Ссылка на статью:
// Ученые записки УлГУ. Сер. Математика и информационные технологии. 2023. № 2, с. 35-42.

Поступила: 28.11.2023
Окончательный вариант: 28.11.2023

УДК 519.725

Декодирование кодов Рида-Соломона

Лавриненко А.Д. [*]***, Степанова Е.А.***

[*]anutalavrinenko@gmail.com
УлГУ, Ульяновск, Россия

В работе исследуются алгоритмы декодирования кодов Рида-Соломона на основе алгоритма Берлекэмпа-Месси и алгоритма Сугиямы. Приводятся числовые примеры декодирования кодов Рида-Соломона с использованием данных алгоритмов над полем $GF(2^3)$. Данная работа может помочь в программной реализации декодеров кода Рида-Соломона.

***Ключевые слова**: коды Рида-Соломона, алгоритм Берлекэмпа-Месси, алгоритм Сугиямы*

Введение

Алгебраические алгоритмы декодирования кодов БЧХ (в том числе кодов Рида-Соломона) делятся на две группы: синдромные и бессиндромные. В данной работе рассматриваются коды Рида-Соломона и способы их декодирования с использованием синдромных алгоритмов декодирования: алгоритма Берлекэмпа-Месси и алгоритма Сугиямы. Обсуждение рассматриваемой темы начато в работах [1, 2].

1. Алгоритм декодирования Берлекэмпа-Месси

Кодом Рида-Соломона называется код БЧХ над полем GF(q), где $q > 2$, который имеет длину $q - 1$. Коды Рида-Соломона являются МДР-кодами, то есть кодами с максимально достижимым расстоянием $d = n - k + 1$.

Для приводимых в работе алгоритмов декодирования будет рассмотрен случай, когда в канале связи действуют ошибки (число стираний $s = 0$).

Процесс декодирования делится на несколько шагов, а именно: 1) вычисление компонентов синдромного вектора на основе полученного вектора; 2) нахождение многочлена локаторов ошибок σ(x); 3) нахождение корней многочлена σ(x), по которым определяются позиции ошибок; 4) нахождение значений ошибок (для недвоичных кодов).

Данные алгоритмы синдромного декодирования имеют сложность $O(rn)$ [4]. Если ошибок немного, то такие алгоритмы являются практичными, к примеру, алгоритм Сугиямы хорошо приспособлен к эффективной аппаратной реализации, а алгоритм Берлекэмпа-Месси, имея меньшее число операций в конечном поле, применяется в программных декодерах.

Использование алгоритма Берлекэмпа-Месси для данных кодов позволяет уменьшить сложность нахождения многочлена локаторов ошибок σ(x) до величины порядка t^2.

Ниже приводится алгоритм Берлекэмпа-Месси нахождения многочлена локаторов ошибок.

Алгоритм 1. (алгоритм Берлекэмпа—Месси).

Вход: последовательность $a_1, \dots, a_n$ над некоторым полем F.

Выход: LFSR (L, f(x)) минимальной длины L, для которого:

$$a_j = -\sum_{i=1}^{L} f_i a_{j-i}, \quad j = L+1, L+2, \dots, n$$

1. Определить $r := 0, f(x) := 1, b(x) := 1, L := 0$.

2. Цикл $r := 1, \dots, n$:

2.1. Определить $\Delta := a_r + \sum_{i=1}^{L} f_i a_{j-i}$

2.2. Если Δ=0, то $b(x) := x \cdot b(x)$.

2.3. Если $\Delta \neq 0$:

2.3.1. Если $2L < r$

$buf(x) := f(x) - \Delta \cdot x \cdot b(x)$.

$b(x) := \Delta^{-1} \cdot f(x)$,

$f(x) := buf(x)$,

$L := r - L$.

2.3.2. Иначе (т.е. выполнено $2L \geq r$):

$f(x) := f(x) - \Delta \cdot x \cdot b(x)$,

$b(x) := x \cdot b(x)$.

Для реализации алгоритма декодирования кодов Рида-Соломона лучше использовать алгоритм Берлекэмпа-Месси с упрощенными вычислениями вида x^d.

Алгоритм 2. (алгоритм Берлекэмпа—Месси с упрощенными вычислениями вида x^d).

Вход: последовательность $a_1, \dots, a_n$ над некоторым полем F.

Выход: LFSR (L, f(x)) минимальной длины L, для которого:

$$a_j = -\sum_{i=1}^{L} f_i a_{j-i}, \quad j = L+1, L+2, \dots, n$$

1. Определить $r := 0, f(x) := 1, b(x) := 1, L := 0, d := 0$.

2. Цикл $r := 1, \dots, n$:

2.1. Определить $\Delta := a_r + \sum_{i=1}^{L} f_i a_{j-i}, d := d + 1$

2.2. Если $\Delta \neq 0$:

2.2.1. Если $2L < r$

$buf(x) := f(x) - \Delta \cdot x^d \cdot b(x)$.

$b(x) := \Delta^{-1} \cdot f(x)$,

$f(x) := buf(x)$,

$L := r - L$,

$d := 0$.

2.2.2. Иначе (т.е. выполнено $2L \geq r$):

$f(x) := f(x) - \Delta \cdot x^d \cdot b(x)$.

Алгоритм декодирования кодов Рида-Соломона на основе алгоритма Берлекэмпа-Месси имеет следующий вид [3].

Алгоритм 3 (декодирование кода Рида-Соломона на основе алгоритма Берлекэмпа-Месси).

Вход: полученный вектор v.

Выход: исходный кодовый вектор u, если произошло не более $\left[\frac{d-1}{2}\right]$ ошибок.

1. Определяется $t = \left[\frac{d-1}{2}\right]$. Находятся компоненты $S_1, S_2, \ldots, S_{2t}$ синдромного вектора: $S_i = v(\alpha^i)$, $i = 1,2, \ldots, 2t$. Если синдромный вектор нулевой, то алгоритм завершается и возвращается u = v.

2. Для последовательности $S_1, S_2, \ldots, S_{2t}$ с помощью алгоритма 1 находится значение многочлена локаторов ошибок σ(x). Пусть l = deg σ(x).

3. Отыскиваются l корней многочлена σ(x) с использованием алгоритма 4, приведенного ниже. При этом локаторы ошибок – это величины, обратные корням многочлена σ(x).

4. По формулам Форни

$$Y_i = \frac{X_i^{-1} w(X_i^{-1})}{\prod_{j \neq i}(1 - X_j X_i^{-1})}, \qquad i = 1, \ldots, l$$

где $l = \deg \sigma(x), \omega(x) = \sigma(x)S(x) \pmod{x^{2t}}$, находятся значения ошибок Y_i.

У вектора v из X_i символа вычитается значение Y_i, $i = 1, \ldots, l$. При этом получается кодовый вектор u.

Для нахождения корней многочлена локаторов ошибок σ(x) используется следующий алгоритм решения квадратного уравнения над полем $GF(2^m)$ [4].

Алгоритм 4 (решение квадратного уравнения над полем $GF(2^m)$).

Вход: многочлен $ax^2 + bx + c$ над $GF(2^m)$, $a \neq 0$.

Выход: корни исходного многочлена (в случае его приводимости), либо заключение о том, что многочлен неприводим над $GF(2^m)$.

1. Если $c = 0$, то $x_1 = 0$, $x_2 = a^{-1}b$ – корни исходного уравнения. На этом алгоритм заканчивается.
2. Если $b = 0$, то уравнение имеет корень: $x_1 = x_2 = (a^{-1}c)^{2^{m-1}}$ кратности 2. На этом алгоритм завершается.

3. В исходном уравнении произведем замену $x = b\tilde{x}/a$. Тогда

$ax^2 + \mathrm{bx} + \mathrm{c} = \frac{b^2}{a}\left(\tilde{x}^2 + \tilde{x} + \frac{ac}{b^2}\right) = \gamma(\tilde{x}^2 + \tilde{x} + \beta)$, где $\beta = \frac{ac}{b^2}$, $\gamma = \frac{b^2}{a}$. Рассмотрим уравнение:

$$\tilde{x}^2 + \tilde{x} + \beta = 0$$

Если $Tr_m(\beta) = 1$, то исходный многочлен неприводим над GF(2^m). На этом алгоритм завершается.

4. Находим первый корень y_1 уравнения $\tilde{x}^2 + \tilde{x} + \beta = 0$ с помощью формулы

$$y = \sum_{i=0}^{m-2}\left(\sum_{j=0}^{i}\beta^{2^j}\right)\delta^{2^i},$$

где δ – произвольный фиксированный элемент поля GF(2^m) с условием $Tr_m(\beta) = 1$.

При этом если m нечетно, то находим по формуле

$$y_1 = \beta^2 + \beta^{2^3} + \beta^{2^5} + \cdots + \beta^{2^{m-2}}.$$

Второй корень равен $y_2 = y_1 + 1$.

5. Корнями исходного уравнения будут:

$$x_1 = \frac{by_1}{a}, x_2 = \frac{by_2}{a}.$$

Далее приводится пример декодирования входного вектора v на основе алгоритма Берлекэмпа-Месси, в результате чего в нем будут исправлены ошибки.

Пример 1. Рассмотрим расширение поля $\mathrm{GF}(2)$? $\mathrm{GF}(2^3)$. Пусть поле $\mathrm{GF}(2^3)$ строится на основе примитивного многочлена $\mathrm{p(x)} = x^3 + \mathrm{x} + 1$, α – примитивный элемент поля:

$\alpha^0 = 1 = 100$
$\alpha^1 = \alpha = 010$
$\alpha^2 = \alpha^2 = 001$
$\alpha^3 = 1 + \alpha = 110$
$\alpha^4 = \alpha + \alpha^2 = 011$
$\alpha^5 = 1 + \alpha + \alpha^2 = 111$
$\alpha^6 = 1 + \alpha^2 = 101$
$\alpha^7 = 1 = 100$

Рассмотрим код Рида-Соломона с параметрами n = 7, k = 3, d = 5. В этом случае код исправляет не более двух ошибок.

Рассмотрим случай возможности исправления до двух ошибок. Пусть на приемном конце получен вектор:

$$\mathrm{v} = (\alpha^2, 1, \alpha, \alpha^4, \alpha^2, \alpha^6, 1),$$

в котором не более двух ошибок. Используем для декодирования алгоритм 3.

1. Полагаем, что $t = \left[\frac{d-1}{2}\right] = \left[\frac{5-1}{2}\right] = 2$. Перезапишем вектор v в виде многочлена:

$$\mathrm{v(x)} = \alpha^2 + \mathrm{x} + \alpha\mathrm{x}^2 + \alpha^4\mathrm{x}^3 + \alpha^2\mathrm{x}^4 + \alpha^6\mathrm{x}^5 + \mathrm{x}^6$$

Вычислим компоненты синдрома для вектора v:

$S_1 = v(\alpha) = \alpha^2 + \alpha + \alpha^3 + \alpha^7 + \alpha^6 + \alpha^{11} + \alpha^6 = \alpha^2 + \alpha + \alpha^3 + 1 + \alpha^4 = \alpha$

$S_2 = v(\alpha^2) = \alpha^2 + \alpha^2 + \alpha^5 + \alpha^{10} + \alpha^{10} + \alpha^{16} + \alpha^{12} = \alpha^5 + \alpha^2 + \alpha^5 = \alpha^2$

$S_3 = v(\alpha^3) = \alpha^2 + \alpha^3 + \alpha^7 + \alpha^{13} + \alpha^{14} + \alpha^{21} + \alpha^{18} = \alpha^2 + \alpha^3 + 1 + \alpha^6 + 1 + 1 + \alpha^4 = \alpha^6$

$S_4 = v(\alpha^4) = \alpha^2 + \alpha^4 + \alpha^6 + \alpha^{16} + \alpha^{18} + \alpha^{26} + \alpha^{24} = \alpha^2 + \alpha^4 + \alpha^2 + \alpha^2 + \alpha^4 + \alpha^5 + \alpha^3 = 0$

Синдромный многочлен будет иметь следующий вид:

$S(x) = \alpha + \alpha^2 x + \alpha^6 x^2$

2. На вход алгоритма 1 передаем последовательность $S_1 = \alpha,\ S_2 = \alpha^2,\ S_3 = \alpha^6$.

Таблица 1. Нахождение многочлена локаторов ошибок с применением алгоритма Берлекэмпа-Месси

r	Δ	σ(x)	b(x)	L
0		1	1	0
1	α	$1 + \alpha x$	α^6	1
2	0	$1 + \alpha x$	$\alpha^6 x$	1
3	α^6	$1 + \alpha x + \alpha^3 x$	$\alpha^3 + \alpha^6 x$	2
4	α^4	$1 + \alpha^3 x + x^2$	$\alpha^3 x + \alpha^4 x^2$	2

Получаем $\sigma(x) = 1 + \alpha^3 x + x^2$.

3. Найдем корни многочлена $\sigma(x) = x^2 + \alpha^3 x + 1$, используя алгоритм 3.

Вычислим $\gamma = \frac{b^2}{a} = \alpha^6, \beta = \frac{ac}{b^2} = \frac{1}{\alpha^6} = \alpha$.

Запишем $1 + \alpha^3 x + x^2 = \alpha^6 (\tilde{x}^2 + \tilde{x} + \alpha)$.

Найдем значения $y_1 = \beta^2 = \alpha^2,\ y_2 = \beta^2 + 1 = \alpha^2 + 1 = \alpha^6$.

Отсюда $x_1 = \frac{\alpha^6}{1} \cdot \alpha^2 = \alpha^5,\ x_2 = \frac{\alpha^3}{1} \cdot \alpha^6 = \alpha^2$.

Получим $X_1 = x_1^{-1} = \alpha^2,\ X_2 = x_2^{-1} = \alpha^5$.

2, 5 – позиции ошибок в исходном векторе.

4. После того, как все локаторы ошибок известны, воспользуемся формулой Форни для нахождения значений ошибок.

$$Y_i = \frac{X_i^{-1} w(X_i^{-1})}{\prod_{j \neq i}(1 - X_j X_i^{-1})}, \qquad i = 1, \dots, s$$

Найдем $\omega(x)$:

$\omega(x) = \sigma(x)S(x) \pmod{x^{2t}} = (1 + \alpha^3 x + x^2) \cdot (\alpha + \alpha^2 x + \alpha^6 x^2) = \alpha + \alpha x$.

Получим $Y_1 = \frac{X_1^{-1} w(X_1^{-1})}{1 - X_2 X_1^{-1}} = \frac{\alpha^5(\alpha + \alpha^6)}{1 + \alpha^{10}} = \alpha^2$, $Y_2 = \frac{X_2^{-1} w(X_2^{-1})}{1 - X_1 X_2^{-1}} = \frac{\alpha^2(\alpha + \alpha^3)}{1 + \alpha^4} = \alpha^4$.

α^2, α^4 – значения ошибок.

Таким образом, вектор ошибок $e = (0, 0, \alpha^2, 0, 0, \alpha^4, 0)$.

Получаем кодовый вектор u:

$$u = v - e = (\alpha^2, 1, \alpha, \alpha^4, \alpha^2, \alpha^6, 1) - (0, 0, \alpha^2, 0, 0, \alpha^4, 0) = (\alpha^2, 1, \alpha^4, \alpha^4, \alpha^2, \alpha^3, 1).$$

2. Алгоритм декодирования Сугиямы.

В отличие от алгоритма Берлекэмпа—Месси алгоритм Сугиямы имеет большее число операций в конечном поле. При этом алгоритм Сугиямы хорошо приспособлен к эффективной аппаратной реализации [3].

Ниже приводится алгоритм декодирования кодов Рида-Соломона на основе алгоритма Сугиямы.

Алгоритм 5 (декодирование кода Рида-Соломона на основе алгоритма Сугиямы).

Вход: принятый вектор v.

Выход: исходный кодовый вектор u, если произошло не более $\left[\frac{d-1}{2}\right]$ ошибок.

1. Пусть $t = \left[\frac{d-1}{2}\right]$, где $d = n - r + 1$— кодовое расстояние кода РС. Находятся компоненты $S_1, S_2, .. S_{2t}$ синдромного вектора: $S_i = v(\alpha^i), i = 1,2, \dots, 2t$. Если они все равны нулю, то ошибок нет и алгоритм завершается с возвращением кодового вектора $u = v$. На основе синдромных компонент составляется синдромный многочлен

$$S(x) = \sum_{i=1}^{2t} S_i x^{i-1}.$$

2. Пусть $r_{-1}(x) = x^{2t}, r_0(x) = S(x), v_{-1}(x) = 0, v_0(x) = 1$. Производится последовательность вычислений обобщенного алгоритма Евклида:

$$r_{i-2}(x) = r_{i-1}(x)q_{i-1}(x) + r_i(x),$$
$$v_i(x) = v_{i-2}(x) - v_{i-1}(x)q_{i-1}(x), \qquad i \geq 1,$$

до тех пор, пока для некоторого $r_j(x)$ не будет выполнено условие:

$$\deg r_{j-1}(x) \geq t, \ \deg r_j(x) \leq t - 1.$$

Тогда

$$\sigma(x) = \lambda v_j(x), \qquad \omega(x) = \lambda r_j(x),$$

где константа $\lambda \in GF(q)$ задается так, чтобы удовлетворялось условие $\sigma(0) = 1$.

Пусть $s = \deg \sigma(x)$. Тогда вектор v содержит s ошибок.

3. Отыскиваются s корней многочлена $\sigma(x)$ последовательной подстановкой в него ненулевых элементов поля $GF(q)$. При этом локаторы ошибок — это величины, обратные корням многочлена $\sigma(x)$.
4. Находятся значения ошибок $Y_1, Y_2, .. Y_s$, с помощью метода Форни:

$$Y_i = \frac{X_i^{-1}\omega(X_i^{-1})}{\prod_{j \neq i}(1 - X_j X_i^{-1})}, \qquad i = 1, \dots, s.$$

У вектора v из X_i символа вычитается значение Y_i, i $= 1, \ldots,$ s. При этом получается кодовый вектор u.

После исправления ошибок и получения исходного кодового вектора u происходит извлечение исходного информационного вектора b.

Далее приводится пример декодирования входного вектора v на основе алгоритма Сугиямы, в результате чего в нем будут исправлены ошибки.

Пример 2. Пусть $GF(2^3)$ – поле на основе примитивного многочлена $x^3 + x + 1$ с примитивным элементом α (Пример 1). Рассмотрим [7,3,5] – код РС над $GF(2^3)$. В этом случае $t = \left[\frac{d-1}{2}\right] = 2$. Поэтому код исправляет две или менее ошибок.

На приемном конце получен вектор:

$$v = (1, \alpha^4, \alpha^5, \alpha, 1, \alpha^4, \alpha^3),$$

$$v(x) = 1 + \alpha^4 x + \alpha^5 x^2 + \alpha x^3 + x^4 + \alpha^4 x^5 + \alpha^3 x^6.$$

Требуется найти соответствующий кодовый вектор u.

1. Определяем $t = \left[\frac{d-1}{2}\right] = 2$. Вычисляем элементы синдрома:

$$S_1 = v(\alpha) = \alpha^5, S_2 = v(\alpha^2) = 0,$$

$$S_3 = v(\alpha^3) = \alpha^2, S_2 = v(\alpha^4) = 1.$$

Синдромный многочлен имеет вид:

$$S(x) = \alpha^5 + \alpha^2 x^2 + x^3.$$

2. Определяем $r_{-1}(x) = x^4, r_0(x) = S(x), v_{-1}(x) = 0, v_0(x) = 1$

Применяя неполный алгоритм Евклида, получаем:

$$r_{-1}(x) = r_0(x)q_0(x) + r_1(x),$$

$$q_0(x) = x + \alpha^2,$$

$$r_1(x) = \alpha^4 x^2 + \alpha^5 x + 1,$$

$$v_1(x) = v_{-1}(x) - v_0(x)q_0(x) = x + \alpha^2;$$

$$r_0(x) = r_1(x)q_1(x) + r_2(x),$$

$$q_1(x) = \alpha^3 x + 1,$$

$$r_2(x) = \alpha^2 x + \alpha^4,$$

$$v_2(x) = v_0(x) - v_1(x)q_1(x) = \alpha^3 x^2 + \alpha^4 x + \alpha^6.$$

Так как $\deg r_1(x) = 2 \geq t$, $\deg r_2(x) = 1 \leq t - 1$, то останавливаемся при значении $j = 2$. Тогда

$$\sigma(x) = \lambda v_2(x) = \lambda(\alpha^3 x^2 + \alpha^4 x + \alpha^6).$$

При $\lambda = \alpha$ получаем $\sigma(0) = 1$, поэтому

$$\sigma(x) = \alpha^4 x^2 + \alpha^5 x + 1, \qquad \omega(x) = \alpha^3 x + \alpha^5$$

3. Корнями многочлена $\sigma(x)$ являются значения $x_1 = \alpha^3$, $x_2 = 1$, поэтому $X_1 = {x_1}^{-1} = \alpha^4$, $X_2 = {x_2}^{-1} = \alpha^0$. Это означает что ошибки в принятом векторе произошли на 0 и 4 позиции.

4. Находим значения ошибок с помощью алгоритма Форни:

$$Y_1 = \frac{X_1^{-1}\omega(X_1^{-1})}{1 - X_2 X_1^{-1}} = \alpha^3, \qquad Y_2 = \frac{X_2^{-1}\omega(X_2^{-1})}{1 - X_1 X_2^{-1}} = \alpha^4,$$

Поэтому

$$e = (\alpha^4, 0, 0, 0, \alpha^3, 0, 0),$$
$$u = (\alpha^5, \alpha^4, \alpha^5, \alpha, \alpha, \alpha^4, \alpha^3).$$

Заключение

В работе приведены алгоритмы декодирования кодов Рида-Соломона на основе алгоритма Берлекэмпа-Месси и алгоритма Сугиямы. Задача декодирования кодов Рида-Соломона имеет большое практическое значение. Даже если поврежден значительный объем информации, коды Рида-Соломона позволяют восстановить большую часть потерянной информации. Использование при декодировании алгоритмов Берлекэма-Месси и Сугиямы позволяет уменьшить вычислительную сложность алгоритма декодирования.

Список литературы

1. Рацеев С.М., Лавриненко А.Д., Степанова Е.А. Об алгоритме Берлекэмпа-Месси и его применении в алгоритмах декодирования // *Вестник Самарского Университета. Естественнонаучная серия.* 2021, №1, с. 44-62.
2. Рацеев С.М., Лавриненко А.Д., Степанова Е.А. О декодировании алгебраических кодов на основе алгоритма Берлекэмпа-Месси // *Ученые записки УлГУ. Сер. Математика и информационные технологии*. 2021, № 2, с. 31–42.
3. Рацеев С. М. *Реализации некоторых криптосистем и корректирующих кодов*: учеб. пособие для вузов. СПб.: Лань, 2024. 288 с.
4. Рацеев С.М. *Элементы высшей алгебры и теории кодирования*: учеб. пособие для вузов, 2-е изд., испр. и доп. СПб.: Лань, 2023. 684 с.

Decoding Reed-Solomon codes

Lavrinenko, A.D.**, Stepanova, E.A.

*anutalavrinenko@gmail.com

Ulyanovsk State University, Russia

The paper investigates algorithms for decoding Reed-Solomon codes based on the Berlekamp-Messi algorithm and the Sugiyama algorithm. Numerical examples of decoding Reed-Solomon codes using these algorithms over the GF(2^3) field are given. This work can help in the software implementation of Reed-Solomon code decoders.

***Keywords**: Reed-Solomon codes, Berlekamp-Massey algorithm, Sugiyama algorithm*