
Review/Outline

Review:

Check matrix criterion for min dist
Vandermonde matrices
Reed-Solomon codes
Models, computations in finite fields

Today:

Hamming codes
Bose-Chaudhuri-Hocquengham (BCH) codes

Hamming codes

Binary Hamming codes correct single errors, with information rates approaching 1. One proof of the minimum distance properties is via simple variant check matrices, a preview of BCH codes. So forget about Vandermonde determinants for the moment.

Fix block length $n = 2^k - 1$. Let g be a *primitive* polynomial of degree k . Let C be the cyclic code generated by $g(x)$. From our discussion of cyclic codes, this is an $[n, n - k]$ -code. Specifically, it is a $[2^k - 1, 2^k - 1 - k]$ -code.

Take a simple variant check matrix

$$H = (1 \quad \beta \quad \beta^2 \quad \dots \quad \beta^{n-1})$$

where β is a root of $g(x) = 0$ in \mathbf{F}_{2^k} . That is, β is a primitive element in \mathbf{F}_{2^k} .

Any two columns of this check matrix are linearly independent, by not being multiples of each other, so the minimum distance of the code arising from this is 3, and it can correct any single error.

Proof: We prove directly that the minimum distance is at least 3. Since the code is linear, this is equivalent to the minimum *weight* being at least 3. If there were v in the code with weight 2, with non-zero entries only at i^{th} and j^{th} places, then

$$0 = v \cdot H^t = \beta^i + \beta^j$$

Without loss of generality, $i < j$. Then

$$\beta^{j-i} + 1 = 0$$

That is, β would satisfy an equation of degree $j - i$ with coefficients in \mathbf{F}_2 . But by hypothesis $g(x) = 0$ is the lowest-degree equation satisfied by β , and g has degree $n > j \geq j - i$, so this is impossible. Thus, the minimum distance is at least 3. ///

Remark: No determinants here, but β is not in \mathbf{F}_2 .

Example: For example, with $k = 3$, $n = 2^3 - 1 = 7$, use *primitive* cubic polynomial $g(x) = x^3 + x + 1$. The cyclic code generated by this gives a binary $[7, 4]$ -code correcting any single error. A generating matrix is made from the coefficients 1101 (in ascending order) of $g(x)$:

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

Let $h(x)$ be the coefficient-reversed form of

$$\frac{x^7 - 1}{g(x)} = \frac{x^7 - 1}{x^3 + x + 1} = x^4 + x^2 + x + 1$$

so $h(x) = 1 + x^2 + x^3 + x^4 = x^4 + x^3 + x^2 + 1$ and a check matrix is

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

From the variant check matrix that the code has minimum distance at least 3. The **rate** of this code is dimension/length = $4/7$.

Example: With $k = 4$, $n = 2^4 - 1 = 15$ use primitive quartic $g(x) = x^4 + x + 1$ as generating polynomial to make a binary cyclic $[15, 11]$ -code, since $15 = 2^4 - 1$ and $11 = 2^4 - 1 - 4$. A generating matrix is

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

The info rate is $11/15$. (As the size of Hamming codes grows the rate approaches 1.) But the **relative error correction**

$$\frac{\text{number of errors correctible}}{\text{block size}}$$

goes to 0 as the block size grows. This is bad. Thus, the useful codes among the Hamming codes are the relatively small ones.

BCH codes

BCH codes use variant check matrices to extend ideas from Hamming and Reed-Solomon codes. BCH codes can keep a fixed alphabet, and correct many errors.

Unfortunately, BCH codes suffer from the same shortcoming as Hamming and RS codes: as block size increases, they become worse and worse in the sense that the relative error correction rate goes to 0 (and/or information rate goes to 0).

Start with a finite field \mathbf{F}_q with q elements, often $q = 2$. Choose block size n , for simplicity relatively prime to q . For $q = 2$, look at *odd* block sizes.

For example, to correct two errors have any bunch of 4 columns be linear independent, via variant) check matrix like

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & (\alpha^3)^2 & \dots & (\alpha^{n-1})^2 \\ 1 & \alpha^3 & (\alpha^2)^2 & (\alpha^3)^3 & \dots & (\alpha^{n-1})^3 \\ 1 & \alpha^4 & (\alpha^2)^4 & (\alpha^3)^4 & \dots & (\alpha^{n-1})^4 \end{pmatrix}$$

with α possibly lying in some larger field \mathbf{F}_{q^m} . with $1, \alpha, \alpha^2, \alpha^3, \dots, \alpha^{n-1}$ are distinct and $n \geq 5$. The determinant of any 4-by-4 matrix made from 4 different columns of this matrix is non-zero Vandermonde, so the code will correct 2 errors.

Similarly we can make check matrices with any 6 columns linear independent (so any 3 errors correctible), any 8 columns linearly independent (so any 4 errors correctible), etc. This much was already done by Reed-Solomon codes using larger and larger alphabets.

As for Hamming codes, we allow the element α to be in a field \mathbf{F}_{q^m} larger than the field \mathbf{F}_q used as the alphabet for the code.

This would be in contrast to the RS codes where we never went outside the finite field \mathbf{F}_q used as the code alphabet. Staying inside \mathbf{F}_q was what required that Reed-Solomon codes use larger and larger \mathbf{F}_q as the block size goes up.

Instead, if we can make check matrices over *larger* fields but keep the code alphabet itself *fixed*, we can make multiple-error-correcting codes using small alphabets.

This trick is an example of taking a **subfield subcode**.

Let α be a primitive element in \mathbf{F}_{q^m} . For simplicity, take

$$\text{block length} = n = q^m - 1$$

For an integer t with $t < n = q^m - 1$ the matrix $H =$

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \dots & (\alpha^{n-1})^2 \\ 1 & \alpha^3 & (\alpha^2)^3 & \dots & (\alpha^{n-1})^3 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \alpha^{t-2} & (\alpha^2)^{t-2} & \dots & (\alpha^{n-1})^{t-2} \\ 1 & \alpha^{t-1} & (\alpha^2)^{t-1} & \dots & (\alpha^{n-1})^{t-1} \end{pmatrix}$$

has the property that the $(t - 1)$ -by- $(t - 1)$ matrix formed from any $t - 1$ columns has non-zero (Vandermonde) determinant.

- In this notation, the quantity t is the **designed distance**.

We must connect such a check matrix with generating polynomials for a cyclic code. For $1 \leq i \leq t-1$, let f_i be the irreducible polynomial with coefficients in \mathbf{F}_q such that

$$f_i(\alpha^i) = 0$$

Since α lies in \mathbf{F}_{q^m} (and is primitive) each irreducible $f_i(x)$ is a factor of $x^{q^m-1} - 1$. Let

$$g(x) = \text{least common multiple of } (f_1, \dots, f_{t-1})$$

Since $n = q^m - 1$ and q are relatively prime, $x^{q^m-1} - 1$ has no repeated factors, so unless two f_i are *the same* their lcm is their *product*.

So the generating polynomial $g(x)$ for the code C with the check matrix H above is the product of the different irreducible polynomials f_i which have roots α^i ($1 \leq i < t$), **not** repeating a given polynomial f_i if two different α^i and α^j are roots of the same f_i . Then the question is: given q , m , block length $n = q^m - 1$, primitive element α , and designed distance t ,

- How can we nicely determine $g(x)$ as a function of t ?

Definition: For a in \mathbf{F}_{q^m} , the **Frobenius map** $\mathbf{F}_{q^m} \rightarrow \mathbf{F}_{q^m}$ is

$$a \rightarrow a^q$$

Theorem: For a, b in the finite field \mathbf{F}_{q^m} , the **Frobenius map** $a \rightarrow a^q$ has the properties

- $(xy)^q = x^q y^q$
- $(x + y)^q = x^q + y^q$
- For a polynomial f with coefficients in \mathbf{F}_q , suppose the equation $f(x) = 0$ has root $a \in F_{q^m}$. Then also $f(a^q) = 0$.

So not only is $\alpha = \alpha^1$ a root of f_1 , but also α^q, α^{q^2} , etc. are all roots of f_1 . This is not an infinite list, because $\alpha^{q^m} = \alpha$ so the list repeats. And $\alpha^2, (\alpha^2)^q, (\alpha^2)^{q^2}$, etc. are roots of f_2 . Further, $\alpha^3, (\alpha^3)^q, (\alpha^3)^{q^2}$, etc. are roots of f_3 . Etc.

A key question: among $\alpha, \alpha^2, \dots, \alpha^{t-1}$, how many *different* polynomials f_i do we need? The more we need, the *larger* the degree of g , and thus the *smaller* the quantity

$$n - \deg g$$

appearing in the *rate* of the code

$$\text{rate} = 1 - \frac{\deg g}{n}$$

In the worst case, even with designed distance $t \ll n$ it can happen that

$$g(x) = \frac{x^n - 1}{x - 1} = \frac{x^{q^m - 1} - 1}{x - 1}$$

which would give us a bad code with rate $\frac{1}{n}$.

Also

- The degree of each irreducible factor f_i of $x^n - 1 = x^{q^m - 1} - 1$ is $\leq m$.

Example: We want a binary code (alphabet \mathbf{F}_2) with block size $n = 2^3 - 1 = 7$. (So $m = 3$ in notation above). What t will give something worthwhile?

We must have $t < n = 7$. To make a binary code that corrects any 2 errors, we have to take $t = 5$. To specify a primitive element α in \mathbf{F}_{2^3} we describe \mathbf{F}_{2^3} as

$$\mathbf{F}_{2^3} = \mathbf{F}_2[x]/P$$

for some primitive cubic polynomial

$$P(x) = x^3 + x + 1$$

and let

$$\alpha = x\text{-mod-}P(x)$$

The check matrix is

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^6 \\ 1 & \alpha^2 & (\alpha^2)^2 & \dots & (\alpha^2)^6 \\ 1 & \alpha^3 & (\alpha^3)^2 & \dots & (\alpha^3)^6 \\ 1 & \alpha^4 & (\alpha^4)^2 & \dots & (\alpha^4)^6 \end{pmatrix}$$

The polynomial $x^n - 1 = x^7 - 1$ factors as

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

(by trial-and-error). By testing, all three irreducible factors are primitive. (We used one of them to define \mathbf{F}_{2^3} .) The α, α^2 , and $\alpha^4 = (\alpha^2)^2$ are obtained by applying the Frobenius automorphism to α , so these are all the roots of $x^3 + x + 1 = 0$. The α^3 cannot be obtained in this manner, yet is not simply 1 (α is primitive), so must be a zero of the other factor $x^3 + x^2 + 1$ of $x^7 - 1$. Thus, the generating polynomial for this code is the product

$$\begin{aligned} g(x) &= (x^3 + x + 1)(x^3 + x^2 + 1) \\ &= x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \end{aligned}$$

A generating matrix is

$$G = (1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1)$$

Too bad: a majority-logic repetition code: send each bit 7 times.

Try again. Enlarge block size to $n = 2^4 - 1 = 15$. Try (again) to make a *binary* code that corrects any 2 errors, so take designed distance $t = 5$. To specify a primitive element α in \mathbf{F}_{2^4} we describe \mathbf{F}_{2^4} as

$$\mathbf{F}_{2^4} = \mathbf{F}_2[x]/P$$

with primitive quartic

$$P(x) = x^4 + x + 1$$

and put

$$\alpha = x\text{-mod-}P(x)$$

The check matrix is $H =$

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{14} \\ 1 & \alpha^2 & (\alpha^2)^2 & \dots & (\alpha^2)^{14} \\ 1 & \alpha^3 & (\alpha^3)^2 & \dots & (\alpha^3)^{14} \\ 1 & \alpha^4 & (\alpha^4)^2 & \dots & (\alpha^4)^{14} \end{pmatrix}$$

The polynomial $x^n - 1 = x^{15} - 1$ factors as

$$\begin{aligned} x^{15} - 1 &= (x - 1)(x^2 + x + 1) \times \\ &\times (x^4 + x^3 + x^2 + x + 1)(x^4 + x + 1)(x^4 + x^3 + 1) \end{aligned}$$

(by trial and error!). By testing, only the last two polynomials are primitive.

So we need to identify which among the factors

$$(x - 1), (x^2 + x + 1), (x^4 + x^3 + x^2 + x + 1),$$

$$(x^4 + x + 1), (x^4 + x^3 + 1)$$

get used to make equations with $\alpha, \alpha^2, \alpha^3, \alpha^4$ as roots. Since $q = 2$, the image α^2 of α under the Frobenius map is the next power of α in the sequence, and $\alpha^4 = (\alpha^2)^2$. To have α^3 be a root we need another polynomial.

Try to be lucky (rather than systematic?).

Since $\alpha^{15} = 1$

$$(\alpha^3)^5 = 1$$

The polynomial $x^5 - 1$ factors

$$x^5 - 1 = (x^4 + x^3 + x^2 + x + 1)(x + 1)$$

Since $\alpha^3 \neq 1$ (since α is primitive)

$$\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1 = 0$$

Thus, the generating polynomial is

$$\begin{aligned} g(x) &= (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) \\ &= x^8 + x^4 + x^2 + x + 1 \end{aligned}$$

Giving generating matrix

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

This is a *binary* $[15, 7]$ -code and has minimum distance at least 5, by construction, so can correct any 2 errors.

Remark: It was not easy to predict that the code would be $[15, 7]$. As the check matrix had only 4 rows and had block size 15, we might have thought the code would have dimension $15 - 4 = 11$. The difference is accounted for (indirectly) by the fact that we restrict to a *binary* code, not with alphabet \mathbf{F}_{16} .

An algorithm to determine the dimension and estimate the minimum distance of BCH codes.

Apply the Frobenius automorphism $x \rightarrow x^p$ repeatedly to the rows of the initial check matrix H for the BCH code, making a larger check matrix, the *Frobenius-stable* check matrix (after all the different possibilities are included, but without repetition). The row rank of this Frobenius-stable check matrix is the ‘true’ row rank:

$$\begin{aligned} & \text{dimension of BCH code} \\ &= \text{length} - \text{row rk Frob-stable chk mx} \end{aligned}$$

And a better estimate of minimum distance can be obtained from the Frobenius stable check matrix: *let t' be largest such that adjacent exponents $1, 2, 3, \dots, t' - 2, t' - 1$ appear as exponents (of the primitive root) in the second column of the Frobenius-stable check matrix.*

Then the minimum distance is *at least t' .*

More efficiently, given the set of exponents (of the primitive root) in the second column of the usual check matrix for a length n BCH code using $GF(p^m)$ over $GF(p)$, to determine the set of such exponents in the Frobenius-stable version repeatedly multiply these exponents by p (reducing modulo $p^m - 1$).

Let r be the number of exponents in the Frobenius-stabilized set: then the actual dimension k of the code is $k = n - r$.

Remark: There is no need to write the whole rows of any check matrix, but only the exponents occurring in the second column.

Example: To determine dimension and estimate minimum distance of the BCH code of length 26 constructed with designed distance 9 using the field extension $GF(3^3)$ of the finite field $GF(3)$, proceed as follows.

The initial set of exponents in the second column of the check matrix is $1, 2, 3, \dots, 8$ (going up to $t - 1$). Repeatedly multiplying by 3 (applying the Frobenius) gives Frobenius-stable set

$1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 18, 19, 20, 21, 24$

which has 18 elements. Thus, the row rank of the Frobenius-stable check matrix is 18, and the dimension of the BCH code is (with length 26)

$$26 - 18 = 8$$

The largest t' so that adjacent exponents $1, 2, \dots, t' - 1$ appear is 13. Thus, the actual minimum distance is ≥ 13 , not just 9.