КОДЫ РИДА–МАЛЛЕРА: ПРИМЕРЫ ИСПРАВЛЕНИЯ ОШИБОК*

И. В. Агафонова

ivagafonova@home.eltel.net

24 марта 2012 г.

Линейные коды с избыточностью

В двоичных кодах с контролем ошибок (синонимы: «помехоустойчивые коды», «корректирующие коды», «коды обнаружения и исправления ошибок», «коды с избыточностью») кодирование и декодирование заключается в преобразованиях строк из нулей и единиц. Двоичные строки длины n в линейных кодах считаются элементами n-мерного векторного пространства $V_n = (\mathbb{Z}_2)^n$ над конечным полем $\mathbb{Z}_2$.

Знаком $\oplus$ будем обозначать, как это часто делается, суммирование в $\mathbb{Z}_2$, а также операцию сложения в V_n, то есть покомпонентное сложение векторов по модулю 2: $u \oplus v = (u_1 \oplus v_1, \ldots, u_n \oplus v_n)$.

Соответствующее скалярное произведение векторов в V_n определим как

$$u \odot v = u_1\, v_1 \oplus u_2\, v_2 \oplus \cdots \oplus u_n\, v_n, \tag{1}$$

в отличие от обычного скалярного произведения n-мерных векторов

$$\langle u, v\rangle = u_1\, v_1 + u_2\, v_2 + \cdots + u_n\, v_n,$$

которое также будет применяться.

Знак $\odot$, применяемый при перемножении двоичного вектора и двоичной матрицы, будет подчёркивать, что скалярное произведение строки на столбец находится по правилу (1).

*Семинар по дискретному гармоническому анализу и геометрическому моделированию «DHA & CAGD»: `http://www.dha.spb.ru/`

Двоичный линейный код представляет собой множество *кодовых слов* — двоичных векторов вида

$$y = x \odot G, \tag{2}$$

где x пробегает всё множество V_k, $k \geqslant 1$, а матрица G, называемая *порождающей матрицей* данного кода, состоит из k линейно независимых[1] двоичных строк длины n при $n > k$. Имея в виду эти два параметра, код называют $[n,k]$-кодом. *Мощность* $[n,k]$-кода (число всех его кодовых слов) равна 2^k.

Формула (2) задаёт правило кодирования исходного вектора $x \in V_k$ посредством его преобразования в вектор $y \in V_n$. Полученный вектор (кодовое слово) передаётся по каналу связи с возможным искажением в процессе передачи, принимается получателем и декодируется. Кодовое слово y содержит больше символов, чем исходный вектор x, и код должен быть построен так, чтобы благодаря избыточной информации с большой вероятностью определять наличие или отсутствие ошибок, либо, более того, исправлять обнаруженные ошибки.

Как мы видим из (2), все кодовые слова являются линейными комбинациями строк матрицы G, то есть линейный код представляет собой линейное пространство с базисом из строк его порождающей матрицы.

Хорошо изученным семейством линейных кодов с избыточностью являются классические двоичные коды Рида–Маллера (сконструированы в 1954 г.).

Код Рида–Маллера, обозначаемый как $RM(r,m)$, определяется параметрами r и m, где $0 \leqslant r \leqslant m$, r — *порядок кода*, 2^m — *длина кода.*

Коды Рида–Маллера 0-го порядка

Порождающая матрица кода $RM(0,m)$ порядка 0, обозначаемая $G_{0,m}$, определяется как

$$G_{0,m} := G_0(m),$$

где $G_0(m)$ — строка из 2^m единиц:

$$G_0(m) = (1, \ldots, 1).$$

При кодировании по формуле (2) при $G = G_{0,m}$ вектор x должен иметь длину 1, а его кодовым словом будет вектор длины 2^m, состоящий из нулей, если $x = 0$, либо из единиц, если $x = 1$, то есть кодирование какого-то сообщения проводится посимвольно, и $RM(0,m)$ — просто код с 2^m-кратным

[1] Линейная зависимость или независимость основывается здесь на определении линейной комбинации векторов как их суммы (операция $\oplus$) с коэффициентами из $\mathbb{Z}_2$, то есть просто суммы каких-то строк матрицы по модулю 2.

повторением каждого символа x. Кодовых слов всего два: нулевой вектор и вектор из единиц.

Если в полученном слове не все символы одинаковы, мы заключаем, что при передаче сообщения имелись ошибки. Декодирование при этом очевидно: если большая часть битов в полученном векторе — единицы, то считаем, что $x = 1$, если нули — считаем, что $x = 0$. Если ошибочными были меньше половины битов, то есть не более $2^{m-1} - 1$, то ошибки будут исправлены. Но такое кодирование очень невыгодно: объём передаваемого сообщения будет в 2^m раз больше объёма полезной информации!

Коды Рида–Маллера 1-го порядка

Зафиксируем $m \geqslant 1$ и составим матрицу $G_1(m)$ из m строк и 2^m столбцов, где j-й столбец (при нумерации от 0) соответствует двоичному представлению числа j, записанному в m битах.

Например,

$$G_1(1) = \begin{pmatrix} 0 & 1 \end{pmatrix}, \quad G_1(4) = \begin{pmatrix} 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \end{pmatrix}.$$

Как давно замечено, строки матрицы $G_1(m)$ можно выписать чисто механически, чередуя при $i = 1, 2, \ldots, m$ в i-й строке группы из 2^{m-i} нулей и такого же числа единиц.

Порождающей матрицей кода $RM(1, m)$ — кода Рида–Маллера порядка 1 длины 2^m — называется матрица $G_{1,m}$, составленная сверху вниз из строк матриц $G_0(m)$, $G_1(m)$:

$$G_{1,1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad G_{1,4} = \begin{pmatrix} 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\ 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\ 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \end{pmatrix}.$$

Матрица $G_{1,m}$ отвечает требованию линейной независимости строк. В этом можно убедиться, составив квадратную подматрицу из столбцов, содержащих не более двух единиц (их номера 0 и 2^i, $i = 0, 1, \ldots, m-1$). В случае $m = 4$

эта подматрица имеет вид

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

При любом m строки и столбцы такой подматрицы, очевидно, линейно независимы.

Кодирование проводится по обычному для линейного кода правилу (2) при $G = G_{1,m}$. Размер исходного слова при этом равен $m+1$, по числу строк матрицы $G_{1,m}$. Так, код $RM(1,4)$ переводит слово $x = 10011$ в кодовое слово $y = xG = 1001\,1001\,1001\,1001$. (Пробелы между группами битов вставлены для более удобного зрительного восприятия длинного вектора).

Код $RM(1,m)$ с базисом из строк матрицы $G_{1,m}$ при $m > 1$ может быть построен рекуррентно на основе векторов $u \in RM(1,m-1)$ конкатенациями (соединениями) $u|u$ и $u|(1 \oplus u)$,[2] так что список кодовых слов можно записать как

$$RM(1,m) = \begin{Bmatrix} RM(1,m-1) \mid RM(1,m-1) \\ RM(1,m-1) \mid \overline{RM}(1,m-1) \end{Bmatrix}, \qquad (3)$$

где набор векторов $\overline{RM}(1,m-1)$ составлен из векторов $u \oplus 1$, $u \in RM(1,m-1)$ (порядок векторов не имеет значения).

Например, код $RM(1,2)$ можно получить, складывая между собой всевозможные подмножества строк матрицы $G_{1,2} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$ и добавив нулевой вектор, а можно — используя $RM(1,1) = \begin{Bmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{Bmatrix}$ и конструкцию (3). В обоих случаях получается набор кодовых слов $RM(1,2) = \begin{Bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{Bmatrix}$.

[2] Это частный случай способа конструирования кодов, называемого конструкцией Плоткина или конструкцией «$u \mid u \oplus v$», см. [1].

Как видно из $RM(1,1)$ и из (3), кодовые слова кода $RM(1,m)$, кроме двух, состоящих только из единиц или нулей, содержат единиц и нулей поровну, по 2^{m-1}.

Код $RM(1,m)$ известен и под другим названием — код Адамара, поскольку его кодовые слова — это строки определённых ниже двоичных матриц A_m, $\overline{A}_m$.

Двоичной матрицей Адамара назовём матрицу

$$A_m = \frac{J + H_m}{2},$$

где H_m — матрица Адамара H_m, строящаяся по рекуррентному правилу

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},\ H_m = \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix},\ m = 2, 3, \dots,$$

а через J обозначена матрица из одних единиц того же размера, что и H_m. Операция сложения здесь, конечно, означает обычное сложение целых чисел.

Матрицу $\overline{A}_m = J - A_m$ составим из дополнений к элементам матрицы A_m (дополнением 0 считаем 1 и наоборот).[3]

Двоичная матрица A_m отличается от соответствующей H_m только заменой всех элементов, равных -1, нулями. Например,

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},\ A_1 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix},\ H_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix},\ A_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

Сравнивая правило построения A_m, $\overline{A}_m$ и конструирование $RM(1,m)$ по (3), убеждаемся, что получаемые коды идентичны.

Кодовое расстояние

Корректирующая способность всякого кода характеризуется его *кодовым расстоянием*, определяемым как минимальное расстояние Хэмминга $\rho(u,v)$ между векторами u, v этого кода, то есть как число несовпадающих компонент этих векторов. Чем больше это расстояние, тем проще распознать кодовое слово, если оно слегка искажено.

Заметим, что расстояние Хэмминга $\rho(u,v)$ измеряется числом единиц в сумме $u \oplus v$ (иными словами, *весом Хэмминга* вектора $u \oplus v$), а эта сумма

[3]Иногда, как, например, в [1], именно матрицу, которая здесь введена как $\overline{A}_m$, называют двоичной матрицей Адамара. Это лишь вопрос удобства обозначений.

для линейного кода сама является кодовым словом, причём при $u \neq v$ — ненулевым. Так что кодовое расстояние кода вычисляется как минимальный вес его ненулевого кодового слова.

Пусть при использовании кода с кодовым расстоянием d по каналу связи принято некоторое двоичное слово, которое мы обозначим y'. Имеются две возможности:

- принятое слово y' принадлежит коду, и тогда мы считаем, что искажений при передаче не было;
- y' не принадлежит коду. Тогда нам ясно, что при передаче были допущены ошибки. Мы исправляем их, заменив y' кодовым словом, ближайшим к нему в смысле расстояния Хэмминга.

В первом случае вывод об отсутствии ошибок будет обоснованным, если кодовое слово не могло измениться так сильно, чтобы совпасть с другим кодовым словом, то есть если мы полагаем, что y' содержит не более $d-1$ изменённых битов. Имея это в виду, говорят, что код с кодовым расстоянием d обнаруживает $d-1$ ошибку в слове.

Во втором случае мы гарантированно восстановим кодовое слово y, которое в процессе передачи преобразовалось в y', если y и y' различаются не более чем в $\left\lfloor \frac{d-1}{2} \right\rfloor$ битах. В этом случае y' находится в $\left\lfloor \frac{d-1}{2} \right\rfloor$-окрестности только одного кодового слова y, которое и будет ближайшим. Говорят, что код с кодовым расстоянием d исправляет $\left\lfloor \frac{d-1}{2} \right\rfloor$ ошибок.

Для $RM(1,m)$ мы уже определили минимальный вес ненулевого кодового слова, он равен 2^{m-1}. Это и есть кодовое расстояние d данного кода, при этом $\left\lfloor \frac{d-1}{2} \right\rfloor = 2^{m-2}-1$, так что $RM(1,m)$ обнаруживает до $2^{m-1}-1$ ошибок в слове и исправляет до $2^{m-2}-1$ ошибок (включительно), это очень хороший показатель. Но скорость передачи информации, которая измеряется отношением длины исходного слова к длине кодового слова, будет для $RM(1,m)$ невысокой, она равна $\frac{m+1}{2^m}$.

Декодирование кодов Рида–Маллера 1-го порядка по принципу максимального правдоподобия

Декодирование по принципу максимального правдоподобия означает выбор из всех возможных кодовых слов того слова y, которое находится на минимальном расстоянии Хэмминга от принятого слова y', и лишь затем — восстановление исходного слова x из его кода y.

Рассмотрим применение принципа максимального правдоподобия к декодированию кода $RM(1,m)$.

Пусть y' — принятое слово длины 2^m, возможно, содержащее ошибки, и пусть Y' означает вектор, полученный из y' заменой всех компонент 0 компонентами -1:

$$Y' = 2y' - 1. \tag{4}$$

Если c — какое-то кодовое слово (будем обозначать это как $c \in RM(1,m)$), то оно, как отмечалось выше, является строкой двоичной матрицы Адамара A_m или её дополнения $\overline{A}_m$. Тогда соответствующий вектор

$$C = 2c - 1$$

есть строка матрицы H_m или $-H_m$ (будем обозначать это как $C \in H_m$ или $C \in -H_m$). Расстояние $\rho(y', c)$ равно расстоянию $\rho(Y', C)$.

При вычислении скалярного произведения $\langle Y', C\rangle$ каждая несовпадающая компонента даст слагаемое -1, а каждая совпадающая даст 1. Таким образом,

$$\langle Y', C\rangle = N_0 - N_1 = N - 2\rho(Y', C),$$

где N_0 — число совпадений компонент Y' и C, $N_1 = \rho(Y', C)$ — число несовпадений, $N = N_0 + N_1$ — длина слова, $N = 2^m$.

Там, где значение $\rho(Y', C)$ минимально, скалярное произведение $\langle Y', C\rangle$ максимально.

Так как $\max\limits_{C\in\pm H_m} \langle Y', C\rangle = \max\limits_{C\in H_m} |\langle Y', C\rangle|$, то

$$\min_{c\in RM(1,m)} \rho(y', c) = \min_{C\in\pm H_m} \rho(Y', C) = \frac{1}{2}\big(N - \max_{C\in H_m} |\langle Y', C\rangle|\big).$$

Алгоритм декодирования принятого вектора Y' следующий:

1) Умножить матрицу Адамара H_m на столбец Y' (или, что даст то же самое, умножить строку Y' на матрицу Адамара).

2) Найти максимальную по абсолютной величине компоненту полученного вектора. Пусть её номер i.

3) Если эта компонента положительна, определить кодовое слово y, ближайшее к y', как равное i-й строке двоичной матрицы Адамара A_m. В противном случае y будет дополнением к этой строке (i-й строкой $\overline{A}_m$).

После этого для восстановления исходного сообщения x по $y \in RM(1,m)$ достаточно воспользоваться уравнениями из системы $y = x \odot G$ с номерами $0, 1, 2, 4, \ldots, 2^{m-1}$:

$$y_0 = x_0, \quad y_{2^i} = x_0 \oplus x_{m-i}, \quad i = 0, 1, 2, \ldots, m-1,$$

то есть

$$
\begin{aligned}
y_0 &= x_0,\\
y_1 &= x_0 \oplus x_m,\\
y_2 &= x_0 \oplus x_{m-1},\\
y_4 &= x_0 \oplus x_{m-2}
\end{aligned}
$$

и так далее.

Складывая равенство $y_0 = x_0$ с каждым из остальных, получаем для компонент вектора x равенства

$$x_0 = y_0, \quad x_{m-i} = y_0 \oplus y_{2^i}, \quad i = 0, 1, 2, \ldots, m-1.$$

В зависимости от знака наибольшей компоненты скалярного произведения мы получим y_0, равное 0 (при $y \in \overline{A}_m$) или 1 (при $y \in A_m$). В первом случае вектор x будет подвектором y, во втором — дополнением к подвектору y.

ПРИМЕР. Декодируем по $RM(1,4)$ вектор $y' = 1001\,1001\,1001\,1110$ (код должен исправлять до 3-х ошибок).

Преобразуем y' по формуле (4):

$$Y' = (1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, 1, 1, -1).$$

Умножим Y' на матрицу Адамара

$$
H_4 = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1\\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1\\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1\\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1\\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1\\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1\\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1\\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1\\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1\\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1\\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1\\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1\\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1\\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1
\end{pmatrix}.
$$

Получим вектор $Y'H_4 = (2, 2, 2, 10, -2, -2, -2, 6, -2, -2, -2, 6, 2, 2, 2, -6)$. Его наибольшая по модулю компонента 10 — третья при нумерации с нуля. Следовательно, ближайшим кодовым словом будет третья строка двоичной матрицы Адамара (опять же при нумерации с 0), то есть вектор $y = 1001\,1001\,1001\,1001$. А исходное слово восстановится по компонентам так: $x_0 = 1\,(= y_0)$, $x_1 = 0\,(= y_8 \oplus 1)$, $x_2 = 0\,(= y_4 \oplus 1)$, $x_3 = 1\,(= y_2 \oplus 1)$, $x_4 = 1\,(= y_1 \oplus 1)$, то есть $x = 10011$.

Коды Рида–Маллера r-го порядка

Определим теперь код $RM(r, m)$ при произвольном $r \leqslant m$.

На основе матрицы $G_1(m)$ выстроим цепочку матриц $G_2(m), \dots, G_r(m)$ по следующему правилу: строки матрицы $G_i(m)$ — это всевозможные поточечные (покомпонентные, побитовые) произведения по i строк из $G_1(m)$.

Число строк матрицы $G_i(m)$ равно C_m^i. Вес каждой строки матрицы $G_i(m)$ равен 2^{m-i}.

Порождающей матрицей кода Рида–Маллера порядка r длины 2^m называется матрица $G_{r,m}$, составленная сверху вниз из строк матриц $G_0(m)$, $G_1(m)$,…,$G_r(m)$ и имеющая размер $k \times 2^m$, где $k = \sum\limits_{i=0}^{r} C_m^i$.

Таким образом, код $RM(r, m)$ классифицируется как $[2^m, k]$-код и состоит из 2^k слов. Предельный случай $r = m$ (и, значит, $k = 2^m$) включён для полноты описания и содержательным не является: в этом случае все векторы длины 2^m — кодовые слова.

Нам удобно строки матриц $G_i(m)$ и соответствующие строки матрицы $G_{r,m}$ проиндексировать следующим образом:

- Единственной строке матрицы $G_0(m)$ присвоим индекс 0. Обозначим $K_0 = \{0\}$.
- Строки матрицы $G_1(m)$ проиндексируем номерами от 1 до m. Обозначим $K_1 = \{1, 2, \dots, m\}$.
- Строкам матрицы $G_i(m)$ при $i > 1$ присвоим составные индексы из номеров строк матрицы $G_1(m)$, умножением которых она образована. Соответствующее множество из всевозможных сочетаний индексов от 1 до m по i обозначим K_i. Строки матрицы будем располагать сверху вниз по возрастанию их индексов из K_i (в лексикографическом порядке).

Приведём полностью матрицу $G_{4,4}$, отделив друг от друга горизонтальными линиями составляющие её блоки $G_r(4)$ при $r = 0, 1, 2, 3, 4$ и выделив подматрицы $G_{r,4}$. Слева в таблице укажем индексы строк.

$$\begin{array}{r} 0\\ 1\\ 2\\ 3\\ 4\\ 12\\ 13\\ 14\\ 23\\ 24\\ 34\\ 123\\ 124\\ 134\\ 234\\ 1234 \end{array}
\left[\begin{array}{cccccccccccccccc}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\ \hline
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\ \hline
0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1\\
0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1\\
0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1\\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\ \hline
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1\\ \hline
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{array}\right]
\begin{array}{llll} G_{0,4}&&&\\ &&&\\ &G_{1,4}&&\\ &&&\\ &&&\\ &&&\\ &&G_{2,4}&\\ &&&\\ &&&G_{3,4}\\ &&&\\ &&&\\ &&&\\ &&&\\ &&&\\ &&&\\ &&& \end{array}$$

Упомянем здесь, что для $RM(r,m)$ при $r > 1$, как и при $r = 1$, существует конструкция Плоткина (см. [1]), которая строит код длины 2^m из кодов длины 2^{m-1}. А именно:

$$RM(r,m) = \{u \,|\, u \oplus v\},$$

где $u \in RM(r, m-1)$, $v \in RM(r-1, m-1)$, а знак $|$ означает конкатенацию.

На основе этой конструкции индукцией по m может быть вычислено кодовое расстояние кода $RM(r,m)$, равное 2^{m-r} .

Кодирование проводится по правилу (2) при $G = G_{r,m}$.

Закодируем, например, кодом $RM(2,4)$ слово $x = 1110\,0011\,001$. (То, что оно должно иметь длину 11, видно из размера матрицы $G_{2,4}$).

Получим $y = x \odot G = 1110\;0001\,0111\,1000$.

Декодирование кодов Рида–Маллера r-го порядка по мажоритарному принципу (алгоритм Рида)

Чтобы декодировать вектор y, полученный из вектора x по правилу (2), надо решить систему уравнений (2) относительно x.

Для $G = G_{r,m}$ эта система имеет вид

$$y[N] = x[K(r)] \odot G[K(r), N],$$

где $K(r) = K_0 \cup K_1 \cup \cdots \cup K_r$ — множество индексов строк матрицы G, $N = \{0, 1, \ldots, 2^m - 1\}$ — множество индексов столбцов матрицы G.

Так как $K(r) = K(r-1) \cup K_r$, эту систему можно переписать в виде

$$y[N] = x[K(r-1)] \odot G[K(r-1), N] + x[K_r] \odot G[K_r, N]. \qquad (5)$$

При решении системы уравнений (5) вначале определяют старшие компоненты $x[K_r]$. Для определения каждого $x_i = x[i]$, $i \in K_r$, получается — как именно, будет показано ниже — набор из 2^{m-r} независимых уравнений вида $x_i = \beta_j$, где каждое β_j вычисляется как некоторая линейная функция от компонент вектора y.

Если все значения β_j, $j \in 1 : 2^{m-r}$, равны между собой, то их общее значение, естественно, и будет искомым значением x_i. Но из-за возможных ошибок вектор y может не быть кодовым словом и не удовлетворять (5), тогда среди β_j могут быть и нули, и единицы. В связи с этим решение принимается по мажоритарному принципу: если единиц среди β_j больше, чем нулей, то присваиваем $x_i := 1$, если наоборот, то $x_i := 0$. Будет обнаружено до $2^{m-r} - 1$ ошибок (сообщать об ошибке будем тогда, когда не все β_j одинаковы) и исправлено до $2^{m-r-1} - 1$ ошибок (исправления корректны, когда число искажённых символов меньше, чем неискажённых).

После того, как $x[K_r]$ определены, их значения подставляют в решаемую систему уравнений (5) и приходят к системе

$$y^{(r-1)}[N] = x[K(r-1)] \odot G[K(r-1), N],$$

где $y^{(r-1)}[N] = y[N] \oplus x[K_r] \odot G[K_r, N]$. Заметим, что $G[K_r, N] = G_r(m)$.

В этой системе тоже определяют старшие компоненты, теперь это $x[K_{r-1}]$, и вновь понижают число уравнений и неизвестных системы. Наконец, приходят к равенству $y^{(0)} = x[0] \odot G[0, N] = x_0\, G_0(m)$, которое определяет $x_0 = x[0]$ по мажоритарному принципу.

Процесс нахождения старших компонент $x[K_r]$ опишем, как это обычно делается, на примере конкретного кода.

Возьмём $RM(2, 4)$. Компоненты вектора x проиндексируем в соответствии со строками матрицы $G_{2,4}$. Компоненты вектора y, как и столбцы матрицы $G_{2,4}$, будут иметь индексы от 0 до 15.

Выпишем первые 4 уравнения решаемой системы:

$$\begin{aligned}
y_0 &= x_0, \\
y_1 &= x_0 \oplus x_4, \\
y_2 &= x_0 \oplus x_3, \\
y_3 &= x_0 \oplus x_3 \oplus x_4 \oplus x_{34}.
\end{aligned}$$

Складывая их, получаем

$$y_0 \oplus y_1 \oplus y_2 \oplus y_3 = x_{34}.$$

Таким же образом, суммируя остальные три четвёрки уравнений, имеем ещё три равенства для x_{34}:

$$y_4 \oplus y_5 \oplus y_6 \oplus y_7 = x_{34},$$
$$y_8 \oplus y_9 \oplus y_{10} \oplus y_{11} = x_{34},$$
$$y_{12} \oplus y_{13} \oplus y_{14} \oplus y_{15} = x_{34}.$$

Оказывается (см. [2]), что и всякую компоненту вектора x с индексом из K_r в точности 2^{m-r} способами можно представить в виде суммы 2^r компонент вектора y так, что каждая компонента y входит только в одну сумму.

Четвёрки соотношений для всех компонент с индексами из K_2 представлены в таблице:

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}
x_{12}	+				+				+				+			
		+				+				+				+		
			+				+				+				+	
				+				+				+				+
x_{13}	+		+						+		+					
		+		+						+		+				
					+		+						+		+	
						+		+						+		+
x_{14}	+	+							+	+						
			+	+							+					
					+	+							+	+		
							+	+							+	+
x_{23}	+		+		+		+									
		+		+		+		+								
									+		+		+		+	
										+		+		+		+
x_{24}	+	+			+	+										
			+	+			+	+								
									+	+			+	+		
											+	+			+	+
x_{34}	+	+	+	+												
					+	+	+	+								
									+	+	+	+				
													+	+	+	+

Итак, у нас теперь по 4 независимых соотношения для перечисленных в таблице переменных x_{ij}. Как говорилось выше, из-за возможных ошибок они могут давать разные значения для x_{ij}, и решение принимается по мажоритарному принципу.

ПРИМЕР. Декодируем принятое слово $y' = 1100\ 0001\,0111\,1000$, зная, что был использован код $RM(2,4)$. Заметим, что код с такими параметрами с гарантией исправит ошибку только в одном бите.

Получаем для старших компонент искомого вектора x наборы значений

- для $x_{12}: 0, 0, 1, 0;$
- для $x_{13}: 0, 1, 1, 1;$
- для $x_{14}: 1, 0, 1, 1;$
- для $x_{23}: 1, 0, 0, 0;$
- для $x_{24}: 0, 1, 1, 1;$
- для $x_{34}: 0, 1, 1, 1.$

По мажоритарному принципу принимаем $x_{12} = 0$, $x_{13} = 1$, $x_{14} = 1$, $x_{23} = 0$, $x_{24} = 0$, $x_{34} = 1$. Строим для определения остальных компонент вектор

$$y^{(1)} = y' \oplus \begin{pmatrix}0 & 1 & 1 & 0 & 0 & 1\end{pmatrix} \odot G_2(4) =$$
$$= \begin{pmatrix}0 & 1 & 1 & 0 & 0 & 1\end{pmatrix} \odot \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1\\ 0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1\\ 0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1\\ 0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \end{pmatrix}.$$

Получаем $y^{(1)} = 1101\,0000\,0000\,1111$.

Решаем систему $x \odot G = y^{(1)}$ относительно $x = (x_0, x_1, x_2, x_3, x_4)$ с матрицей $G = G_{1,4}$.

Для x_1 существует 8 уравнений:

$$x_1 = y_0^{(1)} \oplus y_8^{(1)} = y_2^{(1)} \oplus y_9^{(1)} = y_3^{(1)} \oplus y_{10}^{(1)} = y_4^{(1)} \oplus y_{11}^{(1)} =$$
$$= y_5^{(1)} \oplus y_{12}^{(1)} = y_6^{(1)} \oplus y_{13}^{(1)} = y_7^{(1)} \oplus y_{14}^{(1)} = y_8^{(1)} \oplus y_{15}^{(1)}.$$

Ниже в таблице приведены все пары соотношений для компонент вектора x с индексами из K_1.

	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}	y_{15}
	+								+							
		+								+						
			+								+					
x_1				+								+				
					+								+			
						+								+		
							+								+	
								+								+
	+				+											
		+				+										
			+				+									
x_2				+				+								
									+				+			
										+				+		
											+				+	
												+				+
	+		+													
		+		+												
					+		+									
x_3						+		+								
									+		+					
										+		+				
													+		+	
														+		+
	+	+														
			+	+												
					+	+										
x_4							+	+								
									+	+						
											+	+				
													+	+		
															+	+

Имея в виду, что в заголовке таблицы вектор y теперь понимается как $y^{(1)}$ и равен 1101 0000 0000 1111, определяем из этих соотношений наборы значений:

- для $x_1: 1, 1, 0, 1, 1, 1, 1, 1,$
- для $x_2: 1, 1, 0, 1, 1, 1, 1, 1,$
- для $x_3: 1, 0, 0, 0, 0, 0, 0, 0,$
- для $x_4: 0, 1, 0, 0, 0, 0, 0, 0.$

По мажоритарному принципу принимаем $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$. Строим для определения оставшейся компоненты вектор

$$
\begin{aligned}
&y^{(0)} = y^{(1)} \oplus \begin{pmatrix}1 & 1 & 0 & 0\end{pmatrix} \odot G_1(4) = \\
&= y^{(1)} \oplus \begin{pmatrix}1 & 1 & 0 & 0\end{pmatrix} \odot \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1
\end{pmatrix} = \\
&= 1101\,0000\,0000\,1111 \ \oplus 0000\,1111\,0000\,1111 = 1101\,1111\,1111\,1111.
\end{aligned}
$$

Имеем для x_0 шестнадцать равенств

$$x_0 = y^{(0)}[j],\ j \in 0 : 15,$$

из них 15 предлагают брать $x_0 = 1$, что мы и сделаем. Декодированный вектор:

$$x = (x_0, x_1, x_2, x_3, x_4, x_{12}, x_{13}, x_{14}, x_{23}, x_{24}) = 1110\,0011\,001.$$

Описанные алгоритмы являются простыми и наиболее известными из алгоритмов декодирования кодов Рида–Маллера, но не единственными. Получить информацию о других алгоритмах и их сложности и расширить представление о РМ-кодах можно, ознакомившись с литературой из приводимого ниже списка. Список включает классические книги [1–3], которые могут считаться учебниками по помехоустойчивым кодам, а также литературу, касающуюся

- алгоритмов декодирования кодов Рида–Маллера и некоторых подкодов: [5, 6];
- применения кодов Рида–Маллера в криптографических исследованиях: [4, 5];
- обобщения кодов Рида–Маллера на недвоичный случай: [7] (на эту тему в настоящее время имеется значительное число более поздних публикаций разных авторов).

В [5] содержится обширный список литературы по теме.

ЛИТЕРАТУРА

1. Мак-Вильямс Ф. Дж., Слоэн Н. Дж. А. *Теория кодов, исправляющих ошибки.* М.: Связь, 1979.

2. Блейхут Р. *Теория и практика кодов, контролирующих ошибки.* М.: Мир, 1986.

3. Питерсон У., Уэлдон Э. *Коды, исправляющие ошибки.* М.: Мир, 1976.

4. Логачёв О. А., Сальников А. А., Ященко В. В. *Булевы функции в теории кодирования и криптологии.* М.: МЦНМО, 2004.

5. Кузнецов Ю. В., Шкарин С. А. *Коды Рида–Маллера (обзор публикаций)* //Математические вопросы кибернетики. 1996. Вып. 6. С. 5–50.

6. Сидельников В. М. *Теория кодирования.* М.: Физматлит, 2008.

7. Берлекэмп Э. *Алгебраическая теория кодирования.* М.: Мир, 1971.