Notes on Reed-Solomon codes

arnaucube

January 2023

Abstract

Notes taken while reading about Reed-Solomon codes. Usually while reading papers I take handwritten notes, this document contains some of them re-written to *LaTeX*.

The notes are not complete, don't include all the steps neither all the proofs.

Contents

1 Reed-Solomon Codes overview

In this section we overview the main ideas presented in the Reed-Solomon paper [1] and in Bruce Maggs notes [2].

Reed-Solomon codes appeared in parallel to the BCH codes, and can be described as nonbinary BCH codes. In this section we will focus only in the Reed-Solomon codes, particularly in the encoding and error detection (not correction).

tmp-note: I feel like it is worth to check Galois theory & BCH codes before going to Reed-Solomon codes, but due time constraints and our main goal being FRI, probably we should skip it.

1.1 Idea

Let $p(x) = m_0 + m_1 x + \ldots + m_{k-1} x^{k-1}$, where $m_i \in K$ and $k < 2^n$.

We map k-tuples of K into 2^n-tuples of K, where $K = GF(p^r)$.

$$k \begin{cases} m_0 \\ m_1 \\ \vdots \\ m_{k-1} \end{cases} \in K \quad \longrightarrow 2^n \begin{cases} p(0) \\ p(\alpha) \\ p(\alpha^2) \\ \vdots \\ p(1) \end{cases} \in K$$

The receiver then, can decode the messages by solving simultaneously any k of the 2^n equations

$$\begin{aligned} p(0) &= m_0 \\ p(\alpha) &= m_0 + m_1\alpha + m_2\alpha^2 + \ldots + m_{k-1}\alpha^{k-1} \\ p(\alpha^2) &= m_0 + m_1\alpha^2 + m_2\alpha^4 + \ldots + m_{k-1}\alpha^{2k-2} \\ &\vdots \\ p(1) &= m_0 + m_1 + m_2 + \ldots + m_{k-1} \end{aligned}$$

This system of equations can be solved, as we can see that any k of the $p(\alpha^j)$ equations are linearly independent since they form a Vandermonde matrix

$$V = \begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{k-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{k-1} \\ 1 & x_3 & x_3^2 & \ldots & x_3^{k-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^{k-1} \end{pmatrix}$$

and $\det V$ is a Vandermonde determinant and $\det V \neq 0$ as

$$\det\ V = \prod_{j<i}(x_i - x_j)$$

See Annex A for a proof of the Vandermonde determinant.

1.2 Real world approach

In the practical side, instead of transmitting $k + 2s$ polynomial evaluations, we send k coefficients + $2s$ evaluations.

This is because we are interested in efficiency in the *common* case, where in most of cases there are no errors and we care more about having a fast check that there are no errors than of the error recovering phase.

Furthermore, in our use case in the context of FRI IOP, we are not interested into decoding but only into detecting errors.

1.2.1 Encoding

Let $g(x)$ be the generator polynomial

$$g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^{2s-1})$$

whith α being a primitive element of $GF(p^r)$.

The *encoder* wants to map the message $\{m_0, m_1, \ldots, m_{k-1}\}$ into a polynomial $p(x)$ of degree $< k$ such that

$$p(\alpha_i) = m_i \quad \forall i \in 0, k-1$$

so we interpolate taking $\{m_0, m_1, \ldots, m_{k-1}\}$ as the coefficients of $p(x)$ in order to obtain

$$p(x) = m_0 + m_1 \cdot x + \cdots + m_{k-1}x^{k-1}$$

Then the encoder computes $c(x) = p(x) \cdot g(x)$, and will output the coefficients of $c(x)$, $\{c_i\}^k$.

Notice that $c(x) = q(x)g(x)$, thus $c(x)$ is a multiple of $g(x)$.

1.2.2 Error checking

The *receiver* starts from the received coefficients $\{c'_i\}^k$ of $c'(x)$. $c'(x)$ may contain errors, which we represent by $e(x) = e_0 + e_1x + e_2x^2 + \ldots + e_{k-1}x^{k-1}$, so $c'(x) = c(x) + e(x)$.

In order to check $c(x) \stackrel{?}{=} c'(x)$ (thus, that there are no errors ($e(x) = 0$)), we will do the divisibility check $g(x)|c'(x)$, which if there are no errors the check should pass. This comes from the fact that we've chosen $c(x)$ to be a multiple of $g(x)$. If the check passes, we'll extract the first k elements of $c'(x)$.

If $c'(x)/g(x) = 0$, it means that $c'(x)$ is a multiple of $g(x)$, thus $c(x) = c'(x)$, which means that there are no errors.

In practice, checking divisibility by dividing $c'(x)$ by $g(x)$ could be computationally expensive, that's why we do the following approach.

The check polynomial

From $g(x) = (x - \alpha)(x - \alpha^2) \cdots (x - \alpha^{2s-1})$, we know that

$$g(x) = \prod_{i=0}^{2s-1} (x - \alpha^i) = x^{2s} - 1$$

a proof for $\prod_{i=0}^{n-1}(x - \alpha^i) = x^n - 1$ can be found in [3].

Now, let $g'(x) = \prod_{i=0}^{n-1}(x - \alpha^i)$, which for $n = k + 2s$, is equivalent to

$$g'(x) = \prod_{i=0}^{n-1} (x - \alpha^i) = \prod_{i=0}^{2s-1} (x - \alpha^i) \cdot \prod_{i=2s-1}^{n-1} (x - \alpha^i)$$

where we see clearly that $g'(x)$ is a multiple of $g(x)$, so, we have that $g(x)|g'(x)$, which is the same than saying that

$$g(x)|(x^n - 1),\ for\ 2s < n$$

From here, as we know that $c(x)$ is a multiple of $g(x)$, we can see that there is a unique polynomial $h(x)$ of degree $n - 2s$ such that

$$g(x)h(x) = x^n - 1$$

So, $h(x) = (x^n - 1)/g(x)$, and we can see that

$$h(x) = (x^n - 1)/g(x) = \prod_{2s-1}^{n-1} (x - \alpha^i)$$

The degree of $h(x)$ being $n - 2s$ can be seen from the fact that $deg(g(x)) = 2s$ and $deg(x^n - 1) = n$, thus

$$deg(h(x)) = deg((x^n - 1)/g(x)) = n - 2s$$

which is equal to k.

The polynomial $h(x)$ receives the *check polynomial* name.

From here, to determine if $c'(x)$ is a valid codeword we check:

$$c'(x) \cdot h(x) = 0 \pmod{x^n - 1}$$

It is common to choose $n = p^r - 1$ such as $n = 2^8 - 1$, and $k = 223$ (*RS(255, 223)*), over $GF(2^8)$.

A Vandermonde determinant

This section contains a simple proof of the Vandermonde matrix determinant. The proof was found in [4], here it is written with small modifications and expanded. A different proof can be found in [5].

$$\det\ V(x_1,...,x_n) = \det \begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \dots & x_3^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} \end{pmatrix} = \prod_{j<i}(x_i - x_j)$$

Proof

The equation is true for $n = 2$, as we have:

$$\det\ V(x_1,x_2) = \det \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \end{pmatrix} = x_2 - x_1$$

We now assume that $n \geq 3$. Starting with column n and ending at column 2, we successively apply the following operation: subtract x_1 times column $i-1$ from column i. The determinant remains the same and the resulting matrix is:

$$= \det \begin{pmatrix} 1 & x_1 - (x_1 \cdot 1) & x_1^2 - (x_1 \cdot x_1) & \dots & x_1^{n-1} - (x_1 \cdot x_1^{n-2}) \\ 1 & x_2 - (x_1 \cdot 1) & x_2^2 - (x_1 \cdot x_2) & \dots & x_2^{n-1} - (x_1 \cdot x_2^{n-2}) \\ 1 & x_3 - (x_1 \cdot 1) & x_3^2 - (x_1 \cdot x_3) & \dots & x_3^{n-1} - (x_1 \cdot x_3^{n-2}) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n - (x_1 \cdot 1) & x_n^2 - (x_1 \cdot x_n) & \dots & x_n^{n-1} - (x_1 \cdot x_n^{n-2}) \end{pmatrix}$$

$$= \det \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & (x_2 - x_1) & x_2(x_2 - x_1) & \dots & x_2^{n-2}(x_2 - x_1) \\ 1 & (x_3 - x_1) & x_3(x_3 - x_1) & \dots & x_3^{n-2}(x_3 - x_1) \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & (x_n - x_1) & x_n(x_n - x_1) & \dots & x_n^{n-2}(x_n - x_1) \end{pmatrix}$$

Applying the Laplace expansion along the first row and factoring out the scalar multiples $(x_i - x_1)$ in each column, we obtain:

$$= \det \begin{pmatrix} x_2 - x_1 & x_2^2 - x_1x_2 & x_2^3 - x_1x_2^2 & \dots & x_2^{n-1} - x_1x_2^{n-2} \\ x_3 - x_1 & x_3^2 - x_1x_3 & x_3^3 - x_1x_3^2 & \dots & x_3^{n-1} - x_1x_3^{n-2} \\ \vdots & \vdots & \vdots & & \vdots \\ x_n - x_1 & x_n^2 - x_1x_n & x_n^3 - x_1x_n^2 & \dots & x_n^{n-1} - x_1x_n^{n-2} \end{pmatrix}$$

in each row, factor out $(x_i - x_1)$

$$= \det \begin{pmatrix} 1(x_2 - x_1) & x_2(x_2 - x_1) & x_2^2(x_2 - x_1) & \dots & x_2^{n-2}(x_2 - x_1) \\ 1(x_3 - x_1) & x_3(x_3 - x_1) & x_3^2(x_3 - x_1) & \dots & x_3^{n-2}(x_3 - x_1) \\ \vdots & \vdots & \vdots & & \vdots \\ 1(x_n - x_1) & x_n(x_n - x_1) & x_n^2(x_n - x_1) & \dots & x_n^{n-2}(x_n - x_1) \end{pmatrix}$$

$$= \prod_{2 \le i \le n} (x_i - x_1) \quad \det \begin{pmatrix} 1 & x_2 & x_2^2 & \dots & x_2^{n-2} \\ 1 & x_3 & x_3^2 & \dots & x_3^{n-2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-2} \end{pmatrix}$$

now, by induction

$$= \prod_{i=2}^{n} (x_i - x_1) \quad \cdot \prod_{2 \le j < i \le n} (x_i - x_j) \quad = \prod_{1 \le j < i \le n} (x_i - x_j)$$

□

References

[1] G. Solomon I. S. Reed. Reed-Solomon Codes paper. `https://faculty.math.illinois.edu/~duursma/CT/RS-1960.pdf`.

[2] Bruce Maggs. Reed-solomon. `https://courses.cs.duke.edu/spring10/cps296.3/rs_scribe.pdf`.

[3] Alex Kampa. Lagrange bases in subgroups of fp*: a hands-on introduction. `https://github.com/aragon/research/blob/main/blog/001_Lagrange/lagrange.pdf`.

[4] Pedro Tamaroff. Vandermonde determinant proof. `https://math.stackexchange.com/questions/525334/vandermonde-determinant-by-induction`.

[5] Thomas W. Judson. Abstract algebra, theory and applications. *sec.22.2*.