Generalised Reed-Solomon codes. Decoding methods

PDF created on November 22, 2018

Decoding GRS codes with Euclid's algorithm + Forney's method

Reminder:

- Parity-check matrix of GRS code ($d-2=n-k-1$):

$$H=\begin{pmatrix} v_1 & v_2 & \dots & v_n \\ v_1\alpha_1 & v_2\alpha_2 & \dots & v_n\alpha_n \\ \vdots & \vdots & \vdots & \vdots \\ v_1\alpha_1^{d-2} & v_2\alpha_2^{d-2} & \dots & v_n\alpha_n^{d-2} \end{pmatrix}.$$

- We can calculate:

  The syndrome of the received word $\mathbf{y}\in\mathbb{F}^n$:

$$\mathbf{s}^\intercal=(s_0,s_1,\dots,s_{d-2})^\intercal=H\mathbf{y}^\intercal\,,$$

  and the syndrome polynomial:

$$S(x)=\sum_{l=0}^{d-2}s_l x^l\,.$$

- Unknown values:

  Error vector:

$$\mathbf{e}=\mathbf{y}-\mathbf{c}\,,\quad J=\{j\,|\,e_j\neq 0\}\text{– positions of errors.}$$

  Error locator:

$$\Lambda(x)=\prod_{j\in J}(1-\alpha_j x)$$

  Error evaluator:

$$\Gamma(x)=\sum_{j\in J}e_j v_j\prod_{m\in J\setminus\{j\}}(1-\alpha_m x)\,.$$

- Key equation of decoding of GRS codes:

  1. $\gcd(\Lambda,\Gamma)=1$.
  2. $\deg\Gamma=|J|-1$, $\deg\Lambda=|J|\le\tau=\left\lfloor\frac{d-1}{2}\right\rfloor$
  3. $S(x)\Lambda(x)\equiv\Gamma(x)\,(\mod x^{d-1})$.

     Additionally: check that $\Lambda(0)=1$.

Summary. Methods of decoding of GRS codes.

Step 1. Calculate syndrome polynomial of received word $\mathbf{y} \in \mathbb{F}^n$:

$$\mathbf{s}^\intercal = (s_0, s_1, \ldots, s_{d-2})^\intercal = H\mathbf{y}^\intercal\,,$$

$$S(x) = \sum_{l=0}^{d-2} s_l x^l\,.$$

Step 2. Solve the key equation.

Peterson-Gorenstein-Zierler

- Solve the third equation of the key equation by assuming $\Lambda(x) = \lambda_0 + \lambda_1 x + \cdots + \lambda_\tau x^\tau$ and $\Gamma(x) = \gamma_0 + \gamma_1 x + \cdots + \gamma_{\tau-1} x^{\tau-1}$ and equating coefficients of equal powers of x.[1]
- Calculate $d(x) = \gcd(\Lambda(x), \Gamma(x))$ and if $\deg d > 0$ (i.e. if $d(x)$ is not constant), divide both $\Lambda(x)$ and $\Gamma(x)$ by $d(x)$:
$$\Lambda(x) \leftarrow \frac{\Lambda(x)}{d(x)}\,, \quad \Gamma(x) \leftarrow \frac{\Gamma(x)}{d(x)}\,.$$
This will ensure the first equation in the key equation.
- If $c = \Lambda(0) \neq 1$, divide:
$$\Lambda(x) \leftarrow c^{-1} \cdot \Lambda(x)\,, \quad \Gamma(x) \leftarrow c^{-1} \cdot \Gamma(x)\,.$$

Euclid's algorithm

- Apply (extended) Euclid's algorithm (see the algorithm below) to
$$a(x) \leftarrow x^{d-1} \text{ and } b(x) \leftarrow S(x)\,,$$
to produce
$$\Lambda(x) \leftarrow t_h(x) \text{ and } \Gamma(x) \leftarrow r_h(x)\,,$$
where h is the smallest index i for which $\deg r_i < \frac{d-1}{2}$.
- If $c = \Lambda(0) \neq 1$, divide:
$$\Lambda(x) \leftarrow c^{-1} \cdot \Lambda(x)\,, \quad \Gamma(x) \leftarrow c^{-1} \cdot \Gamma(x)\,.$$

Step 3. Calculate error values.

Peterson-Gorenstein-Zierler

Find roots of $\Lambda(x)$. They are exactly $\{\alpha_j^{-1} \mid \text{there is error in position } j\}$. Calculate straightforward from definition of $\Gamma(x)$ the values of errors.

Forney's algorithm

For $j = 1, 2, \ldots, n$ calculate:

$$e_j = \begin{cases} -\frac{\alpha_j}{v_j} \cdot \frac{\Gamma(\alpha_j^{-1})}{\Lambda'(\alpha_j^{-1})} & \text{if } \Lambda(\alpha_j^{-1}) = 0\,, \\ 0 & \text{otherwise.} \end{cases}$$

Note. The methods in Step 2 and 3 can be combined voluntarily. For example, you could solve step 2 by Peterson-Gorenstein-Zierler and then step 3 – by Forney's algorithm; and so on.

---

[1] This is equivalent to solving the following system of linear equations (in matrix form):

$$\left(\begin{array}{cccccc} s_0 & 0 & 0 & \cdots & 0 & 0 \\ s_1 & s_0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{\tau-1} & s_{\tau-2} & s_{\tau-3} & \cdots & 0 & 0 \\ \hline s_\tau & s_{\tau-1} & s_{\tau-2} & \cdots & s_1 & s_0 \\ s_{\tau+1} & s_\tau & s_{\tau-1} & \cdots & s_2 & s_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{d-2} & s_{d-3} & s_{d-4} & \cdots & s_{d-\tau-1} & s_{d-\tau-2} \end{array}\right) \cdot \begin{pmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_\tau \end{pmatrix} = \left(\begin{array}{c} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{\tau-1} \\ \hline 0 \\ 0 \\ \vdots \\ 0 \end{array}\right)$$

Note that lower equations (below the line) do not involve γ coefficients so they can be solved first. Then, the obtained values can be used in the upper equations to find γ coefficients.

Extended Euclid's algorithm

$$
\begin{array}{l}
r_{-1}(x) = a(x); \quad r_0(x) = b(x); \\
s_{-1}(x) = 1; \qquad\; s_0(x) = 0; \\
t_{-1}(x) = 0; \qquad\; t_0(x) = 1; \\
\textbf{for } (\ i = 1;\ r_{i-1}(x) \neq 0;\ i{+}{+}\ )\ \{ \\
\qquad r_{i-2}(x) = \underline{q_i(x)} \cdot r_{i-1}(x) + \underline{r_i(x)}; \\
\qquad s_{i-2}(x) = q_i(x) \cdot s_{i-1}(x) + \underline{s_i(x)}; \;\leftarrow\; \text{not needed for decoding} \\
\qquad t_{i-2}(x) = q_i(x) \cdot t_{i-1}(x) + \underline{t_i(x)}; \\
\}
\end{array}
$$

Note 1. Underlined values are to be found on ith iteration. Note that quotient $q_i(x)$ is the same during one iteration and it is defined from polynomial division with remainder of $r_{i-2}(x)$ by $r_{i-1}(x)$. Hence, you first find $q_i(x)$ and $r_i(x)$, and then use the obtained $q_i(x)$ to calculate $t_i(x)$.

Note 2. For decoding of GRS codes with (extended) Euclid's algorithm you don't need polynomials $s_i(x)$, so you can omit the second line inside the loop.

Note 3. It might be helpful to see the symmetry of the iterations – note that all $\{r_i(x)\}$, $\{s_i(x)\}$ and $\{t_i(x)\}$ are obtained recursively by the same rule from two preceding iterations.