Linear Algebra

18.12.2012

# Linear Algebra - Course 12

# Chapter 4. Introduction to Linear Codes

# Part I

**Coding Theory:**

Starting points:

* Shannon 1948: Information Theory
* Hamming 1950: Error-Correcting Codes

Main classes of codes:

* source coding: data compression
* channel coding: error-correcting codes

**A first example:**

*EAN-13 International Article Number*

It is a sequence of 13 digits a1, a2, ... , a13 that identifies a product. Digit a13 is a check digit that is computed as *a13 = 10 - (a1 + 3a2 + a3 + 3a4 + \_ \_ \_ + a11 + 3a12) mod 10*:

Digits are written in binary, black bars for 1, white bars for 0.

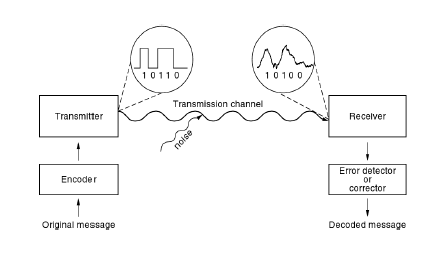
In particular:

ISBN (International Standard Book Number)

UPC (Universal Product Code) etc.

**Error-correcting (detecting) codes**

General scheme:



Diferent codes are suitable for diferent applications:

* satellite and space transmissions
* credit cards
* CD's, DVD's etc.

**The Coding Problem**

* We discuss binary *codes*. Most results may be generalized to codes over finite fields.
* We consider binary symmetric channels: the probability of 1 being changed into 0 is the same as that of 0 being changed into 1.
* We talk about (n; k)-codes:



There are 2^k possible messages, and so 2k code words.

There are 2^n possible words received.

**Aim:**

Find the right balance between k and n - k.

**Two simple codes – The (3,2) - parity check code**

The check digit is the sum modulo 2 of the message digits.

Encoding:

|  |  |
| --- | --- |
| **Message** | **Code word** |
| 00 | 000 |
| 01 | 101 |
| 10 | 110 |
| 11 | 011 |

* How many error can this code detect/correct?

Decoding:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Received words | 101 | 111 | 100 | 000 | 110 |
| Parity check | passes | fails | fails | passes | passes |
| Decoded words | 01 | - | - | 00 | 10 |

**Two simple codes – The (3,1) – repeating code**

* The two check digits repeat the message digit.
* Encoding:

|  |  |
| --- | --- |
| Message | Code word |
| 0 | 000 |
| 1 | 111 |

* How many errors can this code detect/correct?

Decoding:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Received words | 111 | 010 | 011 | 000 |
| Decode words | 1 | 0 | 1 | 0 |

**Hamming distance**

**Definition**:

The Hamming distance between 2 words of the same length is the number of positions in which they differ.

**Notation**: d(u; v).

**Example**: d(101; 100) = 1, d(110; 001) = 3, d(101; 011) = 2.

**Theorem:**

The Hamming distance is a metric on the set Z2n of words of length n, that is, the following properties hold for every u,v, w belonging to Z2n:

(1) d(u; v) = d(v; u).

(2) d(u; v) + d(v;w) \_ d(u;w).

(3) d(u; v) \_ 0 with equality if and only if u = v.

* In an (n; k)-code, the 2n received words can be thought of as placed at the vertices of an n-dimensional cube with unit sides.
* The Hamming distance between two words is the shortest distance between their corresponding vertices along the edges of the n-cube.
* The 2k code words form a subset of the 2n vertices, and the code has better error-correcting and error-detecting capabilities the farther apart these code words are.
* Cube representations of the (3; 2)-parity check and (3; 1)-repeating codes:



**Error correction/detection capabilities:**

**Theorem:**

A code detects all sets of t or fewer errors <==> the minimum Hamming distance between code words is at least t + 1.

**Theorem:**

A code is capable of correcting all sets of t or fewer errors <==> the minimum Hamming distance between code words is at least 2t +1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code | Minimum distance between words | No. of detectable errors | No. of correctable errors | Information rate |
| (n,k) - code | d | d-1 | ≤ (d-1)/2 | k/n |
| (3,2) – parity check code | 2 | 1 | 0 | 2/3 |
| (3,1) – repeating code | 3 | 2 | 1 | 1/3 |

**Polynomial Representation**

A binary n-digit word a0a1 ... an-1 may be identified with a polynomial:

a0 + a1X + ... + an-1Xn-1 belonging to Z2[X].

**Definition**:

Let p belonging to Z2[X] be of degree n - k. The polynomial code generated by p is an (n; k) -code whose code words are those polynomials of degree less than n which are divisible by p. Then the polynomial p is called the generator of the code.

* A message of length k is represented by a polynomial m belonging to Z2[X] of degree less than k.
* Since the message is stored in the right hand side of a word, the message digits are carried by the higher-order coeficients of a polynomial. So we consider m · Xn-k.

To encode the message polynomial m we first use the Division Algorithm to find unique q, r belonging to Z2[X] such that:



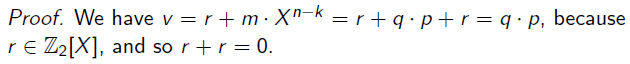
Then the code polynomial is:

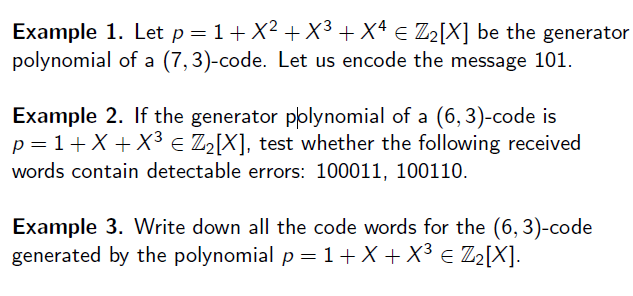


The check digits of the message are carried by r .

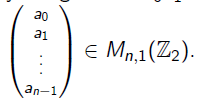
**Theorem**

With the above notation, the code polynomial v is divisible by p.





**Matrix Representation**

A binary n-digit word a0a1 . . . an-1 may be identified with a matrix:

For an (n,k) - code, we see the 2k possible messages as the elements of the vector space Z2k over Z2, and the 2n possible received words as the elements of the vector space Z2n over Z2.

**Definition**

An *encoder* is an injective function (or equivalently:

An (n,k) – code is called *linear* if the encoder is a linear map.

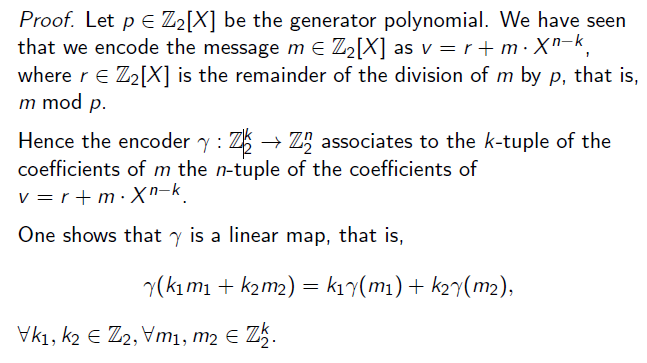
From now on we will discuss only linear codes.

An example: Reed-Solomon code, used for CD's, DVD's, Blu-ray discs etc.

**A class of linear codes**

**Theorem**

Any (n,k) – code generated by a polynomial of degree n-k is linear.



**Generator Matrix**

**Definition**

Consider a linear (n,k) – code, with encoder: 

Let E, E` be the canonical bases of the Z2 vector spaces Z2k  and Z2n  respectively.

Then the matrix G = is called the *generator matrix* of the code.

A message m encodes as .

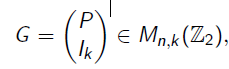
But for m we have 

Hence: A message encodes as G · m.

**Theorem**

1. The code words of the (n,k) - code are the vectors in the subspace Im of Z2n. Hence a binary (n, k) - code means a k-dimensional subspace of the vector space Zn2.
2. The columns of G form a basis of this subspace, and so a vector is a code vector if and only if it is a linear combination of the columns of G.

**Remark**. A code word contains the message digits on the last k positions. Hence the generator matrix G of an (n, k) - code is always of the form:

**Parity Check Matrix**

**Definition**

With the above notation, the matrix is called the *parity check matrix* of the code.



**Theorem**

Consider a linear (n,k) – code with the parity check matrix.

Then a received vector u belonging to Z2n is *a code vector* if and only if H · u = 0.

**Matrix Representation – Examples**:

