Оглавление

Лекция 11: Избыточное кодирование. Криптография.

22.11.2023

0.1 Код Хэмминга

При передаче или хранении данных возможны искажения. Аппаратное обеспечение не идеально. Хочется научиться находить и исправлять такие ошибки. Для этого нужно передавать какую-то избыточную информацию.

Избыточное кодирование — это вид кодирования, использующий избыточное количество информации с целью последующего контроля целостности данных при записи/воспроизведении информации или при её передаче по линиям связи.

Код Хэмминг - это алгоритм, который позволяет закодировать какое-либо информационное сообщение определённым образом, после передачи определить, появилась ли какая-то ошибка в этом сообщении во время его передачи, и, при возможности, восстановить это сообщение.

Рассмотрим самый простой код Хэмминга (может исправлять только одну ошибку). Также существуют более совершенные модификации данного алгоритма, которые позволяют обнаруживать большее количество ошибок.

Алгоритм (Код Хэмминга). Пусть требуется передать какое-либо сообщение a, состоящее из n битов (например, $a = [0, 1, \ldots, 1]$). Тогда, для построения кода Хэмминга, потребуется передать всего $N + 1 (N \geq n)$ битов, среди которых будут контрольные (необходимо распознать $N+1$ положение ошибки, значит справедливо $2^{N-n} \geq N + 1$)

1. Перенумеруем биты от 1 до N, номера, являющиеся степенями двойки отводятся под контрольные биты.
2. Контрольному биту 2^i сопостовляется множество $p_i = \{j : j \& i = i\}$, где j – номер в коде, & – побитовое И. Другими словами, 2^i-ому контрольному биту сопоставляются номера кода j, такие, что в двоичном представлении j на $\log_2 i$ позиции стоит единица. (порядок нумерации для двоичных чисел)

3. 2^i-ый контрольный бит принимает такое значение, чтобы:

$$\bigoplus_{j \in P_i} a[j] = 0 \text{ – контрольное соотношение}$$

Декодирование: Если все контрольные соотношения сходятся, тогда сообщение передано без ошибок. Отбрасываем контрольные биты и выводим сообщение.

1. если не сошлись соотношения, соответствующие множествам $P_1, P_2, \ldots, P_k$, то берем номер $err = \min\{P_1 \cap P_2 \cap \ldots \cap P_k\}$ — номер с ошибкой.
2. Берем $a[err] = \neg a[err]$
3. если не сошлось всего одно соотношение, соответствующее P_i, тогда контрольный бит передался с ошибкой. Просто отбрасываем его и все контрольные биты и передаем сообщение.

Пример. $a = [1, 0, 0, 1, 1, 1, 1]$

Добавляем контрольные биты, меняем нумерацию и состовляем множества для контрольных битов:

i	1	2	3	4	5	6	7	8	9	10	11
a			1		0	0	1		1	1	1
P_0	1		3		5		7		9		11
P_1		2	3			6	7			10	11
P_2				4	5	6	7				
P_3								8	9	10	11

Вычисляем XOR для $P_0, \ldots, P_3$. соответствующие им контрольные биты с номерами 1, 2, 4, 8 принимают значения: 0, 0, 1, 1. Тогда код Хэмминга для собощения:

$$a = [0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1]$$

Пусть передали сообщение с ошибкой: 00111**1**011111. Вычисляем контрольные соотношения, если не сошлись с исходыми, то ищем номер, на котором ошибка и заменяем ее: $err = \min\{\{1, 3, \ldots, 11\} \cap \{4, 5, 6, 7\}\} = 5$.

$a[err] = \neg 1 = 0$. Передаваемое сообщение исправлено.

0.2 Шифрование

Определение 1. Сообщение — то, что защищается от несанкционированного доступа.
Криптографический алгоритм — алгоритм, который используется для шифрования или дешифрования исходного сообщения.
Ключ — вспомогательная информация, используемая алгоритмом.

Пример. (Шифрование с закрытым ключом) $\{a_i\}$ – псевдослучайная последовательность битов – ключ
$\{b_i\}$ – исходное сообщение
$\{c_i\}$ – зашифрованное сообщение, полученное следующим образом:

$$\forall i : c_i = a_i \oplus b_i$$

Для шифрования и дешифрования используется один и тот же ключ.

Алгоритм. (RSA)

1. Выбираются два больших простых числа p и q, $n = pq$.
2. Вычисляется функция Эйлера $\varphi(n) = (p-1)(q-1)$.
3. Выбирается целое число $e \in (1, \varphi(n))$, такое, что $\gcd(e, \varphi(n)) = 1$.
4. Вычисляется число d, обратное к e по модулю $\varphi(n)$, то есть такое, что $ed \equiv 1 \pmod{\varphi(n)}$.

Пара (n, e) – открытый ключ, пара (n, d) – закрытый ключ.

Замечание. Потребуется теорема Эйлера и единственность d (следует из линейного представления НОД).

На лекции это было вынесено в 2 отедльные леммы (не знаю зачем)

Алгоритм. (Применение RSA)

- **Шифрование:** Пусть m — шифруемое сообщение. Предположим, что m представим в виде числа от 0 до $n-1$ (иначе разобьем на блоки). Отправитель вычисляет зашифрованное сообщение: $c \equiv m^e \mod n$
- **Дешифрование:** Получатель получает c и вычисляет $c^d \equiv m^{ed} \equiv m \mod n$

Замечание. Асимметричное шифрование — довольно ресурсоемкая процедура, в отличие от симметричного. Как упростить жизнь?

1. Отправитель генерирует ключ сессии b.
2. Исходное сообщение a шифруется симметричным алгоритмом при помощи
3. ключ сессии b шифруется открытым ключом e — получим цифровой конверт x.
4. c и x передаются получателю.

5. Получатель из x восстанавливает b при помощи своего закрытого ключа, а затем восстанавливает a из c при помощи b.