Problem Set 1

Daniel V. Rostovtsev
Date: 17 April, 2018

Problem 1

(SEE ATTACHED .txt FOR SOLUTION TO PROBLEM ONE, IT WAS DONE COMPUTATIONALLY)

Problem 2

(a) Show the matrix A with an additional parity column. Explain how to recover the erased bits in $\hat{A}$ using the parity column. Can you recover them all?

Finging A with parity column:

0	1	1	1	1
0	1	1	0	0
1	1	1	1	0
1	0	1	1	1

Note $\hat{A}$ is as follows:

0	1	1	E
0	E	1	E
1	1	1	E
1	E	1	1

Note that $\oplus$, or XOR in binary can be considered formally as addition in the field $\mathbb{F}_2$. The only way to recover an erased bit is to find a unique solution to the equation

$$a_i + b_i + c_i + d_i = p_i \text{ (in } \mathbb{F}_2)$$

where each p_i is the parity bit, and a_i, b_i, c_i, d_i are either the read bits or unknown erased bits. Since one linear equation can only yield a unique solution if there is only one unknown in a field (in this case, $\mathbb{F}_2$), it follows that there can only be one erasure per row if all the erased bits are to be recovered. So, using the parity bit, this is all that can be recovered from $\hat{A}$:

0	1	1	(1)
0	(1,0)	1	(0,1)
1	1	1	(1)
1	(0)	1	1

(where b_4 is 1 if b_2 is 0, and b_4 is 0 if b_2 is 1)

(b) Assume that you have an $n \times n$ matrix with an additional parity column. We define an *erasure pattern* as a subset of entries in the $n \times (n+1)$ matrix with an additional parity column. What are the erasure patterns that can be corrected with an additional parity column?

Using the same argument in A with linear equations in a field, but altered slightly to account for potential erasures of the parity bit.

$$a_i + b_i + c_i + d_i = p_i \implies a_i + b_i + c_i + d_i + p_i = 0 \text{ in } \mathbb{F}_2$$

Which is a linear equation in $\mathbb{F}_2$ including the parity bit. Thus a maximum of one bit can be erased per row, and all recoverable erasure patterns are patterns with no more than one bit erased per row.

(c) To improve your erasure correcting scheme you decide to add both a parity column and a parity row. Show the matrix A with the additional parity column and parity row. Explain how to recover the erased bits in $\hat{A}$. Can you recover all of them?

A with additional parity column and row:

0	1	1	1	1
0	1	1	0	0
1	1	1	1	0
1	0	1	1	1
0	1	0	1	

Each row or column with at least one unkown contributes to a system of linear equations in $\mathbb{F}_2$ with the computed parity bits. If the number such of linear equations is greater than or equal to the number of unknowns, then all the unknowns can be recovered. In the case of $\hat{A}$, there are 4 rows with at least one unknown, and 2 columns with at least one unknown. Thus there are 6 linear equations. There are 5 erasures, so there are 5 unknowns. 6 linear equations can solve for 5 unknowns in a field, so all E can be recovered. This can be done with linear algebra, or just solving the grid like a sudoku puzzle. For $\hat{A}$ this is trivial. The first, third and fourth rows can be solved immediately with the parity column as in part (a). Then there is only one unknown in the second and fourth columns, which can be solved immediately with the additional parity row.

(d) Assume that the matrix A with the additional parity column and parity row. We define an *erasure pattern* as a subset of entries in the $(n+1) \times (n+1)$ matrix. What are the erasure patterns that can be corrected in an $n \times n$ matrix with additional parity column and parity row.

Using the same logic as part (c), an erasure pattern will yield a system of equations with $r+c$ equations, where r is the number of rows with an unknown, and c is the number of columns. If the number of erasures, n is less than or equal to $r+c$, then the erasure can be corrected.

(e) Assume that you stored the matrix A with an additional parity column and parity row. Can you correct the error in $\tilde{A}$? What are the correctable error patterns that can be corrected in an $n \times n$ matrix with additional parity column and parity row?

No errors can be corrected with an additional parity column and parity row unless the matrix is a 1×1.

A matrix is correctable if the given parity column and rows are exactly able to identify the locations of all the errors, so that they can be altered. That means that given parity column and row errors correspond to unique errors. But, for $n \geq 2$, there are many $n \times n$ matrices that correspond to the same error code! We can show this by solving explicitly for the number of $n \times n$ matrices with a given error code. There are n^2 unknown entries of this matrix. Each entry in the parity column and parity row corresponds to a single linear equation on values of those n^2 unknown entries. There are only $2n$ such relations, n coming from the parity row, and n coming from the parity column. It follows that for $n \geq 3$, there are more unknowns than relations, so multiple matrices correspond to the same error, and can never be corrected. Now to check the special cases of $n = 2$ and $n = 1$.

For $n = 2$: $\begin{array}{|c|c|c|}\hline x_1 & x_2 & p_1 \\ \hline x_3 & x_4 & p_2 \\ \hline p_3 & p_4 & \\ \hline \end{array}$ has a unique code if and only if the following equation is uniquely solvable:

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}$$

Which is true if and only if $A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$ is invertible in $M_{4\times 4}(\mathbb{F}_2)$, which is true if and only if $\det(A)$ is nonzero in $\mathbb{F}_2$. But:

$$\det \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} = \det \begin{pmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} + \det \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = \det \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} + \det \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = 1 + 1 = 0$$

So no matrix of size 2×2 can be corrected. Obviously a matrix of size one can be corrected though. The entry is simply equal to the parity bit.

(f) (SEE ATTACHED .txt FOR SOLUTION TO (F)(i-iii), IT WAS DONE COMPUTATIONALLY)

(g) (i) Is $[101, 111]$ a single insertion correcting code? Prove your claim. Is $[000, 101]$ a single insertion correcting code? Prove your claim.

Proof. Using pt (ii) of this section, and the preceding part, $[101, 111]$ is not a single insertion correcting code since it is not single deletion correcting, and $[000, 101]$ is single insertion correcting because it is single deletion correcting. □

(ii) Prove that a code is single insertion correcting if and only it is single deletion correcting.

Proof. Let $\{c_i\}$ be a code of length l. Let D be the set of all deletions d_k of the k-th entry if c_i, and let I be the set of all insertions $i_k(b)$ of a bit b in the k-th entry of c_i. A code is single deletion correcting if $D(c_i) \cap D(c_j) = \emptyset$ for all $i \neq j$. Likewise, a code is single insertion correcting if $I(c_i) \cap I(c_j) = \emptyset$ for all $i \neq j$. Suppose a code is not single deletion correcting. Then there exists deletions d_a, d_b such that $d_a(c_i) = d_b(c_j)$ for $i \neq j$. Let $i_a(b_1)$ and $i_b(b_2)$ be the reverse insertions of d_a and d_b, and $i_c(b_3)$ be some arbitrary insertion. By definiton $i_a(b_1) \circ d_a(x) = i_b(b_2) \circ d_b(x) = x$, for all codewords x. It then follows that:

$$\begin{aligned} d_a(c_i) = d_b(c_j) &\implies i_c(b_3) \circ i_a(b_1) \circ d_a(c_i) = i_c(b_3) \circ i_b(b_2) \circ d_b(c_j) \\ &\implies i_c(b_3)(c_i) = i_c(b_3)(c_j) \\ &\implies I(c_i) \cap I(c_j) \neq \emptyset \end{aligned}$$

Which means that $\{c_i\}$ is not single insertion correcting, either. By contrapositive, this means single insertion correcting implies single deletion correcting. Now to go in the other direction. Suppose $\{c_i\}$ is not single insertion correcting. Then there exist $i_a(b_1), i_b(b_2) \in I$ such that $i_a(b_1)(c_i) = i_b(b_2)(c_j)$. Let $d_a, d_b \in D$ be the inverses of $i_a(b_1)$ and $i_b(b_2)$, and let $d_c \in D$ be some arbitrary deletion. It then follows that:

$$\begin{aligned} i_a(b_1)(c_i) = i_b(b_2)(c_j) &\implies d_c \circ d_a \circ i_a(b_1)(c_i) = d_c \circ d_b \circ i_b(b_2)(c_j) \\ &\implies d_c(c_i) = d_c(c_j) \\ &\implies D(c_i) \cap D(c_j) \neq \emptyset \end{aligned}$$

Which means by contrapositive that single deletion correction implies single insertion correction. Now both directions have been proven, and it follows a code is single deletion correcting if and only if it is single insertion correcting. □