§ 1. Примеры кодов

1. Коды повторения можно построить любой длины n над любым алфавитом A. Сообщение, состоящее из 1 буквы алфавита, кодируется повторением этой буквы n раз. Декодирование производится по правилу большинства.

Основной случай здесь — алфавит $\{0,1\}$. Например, двоичный код повторения длины 9 исправляет 4 ошибки, но двоичный код длины 8 исправляет только 3 ошибки, хотя обнаруживает 4 (неполное декодирование). Ясно, что правило большинства всегда определяет победителя для двоичных кодов повторения нечетной длины.

2. Коды с проверкой на четность образуют старейшее семейство кодов, применяемых на практике. Код с проверкой на четность длины n состоит из всех двоичных (если алфавит $A = \{0,1\}$) n-ок с четным числом единиц. Любое подмножество $n-1$ координатных позиций может рассматриваться в качестве носителей информации, а оставшаяся позиция — в качестве «проверочного» бита. Появление одного ошибочного бита сразу обнаруживается, так как четность в принятом слове окажется нечетной. Невозможно определить место ошибки, но по крайней мере ее присутствие чувствуется.

Если алфавит допускает аддитивную операцию (сложение), может быть определен вариант кода проверки на четность. В таком случае код состоит из всех n-ок, суммы координат у которых равны 0.

3. $[7,4]_2$-код Хэмминга описывается в работе Шеннона. Это эффективный код, дающий полное исправление одинарных ошибок и имеющий скорость передачи $C = \frac{4}{7}$.

Пусть $x_1, \ldots, x_7$ — блок из 7 символов (0 или 1). Из них x_3, x_5, x_6, x_7 — символы сообщения, произвольно выбираемые на стороне источника. Остальные 3 символа избыточны и вычисляются исходя из условий:

x_4 выбирается так, чтобы $\alpha = x_4 + x_5 + x_6 + x_7$ было четным,

x_2 выбирается так, чтобы $\beta = x_2 + x_3 + x_6 + x_7$ было четным,

x_1 выбирается так, чтобы $\gamma = x_1 + x_3 + x_5 + x_7$ было четным.

Когда принят блок из 7 символов, вычисляются величины α, β, γ и принимаются равными 0, если четно, и 1, если нечетны. Двоичное число $\alpha\beta\gamma$ тогда дает индекс ошибочного символа x_i (если 0, то ошибки не было).

Этот код можно наглядно представить в виде диаграммы Венна. Символы сообщения занимают центр диаграммы, а каждый круг дополняется так, чтобы обеспечить наличие в нем четного числа единиц. Например, если получено сообщение, для которого в диаграмме круги A и B имеют нечетное число единиц, а C — четное, то символ в $A \cap B \cap \overline{C}$ должен считаться ошибочным.

4. Расширенный двоичный код Хэмминга получается добавлением в начало каждого кодового слова нового символа, который проверяет четность кодового слова. Добавим к $[7,4]_2$-коду Хэмминга начальный символ x_0, исходя из условий:

x_0 выбирается так, чтобы $x_0 + x_1 + \ldots + x_7$ было четным.

В результате получается расширенный $[8,4]_2$-код Хэмминга, который позволяет не только корректировать одинарные ошибки (как обычный код Хэмминга), но и обнаруживать двойные.

5. $[4,2]_3$-код Хэмминга. Это код из 9 четверок $(a,b,c,d) \in A^4$ над троичным алфавитом $A = \{0,1,2\}$. Снабдим множество A аддитивной операцией — сложением по модулю 3. Первые две координатные позиции: a, b имеют информационную функцию, так что пара $(a,b) \in A^2$ выбирается произвольно (именно поэтому будет 9 кодовых слов). Третий символ вычисляем как сумму по модулю 3 первых двух:

$$a + b = c$$

(например, если $(a,b) = (1,0)$, то $c = 1$). Последний символ должен удовлетворять равенству

$$b + c + d = 0$$

(т.е. в нашем примере четверка имеет вид $(1,0,1,2)$). Два выписанных уравнения представляют собой проверочные уравнения для троичных кодовых слов; так же, как для двоичных кодов, они могут использоваться для декодирования. Полный список кодовых слов таков:

$$0000, 0111, 0222, 1012, 1120, 1201, 2021, 2102, 2210.$$

Задача. Составить для данного кода декодирующий алгоритм, исправляющий все одинарные ошибки.

6. Обобщенный код Рида–Соломона. Мы рассмотрим код длины $n = 27$ над алфавитом, в качестве которого взято поле $\mathbb{R}$. Это — не настоящий пример, так как алфавит бесконечный.

Выберем 27 различных вещественных чисел $\alpha_1, \ldots, \alpha_{27}$. Исходным сообщением будет семерка вещественных чисел $(f_0, f_1, \ldots, f_6)$, а кодировать ее будем 27-кой $f = (f(\alpha_1), \ldots, f(\alpha_{27}))$, где $f(x) = f_0 + f_1 x + f_2 x^2 + \ldots + f_6 x^6$.

Согласно основному предположению теории кодирования, если принятые слова близки друг к другу, то они должны происходить от одного кодового слова; иначе говоря, кодовые слова должны быть удалены друг от друга как можно дальше. Кодовые слова, соответствующие f и g, будут различаться в i-й позиции, если многочлены f и g различаются в α_i, и будут равны в i-й позиции, если α_i есть корень многочлена $h(x) = f(x) - g(x)$. Но такое может произойти максимум 6 раз, так как степень h не больше 6. Следовательно: различные кодовые слова различаются самое меньшее в $21\ (= 27 - 6)$ координатных позициях. Таким образом, два кодовых слова различаются очень значительно. На самом деле данный код исправляет до 10 ошибок.