Coding Theory

(Cyclic codes)

Lector: Nikolai L. Manev

Institute of Mathematics and Informatics, Sofia, Bulgaria

Definition

Cyclic codes were first discussed by E. Prange in the period 1957 – 1959. They are important practical error control codes for a variety of reasons. For example, they can be implemented using high-speed shift-register-based encoder and decoders that is of great interest in many applications.

Definition. A code $\mathcal{C}$ is *cyclic* if any cyclic shift of a codeword is also a codeword, that is, $(c_1, \ldots, c_n) \in \mathcal{C}$ implies $(c_n, c_1, \ldots, c_{n-1}) \in \mathcal{C}$.

Example. The binary [3,2,2] code $\mathcal{C} = \{000,\ 110,\ 011,\ 101\}$ is cyclic.

Polynomial representation 1

Consider the quotient ring $\mathcal{F}_n = \mathbb{F}[x]/(x^n - 1)$, $\mathbb{F} = GF(q)$. We associate any vector $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{F}^n$ with the polynomial $v(x) = v_0 + v_1 x + \cdots + v_{n-1} x^{n-1} \in \mathcal{F}_n$

Hence, we define an one-to-one mapping between $\mathbb{F}^n$ and $\mathcal{F}_n$. It is easy to check that this mapping is an isomorphism $\mathbb{F}^n \cong \mathcal{F}_n$ as linear spaces. Therefore, we can consider any linear code $\mathcal{C}$ of block length n as a linear subspace of $\mathcal{F}_n$.

Proposition. The code $\mathcal{C} \subset \mathbb{F}^n$ is a cyclic linear code if and only if $\mathcal{C}$ is an ideal in $\mathcal{F}_n$.

Corollary. If $\mathcal{C}$ is a cyclic linear code of length n, then there exists an unique monic polynomial $g(x) \in \mathcal{F}_n$ such that $\mathcal{C} = (g(x))$ and its degree is less or equal to the degree of any other polynomial of $\mathcal{C}$.
$g(x)$ is called the *generator polynomial* of $\mathcal{C}$.

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

Theorem. Let $\mathcal{C} = (g(x))$ be a cyclic code of length n with generator polynomial $g(x) = g_0 + g_1 x + \cdots + g_r x^r$. Then

1. Every codeword $c(x) \in \mathcal{C}$ can be express uniquely as $c(x) = i(x).g(x)$, where $\deg i(x) < k = n - r$;
2. $\dim \mathcal{C} = n - \deg g(x) = n - r = k$;
3. The following $(n - r) \times n$ matrix is a generator matrix of $\mathcal{C}$:

$$\mathbf{G} = \begin{pmatrix} g_0 & g_1 & \dots & g_r & 0 & 0 & \dots & 0 \\ 0 & g_0 & \dots & g_{r-1} & g_r & 0 & \dots & 0 \\ \dots & & & & & & & \\ 0 & 0 & \dots & & & g_0 & \dots & g_r \end{pmatrix};$$

4. If $h(x) = h_0 + h_1 x + \cdots + h_k x^k, \quad k = n - r$, then the following $r \times n$ matrix is a parity-check matrix of $\mathcal{C}$:

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & \dots & 0 & h_k & \dots & h_1 & h_0 \\ 0 & 0 & \dots & h_k & h_{k-1} & \dots & h_0 & 0 \\ & & & & & & & \\ h_k & h_{k-1} & \dots & \dots & h_0 & \dots & 0 & 0 \end{pmatrix}.$$

Note that every polynomial $s(x) = g(x)u(x) \in \mathcal{C}$ with $(u(x), h(x)) = 1$ generates $\mathcal{C} = (g(x))$ in $\mathcal{F}_n$, but only $g(x)$ satisfies the point 1. of the theorem.

Theorem. Let $\mathcal{C} = (g(x))$ be a cyclic $[n, n-r]$ code with parity-check polynomial $h(x) = (x^n - 1)/g(x)$. Then $\mathcal{C}^\perp$ is generated by $h^*(x)$, the reciprocal polynomial of $h(x)$.

Examples

Let $\mathbb{F} = GF(2)$ and $n = 7$. Then

$$x^7 - 1 = (x+1)(x^3 + x + 1)(x^3 + x^2 + 1) = (x+1)p(x)p^*(x).$$

Consider $C = (p(x))$. Its parity-check polynomial is

$$h(x) = (x+1)p^*(x) = x^4 + x^2 + x + 1.$$

Hence

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, \qquad \mathbf{H} = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Therefor, $\mathcal{C}$ is equivalent to the $[7, 4, 3]_2$ Hamming code.

Encoding cyclic codes

Let $\mathcal{C} = (g(x))$ be an $[n, k]$ cyclic code with $\deg g(x) = n - k$. Let $i(x) = i_0 + i_1 x + \cdots + i_{k-1} x^{k-1}$ be the polynomial corresponding to the message $\mathbf{i} = (i_0, i_1, \ldots, i_{k-1}) \in \mathbb{F}^k$.

- *non-systematic encoding*: $\mathbf{i} \to c(x) = i(x)g(x)$. It corresponds to $\mathbf{c} = \mathbf{i}\mathbf{G}$. Simple, but it is not systematic.

Encoding cyclic codes

Let $\mathcal{C} = (g(x))$ be an $[n, k]$ cyclic code with $\deg g(x) = n - k$. Let $i(x) = i_0 + i_1 x + \cdots + i_{k-1} x^{k-1}$ be the polynomial corresponding to the message $\mathbf{i} = (i_0, i_1, \ldots, i_{k-1}) \in \mathbb{F}^k$.

- *non-systematic encoding*: $\mathbf{i} \to c(x) = i(x)g(x)$. It corresponds to $\mathbf{c} = \mathbf{i}\mathbf{G}$. Simple, but it is not systematic.
- *systematic encoding*: $c(x) = -r(x) + x^{n-k} i(x)$, where $x^{n-k} i(x) = q(x).g(x) + r(x)$, with $\deg r(x) < n - k$. The first $n - k$ positions of $\mathbf{c}$ are the parity-check symbols, and the last k ones are the information symbols. The corresponding generator matrix is $\mathbf{G} = (\mathbf{A} | \mathbf{I}_k)$, where the rows of $\mathbf{A}$ are the reminders of the division of $x^{n-k}, x^{n-k+1}, \ldots, x^{n-1}$ by $g(x)$. Respectively, the parity-check matrix is $\mathbf{H} = (\mathbf{I}_r | -\mathbf{A}^\tau)$.

Decoding cyclic codes

Let $[f(x)]_{g(x)}$ denote the reminder of $f(x)$ modulo $g(x)$.

Definition. Let $\mathcal{C} = (g(x))$ and $v(x) \in \mathcal{F}_n$ be the polynomial corresponding to $\mathbf{v} \in \mathbb{F}^n$. The *syndrome polynomial,* $s(x)$, of $v(x)$ (with respect to $\mathcal{C}$) is the reminder of $f(x)$ modulo $g(x)$, i.e., $s(x) = [v(x)]_{g(x)}$. In the case of systematic encoding $s(x) \leftrightarrow \mathbf{s} = \mathbf{v}\mathbf{H}^{\tau}$.

The vector $v \in \mathbb{F}^n$ is a codeword if and only if $s(x) = 0$.

Decoding cyclic codes

Let $[f(x)]_{g(x)}$ denote the reminder of $f(x)$ modulo $g(x)$.

Definition. Let $\mathcal{C} = (g(x))$ and $v(x) \in \mathcal{F}_n$ be the polynomial corresponding to $\mathbf{v} \in \mathbb{F}^n$. The *syndrome polynomial,* $s(x)$, of $v(x)$ (with respect to $\mathcal{C}$) is the reminder of $f(x)$ modulo $g(x)$, i.e., $s(x) = [v(x)]_{g(x)}$. In the case of systematic encoding $s(x) \leftrightarrow \mathbf{s} = \mathbf{v}\mathbf{H}^\tau$.

The vector $v \in \mathbb{F}^n$ is a codeword if and only if $s(x) = 0$.

Let a codeword $c(x)$ be sent across the channel and the vector $v(x) = c(x) + e(x)$ be received. Then the syndrome is $s(x) = [v(x)]_{g(x)} = [e(x)]_{g(x)}$. A decoder based on the nearest neighbour decoding rule looks for the polynomial $e(x)$ of the smallest weight with $s(x) = [e(x)]_{g(x)}$. Hence, a decoder for the general class of cyclic codes realizes syndrome table decoding.

Decoding cyclic codes 2

The properties of cyclic codes allows the size of the syndrome table to be cut. Nevertheless, the complexity of decoders for general cyclic codes increase exponentially with code length and with the number of errors to be corrected per codeword. For completeness we describe such one decoder.

Decoding cyclic codes 2

The properties of cyclic codes allows the size of the syndrome table to be cut. Nevertheless, the complexity of decoders for general cyclic codes increase exponentially with code length and with the number of errors to be corrected per codeword. For completeness we describe such one decoder.

Meggitt's decoder: Let $\mathcal{C} = (g(x))$ be an $[n, k]$ cyclic t-error correcting code. Instead of storing all error pattern $e(x)$ of weight $\leq t$, we store only one representative $e(x)$ for all its cyclic shifts, for example, $e(x)$ with $e_{n-1} \neq 0$. Also, we use a modified syndrome polynomial: $s(x) = [x^{n-k}v(x)]_{g(x)}$.

The decoder is based on the two simple facts:

1. The syndrome of the cyclic shift of $v(x)$ is $[xs(x)]_{g(x)}$
2. The syndrome of $v(x) - e_{n-1}x^{n-1}$ is $s(x) - e_{n-1}x^{n-k-1}$.

Decoding cyclic codes 3

Algorithm.

1. At receiving $\mathbf{v}$ compute its syndrome $s(x) = [x^{n-k}v(x)]_{g(x)}$. If $s(x) = 0$ go to step 6.
2. Look for the syndrome in the syndrome table. If $s(x)$ is in the table go to step 4. Otherwise, go to step 3.
3. Shift to right $v(x)$ ($v(x) := [xv(x)]_{x^n-1}$) and calculate $s(x) := [xs(x)]_{g(x)}$. Go to step 2.
4. Calculate $v(x) := v(x) - e_{n-1}x^{n-1}$ and $s(x) := s(x) - e_{n-1}x^{n-k-1}$. If $s(x) \neq 0$ go to step 2.
5. After n shifts stop. If $s(x)$ is not found in the syndrome table give "more than t errors".
6. $c(x) := v(x)$ and stop.

Burst error detection and correction

In many real life channels errors occur in burst. For example, they can be caused by transient or intermittent hardware faults or by surface contamination of the storage media. Burst errors are prevalent in mobile communication channels that suffer from multipath fading.

Definition. A *burst of length* b is a vector whose nonzero components are among b successive positions, the first and the last of which are nonzero.

The polynomial form of a burst-error pattern of length b is $e(x) = x^l b(x)$, where $\deg b(x) = b - 1$.

A code designed to correct bursts of length $\leq b$ must have unique syndromes for every such an error pattern. Similarly, for detecting such errors no burst of length $\leq b$ must be a codeword.

Fire codes

A classical example of burst-error correcting codes is the class of Fire codes.

Definition. Let $m \geq b$ be integers and $\mathbb{F} = GF(q)$. A *Fire code* is a cyclic code $\mathcal{C} = (g(x))$ with

$$g(x) = (x^{2b-1} - 1)p(x),$$

where $p(x)$ is an irreducible polynomial over $\mathbb{F}$ of degree m and $p(x)$ does not divide $x^{2b-1} - 1$. The block length of $\mathcal{C}$ is the smallest integer n such that $g(x)$ divides $x^n - 1$.

Example. Consider the binary Fire code with $m = b = 3$ and $p(x) = x^3 + x + 1$. $p(x)$ does not divide $x^5 - 1$. Thus,

$$g(x) = (x^5 + 1)p(x) = x^8 + x^6 + x^5 + x^3 + x + 1,$$

and $\mathcal{C}$ is $[35, 27]$ code correcting all bursts of length up to 3. (35 is the l.c.m. of 5 and 7, the order of $p(x)$.)

Cyclic redundancy check (CRC) codes 1

The CRC codes are the most frequently used error detecting codes owing to their very good error detection performance and extremely simple encoder and decoder implementations. The most of ARQ (automatic-repeat request) protocols are based on such codes. CRC codes are mainly constructed by shortening cyclic codes, and herein, we refer to the shorten cyclic codes as CRC codes.

Definition. Let $\mathcal{C} = (g(x))$ be an $[n, k]$ systematic cyclic code. A shorten $[n-l, k-l]$ code $\check{\mathcal{C}}$ obtained by deleting the $l < k$ rightmost coordinates (they are information positions) is called *shorten cyclic code, or polynomial code*. The code $\check{\mathcal{C}}$ consists of all polynomials of $\mathcal{C}$ with degree $\leq n-1-l$. The encoding procedure is the same as one of the code $\mathcal{C}$, but it is almost always noncyclic.

Cyclic redundancy check (CRC) codes 2

It is common practice to select CRC generator polynomial of the form $g(x) = (x+1)f(x)$, where very often $f(x)$ is a primitive polynomial. The $(x+1)$ factor ensures that all odd-weight error patterns are detectable (the code has only even-weight codewords).

CRC code	Generator polynomial	d
CRC-7	$x^7+x^4+1=(x+1)(x^2+x+1)(x^4+x^3+1)$	3
CRC-12	$x^{12}+x^{11}+x^3+x^2+x+1=(x+1)(x^{11}+x^2+1)$	4
CCITT-16	$x^{16}+x^{12}+x^5+1=$ $(x+1)(x^{15}+x^{14}+x^{13}+x^{12}+x^4+x^3+x^2+x+1)$	4
ANSI-16	$x^{16}+x^{15}+x^2+1=(x+1)(x^{15}+x+1)$	
SDLC	$x^{16}+x^{15}+x^{13}+x^7+x^4+x^2+x+1=$ $(x+1)^2(x^{14}+x^{13}+x^{12}+x^{10}+\cdots+x+1)$	

Cyclic redundancy check (CRC) codes 3

The *coverage ratio,* λ of a q-ary $[n, k]$ CRC code $\mathcal{C} = (g(x))$ is defined to be the ratio of the number of non-codewords to the number of all n-tuples:

$$\lambda = \frac{q^n - q^k}{q^n} = 1 - \frac{1}{q^{n-k}}.$$

CRC codes are good at performing burst-error detection. Since no codeword is a polynomial of degree $< n - k$, the code $\mathcal{C}$ detects all burst-error of length $\leq n - k$.

Proposition. The code $\mathcal{C}$ can detect the fraction

$$\gamma_{n-k+1} = 1 - \frac{1}{q^{n-k-1}(q-1)}$$

of all burst-error patterns of length $n - k + 1$, and

$$\gamma_b = 1 - \frac{1}{q^{n-k}} \quad \text{for length } b > n - k + 1$$

The end of the part

Thank You for Attention!