Transmission and Coding of Information

José Luis Ruiz

July 2018

Departament de Matemàtiques
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya
jose.luis.ruiz@upc.edu

Introduction

Mathematical Theory of Information

The mathematical theory of information begins in 1948 with Claude Shannon's seminal paper "A Mathematical Theory of Communication". For Shannon:

> *The fundamental problem of communication is that of reproducing at one point exactly or approximately a message selected at another point.*

Shannon gives a mathematical definition of information and studies how to transmit a message through a communication channel efficiently and reliably.

Error-Correcting Codes: A Brief History

In 1950, Richard Hamming, at Bell's Laboratories, constructs an error-correcting code for the first time. The Hamming code can correct up to one error.

In 1954, Reed and Muller introduced a family of error-correcting codes capable to correct an arbitrary number of errors. They have been used in Mariner missions from 1969 to 1977.

From 1958 to 1960, BCH and Reed-Solomon codes were discovered. They have been used for space missions and in CDs and DVDs since then.

Communication Channels

Simple model of communication channel

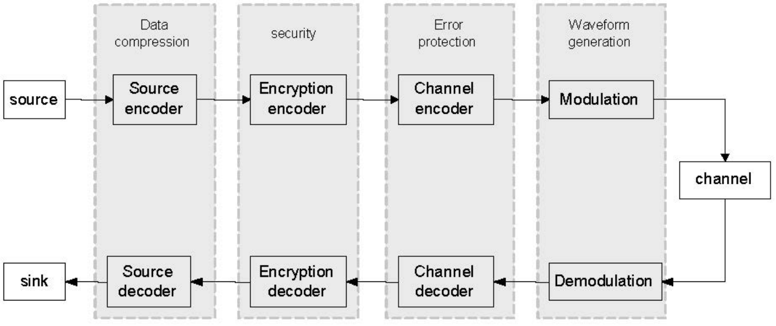
System consisting of three parts:

- a source of information,
- a physical channel, and
- a receiver or sink.

Issues to consider

- Low capacity of the channel.
- Security or the espionage problem.
- Errors in the messages caused by noise in the channel.

Communication Channels



Low Capacity of the Channel

Source coding

Data compression: substitute the data source with a compressed version of it.

Main methods in data compression

- Lossless compression: statistical, arithmetical or dictionary techniques. Used for text, executables, etc.
- Lossy compression: jpeg, fractal compression. Used for graphics, video, music, etc.

Source Coding: Shannon's First Theorem

Information and redundancy

A message is composed of *information* and *redundancy*, and both can be measured.

Source coding

Source coding is the process of taking the redundancy out of a message produced by an information source.

Theorem

A message can *compressed* until only information remains, and no more.

Roughly, the information contained in a message is the number of bits we get with the best compressor.

Source Coding: Example of Statistical Compression

Alphabet: $A = \{a, b, c, d, e\}$.

Message: $M = aaabccedabdb \in A^*$ has length 12.

Frequencies of letters appearing in M:

$$p(a) = \frac{1}{3}, \quad p(b) = \frac{1}{4}, \quad p(c) = p(d) = \frac{1}{6}, \quad p(e) = \frac{1}{12}$$

With a fixed-length ASCII-like binary encoding we need 36 bits to encode M:

$$a \to 000, \quad b \to 001, \quad c \to 010, \quad d \to 011, \quad e \to 100.$$

Source Coding: Example (continued)

Variable-length encoding: we associate a binary word with each letter such that the most frequent letters have shorter words. For example,

$$a \to 00, \quad b \to 01, \quad c \to 10, \quad d \to 110, \quad e \to 111.$$

Now, we only need 27 bits to encode M.

Remark

- $\mathcal{C} = \{00, 01, 10, 110, 111\}$ is a *code*: there is no ambiguity in the messages we can form with the words in $\mathcal{C}$.
- This is an example of a *Huffman encoding*.

Errors in the Channel: Channel Encoding

Channel encoding

- To recover all or most of the original message we add some *redundancy* to it.
- This process is called *channel encoding*.

Error detection & error correction

- Error Detection: parity check bit, cyclic redundancy codes (CRC), etc. Used only when retransmission is possible and cheap (disk reading, local network communication, etc).
- Error Correction: used when retransmitting the message is not possible or is too much expensive (CD-ROMs or DVD reading, satellite communication, etc)

Remark

- Intuitively, the greater the redundancy we add, the more likely is to recover the original information, *but* the lower the transmission rate will be.
- High reliability and low transmission rate seem incompatible objectives.

Theorem

Both problems can be solved at the same time. That is, it is possible to correct as many errors as we want adding only a controlled redundancy.

Example

The Binary Memoryless Symmetric Channel

- A source of information emits two symbols $\{0, 1\}$.
- We send these bits through a channel with error probability $p < 1/2$. That is:

$$p(0|1) = p(1|0) = p, \quad p(0|0) = p(1|1) = 1 - p.$$

Conditional probabilities: $p(i|j)$ is the probability of receiving j when i has been sent.

Example (cont.)

Do nothing: send every bit as it is

- Decision scheme: accept every bit as it arrives.
- Probability of getting a bit of information incorrectly: p.

Use a repetition code: send every bit three times

- Decision scheme: majority decision.
- Probability of getting a bit of information incorrectly: $p^3 + 3p^2(1 - p)$.
- Cost: we have added a 66.6% of redundancy and the transmission rate is three times bigger (e.g., it takes 3 times as long to send the same information).

Example (cont.)

Use a Hamming code

- Given four bits of information $x_1x_2x_3x_4$, compute:

$$x_5 = x_2 + x_3 + x_4, \quad x_6 = x_1 + x_3 + x_4, \quad x_7 = x_1 + x_2 + x_4$$

  (sums modulo 2) and send $x_1x_2x_3x_4x_5x_6x_7$.
- The Hamming code consists of the binary vectors in $\{0,1\}^7$ that are solutions to this linear system.

This an example of a *linear code*.

Example (cont.)

Decision scheme for the Hamming code

- After receiving $y_1y_2y_3y_4y_5y_6y_7$, compute the *syndromes*:

$$
\begin{aligned}
s_1 &= y_4 + y_5 + y_6 + y_7, \\
s_2 &= y_2 + y_3 + y_6 + y_7, \\
s_3 &= y_1 + y_3 + y_5 + y_7
\end{aligned}
$$

- If $s_1s_2s_3 = 000$, (we decide that) there is no error.
- Else, the error location is $s_1s_2s_{3(2)}$, representation of an integer to the base 2.

Example (cont.)

Example of decoding with the Hamming code

- If the information message is 0101, we send 0101 010.
- Suppose we get 0101110 (error in the 5th position).
- The syndromes are $s_1 = 1$, $s_2 = 0$, $s_3 = 1$.
- Hence we decide there is an error in the position $101_{(2)} = 5$.

Some Properties of the Hamming Code

Remark

- This Hamming code corrects up to 1 error in a word of length 7.
- Probability of a decision error: $1-(1-p)^7-7p(1-p)^6$.
- Comparison: the probability of a decision error for every 4 information bits is of 0.00203 with the Hamming code, and is of 0.00259 with the repetition code of length 3.

Information and Entropy

Intuitive Idea of Information

- The concept of information is tied to the concepts of probability and uncertainty.
- Information can be measured by the amount of uncertainty resolved.
- The uncertainty is described by means of the concept of probability.
- In fact, some messages are much likelier than others, and information implies surprise.

Amount of Information = Amount of Uncertainty

The amount of *information* contained in an event can also be understood as the quantity of uncertainty:

- *before* an event happens we have *uncertainty* about the result;
- *after* an event happens we get *information*.

So the amount of information is equal to the amount of uncertainty resolved.

Definition of Information

Let (Ω, p) be a finite probability space.

Definition of information

If $A \subseteq \Omega$ is an event, *the amount of information contained in* A is:

$$I(A) = \log_2 \frac{1}{p(A)}.$$

- Information is measured in *bits*.
- The information function is decreasing and additive: if A and B are independent events, then $I(A \cap B) = I(A) + I(B)$. (The function $f(x) = \log(1/x)$ is decreasing and additive: $f(xy) = f(x) + f(y)$.)

Examples

A perfect coin

For a perfect coin: $p(\text{heads}) = p(\text{tails}) = 1/2$. Hence, each event (heads or tails) contains 1 bit of information:

$$\log_2 \frac{1}{1/2} = \log_2 2 = 1.$$

In fact, this was the first use of the term *bit*, as a unit of information.

A non-perfect coin

However, if $p(\text{heads}) = 1/4$ and $p(\text{tails}) = 3/4$, then:

$$I(\text{heads}) = \log_2 4 = 2, \quad I(\text{tails}) = \log_2(4/3) = 0.415037$$

Entropy

Let X be a random variable taking values $x_1, \ldots, x_n$.

Definition of entropy

Define the *entropy* of X as the average amount of information:

$$H(X) = \sum_{i=1}^{n} p(X = x_i) \cdot I(X = x_i) = \sum_{i=1}^{n} p(x_i) \cdot \log \frac{1}{p(x_i)}$$

Convention: $0 \cdot \infty = 0$ (why?).

Example

If X is equally likely, then: $p(x_i) = 1/n$, and $H(X) = \log(n)$.

Properties of the entropy

1. $H(p_1, \ldots, p_n)$ is continuous: a small change in the probabilities implies a small change in the entropy.
2. $H(p_1, \ldots, p_n)$ is symmetric: the entropy only depends on the probabilities, not on their order.
3. $H(p_1, \ldots, p_n) \geq 0$, and $H = 0 \iff \exists i, p_i = 1$.
4. $H(p_1, \ldots, p_n, 0) = H(p_1, \ldots, p_n)$: an impossible case doesn't contribute to the entropy.
5. $H(1/n, \ldots, 1/n) < H(1/(n+1), \ldots, 1/(n+1))$: the entropy is greater for an equally likely distribution with $n+1$ values than for one with n values.
6. $H(p_1, \ldots, p_n) \leq H(1/n, \ldots, 1/n)$, and the equality holds iff $\forall i, p_i = 1/n$. For instance, there is more uncertainty in a perfect dice than in a loaded one.

Joint Entropy

If X, Y are random variables, $H(X, Y)$ denotes the entropy of the joint distribution and is called the *joint entropy* of the variables.

Proposition

1. $H(X, Y) \leq H(X) + H(Y)$
2. $H(X, Y) = H(X) + H(Y)$ iff X and Y are independent variables.

Remark

The joint entropy is the amount of information we get when we observe both variables at the same time.

Conditioned Entropy

Let X and Y be random variables. Then $X \mid Y = y$ is a random variable whose entropy is:

$$H(X \mid Y = y) = \sum_{i=1}^{n} p(x_i \mid y) \log \frac{1}{p(x_i \mid y)}.$$

Conditioned entropy

The average of all these entropies is called the *conditioned entropy* (even though it is not a true entropy!):

$$H(X \mid Y) = \sum_{j} p(y_j) H(X \mid Y = y_j)$$

Proposition

1. $H(X, Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y)$.
2. $H(X \mid Y) \leq H(X)$, *and the equality holds iff* X *and* Y *are independent.*

Remark

$H(X \mid Y)$ measures the uncertainty that remains in X after having observed Y.

Mutual Information

Mutual information

We define the *mutual information* of the random variables X and Y as:

$$I(X, Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X).$$

Proposition

$I(X, Y) = 0$ iff X and Y are independent variables.

Remark

$I(X, Y) =$ amount of information in X minus the amount of information *still* in X after knowing Y (and conversely) $=$ amount of information that each variable contains about the other.

Mnemonic Rule

Represent $H(X)$ as the set U and $H(Y)$ as the set V. Then:

$$\begin{aligned} H(X,Y) &\longleftrightarrow U \cup V \\ H(X \mid Y) &\longleftrightarrow U - V \\ H(Y \mid X) &\longleftrightarrow V - U \\ I(X,Y) &\longleftrightarrow U \cap V. \end{aligned}$$

Codes

Coding an alphabet into another one

- We want to send the information generated by an information source through a communication channel.
- The source emits symbols from an alphabet B, but the channel only accepts symbols from another alphabet A.
- Hence, we need a one-to-one mapping $f\colon B \to A^*$ that assigns to each symbol $b \in B$ a word $f(b) \in A^*$.
- Moreover, we need that any word of A^* that is a concatenation of words of the image $\mathcal{C} = f(B)$ decomposes uniquely as such concatenation.
- A set $\mathcal{C}$ with a property like that is called a *code*.

Alphabets. Concatenation

Definition

An alphabet is a finite set $A = \{a_1, \dots, a_q\}$, $q = |A|$.

- A word over A is a finite sequence of symbols $x = a_{i_1} \cdots a_{i_n}$.
- $n = \ell(x)$ is the length of the word x.
- λ is the empty word (has no symbol), $\ell(\lambda) = 0$.
- $A^* = \{\text{words of length} \geq 0\}$, $A^n = \{x \in A^* \mid \ell(x) = n\}$.

Concatenation

If $x = d_1 \cdots d_r, y = e_1 \cdots e_s \in A^*$:

$$xy := d_1 \cdots d_r e_1 \cdots e_s.$$

We have: $\ell(xy) = \ell(x) + \ell(y)$.

Factorizations, Prefixes and Codes

Factorization

A *factorization* of a word $x \in A^*$ is an expression $x = x_1 x_2 \cdots x_k$, where $x_i \in A^*$.

Prefixes

Let $x, y \in A^*$. The word x is a *prefix* of the word $y \iff \exists u \in A^* : y = xu$.

Codes

A subset $\mathcal{C} \subset A^*$ is a *code* if every message $M \in \mathcal{C}^*$ admits a unique factorization as a concatenation of words of $\mathcal{C}$:

$$c_1 c_2 \cdots c_r = d_1 d_2 \cdots d_s, \quad c_i, d_j \in \mathcal{C}$$

$$\Downarrow$$

$$r = s, \quad \text{and} \quad c_i = d_i, \quad i = 1, \ldots, r$$

Examples

Examples of codes

1. The Morse code over the alphabet $\{\bullet, -, \mathit{space}\}$.
2. Block codes of length $n \geq 1$.
3. The repetition code of length n over an alphabet A: $\mathrm{Rep}(n, A) = \{a \cdots a : a \in A\}$.

Example of a set that is not a code

$\mathcal{C} = \{a, c, ad, abb, bad, deb, bbcde\} \subset \{a, b, c, d, e\}^*$ is not a code: $abbcdebad = a|bbcde|bad = abb|c|deb|ad$.

Remark

Sardinas–Paterson algorithm decides whether a finite subset of A^* is a code or not, giving an ambiguous message if not.

Prefix Codes

Definition

A subset $\mathcal{C} \subset A^*$ is a *prefix* subset if no word of $\mathcal{C}$ is a prefix of another one; that is: $u, v \in \mathcal{C} \Rightarrow u$ is not a prefix of v.

Proposition

Every prefix set is a code.

Remark

Such a code is called a *prefix code* or an *instantaneous code* for the decoding can be done in *real-time*.

Example

The set $\{0, 10, 110, 1110\}$ is a binary prefix code.

Definition

An *encoding* of an alphabet B over another alphabet A is a one-to-one mapping $f\colon B \to A^*$ such that the image $f(B) \subset A^*$ is a code over A.

Equivalently, f is an encoding iff the induced mapping:

$$B^* \to A^*, \quad b_1 b_2 \cdots b_r \mapsto f(b_1)f(b_2)\cdots f(b_r)$$

is one-to-one.

Kraft's Inequality

Let $\mathcal{C} = \{x_1, \ldots, x_n\}$ be a q-ary code ($q = |A|$). Then:

$$\sum_{i=1}^{n} \frac{1}{q^{\ell(x_i)}} \leq 1.$$

Kraft-Macmillan's Theorem

Let $q, n, \ell_1, \ldots, \ell_n$ be positive integers such that:

$$\sum_{i=1}^{n} \frac{1}{q^{\ell_i}} \leq 1.$$

Then there exists a q-ary prefix code with n words of lengths $\ell_1, \ldots, \ell_n$.

Example

Remark

Kraft's inequality can be used to show that a given set is *not* a code (but not to show that it is!).

Example

For the set $\mathcal{C} = \{1, 00, 01, 111, 101\} \subseteq \{0, 1\}^*$ we have:

$$\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^3} > 1,$$

so $\mathcal{C}$ is not a code. That $\mathcal{C}$ is not a code can also be proved by considering the ambiguous message 111.

Corollary

If there exists a q-ary code composed of n words with lengths $\ell_1, \ldots, \ell_n$, then there is another code with the same properties that is also a prefix code.

Therefore, there is no loss of generality if we always work with prefix or instantaneous codes, for if in a given situation we have a code that is not a prefix code, we can substitute it by another one that is so and has the same cardinality and word-lengths.

Source Coding

Definition

A *Discrete Memoryless Information Source* (DMIS) is a pair $S = (A, X)$ consisting of a finite alphabet $A = \{a_1, \ldots, a_n\}$ and a finite probability distribution $X = \{p_1, \ldots, p_n\}$, where p_i is the probability that S emits the symbol a_i: $p_i = p(a_i)$.

'*Memoryless*' means that symbols are emitted independently of each other; that is:

$$p(a_{i_1} \cdots a_{i_k}) = p(a_{i_1}) \cdots p(a_{i_k}) = p_{i_1} \cdots p_{i_k}.$$

Entropy of a source

The entropy $H(S)$ of a DMIS S is the average amount of information given by S.

Average Length of an Encoding

Let $S = (A, X)$ be a DMIS and $f\colon A \to B^*$ an encoding, $q = |B|$. Then $f(A) = \mathcal{C} = \{x_1, \ldots, x_n\} \subseteq B^*$ is a code, where $x_i = f(a_i)$.

Definition

- The *average length* of the encoding f is:

$$\widetilde{\ell}_f(S) = \widetilde{\ell}_{\mathcal{C}}(S) = \sum_{i=1}^{n} p(a_i)\ell(x_i).$$

- The *minimum length* of S over q-ary alphabets is:

$$\widetilde{\ell}_q^{\min}(S) = \min_f \widetilde{\ell}_f(S).$$

  It only depends on q and S. This minimum always exists.

Definition

A *Huffman encoding* or an optimal encoding for a DMIS S is a prefix code $\mathcal{C}$ whose length is equal to the minimum length of the source; that is:

$$\widetilde{\ell}_{\mathcal{C}}(S) = \widetilde{\ell}_q^{\min}(S).$$

If no confusion arises, we also call $\mathcal{C} = f(A)$ a Huffman code for the source (note that the words of $\mathcal{C}$ are implicitly ordered).

Proposition 1

Assume that $p_1 \geq p_2 \geq \cdots \geq p_n$. Then there is a Huffman code $\mathcal{C} = \{x_1, x_2, \ldots, x_n\}$ such that $\ell(x_1) \leq \ell(x_2) \leq \cdots \leq \ell(x_n)$.

Proposition 2

Assume that $p_1 \geq p_2 \geq \cdots \geq p_n$. Then there is a Huffman code $\mathcal{C} = \{x_1, \ldots, x_n\}$ such that x_{n-1} and x_n differ only in the last symbol.

Binary Huffman Encodings: Construction

Huffman Algorithm

Let $S = (A, X)$ be DMIS with n symbols and probabilities $p_1 \geq \cdots \geq p_n$. Consider the source S' with $n-1$ symbols and probabilities $p_1, \ldots, p_{n-2}, p_{n-1} + p_n$.

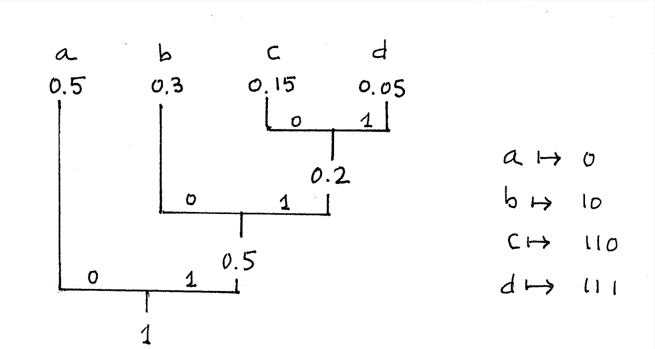
If $\mathcal{C}' = \{x_1, \ldots, x_{n-1}\}$ is a Huffman code for S', then:

$$\mathcal{C} = \{x_1, \ldots, x_{n-2}, x_{n-1}0, x_{n-1}1\}$$

is a Huffman code for S.

Example of Binary Huffman Algorithm

$A = \{a, b, c, d\}$, $p(a) = 0.5$, $p(b) = 0.3$, $p(c) = 0.15$, $p(d) = 0.05$



$\widetilde{\ell}^{\min}(S) = 0.5 \cdot 1 + 0.3 \cdot 2 + 0.15 \cdot 3 + 0.05 \cdot 3 = 1.7$ bits/symbol.

Compare with $H_2(S) = 1.64773$ (bits of information).

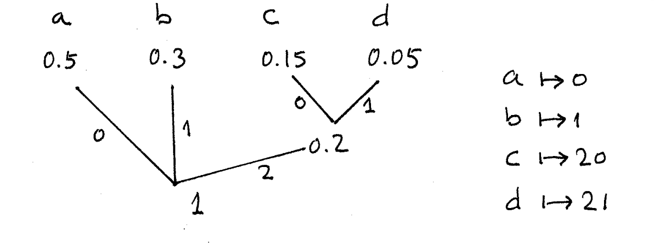
Algorithm

- Construct a q-ary tree adding up in each step the q smallest probabilities, except in the first step.
- In the first step, add up the q_0 smallest probabilities, where $2 \leq q_0 \leq q$ and $(q-1) \mid (n-q_0)$.

From these conditions it follows that:

1. the integer q_0 is unique;
2. in the following steps we always have q probabilities to add up.

$A = \{a, b, c, d\}$, $p(a) = 0.5$, $p(b) = 0.3$, $p(c) = 0.15$, $p(d) = 0.05$



$\widetilde{\ell}^{\min}(S) = 0.5 \cdot 1 + 0.3 \cdot 1 + 0.15 \cdot 2 + 0.05 \cdot 2 = 1.2$ ternary digits/symbol.

Compare with $H_3(S) = 1.0396$.

Theorem

If S is a discrete memoryless information source, then:

$$H_q(S) \leq \widetilde{\ell}_q^{\min}(S) < H_q(S) + 1.$$

Remarks

- H_q is the entropy computed with $\log_q$.
- This theorem says that a source cannot be compressed below its entropy (first inequality).

In the light of Shannon's first theorem, we give the following definition.

Definition

Given an encoding $A \to \mathcal{C} \subset B^*$ of a source $S = (A, X)$, we define its *efficiency* as:

$$\mathrm{eff}(\mathcal{C}) = \frac{H_q(S)}{\widetilde{\ell}_{\mathcal{C}}(S)}.$$

We have:

$$0 \leq \mathrm{eff}(\mathcal{C}) \leq \frac{H_q(S)}{\widetilde{\ell}_q^{\min}(S)} \leq 1.$$

Sometimes it is useful to send the information symbols in packets of length k instead of one at a time.

k-th extension of a source

The *k-th extension* of a source S is:

$$S^k = (A^k, X^k), \quad p(a_{i_1} \cdots a_{i_k}) = p(a_{i_1}) \cdots p(a_{i_k})$$

(recall that S is memoryless).

Proposition

$H(S^k) = kH(S)$.

Extensions in the limit

Consider all the extensions S^k of a DMIS source S, $k \geq 1$.

If we apply the first Shannon theorem to each of them, we get:

$$kH_q(S) = H_q(S^k) \leq \widetilde{\ell}_q^{\min}(S^k) < H_q(S^k) + 1 = kH_q(S) + 1.$$

Dividing by k and taking limits as $k \to +\infty$, we get:

$$H_q(S) \leq \frac{\widetilde{\ell}_q^{\min}(S^k)}{k} < H_q(S) + \frac{1}{k}$$

$$\lim_{k \to +\infty} \frac{\widetilde{\ell}_q^{\min}(S^k)}{k} = H_q(S)$$

This means that by using a suitable extension of S we can approach the entropy of the original source as much as we like.

Example of Extensions

$A = \{a, b, c, d\}$
$p(a) = 0.5, p(b) = 0.3, p(c) = 0.15, p(d) = 0.05$
$H_2(S) = 1.64773$

- Huffman code for S: $\widetilde{\ell}_2^{\min}(S) = 1.7$ bits/symbol of S.
- Huffman code for S^2: $\widetilde{\ell}_2^{\min}(S^2) = 3.3275$ bits/symbol of S^2. This represents 1.66375 bits/symbol of S.

Channel Coding

Discrete memoryless Channels

Definition

A *discrete memoryless channel* $K = (A, B, Q)$ consists of:

- an input alphabet $A = \{a_1, \ldots, a_s\}$,
- an output alphabet $B = \{b_1, \ldots, b_t\}$,
- a matrix of probabilities:

$$Q = \begin{bmatrix} p(b_1 \mid a_1) & \ldots & p(b_t \mid a_1) \\ \vdots & & \vdots \\ p(b_1 \mid a_s) & \ldots & p(b_t \mid a_s) \end{bmatrix}, \quad (\textit{channel matrix})$$

  where $p(b_j \mid a_i)$ is the probability of receiving b_j if a_i was sent.

1. Q is a *stochastic* matrix: each row is a probability distribution: $\sum_{j=1}^{t} p(b_j \mid a_i) = 1$, for each i.
2. The binary symmetric channel (BSC): $A = B = \{0, 1\}$, $0 \le p < 1/2$ and:

$$Q = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$

   $p = p(0|1) = p(1|0)$ is called the *error probability*.
3. The binary channel with erasure: $A = \{0, 1\}$, $B = \{0, 1, e\}$, and:

$$Q = \begin{bmatrix} 1-\varepsilon & 0 & \varepsilon \\ 0 & 1-\varepsilon & \varepsilon \end{bmatrix}$$

Types of Channels

Lossless: the output completely determines the input. Each column of Q has, at most, one non-zero element.

Determinist: the input completely determines the output. Each row of Q has, at most, one non-zero element (that must be a 1).

Noiseless: it is a lossless and determinist channel. We have $|A| = |B|$ and Q is the identity matrix, except for a permutation of rows if necessary.

Useless: the input and the output are independent. All the rows of Q are equal.

Symmetric: all the rows and columns of Q contain the same numbers, except for permutations. For instance, the binary symmetric channel.

Output distribution of a channel

The channel matrix Q transforms an input distribution $X = (p_1, \ldots, p_s)$ into an output distribution $Y = (q_1, \ldots, q_t)$ as:

$$(p_1, \ldots, p_s) \cdot Q = (q_1, \ldots, q_t).$$

So: $q_i = \sum_{j=1}^{s} p_j \cdot p(b_i \mid a_j)$. We write: $Y = K(X)$.

- If we input a discrete memoryless source $S = (A, X)$, then we get an output source $T = (B, K(X))$.
- The joint distribution of X and $Y = K(X)$ depends on both X and Q.
- The mutual information $I(X, Y)$ is the amount of information about X that goes through the channel.

Definition

The *capacity* of a channel K is the maximum value of $I(X, Y)$ as the input distribution X varies. It is denoted as $\gamma(K)$.

- This maximum value always exists because $I(X, Y)$ is a continuous function of the variables $p_1, \ldots, p_s$ defined on a *compact* set.
- If $\gamma(K) = I(X, K(X))$, for a particular X, we say that the capacity is *attained* at X.
- $I(X, Y) = H(X) - H(X \mid Y) \leq H(X) \leq \log(s)$.
- The computation of $\gamma(K)$, for a general K, is a non-trivial problem of optimization.

Lossless channel

For any input distribution X, we have $H(X \mid Y) = 0$. So:

$$I(X, Y) = H(X) - H(X \mid Y) = H(X)$$

and the maximum value is $\log(s)$, where s is the number of inputs. Hence:

$$\gamma(K) = \log(s)$$

Moreover, the capacity is attained at an equally likely input distribution.

Determinist channel

For any input distribution X, we have $H(Y \mid X) = 0$. So if X is the distribution which the capacity is attained at, then:

$$\gamma(K) = \max_X (H(Y) - H(Y \mid X)) = \max_X H(Y) \leq \log(t),$$

where t is the number of outputs.

- If there exists an input X that gives an equally likely output Y, then $\gamma(K) = \log(t)$.
- In the general case, we cannot say more.

Noiseless channel

This is a lossless channel, so we have $\gamma(K) = \log(s)$, where s is the number of inputs.

The capacity is attained at an equally likely input.

Useless channel

We have $H(X) = H(X \mid Y)$, because X and Y are independent. Hence $\gamma(K) = 0$.

Capacity of a Symmetric Channel

Theorem

Let $K = (A, B, Q)$ be a symmetric channel. Let $\alpha_1, \ldots, \alpha_s$ be the values of any row of Q. Then:

$$\gamma(K) = \log(s) - H(\alpha_1, \ldots, \alpha_s).$$

In particular, the capacity of the binary symmetric channel with error probability p is:

$$1 - H(p, 1-p) = 1 + p\log(p) + (1-p)\log(1-p).$$

Let S be a DMIS and let K be a binary symmetric channel with error probability $p < 1/2$.

Channel encodings

A *channel encoding* for S consists of:

1. a bijective mapping $\mathrm{cod}\colon A_S \to \mathcal{C} \subseteq \{0,1\}^n$, the *encoding mapping*;
2. a *decoding mapping* $\mathrm{dec}\colon \{0,1\}^n \to \mathcal{C} \cup \{?\}$

- The decoding mapping is also called a *decision scheme* or a *decoding scheme*.
- A decision scheme may be only defined in a subset of $\{0,1\}^n$. If so, it is called an *incomplete decision scheme*.

Maximum Likelihood Schemes (MLS)

Let $f\colon \{0,1\}^n \to \mathcal{C} \cup \{?\}$ be a decision scheme. Let $x \in \{0,1\}^n$ be the received word.

- If $f(x) \in \mathcal{C}$, then we *decide* that $f(x)$ is the word sent.
- If $f(x) = ?$, then we *announce* that a decoding error has occurred.

Aim of the decoding process

To maximize the probability that $f(x)$ be the word sent.

Definition

A decision scheme f is called a *maximum likelihood scheme* if $p(\text{we get } x \mid f(x) \text{ was sent}) = \max_{c \in \mathcal{C}} p(\text{we get } x \mid \text{we sent } c)$.

$f(x) \in \mathcal{C}$ satisfies that for no other word of $\mathcal{C}$ is more likely to have received x.

We have:

$$p(\text{error in } k \text{ fixed positions}) = p^k(1-p)^{n-k}.$$

Therefore, if $c \in \mathcal{C}$ differs from $x \in \{0,1\}^n$ in k positions then:

$$p(\text{we get } x \mid \text{we sent } c) = p^k(1-p)^{n-k}.$$

But: $p < 1/2 \Rightarrow 1-p > p$, so $p^k(1-p)^{n-k}$ is *big* for $n-k$ big and k small.

Theorem

For a BSC, the maximum likelihood scheme consists of finding the codeword with the minimum number of differences with the word received.

This kind of decoding is called *proximity decoding*.

Transmission Rate of a Code

Let $\mathcal{C} \subset A^n$ be a block code of length n. Let $M = |\mathcal{C}|$ the number of codewords and $q = |A|$ the number of symbols.

Definition

We define the *transmission rate* of $\mathcal{C}$ as:

$$R(\mathcal{C}) = \frac{\log_q(M)}{n}, \quad 0 < R(\mathcal{C}) < 1.$$

It is a measure of efficiency of $\mathcal{C}$.

- Aim: to maximize both the probability of correct decoding and the efficiency of a code, at the same time.
- Shannon's second theorem basically states that we can get both results if we take R less than the channel capacity.

Shannon's Second Theorem for Binary Symmetric Channels

Theorem

Let K be a BSC with probability error $p < 1/2$. Let $R < \gamma(K)$. Then for every $\varepsilon > 0$ there exists a length n_ε and a code $\mathcal{C} \subset \{0,1\}^{n_\varepsilon}$ such that $R(\mathcal{C}) \geq R$ and the probability of error decoding is $< \varepsilon$.

Remark

The proof is not constructive, so we know such codes exist but we don't know how to construct them.

Example

If $p = 0.001$, then $\gamma = 1 + p\log(p) + (1-p)\log(1-p) = 0.919$. So we can transmit with a transmission rate of the 90% and error probability less than any prefixed $\varepsilon > 0$.

Block Codes

Definition

A *block code* of length n over an alphabet A is a nonempty subset of A^n. Block codes are prefix codes.

Examples

- *Trivial* codes: $|\mathcal{C}| = 1$ and A^n (*total* code).
- *Repetition* codes: $\mathrm{Rep}(n, A)$.
- The *even binary code*: $\mathcal{C} \subseteq \{0,1\}^n$ consisting of the words with an even number of 1's. It can be described as:

$$\mathcal{C} = \{x_1 \cdots x_n \mid x_1 + \cdots + x_n = 0\}.$$

  We have $|\mathcal{C}| = 2^{n-1}$.

Example: a Hamming Code

The *binary Hamming code* of length 7: it is the set of words $x_1 \dots x_7 \in \{0,1\}^7$ such that:

$$x_4 + x_5 + x_6 + x_7 = 0$$
$$x_2 + x_3 + x_6 + x_7 = 0$$
$$x_1 + x_3 + x_5 + x_7 = 0$$

(arithmetic mod 2).

There are 16 codewords.

Hamming Distance

Definition

The *Hamming distance* between two words $x, y \in A^n$ is:

$$d(x, y) = |\{i \mid x_i \neq y_i\}| \qquad \text{(number of differences)}$$

where $x = x_1 \ldots x_n$, $y = y_1 \ldots y_n$.

If $S \subset A^n$ and $x \in A^n$: $d(x, S) = \min\{d(x, y) \mid y \in S\}$.

Proposition

$d \colon A^n \times A^n \to \mathbb{N}$ is a distance function; that is:

1. $d(x, y) \geq 0$, and $d(x, y) = 0 \iff x = y$;
2. $d(x, y) = d(y, x)$;
3. $d(x, z) \leq d(x, y) + d(y, z)$

Balls in A^n

The concept of Hamming distance allows us to think geometrically in A^n as if we were in the Euclidean space.

Definition

For $x \in A^n$ and $r \in \mathbb{N}$, the *ball* of center x and radius r is:

$$B_r(x) = \{y \in A^n \mid d(y,x) \le r\}.$$

Remark

- $B_r(x)$ contains all the words that differ from x in at most r positions.
- If we send the word x and r or fewer errors occur, then we get a word of $B_r(x)$.
- $|B_r(x)| = \sum_{i=0}^{r} \binom{n}{i}(q-1)^i$.

Minimum Distance of a Code

Defintion

The *minimum distance* $d(\mathcal{C})$ of a block code $\mathcal{C}$ is:

$$d(\mathcal{C}) = \min\{d(x,y) \mid x, y \in \mathcal{C}, x \neq y\}.$$

- It is the least number of differences between two distinct codewords.
- It is the most important parameter of the code (and the most difficult to find).
- A *good* code has to have many words, a big minimum distance (so that its codewords differ a lot among them), and short length.

Parameters of a Code

The type of a code

A q-ary code has *type* or *parameters* $(n, M, d)_q$ if it has length n, M words and minimum distance d.

Examples

- $d(A^n) = 1$.
- $d(\mathrm{Rep}(n)) = n$.
- The minimum distance of the binary even code is 2.
- The minimum distance of the binary Hamming code of length 7 is 3.

Other Parameters

- The *tangency radius* ρ is the greatest radius such that the balls with centers in the codewords and radius ρ are pairwise disjoint.
- The *information rate* $R = k/n$ tells us the rate of information symbols per codeword.
- The *redundancy* $n - k$. It is also called the number of parity check symbols.
- The *redundancy rate* $1 - R = (n - k)/n$.

Tangency Radius

The tangency radius of a code $\mathcal{C}$ is:

$$\rho = \rho(\mathcal{C}) = \max\{r \geq 0 \mid B_r(x) \cap B_r(y) = \emptyset, \forall x, y \in \mathcal{C}, x \neq y\}.$$

- If we send a word x and the channel introduces at most ρ errors, then we get a word $y \in B_\rho(x)$.
- So if we look for the closest codeword to y at distance $\leq \rho$, we get a unique answer: x.
- We say $\mathcal{C}$ can correct up to ρ errors.

Theorem

The tangency radius is given by: $\rho = \left\lfloor \frac{d-1}{2} \right\rfloor$.

Optimal codes

A code $\mathcal{C}$ of type $(n, M, d)_q$ is *optimal* if M is the largest cardinality among codes of length n and minimum distance d.

- We write $A_q(n, d)$ for the cardinality of an optimal $(n, M, d)_q$ code.
- $A_q(n, d)$ is difficult to compute. In many cases only lower and upper bounds are known.

For instance, for $q = 2$:

$d \downarrow / n \rightarrow$	5	6	7	8	9	10	11
3	4	8	16	20	40	72–79	144–158
5	2	2	2	4	6	12	24

Hamming Bound and Gilbert-Varshamov Bound

Hamming bound or the sphere packing bound

$$A_q(n,d) \leq \frac{q^n}{\sum_{i=0}^{\rho} \binom{n}{i}(q-1)^i}$$

where $\rho = \lfloor (d-1)/2 \rfloor$ is the tangency radius.

Gilbert-Varshamov bound

$$A_q(n,d) \geq \frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i}(q-1)^i}$$

Perfect Codes

Definition

A code is *perfect* if it is optimal and fulfills the Hamming bound.

- That is, $\mathcal{C}$ is perfect iff $|\mathcal{C}| = \frac{q^n}{\sum_{i=0}^{\rho} \binom{n}{i}(q-1)^i}$.
- In other words, there is no word outside the (pairwise disjoint) balls of radius ρ centered in the codewords.

Trivial perfect codes

The binary repetition code $\mathrm{Rep}_2(n)$ of odd length n, the total code over any alphabet and the codes of cardinality 1 are perfect codes. These are called the *trivial* perfect codes.

Hamming codes

Let $q = p^r$ be a power of a prime p, $r \geq 1$. Then the only non-trivial perfect q-ary codes are those codes whose parameters are $(n = (q^s - 1)/(q - 1), q^{n-s}, 3)_q$. For instance, the Hamming codes are the only linear and perfect codes with these parameters.

Golay codes

These are the codes G_{23}, with parameters $(23, 2^{12}, 7)_2$, and G_{11}, with parameters $(11, 3^6, 5)_3$. (Check that these parameters satisfy the Hamming bound.)

Equivalence of Codes

Definition

Two codes over the same alphabet A are *equivalent* if one of them can be obtained from the other performing a finite sequence of the following two operations:

1. apply a permutation of A to all the codewords in a fixed position:
$$x_1 \dots x_i \dots x_n \mapsto x_1 \dots \sigma(x_i) \dots x_n, \quad \sigma \in S_A$$
2. apply a permutation of the positions in all the codewords:
$$x_1 \dots x_n \mapsto x_{\pi(1)} \dots x_{\pi(n)}, \quad \pi \in S_n$$

Proposition

Equivalent codes have the same parameters.

Detection of Errors

Let $\mathcal{C}$ be a code of type $(n, M, d)_q$.

Error detection

We say that $\mathcal{C}$ is able to *detect* s errors if $B_s(u) \cap \mathcal{C} = \{u\}$, for any $u \in \mathcal{C}$.

- That is, there is no word of $\mathcal{C} - \{u\}$ at distance $\leq s$ from u.
- However, there could exist more than one codeword at distance $\leq s$ from a given word $x \in A^n$.
- In this case $x \notin \mathcal{C}$ and we don't know how to correct x.

Correction of Errors

Let $\mathcal{C}$ be a code of type $(n, M, d)_q$.

Error correction

We say that $\mathcal{C}$ is able to *correct* t errors if for any $u, v \in \mathcal{C}$, $u \neq v$, and any $x \in B_t(u)$ we have $d(x, u) < d(x, v)$.

That is, all the words of $B_t(u)$ have the codeword u as the closest codeword.

Let $\mathcal{C}$ be a code of type $(n, M, d)_q$.

Definition

We say that $\mathcal{C}$ is able to detect s errors and correct t errors *simultaneously* if for any $u, v \in \mathcal{C}$, $u \neq v$, and any $x \in A^n$ we have:

$$d(x, u) \leq s \Rightarrow d(x, v) > t.$$

That is, if $v \in \mathcal{C}$ is the closest codeword to $x \in A^n$ (it is unique), then x differs from the remainder codewords at least in $s + 1$ positions.

Criterion for Detection and Correction

Theorem

1. $\mathcal{C}$ can detect s errors $\iff s \leq d-1$.
2. $\mathcal{C}$ can correct t errors $\iff t \leq \lfloor (d-1)/2 \rfloor \iff 2t \leq d-1$.
3. $\mathcal{C}$ can detect s errors and correct t errors simultaneously $\iff s + t < d$.

Examples

- A code with $d = 3$ can be used to detect up to 2 errors or to correct 1 error, but cannot be used to do both things simultaneously.
- A code with $d = 4$ can be used to detect up to 3 errors or to correct 1 error or to detect 2 errors and correct 1 at the same time.

Algorithm for Correction: Sketch

- Input: $x \in A^n$, the word received.
- Main step: *look for* the closest codeword $u \in \mathcal{C}$ to x.
- If $d(x, u) \leq t$, correct x to u.
- If $d(x, u) > t$, announce that at least t symbols are incorrect.

The most difficult step in this algorithm is to look for the closest codeword. This is the point which we will focus on.

Finite Fields

What Do We Need Finite Fields For?

- Codes over alphabets with an algebraic structure are easier to deal with. E.g., $\{0, 1\} = \mathbb{Z}_2$.
- Even with a binary channel, it could be convenient to perform the computations in a finite field. For instance, we can put a field structure on $\{00, 01, 10, 11\}$ and treat binary strings of length 2 as symbols.
- Many algorithms for binary codes make extensive use of large fields containing $\mathbb{Z}_2$.
- For burst error-correcting purposes we consider single bytes as elements of a finite field of 256 elements.

A *field* is a set $\mathbb{K}$ with two operations $(+, \cdot)$ satisfying the following properties.

Axioms for the sum

S1) Associative: $a + (b + c) = (a + b) + c$.

S2) Commutative: $a + b = b + a$.

S3) Existence of a neutral element: there is an element $0 \in \mathbb{K}$ such that $a + 0 = a$, for all $a \in \mathbb{K}$.

S4) Existence of inverses: for all $a \in \mathbb{K}$ there is an $a' \in \mathbb{K}$ such that $a + a' = 0$.

Definition of Field

Axioms for the product

P1) Associative: $a(bc) = (ab)c$.

P2) Commutative: $ab = ba$.

P3) Existence of a neutral element: there is an element $1 \in \mathbb{K}$ such that $a \cdot 1 = a$, for all $a \in \mathbb{K}$.

P4) Existence of inverses: for all $a \in \mathbb{K} - \{0\}$ there is an $\overline{a} \in \mathbb{K}$ such that $a \cdot \overline{a} = 1$.

Distributive law

Both operations are related by the distributive law:

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

Remarks

- A *ring* is a set with two operations satisfying all the above properties except maybe P4 (the existence of inverses with respect to the product).
- The inverse element of a with respect to the sum is $-a$. So that $a + (-a) = 0$.
- The inverse element $\overline{a}$ of $a \neq 0$ with respect to the product is denoted by $a^{-1} = 1/a$. So that $a \cdot \frac{1}{a} = 1$.
- Therefore a field is a set where we can add, subtract, multiply and divide (by a nonzero element).
- If $\mathbb{K}$ is a field and $a \cdot b = 0$ in $\mathbb{K}$, then $a = 0$ or $b = 0$.

Examples of Fields and Rings

- The set of integers $\mathbb{Z}$ is a ring but not a field (e.g., 2 has no multiplicative inverse).
- The set of rational numbers $\mathbb{Q}$ is a field.
- Other familiar fields: $\mathbb{R}$, $\mathbb{C}$. Recall:

$$\mathbb{C} = \{a + bi \mid a, b \in \mathbb{R}, i^2 = -1\}$$

- If $\mathbb{K}$ is a field, then the set of polynomials with coefficients in $\mathbb{K}$ in an indeterminate X

$$\mathbb{K}[X] = \{a_0 + a_1 X + \cdots + a_n X^n \mid n \in \mathbb{N}, a_i \in \mathbb{K}\}$$

  is a ring but not a field (e.g., X has no multiplicative inverse).

Proposition

Let $\mathbb{F}$ be a finite field. Then there exists a prime number p such that $1 + \cdots^{p} + 1 = 0$ in $\mathbb{F}$. This prime p is called the *characteristic* of $\mathbb{F}$ and is denoted by $p = \mathrm{char}(\mathbb{F})$.

Proof.

Consider the sums $r_{\mathbb{F}} := 1 + \cdots^{r} + 1$, $r \in \mathbb{N}$. As $\mathbb{F}$ is finite, there exist r, s such that $r_{\mathbb{F}} = s_{\mathbb{F}}$. That is $p_{\mathbb{F}} = 0$, for some $p \in \mathbb{N}$. But then p must be a prime number, because $\mathbb{F}$ is a field. $\square$

As a consequence, if $\mathrm{char}\,\mathbb{F} = p$ then $0, 1, 2, \cdots, (p-1) \in \mathbb{F}$ and the arithmetic of this subset is just the arithmetic modulo p.

- In fact, the set $\mathbb{Z}_p = \{0, 1, \cdots, p-1\}$ with the sum and product modulo p is a finite field. Recall that

  $$k \in \mathbb{Z} \text{ has an inverse mod } p \iff \gcd(k, p) = 1$$

  Thus any element $k \in \mathbb{Z}_p - \{0\}$ has a multiplicative inverse in $\mathbb{Z}_p$.
- Any field $\mathbb{F}$ of characteristic p has the finite field $\mathbb{Z}_p$ as a subset (we say that $\mathbb{Z}_p$ is a subfield of $\mathbb{F}$ or that $\mathbb{F}$ is a field extension of $\mathbb{Z}_p$).

Are there more finite fields besides the fields $\mathbb{Z}_p$?

An Example of a Finite Field with 4 Elements

- Start with $\mathbb{R}$ and the irreducible polynomial $x^2 + 1$. *Invent* a root for it: $i^2 = -1$, and consider expressions like $a + bi$, with $a, b \in \mathbb{R}$. The sum and product of expressions like these are defined as if they were polynomials. We get a new field: $\mathbb{C}$, and $\mathbb{R} \subset \mathbb{C}$.
- Start with $\mathbb{Z}_2$ and the irreducible polynomial $x^2 + x + 1$. *Invent* a root for it: $\alpha^2 = \alpha + 1$, and consider expressions like $a + b\alpha$, where $a, b \in \mathbb{Z}_2$. The sum and product of expressions like these are defined as if they were polynomials. We get a new field: $\mathbb{F}_4$, and $\mathbb{Z}_2 \subset \mathbb{F}_4$.

$$\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2 = \alpha + 1\}.$$

The usual process of defining $\mathbb{C}$ from $\mathbb{R}$ consists of *inventing* a root i for the irreducible polynomial X^2+1 and then $\mathbb{C}=\{a+bi \mid a,b\in\mathbb{R}, i^2=-1\}$. The arithmetic in $\mathbb{C}$ is defined using the polynomial arithmetic:

$$(a+bX)+(c+dX)=(a+c)+(b+b)X$$
$$(a+bX)\cdot(c+dX)=ac+(ad+bc)X+bdX^2$$

and then we *pretend* that '$X^2=-1$'. If we do congruences in $\mathbb{R}[X]$ modulo X^2+1, then $X^2+1\equiv 0 \pmod{X^2+1}$.

How to Build a Field: $\mathbb{F}_4$

- Let's begin with $\mathbb{F}_2 = \{0, 1\}$ and an irreducible polynomial of degree 2 with coefficients in $\mathbb{F}_2$: $X^2 + X + 1$ is the only one!
- *Invent* a root α for it, so $\alpha^2 = \alpha + 1$.
- Then define $\mathbb{F}_4 = \{a + b\alpha \mid a, b \in \mathbb{F}_2\} = \{0 = 00, 1 = 01, \alpha = 10, 1 + \alpha = 11\}$.
- Finally the sum and the product in $\mathbb{F}_4$ are defined using the arithmetic of $\mathbb{F}_2[X]$ modulo $X^2 + X + 1$.

How to Build a Field: General Case

In general, we can start with a field $\mathbb{K}$ and perform the same sequence of actions:

- pick up an irreducible polynomial $f(X) \in \mathbb{K}[X]$ of degree n;
- invent a root for it α; that is, we require that $f(\alpha) = 0$;
- define $\mathbb{L}$ as the set of polynomial expressions in α of degree $\leq n - 1$;
- the arithmetic in $\mathbb{L}$ is defined through the arithmetic of polynomials and using the *substitution rule* $f(\alpha) = 0$.

Working Description of $\mathbb{F}_{p^n}$

- Fix an irreducible polynomial of degree n over $\mathbb{F}_p[X]$:
$$f(X) = X^n + f_{n-1}X^{n-1} + \cdots + f_0.$$
- Do congruences mod $f(X)$ and set $\alpha = \overline{X}$.
- As $\overline{f(X)} = f(\overline{X}) = f(\alpha) = 0$, α is a new root of $f(X)$.
- The elements of the new field $\mathbb{F}_{p^n}$ can be described as polynomials expressions in α up to degree $n-1$:
$$a_{n-1}\alpha^{n-1} + \cdots + a_1\alpha + a_0,$$
where $a_i \in \mathbb{F}_p$.
- Addition and multiplication of such expressions is done as polynomials and taking into account the *substitution rule* $f(\alpha) = 0$.

Order of an Element

Let $\mathbb{F}_q$ be a finite field (q is a power of a prime).

Multiplicative order

Define the *order* of a non-zero element $\beta \in \mathbb{F}_q$ as the least integer $t \geq 1$ such that $\beta^t = 1$. We write $\mathrm{ord}(\beta)$ for this integer.

Proposition

1. If $\beta^r = 1$, then $\mathrm{ord}(\beta) \mid r$.
2. For all $\beta \neq 0$, $\beta^{q-1} = 1$.
3. For all $\beta \neq 0$, $\mathrm{ord}(\beta) \mid q-1$.
4. We have: $\mathrm{ord}(\beta^k) = \frac{\mathrm{ord}(\beta)}{\gcd(k, \mathrm{ord}(\beta))}$.
5. There exists an element β whose order is $q-1$.

Primitive Elements

Definition

An element $\beta \in \mathbb{F}_q - \{0\}$ is called a *primitive* element if its order is $q-1$.

If β is a primitive element, then every nonzero element of $\mathbb{F}_q$ can be written as a power of β: $\mathbb{F}_q = \{0, 1, \beta, \beta^2, \ldots, \beta^{q-2}\}$ and $\beta^i \beta^j = \beta^{i+j \pmod{q-1}}$, because $\beta^{q-1} = 1$.

Remark

When using the polynomial representation, addition is easy and multiplication is hard. But if we use the representation of elements as powers of a primitive element, then multiplication is easy, but addition is hard.

Definition

An irreducible polynomial $f(X) \in \mathbb{F}_p[X]$ is called a primitive polynomial if $\alpha = \overline{X}$ is a primitive element in the finite field it defines $\mathbb{F}_q = \mathbb{F}_p[X]/f(X)$.

Example

- $X^4 + x + 1$ is primitive polynomial.
- $X^4 + X^3 + x^2 + X + 1$ is not a primitive polynomial. In this case, it is more difficult to find a primitive element.

Let $\beta \in \mathbb{F}_q$ be a primitive element.

Discrete logarithm

If $\gamma \in \mathbb{F}_q^*$, then there is an integer i such that $\gamma = \beta^i$. We call this integer i the *discrete logarithm* of γ with respect to β: $i = \log_\beta(\gamma)$. It is defined mod $q-1$.

Zech's logarithm

If $\beta^m \neq -1$, we define the Zech's logarithm of m as the integer $Z(m)$ mod $q-1$ such that:

$$1 + \beta^m = \beta^{Z(m)}.$$

- When lots of computations have to be done in a finite field it is very useful to construct the field table.
- This table contains a correspondence between the polynomial expressions in terms of α and the powers of a primitive element β.
- We can save a lot of work if α is already a primitive element.
- In most cases it is also useful to have the Zech logarithms of the field elements.
- Using Zech's logarithms, we have $\beta^m + \beta^n = \beta^{m+Z(n-m)}$.

Example

$\mathbb{F}_8 = \mathbb{F}_2[X]/(X^3 + X + 1)$.
$\alpha = \overline{X}$ is a primitive element.

$p(\alpha)$	α^i	$\alpha^{Z(i)}$	$p(\alpha)$	α^i	$\alpha^{Z(i)}$
1	1	–	$\alpha^2 + \alpha$	α^4	α^5
α	α	α^3	$\alpha^2 + \alpha + 1$	α^5	α^4
α^2	α^2	α^6	$\alpha^2 + 1$	α^6	α^2
$\alpha + 1$	α^3	α			

Linear Codes

Linear Codes

Fix a finite field $\mathbb{F}_q$ ($q = p^m$, p a prime number): this will be the alphabet. ($p = 2$ in applications.) We will denote by $\mathbf{0}$ the all-zeros vector and by $\mathbf{1}$ the all-ones vector:

$$\mathbf{0} = 00 \cdots^{n} 0, \quad \mathbf{1} = 11 \cdots^{n} 1.$$

Definition

A *linear code* of length n over $\mathbb{F}_q$ is a linear subspace $\mathcal{C} \subseteq \mathbb{F}_q^n$. This means:

1. $x, y \in \mathcal{C} \Rightarrow x + y \in \mathcal{C}$,
2. $x \in \mathcal{C}, \lambda \in \mathbb{F}_q \Rightarrow \lambda x \in \mathcal{C}$.

In the binary case, this second condition is always fulfilled.

How to Give a Linear Code

Parameters

Parameters of a linear code: $[n, k, d]_q$, where n is the length, $k = \dim(\mathcal{C})$ is the dimension and d is the minimum distance. Therefore: $|\mathcal{C}| = q^k$.

Generating matrix

Matrix whose rows are a basis of the linear code. It has k rows and rank k.

Parity-check matrix

Matrix associated with a homogeneous linear system of equations whose solutions are the codewords. It has $n - k$ rows and rank $n - k$.

Weight and Minimum Distance

Weight of a word

Define the *weight* $|x|$ of a word $x \in \mathbb{F}_q^n$ as the number of non-zero coordinates; that is $|x| = d(x, 0)$.

Remark

1. $d(x, y) = |x - y|$, if $x, y \in \mathbb{F}_q^n$.
2. If $\mathcal{C}$ is a linear code, then $d(\mathcal{C}) = \min\{|x| \mid x \in \mathcal{C}, x \neq \mathbf{0}\}$.

Generating matrix

A generating matrix for $\mathcal{C}$ of dimension k is a matrix whose k rows are the vectors of a basis. A linear code admits many different generating matrices.

Examples

1. Two possible generating matrices for the binary even code $\mathcal{C} = \{000, 011, 101, 110\}$:

$$G_1 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad G_2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

2. The repetition code $\mathrm{Rep}_q(n)$ is generated by the codeword **1**.

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

Let $\mathcal{C}$ be a $[n, k]_q$-linear code with generating matrix G.

Encoding mapping

The encoding mapping associated to G is:

$$\begin{aligned} \mathrm{enc}_G \colon \mathbb{F}_q^k &\to \mathcal{C} \subseteq \mathbb{F}_q^n \\ a_1 a_2 \ldots a_k &\mapsto (a_1, a_2, \ldots, a_k) \cdot G = (x_1, x_2, \ldots, x_n) \in \mathcal{C} \end{aligned}$$

enc_G depends on G: if we change the generating matrix, then the encoding mapping is different, though the code does not change.

Example

- The binary even code of length 3 admits the following two different encoding mappings:

$$\begin{aligned} \text{with } G_1\text{:} \quad & a_1a_2 \mapsto (a_1, a_2)G_1 = (a_1, a_2, a_1 + a_2) \\ \text{with } G_2\text{:} \quad & a_1a_2 \mapsto (a_1, a_2)G_2 = (a_1, a_1 + a_2, a_2) \end{aligned}$$

- We observe that using G_1 the information bits a_1 and a_2 are at the beginning of the codeword, so it is easier to extract them in the decoding process.
- When the information symbols are at the beginning of the codeword the encoding is called *systematic*.

Systematic Encodings

Definition

A linear code $\mathcal{C}$ is *systematic* if it admits a systematic encoding; that is, an encoding with a generating matrix of the shape $G = (I_k \mid A)$, where I_k is the $k \times k$ identity matrix.

Thus a systematic encoding has the form:

$$a_1 a_2 \dots a_k \mapsto (a_1, a_2, \dots, a_k)(I_k \mid A) = (a_1, a_2 \dots, a_k, \dots)$$

Proposition

Every linear code is equivalent to a systematic code.

Parity-Check Matrix

A $[n, k]_q$ linear code can be defined as the set of solutions of an homogeneous linear system of rank $n - k$. The associated matrix to such a system is called a *parity-check matrix* (or simply a check matrix) for the code. It is usually denoted by H. Thus:

$$x_1 x_2 \dots x_n \in \mathcal{C} \iff H \cdot x^t = \mathbf{0}^t,$$

where A^t denotes the transpose of a matrix A.

We will always assume that the linear system has the exact number of independent equations, so H has $n - k$ independent rows.

Theorem

Let $\mathcal{C}$ be a $[n, k]_q$ linear code with generator matrix G and parity-check matrix H.

1. $H \cdot G^t = 0$.
2. If $G = (I_k \mid A)$ is systematic, then $H = (-A^t \mid I_{n-k})$ is a parity-check matrix.

A parity-check matrix of the form $(-A^t \mid I_{n-k})$ is said to be in standard form.

Examples

- Let $\mathcal{C}$ the even binary code of length 3. Then:

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

- The repetition code $\mathrm{Rep}_q(n)$ can be defined by the equations $x_i = x_n$, $i = 1, 2 \ldots, n-1$, so a parity-check matrix is:

$$H = \begin{bmatrix} & \vdots & -1 \\ I_{n-1} & \vdots & \vdots \\ & \vdots & -1 \end{bmatrix}$$

Example: Extending a Code by a Parity Bit

Let $\mathcal{C}$ be a linear code with parity-check matrix H. Let $\hat{\mathcal{C}}$ the code obtained from $\mathcal{C}$ adding a parity bit:

$$\hat{\mathcal{C}} = \left\{x_1 \ldots x_n x_{n+1} \mid x_1 \ldots x_n \in \mathcal{C}, \textstyle\sum_{i=1}^{n+1} x_i = 0\right\}.$$

Then a parity-check matrix for $\hat{\mathcal{C}}$ is:

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

Fundamental Property of the Parity-Check Matrix

A fundamental property of the parity-check matrix is that the minimum distance of the code can be *read* from H itself (though in practice it is not so easy!).

Theorem

Let H be the parity-check matrix of a linear code $\mathcal{C}$ of type $[n, k, d]_q$. Then $d(\mathcal{C}) = d$ if, and only if, the following two conditions hold:

- H has d linearly dependent (l.d.) columns, and
- every set of $d - 1$ columns is linearly independent (l.i.).

Let's apply the theorem for $d = 3$.

- $d = 3 \iff H$ has 3 l.d. columns but every set of 2 columns is l.i.
- We have to construct a matrix H where no column is a multiple of any other and with 3 l.d. columns.
- A *Hamming code* is a code with a parity-check matrix satisfying this property and as many columns as possible.

Theorem

The maximum number of nonzero vectors of $\mathbb{F}_q^r$ such that no one is a multiple of any other is:

$$\frac{q^r - 1}{q - 1} = q^{r-1} + \cdots + q + 1.$$

Parameters of a Hamming code

The parameters of a Hamming code are:

$$n = \frac{q^r - 1}{q - 1}, \quad k = n - r, \quad d = 3.$$

r is called the codimension of the Hamming code.

Hamming Codes: Standard Form

Let $\beta \in \mathbb{F}_q^*$ be a primitive element.

- We define an linear order in $\mathbb{F}_q$ with respect to β as:

$$0 < 1 < \beta < \beta^2 < \cdots < \beta^{q-2}.$$

- This order determines a lexicographic order in the linear space $\mathbb{F}_q^r$.
- Pick up from each set $\{\lambda v \mid \lambda \neq 0\}$ the unique vector whose first non-zero coordinate is 1 and write all these vectors in lexicographic order.
- The resulting matrix is called a parity-check matrix in standard form. We denote by $\mathrm{Ham}_q(r)$ this Hamming code of codimension r.

Hamming Codes: Examples

- $\mathrm{Ham}_2(2)$: $H = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$.
- $\mathrm{Ham}_2(3)$: $H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$.
- $\mathrm{Ham}_4(2)$: $\mathbb{F}_4 = \mathbb{F}_2[X]/(X^2 + X + 1)$, $\alpha = \overline{X}$.

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

- $\mathrm{Ham}_9(2)$: $\mathbb{F}_9 = \mathbb{Z}_3[X]/(X^2 - X + 2)$, $\alpha^2 = \alpha + 1$ is primitive. We have $r = 2$, so $n = 10$, $k = 8$. We get a $[10, 8, 3]_9$-code.

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 \end{bmatrix}$$

Let $\mathcal{C}$ be a linear code of type $[n, k, d]_q$ with parity-check matrix H.

Syndrome of a word

Define the *syndrome* of a word $y \in \mathbb{F}_q^n$ as the word $s(y)$ of length $n - k$ such that $s(y)^t = Hy^t$.

Remark

The codewords are precisely those words with syndrome equal to the zero vector:

$$\mathcal{C} = \{x \in \mathbb{F}_q^n \mid s(x) = \mathbf{0}\}.$$

Error pattern

Assume that we send $x \in \mathcal{C}$ and the channel introduces the error e. Hence we receive the word $y = x + e$. The vector e is called the *error pattern*.

Properties

- $s(y) = s(x + e) = s(e)$.
- If the error pattern e has weight 1, then the $s(y)$ is a multiple of the i-th column of H.
- In general, if e has weight r, then the $s(y)$ is a linear combination of r columns of H (corresponding to the nonzero positions of the error pattern e).

Coset associated to a syndrome

Recall that the parity-check matrix H has maximal rank.

- For any vector $s \in \mathbb{F}_q^{n-k}$ (a possible syndrome), the linear system $Hy^t = s^t$ is always compatible.
- The solutions of $Hy^t = s^t$ consist of the words $y \in \mathbb{F}_q^n$ that have syndrome s.
- We define the *coset* associated with the syndrome s to be the set of solutions to $Hy^t = s^t$.

Proposition

The coset associated with s is $y_0 + \mathcal{C} = \{y_0 + x \mid x \in \mathcal{C}\}$, where y_0 is any word with syndrome $s(y_0) = s$.

Properties of Cosets

1. Two vectors define the same coset iff they have the same syndrome:

$$y + \mathcal{C} = z + \mathcal{C} \iff s(y) = s(z).$$

2. The difference of two vectors belonging to the same coset is a codeword. Reciprocally: if the difference of two vectors is a codeword, then they belong to the same coset:

$$y + \mathcal{C} = z + \mathcal{C} \iff y - z \in \mathcal{C}.$$

3. Either two cosets are disjoint or they are the same (that is, if two cosets have a non-empty intersection, then they are equal):

$$(y + \mathcal{C}) \cap (z + \mathcal{C}) = \emptyset \quad \text{or} \quad y + \mathcal{C} = z + \mathcal{C}.$$

Leaders of a Coset

Definition

A leader of a coset $y + \mathcal{C}$ is a word ℓ of minimal weight. We write $\ell(s(y))$, because in fact it depends on the syndrome of y, and not directly on y.

Properties

If $\ell \in y + \mathcal{C}$ is a leader, then:

- $y - \ell$ is the closest codeword to y.
- Hence, in a proximity decoding algorithm, we decode y as $y - \ell$.
- If $|\ell| \leq \lfloor (d-1)/2 \rfloor$, then ℓ is the unique leader of the coset.

Decoding Algorithm for Linear Codes (Error-Correcting)

Preprocessing

1. List all possible syndromes: all vectors of $\mathbb{F}_q^{n-k}$.
2. For each syndrome s, compute a leader $\ell(s)$ in the coset of words with syndrome s: $\ell(s)$ is a solution of $Hy^t = s^t$ with minimum weight.

Algorithm

Assume we get $y \in \mathbb{F}_q^n$ from the channel.

1. Compute the syndrome $s = s(y)$.
2. Decode y as $y - \ell(s)$.

Some Remarks

- The preprocessing is only useful when we have to decode lots of words.
- A leader with weight $\leq \rho$ is unique in its coset.
- If we only list those leaders that are unique (that is, with weight $\leq \rho$), then the decoding is *incomplete*. In this case, the algorithm doesn't decide how to process a word with more errors than the error-correcting capability of the code.

An Example of Decoding

Let $\mathcal{C}$ be a $[4,2,2]_2$-code with:

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

syndrome	00	11	01	10
leader	0000	1000	0100	0010

- $\rho = 0$, so $\mathcal{C}$ cannot be used for error-correcting.
- Leaders of weight 1 are not unique in general.
- The decoding of $y = 1111$ is done as:

  $$s = s(y) = (0,1), \quad y - \ell(01) = 1111 - 0100 = 1011.$$

  But if 0001 is chosen as a leader for the syndrome 01, then y is corrected as 1110 instead.

An Example of Incomplete Decoding

Consider the $[5, 2, 3]_2$-code with check matrix:

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

A table of syndrome-leader for an incomplete decoding is (we list only those leaders whose weight is ≤ 1):

syndrome	000	110	011	100	010	001
leader	00000	10000	01000	00100	00010	00001

- $y = 10011$: $s = s(y) = yH^t = 101$ is not on the table. So the algorithm doesn't decide.
- $y = 11111$: $s(y) = 010$ and y is decoded as $y - \ell(010) = 11111 - 00010 = 11101$.

Let $\mathcal{H}$ be a Hamming code in standard form.

- Hamming codes are perfect, so a complete decoding algorithm is possible.
- Assume we get $y = x + e$, where $x \in \mathcal{H}$ and $|e| \leq 1$.
- Compute $s(y) = s(e)$. If $s(y) = 0$, then assume that no error has ocurred.
- Assume $s(y) \neq 0$. Let λ be the first non-zero coordinate in $s(y)$. As $\mathcal{H}$ is in standard form, $\lambda^{-1}s(y)$ is a column of H, say the i-th column.
- The leader corresponding to this syndrome is $\lambda \cdot e_i$, that has all zeros except for the i-th coordinate that is λ. Hence y is decoded as:

$$y - (0, \dots, \lambda^{i)}, \dots, 0)$$

Decoding Hamming Codes: Algorithm

1. Compute the syndrome $s(y)$ of the word received y.
2. If $s(y) = 0$, we assume that there is no error.
3. If $s(y) \neq 0$, then $s(y) = \lambda H_i$, where H_i is the i-th column of H. The we announce that an error in the position i has occurred of *magnitude* λ and we decode y as:

$$y - (0, \ldots, \lambda^{i)}, \ldots, 0).$$

Decoding Hamming Codes: Binary Case

In the binary case, the columns of H are the representations of integers from 1 to $n = 2^r - 1$ to the base 2 (this is the imposed lexicographic order). Then we can substitute step 3 in the previous algorithm by:

3. If $s(y) \neq 0$, then $s(y)$ is the representation of the error position to the base 2.

Extended Binary Hamming Codes

Let $\mathcal{H} = \mathrm{Ham}_2(r)$ be the binary Hamming code of codimension r. Let $\hat{\mathcal{H}}$ be the code obtained from $\mathcal{H}$ by adding a parity bit:

$$\hat{\mathcal{H}} = \left\{x_1 \dots x_n x_{n+1} \mid x_1 \dots x_n \in \mathcal{H}, \textstyle\sum_{i=1}^{n+1} x_i = 0\right\}$$

As $d(\mathcal{H}) = 3$ is odd, we know that $d(\hat{\mathcal{H}}) = 4$. So $\hat{\mathcal{H}}$ can be used to simultaneously correct 1 error and detect 2 errors.

Let's see an algorithm for doing so.

Algorithm

Assume we receive the word y. Compute the syndrome $s(y) = (s_1, \dots, s_r, s_{r+1})$. Observe that $(s_1, \dots, s_r)$ can be computed using the parity-check matrix H of $\mathcal{H}$.

1. If $s(y) = 0$, announce no error.
2. If $s(y) \neq 0$ but $s_{r+1} = 1$ and $(s_1, \dots, s_r) \neq 0$, announce there is 1 error in the position given by $(s_1 s_2 \dots s_r)_{(2)}$.
3. If $s(y) \neq 0$ but $s_{r+1} = 1$ and $(s_1, \dots, s_r) = 0$, announce there is 1 error in the last position.
4. If $s(y) \neq 0$ but $s_{r+1} = 0$ and $(s_1, \dots, s_r) \neq 0$, announce 2 errors and ask for retransmission.

Let $\mathcal{C}$ be the linear code over $\mathbb{Z}_{11}$ of length 10 defined by the equation:

$$x_1 + 2x_2 + 3x_3 + \cdots + 10x_{10} = 0$$

(we represent 10 by the letter X).

The International Standard Book Number (ISBN) code $\mathcal{I}$ is the subcode of $\mathcal{C}$ whose words have no X in the first 9 positions.

Proposition

Let $x \in \mathcal{I}$ and $y \in \mathbb{Z}_{11}^{10}$.

1. *If $d(x, y) = 1$, then $s(y) \neq 0$ and if we know either the error location or the error magnitude, we can correct the error.*
2. *If y differs from x in the transposition of two digits, then $s(y) \neq 0$ and we can detect this error.*

Let $\mathcal{C}$ be the linear code over $\mathbb{Z}_{23}$ of length 8 given by the equation:

$$x_{-1} + \sum_{i=0}^{7} 10^i x_i = 0.$$

That is:

$$x_{-1} = x_{14} + x_7 + 6x_6 + 19x_5 + 18x_4 + 11x_3 + 8x_2 + 10x_1 + x_0.$$

The DNI code $\mathcal{D}$ is the subcode of $\mathcal{C}$ whose words $x_{-1}x_0 \dots x_7$ satisfy $x_i \in \{0, 1, \cdots, 9\}$, for $0 \leq i \leq 7$. Moreover, we substitute x_{-1} by a letter according to the following table:

0–T	1–R	2–W	3–A	4–G	5–M	6–Y	7–F
8–P	9–D	10–X	11–B	12–N	13–J	14–Z	15–S
16–Q	17–V	18–H	19–L	20–C	21–K	22–E	

Proposition

Let $x \in \mathcal{D}$ and $y \in \mathbb{Z}_{23}^{9}$.

1. *If $d(x, y) = 1$, then $s(y) \neq 0$ and if we know either the error location or the error magnitude, we can correct the error.*
2. *If y differs from x in the transposition of two digits, then $s(y) \neq 0$ and we can detect this error.*

Another Look at Hamming Codes

- Recall that the columns of the check matrix H of the Hamming code $\mathrm{Ham}_2(r)$ are the non-zero binary words of length r, $r \geq 2$.
- Consider a primitive element $\alpha \in \mathbb{F}_{2^r}$. The elements of $\mathbb{F}_{2^r}^*$ can be written as non-zero binary words of length r and also as powers of α.
- So H can be written in a *simplified* form as:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \end{bmatrix}, \quad (n = 2^r - 1)$$

where α^i has to be substituted by the corresponding binary string (the coefficients of the polynomial expression in α).

Binary words as polynomials

- From now on, we'll identify binary words length n with polynomials up to degree $n-1$:

$$a = a_0 a_1 \dots a_{n-1} \in \mathbb{F}_2^n \leftrightarrow a(t) = a_0 + a_1 t + \cdots + a_{n-1} t^{n-1} \in \mathbb{F}_2[t]_n$$

where $\mathbb{F}_2[t]_n = \{a(t) | \deg(a(t)) < n\}$.

- In this setting, $a = a_0 a_1 \dots a_{n-1} \in \mathbb{F}_2^n$ belongs to $\mathrm{Ham}_2(r)$ iff

$$H \cdot a^t = a_0 + a_1\alpha + \cdots + a_{n-1}\alpha^{n-1} = 0.$$

- That is: $a(t) \in \mathrm{Ham}_2(r) \iff a(\alpha) = 0$.
- But if $a(\alpha) = 0$, then $a(\alpha^2) = 0$.
- What happens if we also impose that $a(\alpha^3) = 0$?

A BCH Code that Corrects 2 Errors

BCH = Bose—Chaudhuri—Hocquenhe

- Let $\mathcal{B}$ be the binary code of length 15 defined by:

$$\mathcal{B} = \{a(t) = a_0 + a_1 t + \cdots + a_{14} t^{14} \mid a(\alpha) = a(\alpha^3) = 0\}$$

  where $\alpha \in \mathbb{F}_{16}$ is a primitive element.
- A parity-check matrix in simplified form is given by:

$$K = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{12} \end{bmatrix}$$

  (the second row is the cube of the first row).
- We'll see that $d \geq 5$, so $\mathcal{B}$ can correct at least 2 errors.

$\mathbb{F}_{16} = \mathbb{F}_2[X]/(x^4 + x + 1)$, $\alpha = \overline{X}$ is primitive.

Write $a_3 a_2 a_1 a_0 = a_3\alpha^3 + a_2\alpha^2 + a_1\alpha + a_0$.

0000	0	1011	α^7
0001	1	0101	α^8
0010	α	1010	α^9
0100	α^2	0111	α^{10}
1000	α^3	1110	α^{11}
0011	α^4	1111	α^{12}
0110	α^5	1101	α^{13}
1100	α^6	1001	α^{14}

Strategy for Decoding

Let $y \in \mathbb{F}_2^{15}$ be the word received and write its syndrome as $s(y) = (s_1, s_2)$, where $s_i \in \mathbb{F}_{16}$.

1. If there is no error, then $(s_1, s_2) = (0, 0)$.
2. If there is just 1 error in the position i, then the syndrome is the column i of K:
$$(s_1, s_2) = (\alpha^i, \alpha^{3i}).$$
In this case $s_2 = s_1^3$.
3. If there are 2 errors in the positions i and j, then the syndrome is the sum of the columns i and j of K:
$$(s_1, s_2) = (\alpha^i, \alpha^{3i}) + (\alpha^j, \alpha^{3j})$$

Hence, the strategy is: find out whether the syndrome is equal to a column ($s_2 = s_1^3$) or to a sum of two columns of K.

Strategy for Decoding

Assume there are two errors (then $s_1 \neq 0$). Let $a = \alpha^i$ and $b = \alpha^j$. Assume $(s_1, s_2) = (a, a^3) + (b, b^3) = (a + b, a^3 + b^3)$. We have to solve the non-linear system:

$$a + b = s_1, \quad a^3 + b^3 = s_2.$$

Use $(a + b)^3 = a^3 + a^2b + ab^2 + b^3 = (a^3 + b^3) + ab(a + b)$, so we have $s_1^3 = s_2 + abs_1$ and we can compute ab from s_1 and s_2:

$$ab = \frac{s_1^3 - s_2}{s_1}$$

But now we know $a + b = s_1$ and $ab = (s_1^3 - s_2)/s_1$ and a, b are the solutions to the quadratic equation:

$$T^2 - (a + b)T + ab = 0.$$

A Decoding Algorithm for $\mathcal{B}$

1. If $(s_1, s_2) = (0, 0)$: no error.
2. If $s(y) \neq 0$ and $s_1^3 = s_2$, then there is 1 error in the position $i = \log_\alpha(s_1)$.
3. If $s(y) \neq 0$ and $s_1^3 \neq s_2$, then we try to solve the quadratic equation
$$T^2 + s_1 T + (s_1^3 + s_2)/s_1 = 0.$$
If α^i and α^j are the solutions, then there are 2 errors in the positions i and j.
4. If the above quadratic equation has no solutions, then the channel has introduced 3 or more errors.

We receive $y = 11111\,11111\,11111 = \sum_{i=0}^{14} t^i$. Compute the syndrome:

$$Ky^t = \begin{bmatrix} y(\alpha) \\ y(\alpha^3) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

because:

$$y(\alpha) = \sum_{i=0}^{14} \alpha^i = \frac{\alpha^{15} - 1}{\alpha - 1} = 0, \qquad y(\alpha^3) = \sum_{i=0}^{14} (\alpha^3)^i = \frac{\alpha^{30} - 1}{\alpha^3 - 1} = 0$$

Hence $y \in \mathcal{B}$ and there is no error.

Examples of Decoding: 1 Error

We receive

$$\begin{aligned} y &= 10011\,10011\,00001 \\ &= 1 + t^3 + t^4 + t^5 + t^8 + t^9 + t^{14} \end{aligned}$$

Compute the syndrome:

$$s_1 = y(\alpha) = \alpha^9, \qquad s_2 = y(\alpha^3) = \alpha^{12}$$

Now $s_1^3 = (\alpha^9)^3 = \alpha^{27} = \alpha^{12} = s_2$, so there is 1 error in the position 9.

Hence we decode y as $10011\,1001\mathbf{0}\,00001$.

Examples of Decoding: 2 Errors

We receive

$$\begin{aligned} y &= 11111\,11110\,01111 \\ &= 1 + t + t^2 + t^3 + t^4 + t^5 + t^6 + t^7 + t^8 + t^{11} + t^{12} + t^{13} + t^{14} \end{aligned}$$

Compute the syndrome:

$$s_1 = y(\alpha) = \alpha^{13}, \qquad s_2 = y(\alpha^3) = \alpha^{11}$$

Now: $s_1^3 = \alpha^9 \neq s_2$, so there are ≥ 2 errors. We solve the quadratic equation:

$$T^2 + s_1 T + (s_1^3 + s_2)/s_1 = T^2 + \alpha^{13} T + \alpha^4 = 0.$$

The solutions are $T = \alpha^9$ and $T = \alpha^{10}$. So the errors are in the positions 9 and 10 and we decode y as 11111 1111**1** **1**1111.

Examples of Decoding: ≥ 3 Errors

We receive

$$\begin{aligned} y &= 10111\,11110\,01111 \\ &= 1 + t^2 + t^3 + t^4 + t^5 + t^6 + t^7 + t^8 + t^{11} + t^{12} + t^{13} + t^{14} \end{aligned}$$

Compute the syndrome:

$$s_1 = y(\alpha) = \alpha^{12}, \qquad s_2 = y(\alpha^3) = \alpha^5$$

Now: $s_1^3 = \alpha^6 \neq s_2$, so there are ≥ 2 errors. We try to solve the quadratic equation:

$$T^2 + s_1 T + (s_1^3 + s_2)/s_1 = T^2 + \alpha^{12} T + \alpha^{12} = 0.$$

This equation has no solution in $\mathbb{F}_{16}$, so the channel has introduced ≥ 3 errors.

Cyclic Codes

Cyclic Codes: Introduction

Cyclic codes form an important class of linear codes.

- Encoders and decoders for cyclic codes can be implemented using linear shift registers.
- Well-known families of codes such as BCH codes or Reed-Solomon codes are cyclic codes.
- Lower bounds for the minimum distance can be easily computed in many cases.
- They are commonly used for detection purposes in contexts such as Ethernet communication protocol, where they are known as CRC: *cyclic redundancy codes*.

Cyclic Codes

Definition

A linear code $\mathcal{C}$ over $\mathbb{F}_q$ is *cyclic* if it satisfies:

$$c_0c_1\dots c_{n-1} \in \mathcal{C} \Rightarrow c_{n-1}c_0\dots c_{n-2} \in \mathcal{C}.$$

Remarks

- We identify the vector space $\mathbb{F}_q^n$ with $\mathbb{F}_q[x]_n$:

$$c = c_0c_1\dots c_{n-1} \leftrightarrow c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$$

- Multiplication by x shifts the symbols one position to the right:

$$\begin{aligned} x \cdot c(x) &= x(c_0 + c_1x + \cdots + c_{n-1}x^{n-1}) \\ &= c_0x + c_1x^2 + \cdots + c_{n-2}x^{n-1} + c_{n-1}x^n \end{aligned}$$

Cyclic Codes

Remarks

- If we had "$x^n = 1$", then $xc(x) = c_{n-1} + c_0 x + \cdots + c_{n-2}x^{n-2}$, and the condition for a code to be cyclic would be:

$$c(x) \in \mathcal{C} \Rightarrow xc(x) \in \mathcal{C}.$$

- Thus we need to do congruences mod $x^n - 1$.
- Hence, a linear code $\mathcal{C}$ is cyclic if:

$$c(x) \in \mathcal{C} \Rightarrow xc(x) \bmod (x^n - 1) \in \mathcal{C}.$$

- So when we interpret codewords of a cyclic code as polynomials, we must perform the operations mod $x^n - 1$.

Proposition

Let $\mathcal{C}$ be a cyclic code of length n over $\mathbb{F}_q$.

1. If $c(x) \in \mathcal{C}$ and $p(x) \in \mathbb{F}_q[x]$, then $p(x)c(x) \in \mathcal{C}$.
2. There exists a unique polynomial $g(x) \in \mathcal{C}$ of degree $r < n$ such that $\mathcal{C} = \{p(x)g(x) \mid p(x) \in \mathbb{F}_q[x]\}$. $g(x)$ is the polynomial of least degree of $\mathcal{C}$.
3. $g(x)$ is a divisor of $x^n - 1$.
4. $\mathcal{C}$ has dimension $k = n - \deg(g) = n - r$.

Definition

$g(x)$ is called *the generating* polynomial of $\mathcal{C}$ and $h(x) = (x^n - 1)/g(x)$ is called the *parity-check* polynomial.

Cyclic Codes: Properties

Remarks

- As $g(x) \mid (x^n - 1)$, there are as many cyclic codes of length n as 2^m, where m is the number of irreducible factors of $x^n - 1 \in \mathbb{F}_q[x]$.
- The codewords have the following shape ($r = \deg(g)$):

$$c(x) = (p_0 + p_1 x + \cdots + p_{n-r-1} x^{n-r-1}) g(x).$$

- The information symbols are $p_0 p_1 \ldots p_{n-r-1}$, so the previous equality can be seen as an encoding mapping for $\mathcal{C}$ (but not a systematic one).
- $c(x) \in \mathcal{C} \iff c(x)h(x) = 0 \pmod{x^n - 1}$.

Let $\mathcal{C}$ be a cyclic code with generating and parity-check polynomials:

$$g(x) = g_0 + g_1 x + \cdots + g_{n-k} x^{n-k}$$
$$h(x) = h_0 + h_1 x + \cdots + h_k x^k$$

From g and h we can write a generating matrix and a parity-check matrix for $\mathcal{C}$:

- $G = \begin{bmatrix} g_0 & g_1 & \dots & g_{n-k} & 0 & \dots & 0 \\ & & & \dots & & & \\ 0 & \dots & 0 & g_0 & g_1 & \dots & g_{n-k} \end{bmatrix}$
- $H = \begin{bmatrix} h_k & h_{k-1} & \dots & h_0 & 0 & \dots & 0 \\ & & & \dots & & & \\ 0 & \dots & 0 & h_k & h_{k-1} & \dots & h0 \end{bmatrix}$

A non-systematic encoding

If $g(x)$ is the generator polynomial, then:

$$p(x) = p_0 + p_1 x + \cdots + p_{k-1} x^{k-1} \mapsto p(x)g(x)$$

is a non-systematic encoding mapping.

A systematic encoding

1. Multiply $p(x)$ by x^{n-k}.
2. Divide $x^{n-k}p(x)$ by $g(x)$: $x^{n-k}p(x) = g(x)q(x) + r(x)$, where $\deg(r) < \deg(g)$.
3. Encode $p(x)$ as $x^{n-k}p(x) - r(x)$:

$$x^{n-k}p(x) - r(x) = (-r_0, -r_1, \ldots, -r_{n-k-1}, p_0, p_1, \ldots, p_{k-1}).$$

Example

In $\mathbb{F}_2[x]$: $x^7 - 1 = (1 + x + x^3)(1 + x + x^2 + x^4)$.

Let $\mathcal{C}$ be the cyclic code of length 7 with $g(x) = 1 + x + x^3$ and $h(x) = 1 + x + x^2 + x^4$.

$p(x) = 1 + x^3$ can be encoded as follows:

- non-systematic encoding:
$$1 + x^3 \mapsto (1 + x^3)g(x) = 1 + x + x^4 + x^6$$
$$1001 \mapsto 11001010.$$
- systematic encoding:
  1. $(1 + x^3)x^3 = x^3 + x^6$;
  2. $x^6 + x^3 = g(x)(x + x^3) + (x + x^2)$;
  3. and finally: $1 + x^3 \mapsto (x + x^2) + (x^3 + x^6) = 0111001$

Cyclic Redundancy Code: CRC

- When a cyclic code is used for error-detecting is called a cyclic redundancy code or a CRC.
- All generator polynomials have the form $(1+x)\overline{g}(x)$, so a CRC contains the even binary code (all codewords have an even number of ones — prove it!).
- CRC are used in:
  - popular error-detecting signatures,
  - checksums,
  - TCP-IP,
  - disk interfaces, network software and hardware, program loaders, backup software, revision-control systems, etc

Remarks

- The number k of information symbols can change from one data packet to another one. But the number of parity-check bits is always the same: the degree of $g(x)$.
- For example, in the Ethernet protocol, $32 = n - k$. In this protocol, the length of the data packet k can varied from 512 to 12144 bits.
- Minimum distance d in function of the code length n:

n	90–123	124–203	204–300	301–3006	3007–12144
d	8	7	6	5	4

Some Standard CRC

CRC-4	$x^4 + x^3 + x^2 + x + 1$
CRC-7	$x^7 + x^6 + x^4 + 1$
CRC-8	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$
CRC-8 (GSM)	$x^8 + x^7 + x^4 + x^3 + x + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
CRC-16 (ANSI)	$x^{16} + x^{15} + x^2 + 1$
CRC-16 (CCITT)	$x^{16} + x^{12} + x^5 + 1$
CRC-16 (SDLC)	$x^{16} + x^{15} + x^{13} + x^7 + x^4 + x^2 + x + 1$
CRC-24	$x^{24} + x^{23} + x^{14} + x^{12} + x^8 + 1$
CRC-24 (GSM 3rd gen.)	$x^{24} + x^{23} + x^6 + x^5 + x + 1$
CRC-32 (Ethernet)	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11}$ $+ x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Example: CRC-16 CCITT

- The recommended 16-bit CRC by the CCITT (International Communication Standards) has generator polynomial:

$$g(x) = x^{16} + x^{12} + x^5 + 1 = \texttt{0x11021}$$

- A message is considered as a long binary string and then as a binary polynomial $m(x)$.
- The CRC-16 of $m(x)$ is the remainder of the division of $m(x)$ by `0x11021`.
- For example: the ASCII string corresponding to 'DOG' is `0110 0100 0110 1111 0110 0111` or the binary polynomial $x^{22}+x^{21}+x^{18}+x^{14}+x^{13}+x^{11}+x^{10}+x^9+x^8+x^6+x^5+x^2+x+1$. Its CRC-16 is $x^{14} + x^9 + x^8 + x^6 + x^2 + 1 = \texttt{0x8389}$.

Example: CRC-16 CCITT

- To compute the CRC-16 of a message $m(x)$, we read a bit at a time and compute the new CRC from the last CRC computed.
- If $m(x) = g(x)q(x) + r(x)$, with $\deg(r) \leq 15$, and b is the last bit read, then the new message is $m(x)x + b$ and:

$$\begin{aligned} m(x)x + b &= g(x)q(x)x + r(x)x + b \\ &\equiv r(x)x + b \pmod{g(x)} \end{aligned}$$

- That is, the CRC-16 of the new message is the same as the CRC-16 of $r(x)x + b$. So we shift the old CRC-16 one position to the left and then make a XOR with b.

Example: CRC-16 CCITT

This is what the following C function does.

```c
short BitwiseCRC16 (int bit, short crc) {
  long longcrc = crc ;
  longcrc = (longcrc << 1)^bit ; /* next bit */
  if (longcrc & 0x10000)
    longcrc ^= 0x11021 ; /* reduce */
  return longcrc ;
}
```

Remarks

- As cyclic codes are linear, we can use the general algorithm of syndromes and leaders for decoding them.
- If a word y has an error in position i, then applying $n-1-i$ cyclic shifts to y we can put that error in position $n-1$.
- So it is enough to construct the table of syndromes-leaders for those leaders of weight $\leq \rho$ (incomplete decoding) that have an error in the last position.
- Meggitt's algorithm is based on this observation.

Let $\mathcal{C}$ be a binary cyclic code of length n with generating polynomial $g(x)$.

Definition

The syndrome $s(y)$ a word $y(x) \in \mathbb{F}_2[x]_n$ is the remainder of the division of $y(x)$ by $g(x)$.

Theorem

If $y^{(i)}(x)$ is i-th cyclic shift of $y(x)$, then:

$$s(y^{(i)}(x)) = s(x^i s(y(x))).$$

That is, if we apply a cyclic shift to $y(x)$, the syndrome of the new word is obtained multiplying $s(y)$ by x and reducing the result mod $g(x)$.

Meggitt's Algorithm

Preprocessing

Compute the list L of leaders of degree $n-1$ and weight $t \leq \rho$ together with their syndromes.

Algorithm

Let $s(x) = s(y(x))$ be the syndrome of the word received $y(x)$.

1. If $s(x) = 0$: no error.
2. $i = n - 1$.
   While $i \geq 0$ do:
   - if $s(x) \in L$: correct position i
   - $s(x) := s(xs(x))$
   - $i = i - 1$

Meggitt's Algorithm: Example

- $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ generates a cyclic code with $n = 15$, $k = 7$ and $d = 5$ ($\rho = 2$).
- The list L contains the leaders of degree 14 and weight ≤ 2; that is, the leaders of the form x^{14} and $x^i + x^{14}$, with $0 \leq i \leq 13$.

$\ell(x)$	$s(x)$	$\ell(x)$	$s(x)$
x^{14}	$x^7 + x^6 + x^5 + x^3$	$x^7 + x^{14}$	$x^6 + x^5 + x^3$
$1 + x^{14}$	$x^7 + x^6 + x^5 + x^3 + 1$	$x^8 + x^{14}$	$x^5 + x^4 + x^3 + 1$
$x + x^{14}$	$x^7 + x^6 + x^5 + x^3 + x$	$x^9 + x^{14}$	$x^7 + x^4 + x^3 + x + 1$
$x^2 + x^{14}$	$x^7 + x^6 + x^5 + x^3 + x^2$	$x^{10} + x^{14}$	$x^3 + x^2 + 1$
$x^3 + x^{14}$	$x^7 + x^6 + x^5$	$x^{11} + x^{14}$	$x^7 + x^6 + x^5 + x^4 + x^2 + 1$
$x^4 + x^{14}$	$x^7 + x^6 + x^5 + x^4 + x^3$	$x^{12} + x^{14}$	$x^7 + x^6 + x^4 + x$
$x^5 + x^{14}$	$x^7 + x^6 + x^3$	$x^{13} + x^{14}$	$x^7 + x^4 + x^3 + x^2$
$x^6 + x^{14}$	$x^7 + x^5 + x^3$		

Meggitt's Algorithm: Example

- Let us apply the algorithm to:

$$y(x) = 1 + x^2 + x^5 + x^6 + x^7 + x^8 + x^9 + x^{10} + x^{14}$$
$$s(y) = 1 + x^3 + x^6 \notin L$$

i	$s(x)$
14	$x^6 + x^3 + 1$
13	$x^7 + x^4 + x$
12	$x^7 + x^6 + x^5 + x^4 + x^2 + 1 \in L \quad (*)$
$\ldots$	$\ldots$
9	$x^7 + x^6 + x^5 + x^3 + x^2 \in L$
$\ldots$	$\ldots$

- There are two errors in positions 9 and 12. Corrected word: $1 + x^2 + x^5 + x^6 + x^7 + x^8 + x^{10} + x^{12} + x^{14}$.
- If we look at the leader corresponding to the step (*), we catch the second error!

Definition

The roots of the polynomial $x^n - 1$ in some finite field $\mathbb{F}_{2^m}$ are called the *roots of unity*.

Properties

- If n is odd, there are exactly n distinct roots of unity.
- There exists a root of unity ω such that all roots of unity are $1, \omega, \omega^2, \ldots, \omega^{n-1}$.
- ω is called a *primitive* root of unity.

Factoring $x^n - 1$

Strategy for factoring $x^n - 1$: look for the smallest $\mathbb{F}_{2^m}$ that contains all the roots of unity.

- If $\omega \in \mathbb{F}_{2^m}$ is a primitive root of unity, then $n = \mathrm{ord}(\omega) \mid (2^m - 1)$.
- Take $m \geq 1$ as the least integer such that $2^m \equiv 1 \pmod{n}$. m is called the *order* of 2 mod n.
- Let $\alpha \in \mathbb{F}_{2^m}$ be a primitive element (so of order $2^m - 1$).
- If $r = (2^m - 1)/n$, then the order of $\omega = \alpha^r$ is n.
- We can factor $x^n - 1$ as:

$$\begin{aligned} x^n - 1 &= (x-1)(x-\omega)\cdots(x-\omega^{n-1}), && \text{over } \mathbb{F}_{2^m}[x] \\ x^n - 1 &= f_1(x)f_2(x)\cdots f_s(x), && \text{over } \mathbb{F}_2[x] \end{aligned}$$

Cyclotomic Classes mod n

Proposition

Let $p(x) \in \mathbb{F}_2[x]$ and assume that $p(\beta) = 0$, where $\beta \in \mathbb{F}_{2^m}$. Then $p(\beta^2) = 0$, $p(\beta^4) = 0$, and so on.

Definition

The cyclotomic class of an integer $j \in \mathbb{Z}_n$ is the set:

$$C_j = \{j, 2j, 2^2 j, \ldots, 2^{r-1} j\} \bmod n$$

where r is the least integer ≥ 1 such that $2^r j \equiv j \pmod{n}$.

Cyclotomic Classes mod n

Remarks

- The cyclotomic classes mod n form a partition of $\mathbb{Z}_n$.
- To compute the factors $f_i(x)$ we group the roots of unity whose exponents belong to the same cyclotomic class.
- There is an irreducible factor of $x^n - 1$ over $\mathbb{F}_2[x]$ for each cyclotomic class mod n.

Examples of Cyclotomic Classes

Cyclotomic classes mod 7 and irreducible factors of $x^7 - 1$

$C_0 = \{0\}$, $C_1 = \{1, 2, 4\}$, $C_3 = \{3, 6, 5\}$.

Factors: one of degree 1 and two of degree 3.

Cyclotomic classes mod 11 and irreducible factors of $x^{11} - 1$

$C_0 = \{0\}$, $C_1 = \{1, 2, 4, 8, 5, 10, 9, 7, 3, 6\}$

Factors: $x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$ and $x - 1$.

Cyclotomic classes mod 23 and irreducible factors of $x^{23} - 1$

$C_0 = \{0\}$, $C_1 = \{1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12\}$,
$C_5 = \{5, 10, 20, 17, 11, 22, 21, 19, 15, 7, 14\}$.

Factors: three factors of degrees 1, 11 and 11.

Theorem

Then the irreducible factors of $x^n - 1$ are the polynomials:

$$f_C(x) = \prod_{j \in C} (x - \omega^j),$$

where C is a cyclotomic class mod n.

Factorization of $x^7 - 1$ in $\mathbb{F}_2[x]$.

- Order of 2 mod 7 is 3.
- Now: $r = (2^3 - 1)/7 = 1$. Take $h(x) = 1 + x^2 + x^3$ and then:

$$\mathbb{F}_8 = \mathbb{F}_2[x]/h(x), \quad \alpha = \bar{x}, \quad \omega = \alpha^1 = \alpha.$$

- Finally:

$$\begin{aligned}
f_{C_0} &= x - 1 \\
f_{C_1} &= (x - \alpha)(x - \alpha^2)(x - \alpha^4) = x^3 + x^2 + 1 \\
f_{C_3} &= (x - \alpha^3)(x - \alpha^5)(x - \alpha^6) = x^3 + x + 1
\end{aligned}$$

Factorization of $x^{23} - 1$ in $\mathbb{F}_2[x]$.

- Order of 2 mod 23 is 11.
- Now: $r = (2^{11} - 1)/23 = 89$. If $\alpha \in \mathbb{F}_{2^{11}}$ is a primitive element, then $\omega = \alpha^{89}$ is a 23th root of unity.
- Finally:

$$
\begin{aligned}
f_{C_0} &= x - 1 \\
f_{C_1} &= \prod_{j \in C_1} (x - \omega^j) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1 \\
f_{C_5} &= \prod_{j \in C_5} (x - \omega^j) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1
\end{aligned}
$$

Binary BCH Codes

Roots of a Cyclic Code

Let $\mathcal{C}$ be a binary cyclic code of odd length n with generating polynomial $g(x) \mid (x^n - 1)$.

- Let $\mathbb{F}_{2^m}$ be a finite field where $x^n - 1$ decomposes $(n \mid (2^m - 1))$. Then:
$$g(x) = (x - \alpha_1) \cdots (x - \alpha_r), \quad r = \deg(g),$$
where $\alpha_1, \ldots, \alpha_r \in \mathbb{F}_{2^m}$ are r distinct elements.
- Then the code $\mathcal{C}$ can be described as follows:
$$c(x) \in \mathcal{C} \iff c(\alpha_i) = 0, \quad i = 1, \ldots, r$$

Definition

The elements $\alpha_1, \ldots, \alpha_r \in \mathbb{F}_{2^m}$ are called the *roots* of $\mathcal{C}$ and $\{\alpha_1, \ldots, \alpha_r\}$ is called the *root set* of $\mathcal{C}$.

Remark

- If ω is a root of unity, then the roots of a cyclic code can be written as $\alpha_1 = \omega^{i_1}, \dots, \alpha_r = \omega^{i_r}$
- Can we choose the exponents $i_1, \dots, i_r$ such that $\mathcal{C}$ has *good properties*? This means: can we control the minimum distance by means of the exponents?

Proposition

If there are $\delta - 1$ consecutive powers of ω among the roots of $\mathcal{C}$, then $d(\mathcal{C}) \geq \delta$.

Definition

This parameter δ is called the *designed distance* of the code.

Data to build a BCH code

- n: an odd integer (the length).
- $m \geq 1$: an integer such that $2^m \equiv 1 \pmod{n}$ (so that $x^n - 1$ factors in $\mathbb{F}_{2^m}$).
- $\omega \in \mathbb{F}_{2^m}$: a primitive root of unity: $\omega^n = 1$.
- $\delta \geq 2$: the designed distance.
- $\ell \geq 1$: an offset.

Definition

Define $\mathrm{BCH}_\omega(\delta, \ell)$ as the binary cyclic code of length n whose root set is the smallest set containing the roots:

$$\omega^\ell, \omega^{\ell+1}, \ldots, \omega^{\ell+\delta-2}.$$

Strict and primitive BCH codes

- If $\ell = 1$, the BCH code is called *strict*.
- If $n = 2^m - 1$ and $\omega = \alpha$ is a primitive element of $\mathbb{F}_{2^m}$, the BCH code is called *primitive*.

- For strict and primitive BCH codes, the data are: the length $n = 2^m - 1$, the designed distance δ, $\omega = \alpha \in \mathbb{F}_{2^m}$ a primitive element.
- In this case the roots of the BCH code are at least $\alpha, \alpha^2, \ldots, \alpha^{\delta-2}$.
- The minimum distance is $d \geq \delta$.
- The dimension k can be easily computed in every concrete case.

Examples

Let $q = 2$, $m = 4$, $\alpha \in \mathbb{F}_{16}$ primitive. The cyclotomic classes mod 15 are $C_1 = \{1, 2, 4, 8\}$, $C_3 = \{3, 6, 12, 9\}$, $C_5 = \{5, 10\}$ and $C_7 = \{7, 14, 13, 11\}$.

- We have:
$$\mathrm{BCH}(\delta = 4) = \mathrm{BCH}(\delta = 5) \Rightarrow d_4 = d_5 \geq 5$$
$$\mathrm{BCH}(\delta = 6) = \mathrm{BCH}(\delta = 7) \Rightarrow d_6 = d_7 \geq 7$$
- The roots of BCH(5) are $\alpha, \alpha^2, \alpha^3, \alpha^4$. The cyclotomic classes involved are C_1 and C_3. Hence the generating polynomial is $g(x) = f_{C_1}(x) f_{C_3}(x)$ whose degree is $|C_1| + |C_3| = 8$.
- Moreover, $c(x) \in \mathrm{BCH}(5)$ iff $c(\alpha) = c(\alpha^3) = 0$ (a power from each cyclotomic class is enough).

Let $\mathcal{B}$ be a binary, strict and primitive BCH code of length n.

- Write the word received $y(x)$ as:

$$y(x) = c(x) + e(x), \quad e(x) = \sum_{i=0}^{n-1} e_i x^i = (e_0, e_1, \dots, e_{n-1})$$

  where $c(x) \in \mathcal{B}$ and $e(x)$ is the error vector (or polynomial).

- Assume that $1 \leq |e| = t \leq \rho$. Write:

$$e(x) = x^{i_1} + x^{i_2} + \cdots + x^{i_t}, \quad (i_1 < i_2 < \cdots < i_t).$$

Error locators and error location polynomial

Error locators: $\eta_1 = \alpha^{i_1}, \eta_2 = \alpha^{i_2}, \ldots, \eta_t = \alpha^{i_t}$.
Error location polynomial: $\ell(x) = (1 - \eta_1 x) \cdots (1 - \eta_t x)$.

Remark

The polynomial $\ell(x)$ has t distinct roots, the inverses of the error locators η_j.
Hence: $\gcd(\ell(x), \ell'(x)) = 1$.

The Key Equation

Syndrome

We define the syndrome of the word $y(x)$ as the polynomial $s(x) = \sum_{i=0}^{\delta-2} s_i x^i$, where:

$$s_i = e(\alpha^{i+1}) = y(\alpha^{i+1}).$$

Key Equation

1. The polynomials $\ell(x)$ and $s(x)$ satisfy the *key equation*

$$\ell(x)s(x) \equiv \ell'(x) \pmod{x^{\delta-1}}.$$

2. If there is a polynomial $\ell(x)$ of degree $\leq t = \lfloor(\delta-1)/2\rfloor$ and distinct roots satisfying this equation, then it is unique up to a factor of $\mathbb{F}_{2^m}$.

Remark

- Write the Key Equation as $a(x)x^{\delta-1} + \ell(x)s(x) = \ell'(x)$, for some $a(x)$.
- A typical step of Euclid's algorithm produces $a_k(x)$, $b_k(x)$ and $r_k(x)$ such that $a_k(x)x^{\delta-1} + b_k(x)s(x) = r_k(x)$.
- If we want $r_k(x) = \ell'(x)$ and $b_k(x) = \ell(x)$, then we must have:
  $$\deg r_k(x) < \frac{\delta-1}{2}, \quad \deg b_k(x) \leq \frac{\delta-1}{2}$$
  and hence $\deg r_k(x) < (\delta-1)/2$ and $\deg r_{k-1}(x) \geq (\delta-1)/2$.
- We can set $\ell(x) = \frac{b_k(x)}{b_k(0)}$.

Decoding Algorithm: Euclidean Algorithm

1. Compute the syndrome polynomial:
$$s(x) = s_0 + s_1 x + \cdots + s_{\delta-2}x^{\delta-2},$$
where $s_i = y(\alpha^{i+1})$, $i = 0, \ldots, \delta - 2$.
2. If $s(x) = 0$, then there is no error.
3. Otherwise, apply Euclid's algorithm to $x^{\delta-1}$ and $s(x)$ until we get polynomials $r_k(x)$, $a_k(x)$ and $b_k(x)$ such that:
$$r_k(x) = a_k(x)x^{\delta-1} + b_k(x)s(x)$$
and $\deg r_k(x) < (\delta - 1)/2$ and $\deg r_{k-1}(x) \geq (\delta - 1)/2$.
4. Let $\ell(x) = b_k(x)/b_k(0)$. This is the error locator polynomial.
5. Find its roots $\xi_1, \ldots, \xi_t$.
6. Then $\eta_i = \xi_i^{-1}$ are the error locators and $\log_\alpha(\eta_i)$ the error positions.

Example of Decoding

The code $\mathcal{B}$

$\mathcal{B} = \text{BCH}(7)$ of length $15 = 2^4 - 1$. $\mathbb{F}_{16} = \mathbb{F}_2[x]/(x^4 + x + 1)$, $\alpha \in \mathbb{F}_{16}$ is primitive. Then $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6$ are among the roots of $\mathcal{B}$. The cyclotomic classes of 2 mod 15 that we need are $C_1 = \{1, 2, 4, 8\}$, $C_3 = \{3, 6, 9, 12\}$ and $C_5 = \{5, 10\}$. So $g(x) = f_{C_1}(x) f_{C_3}(x) f_{C_5}(x)$, its degree is 10, and $k = \dim \mathcal{B} = 5$.

Assume we receive the word:

$$\begin{aligned} y &= 111011100000110 \\ &= 1 + x + x^2 + x^4 + x^5 + x^6 + x^{12} + x^{13}. \end{aligned}$$

Example of Decoding

Compute the syndrome polynomial:

$$
\begin{aligned}
s_0 &= y(\alpha) = 1 + \alpha + \alpha^2 + \alpha^4 + \alpha^5 + \alpha^6 + \alpha^{12} + \alpha^{13} = \alpha^6 \\
s_1 &= y(\alpha^2) = y(\alpha)^2 = \alpha^{12} \\
s_2 &= y(\alpha^3) = 1 + \alpha^3 + \alpha^6 + \alpha^{12} + \alpha^{15} + \alpha^{18} + \alpha^{36} + \alpha^{39} = \alpha^8 \\
s_3 &= y(\alpha^4) = y(\alpha)^4 = \alpha^9 \\
s_4 &= y(\alpha^5) = 1 + \alpha^5 + \alpha^{10} + \alpha^{20} + \alpha^{25} + \alpha^{30} + \alpha^{60} + \alpha^{65} = \alpha^{10} \\
s_5 &= y(\alpha^6) = y(\alpha^3)^2 = \alpha
\end{aligned}
$$

That is: $s(x) = \alpha^6 + \alpha^{12}x + \alpha^8x^2 + \alpha^9x^3 + \alpha^{10}x^4 + \alpha x^5$.

Apply Euclid's algorithm to x^6 and $s(x)$ until we get a remainder of degree less than $(\delta - 1)/2 = 3$.

$$
\begin{aligned}
r_{-1}(x) &= x^6 & & \\
r_0(x) &= s(x) & q_1(x) &= \alpha^{14}x + \alpha^8 \\
r_1(x) &= \alpha^{13}x^4 + \alpha^{12}x^3 + \alpha^6x^2 + \alpha^{14} & q_2(x) &= \alpha^3x + \alpha^7 \\
r_2(x) &= \alpha^4x^3 + \alpha^3x^2 + \alpha^7x & q_3(x) &= \alpha^9x \\
r_3(x) &= \alpha^{11}x^2 + \alpha^{14} & &
\end{aligned}
$$

Example of Decoding

Now we compute the polynomials $b_k(x)$:

$$
\begin{aligned}
b_{-1}(x) &= 0 \\
b_0(x) &= 1 \\
b_1(x) &= q_1(x) = \alpha^{14}x + \alpha^8 \\
b_2(x) &= b_0(x) - b_1(x)q_2(x) = \alpha^2x^2 + \alpha x \\
b_3(x) &= b_1(x) - b_2(x)q_3(x) = \alpha^{11}x^3 + \alpha^{10}x^2 + \alpha^{14}x + \alpha^8
\end{aligned}
$$

and $b_3(0) = \alpha^8$, so the error locator polynomial is:

$$
\ell(x) = \frac{b_3(x)}{b_3(0)} = \alpha^3x^3 + \alpha^2x^2 + \alpha^6x + 1.
$$

Finally we find the roots of $\ell(x)$:

$$\ell(1) = \alpha^3 + \alpha^2 + \alpha^6 + 1 = 1 \qquad \ell(\alpha^3) = \alpha^{12} + \alpha^8 + \alpha^9 + 1 = 1$$
$$\ell(\alpha) = \alpha^6 + \alpha^4 + \alpha^7 + 1 = \alpha^8 \qquad \ell(a^4) = 1 + \alpha^{10} + \alpha^{10} + 1 = 0$$
$$\ell(\alpha^2) = \alpha^9 + \alpha^6 + \alpha^8 + 1 = \alpha$$

So α^4 is a root and $\eta_1 = \alpha^{-4} = \boxed{\alpha^{11}}$ is an error locator.

Now we divide $\ell(x)$ by $x - \alpha^4$ and the quadratic equation we get has roots α^{10} and α^{13}. So the other error locators are $\eta_2 = \alpha^{-10} = \boxed{\alpha^5}$ and $\eta_3 = \alpha^{-13} = \boxed{\alpha^2}$.

Conclusion: there are 3 errors at positions: 2, 5 and 11; and the most likely word sent is:

$$y(x) - (x^2 + x^5 + x^{11}) = 11\mathbf{0}01\mathbf{0}10000\mathbf{1}110.$$

Reed-Solomon Codes

Reed-Solomon (RS) Codes

- Introduced by Reed and Solomon in 1960.
- First efficient decoding algorithm by Berlekamp (1968) and Massey (1969)
- RS codes are special BCH codes, so also cyclic codes (and linear codes)
- These codes are Maximum Distance Separable (MDS) codes: d is maximum for fixed n and k: $d = n - k + 1$.
- RS codes are never binary codes: they are defined over some $\mathbb{F}_{2^r}$, $r > 1$.
- Applications: used by NASA (from Voyager 1977) and the European Space Agency; CD-ROM, Audio Compact Disc, and DVD-ROM; Pay per view TV and TDT...

Finite Fourier Transform (FFT)

- $r \geq 2$ an integer, $q = 2^r$, $n = q - 1$.
- $\beta \in \mathbb{F}_q^*$ a primitive element: $\beta^{q-1} = \beta^n = 1$. (Observe that β^{-1} is also primitive.)
- $\mathbb{F}_q[x]_n = \{p(x) \in \mathbb{F}_q[x] \mid \deg p(x) < n\}$.

The FFT of a polynomial

Define the mapping $\mathcal{F}_\beta : \mathbb{F}_q[x]_n \longrightarrow \mathbb{F}_q[x]_n$ as:

$$\mathcal{F}_\beta(a(x)) = \sum_{i=0}^{n-1} a(\beta^i) x^i$$

The polynomial $A(x) = \mathcal{F}_\beta(a(x))$ is called the *Finite Fourier Transform* (FFT) of $a(x)$.

Properties of the mapping $\mathcal{F}_\beta$

- $\mathcal{F}_\beta$ is a bijective linear mapping.
- The inverse mapping is:

$$\mathcal{F}_{\beta^{-1}}(A(x)) = \sum_{j=0}^{n-1} A(\beta^{-j})x^j$$

- $a(x) = \mathcal{F}_{\beta^{-1}}(A(x))$ is called the inverse FFT of $A(x)$.

Let k be an integer such that $1 \leq k \leq n$ and $d = n - k + 1$.

Definition

The RS code $\mathcal{R} = \mathcal{R}(2^r, d)$ is the set of polynomials of degree less than n that are the FFT of polynomials of degree less than k:

$$\mathcal{R} = \{\mathcal{F}_\beta(a(x)) : a(x) \in \mathbb{F}_q[x]_k\}$$

Remark

Hence, $A(x) \in \mathcal{R}$ if, and only if, its inverse FFT $\mathcal{F}_{\beta^{-1}}(A(x))$ has degree less than k. That is, if:

$$A(\beta) = A(\beta^2) = \cdots = A(\beta^{d-1}) = 0$$

(Observe that $\beta^{-j} = \beta^{n-j}$, because $\beta^n = 1$.)

Properties

- Parameters: $\mathcal{R}$ has length $n = q - 1$, dimension k and minimum distance $d = n - k + 1$.
- Syndrome polynomial of $A(x)$:

$$s(x) = A(\beta) + A(\beta^2)x + \cdots + A(\beta^{d-1})x^{d-2}.$$

- Encoding: to encode a polynomial of degree less than k, we compute its FFT (non-systematic encoding!):

$$a(x) \in \mathbb{F}_q[x]_k \mapsto A(x) = \sum_{i=0}^{n-1} a(\beta^i)x^i \in \mathbb{F}_q[x]_n.$$

Properties

- A Reed-Solomon code is a cyclic code:

$$A(x) \in \mathcal{R} \iff A(\beta) = \cdots = A(\beta^{d-1}) = 0$$

  which is equivalent to $A(x)$ being a multiple of the generating polynomial:

$$g(x) = (x - \beta) \cdots (x - \beta^{d-1}).$$

- A Reed-Solomon code is BCH code over $\mathbb{F}_q$, with designed distance d, because its generating polynomial vanishes at $d - 1 = n - k$ consecutive powers of β, which is a root of unity.

Example: $\mathcal{R}(16, 9)$. Encoding

Let $\mathbb{F}_{16} = \mathbb{F}_2[x]/(x^4 + x + 1)$. $\alpha = \bar{x} \in \mathbb{F}_{16}$ is a primitive element. Consider the Reed-Solomon code $\mathcal{R}(16, 9)$, with parameters: $n = 15$, $k = 16 - 9 = 7$, $d = 9$.

The word $m = \alpha^6 0 \alpha^5 1 \alpha 0 \alpha^2$, or as polynomial:

$$m(x) = \alpha^6 + \alpha^5 x^2 + x^3 + \alpha x^4 + \alpha^2 x^6,$$

is encoded as its FFT $M(x) = \sum_{j=0}^{14} m(\alpha^j) x^j$:

$$M(x) = \alpha^{13} + \alpha^6 x + \alpha^{14} x^2 + \alpha^4 x^3 + \alpha^2 x^4 + \alpha^2 x^5 + \alpha x^6 + \alpha^4 x^7 + \\ + \alpha^2 x^8 + \alpha^4 x^9 + \alpha^{12} x^{11} + \alpha^4 x^{12} + \alpha^{10} x^{13} + \alpha^9 x^{14}$$

Example: $\mathcal{R}(16,9)$. Decoding

Consider the word: $N = \alpha^5\alpha^9 0001\alpha^4\alpha^3\alpha 000\alpha^{13}\alpha\alpha^8$, or as polynomial:

$N(x) = \alpha^5 + \alpha^9 x + x^5 + \alpha^4 x^6 + \alpha^3 x^7 + \alpha x^8 + \alpha^{13} x^{12} + \alpha x^{13} + \alpha^8 x^{14}$.

- First, compute the inverse FFT $n(x) = \sum_{j=0}^{14} N(\alpha^{-j})x^j$:

$$n(x) = \alpha^{11} + \alpha x + \alpha^4 x^3 + \alpha^{13} x^4 + \alpha^{12} x^5 + \alpha^8 x^6 + \alpha x^7 + \alpha^{14} x^8 + \\ + \alpha^4 x^9 + \alpha^{14} x^{10} + \alpha^{11} x^{11} + \alpha^6 x^{12} + \alpha^8 x^{13} + \alpha^3 x^{14}$$

Example: $\mathcal{R}(16, 9)$. Decoding

- As $\deg(n(x)) \geq 8$, we know there have been errors.
- The syndromes are the last eight coefficients of $n(x)$:

$$s_0 = \alpha^3, \quad s_1 = \alpha^8, \quad s_2 = \alpha^6, \quad s_3 = \alpha^{11},$$
$$s_4 = \alpha^{14}, \quad s_5 = \alpha^4, \quad s_6 = \alpha^{14}, \quad s_7 = \alpha$$

- Write the syndrome polynomial:

$$s(x) = \alpha x^7 + \alpha^{14} x^6 + \alpha^4 x^5 + \alpha^{14} x^4 + \alpha^{11} x^3 + \alpha^6 x^2 + \alpha^8 x + \alpha^3$$

Setting

- We send $A(x) \in \mathcal{R}$ and we get $B(x) = A(x) + E(x)$, where $E(x)$ is the error introduced by the channel.
- Apply the inverse FFT to the equality $B(x) = A(x) + E(x)$:

$$b(x) = a(x) + e(x)$$

  where $\deg a(x) \leq k - 1$, so we have that:

$$e_k = b_k, \ldots, e_{n-1} = b_{n-1}.$$

- These are the coefficients of the syndrome polynomial.
- We only have to compute the coefficients e_j, for $j = 0, \ldots, k - 1$.

Error Locator and Evaluating Polynomials

- The *error locator* polynomial has degree t (the number of errors) and vanishes at β^i iff an error has happened at position i:

$$\sigma(x) = \prod_i (x - \beta^i) = \sigma_0 + \sigma_1 x + \cdots + \sigma_{t-1} x^{t-1} + x^t = x^t \ell\left(\frac{1}{x}\right)$$

  where $\ell(x)$ is the polynomial introduced for BCH codes.

- The *evaluating* polynomial is used to compute the magnitudes ε_i of the errors:

$$ev(x) = \sum_{i=1}^{t} \eta_i \varepsilon_i \prod_{j \neq i} (1 - \eta_j x)$$

Error locator and evaluating polynomials: properties

1. In the binary case: $\varepsilon_i = 1$, for all i.
2. In the binary case: $ev(x) = \ell'(x)$.
3. $\deg \ell(x) = t \geq 1$, $\deg ev(x) < t$.
4. $\gcd(\ell(x), ev(x)) = 1$.
5. *Key Equation*: $ev(x) \equiv \ell(x)s(x) \pmod{x^{d-1}}$.
6. We can find $\ell(x)$ (and $\sigma(x)$) applying the extended Euclidean algorithm to x^{d-1} and $s(x)$.

A backward recurrence

- For RS codes, we solve a linear recurrence to find the unknown terms e_j, instead of finding the evaluating polynomial to compute the magnitudes of the errors.
- Set the initial conditions (already known):

$$e_{n-1} = b_{n-1}, \dots, e_k = b_k.$$

- Compute the remaining e_j from the backward recurrence:

$$e_j = \sigma_{t-1} e_{j+1} + \cdots + \sigma_0 e_{j+t}$$

  if $j = k-1, k-2, \dots, 0$, where the σ_i are the coefficients of the error locator polynomial $\sigma(x)$.

Example: $\mathcal{R}(16, 9)$. Decoding

- Recall that the syndrome polynomial is:

$$s(x) = \alpha x^7 + \alpha^{14}x^6 + \alpha^4x^5 + \alpha^{14}x^4 + \alpha^{11}x^3 + \alpha^6x^2 + \alpha^8x + \alpha^3.$$

- We apply the extended euclidean algorithm to $x^{d-1} = x^8$ and $s(x)$ until we get a remainder of degree less than $(d-1)/2 = 4$.

$$\begin{aligned}
r_{-1} &= x^8 & & \\
r_0 &= s(x) & q_1 &= \alpha^{14}x + \alpha^{11} \\
r_1 &= \alpha^5x^6 + \alpha^{12}x^5 + \alpha^{14}x^4 + \alpha^4x^3 + \alpha^4x^2 + \alpha x + 1 & q_2 &= \alpha^{11}x + \alpha \\
r_2 &= \alpha^{14}x^5 + \alpha^{14}x^4 + \alpha^{14}x^3 + \alpha^8x^2 + \alpha^{12}x + \alpha^9 & q_3 &= \alpha^6x + 1 \\
r_3 &= \alpha^5x^4 + \alpha^4x^3 + \alpha^{11}x^2 + \alpha^6x + \alpha^7 & q_4 &= \alpha^9x + \alpha^{12} \\
r_4 &= \alpha^{13}x^3 + x^2 + \alpha^8x + \alpha^{14} & &
\end{aligned}$$

Example: $\mathcal{R}(16, 9)$. Decoding

- Now we compute the polynomial $b_4(x)$:

$$b_2 = 1 + q_1 q_2 = \alpha^{10} x^2 + \alpha^2 x + \alpha^6$$
$$b_3 = b_1 + b_2 q_3 = \alpha x^3 + \alpha x^2 + \alpha x + \alpha^4$$
$$b_4 = b_2 + b_3 q_4 = \alpha^{10} x^4 + \alpha^9 x^3 + \alpha^{13} x^2 + \alpha^2 x + \alpha^{11}$$

  So the number of errors is 4.

- Now we know that:

$$\ell(x) = \alpha^{-11} b_4(x) = \alpha^{14} x^4 + \alpha^{13} x^3 + \alpha^2 x^2 + \alpha^6 x + 1$$
$$\sigma(x) = x^4 \ell\left(\frac{1}{x}\right) = \alpha^{14} + \alpha^{13} x + \alpha^2 x^3 + \alpha^6 x^3 + t^4$$

  That is: $\sigma_0 = \alpha^{14}$, $\sigma_1 = \alpha^{13}$, $\sigma_2 = \alpha^2$, $\sigma_3 = \alpha^6$.

- The zeros of $\sigma(x)$ are $\alpha^3, \alpha^4, \alpha^9, \alpha^{13}$. So the error positions are: $3, 4, 9, 13$ (but this algorithm does not use this information).

- Instead we solve the linear recurrence:

$$\begin{aligned} e_j &= \sigma_3 e_{j+1} + \sigma_2 e_{j+2} + \sigma_1 e_{j+3} + \sigma_0 e_{j+4} \\ &= \alpha^6 e_{j+1} + \alpha^2 e_{j+2} + \alpha^{13} e_{j+3} + \alpha^{14} e_{j+4} \end{aligned}$$

with the initial conditions:

$$e_7 = \alpha, \quad e_8 = \alpha^{14}, \quad e_9 = \alpha^4, \quad e_{10} = \alpha^{14}, \ldots$$

- We get:

$$\begin{array}{llll} e_6 = 0, & e_5 = \alpha^{12}, & e_4 = \alpha^6, & e_3 = \alpha^{10}, \\ e_2 = 0, & e_1 = \alpha, & e_0 = \alpha^3 & \end{array}$$

Example: $\mathcal{R}(16, 9)$. Decoding

- Hence the inverse FFT of the error vector is:

$$e = (\alpha^3, \alpha, 0, \alpha^{10}, \alpha^6, \alpha^{12}, 0, \alpha, \alpha^{14}, \alpha^4, \alpha^{14}, \alpha^{11}, \alpha, \alpha^8, \alpha^3).$$

- Finally, the information symbols are:

$$n + e = (\alpha^5, 0, 0, \alpha^2, 1, 0, \alpha^9, 0, \ldots, 0).$$

- We can check our computations by calculating the FFT of this last word. We have to get the word N modified at positions $3, 4, 9, 13$. Indeed, this FFT is:

$$\alpha^{12}t^{13} + \alpha^{10}t^9 + \alpha^{14}t^4 + \alpha^{14}t^3.$$

A decoding algorithm for RS codes

Input: the received word $B(x)$.
Output: the information symbols or $*$ (more than ρ errors).

1. Compute the inverse FFT of $B(x)$: $b(x) = \mathcal{F}_{\beta^{-1}}(B(x))$.
2. Set $e_j = b_j$, for $j = k, \ldots, n-1$. The syndrome polynomial of $B(x)$ is $s(x) = \sum_{i=0}^{d-2} b_{i+k}x^i$.
3. Apply the euclidean algorithm to x^{d-1} and $s(x)$ to compute the error-locator polynomial $\ell(x)$ (of degree $t \leq \rho$, say).
4. If $\ell(0) = 0$, then stop and return $*$; else continue.
5. If $\ell(x)$ does not have t simple roots, then stop and return $*$; else continue.

A decoding algorithm for RS codes

6. Compute the coefficients of the reciprocal polynomial of $\ell(x)$:

$$\sigma(x) = x^t \ell\left(\frac{1}{x}\right) = \sum_{i=0}^{t} \sigma_i x^i$$

7. With the initial conditions computed in 2, solve the linear recurrence:

$$e_j = \sigma_{t-1} e_{j+1} + \cdots + \sigma_0 e_{j+t}, \quad j = k-1, \ldots, 0$$

8. Compute the polynomial $a(x) = b(x) + e(x)$.
9. If $\deg a(x) \geq k$, then stop and return $*$; else continue.
10. The information symbols are the coefficients of $a(x)$.