



Современная теория информации

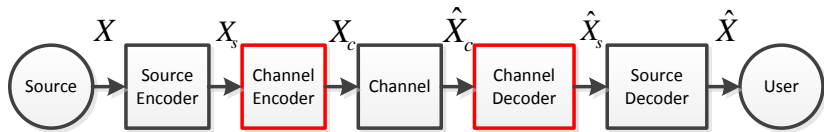
Лекция 12. Коды, исправляющие ошибки.

Содержание лекции

- 1 Конечные поля.
- 2 Линейные пространства над конечным полем.
- 3 Линейные коды.
- 4 Коды Хемминга.

Коды, исправляющие ошибки

Кодер и декодер канала

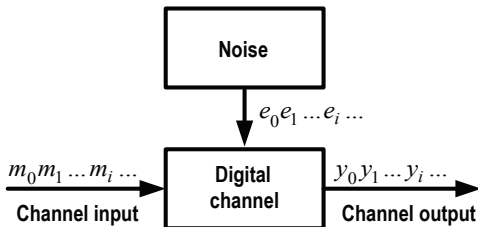


Как исправлять ошибки в канале?

Коды, исправляющие ошибки

Модель канала

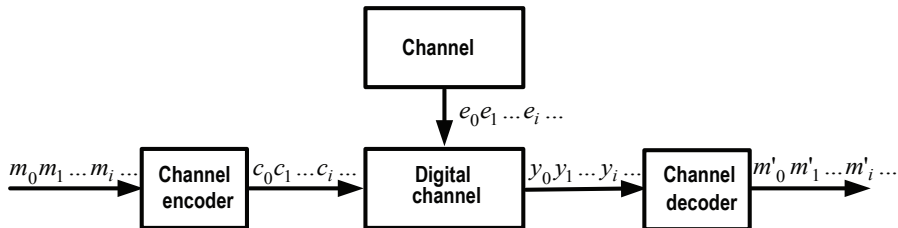
- Пусть $m_0, m_1, \dots, m_i, \dots$ – последовательность символов, которая передаётся по цифровому каналу, $m_i \in \{0, 1\}$;
- $y_0, y_1, \dots, y_i, \dots$ – символы на выходе канала, $y_i \in \{0, 1\}$;
- Ошибки канала моделируются внешним источником, который генерирует двоичный вектор ошибок $e_0, e_1, \dots, e_i, \dots$, где $e_i = 1$ – в случае ошибки, $e_i = 0$ – иначе.



Коды, исправляющие ошибки

Основная идея

- Для надёжной передачи данных используются коды, исправляющие ошибки.
- Основная идея заключается в том, что кодер канала генерирует избыточные (проверочные) символы, которые передаются вместе с информационными символами.
- Декодер канала детектирует и исправляет ошибки.



Коды, исправляющие ошибки

Конечные поля

Конечное множество X называется *конечным полем* (или полем Галуа) если имеют место следующие свойства:

- На множестве определены операции сложения и умножения, результат этих операций принадлежит множеству.
- Для любых элементов $x_i, x_j, x_k \in X$ выполняются следующие равенства:

$$(x_i + x_j) + x_k = x_i + (x_k + x_j),$$

$$(x_i \cdot x_j) \cdot x_k = x_i \cdot (x_k \cdot x_j),$$

$$x_i + x_j = x_j + x_i,$$

$$x_i \cdot x_j = x_j \cdot x_i,$$

$$(x_i + x_j) \cdot x_k = x_i \cdot x_k + x_j \cdot x_k.$$

- Существует нулевой элемент для сложения $x_0 \in X$ и единичный элемент для умножения $x_1 \in X$:

$$x_i + x_0 = x_0 + x_i = x_i,$$

$$x_i \cdot x_1 = x_1 \cdot x_i = x_i.$$

Коды, исправляющие ошибки

Конечные поля

- Для каждого элемента $x_i \in X$ существует единственный элемент $x_j \in X$, обратный для сложения

$$x_i + x_j = 0,$$

- Для каждого элемента $x_i \in X$, кроме x_0 , существует единственный элемент $x_j \in X$, обратный для умножения

$$x_i \cdot x_j = 1.$$

В литературе, конечное поле обозначается как $GF(q)$, где q – число элементов в поле.

Коды, исправляющие ошибки

Конечные поля. Пример.

Рассмотрим конечное поле $GF(5) = \{0, 1, 2, 3, 4\}$, которое включает в себя все вычеты по модулю 5. Здесь $x_0 = 0$, $x_1 = 1$.

Таблица: Таблица сложения (слева) и умножения (справа) в поле $GF(5)$

	0	1	2	3	4		0	1	2	3	4
0	0	1	2	3	4	0	0	0	0	0	0
1	1	2	3	4	0	1	0	1	2	3	4
2	2	3	4	0	1	2	0	2	4	1	3
3	3	4	0	1	2	3	0	3	1	4	2
4	4	0	1	2	3	4	0	4	3	2	1

Коды, исправляющие ошибки

Линейные пространства над конечным полем

Пусть X – конечное поле и $\mathbf{x} = (x_1, x_2, \dots, x_n) \in X^n$ – вектор, каждая компонента которого принадлежит X . Тогда сумма векторов $\mathbf{x} \in X^n$ and $\mathbf{y} \in X^n$ определяется как

$$\mathbf{x} + \mathbf{y} = (x_1 + y_1, x_2 + y_2, \dots, x_n + y_n). \quad (1)$$

Если $c \in X$, тогда умножение вектора \mathbf{x} на скаляр c определяется как

$$c \cdot \mathbf{x} = (c \cdot x_1, c \cdot x_2, \dots, c \cdot x_n). \quad (2)$$

Пример. Рассмотрим два вектора $\mathbf{x} = (1, 2, 3, 4)$ и $\mathbf{y} = (4, 2, 3, 1)$, каждый элемент которого принадлежит $GF(5)$. Тогда $\mathbf{x} + \mathbf{y} = (0, 4, 1, 0)$. Пусть $c = 2$, тогда $c \cdot \mathbf{x} = (2, 4, 1, 3)$.

Коды, исправляющие ошибки

Линейные пространства над конечным полем

Множество векторов \mathbf{V} формируют *линейное пространство*, если это множество является замкнутым по отношению к операциям сложения и умножения на скаляр. Это означает, что для любого $k = \{1, 2, 3, \dots\}$, вектор

$$\mathbf{z} = \sum_{i=1}^k c_i \cdot \mathbf{x}_i \quad (3)$$

принадлежит \mathbf{V} для любого $c_i \in X$ и $\mathbf{x}_i \in X^n$. Правая часть (3) называется *линейной комбинацией* векторов $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$.

Пример. Рассмотрим два множества векторов V_1 и V_2 над полем $GF(2) = \{0, 1\}$.

$$\mathbf{v}_1 = \left\{ \begin{array}{c} 000 \\ 110 \\ 011 \\ 101 \end{array} \right\}, \mathbf{v}_2 = \left\{ \begin{array}{c} 000 \\ 100 \\ 010 \\ 001 \end{array} \right\}.$$

Множество \mathbf{V}_1 является линейным пространством, потому что сумма любой пары векторов из \mathbf{V}_1 принадлежит \mathbf{V}_1 . Множество \mathbf{V}_2 не является линейным пространством.

Коды, исправляющие ошибки

Линейные пространства над конечным полем

Подмножество линейного пространства, для которого выполняются все свойства линейного пространства, называется *линейным подпространством*.

Пример. Рассмотрим два множества векторов V_1 и V_2 в поле $GF(2) = \{0, 1\}$.

$$\mathbf{V}_1 = \left\{ \begin{pmatrix} 000 \\ 001 \\ 010 \\ 100 \\ 101 \\ 110 \\ 111 \end{pmatrix} \right\}, \mathbf{V}_2 = \left\{ \begin{pmatrix} 000 \\ 110 \\ 011 \\ 101 \end{pmatrix} \right\}.$$

Множество \mathbf{V}_2 является линейным подпространством пространства \mathbf{V}_1 .

Коды, исправляющие ошибки

Линейные пространства над конечным полем

Векторы $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ называются *линейно независимыми*, если равенство

$$c_1 \cdot \mathbf{x}_1 + c_2 \cdot \mathbf{x}_2 + \dots + c_k \cdot \mathbf{x}_k = \mathbf{0}, \quad (4)$$

где $\mathbf{0}$ – нулевой вектор, выполняется, если

$$c_1 = c_2 = \dots = c_k = 0. \quad (5)$$

Пример. Векторы $(0, 1, 1, 1)$ и $(0, 2, 2, 2)$ над полем $GF(5)$ являются линейно зависимыми, поскольку $2 \cdot (0, 2, 2, 2) + (0, 1, 1, 1) = (0, 0, 0, 0)$.

В каждом линейном пространстве существуют линейно независимые векторы $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$, такие что каждый вектор

$$\mathbf{x}_i = c_1 \cdot \mathbf{x}_1 + c_2 \cdot \mathbf{x}_2 + c_k \cdot \mathbf{x}_k. \quad (6)$$

Такие векторы называются *базисными векторами* линейного пространства.

Коды, исправляющие ошибки

Линейные пространства над конечным полем

Пример. Рассмотрим два множества векторов \mathbf{V}_1 и \mathbf{V}_2 над полем $GF(2) = \{0, 1\}$.

$$\mathbf{V}_1 = \left\{ \begin{array}{c} 000 \\ 001 \\ 010 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \right\}, \mathbf{V}_2 = \left\{ \begin{array}{c} 100 \\ 010 \\ 001 \end{array} \right\}.$$

Векторы, принадлежащие \mathbf{V}_2 являются базисными векторами для линейного пространства \mathbf{V}_1 .

Коды, исправляющие ошибки

Порождающая матрица линейного пространства

Пусть векторы $\mathbf{g}_1 = (g_{11}, \dots, g_{1n})$, $\mathbf{g}_2 = (g_{21}, \dots, g_{2n})$, ..., $\mathbf{g}_k = (g_{k1}, \dots, g_{kn})$ являются базисными векторами для линейного пространства \mathbf{V}_k . Тогда каждый вектор \mathbf{x} в \mathbf{V}_k может быть представлен как линейная комбинация базисных векторов:

$$\mathbf{x} = m_1 \cdot \mathbf{g}_1 + m_2 \cdot \mathbf{g}_2 + \dots + m_k \cdot \mathbf{g}_k. \quad (7)$$

Выражение (7) может быть записано как:

$$\mathbf{x} = (m_1, m_2, \dots, m_k) \cdot \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1n} \\ g_{21} & g_{22} & \dots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \dots & g_{kn} \end{bmatrix} = \mathbf{m} \cdot \mathbf{G}, \quad (8)$$

где $\mathbf{m} = (m_1, m_2, \dots, m_k)$ и $k \times n$ матрица \mathbf{G} имеет в качестве своих строк базисные векторы линейного пространства \mathbf{V}_k . Matrix \mathbf{G} называется *порождающей матрицей* линейного пространства \mathbf{V}_k .

Коды, исправляющие ошибки

Проверочная матрица линейного пространства

Произвольный вектор $\mathbf{x} = (x_1, \dots, x_n) \in \mathbf{V}_k$ удовлетворяет следующей системе линейных уравнений:

$$\begin{cases} h_{11} \cdot x_1 + \dots + h_{1n} \cdot x_n = 0, \\ h_{21} \cdot x_1 + \dots + h_{2n} \cdot x_n = 0, \\ \dots\dots\dots \\ h_{r1} \cdot x_1 + \dots + h_{rn} \cdot x_n = 0, \end{cases} \quad (9)$$

где $r = n - k$, или в матричной записи

$$\mathbf{x} \cdot \mathbf{H}^T = \mathbf{0}. \quad (10)$$

Система уравнений (10) проверяет, что вектор \mathbf{x} принадлежит линейному пространству \mathbf{V}_k . Поэтому \mathbf{H} называется *проверочной матрицей* линейного пространства \mathbf{V}_k .

$$\mathbf{x} \cdot \mathbf{H}^T = \mathbf{m} \cdot \mathbf{G} \cdot \mathbf{H}^T = \mathbf{0}. \quad (11)$$

$$\mathbf{G} \cdot \mathbf{H}^T = \mathbf{0}. \quad (12)$$

Коды, исправляющие ошибки

Линейные коды


Линейным q -ичным кодом длины n с k информационных символов, или (n, k) -кодом над полем $GF(q)$, называется k -мерное подпространство линейного n -мерного пространства всех векторов над полем $GF(q)$.

Линейный (n, k) -код задаётся базисными векторами $\mathbf{g}_1 = (g_{11}, \dots, g_{1n})$, $\mathbf{g}_2 = (g_{21}, \dots, g_{2n})$, ..., $\mathbf{g}_k = (g_{k1}, \dots, g_{kn})$, $g_{ij} \in GF(q)$ или порождающей матрицей

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} & \cdot & g_{1n} \\ g_{21} & g_{22} & \cdot & g_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ g_{k1} & g_{k2} & \cdot & g_{kn} \end{bmatrix}, \quad (13)$$

при этом кодовое слово $\mathbf{c} = (c_1, c_2, \dots, c_n)$ является линейной комбинацией базисных векторов:

$$\mathbf{c} = m_1 \cdot \mathbf{g}_1 + \dots + m_k \cdot \mathbf{g}_k = \mathbf{m} \cdot \mathbf{G}, \quad (14)$$

где $\mathbf{m} = (m_1, m_2, \dots, m_k)$ – информационная последовательность. 

Коды, исправляющие ошибки

Линейные коды

Если порождающая матрицы \mathbf{G} может быть представлена как

$$\mathbf{G} = [\mathbf{I}_k | \mathbf{G}_2], \quad (15)$$

где \mathbf{I}_k - единичная матрица, размером $k \times k$, то эта матрица имеет *левую каноническую форму*.

Если порождающая матрица имеет каноническую форму, то линейный код называется *систематическим*. Для систематического кода, кодовое слово представляется как

$$\mathbf{c} = \mathbf{m} \cdot \mathbf{G} = \mathbf{m} \cdot [\mathbf{I}_k | \mathbf{G}_2] = (\mathbf{m}, \mathbf{m} \cdot \mathbf{G}_2), \quad (16)$$

то есть, кодовое слово состоит из двух подслов: левое подслово это информационная последовательность $\mathbf{m} = (m_1, m_2, \dots, m_k)$ длины k , а правое подслово состоит из $r = n - k$ проверочных символов.

Коды, исправляющие ошибки

Линейные коды

Пример. Пусть порождающая матрица с длиной кодового слова $n = 7$ и длиной информационной последовательности $k = 4$ имеет вид:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Необходимо сформировать кодовое слово \mathbf{c} для информационной последовательности $\mathbf{m} = (0101)$.

$$\mathbf{c} = \mathbf{m} \cdot \mathbf{G} = (0101) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = (0101010).$$

Коды, исправляющие ошибки

Свойства линейных кодов

Пусть x и y – два слова из X^n . Расстоянием Хемминга $d(x, y)$ между x и y называется число позиций, в которых эти слова различаются.

Рассмотрим $c = (c_1, \dots, c_M)$ как кодовое слово длины n над алфавитом X . Минимальное расстояние d кода C это минимальное расстояние Хемминга между любыми парами кодовых слов из C .

Теорема 1. Минимальное расстояние линейного кода C равно минимуму из весов ненулевых кодовых слов.

Theorem 2. Если любые $d - 1$ столбцов проверочной матрицы H линейного кода линейно независимы, то минимальное расстояние кода будет по меньшей мере d . Если при этом найдутся d линейно зависимых столбцов, то минимальное расстояние кода равно d .

Теорема 3. Код с минимальным расстоянием d исправляет любые ошибки кратности $t = (d - 1)/2$ и обнаруживает ошибки кратности $t \leq d - 1$.

Коды, исправляющие ошибки

Коды Хемминга

Коды Хемминга это линейные коды, которые обеспечивают минимально возможное количество проверочных символов для минимального кодового расстояния $d = 3$. Рассмотрим метод построения порождающей и проверочной матрицы для систематического кода Хемминга над полем $GF(2)$.

Пусть порождающая матрица кода \mathbf{G} представляется в левой канонической форме:

$$\mathbf{G} = [\mathbf{I}_k | \mathbf{G}_2], \quad (17)$$

где \mathbf{I}_k – единичная подматрица размером $k \times k$ и \mathbf{G}_2 – подматрица размером $r \times k$.

Проверочная матрица \mathbf{H} может быть записана как:

$$\mathbf{H} = [\mathbf{H}_1 | \mathbf{H}_2], \quad (18)$$

где \mathbf{H}_1 – подматрица размером $r \times k$, \mathbf{H}_2 – подматрица размером $r \times r$.

$$\mathbf{0} = \mathbf{G} \cdot \mathbf{H}^T = [\mathbf{I}_k | \mathbf{G}_2] \begin{bmatrix} \mathbf{H}_1^T \\ \mathbf{H}_2^T \end{bmatrix} = \mathbf{H}_1^T + \mathbf{G}_2 \cdot \mathbf{H}_2^T. \quad (19)$$

Коды, исправляющие ошибки

Коды Хемминга

Пусть проверочная матрица \mathbf{H} представлена в правой канонической форме, то есть \mathbf{H}_2^T – единичная подматрица. Тогда,

$$\mathbf{0} = \mathbf{H}_1^T + \mathbf{G}_2 \cdot \mathbf{H}_2^T = \mathbf{H}_1^T + \mathbf{G}_2. \quad (20)$$

С учетом работы в поле $GF(2)$, из (20) следует:

$$\mathbf{G}_2 = \mathbf{H}_1^T. \quad (21)$$

- Матрица \mathbf{H} не должна содержать нулевых столбцов.
- Из Теоремы 2 следует, что для построения линейного кода с расстоянием $d = 3$ любые два столбца \mathbf{H} должны быть линейно независимы. В случае $GF(2)$ это означает, что любые два столбца \mathbf{H} должны быть различными (это возможно, когда $n = k + r \leq 2^r - 1$). При этом, матрица \mathbf{H} должна быть записана в правой канонической форме, то есть содержать единичную подматрицу в правой части.

Коды, исправляющие ошибки

Построение кода Хемминга

Input.

Длина кодового слова n .

Длина информационной последовательности k .

Step 1.

$$r := n - k.$$

Сформировать \mathbf{H}_1 размером $r \times k$ из k различных столбцов, каждый из которых содержит больше одной единицы.

Step 2.

Сформировать \mathbf{H}_2 как единичную матрицу $r \times r$.

$$\mathbf{H} := [\mathbf{H}_1 | \mathbf{H}_2].$$

Step 3.

Сформировать \mathbf{G}_1 как единичную матрицу $k \times k$.

$$\mathbf{G} := [\mathbf{G}_1 | \mathbf{H}_1^T].$$

Коды, исправляющие ошибки

Построение кода Хемминга

Пример. $n = 7$, $k = 4$. Скорость кода $R = \frac{k}{n} = \frac{4}{7}$.

$$\mathbf{H}_1 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix},$$

$$\mathbf{H} = [\mathbf{H}_1 | \mathbf{H}_2] = \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right]$$

$$\mathbf{G} = [\mathbf{G}_1 | \mathbf{H}_1^T] = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} \right].$$

Коды, исправляющие ошибки

Расширенные коды Хемминга

- Расширенные коды Хемминга обеспечивают минимально возможное количество проверочных символов для минимального кодового расстояния $d = 4$.
- Увеличение минимального расстояния кода достигается путем замены одного информационного символа исходного кода Хемминга на проверочный символ.
- Поэтому проверочная матрица такого кода должна иметь на одну строку больше, чем в коде Хемминга.

Коды, исправляющие ошибки

Расширенные коды Хемминга

- В соответствии с теоремой 2 в проверочной матрице расширенного кода Хемминга любые три столбца должны быть линейно независимые.
- В случае кода над полем $GF(2)$ это означает, что
 - ▶ любые два столбца матрицы **H** должны быть различны;
 - ▶ матрица **H** не должна содержать нулевого столбца;
 - ▶ любые два столбца в сумме не должны давать третий.

Коды, исправляющие ошибки

Расширенные коды Хемминга

Обозначим через \mathbf{H}^3 проверочную матрицу кода Хемминга с минимальным расстоянием $d = 3$. Тогда матрица

$$\mathbf{H}^4 = \left[\begin{array}{c|c} 1 & 1 & 1 & \dots & 1 \\ \hline & & & \mathbf{H}^3 & \end{array} \right] \quad (22)$$

удовлетворяет приведенным выше требованиям для кода с минимальным расстоянием $d = 4$.

- Во-первых, матрица \mathbf{H}^4 не содержит двух одинаковых столбцов и не содержит нулевой столбец, поскольку в матрице \mathbf{H}^3 нет нулевых столбцов и все столбцы разные.
- Во-вторых, сумма по модулю 2 любых двух столбцов не равна никакому третьему столбцу, так как сумма двух столбцов будет содержать 0 в первом разряде, тогда как все столбцы в \mathbf{H}^4 в первом разряде равны 1.

Коды, исправляющие ошибки

Расширенные коды Хемминга

- Для формирования порождающей матрицы расширенного кода Хемминга проверочную матрицу H^4 необходимо представить в правой канонической форме.
- Так как строки матрицы H^4 являются базисным векторами линейного пространства, то каждый базисный вектор можно заменить другим базисным вектором, который является линейной комбинацией исходных базисных векторов.
- В случае кодов над полем $GF(2)$ можно менять любую строку матрицы H^4 на сумму двух или более строк по модулю 2 до тех пор, пока матрица H^4 не примет правую каноническую форму.
- Затем, аналогично коду Хемминга можно построить порождающую матрицу.

Коды, исправляющие ошибки

Расширенные коды Хемминга. Пример построения.

Требуется построить проверочную и порождающую матрицу расширенного кода Хемминга с длиной кодового слова $n = 7$ и информационной последовательностью длины $k = 3$. В соответствии с шагом 1 алгоритма формируется подматрица

$$\mathbf{H}_1^3 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix},$$

затем в соответствии с шагом 2 формируется единичная матрица \mathbf{H}_2^3 размером 3×3 , и формируется проверочная матрица в виде

$$\mathbf{H}^3 = [\mathbf{H}_1^3 | \mathbf{H}_2^3] = \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right].$$

Коды, исправляющие ошибки

Расширенные коды Хемминга. Пример построения.

В соответствии с шагом 3 формируется проверочная матрица расширенного кода как

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Затем сумма всех строк \mathbf{H}_4 по модулю 2 записывается в первую строку:

$$\mathbf{H}_4 = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Коды, исправляющие ошибки

Расширенные коды Хемминга. Пример построения.

Для формирования правой канонической формы первая строка матрицы \mathbf{H}_4 складывается со второй, результат сложения записывается во вторую строку. Затем первая строка складывается с третьей, результат сложения записывается в третью строку. В итоге матрица \mathbf{H}_4 представляется как

$$\mathbf{H}_4 = [\mathbf{H}_1^4 | \mathbf{H}_2^4] = \left[\begin{array}{ccc|cccc} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array} \right].$$

В соответствии с шагом 4 формируется единичная подматрица \mathbf{G}_1^4 размером 3×3 и порождающая матрица в виде

$$\mathbf{G} = [\mathbf{G}_1^4 | (\mathbf{H}_1^4)^T] = \left[\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array} \right].$$

Коды, исправляющие ошибки

Декодирование кода Хемминга по минимуму расстояния

Пример. Пусть сообщение $\mathbf{m} = (101)$ передаётся при помощи расширенного кода Хемминга, имеющего порождающую матрицу

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Кодер канала формирует кодовое слово \mathbf{c} как

$$\mathbf{c} = \mathbf{m} \cdot \mathbf{G} = (101) \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} = (1011010).$$

Предположим, что произошла одна ошибка, то есть вектор ошибок $\mathbf{e} = (0010000)$. Тогда декодер принял сообщение $\mathbf{y} = \mathbf{c} + \mathbf{e} = (1001010)$.

Коды, исправляющие ошибки

Декодирование кода Хемминга по минимуму расстояния

Декодер попарно вычисляет расстояние $d(y, c_i)$ между $y = (1001010)$ и словами из C и строит таблицу:

i	m_i	$c_i = m_i \cdot G$	$d(y, c_i)$
0	000	0000000	2
1	001	0010111	4
2	010	0101011	4
3	011	0111100	6
4	100	1001101	6
5	101	1011010	1
6	110	1100110	3
7	111	1110001	5

Так как минимальное расстояние $d(y, c_5) = 1$ соответствует кодовому слову c_5 , то декодер принимает решение, что передавалось кодовое слово $c = (1011010)$ и исходное сообщение $m = (101)$.

Коды, исправляющие ошибки

Синдромное декодирование кода Хемминга

Из свойств проверочной матрицы следует, что $\mathbf{c} \cdot \mathbf{H}^T = 0$. Поэтому

$$\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T = (\mathbf{c} + \mathbf{e}) \cdot \mathbf{H}^T = \mathbf{c} \cdot \mathbf{H}^T + \mathbf{e} \cdot \mathbf{H}^T = \mathbf{e} \cdot \mathbf{H}^T, \quad (23)$$

называемый *синдромом* зависит только от вектора ошибки \mathbf{e} и не зависит от кодового слова \mathbf{c} . Если ошибка не произошла, то $\mathbf{s} = \mathbf{0}$.

i	\mathbf{e}_i	$\mathbf{s}_i = \mathbf{e}_i \cdot \mathbf{H}^T$
0	1000000	011
1	0100000	101
2	0010000	110
3	0001000	111
4	0000100	100
5	0000010	010
6	0000001	001

Коды, исправляющие ошибки

Работа декодера в режиме обнаружения ошибок

Пусть для передачи сообщения $\mathbf{m} = (101)$ использовался расширенный код Хемминга и было сформировано кодовое слово $\mathbf{c} = (1011010)$. Пусть в процессе передачи возникли две ошибки и вектор ошибок $\mathbf{e} = (0010001)$. Тогда на вход декодера канала поступает сообщение $\mathbf{y} = \mathbf{c} + \mathbf{e} = (1001011)$.

Декодер канала вычисляет синдром

$$\mathbf{s}^* = \mathbf{y} \cdot \mathbf{H}^T = (1001011) \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = (0110).$$

Так как синдром \mathbf{s}^* отличен от нулевого, то декодер канала выносит решение о том, что в канале произошла ошибка.

Коды, исправляющие ошибки

Работа декодера в режиме обнаружения ошибок. Число ошибок превышает $d - 1$

Пусть для передачи сообщения $\mathbf{m} = (101)$ было сформировано кодовое слово $\mathbf{s} = (1011010)$. Пусть в процессе передачи возникли четыре ошибки и вектор ошибок $\mathbf{e} = (1110001)$. Тогда на вход декодера канала поступает сообщение $\mathbf{y} = \mathbf{s} + \mathbf{e} = (0101011)$. Декодер канала вычисляет синдром

$$\mathbf{s}^* = \mathbf{y} \cdot \mathbf{H}^T = (0101011) \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = (0000).$$

Так как синдром \mathbf{s}^* является нулевым, то декодер канала выносит ошибочное решение о том, что в канале не было ошибок.