EXAMPLES: DECODING ALGORITHM

Contents

This note contains the detailed computations of the examples of decoding $\mathrm{BCH}(4,3)$ and $\mathrm{RS}(4,3)$ that were presented in the last lecture, and some tips for creating further examples to practise the decoding algorithm. I recommend you to *work through the examples by yourself before you read the full computation from this note*; this note will be less helpful, otherwise.

As usual, consider the field $F = \mathbb{B}[\alpha]/\alpha^4 + \alpha + 1$ with 16 elements with α our preferred primitive element.

1. Example: Decoding $\mathrm{BCH}(4,3)$

Suppose that a message is transmitted via $\mathrm{BCH}(4,3)$. As usual, we identify $(v_{14}, \cdots, v_1, v_0) \in \mathbb{B}^{15}$ with a polynomial $v_{14}X^{14} + \cdots + v_1 X + v_0 \in \mathbb{B}[X]$.

$$\begin{aligned}
\text{transmitted word:}\quad & c(X) = X^{14} + X^9 + X^7 + X^4 + X^3 + X + 1\\
\text{received word:}\quad & d(X) = X^{14} + X^9 + X^7 \phantom{{}+ X^4 + X^3} + X + 1\\
\text{error polynomial:}\quad & e(X) = \phantom{X^{14} + X^9 + X^7 +{}} X^4 + X^3\\
\text{error positions:}\quad & 3, 4\\
\text{error locator:}\quad & l(z) = (1-\alpha^3 z)(1-\alpha^4 z) = 1 + \alpha^7 z + \alpha^7 z^2\\
\text{error evaluator:}\quad & w(z) = \alpha^3(1-\alpha^4 z) + \alpha^4(1-\alpha^3 z) = \alpha^7
\end{aligned}$$

Exercise 1.0.1. Check that the transmitted word *is* a codeword of $\mathrm{BCH}(4,3)$ using each of the following strategies:

(1) Find the generator polynomial $g_{4,3}(X) \in \mathbb{B}[X]$ and show that the transmitted word $c(X)$ is divisible by $g_{4,3}(X)$.
(2) Find the check polynomial $h_{4,3}(X)$ and show that $c(X)h_{4,3}(X)$ is divisible by $X^{15} - 1$; or equivalently, show that the ith coefficient and $(i+15)$th coefficients of $c(X)h_{4,3}(X)$ are same for all $i = 0, \cdots, 14$.
(3) Show that the syndrome polynomial is zero; i.e., show that $c(\alpha) = c(\alpha^3) = c(\alpha^5) = 0$.

Now, assume that we only know the received word $d(X)$, and try to recover $c(X)$ via the decoding algorithm. Try to *work out the details by yourself*, if you have not done so, before you read the section below.

1.1. Find the syndrome polynomial.

First compute the odd syndromes:

$$d(\alpha) = \alpha^7, \quad d(\alpha^3) = \alpha^8, \quad d(\alpha^5) = \alpha^{10} \tag{1.1.1}$$

The even syndromes can easily computed from the odd syndromes, as follows:

$$d(\alpha^2) = (d(\alpha))^2 = \alpha^{14}, \quad d(\alpha^4) = (d(\alpha^2))^2 = \alpha^{13}, \quad d(\alpha^6) = (d(\alpha^3)^2 = \alpha \tag{1.1.2}$$

(Of course, you can directly evaluate $d(\alpha^{2i})$ and obtain the same answer, but this involves more calculation.)

From (1.1.1) and (1.1.2) we obtain the syndrome polynomial.

$$s(z) := \alpha z^5 + \alpha^{10}z^4 + \alpha^{13}z^3 + \alpha^8 z^3 + \alpha^{14}z + \alpha^7$$

Exercise 1.1.3. Compute (1.1.1) and justify (1.1.2).

1.2. **Euclid's algorithm.** We run Euclid's algorithm for $s(z)$ and z^6:

Step 1: $z^6 = (\alpha^{14}z + \alpha^8)s(z) + r_1(z)$ where

$$r_1(z) := \alpha^{10}z^4 + \alpha^{10}z^3 + \alpha^{12}z^2 + \alpha^{10}z + 1.$$

Step 2: $s(z) = (\alpha^6 z + \alpha^{13})r_1(z) + r_2(z)$ where $r_2(z) = \alpha^5$.

We terminate the process since $deg(r_2(z)) = 0 < 3$.

Remark 1.2.1. If the algorithm terminated in s steps and produces $r_s(z) = 0$, then it indicates that there are more than 3 error bits in the message block. (Mode A)

We put all these together.

$$\begin{aligned} \alpha^5 &= s(z) + (\alpha^6 z + \alpha^{13})r_1(z) \qquad \ldots\textbf{Step 2} \\ &= s(z) + (\alpha^6 z + \alpha^{13})\Big((\alpha^{14} + \alpha^8)s(z) + z^6\Big) \qquad \ldots\textbf{Step 1} \\ &\equiv (\alpha^5 z^2 + \alpha^5 z + \alpha^{13})s(z) \bmod z^6 \end{aligned}$$

Note that the constant term of $u_2(z) := \alpha^5 z^2 + \alpha^5 z + \alpha^{13}$ is α^{13}, and its inverse is α^2. Therefore

$$\begin{aligned} l(z) &= \alpha^2 \cdot u_2(z) = \alpha^7 z^2 + \alpha^7 z + 1 \\ w(z) &= \alpha^2 \cdot r_2(z) = \alpha^7 \end{aligned}$$

Remark 1.2.2. If the algorithm terminated in s and the constant term of $u_s(z)$ is zero, then it indicates that there are more than 3 error bits in the message block. (Mode B1)

1.3. **Finding the roots of $l(z)$.** The only general method of finding all the roots of $l(z)$ is "exhaustive search"; i.e. compute $l(1), l(\alpha), \cdots$ until you find all the roots. (Note that the constant term of $l(z)$ is 1 so $z = 0$ cannot be a root.) We obtain that α^{-3} and α^{-4} are the roots of $l(z)$.

Remark 1.3.1. Here is a useful tip for finding the roots of $l(z)$. Since $l(z)$ is of degree 2, we know that it "should" factor as $l(z) = (1 - \alpha^r z)(1 - \alpha^s z) = 1 + (\alpha^r + \alpha^s)z + \alpha^r\alpha^s z^2$. If you have found one root, say α^{-3} for example, then we may take $r = 3$ and we also have $\alpha^3\alpha^s = \alpha^7$ by comparing the coefficients of z^2. Therefore $\alpha^s = (\alpha^3)^{-1}\alpha^7 = \alpha^4$, so α^{-4} is the other root.

Remark 1.3.2. If $l(z)$ has degree > 2, then the following trick can be useful. Let $l(z) = \alpha^9 z^3 + z^2 + \alpha^{12}z + 1$, for example. Clearly 0 is not a root, so we try plugging in $z = 1, \alpha^{-1}, \alpha^{-2}, \cdots$. We see $l(1) \neq 0$ and $l(\alpha^{-1}) \neq 0$, but we see $l(\alpha^{-2}) = 0$. First, make sure that it is *not* a multiple root by checking $\frac{d}{dz}l(\alpha^{-2}) \neq 0$. Indeed $\frac{d}{dz}l(z) = \alpha^9 z^2 + \alpha^{12}$ does not have α^{-2} as a root. (Note that if $l(z)$ has a multiple root, then it means that the decoding algorithm cannot work and that there are more than 3 error-bits in the message block. See Remark 1.3.3.) Since α^{-2} is a root of $l(z)$, it implies that $1 - \alpha^2 z$ divides $l(z)$ and all the other roots of $l(z)$ are roots of $l(z)/(1 - \alpha^2 z) = \alpha^7 z^2 + \alpha^7 z + 1$. Also note that 1 and α^{-1} are not roots of $l(z)/(1-\alpha^2 z)$ as they are not roots of $l(z)$; and α^{-2} is not a root of $l(z)/(1-\alpha^2 z)$ as it is not a multiple root of $l(z)$. Now, continue searching for roots of $l(z)/(1-\alpha^2 z)$, *starting from* $z = \alpha^{-3}, \alpha^{-4}, \alpha^{-5}, \cdots$.

Remark 1.3.3. If $l(z)$ has a multiple root, then it indicates that there are more than 3 error bits in the message block. (Mode B2) If $l(z)$ does not split into linear factors (i.e., it does not have as many roots as its degree), then it indicates that there are more than 3 error bits in the message block. (Mode B3)

1.4. **"Error evaluation" and finding error polynomial.** Since we know that the error positions are $3, 4$, the only possible error polynomial is $X^4 + X^3$. Therefore, the corrected codeword is

$$d(X) + e(x) = X^{14} + X^9 + X^7 + X^4 + X^3 + X + 1,$$

which coincides with the "transmitted word".

If you want to check if there was any arithmetic error, you can check whether this result is consistent with error evaluator. Using the formula (deduced from the definition of $w(z)$:

$$e_i = w(\alpha^{-i})\alpha^{-i} \prod_{j \in M \setminus \{i\}} (1 - \alpha^{j-i})^{-1}, \tag{1.4.1}$$

we compute $e_3 = w(\alpha^{-3})\alpha^{-3}(1 - \alpha^4 \cdot \alpha^{-3})^{-1} = 1$ and $e_4 = w(\alpha^{-4})\alpha^{-4}(1 - \alpha^3\alpha^{-4})^{-1} = 1$. If either e_4 or e_3 is not 1, it indicates that an arithmetic error was made during the algorithm (Mode C).

Exercise 1.4.2. This exercise is not very important but quite amusing. (It is more or less a "thought exercise".) Fix $F = \mathrm{GF}(2^k)$ and a primitive element $\alpha \in F$. Consider "small enough" t. Show that without any arithmetic error, the "error evaluation step" in the decoding algorithm for $\mathrm{BCH}(k, t)$ should always give 1 whenever there are at most t errors. (*Hint:* By the natural inclusion $\mathbb{B}^{2^k-1} \subset F^{2^k-1}$, a codeword of $\mathrm{BCH}(k, t)$ is necessarily a codeword of $\mathrm{RS}(k, t)$. Also the decoding algorithms for $\mathrm{BCH}(k, t)$ and $\mathrm{RS}(k, t)$ are "compatible" with this inclusion.)

2. Example: Decoding RS(4, 3)

The way it works is more or less the same as the decoding algorithm for $\mathrm{BCH}(4, 3)$, except that the "error evaluation step" cannot be skipped; indeed, non-zero coefficients of error polynomial does not have to be 1.

Suppose that a message is transmitted via $\mathrm{RS}(4, 3)$. As usual, we identify $(v_{14}, \cdots, v_1, v_0) \in F^{15}$ with a polynomial $v_{14}X^{14} + \cdots + v_1X + v_0 \in F[X]$.

Let $c(X)$ and $d(X)$ respectively denote the transmitted and received words.

$$\begin{aligned}
c(X) &:= \alpha^5X^{14} + \alpha^7X^{13} + \alpha^6X^{12} + \alpha^{10}X^{11} + \alpha^{14}X^{10} + \alpha^{14}X^9 + X^8 \\
&\quad +\alpha^6X^7 + \alpha^{10}X^6 + X^5 + \alpha^6X^4 + \alpha^{10}X^3 + X^2 + \alpha^6X + \alpha^{10} \\
d(X) &:= \alpha^5X^{14} + \alpha^7X^{13} + \alpha^6X^{12} + \alpha^{10}X^{11} + \alpha^{14}X^{10} + \alpha^{14}X^9 + \alpha^{10}X^8 \\
&\quad +X^7 + \alpha^{13}X^6 + X^5 + \alpha^6X^4 + \alpha^{10}X^3 + X^2 + \alpha^6X + \alpha^{10}
\end{aligned}$$

The error polynomial is $e(X) = \alpha^5X^8 + \alpha^{13}X^7 + \alpha^9X^6$ and the error positions are $6, 7, 8$.

Let $l(X)$ and $w(X)$ respectively denote the error locator and the error evaluator polynomial.

$$\begin{aligned}
l(X) &:= (1 - \alpha^6z)(1 - \alpha^7z)(1 - \alpha^8z) = 1 + \alpha z + \alpha^8z^2 + \alpha^6z^3 \\
w(X) &:= \alpha^9\alpha^6(1 - \alpha^7z)(1 - \alpha^8z) + \alpha^{13}\alpha^7(1 - \alpha^6z)(1 - \alpha^8z) + \alpha^5\alpha^8(1 - \alpha^6z)(1 - \alpha^7z) \\
&= \alpha^6z^2 + \alpha^3z + \alpha^9
\end{aligned}$$

Exercise 2.0.1. Check that the transmitted word *is* a codeword of $\mathrm{RS}(4, 3)$ using each of the following strategies:

(1) Find the generator polynomial $g^{\mathrm{RS}}_{4,3}(X) \in \mathbb{B}[X]$ and show that the transmitted word $c(X)$ is divisible by $g^{\mathrm{RS}}_{4,3}(X)$.

(2) Show that the syndrome polynomial is zero; i.e., show that $c(\alpha) = c(\alpha^2) = \cdots = c(\alpha^6) = 0$.

(3) (This method seems very complicated, but it works.) Find the check polynomial $h_{4,3}^{\mathrm{RS}}(X)$ and show that $c(X)h_{4,3}^{\mathrm{RS}}(X)$ is divisible by $X^{15} - 1$; or equivalently, show that the ith coefficient and $(i + 15)$th coefficients of $c(X)h_{4,3}(X)$ are same for all $i = 0, \cdots, 14$.

Unlike Exercise 1.0.1, it is *not* enough in (2) to check just for odd powers of α; i.e. $c(\alpha) = c(\alpha^3) = c(\alpha^5) = 0$. Explain why.

Now, assume that we only know the received word $d(X)$, and try to recover $c(X)$ via the decoding algorithm. Try to *work out the details by yourself*, if you have not done so, before you read the section below.

2.1. Find the syndrome polynomial.

$$s(z) = \sum_{i=1}^{6} d(\alpha^i)z^{i-1} = \alpha^4 z^5 + \alpha^4 z^4 + \alpha^{13} z^3 + \alpha^8 z^2 + \alpha^{12} z + \alpha^9 \tag{2.1.1}$$

Exercise 2.1.2. Compute (2.1.1), and explain why $d(\alpha^{2j})$ does not have to be equal to $(d(\alpha^j))^2$.

2.2. Euclid's algorithm. We run Euclid's algorithm for $s(z)$ and z^6:

Step 1: $z^6 = (\alpha^{11}z + \alpha^{11})s(z) + r_1(z)$ where

$$r_1(z) := \alpha^7 z^4 + \alpha^{14} z^3 + \alpha^5 z^2 + \alpha^4 z + \alpha^5$$

Step 2: $s(z) = (\alpha^{12}z+\alpha^6)r_1(z)+r_2(z)$ where $r_2(z) = \alpha^{12}z^3+\alpha^{14}z^2+\alpha^6 z+\alpha^2$.
Step 3: $r_1(z) = (\alpha^{10}z + \alpha^7)r_2(z) + r_3(z)$ where $r_3(z) = \alpha^3 z^2 + z + \alpha^6$.

We terminate the process since $deg(r_3(z)) = 2 < 3$.

Remark 2.2.1. If the algorithm terminated in s steps and produces $r_s(z) = 0$, then it indicates that there are more than 3 error bits in the message block. (Mode A)

We put all these together.

$$\begin{aligned}
r_3(z) &= r_1(z) + (\alpha^{10}z + \alpha^7)r_2(z) && \ldots\textbf{Step 3}\\
&= r_1(z) + (\alpha^{10}z + \alpha^7)\Big(s(z) + (\alpha^{12}z + \alpha^6)r_1(z)\Big) && \ldots\textbf{Step 2}\\
&= (\alpha^{10}z + \alpha^7)s(z) + (\alpha^7 z^2 + z + \alpha^6)r_1(z)\\
&= (\alpha^{10}z + \alpha^7)s(z) + (\alpha^7 z^2 + z + \alpha^6)\Big((\alpha^{11}z + \alpha^{11})s(z) + z^6\Big) && \ldots\textbf{Step 1}\\
&\equiv (\alpha^3 z^3 + \alpha^5 z^2 + \alpha^{13} z + \alpha^{12})s(z) \bmod z^6
\end{aligned}$$

Note that the constant term of $u_3(z) := \alpha^3 z^3 + \alpha^5 z^2 + \alpha^{13} z + \alpha^{12}$ is α^{12}, and its inverse is α^3. Therefore

$$\begin{aligned}
l(z) &= \alpha^3 \cdot u_3(z) = \alpha^6 z^3 + \alpha^8 z^2 + \alpha z + 1\\
w(z) &= \alpha^3 \cdot r_3(z) = \alpha^6 z^2 + \alpha^3 z + \alpha^9
\end{aligned}$$

Remark 2.2.2. If the algorithm terminated in s and the constant term of $u_s(z)$ is zero, then it indicates that there are more than 3 error bits in the message block. (Mode B1)

2.3. **Finding the roots of $l(z)$.** We follow the strategy outlined in Remark 1.3.2. Clearly 0 is not a root of $l(z)$, so we try plugging in $z = 1, \alpha^{-1}, \alpha^{-2}, \cdots$. We see $l(1), \cdots, l(\alpha^{-5}) \neq 0$ and $l(\alpha^{-6}) = 0$. Since $\frac{d}{dz} l(z) = \alpha^3 z^2 + \alpha$ does not have α^{-6} as a root, it is not a multiple root of $l(z)$. (Note that if $l(z)$ has a multiple root, then it means that the decoding algorithm cannot work and that there are more than 3 error-bits in the message block. See Remark 1.3.3.) It follows that $1 - \alpha^6 z$ divides $l(z)$ and all the other roots of $l(z)$ are roots of $l(z)/(1 - \alpha^6 z) = z^2 + \alpha^{11} z + 1$. Also note that $1, \cdots, \alpha^{-5}$ are not roots of $l(z)/(1 - \alpha^6 z)$ as they are not roots of $l(z)$; and α^{-6} is not a root of $l(z)/(1 - \alpha^6 z)$ as it is not a multiple root of $l(z)$. Now, continue searching for roots of $z^2 + \alpha^{11} z + 1$, *starting from* $z = \alpha^{-7}, \alpha^{-8}, \cdots$.

Indeed, α^{-7} is a root of $z^2 + \alpha^{11} z + 1$, so we have $(1 - \alpha^7 z)(1 - \alpha^r z) = 1 + \alpha^{11} z + z^2$ for some r. In particular, we have $\alpha^r = (\alpha^7)^{-1} = \alpha^8$. Therefore, we found all the roots of $l(z)$; namely, $\alpha^{-6}, \alpha^{-7}, \alpha^{-8}$. So the error positions are $6, 7, 8$.

Remark 2.3.1. If $l(z)$ has a multiple root, then it indicates that there are more than 3 error bits in the message block. (Mode B2) If $l(z)$ does not split into linear factors (i.e., it does not have as many roots as its degree), then it indicates that there are more than 3 error bits in the message block. (Mode B3)

2.4. **"Error evaluation" and finding error polynomial.** Contrary to the case of $\mathrm{BCH}(k, t)$, this step cannot be skipped. Since we know that the error positions are $6, 7, 8$, the error polynomial is of the form $e(X) = e_8 X^8 + e_7 X^7 + e_6 X^6$. We find e_i using the formula (1.4.1), and obtain:

$$\begin{aligned} e_8 &= w(\alpha^{-8})\alpha^{-8}(1 - \alpha^6\alpha^{-8})^{-1}(1 - \alpha^7\alpha^{-8})^{-1} = \alpha^5 \\ e_7 &= w(\alpha^{-7})\alpha^{-7}(1 - \alpha^6\alpha^{-7})^{-1}(1 - \alpha^8\alpha^{-7})^{-1} = \alpha^{13} \\ e_6 &= w(\alpha^{-6})\alpha^{-6}(1 - \alpha^7\alpha^{-6})^{-1}(1 - \alpha^8\alpha^{-6})^{-1} = \alpha^9 \end{aligned}$$

Therefore we obtain:

$$\begin{aligned} e(X) &= \alpha^5 X^8 + \alpha^{13} X^7 + \alpha^9 X^6 \\ d(X) + e(x) &= \alpha^5 X^{14} + \alpha^7 X^{13} + \alpha^6 X^{12} + \alpha^{10} X^{11} + \alpha^{14} X^{10} + \alpha^{14} X^9 + X^8 \\ &\quad + \alpha^6 X^7 + \alpha^{10} X^6 + X^5 + \alpha^6 X^4 + \alpha^{10} X^3 + X^2 + \alpha^6 X + \alpha^{10}, \end{aligned}$$

Clearly, the corrected word coincides with the "transmitted word".

3. Suggestion for more practice

Let $F = \mathrm{GF}(2^k)$ and fix a primitive element $\alpha \in F$. (You may take $k = 4$ and the usual α, if you want.) This section will sketch how to create more examples to practise the decoding algorithm for $\mathrm{BCH}(k, t)$ and $\mathrm{RS}(k, t)$. Let me first describe an approach that is quite unlikely to be successful: take a random polynomial $d(X) \in \mathbb{B}[X]$ or $F[X]$ of degree $\leqslant 2^k - 2$ and run the decoding algorithm. In fact, the number of words that can be obtained by modifying at most t binary bits from a codeword of $\mathrm{BCH}(k, t)$ is far smaller than $2^{2^k - 1} = |\mathbb{B}^{2^k - 1}|$ unless $t = 1$ (in which case $\mathrm{BCH}(k, 1) = \mathrm{Ham}(k)$ is 1-perfect), and similarly the number of words that can be obtained by modifying at most t symbols from a codeword of $\mathrm{RS}(k, t)$ is far smaller than $2^{k(2^k - 1)} = |F^{2^k - 1}|$. Therefore, the random polynomial $d(X)$ you pick is quite unlikely to occur as a weight $\leqslant t$ error of a codeword of $\mathrm{BCH}(k, t)$ or $\mathrm{RS}(k, t)$.

Exercise 3.1. Verify the claim above for $\mathrm{BCH}(4, t)$ with $t = 2, 3$ and $\mathrm{RS}(4, t)$ with $t = 2, 3, 4$. (Use either calculator or some crude estimates.)

Another approach is to produce a codeword of $\mathrm{BCH}(k, t)$ or $\mathrm{RS}(k, t)$ and modify it at most t symbols. This could be a reasonable amount of computation for

BCH$(4,t)$, but could be extremely time-consuming for RS$(4,t)$. First of all, it involves the nuisance of "finding a codeword". You would realise this if you try to expand the generator polynomial $g^{\mathrm{RS}}_{4,3}(X) = (X-\alpha)\cdots(X-\alpha^6)$. It is not just that, but you need to multiply it against some other polynomial $u(X)$ to create a codeword – of course, you may take $u(X) = 1$, but then you may be unhappy because the example you create would not look very "realistic". Even after going through such a nuisance, finding the syndrome polynomial could also be a nuisance. (Imagine that you are computing $d(\alpha), \cdots, d(\alpha^{2t})$ for some polynomial $d(X) \in F[X]$ of degree 14!) On the other hand, this could be a reasonable task for RS$(3,2)$ where $F = \mathbb{B}[\alpha]/\alpha^3 + \alpha + 1$ is a field with 8 elements and $\alpha \in F$ is a primitive element. In this case, codewords and received words are polynomials of degree $\leqslant 6$. (Also the generator polynomial $g^{\mathrm{RS}}_{3,2}(X)$ is not too difficult to expand.)

If you still want to stick with RS$(4,t)$, then here is how one could create examples without going through the nuisances I described above. Write down some "error polynomial" $e(X)$ of degree $\leqslant 14$ with no more than t non-zero terms. Note that if $d(X) = c(X) + e(X)$ for any codeword $c(X)$, we have $d(\alpha^i) = e(\alpha^i)$ for $i = 1, \cdots, 2t$. So in particular, if you write down an error polynomial, you can obtain the corresponding syndrome polynomial $s(z)$. Once a syndrome polynomial is given, you can run the decoding algorithm to recover the error polynomial. (Though you cannot recover the transmitted word as the received word is not given, this computation only involves adding two polynomials which is easy.) This way, you do not have to get into the unpleasant computation of evaluating a degree-14 polynomial at $2t$ values. One could also try an error polynomial with *more than t* non-zero terms and run the decoding algorithm. You may see various error modes discussed in the lecture notes or obtain a wrong error polynomial.