SATBAYEV UNIVERSITY

Institute of automation and information technologies

Department Electronics, telecommunications and space technologies

Lecture 13
Noise-Resistant Codes.
Methods for Constructing Hamming Codes

Lecturer Dosbayev Zh.

SATBAYEV UNIVERSITY

Content

- Noise-Resistant codes and its parameters
- Classification of noise-resistant codes
- Introduction to Hamming Codes
- Constructing Hamming Codes – Overview
- Methods for constructing and using Hamming codes
- Advantages of Hamming Codes
- Conclusion

Noise-Resistant codes

The purpose of noise–resistant coding is to protect information from interference and errors during transmission and storage of information. Noise-tolerant coding is necessary to eliminate errors that occur during the transmission and storage of information. When transmitting information over a communication channel, interference, errors occur and a small part of the information is lost.

Noise-Resistant coding parameters

The first parameter, the code rate **R**, characterizes the proportion of information («useful») data in a message and is determined by the expression:

;

where

- **n** is the number of characters of the encoded message (encoding result);
- **m** is the number of verification characters added during encoding;
- **k** is the number of information characters.

The parameters n and l are often given together with the name of the code for its unambiguous identification. For example, the Hemming code (7,4) means that 4 characters come to the input of the encoder, 7 characters come to the output, Reed-Solomon (15, 11), etc.

Noise-Resistant coding parameters

The second parameter, **the multiplicity of detected errors**, is the number of erroneous characters that the code can detect.

The third parameter, **the multiplicity of errors to be corrected**, is the number of erroneous characters that the code can correct (indicated by the letter t).

SATBAYEV UNIVERSITY

Classification of noise-resistant codes

1 > 2

Continuous

- the encoding and decoding process is continuous. Convolutional code is a special case of continuous code. One character was received at the input of the encoder, respectively, several appeared at the output, i.e. several outputs are formed for each input character, since redundancy is added.

Block

- the process of encoding and decoding is carried out in blocks. From the point of view of understanding the work, the block code is simpler, we break the code into blocks and each block is encoded separately.

SATBAYEV UNIVERSITY

According to the alphabet used:

Binary

They operate with bits.

Non-binary (Reed-Solomon codes).

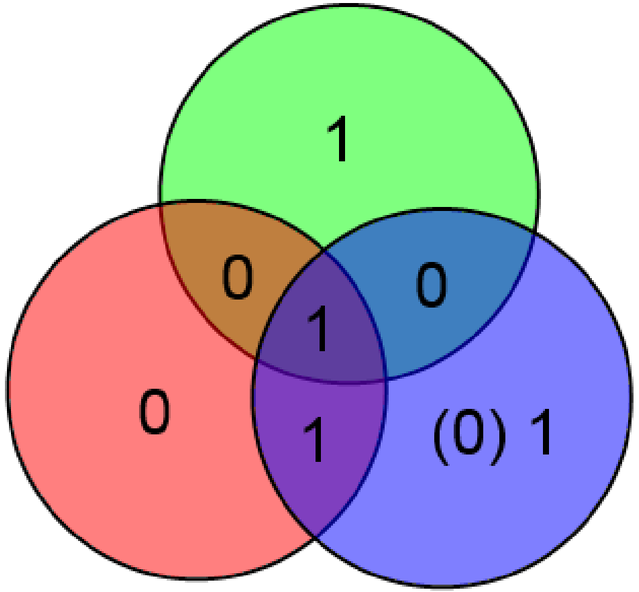
They operate with larger symbols. If the information is initially binary, you need to turn these bits into characters.

For example, there is a sequence 110 110 010 100 and you need to convert them from binary characters to non—binary, take groups of 3 bits — this will be one character, 6, 6, 2, 4 - non-binary noise-resistant codes work with these non-binary characters.

SATBAYEV UNIVERSITY

Hamming Code

Hamming codes are a family of error-correcting codes invented by Richard Hamming in the 1950s to detect and correct errors in digital data transmission. They're widely used in computer memory (ECC RAM) and telecommunications to enhance data integrity in environments prone to noise and transmission errors.



SATBAYEV UNIVERSITY

The Hamming code (7,4) is 4 bits at the input of the encoder and 7 at the output, hence 3 verification bits. From 1 to 4 information bits, from 6 to 7 verification bits (see Table. above). The fifth verification bit y5 is the sum modulo two 1-3 information bits. The sum modulo 2 is the calculation of the parity bit.

$$y_5 = x_1 \oplus x_2 \oplus x_3,$$
$$y_6 = x_2 \oplus x_3 \oplus x_4,$$
$$y_7 = x_1 \oplus x_2 \oplus x_4.$$

Formation of verification bits

$$s_1 = y_1 \oplus y_2 \oplus y_3 \oplus y_5$$
$$s_2 = y_2 \oplus y_3 \oplus y_4 \oplus y_6$$
$$s_3 = y_1 \oplus y_2 \oplus y_4 \oplus y_7$$

Decoding the Hamming code through the syndrome

x_1	x_2	x_3	x_4			
y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	0	0	0	0
1	0	0	0	1	0	1
0	1	0	0	1	1	1
1	1	0	0	0	1	0
0	0	1	0	1	1	0
1	0	1	0	0	1	1
0	1	1	0	0	0	1
1	1	1	0	1	0	0
0	0	0	1	0	1	1
1	0	0	1	1	1	0
0	1	0	1	1	0	0
1	1	0	1	0	0	1
0	0	1	1	1	0	1
1	0	1	1	0	0	0
0	1	1	1	0	1	0
1	1	1	1	1	1	1

Encoding table for (7,4)

Method 1: Choosing the Code Length and Parity BitsTotal Bits (n):

Sum of data bits (k) and parity bits (r).

Relationship: n=k+r

Hamming Condition: For a code to correct single-bit errors, the number of parity bits r must satisfy:

$$\geq n+1$$

Example: For 4 data bits, 3 parity bits are needed, as =8≥4+3+12 3 =8≥4+3+1.

Method 2: Placing the Parity Bits

Parity Bits Placement: Positions that are powers of 2 (1, 2, 4, etc.).
Data Bits Placement: Fill in remaining positions.
Example: For a 7-bit Hamming code, data bits are in positions 3, 5, 6, and 7, with parity bits in 1, 2, and 4.

Method 3: Error Detection and Correction

Syndrome Calculation: Parity bits are recalculated at the receiver's end to form a binary number (syndrome).

Error Localization:

Syndrome:

- Indicates the exact position of an error.
- If syndrome = 0, no error; if non-zero, the position it points to is incorrect.

Advantages of Hamming Codes

1. Single-Bit Error

- **Correction Capability:** Hamming codes are designed to correct single-bit errors, making them highly effective in environments where errors are typically minor.
- **Benefit:** This correction ability allows data integrity to be maintained without needing retransmission for minor errors, making them suitable for reliable systems.

2. Double-Bit Error Detection

- **Detection:** While Hamming codes can only correct single-bit errors, they can detect double-bit errors.
- **Application:** This detection capability adds another layer of reliability by flagging data that may need to be retransmitted if a double error is detected.

3. Low Redundancy

- **Efficiency:** Hamming codes achieve error correction with minimal additional bits, especially for shorter data lengths.
- **Advantage:** Compared to more complex error-correcting codes, Hamming codes add only a small amount of redundancy, which makes them bandwidth-efficient and ideal for storage and memory applications.

Conclusion

In summary, **Noise-Resistant Codes** are essential for maintaining data integrity in communication systems subject to noise. Among these codes, **Hamming codes** provide a straightforward yet powerful method for detecting and correcting single-bit errors, making them ideal for applications requiring basic error correction with minimal overhead.

Hamming codes stand out due to their **low redundancy, ease of implementation**, and **reliable performance** in correcting minor errors. These features make them widely applicable in memory systems, digital storage, and basic communication systems, where they contribute to data reliability without needing complex processing.

Looking forward, while Hamming codes are foundational, advancing technologies continue to drive the development of more complex codes that can handle multi-bit errors and meet the demands of high-throughput systems. Nonetheless, Hamming codes remain a practical choice for low-complexity, high-efficiency error correction, illustrating how fundamental coding principles continue to support reliable digital communication.