = (0100110|0000000|\cdots|0000000)$$

Note that if $\mathbf{c}^V \neq 0$ then there must be a nonzero row and its weight is ≥ 3 since it is a codeword of a (7,4,3) code. But once there is a single one in some row the number of ones in the corresponding column must be also ≥ 3 (it is the weight of $\mathbf{c}_1$)

We want to devise an ad hoc error-correction method that can correct up to four bit errors for at least some "nice" distributions of errors in a 7×7 array of received bits. Consider the various number of rows that can be affected by up to four errors.

> The following ad hoc error-correction procedure is one method for correcting up to four bit errors in a 7×7 codeword array. Let R be the set of rows in which errors are detected and let C be the set of columns in which errors are detected (simply check whether received rows/columns are the codewords of the Hamming (7,4,3) code). Apply the following correction procedure:
>
> (a) *Case 1* $|R| \geq |C|$. Run through the rows in R. For each row $\mathbf{r} \in R$, correct r using the nearest-neighbor error correction for the (7,4) Hamming code. Then apply error detection to each column of the received array. If any columns not in C show errors, **undo the correction performed on** r.
>
> Example:
>
> $$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$
>
> In this case $|R| = 4$ and $|C| = 2$. The errors are detected, and then any single error in each row is corrected using Hamming code correction so that
>
> $$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$
>
> *Case 2*: $|R| < |C|$. Proceed as above, with the roles of rows and columns interchanged.
>
> (b) Update R and C. If both are now empty, stop. (in the above example we are done - no more errors either in rows or in columns)
>
> (c) But it might be the case that we miscorrect some columns or rows so we are not done in the first step. We need to undo the correction of such a row.

Example:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

The green colour indicates a new error introduced by error correction. Since this column was not detected to have errors originally we must undo error correction of this row. Now we proceed with error correction of the columns,

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

There are a few more complicated patterns that might be discussed but we stop here.

33. Show that a perfect binary $[n, M, 7]$ code has $n = 7$ or $n = 23$.

The Sphere Packing Bound for a binary $[n, M, 7]$ code says that:

$$M\left\{\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \binom{n}{3}\right\} = 2^n.$$

Thus,

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \binom{n}{3} = 2^r.$$

Rewriting this we have,

$$\begin{aligned} 1 + n + \frac{1}{2}n(n-1) + \frac{1}{6}n(n-1)(n-2) &= 2^r, \\ 6(n+1) + 3n(n-1) + n(n-1)(n-2) &= 3 \times 2^{r+1}, \\ 6(n+1) + n(n-1)\{3 + (n-2)\} &= 3 \times 2^{r+1}, \\ (n+1)\{n^2 - n + 6\} &= 3 \times 2^{r+1}, \\ (n+1)\{(n+1)^2 - 3(n+1) + 8\} &= 3 \times 2^{r+1}. \quad (*) \end{aligned}$$

A bit of number theory implies that $n + 1$ must be divisible by some power of 2. First note that writing (*) modulo 8 we have,

$$(n+1)\{(n+1)^2 - 3(n+1)\} \equiv 0 \pmod 8,$$

as $8|2^{r+1}$ for $r \geq 2$ which is always the case. Thus $8|(n+1)$.
If 16 divides $n + 1$, then the second term on the LHS is divisible by 8 but not by

16. In general the second term on the LHS is then of the form $(2k+1)8$ for $k \geq 0$. If it is 8 ($k = 0$), then

$$(n+1)^2 - 3(n+1) = 0,$$

which is impossible, since $n \geq 7$. If it is 24 ($k = 1$), then

$$(n+1)^2 - 3(n+1) - 16 = 0,$$

which is also impossible, as the discriminant is 73.
Therefore, $n + 1$ divides 24 (as we have 3 on the RHS), so that $n = 7, 11, 23$. Now, $n = 11$ does not satisfy Equation (*). So, $n = 7$ or 23. In fact, perfect codes of these lengths exist, the repetition code of length 7 and the Golay code, respectively.

34. (Bounds): Use the Sphere Packing Bound to find an upper bound for M of a binary $[5, M, 3]$ code.

For $q = 2$, the Sphere Packing Bound for an $[n, M, 2e+1]$ code is,

$$M\left\{\binom{n}{0} + \binom{n}{1} + \ldots + \binom{n}{e}\right\} \leq 2^n.$$

For $n = 5, d = 3, e = 1$ we get,

$$M\left\{1 + \binom{5}{1}\right\} \leq 32,$$

therefore $M \leq 5$.

35. (Bounds II): The previous exercise gives an upper bound for $A_2(5,3)$. Now, show by construction that $A_2(5,3) = 4$.

The idea is to exhaustively find the maximal number of the codewords. Choose two words in C as $a_1 = 00000$, and w.l.o.g. $a_2 = 11100$. Since $d(x, a_1) \geq 3$ for any x in C the only other possible elements of C are the 9 words with three 1s, apart from a_2, the 5 words with four 1s, and $u = 11111$. As $d(u, a_2) = 2$, we have $u \notin C$.
$wt(c) = 3$: 11010, 11001, 10110, 10101, 10011, 01110, 01101, 01011, 00111;
$wt(c) = 4$: 11110, 11101, 11011, 10111, 01111.
The words with three 1s and two of the first three coordinates 1 are at distance 2 from a_2. This leaves

$$b_1 = 10011, \quad b_2 = 01011, \quad b_3 = 00111.$$

Similarly, the first two words with four 1s are at distance 1 from a_2. This leaves

$$c_1 = 11011, \quad c_2 = 10111, \quad c_3 = 01111.$$

Now, $d(b_i, b_j) = 2$, $d(c_i, c_j) = 2$ for $i \neq j$. So there can only be one b_i and one c_j in C. Hence $|C| \leq 4$.
In fact, taking $b_1 \in C$, the only possibility is c_3. This gives $C = \{a_1, a_2, b_1, c_3\}$ as a [5, 4, 3] code.

36. (Even weight codes): Show that if there is a binary $[n, M, d]$ code with d even then there is an $[n, M, d]$ code in which all codewords have even weight.

   Let C be an $[n, M, d]$ code with d even. Then we can puncture C to get an $[n-1, M, d-1]$ code C' (assuming no coordinate is zero for all codewords). From this code we can get $\overline{C'}$ as the extended code by, calculating $c_n = \sum_{i=1}^{n-1} c_i \pmod 2$. The extended code $\overline{C'}$ is again $[(n, M, d)]$ but all the codewords have even weight. As an example consider,

$$C = \left\{ \begin{array}{cccc} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array} \right. \rightarrow C' = \left\{ \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right. \rightarrow \overline{C} = \left\{ \begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array} \right.$$

37. (Extended code): Let C be a binary code of length n. Form a binary code C' of length $n+1$ as follows:

$$x = x_1 x_2 \ldots x_n \in C \Rightarrow x' = x_1 x_2 \ldots x_n x_{n+1} \in C'$$

where

$$x_{n+1} = \left\{ \begin{array}{ll} 1 & if \;\; w(x) \;\; is \;\; odd \\ 0 & if \;\; w(x) \;\; is \;\; even \end{array} \right.$$

Show that, if C is linear, then C' is also linear; it is called the extended code.

   Let

$$\begin{array}{rcl} C_0 & = & \{x_1 x_2 \ldots x_{n+1} \in V(n+1, 2) | x_1 x_2 \ldots x_n \in C; x_{n+1} \in \mathbb{F}_2\}, \\ C_1 & = & \{x_1 x_2 \ldots x_{n+1} \in V(n+1, 2) | x_1 + x_2 + \ldots + x_n + x_{n+1} = 0\}. \end{array}$$

   Then C_0 and C_1 are both subspaces of $V(n+1, 2)$, and $C' = C_0 \cap C_1$. Hence C' is a subspace.
   Now for $x, y \in V(n, 2)$,

$$w(x+y) = \sum_{i=1}^{n} (x_i + y_i) = \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i = w(x) + w(y) \pmod 2.$$

   To check that C' is linear, only one condition is required: $x', y' \in C' \Rightarrow x' + y' \in C'$. There are three cases.
   (1) $w(x)$ even, $w(y)$ even; then $w(x+y)$ is even. So

$$(x+y)' = (x_1 + y_1, \ldots, x_n + y_n, 0)$$

$$= (x_1, \ldots, x_n, 0) + (y_1, \ldots, y_n, 0) = x' + y'.$$

   Thus the mapping is linear in this case.
   (2) $w(x)$ odd, $w(y)$ odd; then $w(x+y)$ is even. So

$$(x+y)' = (x_1 + y_1, \ldots, x_n + y_n, 0)$$

$$= (x_1, \ldots, x_n, 1) + (y_1, \ldots, y_n, 1) = x' + y'.$$

Thus the mapping is linear in this case too.
(3) $w(x)$ odd, $w(y)$ even; then $w(x+y)$ is odd. So

$$(x+y)' = (x_1+y_1,\ldots,x_n+y_n,1)$$

$$= (x_1,\ldots,x_n,1)+(y_1,\ldots,y_n,0) = x'+y'.$$

The mapping is linear in this case too, that is C' is linear.

38. (Constructing new codes II): Show that if a binary $(n,k,2t+1)$ code exists, then so does an $(n+1,k,2t+2)$ code.

Let C be an $(n,k,2t+1)$ code. We construct C', an $(n+1,k,2t+2)$ code from C as follows:
Let $\mathbf{x} = (x_1,\ldots,x_n) \in C$. Corresponding to $\mathbf{x}$, C' contains the codeword $\mathbf{x}' = (x_1,\ldots,x_n,\sum_{i=1}^{n} x_i)$.

Clearly, C' is an $(n+1,k,d)$ code - we want to show that

$$d \geq 2t+2,$$

i.e., for all $\mathbf{x},\mathbf{y} \in C$ we have

$$d(\mathbf{x}',\mathbf{y}') \geq 2t+2.$$

Notice that

$$d(\mathbf{x}',\mathbf{y}') \geq d(\mathbf{x},\mathbf{y}).$$

Hence, it suffices to consider the case

$$d(\mathbf{x},\mathbf{y}) = 2t+1.$$

Let us permute the coordinates of C so that $\mathbf{x}$ and $\mathbf{y}$ agree in the first $m = n-(2t+1)$ positions and disagree in the last $2t+1$ positions. Now consider the quantity

$$\sum_{i=1}^{n}(x_i+y_i),$$

which equals to "(2× # 1's in the first $m = n-(2t+1)$ positions + $(2t+1)$)".

The parity is clearly odd and therefore we get that exactly one of $\sum_{i=1}^{n} x_i$ and $\sum_{i=1}^{n} y_i$ is one (modulo 2) and the other is zero. Thus $\mathbf{x}$' and $\mathbf{y}$' disagree in the last coordinate and so the distance between them is $2t+2$.

39. (Construction III): Given an (n,k,d) linear code over $\mathbb{F}_q$, can one always construct an $(n+1,k,d+1)$ linear code ?

No, we show the following counter-example: Let C be the binary linear code containing all vectors of length 3 with even Hamming weight. This is a (3, 2, 2) code. If the claim is true, then there must exist a (4, 2, 3) code. Take simply any two vectors of weight 3 in $\mathbb{F}_2^4$, say (1110) and (0111). Clearly we cannot get distance 3 for any choice of the basis.

40. (Concatenation of codes):

41. Let C_1 and C_2 be two linear codes over $\mathbb{F}_q$. Show that $C = \{(c_1||c_2) : c_1 \in C_1, c_2 \in C_2\}$ (where $||$ stands for concatenations) is a linear code with $d = \min\{d_1, d_2\}$.

    We first show that C is a linear code. Let $x, y \in C$ and $\alpha, \beta \in \mathbb{F}_q$. We show that $\alpha x + \beta y \in C$. From the definition of C: $x = x_1||x_2$ and $y = y_1||y_2$ where $x_1, y_1 \in C_1$ and $x_2, y_2 \in C_2$. Now,

    $$\alpha x + \beta y = \alpha(x_1||x_2) + \beta(y_1||y_2) = \alpha x_1 + \beta y_1||\alpha x_2 + \beta y_2$$

    , where by the linearity of C_1 and C_2, $\alpha x_1 + \beta y_1 \in C_1$ and $\alpha x_2 + \beta y_2 \in C_2$ and therefore $\alpha x + \beta y \in C$.
    We now show that $d = \min\{d_1, d_2\}$. Since every codeword in C is a concatenation of a codeword form C_1 and a codeword from C_2, it is clear that $d \geq \min\{d_1, d_2\}$. Now, assume w.lo.g, that $d_1 \leq d_2$. The codeword$(c_1||\mathbf{0})$ where $c_1 \in C_1$ such that $wt(c_1) = d_1$ and $\mathbf{0}$ is the all-zero codeword in C_2, has weight d_1 and therefore $d = \min\{d_1, d_2\}$.

42. Hadamard matrices: Recall that an $n \times n$ matrix H all of whose entries are from $\{+1, -1\}$ is a Hadamard matrix if $H \cdot H^T = n \cdot I$ where the matrix product is over the reals and I is the $n \times n$ identity matrix.

    (a) Show that if there is an $n \times n$ Hadamard matrix then n is either 1 or 2 or a multiple of 4.

    Let $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ be three distinct rows of a Hadamard matrix. (So we are assuming $n \geq 3$.) For $i, j \in \{1, -1\}$, let

    $$S_{i,j} = \{k | a_k = i \cdot b_k \text{ and } a_k = j \cdot c_k\}.$$

    Let $\alpha = |S_{1,1}|$, $\beta = |S_{1,-1}|$, $\gamma = |S_{-1,1}|$, and $\delta = |S_{-1,-1}|$. Then $\alpha + \beta$ counts the number of coordinates where $\mathbf{a}$ equals $\mathbf{b}$ and so $\alpha + \beta = n/2$. Similarly $\alpha + \gamma$ counts the number of coordinates where $\mathbf{a}$ equals $\mathbf{c}$ and so $\alpha + \gamma = n/2$. Finally, $\alpha + \delta$ counts the number of coordinates where $\mathbf{b}$ equals $\mathbf{c}$ and so $\alpha + \delta = n/2$. And of course, $\alpha + \beta + \gamma + \delta = n$. Solving the 4×4 linear system above, we get $\alpha = \beta = \gamma = \delta = n/4$. Since each is an integer, we have n must be a multiple of 4.

    (b) Given an $n \times n$ Hadamard matrix H_n and an $m \times m$ Hadamard matrix H_m, construct an $(nm) \times (nm)$ Hadamard matrix.

    Let $\mathbb{F}$ be any field (say rationals, for this problem). For vectors $\mathbf{a} \in \mathbb{F}^n$ and $\mathbf{b} \in \mathbb{F}^m$, let $\mathbf{a} \otimes \mathbf{b} \in \mathbb{F}^{nm}$ denote their outer product (aka tensor product), namely the vector whose ij-th coordinate is $a_i \cdot b_j$. Note that if $\mathbf{a}$, $\mathbf{b}$ are $+1/-1$ vectors, then so is $\mathbf{a} \otimes \mathbf{b}$. Furthermore, if $\mathbf{a}, \mathbf{c} \in \mathbb{F}^n$ and $\mathbf{b}, \mathbf{d} \in \mathbb{F}^m$,

$$\langle \mathbf{a} \otimes \mathbf{b}, \mathbf{c} \otimes \mathbf{d} \rangle = \sum_{ij} a_i b_j c_i d_j = (\sum_i a_i c_i)(\sum_j b_j d_j) = \langle \mathbf{a}, \mathbf{c} \rangle \cdot \langle \mathbf{b}, \mathbf{d} \rangle.$$

We show how to use tensor products to build a big Hadamard matrix from two smaller ones.
Let $\mathbf{u}_1 \ldots, \mathbf{u}_n$ be the rows of H_n and let $\mathbf{v}_1 \ldots, \mathbf{v}_m$ be the rows of H_m. By the condition $H_n H_n^T = nI$, we have $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0$ if $i \neq j$. (Similarly for the $\mathbf{v}_i$'s.) Let H_{nm} be the matrix whose rows are $\mathbf{u}_i \otimes \mathbf{v}_j$ for all $i \in [n]$, $j \in [m]$. As noted above, this is a $+1/-1$ matrix. Thus the diagonal entries of $H_{nm} H_{nm}^T$ are all nm as required. Now consider the off-diagonal entry $(H_{nm} H_{nm}^T)_{(ij),(kl)} = \langle \mathbf{u}_i \otimes \mathbf{v}_j, \mathbf{u}_k \otimes \mathbf{v}_l \rangle = \langle \mathbf{u}_i, \mathbf{u}_k \rangle \cdot \langle \mathbf{v}_j, \mathbf{v}_l \rangle$. Since at least one of the conditions $i \neq k$ or $j \neq l$ holds, we have the above inner product is zero. This proves the off-diagonal entries are zero as required.

43. Suppose C is a code of length n over the q-ary alphabet A. Let $w, x \in C$, $w \neq x$, and $v \in A^n$. In terms of these vectors answer the following (and generalize):

(a) What does it mean to say that C is t-error-detecting? What does it mean to say that C is t-error-correcting? Prove that if C is $2t$-error-detecting, then C is t-error-correcting. Hint: Use the triangle inequality and show that if C is not t-error correcting then it is not $2t$-error detecting.

C is t-error-detecting if there do not exist words $w, x \in C$ with $d(w,x) \leq t$. C is t-error-correcting if there do not exist words $v \in A^n$ and $w, x \in C$ such that $w \neq x$ and

$$d(v,w) \leq t, \;\; d(v,x) \leq t.$$

Suppose C fails to be t-error-correcting. Then there are v, w, x as above. By the triangle inequality, we have

$$d(w,x) \leq d(v,w) + d(v,x) \leq t + t = 2t,$$

and so C is not $2t$-error-detecting. Hence if C is $2t$-error-detecting, then it is t-error-correcting.

(b) Show that if C is t-error-correcting, then

$$|C| \leq \frac{q^n}{\binom{n}{0} + (q-1)\binom{n}{1} + (q-1)^2\binom{n}{2} + \ldots + (q-1)^t\binom{n}{t}}.$$

Hint: Use the concept of disjoint spheres.

If C is t-error-correcting, then the spheres $S(x,t)$ for $x \in C$ must be disjoint. Thus we have,

$$\begin{aligned} |\cup_{x \in C} S(x,t)| &= \sum_{x \in C} |S(x,t)| \\ &= \sum_{x \in C} \left(\binom{n}{0} + \binom{n}{1}(q-1) + \ldots + \binom{n}{t}(q-1)^t \right). \end{aligned}$$

But $\cup_{x\in C} S(x,t)$ is a subset of A^n, which contains exactly q^n words. And so

$$|C| \cdot \left(\binom{n}{0} + \binom{n}{1}(q-1) + \ldots + \binom{n}{t}(q-1)^t \right) \leq q^n,$$

which gives the result.

(c) Suppose that $n^2+n+1 > 2^l$ for some integer l, and that C is a binary linear (n,k)-code which is 2-error-correcting. Prove that $k < n-l+1$.

By the above inequality, we have

$$|C| \leq \frac{2^n}{\binom{n}{0}+\binom{n}{1}+\binom{n}{2}} = \frac{2^n}{1+n+\frac{n(n-1)}{2}}.$$

Thus,

$$2^k \leq \frac{2^n}{\frac{1}{2}(2+n+n^2)} < \frac{2^n}{\frac{1}{2}(2^l)} = 2^{n-l+1}.$$

Thus, $k < n-l+1$.

44. This problem concerns the bounds on codes.

(a) Let B be an alphabet of size q and $C \subset B^n$ be a q-ary block code of length n.

(i) Define (mathematically) the Hamming distance d on B^n.

For $x, y \in B^n$ we define,

$$d(x,y) = \#\{x_i \neq y_i; i = 1, \ldots, n\}.$$

(ii) Define the minimum distance $d(C)$ of the code C.

$$d(C) = \min_{x,y\in C; x\neq y} d(x,y).$$

(b) Let the parameter $A_q(n,d)$ define the maximum number of codewords of length n over B such that the Hamming distance between any two codewords is $\geq d$. State and prove the sphere-packing bound for $A_q(n,d)$.
Hint: How many disjoint spheres of *suitable* radius can be packed in the space.

Since $d(C) = d$ we know that C is $t = \frac{d-1}{2}$ error-correcting code. The bound is,

$$A_q(n,d) \leq q^n/|S(x,t)|.$$

In the class a different notation was used $A_q(n,d) \leq q^n/V_q(n,t)$, so obviously $V_q(n,t) = |S(x,t)|$. Note that

$$V_q(n,t) = \binom{n}{0} + (q-1)\binom{n}{1} + (q-1)^2\binom{n}{2} + \ldots + (q-1)^t\binom{n}{t},$$

hence the bound is the same as in Problem 1.

(c) Then prove that $A_q(n-1,d) \geq A_q(n,d)/q$.

See the previous exam. Let C be a code with $A_q(n,d)$ codewords. W.l.o.g. consider the last coordinate of C that is c_n. Then we can sort the codewords w.r.t. this coordinate, i.e. split C into q sets. Then there must be some value in this coordinate such that $|\{c \in C : c_n = a\}| \geq A_q(n,d)/q$ for some $a \in B$. Then we take the codewords of C for which $c_n = a$ and delete (puncture) this coordinate. The code is obviously an $[n-1, M', d]$ code with $M' \geq A_q(n,d)/q$. Also, deleting this coordinate does not affect d.

(d) In each of the following cases either construct a code with the specified parameters or explain why no such code exists.

(i) A 5-ary [7, 26, 6] code.

Since all codewords differ from one another in at least 6 places, the 2-coordinate words we get by deleting the last 5 coordinates of each codeword must all be distinct. Hence a 5-ary [7,M, 6] code must have $M \leq 5^2 = 25$. Hence, 5-ary [7, 26, 6] code does not exist.

(ii) A 5-ary $[8, 130, 6]$ code.

A 5-ary [8, 130, 6] code does not exist by (i) and part (c).

45. Let C be the binary linear code with generator matrix

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

(a) Write down a parity check matrix H for C. Explain how the minimum distance of C may be deduced from H. Find $d(C)$.

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

$d(C) = d$ if and only if no set of $(d-1)$ columns of H is linearly dependent, but some set of d columns is. In this case, no pair of columns is linearly dependent but the first 3 columns sum to 000, hence are linearly dependent. So $d(C) = 3$.

(b) How many cosets does C have? How many cosets are led by weight 1 vectors ? Does any coset have a weight 2 coset leader?

C has $|Z_2^6|/|C| = 2^6/2^3 = 8$ cosets. Since $d(C) = 3$, every weight 1 vector is a coset leader, so 6 of the cosets have weight 1 leaders. C itself is led by 000000, so that leaves 1 coset unaccounted for. There are $\binom{6}{2}$ weight 2 vectors in Z_2^6, and the question is whether all of these are contained in the cosets with weight 1 leaders. But in order to lie in coset $\mathbf{v} + C$ with $w(v) = 1$, a vector must be at distance 1 from one of the codewords of C. Now,

$$C = \{f000000; 100101; 010110; 110011; 001011; 101110; 011101; 111000\},$$

so the only weight 2 vectors with this property are,

$$000101; 100001; 100100; 000110; 010010; 010100; 000011; 001001;$$

$$001010; 011000; 101000; 110000.$$

Hence the remaining weight 2 vectors, namely 100010, 010001 and 001100 must lie in the 8th coset, and any of these 3 may be chosen as the coset leader.

(c) Construct a syndrome look-up table for C. Hence, or otherwise, decode the received vectors 100110, 011101 and 101001.

We compute $S(\mathbf{v}_r) = \mathbf{v}_r H^T$ for each of the coset leaders $\mathbf{v}_1, \ldots, \mathbf{v}_8$. Since there are 3 different choices for $\mathbf{v}_8$ 3 different tables are possible (they differ only in the last row). Choosing 100010 as our weight 2 coset leader, we obtain:

coset leader	syndrome
000000	000
100000	101
010000	110
001000	011
000100	100
000010	010
000001	001
100010	111

$S(100110) = 011$ so 100110 lies in $001000 + C$. Hence we correct it by subtracting 1 from its 3rd digit: 100110 → 101110.
$S(011101) = 000$ so 011101 is a codeword and needs no correction.
$S(101001) = 111$ so 101001 lies in $100010 + C$. Hence we correct it $101001 \to 101001 - 100010 = 001011$. Note that your answer will be different if you chose a different weight 2 coset leader.

(d) Puncturing the code means deleting some coordinates of the code. Discuss the effect of puncturing on the minimum distance and the rate of the code.

Puncturing may or may not decrease the minimum distance, hence $d' \leq d$. Anyway, since $n' < n$ and the dimension is the same it means that rate $R = k/n'$ is increased.

46. A (6, 3) linear block code C over GF(2) is defined by the following parity check matrix,

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

(a) Find the generator matrix of C.

The parity check matrix is simply obtained from H as,

$$G = [A^T \; I_3] = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

(b) The parity check matrix H does not allow the presence of the codewords of weight < 3 (apart from the all zero codeword). Explain why ?

We cannot have a codeword of weight 2 since then $Hc^T = 0$ is not satisfied due to the properties of H that no 2 columns of H are linearly dependent.

(c) Suppose that the code is used for error detection only over a binary symmetric channel with error rate $p = 10^{-3}$. Find the probability of undetected error.
Hint: W.l.o.g. assume that all zero codeword was transmitted. Be careful with the interpretation of undetected error (for Pavel only : of course the error must be a codeword)

An undetected error occurs if and only if the error pattern is a codeword ! The weight distribution of the code is (1, 0, 0, 4, 3, 0, 0), and there are 4 possibilities that an error of weight 3 and 3 possibilities that an error of weight 4 goes undetected.Therefore,

$$\begin{aligned} P_e &= 4p^3(1-p)^3 + 3p^4(1-p)^2 \\ &= 4 \cdot 10^{-9}(0.999)^3 + 3 \cdot 10^{-12}(0.999)^2 \approx 4 \times 10^-9. \end{aligned}$$

(d) Suppose that the code is used for erasure correction over a binary erasure channel with erasure probability $\epsilon = 10^{-2}$. How many erasures the code can always correct ?

Since $d = 3$ the code can always correct 2 erasures.

(e) Decode the received word $(\_\,\_\,\_\,110)$

Determining erasure values corresponds to solving a system of 3 equations. Thus we need linear independency of these equations which comes from the independency of the columns of H. Denoting the first 3 positions by r_1, r_2, r_3 from $H\mathbf{c}^T = 0$ we get,

$$\begin{aligned} r_1 &= 1 \\ r_2 &= 1 \\ r_3 &= 0 \end{aligned}$$

Thus, $r \rightarrow c = (110110)$.

(f) For the erasure probability $\epsilon = 10^{-2}$ find the probability of decoder failure, that is, the probability that the transmitted codeword cannot be determined from the unerased bits (for sufficiently many erasures)

Hint: One erasure weight need to be carefully investigated.

Since any 4 columns or more of H are linearly dependent we cannot correct 4 or more erasures. However, some erasures of weight 3 can be corrected as in (e) but those submatrices of H of size 3×3 whose columns are not linearly independent are those erasure patterns that cannot be corrected. The number of such submatrices is 4 (by inspection of H) and therefore,

$$\begin{aligned} Pfail &= 4\epsilon^3(1-\epsilon)^3 + \binom{6}{4}\epsilon^4(1-\epsilon)^2 + \binom{6}{5}\epsilon^5(1-\epsilon) + \binom{6}{6}\epsilon^6 = \\ &= 4 \cdot 10^{-6}(0.99)^3 + 15 \cdot 10^{-8}(0.99)^2 + 6 \cdot 10^{-10}(0.99) + 10^{-12} = \\ &\approx 4 \cdot 10^{-6}(0.99)^3 + 15 \cdot 10^{-8}(0.99)^2 = 3.9 \times 10^{-6}. \end{aligned}$$

47. This question considers the bounds on codes. Actually, the results in this problem establish the so-called Plotkin bound.

(a) What is meant by a binary $[n, M, d]$-code, i.e. explain the notation ?

An $[n, M, d]$-code C is a code with M codewords over a binary alphabet, all of length n, such that $d(u, v) \geq d$ for all distinct $u, v \in C$.

(b) Suppose C is a binary $[n, M, d]$-code. Regard the codewords as vectors over $GF(2)$, and define an $\binom{M}{2} \times n$ binary matrix D as follows:
The rows of D correspond to all (unordered) pair of codewords in C. The row corresponding to codewords $\mathbf{u}$ and $\mathbf{v}$ is simply the vector (modulo 2 bitwise) sum of $\mathbf{u}$ and $\mathbf{v}$. (The ordering of the rows of D is not significant.) Write down the array D for the particular code

$$C = \{000000, 001111, 111001, 110110\}.$$

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

(c) Now suppose C is an arbitrary $[n, M, d]$-code. Prove that the number of 1s in D is at least $\binom{M}{2}d$.

Since any row of D is the sum of distinct $\mathbf{u}$ and $\mathbf{v}$ we must have that $wt(D_i) \geq d$ for any row D_i, $1 \leq i \leq \binom{4}{2}$. Therefore $wt(D) \geq \binom{M}{2}d$. Note that $\binom{M}{2} = 6$ for $M = 4$ and $d = 4$.

(d) Prove that the number of 1s in D is at most $nM^2/4$. (Hint: consider D columnwise.)

For any column, say i, the entry is one if the ith coordinates of corresponding $\mathbf{u}$ and $\mathbf{v}$ are such that $\mathbf{u}_i \neq \mathbf{v}_i$. Let the number of 1's in the ith position of the codewords of C is j and the number of codewords of C in the ith coordinate equal to 0 is $M - j$. Then, for each coordinate the number of 1's is $j(M - j) \leq M^2/4$. Since there are n columns $wt(D) \leq nM^2/4$.

(e) Deduce that $M \leq 2d/(2d - n)$, provided that $2d > n$.

From c) and d) we have $\binom{M}{2}d \leq nM^2/4$ which gives $dM(M - 1)/2 \leq nM^2/4$ and therefore, $M \leq 2d/(2d - n)$.