MAT 312/AMS 351. Notes on binary codes: linear, error-detecting and correcting, efficient.

§1. A *binary code* C of length n is a subset of the set $\mathbf{B}^n$ of all binary n-tuples $(x_1, x_2, \ldots, x_n)$ where $x_i = 0$ or 1.

The set $\mathbf{B}^n$ forms a group under componentwise addition *mod* 2. (In this way it is isomorphic to $\mathbf{Z}_2 \times \cdots \times \mathbf{Z}_2$, n times). Moreover the scalar product $\mathbf{Z}_2 \times \mathbf{B}^n \to \mathbf{B}^n$, which takes $(0, (x_1, x_2, \ldots, x_n))$ to $(0, 0, \ldots, 0)$ and $(1, (x_1, x_2, \ldots, x_n))$ to $(x_1, x_2, \ldots, x_n)$, makes $\mathbf{B}^n$ into a $\mathbf{Z}^2$-vector space; the operations are exactly analogous to vector addition and scalar multiplication in $\mathbf{R}^n$.

The binary code C is called linear if it is a *subgroup* of $\mathbf{B}^n$ (this is the same as requiring it to be a *subspace* of the vector space $\mathbf{B}^n$).

We define (as on p.234) the *distance* between two codewords c_1 and c_2 as the number of places in which they are different (this number can range from 0 to n). Then the minimum distance between different codewords in C measures the possibilities of C for error-detection and correction (Theorem 5.4.2); we'll call this number the *quality* of the code, and write it $Q(C)$.

If the code C is linear, then $Q(C)$ can be determined from inspection of the set of codewords: it is the smallest number of 1s (the "weight") of a non-zero codeword (Theorem 5.4.3).

§2. One way of defining a linear code (this presentation is different from the book's) is to consider a linear transformation $h : \mathbf{B}^n \to \mathbf{B}^m$ for some $m < n$ and to define $C = C_H$ as the "kernel" (or "null-space") of h; this is the set of all n-tuples $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ which h sends to the identity $\mathbf{0} \in \mathbf{B}^m$ ($\mathbf{0} = (0, \ldots, 0)$, m components). In set notation, $C_h = \{\mathbf{x} \in \mathbf{B}^n | h(\mathbf{x}) = \mathbf{0}\}$. (Note that in this context "linear transformation" means no more than the requirement $h(\mathbf{x}_1 + \mathbf{x}_2) = h(\mathbf{x}_1) + h(\mathbf{x}_2)$).

The reason for defining a linear code this way is that when we express the linear transformation h by a matrix, useful information may be determined directly from that matrix, without a detailed examination of the set of codewords.

We will follow the convention of the book by representing n-tuples as row vectors, and representing h by a matrix acting on the *right*. Thus if $n = 3$, $m = 2$, and h is the linear transformation: $h((x_1, x_2, x_3)) = (x_1 + x_3, x_1 + x_2 + x_3)$. The corresponding matrix, with respect to the standard bases in $\mathbf{B}^3$ and $\mathbf{B}^2$ would be, with our convention of right action,

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

since

$$(x_1, x_2, x_3) \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} = (x_1 + x_3, x_1 + x_2 + x_3).$$

§3. One useful type of transformation (matrix) is a *canonical parity-check matrix*. In this case, with m and n as above, H has the form of an $(n-m) \times m$ matrix A on top of an $m \times m$ identity matrix.

Example:

$$n = 6, \quad m = 3, \quad A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In this case, $(x_1, x_2, x_3, x_4, x_5, x_6)H = (x_1 + x_3 + x_4, x_1 + x_2 + x_5, x_2 + x_6)$. If we consider $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$ as a word of the code defined by $\mathbf{x}H = \mathbf{0}$, we can interpret x_4, x_5, x_6 as parity check bits: x_4 should be 1 if the number of 1s among x_1 and x_3 is odd; x_5 should be 1 if the number of 1s among x_1 andx_2 is odd; x_6 should be 1 if x_2 is 1.

§4. Error-detection and correction. Note first (compare the examples above) that if $\mathbf{e}_1 = (1, 0, \ldots, 0)$ is the first standard basis vector for $\mathbf{B}^n$, then $\mathbf{e}_1 H$ produces exactly the first row of H; similarly $\mathbf{e}_2 H$ is the 2nd row of H, etc. Now the collection $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ comprises *all* the words in $\mathbf{B}^n$ with exactly one "1". Consider the code C defined by $\mathbf{x}H = \mathbf{0}$. If H has no non-zero rows, then none of $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ can satisfy that equation. Consequently all the nonzero words of C have at least two "1"s. This argument proves:

Proposition 1. If the matrix H has no row with all zeros, then the code defined by $\mathbf{x}H = \mathbf{0}$ can be used for single-error detection.

Example: The code C defined by the 6×3 matrix H above can be listed by assigning arbitrary values to the bits numbered 1, 2, 3 (we can think of these as information bits; then the values of bits 4, 5, 6 are determined as explained above. There will therefore be eight words in C; it is convenient to list the information parts using the binary numbers for 0 to 7, and then compute the check bits.

word no.	in binary	complete word
0	000	000000
1	001	001100
2	010	010011
3	011	011111
4	100	100110
5	101	101010
6	110	110101
7	111	111001

.

Proposition 2. If in the matrix H no row is zero and no two rows are equal, then the code defined by $\mathbf{x}H = \mathbf{0}$ can be used for single error correction.

Proof: We need to show that every nonzero code-word has at least three "1"s. We already know that since H has no zero row there cannot be a codeword with exactly one "1". On the other hand, a codeword with exactly two "1"s would be of the form $\mathbf{e}_i + \mathbf{e}_j$, with $i \neq j$. (For example, $(010100) = \mathbf{e}_2 + \mathbf{e}_4$). Applying H to such a word would give the sum of the like-numbered rows. (For example, with H above, $(010100)H = (\mathbf{e}_2 + \mathbf{e}_4)H = (011) + (100) = (111)$). The product with H can only come out to be zero of those two rows add up to zero, i.e. if they are identical. So if no two rows of H are equal, then no word $\mathbf{x}$ satisfying $\mathbf{x}H = \mathbf{0}$ can have exactly two "1"s. Since exactly one "1" has been excluded, a nonzero word must have at least three "1"s. Q.E.D.

§5. Efficiency. We would like to maximize the ratio of information bits to check bits and still have a code admitting single error correction. Suppose we have r ckeck bits; we can suppose our matrix H is in canonical parity-check matrix form, so the bottom r rows are $\mathbf{e}_1, \ldots, \mathbf{e}_r$. There are 2^r possible length r binary numbers, running from $(0, 0, \ldots, 0)$ to $(1, 1, \ldots 1)$. As extra rows in our matrix we must exclude $(0, 0, \ldots, 0)$ as well as the rows $\mathbf{e}_1, \ldots, \mathbf{e}_r$ we used at the bottom. This leaves $2^r - 1 - r$ possiblities; each one corresponds to a possible information bit. To maximize efficiency, we use them all. For example,

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

has 3 check-bits and $2^3 - 1 - 3 = 4$ information bits. Such a code is called a *perfect code*; it can be shown to be the most efficient way of encoding 2^4 symbols with single error detection. Similarly

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

has 4 check-bits and $2^4 - 1 - 4 = 11$ information bits; it is also a perfect code; the most efficient way of encoding 2^{11} symbols with single error detection.

§6. Hamming codes. Suppose $\mathbf{x}$ is a codeword in the perfect code C defined as above by a matrix H. A transmission error in the i-th position means that a 0 has been changed to a 1 or vice-versa; in either case, the transmitted word is $\mathbf{x}+\mathbf{e}_i$. Applying H to the transmitted word gives $H(\mathbf{x})+H(\mathbf{e}_i) = \mathbf{0}+H(\mathbf{e}_i) =$ the ith row of H. In the matrix defining a perfect code with r check bits, each binary number between 1 and r appears as a row. If the rows of H are rearranged so that the ith row is exactly the binary number i, and that new matrix is used to define the code, then the result of applying H to a transmitted word will be either $\mathbf{0}$ (if there was no error) or *the binary number of the bit where the error occurred*.

Example (r=3).

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

Here the check-bits are in position 1, 2, 4. The 16 words of the code can be efficiently generated by using (0000) up to (1111) for the information bits x_3, x_5, x_6, x_7 and adjusting the check bits accordingly: $x_1 = x_3 + x_5 + x_7$, $x_2 = x_3 + x_6 + x_7$, $x_4 = x_5 + x_6 + x_7$.

word no.	in binary	complete word
0	0000	*00*0*0*000
1	0001	*11*0*1*001
2	0010	*01*0*1*010
3	0011	*10*0*0*011
4	0100	*10*0*1*100
5	0101	*01*0*0*101
6	0110	*11*0*0*110
7	0111	*00*0*1*111
8	1000	*11*1*0*000
9	1001	*00*1*1*001
10	1010	*10*1*1*010
11	1011	*01*1*0*011
12	1100	*01*1*1*100
13	1101	*10*1*0*101
14	1110	*00*1*0*110
15	1111	*11*1*1*111

.

(Here the checkbits are shown in *italic*). This is a *Hamming code.*

Suppose that word number 6, $\mathbf{x} = (1100110)$, was transmitted with an error in bit 5, so as $\mathbf{x}' = (1100010)$. Applying H to the transmitted word gives $\mathbf{x}'H = (101)$, signalling an error in position 5. The word can then be corrected by adding (0000100) to $\mathbf{x}'$. Thus the Hamming code doesn't just allow a single error to be corrected; it shows you immediately how to do it.

Exercises:

1. Consider the code C_H defined by $\mathbf{x}H = 0$ for this matrix H:

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

List the eight codewords of C_H. (I.e., give all the solutions of $(x_1, x_2, x_3, x_4, x_5, x_6)H = 0$). Give an example of a single error, in the transmission of one of the codewords of C_H, which cannot be detected.

2. Consider the code C_H defined by $\mathbf{x}H = 0$ for this matrix H:

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

List the eight codewords of C_H. (I.e., give all the solutions of $(x_1, x_2, x_3, x_4, x_5, x_6)H = 0$). Give an example of a single error, in the transmission of one of the codewords of C_H, that cannot be corrected.

3. Suppose the Hamming code of §6 is used to transmit text, by assigning A to word 0, B to word 1, ..., P to word 15, following alphabetical order. An 8-letter message is encoded and transmitted. What is received is

   0101100 0010110 0010100 1010101 1001001 1001101 0010000 0111101.

   Assuming that each codeword has been transmitted with at most a single error, reconstruct the original message.

Anthony Phillips
April 10, 2012