Codes and Fields

Part 1
Coding for the Binary Symmetric Channel

Oliver Pretzel

Ref's Oliver Pretzel:
Error-Correcting Codes
and Finite Fields,
Clarendon Press, Oxford
1992

Codes and Fields

Part 1
Coding for the Binary Symmetric Channel

Oliver Pretzel

Ref: Oliver Pretzel;
Error-Correcting Codes
and Finite Fields,
Clarendon Press, Oxford
1992

Chapter 1.1

Introduction

1. The situation we wish to model is as follows. Information is sent via a *channel* which is prone to errors. The distorted information is processed at the receiving end to restore as nearly as possible the original message.

EXAMPLES. Telecommunication, satellite pictures, cash dispensers, computer file transfer, compact disc players, talking in a noisy pub.

There are two fundamental approaches for dealing with errors:

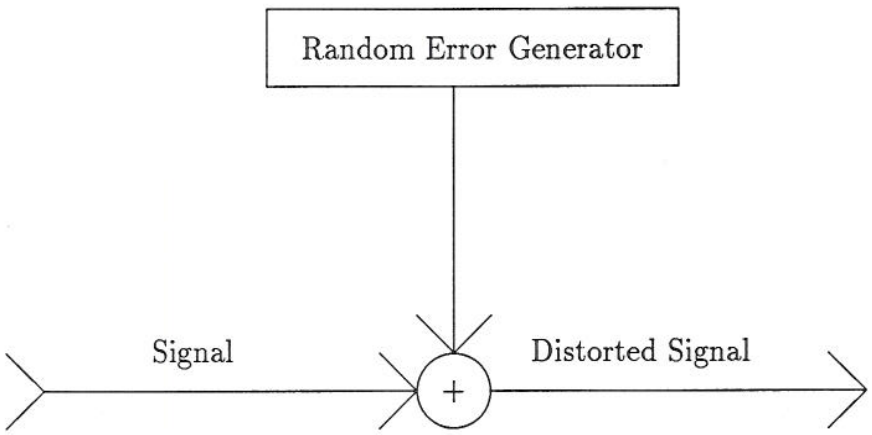
(a) *Enhancement.* This is statistical guessing to improve image, at the receiving end only.

(b) *Planned Redundancy.* This artificially adds redundancy before transmission to help remove errors at the receiving end.

2. Our course deals with the second topic. We assume the message is transmitted in *binary* form.

$$\mathbb{B} = \{0, 1\} \quad +, \quad \times.$$

We shall further assume that errors are *random* and independent of input (*symmetric*). We shall always use p for the probability of an error occurring in a single bit ($p < 1/2$). We treat *erasures* as though they were full errors.



3. We shall deal with *block codes*. Idea:

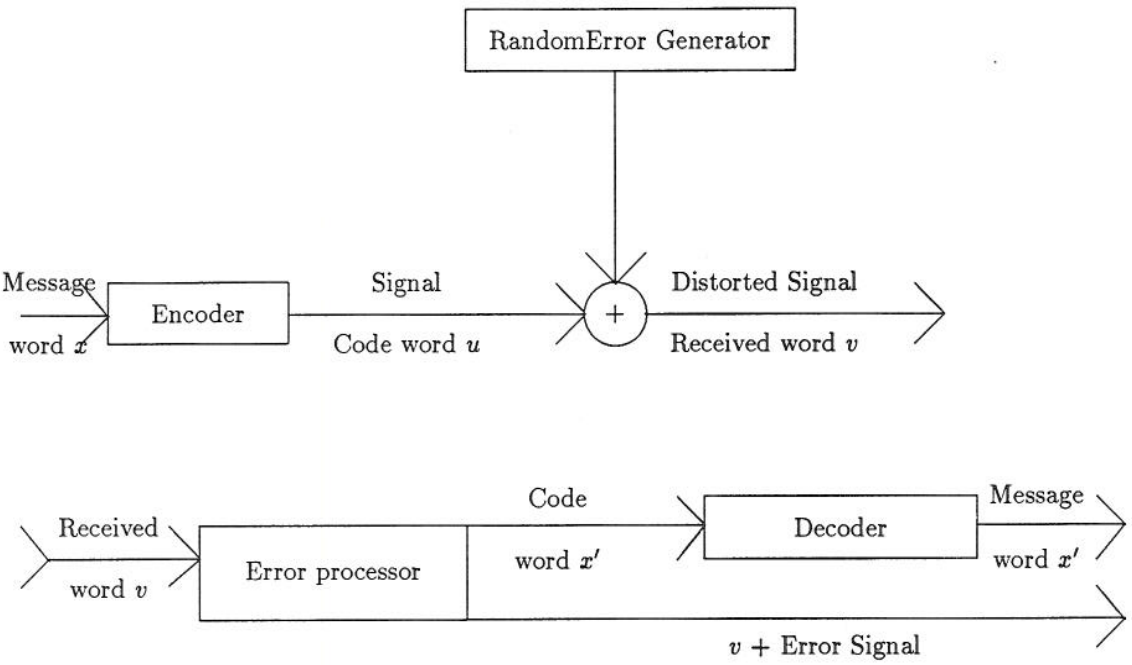
- Divide message into blocks of length m.
- Encode into blocks of length $n > m$ to add redundancy.
- Treat each block of length n as a *word* or vector in $\mathbb{B}^n$.
- Eg. $m = 4, n = 7$. Then 0010 may encode as $w = 0010101$.

DEFINITION. The *rate* of the code is m/n. It is a measure of the redundancy added to the signal. Errors produce *patterns* or *error words* which are added to code word termwise. To distinguish between code words and words which may contain errors, the latter are called *received words*.

Eg. if $w = 0010101$, the error word $e = 0100100$ produces the distorted word 0110001.

Thus error pattern = received word − transmitted word.

We can now complete our picture of the process. At the receiving end we have an *error processor* which either converts the received word back into a code word (with a high probability of correctness), or passes it through with an error message. These two possibilities are called *error correction* and *error detection*. Clearly, the first requires more resources.



The error processor and decoder are often lumped together and just called the decoder. It is important to recognize that to determine a coding system we need

(1) a set of codewords (the code),

(2) a bijective map of message blocks onto codewords (for encoding and decoding), and

(3) and error processing unit.

The performance of the code can be affected by the choice of any of the three.

4. In order to assess the performance of our codes we need some very elementary probability theory.

DEFINITION. The *weight* of a word or error pattern is the number of non-zero entries.

PROPOSITION. *The probability of a particular error pattern of weight k occurring in a received word of length n is $p^k(1-p)^{n-k}$.*

The probability of some *error pattern of weight k occurring is $\binom{n}{k}p^k(1-p)^{n-k}$, where $\binom{n}{k}$ is the binomial coefficient 'n choose k'.*

5. Examples of Codes

We assume we have to transmit a message of 10000 bits along a channel with error probability $p = 1/1000$. With no coding the probability of successful transmission is $.999^{10000} \approx .000045$.

(a) **The (11,10) parity check code.** To each message block of 10 bits add a parity check so that the no. of 1's is even.

Rate: 10/11.

This can detect up to one error in each transmitted word of length 11. It cannot correct any errors.

Probability of no error in word:	$(.999)^{11} \approx .989055.$
Probablity of one error in word:	$(.999)^{10} \cdot 11/1000 \approx .\underline{010890}.$
	$\approx .999945.$
Probability of correct transmission:	$(.989055)^{1000} \approx .000016.$
Probability of no undetected error:	$(.999945)^{1000} \approx .946.$

(b) **Triple repetition (3,1) code.** Repeat each bit three times, $1 \to 111$, $0 \to 000$.

Rate: 1/3.

(a) Error Detecting. Can detect two errors in block of three. So only undetectable error pattern is 111 which has probability 10^{-9}.

Probability of correct transmission:	$(.999)^{30000} \approx 10^{-13}.$
Probability of no undetected error:	$(1 - 10^{-9})^{10000} \approx .99999.$

(b) Error correcting. One error is more likely in a block than two. So can decode by majority logic.

Probability of no error in word:	$(.999)^{3} \approx .997003.$
Probability of one error in word:	$(.999)^{(2\cdot 3)/1000} \approx .\underline{002994}.$
	$\approx .999997.$
Hence probability of correct transmission:	$(.999997)^{10000} \approx .97.$

But this is also the probability of no undetected error.

(c) **Triple parity check (6,3) code.** Divide message into blocks of three (a, b, c). Encode as (a, b, c, x, y, z) with $x = a + b, \quad y = a + c, \quad z = b + c.$

This will correct one error in block of six. If error is in a,b or c the two equations involving it will fail. If error is in x,y, or z the single equation involving it will fail. It will also detect the three error patterns $(1, 0, 0, 0, 0, 1)$, $(0, 1, 0, 0, 1, 0)$, and $(0, 0, 1, 1, 0, 0)$ as these cause all three equations to fail.

Rate: 1/2.

Probability of no error in word:	$(.999)^6$	$\approx .994015.$
Probability of one error in word:	$6(.999)^5/1000$	$\approx \underline{.005970}.$
		$\approx .999985.$
Probability of 3 error patterns above:	$3(.999)^4/10^6$	$\approx .000003.$
Probability of correct transmission:	$(.999985)^{10000/3}$	$\approx .951.$
Probability of no undetected error:	$(.999987)^{10000/3}$	$\approx .957.$

6. SHANNON'S THEOREM. *There exist codes with probability of correct transmission arbitrarily close to 1, and rate arbitrarily close to the channel capacity $C(p)$:*

$$C(p) = 1 + p.\log_2(p) + (1-p)\log_2(1-p).$$

Comment. $C(.5) = 0$, $C(.999) = .9886$. Shannon's Theorem only applies to very long messages. He uses random codes with very large block length. These are practically impossible to decode efficiently. In practice codes are highly structured to achieve efficient decoding. They are much better than our examples, but nowhere near Shannon's bound.

Chapter 1.2

Codes, Weight and Distance

Although encoding, error processing and decoding are integral parts of any coding scheme, we shall use the word 'code' to speak of the set of codewords.

1. DEFINITION. Let A be a finite alphabet. We denote the set of all *words* or sequences of length n with entries in A by A^n. A *block code* C over A of *block length* n is a subset of A^n with at least two elements. If $A = \mathbb{B}$, the binary field 0,1, the code is called *binary*.

EXAMPLES. The examples of Chapter 1 are binary codes of lengths 11, 3 and 6. The Triple Repetition Code has two codewords, the Triple Check Code has 8. Write them all down. How many codewords does the (11,10) Parity Check Code have?

Remark. We shall later discover a need for codes with an enlarged alphabet A, but we shall always require that the alphabet will allow all four arithmetic operations and thus be a *field*. So termwise addition and multiplication by elements of A will *always* make A^n into a vector space. See the appendix for a list of the laws of arithmetic and vector spaces.

2. DEFINITION. The *distance* $d(u, v)$ between two code words u and v is the number of entries in which they differ. The *weight* $w(u)$ of a codeword u is the number of non-zero entries in u.

PROPOSITION. *For binary codes (and whenever A is a field),* $d(u, v) = w(u - v)$.

3. DEFINITION. The *minimum distance* $d(C)$ of a code C is the smallest distance between two distinct codewords of C.

PROPOSITION.

(a) *The code C can detect up to k errors in a received word iff $d(C) \geq k+1$.*

(b) *C can correct up to k errors in a received word iff $d(C) \geq 2k+1$.*

(c) *C can correct up to k errors and at the same time detect up to $(k+j)$ errors iff $d(C) \geq 2k+j+1$.*

Proof. We shall prove (a) and (b). (c) is set as an exercise on Sheet 1.

(a) The code will detect r errors in a received word iff any r changes in a codeword produce a non-codeword.

(b) For the code to be able to correct k errors it is necessary and sufficient that making k changes to a codeword u produces a word that cannot be obtained from any other codeword v by $\leq k$ changes.

Chapter 1.3

Linear Codes

Sometimes these are called group codes.

1. DEFINITION. A code is called *linear* if

(a) it is a subspace of $\mathbb{B}^n$, that is the termwise sum of two code words is a codeword (for larger fields we shall later also require that any scalar multiple of a codeword is a codeword), and

(b) the encoding algorithm is given by multiplying a message word (written as a column) by an $n \times m$ matrix.

This matrix is called a *generator matrix* for the code. The columns of the generator matrix are the codewords that encode the messages $(1,0,\ldots,0)$, $(0,1,\ldots,0)$ etc.The matrix is in *standard form* if it is of the form $(I,A)^\top$, where I is the $m \times m$ identity matrix. Standard form means that the message word forms the first m bits of the codeword.

For a linear code C, the length m of the message words is the *dimension* or *rank* of C, and the length n of the codewords is the *block length* of C. Such a code is called an (n,m)-*code.*

EXAMPLES.

	Standard	Non-Standard
A.	The(11,10)-Parity Check Code:	
	$\begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & & \ldots & \\ 0 & 0 & \ldots & 1 \\ 1 & 1 & \ldots & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ 0 & 1 & 1 & \ldots & 0 \\ & & & \ldots & \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}$

B. The Triple Repeat Code:

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$ none

C. The Triple Check Code:

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

In all cases we gave standard encoding in Chapter 1.

2. PROPOSITION. *For linear codes the minimum distance is equal to the minimum weight of a non-zero codeword.*

3. DEFINITION. A *check matrix* for a linear code C is an $k \times n$ matrix H with the property that for a (column) vector v in $\mathbb{B}^n$, $Hv = \underline{0}$ if and only if $v \in C$. The number k is arbitrary, but we shall show that the smallest possible value for k is $n - m$.

H is said to be in *standard form* if it has the form (D, J), where J is the $(n-m) \times (n-m)$ identity matrix. Standard form (with standard encoding) means that the non-message bits (the check bits) are each given as combinations of the message bits.

Nota Bene. Multiplication by H does not decode. It is used to check for and locate errors.

EXAMPLES.

Standard Non-Standard

A. The(11,10)-Parity Check Code:

$$\begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix}$$ none

B. The Triple Repeat Code:

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

C. The Triple Check Code:

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

4. PROPOSITION. *If e is the error pattern of a received word v of a binary code C with check matrix H, then $He = Hv =$ the sum of the columns of H corresponding to error locations in v.*

Proof. $v = u + e$, where u was the transmitted codeword. So $Hv = Hu + He = \underline{0} + He$.

COROLLARY. *A binary linear code can correct all single errors iff it has a check matrix with distinct non-zero rows.*

DEFINITION. Hv is called the *syndrome* of the received word v.

Remark. If the alphabet is a field larger than $\mathbb{B}$, then we have to take into account multiplication by scalars, but a similar proposition and corollary still hold.

EXAMPLES. Parity Check cannot correct single errors, the other two codes can.

5. PROPOSITION.

(a) *The rank of a check matrix for an (n, m)-code is $n - m$. In particular, H has at least $n - m$ rows.*

(b) *If H is a check matrix for the code C, and K is a matrix obtained from H by adding rows that are linear combinations of the rows of H, then K is a check matrix for the same code.*

Proof.

(a) This is a direct consequence of the Rank and Nullity Theorem of linear algebra.

(b) Obviously, $Kv = \underline{0}$ implies $Hv = \underline{0}$, because the entries of Hv are the first entries of Kv. But if $Hv = \underline{0}$, then $Kv = \underline{0}$, because all the entries of Kv are linear combinations of those of Hv.

6. THEOREM.

(a) *A linear binary code has unique generator and check matrices in standard form if it has either.*

(b) *If these are* $G = \begin{pmatrix} I \\ A \end{pmatrix}$ *and* $H = (B, J)$, *then* $A = B$.

Proof. We assume the code C has block length n and dimension m. So if they exist, G is $n \times m$ and H is $(n-m) \times n$.

If C has generator and check matrices G and H as above, then $HG = \underline{0}$, because the columns of G are codewords. Multiplying out we get $BI + JA = \underline{0}$. But $BI = B$ and $JA = A$. Hence G is determined by H and vice-versa. So if the code has standard generator and check matrices, they are unique.

To complete the proof we must show that given $G = \binom{I}{A}$ and $H = (A, J)$, then $v = Gu$ for some u iff $Hv = \underline{0}$ As $HG = \underline{0}$ it is immediate that for $v = Gu$ it follows that $Hv = GHu = \underline{0}u = \underline{0}$.

Conversely, let $Hv = \underline{0}$ and split v into (u, w) where u consists of the first m bits of v. Then $Hv = \underline{0} \iff Au + Jw = Au + w = \underline{0}$. So $w = Au$. Hence

$$v = \begin{pmatrix} u \\ w \end{pmatrix} = \begin{pmatrix} u \\ Au \end{pmatrix} = \begin{pmatrix} I \\ A \end{pmatrix} u = Gu.$$

Chapter 1.4

Decoding by Cosets

1. Let C be a linear binary (n, m)-code. We can tabulate the elements of $\mathbb{B}^n$ as follows.
First row: elements of C starting with $\underline{0}$.
Later rows: start with a word x of lowest weight that has not yet appeared. Under each codeword u put $u + x$.

EXAMPLE. Triple Check Code:

000000	100110	010101	001011	111000	011110	101101	110011
100000	<u>000110</u>	110101	101011	<u>011000</u>	111110	001101	010011
010000	110110	<u>000101</u>	011011	<u>101000</u>	001110	111101	100011
001000	101110	011101	<u>000011</u>	<u>110000</u>	010110	100101	111011
000100	<u>100010</u>	<u>010001</u>	001111	111100	011010	101001	110111
000010	<u>100100</u>	010111	<u>001001</u>	111010	011100	101111	110001
000001	100111	<u>010100</u>	<u>001010</u>	111001	011111	101100	110010
<u>100001</u>	000111	110100	101010	011001	111111	<u>001100</u>	<u>010010</u>

We have underlined words of weight 2. There is a choice for the order of elements in the last row.

DEFINITION. This table is called a *decoding table* or *coset table* for C.

Two elements of $\mathbb{B}^n$ lie in the same row of the table iff their difference lies in C. Thus no element appears twice. The number of rows is 2^{n-m}. The elements of a row are called a *coset* of C. The first element of the row is called the *coset leader.* A received word is decoded as the codeword heading its column.

2. PROPOSITION.
(a) An error pattern is corrected iff it is a coset leader.
(b) If a received word v is decoded as the codeword u, then for any codeword w: $d(w, v) \geq d(u, v)$.

Proof.
(a) Going to the head of the column is the same as subtracting the coset leader.
(b) Going to any codeword is the same as subtracting some member of the coset of v.

Remarks.
(a) If there is a choice for coset leader, then this will affect the error patterns that are corrected.
(b) Decoding by cosets is nearest codeword decoding and hence maximum likelihood decoding.

COROLLARY. *The code corrects all error patterns of weight up to k if they all occur as coset leaders. Then they are the unique minimal weight elements of their cosets.*

3. PROPOSITION. *If the code C has check matrix H, then two words u and v lie in the same coset of C iff $Hu = Hv$, that is iff they have the same syndrome.*

Proof. u and v lie in the same coset iff $u - v \in C$.

COROLLARY. *The table can be abbreviated to two columns: one column of syndromes and one of coset leaders.*

EXAMPLE. Triple Check Code:

Syndrome	Coset Leader
000	000000
110	100000
101	010000
011	001000
100	000100
010	000010
001	000001
111	100001

Note. The last line is non-unique.

With this abbreviated table 011101 gives syndrome 011 so is decoded by adding (subtracting) 001000, giving 010101. But 001100 is decoded as 101101 while 000000 would have been an equally likely original.

Chapter 1.5

Hamming Codes

1. DEFINITION. The *binary Hamming Code* $\mathrm{Ham}(k)$ has as its check matrix H_k the matrix whose columns are all non-zero binary words of length k.

EXAMPLE. H_3:

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

Remark. Permuting the columns of H_k corresponds to permuting the entries of a word. So there is essentially only one $\mathrm{Ham}(k)$ for each k.

2. PROPOSITION.

(a) $\mathrm{Ham}(k)$ has block length $n = 2^k - 1$ and dimension $m = 2^k - k - 1$.

(b) To every word $v \in \mathbb{B}^n$ there is a unique word $u \in \mathrm{Ham}(k)$ with $d(u, v) \leq 1$. Hence the minimum distance of $\mathrm{Ham}(k)$ is 3.

Proof.

(a) n is the number of non-zero binary words of length k. m is the number of rows of H_k which can be chosen to be in standard form.

(b) The syndrome of v, $H_k v$ is $\underline{0}$ iff $v \in \mathrm{Ham}(k)$ and otherwise it is a unique column of H_k. The corresponding entry is the unique one that can be changed to give a codeword.

3. DEFINITION. An (n, m)-code C is called *r-perfect* if to every vector $v \in \mathbb{B}^n$ there is a unique codeword u with $d(u, v) \leq r$. Perfect codes have maximum possible density among the codes that can correct error patterns of weight up to r (see proposition below) but they are very rare. The only binary perfect codes are the following:

1. The $(2r + 1, 1)$-repetition code is r-perfect.
2. The Hamming codes are 1-perfect.
3. There is a single further perfect code: the 3-perfect (23,12) Golay code.

PROPOSITION. *If there is an r-perfect (n, m)-code then no (n, m')-code with $m' > m$ can correct all error patterns of weight up to r.*

Proof. The *r-ball* $D_r(u) = D$ with centre u consists of all vectors $v \in \mathbb{B}^n$ with $d(u, v) \leq r$. The statement that C is r-perfect says that the r-balls centred on the codewords are disjoint and cover $\mathbb{B}^n$.

Now the number $|D|$ of elements of an r-ball D is independent of its centre. So $2^n = 2^m|D|$. Thus if C' is an (n, m')-code the r-balls centred on the words of C' cannot be disjoint.

Remark. We can calculate $|D|$:

$$|D| = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{r}$$

If we insert this value in the formula $2^n = 2^m|D|$ this places severe restrictions on m, n, and r. Thus the Golay code could not exist but for the fact that $1 + 23 + 23.22/2 + 23.22.21/6 = 1 + 23 + 253 + 1771 = 2048 = 2^{11}$.

4. EXAMPLE. Ham(3): Generator Matrix:

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

So check symbols for the message *abcd* are $x = a + c + d$, $y = a + b + d$, and $z = a + b + c$. For a 10000 bit message on a channel with error probability .001 the probability of correct transmission is:

$$(.999^7 + 7(.001)(.999)^6)^{10000/4} \approx .949,$$

almost as good as Triple Check. But the rate is 4/7 so the code is somewhat more efficient. Triple check requires transmission of 20000 bits while Ham(3) requires 175000.

The Laws of Arithmetic

A. Laws of Addition

A1. $(a+b)+c=a+(b+c)$. [Associative Law]

A2. There exists 0, s.t.for all a, $0+a=a+0=a$ [Zero]

A3. $a+b=b+a$. [Commutative Law]

A4. For every a, there exists $-a$ s.t. $a+(-a)=0$. [Negatives]

D. Mixed Laws

D1. $a(b+c)=ab+ac$.

D2. $(a+b)c=ac+bc$. [Distributive Laws]

M. Laws of Multiplication

M1. $(ab)c=a(bc)$ [Associative Law]

M2. There exists $1\neq 0$, s.t.for all a, $1a=a1=a$. [Identity]

\- - - - - - - - - - - - - - - - RING - - - - - - - - - - - - - - - -

M3. $ab=ba$. [Commutative Law]

\- - - - - - - - - - - - COMMUTATIVE RING - - - - - - - - - - - -

M4. $ab=0\Rightarrow a=0$ or $b=0$. [Cancellation Law]

\- - - - - - - - - - - - - - - - DOMAIN - - - - - - - - - - - - - - - -

M5. For every $a\neq 0$, there exists a^{-1} s.t. $aa^{-1}=a^{-1}a=1$. [Inverses]

\- - - - - - - - - - - - - - - - FIELD - - - - - - - - - - - - - - - -

Examples

Ring:	2×2 real matrices.
Commutative Ring:	diagonal 2×2 real matrices.
Domain:	integers, polynomials with real coefficients.
Field:	real numbers, complex numbers,
	binary field $\mathbb{B}$, $\{0,1\}$ with $1+1=0$,
	ternary field $\mathbb{T}$, $\{0,1,-1\}$ with $1+1=-1$.

The later structures are special cases of earlier ones. So, for example, any field is a domain, and any domain is a ring, but we usually use the strongest name we know to be valid.

Vector Spaces

It is possible to give formal laws describing vector spaces, but for our purposes a vector space is a subset of F^n.

DEFINITION. Let F be a field. Then F^n denotes the set of all n–tuples $\underline{x} = (x_1, \ldots, x_n)$ with $x_1, \ldots, x_n \in F$. Elements of F^n are called *vectors* and elements of F *scalars.* Addition and multiplication by scalars are defined as follows.

$$(x_1, \ldots, x_n) + (y_1, \ldots, y_n) = (x_1 + y_1, \ldots, x_n + y_n)$$
$$a(x_1, \ldots, x_n) = (ax_1, \ldots, ax_n)$$

A *vector space* is a subset V of F^n such that for $\underline{x}, \underline{y} \in V$ and $a \in F$, $\underline{x} + \underline{y} \in V$ and $a\underline{x} \in V$ (V is *closed* under addition and multiplication by scalars.

If we write the elements of F^n as *rows* and A is an $m \times n$–matrix with entries in F. Then the function $\underline{x} \to A\underline{x}$ defines a mapping from F^m to F^n.

DEFINITION. Such a map is called *linear* The set of vectors $\underline{y} \in F^n$ that occur as $\underline{y} = A\underline{x}$ for some $\underline{x} \in F^m$ is called the *image* of A ($\operatorname{im} A$). The set of vectors $\underline{x} \in F^m$ such that $A\underline{x} = \underline{0}$ is called the *kernel* of A ($\ker A$).

PROPOSITION. *The image and kernel of a matrix are vector spaces. To every vector space V in F^n there exist matrices A and B such that $V = \operatorname{im} A$ and $V = \ker B$.*

DEFINITION. Let $V \subseteq F^n$ be a vector space, then the smallest number m such that there exists an $m \times n$–matrix A with V as its image is called the *dimension* of V ($\dim(V)$).

PROPOSITION. *The dimension of F^n is n. If V is a vector space in F^n, then its dimension is $\leq= n$.*

THEOREM Rank and Nullity. *If A is an $m \times n$ matrix then*

$$\dim(\ker A) + \dim(\operatorname{im} A) = n.$$

Coding Theory
Exercise Sheet 1

All codes are linear. n, m, d denote the block length, dimension and minmum distance.

1. Suppose we wish to correct up to k errors and detect the presence of up to $k + r$ errors. Show that we must use a code with minimum distance $2k + r + 1$.

2. Define a code by adding an overall parity check p to the triple check code. So abc is encoded as $abcxyzp$ with

$$a + b + x = a + c + y = b + c + z = a + b + c + x + y + z + p = 0.$$

What are the parameters of this code, including minimum distance? Write down generator and check matrices for the code.

3. Write down a generator and check matrices for the code Ham(4).

4. Compare the performances of the codes of Q2 and Q3 with those of Chapter 1.

5. Just as in Q2 we can extend any Hamming code Ham(k) by adding an overall parity check p to each codeword. The extended code will be denoted by Ham$'(k)$. So Ham$'(3)$ is an (8,4) code. Show that this code can correct single errors and detect double errors. Is that true for all extended Hamming codes?

6. Calculate the error probabilities for Ham(3), Ham$'(3)$, Ham(4) and Ham$'(4)$ for a message of 10000 bits on a channel with error probability .001. Comment on the significance of your result.

Coding Theory
Exercise Sheet 2

1. Let C be a (n, m, d)–code with at least two non-zero codewords. Show that for any $i = 1, \ldots, n$, there is a non-zero code word in C with 0 in the i–th. position. Fix i and define the *shortened code* C' as follows: there is a code word u' of C' for every code word u of C that has 0 in the i–th. position. The word u' is obtained from u by deleting the 0 that occurs in the i–th. position. These are all the code words of C'. C' obviously has block length $n' = n - 1$. Show that C' is linear. Find its dimension m', and minimum distance d' in terms of the data for C. How are the check matrices of C and C' related?

Show that if the bits of the code words of the triple check code are suitably permuted, it can be obtained from Ham(3) by shortening.

2. With C as above, but now assuming that $d \geq 2$. We define the Punctured Code C'' as follows: delete the i–th symbol from all code words of C. Show that C'' is linear. Find the dimension m'', and minmum distance of C and C''. How are the generator matrices of C and C'' related? How are C' and C'' related.

3. Suppose not all the code words of the binary linear code C have even weight. $\hat{C}$ is obtained from C by adding an overall parity check. $\hat{C}'$ and $\hat{C}''$ are obtained from $\hat{C}$ by shortening and puncturing it in the last position, respectvely. What is the relation between $\hat{C}'$, $\hat{C}''$ and C?

Coding Theory
Exercise Sheet 3

1. THE SINGLETON BOUND. Let C be an (n, m, d)–code. Show that $d \leq n - m + 1$. HINT: *puncture C $d-1$ times. What are the block length and dimension of the resulting code?*

Which Hamming Codes (if any) meet this bound?

This question shows that the relation between the minimum distance and the number of check bits is not an easy topic. Do not assume that the number of check bits of a code is simply related to its error correcting capability.

2. Let C be a linear code of block length n greater than the block length $2k-1$ of Ham(k). Show that if C can correct single errors, then the dimension m of C satisfies $n - m > k$.

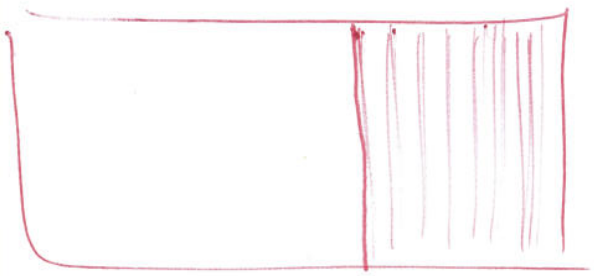
This means that C uses more check bits than Ham(k).

In particular show that a linear code of block length greater than 7 that can correct single errors must have at least 4 check bits.

You can do the last part directly, without refering to the Hamming code. Actually the assumption that C is linear is not necessary, but the question would have to be rephrased for non-linear codes.

3. Let C be an (n, m)–linear code with generator matrix G and check matrix H in standard form. Define the dual code $C^\top$ to have check matrix $G^\top$. Describe the dual codes of the parity check code, triple repetition code, and triple check code of section 1. Find a generator matrix for $C^\top$ in general. What are the block length, and dimension, of $C^\top$? Think about the problem of determining the minimum distance of $C^\top$ (This is hard. It can be shown that if C meeets the bound of Q1, then so does $C^\top$).

$\frac{n-d+1}{m-d+1}$



Coding Theory
Exercise Sheet 4

1. The ternary field T has elements $\{0, 1, -1\}$. Addition is ordinary addition except $1+1=-1$ and $-1+-1=+1$. Multiplication is ordinary multiplication. Let C be the ternary code with check matrix:

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

What are the dimension and block length of the code? Write down a standard form generator matrix for this code. How are standard generator and check matrices related for ternary codes?

Show that this code can correct all single errors in a codeword (so it can locate them and determine their sign). What is the minimum distance of the code? Give a necessary and sufficient condition on the check matrix of a ternary code for the code to be able to correct all single errors.

2.$^{+}$ Let C be a linear code with check matrix H. Show that C can correct all error patterns of weight $\leq k$ iff all sets of $2k$ columns of H are linearly independent (ie. the only linear combination yielding 0 is the one with all coefficients 0).

3.* Try to extend the Hamming check matrix H_4 to produce a check matrix of a double error correcting code.

The main purpose of this exercise is to convince you that it is hard to do. Remember the additional columns must not be linear combinations of the columns of H_4.

$^{+}$ A problem that requires a good grasp of linear algebra.

* A challenging problem that need not be solved.