Лекция №5
по курсу
«Методы и средства передачи информации ч.1»

Лектор: д.т.н., Оцоков Шамиль Алиевич,

email: otsokovShA@mpei.ru

Москва, 2022

Расстояние Хэмминга

Известно, что расстояние между точками в пространстве определяется как длина отрезка прямой, соединяющей эти точки. Оно служит мерой близости точек — чем меньше расстояние, тем ближе друг к другу расположены точки. Если обозначать расстояние между точками a и b через $\rho(a, b)$, то для любых точек a, b и c имеем:

1) $\rho(a,b) \geqslant 0$;
2) $\rho(a,b)=0$ означает, что $a=b$;
3) $\rho(a,b)=\rho(b, a)$
4) $\rho(a,b)+\rho(b,c) \geqslant \rho(a,c)$.

Расстоянием $\rho(x, y)$ между двумя словами x и y назовем число несовпадающих позиций этих слов. Например, расстояние между словами $x=01101$ и $y=00111$ равно 2.

Это расстояние называется расстоянием Хемминга. Почему оно расстояние?

Кодовое расстояние

$$\rho(a, b) = \sum_{i=1}^{n} (a_i + b_i) \mod 2$$

$$\rho(b, c) = \sum_{i=1}^{n} (b_i + c_i) \mod 2$$

$$\rho(a, c) = \sum_{i=1}^{n} (a_i + c_i) \mod 2$$

$$\rho(a, b) + \rho(b, c) = \rho(a, c) \geq \rho(a, c)$$

Кодовое расстояние $d(V)$ определим как минимальное расстояние между различными кодовыми словами из V:

$$d(V) = \min_{x \neq y} \rho(x, y).$$

Кодовое расстояние

Какое кодовое расстояние кода проверки на чётность, кода с повторением?

Кодовое расстояние определяет корректирующие возможности кодов.

Справедлива теорема:

Код способен исправлять любые комбинации из t (и меньшего числа) ошибок тогда и только тогда, когда его кодовое расстояние больше $2t$.

В самом деле, если $d(V)>2t$, то для любых кодовых слов x, y имеем $\rho(x, y)\geqslant 2t+1$. Пусть при передаче некоторого слова x произошло $r\leqslant t$ ошибок, и в результате было принято слово z. Тогда $\rho(x, z)=r\leqslant t$, и в то же время расстояние $\rho(z, y)$ до любого другого кодового слова y больше t. Последнее вытекает из неравенства треугольника:

$$\rho(x, z)+\rho(z, y)\geqslant\rho(x, y)\geqslant 2t+1.$$

Кодовое расстояние

Значит, для восстановления посланного слова (декодирования) необходимо найти кодовое слово x, «ближайшее» к принятому слову z в смысле расстояния Хемминга. Подчеркнем, что это прабило декодирования приводит к правильному результату, если число ошибок в передаваемом слове действительно не превосходило t.

Если же условие $d(V)>2t$ нарушается, то найдутся такие кодовые слова x и y, расстояние между которыми $\rho(x, y)\leqslant 2t$. Тогда может найтись такая комбинация из t ошибок в одном из слов x, что принятое слово z будет находиться от другого слова y не дальше, чем от x. Поэтому нельзя будет определить, какое из слов — x или y — было на самом деле передано

Кодовое расстояние

2. Доказать, что для обнаружения s (или меньшего числа) ошибок необходимо и достаточно, чтобы кодовое расстояние удовлетворяло неравенству $d(V) \geqslant s+1$.

3. Доказать, что для исправления t (и меньшего числа) ошибок и вместе с этим обнаружения s (и меньшего числа) ошибок ($s \geqslant t$) необходимо и достаточно, чтобы кодовое расстояние удовлетворяло неравенству $d(V) \geqslant t+s+1$.

Указание. Проверяем непосредственно неравенство.

Кодовое расстояние

ЛИНЕЙНЫЕ ИЛИ ГРУППОВЫЕ КОДЫ

Большинство рассмотренных выше кодов обладало следующим свойством: сумма (и разность) двух кодовых слов также являлась кодовым словом. Для кода с повторением это свойство очевидно. Ясно оно и для кода с общей проверкой на четность, потому что сумма двух слов с четным числом единиц есть также слово с четным числом единиц.

w(c) – вес кодового слова c (количество ненулевых битов в кодовом слове)

Теорема 3.1.3. *Для линейного кода минимальное расстояние d^* находится из равенства*

$$d^* = \min_{\mathbf{c} \neq 0} w(\mathbf{c}) = w^*,$$

где минимум берется по всем кодовым словам, кроме нулевого.

Доказательство.

$$d^* = \min_{\substack{\mathbf{c}_i, \mathbf{c}_j \in \mathscr{C} \\ i \neq j}} d(\mathbf{c}_i, \mathbf{c}_j) = \min_{\substack{\mathbf{c}_i, \mathbf{c}_j \in \mathscr{C} \\ i \neq j}} d(0, \mathbf{c}_i - \mathbf{c}_j) = \min_{\substack{\mathbf{c} \in \mathscr{C} \\ \mathbf{c} \neq 0}} w(\mathbf{c}).$$

□

Линейные коды

Каждое слово $x_1x_2\ldots x_n$ этого алфавита будем отождествлять с вектором $(x_1, x_2, \ldots, x_n)$ n-мерного пространства L_n (в котором координаты векторов являются элементами F). Вектор, соответствующий кодовому слову, будем называть *кодовым*. Систему проверочных соотношений запишем в виде системы уравнений:

$$\begin{aligned} b_{11}x_1+b_{12}x_2+\ldots+b_{1n}x_n&=0,\\ b_{21}x_1+b_{22}x_2+\ldots+b_{2n}x_n&=0,\\ \ldots\quad\ldots\quad\ldots&\\ b_{m1}x_1+b_{m2}x_2+\ldots+b_{mn}x_n&=0 \end{aligned} \qquad (1)$$

с коэффициентами b_{ij} из F. Код, состоящий из всех слов $x_1x_2\ldots x_n$, для которых справедливы соотношения (1), называют *кодом с проверками на четность*.

Линейные коды

Для такого кода выполняется следующее свойство: если векторы $(a_1, a_2, \ldots, a_n)$ и $(b_1, b_2, \ldots, b_n)$ являются кодовыми, а значит, решениями системы (1), то и их сумма $(a_1+b_1, a_2+b_2, \ldots, a_n+b_n)$ также является решением этой системы и потому кодовым вектором. Справедливо и другое свойство решений системы (1): если α — элемент поля F и $(a_1, a_2, \ldots, a_n)$ — решение системы (1), то и вектор $(\alpha a_1, \alpha a_2, \ldots, \alpha a_n)$ также является решением системы (1).

Линейные коды

Оба отмеченных свойства проверяются непосредственной подстановкой в систему (1) векторов

$$(a_1+b_1,\ a_2+b_2,\ \ldots,\ a_n+b_n) \text{ и } (\alpha a_1,\ \alpha a_2,\ \ldots,\ \alpha a_n).$$

Вместе эти свойства означают, что код с проверками на четность образует линейное подпространство в пространстве L_n всех n-буквенных слов. По этой причине коды с проверками на четность называют *линейными кодами* (двоичные линейные коды называют также *групповыми*). Если кодовое подпространство в пространстве L_n имеет размерность k, то употребляют для большей определенности термин линейный (n, k)-код.

Проверочная матрица

Имеется очень много причин, по которым линейные коды являются важнейшими в теории кодирования. Одна из них связана с удобствами в обнаружении и исправлении ошибок, Другая
причина — это возможность компактного задания кода. Действительно, в случае линейного кода нет необходимости указывать полный список кодовых слов, ведь код вполне определен системой линейных уравнений (1) или матрицей этой системы (*проверочной матрицей*):

$$H = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{pmatrix}.$$

Проверочная матрица

В дальнейшем мы будем предполагать, что строки этой матрицы линейно независимы.

К числу других достоинств линейных кодов, которые связаны с предыдущими, относятся простые алгоритмы кодирования и декодирования, легко реализуемые электронными переключательными схемами. Вообще, можно сказать, что бурное развитие теории кодирования, которое происходило в последние десятилетия, объясняется главным образом тем, что к линейным кодам приложим хорошо развитый аппарат линейной алгебры и теории конечных полей.

Возвращаясь к рассмотренным ранее кодам, легко найдем их проверочные матрицы. Так, для кода с общей проверкой на четность имеем одно проверочное соотношение

Проверочная матрица

$x_1+x_2+\ldots+x_n=0$; соответственно этому проверочная матрица состоит из одной строки и имеет вид

$$H=(111\ldots1).$$

x1 + x3 + x5 + x7 = 0 (mod 2)
x2 + x3 + x6 + x7 = 0 (mod 2)
x4 + x5 + x6 + x7 = 0 (mod 2)

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

Итак, мы имеем три проверочных соотношения:

$$\begin{aligned}s_1&=\alpha_4+\alpha_5+\alpha_6+\alpha_7=0,\\ s_2&=\alpha_2+\alpha_3+\alpha_6+\alpha_7=0,\\ s_3&=\alpha_1+\alpha_3+\alpha_5+\alpha_7=0,\end{aligned}\qquad(5)$$

Линейные коды

Имеется и другой способ матричного задания линейного кода. Он основан на том, что во всяком подпространстве линейного пространства можно выбрать базис, т. е. такую линейно независимую систему векторов, через которые линейно выражаются все вообще векторы подпространства. Пусть

$$\begin{aligned} a_1 &= (a_{11}, a_{12}, \quad , a_{1n}), \\ a_2 &= (a_{21}, a_{22}, \quad , a_{2n}), \\ &\cdot \qquad\qquad \cdot \\ a_k &= (a_{k1}, a_{k2}, \quad , a_{kn}) \end{aligned} \tag{3}$$

— базисные векторы линейного кода. Тогда всевозможные кодовые векторы исчерпываются линейными комбинациями

$$\alpha_1 a_1 + \alpha_2 a_2 + \ldots + \alpha_k a_k, \tag{4}$$

где коэффициенты α_i — любые элементы исходного поля.

Порождающая матрица

Таким образом, система базисных векторов (3) полностью определяет линейный (n, k)-код. Матрица G, составленная из них,

$$G=\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{k1} & a_{k2} & \dots & a_{kn} \end{pmatrix}. \qquad (5)$$

называется *порождающей матрицей* кода.

При использовании линейного кода для передачи сообщений полезно знать и порождающую, и проверочную матрицу. С помощью порождающей матрицы удобно кодировать сообщения. Поскольку линейный (n, k)-код с порождающей матрицей (5) имеет q^k кодовых слов (любой из k коэффициентов в (4) можно выбрать q способами), он позволяет закодировать все сообщения длины k в алфавите из q символов. Если $A=\alpha_1\alpha_2 \dots \alpha_k$ — такое сообщение (α_i — информационные символы), то самое удобное — сопоставить ему кодовый вектор a, совпадающий с линейной комбинацией (4) строк порождающей матрицы. Вектор a нетрудно записать в матричном виде, используя правило умножения матриц:

Порождающая матрица

$$a = A \cdot G = (\alpha_1 \alpha_2 \ldots \alpha_k) \begin{pmatrix} a_{11} & a_{12} & a_{1n} \\ a_{21} & a_{22} & a_{2n} \\ \cdot & \cdot & \cdot \\ a_{k1} & a_{k2} & a_{kn} \end{pmatrix}.$$

Пример. Пусть дана порождающая матрица двоичного линейного кода:

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

Этот код содержит $2^4 = 16$ кодовых слов, которыми можно закодировать все двоичные сообщения длины 4. Если, например, $A = (0101)$, то для соответствующего кодового вектора имеем:

Порождающая матрица

$$a=(0\ 1\ 0\ 1)\begin{pmatrix}1&1&1&0&0&0&0\\1&0&0&1&1&0&0\\0&1&0&1&0&1&0\\1&1&0&1&0&0&1\end{pmatrix}=(0\ 1\ 0\ 0\ 1\ 0\ 1).$$

О роли проверочной матрицы мы скажем в дальнейшем — она используется в основном при декодировании полученных сообщений и для исправления ошибок.

Выясним, как связаны порождающая и проверочная матрицы, и как по одной из них найти другую. Обратимся

Выясним, как связаны порождающая и проверочная матрицы, и как по одной из них найти другую. Обратимся к соотношениям (1). Всякая строка порождающей матрицы, являясь кодовым вектором, удовлетворяет каждому из соотношений (1), т. е. для любых i и j

$$b_{i1}a_{j1}+b_{i2}a_{j2}+\ldots+b_{in}a_{jn}=0. \qquad (7)$$

Другими словами, любая строка порождающей и любая строка проверочной матриц ортогональны друг другу. Матричная запись равенств (7) выглядит так:

$$GH^{T}=0$$

Порождающая матрица

Заметим также, что из соотношений (1) вытекает равенство

$$vH^T=0,$$

справедливое для каждого кодового вектора v.

Для отыскания порождающей матрицы нужно фактически найти $k=n-m$ линейно независимых решений системы (1). Эти решения как раз и будут строками порождающей матрицы.

$$\begin{aligned} &b_{11}x_1+b_{12}x_2+\ldots+b_{1n}x_n=0,\\ &b_{21}x_1+b_{22}x_2+\ldots+b_{2n}x_n=0, \qquad (1)\\ &\ldots\ldots\ldots\\ &b_{m1}x_1+b_{m2}x_2+\ldots+b_{mn}x_n=0 \end{aligned}$$

Порождающая матрица

Заметим, что базис можно выбрать не единственным образом, поэтому порождающая матрица G определена неоднозначно. Последнее, впрочем, верно и в отношении проверочной матрицы H.

Легко указать порождающую матрицу кода с повторением; она имеет вид:

$$G=(1\ 1\ 1\ldots1).$$

является порождающей матрицей этого кода.

При использовании линейного кода для передачи сообщений полезно знать и порождающую, и проверочную матрицу. С помощью порождающей матрицы удобно кодировать сообщения.

Порождающая матрица

В качестве базисных векторов двоичного кода с общей проверкой на четность могут быть взяты, например, следующие $n-1$ векторов:

$$
\begin{aligned}
a_1 &= (1\ 1\ 0\ 0\ \ldots\ 0),\\
a_2 &= (1\ 0\ 1\ 0\ \ \ \ \ \ 0),\\
a_3 &= (1\ 0\ 0\ 1\ \ \ \ \ \ 0),\\
&\ \ \ .\ \ .\ \ .\ \ .\ \ \ \ \ \ \ .\\
a_{n-1} &= (1\ 0\ 0\ 0\ \ \ \ \ \ 1).
\end{aligned}
$$

Поэтому матрица

$$
G = \begin{pmatrix}
1 & 1 & 0 & 0 & \ldots & 0\\
1 & 0 & 1 & 0 & \ldots & 0\\
1 & 0 & 0 & 1 & \ldots & 0\\
. & . & . & . & . & .\\
1 & 0 & 0 & 0 & \ldots & 1
\end{pmatrix}
$$

Порождающая матрица

Пример. Найдем порождающую матрицу кода Хемминга длины 7 с проверочной матрицей (2). В данном случае требуется найти 4 независимых решения следующей системы:

$$\begin{aligned} x_4 + x_5 + x_6 + x_7 &= 0, \\ x_2 + x_3 + x_6 + x_7 &= 0, \\ x_1 + x_3 + x_5 + x_7 &= 0. \end{aligned}$$

Разрешаем систему относительно неизвестных x_1, x_2, x_4:

$$\begin{aligned} x_1 &= x_3 + x_5 + x_7, \\ x_2 &= x_3 + x_6 + x_7, \\ x_4 &= x_5 + x_6 + x_7. \end{aligned} \qquad (8)$$

Порождающая матрица

Неизвестным x_3, x_5, x_6, x_7 можно придавать любые значения; тогда из равенств (8) могут быть определены оставшиеся неизвестные x_1, x_2, x_4. Придавая поочередно одному из неизвестных x_3, x_5, x_6, x_7 значение 1, а остальным — 0, получим 4 решения:

$$a_1=(1110000),\quad a_2=(1001100),\quad a_3=(0101010),\quad a_4=(1101001),$$

которые, как читатель может убедиться, линейно независимы. Составленная из этих решений матрица

$$G=\begin{pmatrix}1&1&1&0&0&0&0\\1&0&0&1&1&0&0\\0&1&0&1&0&1&0\\1&1&0&1&0&0&1\end{pmatrix}$$

и является искомой порождающей матрицей.

По поводу сказанного в этом параграфе возникает множество вопросов. Например, как, зная порождающую и проверочную матрицы, выяснить корректирующие способности данного кода; как реализовать эти способности (т. е. каким должен быть алгоритм декодирования, исправляю-

Порождающая матрица

щий или обнаруживающий ошибки); как, наконец, после исправления ошибок выделить информационные символы?

Остановимся вначале на последнем вопросе. Пусть, исправив ошибки, мы получили некоторый кодовый вектор $a=(a_1, a_2, \ldots, a_n)$. Тогда соответствующие ему информационные символы $\alpha_1, \alpha_2, \ldots, \alpha_k$ определяются из равенства (6). Иными словами, вектор $(\alpha_1, \alpha_2, \ldots, \alpha_k)$ есть решение (как можно показать,— единственное) системы линейных уравнений:

$$\begin{aligned}
&a_{11}\alpha_1+a_{21}\alpha_2+\ldots+a_{k1}\alpha_k=a_1,\\
&a_{12}\alpha_1+a_{22}\alpha_2+\ldots+a_{k2}\alpha_k=a_2,\\
&\ldots\quad\ldots\quad\ldots\\
&a_{1n}\alpha_1+a_{2n}\alpha_2+\ldots+a_{kn}\alpha_k=a_n.
\end{aligned}$$

Порождающая матрица

Линейный (n, k)-код называется *систематическим*, если его порождающая матрица имеет вид

$$G=\begin{pmatrix} 1 \ldots 0 \ldots 0 & p_{11} \ldots p_{1m} \\ 0 \ldots 1 \ldots 0 & p_{21} \ldots p_{2m} \\ \cdot \quad\quad \cdot & \\ 0 \quad 0 \quad\quad 1 & p_{k1} \quad\quad p_{km} \end{pmatrix} \tag{9}$$

или вид

$$G=\begin{pmatrix} p_{11} \ldots p_{1m} & 1\ 0 \ldots 0 \\ p_{21} \ldots p_{2m} & 0\ 1 \quad\ 0 \\ \cdot \quad\quad \cdot & \cdot \quad\quad \cdot \\ p_{k1} \ldots p_{km} & 0\ 0 \ldots 1 \end{pmatrix}, \tag{9'}$$

т. е. если она получается приписыванием к единичной матрице порядка k некоторой матрицы из k строк и $m=n-k$ столбцов (иначе говоря, матрицы порядка $k \times m$).

Порождающая матрица

В случае (9) равенство (6) принимает вид:

$$a = (\alpha_1\alpha_2\ldots\alpha_k)\begin{pmatrix} 1 & 0 & \ldots & 0 & p_{11} & \ldots & p_{1m} \\ 0 & 1 & & 0 & p_{21} & \ldots & p_{2m} \\ \ldots & & & \ldots & \ldots & & \ldots \\ 0 & 0 & & 1 & p_{k1} & \ldots & p_{km} \end{pmatrix} = (\alpha_1\alpha_2\ldots\alpha_k a_{k+1}\ldots a_n).$$

Таким образом, первые k символов любого кодового слова как раз и оказываются информационными, а остальные (они являются проверочными) связаны с информационными символами соотношениями:

$$\begin{aligned} a_{k+1} &= p_{11}\alpha_1 + p_{21}\alpha_2 + \ldots + p_{k1}\alpha_k, \\ a_{k+2} &= p_{12}\alpha_1 + p_{22}\alpha_2 + \ldots + p_{k2}\alpha_k, \\ &\ldots\ldots\ldots\ldots\ldots\ldots \\ a_n &= p_{1m}\alpha_1 + p_{2m}\alpha_2 + \ldots + p_{km}\alpha_k. \end{aligned} \qquad (10)$$

Порождающая матрица

Равенства (10), очевидно, образуют систему проверочных соотношений систематического линейного кода. Очень важно, что всякий линейный код в некотором смысле эквивалентен систематическому (см. дополнение 9).

Дальнейшее прольет свет и на другие вопросы, поставленные выше.