Теория кодирования

Гошин Егор Вячеславович, к.т.н., доцент
кафедры суперкомпьютеров и общей информатики

Лекция 1-1

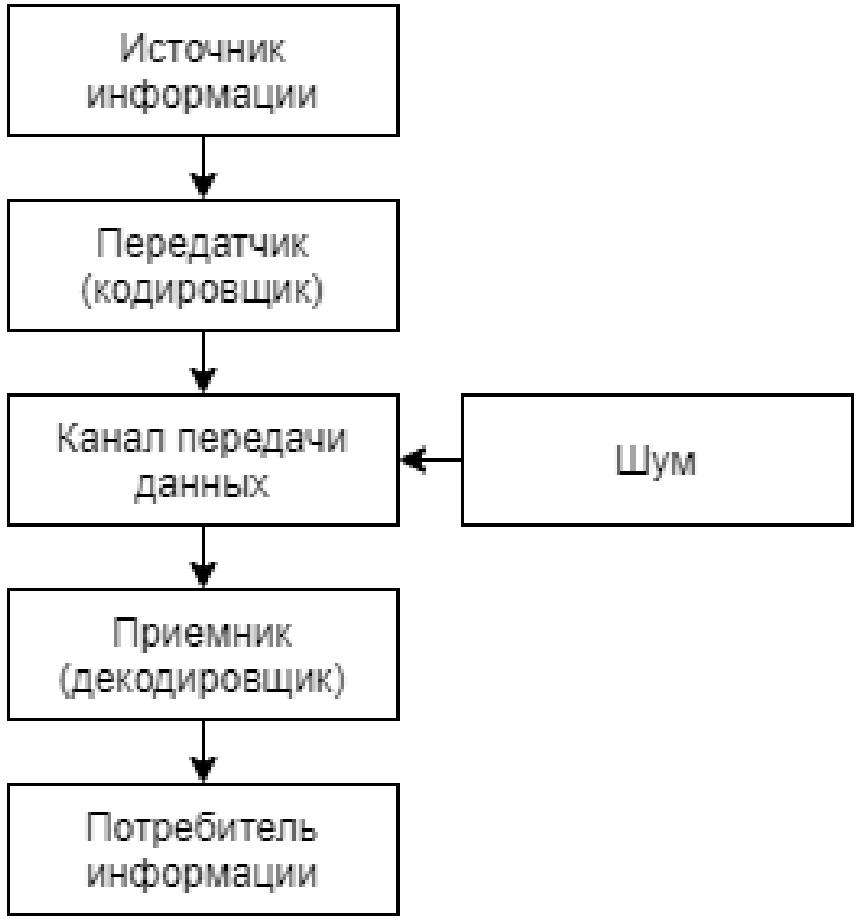
Введение в теорию кодирования

Введение. Задачи теории кодирования

- Теория кодирования изучает методы эффективной и надёжной передачи информации из одного места в другое.

- Эффективное кодирование (сжатие)
- **Помехоустойчивое кодирование**
- Шифрование

- Наличие шума – существенно!



Введение. Задачи теории кодирования

Требования к кодировщику и декодировщику:

- быстрое кодирование информации,
- простая передача закодированного сообщения,
- быстрое декодирование полученных сообщений
- **исправление ошибок, возникающих в канале**,
- передача максимального количества информации за единицу времени.

Основные термины и положения

- Информация передаётся в форме нулей и единиц.
- Каждый 0 или 1 – **разряд** или **знак**.
- Последовательность знаков – **слово**.
- **Длина** слова – число разрядов в этом слове.

  Длина слова 1010110 равна семи.
- Слова передаются по двоичному каналу поразрядно и последовательно.

Базовые положения. Определения

- **Двоичным кодом** C называется набор из двоичных слов. Например, код, состоящий из всех слов длины 2 выглядит как
$$C = \{00, 10, 01, 11\}$$
- **Блочный код** – это код, состоящий из слов одинаковой длины. Эта длина называется длиной кода. В настоящем курсе будут рассматриваться только двоичные блочные коды.
- Слова, принадлежащие коду называются **кодовыми словами**. Число кодовых слов будет обозначаться $|C|$.

Передача данных

- Переданное **кодовое слово** длины n, состоящее из нулей и единиц, принимается как **слово** длины n, состоящее из нулей и единиц, хотя не обязательно то же самое.
- Начало первого слова однозначным образом определено. Кроме того, это означает, что переданная последовательность гарантированно кратна длине кодового слова.
- Шум в канале распределён равномерно. Вероятность верной передачи одного разряда постоянна и равна p.
- Вероятность верной передачи разряда выше вероятности ошибки $p > 0.5$.

Исправление и обнаружение ошибок

- Если принятое слово не является кодовым словом, очевидно, что в ходе передачи возникла ошибка (или несколько). Этот процесс называется обнаружением ошибки.

- В ряде случаев отправленное кодовое слово может быть получено на основе принятого слова с ошибкой. Этот процесс называется исправлением ошибки.

Исправление и обнаружение ошибок

- Задан код $C_1 = \{00, 01, 10, 11\}$.

Исправление и обнаружение ошибок невозможно, поскольку все возможные принятые слова будут кодовыми.

- Задан код $C_2 = \{000000, 010101, 101010, 111111\}$ (**код повторения**).

Пусть принято слово 110101.

Обнаружение ошибки: слово не входит в число кодовых, поэтому ошибка присутствует.

Исправление ошибки: наиболее «близким» и потому наиболее вероятным переданным словом является 010101.

Исправление и обнаружение ошибок

- Задан код $C_3 = \{000, 011, 101, 110\}$ (**код с проверкой на чётность**).
- Пусть принято слово 010.
- **Обнаружение ошибки**: слово не входит в число кодовых, поэтому ошибка присутствует.
- **Исправление ошибки** невозможно, поскольку три кодовых слова $(000, 011, 101)$ одинаково «близки» к принятому слову.

Скорость кода

- **Скоростью кода** называется доля полезного сообщения в каждом кодовом слове. Для кода C длины n скорость кода определяется как

$$\frac{1}{n}\log_2|C|$$

$C_1 = \{00, 01, 10, 11\}$, *скорость кода равна 1*

$C_2 = \{000000, 010101, 101010, 111111\}$, *скорость кода равна 1/3.*

$C_3 = \{000, 011, 101, 110\}$, *скорость кода равна 2/3.*

Зачем нужно обнаружение ошибок?

- Рассмотрим следующий случай:
- Пусть все 2^{11} слов длины 11 являются кодовыми словами, в этом случае обнаружение ошибок невозможно. Пусть надёжность канала равна $p = 1 - 10^{-8}$ и данные передаются со скоростью 10^7 знаков в секунду. Тогда вероятность, что слово будет передано неправильно примерно равно $11p^{10}(1-p) \sim \frac{11}{10^8}$.

$$\frac{11}{10^8} \cdot \frac{10^7}{11} = 0.1$$

Проверка на чётность

- Добавим бит проверки на чётность.
- В этом случае все однократные ошибки будут обнаружены.
- Вероятность двукратной ошибки в одном слове равна $1 - p^{12} - 12p^{11}(1-p)$, что для $p = 1 - 10^{-8}$ приблизительно равно $\frac{66}{10^{16}}$.

$$\frac{66}{10^{16}} \cdot \frac{10^7}{12} = 5.5 \cdot 10^{-9}$$

Поиск наиболее вероятного кодового слова

- Для известных переданного и принятого слов можно вычислить вероятность этого события

$$\phi_p(v, w) = p^{n-d}(1-p)^d$$

- Пример:
- Пусть C – код длины 5. Тогда для любого $v \in C$ вероятность того, что v принято верно равна

$$\phi_p(v, w) = p^5$$

- Пусть $10101 \in C$. Тогда

$$\phi_p(10101,10110) = p^3(1-p)^2$$

- и для $p = 0.9$

$$\phi_p(10101,10110) = 0.9^3 0.1^2 = 0.00729$$

Поиск наиболее вероятного кодового слова

- На практике мы знаем принятое слово w, но не отправленное.

- Однако для каждого кодового слова v можно определить вероятность того, что было принято слово w.

- Предположим, что отправленным словом является то, которое максимизирует вероятность появления принятого:

$$\phi_p(v, w) = \max\{\phi_p(u, w) : u \in C\}$$

Поиск наиболее вероятного кодового слова

- Теорема:
- Пусть имеется двоичный канал с вероятностью правильной передачи $\frac{1}{2} < p < 1$. Пусть v_1 и w отличаются в d_1 позициях, а v_2 и w отличаются в d_2 позициях. Тогда

$$\phi_p(v_1, w) \leq \phi_p(v_2, w)$$

- тогда и только тогда, когда

$$d_1 \geq d_2$$

Поиск наиболее вероятного кодового слова

Пусть принято слово $w = 01100$ по двоичному каналу. Какое из слов $10110,\ \ 10010,\ \ 00101,\ \ 10101$ было отправлено с наибольшей вероятностью?

v	d (число различий в разрядах)
10110	3
10010	4
00101	2
10101	3

Немного базовой алгебры

- Задача, которую мы планируем решать, посвящена поиску эффективного способа определения ближайшего кодового слова.
- Введём несколько понятий.
- Пусть $K = \{0, 1\}$ и K^n – набор всех двоичных слов длины n.
- Определим сложение и умножение элементов K следующим образом:
- $0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 0,$
- $0 \cdot 0 = 0, 0 \cdot 1 = 0, 1 \cdot 0 = 0, 1 \cdot 1 = 1$
- Операции элементов из K^n определяются аналогичным образом поэлементно, например:

$$10110 + 10011 = 00101$$

Немного базовой алгебры

- Можно показать, что K^n – векторное пространство и выполняются следующие свойства для любых слов u, v, w длины n и для любых скаляров a и b.
- 1. $v + w \in K^n$
- 2. $(u + v) + w = u + (v + w)$
- 3. $v + 0 = 0 + v = v$, где 0 – нулевое слово.
- 4. Существует $v \in K^n$: $v + v' = v' + v = 0$
- 5. $v + w = w + v$
- 6. $av \in K^n$
- 7. $a(v + w) = av + aw$
- 8. $(a + b)v = av + bv$
- 9. $(ab)v = a(bv)$
- 10. $1v = v$

Немного базовой алгебры

- Обратим внимание, что если отправлено кодовое слово v , а принято слово w, то в сумме $v + w$ единицы располагаются в разрядах возникновения ошибки, а нули в разрядах без ошибки.

- $v + w$ называется ошибкой или шаблоном ошибки.

Вес и расстояние

- Введём для важных понятия.
- Пусть v – слово длины n. Весом Хэмминга (или просто весом) v называется число единиц в v. Обозначим вес v как $wt(v)$. Например, $wt(100101) = 3$.
- Пусть v и w – слова длины n. Расстоянием Хэмминга (или просто расстоянием) между v и w называют число разрядов, в которых эти слова отличаются.

Свойства веса и расстояния Хэмминга

- Пусть u, v и w – слова длины n и a – разряд. Тогда:
- 1. $0 \leq wt(v) \leq n$
- 2. $wt(0) = 0$
- 3. $wt(v) = 0 \ \rightarrow v = 0$
- 4. $0 \leq d(v, w) \leq n$
- 5. $d(v, v) = 0$
- 6. $d(v, w) = 0 \ \rightarrow v = w$
- 7. $d(v, w) = d(w, v)$
- 8. $wt(v + w) \leq wt(v) + wt(w)$
- 9. $d(v, w) \leq d(v, u) + d(u, w)$
- 10. $wt(av) = a \cdot wt(v)$
- 11. $d(av, aw) = a \cdot d(v, w)$

Кодирование сообщений

В первую очередь необходимо задать значение k – длину бинарного слова, соответствующего сообщению.

Поскольку разные сообщения должны соответствовать различным двоичным словам, значение k должно удовлетворять условию

$$|M| \leq \left|K^k\right| = 2^k$$

Затем определяется число разрядов, которые нужно добавить к каждому кодовому слову длины k для того, чтобы иметь возможность исправлять заданной кратности. Полученное значение, n – длина кодового слова, соответствующего сообщению.

Декодирование методом максимального правдоподобия

Пусть принято слово w.

Полное декодирование методом максимального правдоподобия

Если существует единственное слово v в коде C, такое, что $d(v, w) < d(v_1, w)$ для любых $v_1 \in C$, то w декодируется как v.

Иначе, w декодируется как произвольное из имеющих наименьшее расстояние.

Неполное декодирование методом максимального правдоподобия

Если существует единственное слово v в коде C, такое, что $d(v, w) < d(v_1, w)$ для любых $v_1 \in C$, то w декодируется как v.

Иначе, w не декодируется.

Пример для $|M| = 2, n = 3, C = \{000, 111\}$

Полученное слово	**Ошибка** **000 + w**	**Ошибка** **111 + w**	**Декодированное слово**
000	000	111	000
100	100	011	000
010	010	101	000
001	001	110	000
110	110	001	111
101	101	010	111
011	011	100	111
111	111	000	111

Пример для $|M| = 3, n = 4, C = \{0000, 1010, 0111\}$

Полученное слово	Ошибка 0000 + w	Ошибка 1010 + w	Ошибка 0111 + w	Декодированное слово
0000	0000	1010	0111	0000
1000	1000	0010	1111	???
0100	0100	1110	0011	0000
0010	0010	1000	0101	???
0001	0001	1011	0110	0000
1100	1100	0110	1011	???
1010	1010	0000	1101	1010
1001	1001	0011	1110	???

Полученное слово	Ошибка 0000 + w	Ошибка 1010 + w	Ошибка 0111 + w	Декодированное слово
0110	0110	1100	0001	0111
0101	0101	1111	0010	0111
0011	0011	1001	0100	0111
1110	1110	0100	1001	1010
1101	1101	0111	1010	0111
1011	1011	0001	1100	1010
0111	0111	1101	0000	0111
1111	1111	0101	1000	0111

Код для обнаружения ошибок

- Пусть отправлено $v \in C$ и получено $w \in K^n$. Код C обнаруживает ошибку $v + w$ тогда и только тогда, когда w не является кодовым словом для любого $v \in C$.

- Иначе говоря, ошибка u обнаруживается, если для любого передаваемого слова v декодер после получения $v + u$ может определить, что оно не является кодовым словом.

Пример

- $C = \{001, 101, 110\}$
- Обнаруживаемые ошибки: $\{001, 010, 101, 110\}$
- Не обнаруживаемые ошибки: $\{011, 100, 111\}$

Теорема о связи кодового расстояния и кратности обнаруживаемых ошибок

- Код C с расстоянием d может обнаруживать по меньшей мере все ненулевые шаблоны ошибок веса не более, чем $d-1$. При этом существует по меньшей мере один шаблон ошибок веса d, который не может быть обнаружен.

- Будем называть код обнаруживающим ошибки кратности t, если он позволяет обнаружить все ошибки веса не более t и при этом не позволяет обнаружить хотя бы одну ошибку кратности $t+1$.

Код для исправления ошибок

- Пусть отправлено $v \in C$ и получено $w \in K^n$. Код C исправляет ошибку $v + w$ тогда и только тогда, когда w ближе к v, чем к любому другому кодовому слову из C.
- Иначе говоря, ошибка u обнаруживается, если для любого передаваемого слова v слово $v + u$ ближе к v , чем к другому кодовому слову.
- По аналогии с обнаружением ошибок говорят, что код является исправляющим ошибки кратности t, если он исправляет все ошибки веса не более чем t и не исправляет хотя бы одну ошибку веса $t + 1$.

Коды для исправления ошибок

- $C = \{000, 111\}$
- $u = 010, v = 000$
- $d(000, v + u) = d(000, 010) = 1$
- $d(111, v + u) = d(111, 010) = 2$

- $u = 110, v = 000$
- $d(000, v + u) = d(000, 110) = 2$
- $d(111, v + u) = d(111, 110) = 1$

Связь кодового расстояния и кратности

- Код C с расстоянием d может исправлять все ошибки веса не более, чем $\left[\frac{(d-1)}{2}\right]$. При этом существует по меньшей мере один шаблон ошибок веса $1+\left[\frac{d-1}{2}\right]$, который этот код не исправляет.

Пример

- $C = \{\, 000000,\ 100101,\ 010110,\ 001111,\ 110011,\ 101010,\ 011001,\ 111100\}$
- $d = 3$
- $t = 1$
- $u_1 = 001000$
- $u_2 = 001001$

Лекция 1-2

Линейные коды

Определение линейного кода

- Код C называется линейным, если $v + w$ является кодовым словом из C для v и w из C.
- Таким образом, линейный код замкнут относительно операции сложения.
- Код {000, 111} – линейный. Код {000, 001, 101} – не является линейным.
- Линейный код обязательно включает в себя нулевое кодовое слово.
- Главное достоинство линейного кода – простота нахождения кодового расстояния. Оно равно наименьшей длине ненулевого кодового слова.

Достоинства линейных кодов

- 1. Для линейных кодов процедура декодирования ММП значительно проще и быстрее, чем описанная ранее.
- 2. Кодирование линейных кодов быстрее и требует меньшего объёма памяти, чем для произвольных кодов.
- 3. Наборы обнаруживаемых и исправляемых ошибок могут быть описаны в гораздо более простой форме.

Подпространства

- Непустое подмножество U векторного пространства V является подпространством V, если U замкнуто относительно векторного сложения и скалярного умножения, так, если v и w являются элементами U, то $v + w$ и av являются элементами U для любого скалярного a.
- В частности, если определены только два скаляра: 0 и 1, U является подпространством K^n тогда и только тогда, когда U замкнуто относительно сложения.

Линейная оболочка

- Вектор w называется линейной комбинацией векторов $v_1, v_2, \dots, v_k$, если существует набор скаляров таких, что
$$w = a_1 v_1 + a_2 v_2 + \cdots + a_k v_k$$
- Набор всех линейных комбинаций векторов в заданном множестве $S = \{v_1, v_2, \dots, v_k\}$ называется линейной оболочкой и обозначается $< S >$. Если S – пустое, то $< \mathrm{S} >= \{0\}$.
- В линейной алгебре показано, что для любого подмножества S векторного пространства V, линейная оболочка $< \mathrm{S} >$ является подпространством V.

Код, порождённый подмножеством

- Для векторного пространства K^n существует очень простое описание $< S >$. Поскольку $< S >$ – подпространство, оно является линейным кодом. Будем называть его кодом, порождённым S.
- Теорема:
- Для любого подмножества S из K^n код $C =< U >$, порождённый S состоит исключительно из следующих слов: нулевое, все слова из S и все суммы двух и более слов из S.

Пример

- Пусть $S = \{0100, 0011, 1100\}$, тогда код C, порождённый S состоит из следующих слов:
- $0000, 0100, 0011, 1100, 0100 + 0011 = 0111, 0100 + 1100 = 1000, 0011 + 1100 = 1111, 0100 + 0011 + 1100 = 1011$
- Таким образом,
- $C =< S > = \{0000, 0100, 0011, 1100, 0111, 1000, 1111, 1011\}$

Скалярное произведение

- Если $v = (a_1, a_2, \dots, a_n)$, $w = (b_1, b_2, \dots, b_n)$ векторы из K^n, можно определить скалярное произведение $v \cdot w$ как
$$v \cdot w = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$
- Обратим внимание, что $v \cdot w$ – скаляр, а не вектор.
- Например, для K^5:
$$11001 \cdot 01101 = 1 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 = 0$$

Ортогональность и дуальный код

- Векторы ортогональны, если их скалярное произведение равно нулю.
- Для заданного множества векторов S говорят, что вектор v ортогонален этому множеству, если он ортогонален всем векторам этого множества.
- Множество векторов, ортогональных некоторому множеству S называется комплементарным этому множеству и обозначается как $S^{\perp}$.
- Для векторного пространства K^n если $C =< S >$, $C^{\perp} = S^{\perp}$ является подпространством и называется кодом, дуальным к C.

Независимость векторов

- Множество $S = \{v_1, v_2, \dots, v_k\}$ векторов называется линейно зависимым, если существует множество скаляров $a_1, a_2, \dots, a_k$, из которых хотя бы один ненулевой, таких что:

$$a_1 v_1 + a_2 v_2 + \cdots + a_k v_k = 0$$

иначе это множество называется линейно независимым.

Базис, размерность

Непустое подмножество B векторов из векторного пространства V называется базисом V, если:

- 1. $B =< V >$,
- 2. B – линейно независимое множество.

Любое линейно независимое множество B является базисом $< B >$.

Описание линейного кода через базис

- Базис предоставляет удобный способ описания линейного кода. Для любого векторного пространства V любой вектор $w \in V$ может быть представлен в виде линейной комбинации базисных векторов. И для каждого вектора из этого пространства коэффициенты в линейной комбинации будут уникальными.

- *Линейный код размерности состоит ровно из 2^k векторов.*

- *Пусть $C =< S >$ – линейный код, порождаемый $S \subseteq K^n$. Тогда (размерность C + размерность $C^{\perp}$) = n.*

Матрицы

- Определяют две элементарные операции над строками:

1. Перестановка двух строк.

2. Сложение двух строк и замена получившейся суммой одной из них.

- Две матрицы строчно-эквивалентны если одна из них может получена из другой последовательностью элементарных операций.

Матрица ступенчатого вида

Первая единица в строке называется ведущим элементом. Столбец, в котором есть ведущий элемент, называется ведущим.

Матрица M считается матрицей ступенчатого вида по строкам, если:

- все ненулевые строки (имеющие по крайней мере один ненулевой элемент) располагаются над всеми чисто нулевыми строками;
- ведущий элемент (первый ненулевой элемент строки при отсчёте слева направо) каждой ненулевой строки располагается строго правее ведущего элемента в строке, расположенной выше данной.

Если вдобавок к этому в каждом ведущем столбце присутствует единственная единица, такая матрица называется матрицей приведённого ступенчатого вида по строкам.

Любая матрица может быть приведена к эквивалентной матрице в ступенчатой форме.

Базы для линейного и дуального кодов

- Пусть S непустое подмножество K^n. Рассмотрим алгоритмы, формирующие базис для $C =< S >$ для линейного кода, генерируемого S.

- Алгоритм 1.

1. Сформировать матрицу A, состоящую из строк из S.
2. Используя элементарные операции, сформировать матрицы ступенчатого вида из A.
3. Ненулевые строки из полученной матрицы формируют базис для $C =< S >$.

Пример

- $S = \{11101, 10110, 01011, 11010\}$
- $A = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
- Базис: $\{11101, 01011, 00111\}$

Альтернативный алгоритм

- Алгоритм 2.

1. Сформировать матрицу A, состоящую из столбцов из S.
2. Используя элементарные операции, сформировать матрицу ступенчатого вида из A.
3. Ведущие столбцы полученной матрицы соответствуют базисным столбцам в A.

Пример

- $S = \{11101, 10110, 01011, 11010\}$
- $A = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
- Базис: $\{11101, 10110, 11010\}$

Алгоритм формирования базиса для дуального кода

- Алгоритм 3.

1. Сформировать матрицу A, состоящую из строк из S.
2. Используя элементарные операции, сформировать матрицу приведённого ступенчатого вида из A.
3. G – матрица размерности $k \times n$, состоящая из ненулевых строк полученной матрицы.
4. X – матрица размерности $k \times (n-k)$, полученная из G удалением ведущих столбцов.
5. Сформировать H размерности $n \times (n-k)$ следующим образом:

- В строках H, соответствующих главным столбцам G располагаются строки X.

- В оставшихся $n-k$ строках H располагаются строки единичной матрицы I размерности $(n-k) \times (n-k)$.

6. Столбцы полученной матрицы формируют базис для $\mathrm{C}^{\perp}$.

123

- S=
- $A = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
- $G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, H = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$

Порождающая матрица и кодирование

- Ранг матрицы – число ненулевых строк в матрице ступенчатого вида.
- Если C имеет длину n, расстояние d и размерность k, будем обозначать такой код как линейный код (n, k, d). Эти три параметра в полной мере характеризуют линейный код.
- Если C – линейный код длины n и размерности k, любая матрица, строки которой формируют базис в C называется **порождающей** матрицей.

Свойства порождающей матрицы

- Теорема

Матрица G является порождающей для некоторого линейного кода тогда и только тогда, когда строки G линейно независимы или, что эквивалентно, если ранг матрицы равен числу её строк.

- Теорема

Если G порождающая матрица линейного кода C, тогда любая строчно-эквивалентная матрица также является порождающей матрицей для C. В частности, любой линейный код имеет порождающую матрицу в приведённом ступенчатом виде.

Формирования матрицы линейного кода

- Для того, чтобы найти порождающую матрицу линейного кода, можно сформировать матрицу, строки которой представляют собой кодовые слова из C. Поскольку C, алгоритм 1 можно использовать для формирования базиса C.
- Матрица, строки которой являются базисными векторами, является порождающей матрицей C.

Матрица кода

- Пример:

$C = \{0000, 1110, 0111, 1001\}.$

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Формирование кода с помощью матрицы

- Пусть C – линейный код длины n и размерности k. Если G – порождающая матрица для C и если слово u длины k записано в форме вектора-строки, тогда $v = uG$ – кодовое слово из C, поскольку v – линейная комбинация строк из G, формирующих базис в C.

- На самом деле, если $u = (a_1, \dots, a_k)$ и $G = \begin{bmatrix} g_1 \\ g_2 \\ \dots \\ g_k \end{bmatrix}$

тогда $v = uG = a_1 g_1 + a_2 g_2 + \cdots + a_k g_k$.

Формирование линейного кода

- Теорема

Если G – порождающая матрица линейного кода C длины n и размерности k, тогда $v = uG$ принимает все возможные значения из 2^k слов в C в то время как u принимает все возможные значения из 2^k слов длины k. Таким образом, C представляет собой набор всех слов uG для u из K^2. Более того, $u_1G = u_2G$ тогда и только тогда, когда $u_1 = u_2$.

Кодирование с использованием матрицы

- Теорема говорит о том, что линейный код (n, k, d) заданный порождающей матрицей G позволяет закодировать все сообщения из K^k посредством умножения этих сообщений на порождающую.

- Заметим, что скорость такого кода равна

$$\frac{\log_2 2^k}{n} = \frac{k}{n}$$

Пример

- Пусть C линейный код $(5,3,d)$ с порождающей матрицей G

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

Скорость такого кода равна $\frac{3}{5}$. Все сообщения в K^3 могут быть закодированы. Например, для сообщения $u = 101$

$$v = uG = [1 \quad 0 \quad 1] \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} = [1 \quad 0 \quad 0 \quad 1 \quad 1]$$

Проверочная матрица

- Рассмотрим ещё одну матрицу, связанную с линейным кодом и порождающей матрицей. Матрица H называется проверочной для линейного кода, если её столбцы формируют базис дуального кода $C^{\perp}$. Если C имеет длину n и размерность k, то проверочная матрица должна иметь n строк, $n-k$ столбцов и ранг $n-k$.

Теоремы

- Матрица H является проверочной для некоторого кода C тогда и только тогда, когда столбцы H линейно независимы.

- Если H – проверочная матрица линейного кода C длины n, тогда C состоит из всех слов $v \in K^n$ таких, что $vH = 0$.

Формирование порождающей матрицы

- Если известная порождающая матрица линейного кода C, можно найти проверочную матрицу, используя алгоритм 3, поскольку столбцы H формируют базис $C^{\perp}$.

- Пример.

Для кода $C = \{0000, 1110, 0111, 1001\}$порождающая матрица имеет вид:

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

Можно сформировать H в виде

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

Связь порождающей и проверочной матриц

Матрицы G и H являются порождающей и проверочной матрицами, соответственно, для некоторого линейного кода C тогда и только тогда, когда:

1. Строки G линейно независимы.

2. Столбцы H линейно независимы

3. число строк в G плюс число столбцов в H равно числу столбцов в G и числу строк в H.

4. $GH = 0$