The binary block coding scheme

Coding Technology

Illés Horváth

2025/09/12

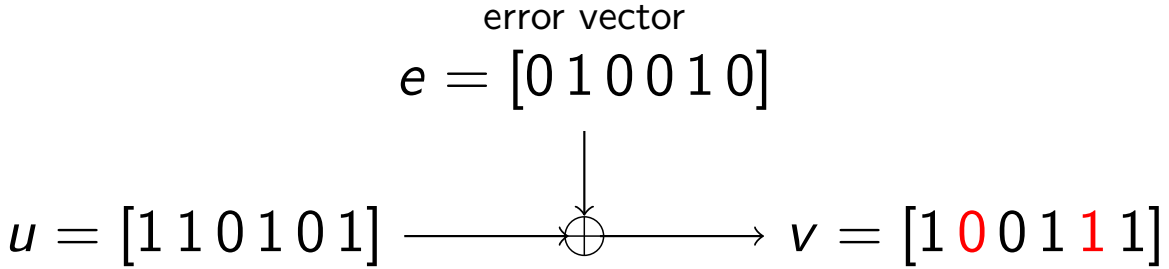
Binary Symmetric Channel

For the Binary Symmetric channel, each transmitted bit is flipped with probability P_b, independently from the input bit and other bits.

The bit flipping can be mathematically described as an additive error: if an input vector u is sent through the channel, the output vector is

$$v = u \oplus e,$$

where the e error vector contains 1's in positions which are flipped:



Error probability

If the error vector has length n and the bit error probability is P_b, then

$$P(w(e) = i) = \binom{n}{i} P_b^i (1 - P_b)^{n-i} = \binom{n}{i} \left(\frac{P_b}{1 - P_b}\right)^i (1 - P_b)^n.$$

P_b is typically small, and $P(w(e) = i)$ decreases fast in i.

Example. If $n = 3$, $P_b = 0.01$, then

i	0	1	2	3
$P(w(e) = i)$	0.970	0.0294	2.970×10^{-4}	10^{-6}

Error probability

Example. We want to transmit a single bit of information through a channel with $P_b = 0.01$. What is the probability that the bit will be received correctly? What if we use a $3\times$ repeater (repetition) code?

If we just transmit the single bit through the channel,

$$P(\text{received correctly}) = 1 - P_b = 0.99.$$

If we use a $3\times$ repeater code, then we transmit the original bit 3 times. The $3\times$ repeater code can correct 1 error, which means that if $w(e) \leq 1$, the receiver will be able to decode the original bit, so

$$P(\text{received correctly}) = P(w(e) \leq 1) = 0.970 + 0.0294 = 0.9994.$$

Error probability

The 'cost' of using the $3\times$ repeater code is that the channel capacity C is effectively reduced to $C/3$ (because we send 3 bits over the channel to transmit a 1-bit message).

This is a general tradeoff when using error correction codes: we can obtain higher probability of correct reception at the cost of reducing channel capacity.

(We will discuss more of this later, for specific codes.)

Block coding

For the repeater code, the original message is a single bit, transmitted several times through the channel.

What if the message to transmit consists of more than 1 bit?

We transmit it bit-by-bit, repeating each bit the same number of times.

We can make more efficient error correction codes if we code blocks of several bits simultaneously.

Block coding

The general setup for block coding is the following: for a $C(n, k)$ code, we first cut up the original message into sections of length k, then for each message section u, we apply a given map $\psi : \{0,1\}^k \to \{0,1\}^n$, and transmit

$$c = \psi(u)$$

through the channel.

Then the channel adds an error e, and the received vector on the other end is

$$v = c \oplus e$$

Based on v, the receiver will have to guess what c was sent through the channel using another function $\phi : \{0,1\}^n \to \{0,1\}^n$:

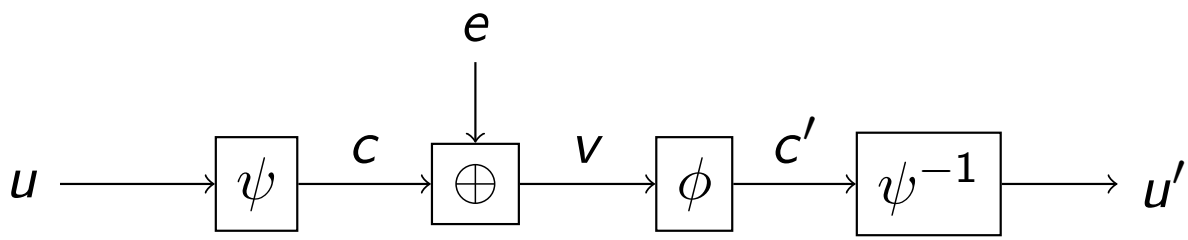
$$c' = \phi(v)$$

Finally, decoding is

$$u' = \psi^{-1}(c')$$

Then this process is repeated for further message sections.

Block coding



Glossary:

- u: message vector
- c: codeword or code vector
- e: error vector
- v: received vector
- c': detected codeword
- u': detected or decoded message

Remark. $n > k$ always holds for any error correction code.

Block coding

Example. For the $3\times$ repeater code, the ψ function is according to the following table:

u	c
0	000
1	111

After c goes through the channel, $v = c \oplus e$ is received on the output, but v can be any binary vector of length 3, and based on v, we have to make a guess as to what c could be (ϕ function).

For the repeater code, it is the majority decision: if there are more 1's than 0's in v, then $c' = \begin{bmatrix}1 & 1 & 1\end{bmatrix}$, and if there are more 0's than 1's in v, then $c' = \begin{bmatrix}0 & 0 & 0\end{bmatrix}$.

Block coding

How to guess the codeword in general?

Theorem

Assume all message vectors have equal probability. Then, for a given received vector v, the most likely codeword c is the one for which $d(c, v)$ is minimal.

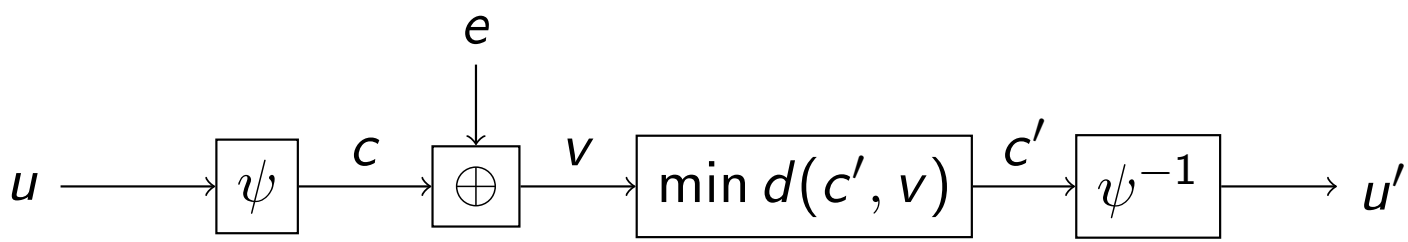
Proof. Bayes tells us that for any possible codeword c,

$$P(c|v) = \frac{P(c \text{ and } v)}{P(v)} = \frac{P(c \text{ and } e = c - v)}{P(v)} = \frac{P(c)P(e = c - v)}{P(v)}.$$

$P(c)$ is constant due to the assumption, $P(v)$ is the same for all c, and $P(e = c - v)$ is minimal when $w(c - v) = d(c, v)$ is minimal.

Block coding scheme

Based on that, we can now update the coding scheme:



How many errors can such a coding scheme detect and correct?

Error detection and correction capabilities

Theorem

Let $d_{\min}$ be the minimal Hamming-distance among codewords. Then the above coding scheme can...

- *detect $d_{\min} - 1$ errors, and*
- *correct $\lfloor \frac{d_{\min}-1}{2} \rfloor$ errors.*

Proof. Within the space $\{0,1\}^n$, the codewords are points such that the minimal distance between any two codewords is $d_{\min}$.

If we start from a codeword and change at most $d_{\min} - 1$ bits, we cannot reach any of the other codewords as they are just too far. So either we end up between two codewords (error detected), or stay at the original codeword (no errors).

Error correction: for each codeword c, consider the sets $\{u : d(c,u) \leq \lfloor \frac{d_{\min}-1}{2} \rfloor\}$ ('balls' of radius $\lfloor \frac{d_{\min}-1}{2} \rfloor$).

These balls are disjoint, so if we start from a codeword and change at most $\lfloor \frac{d_{\min}-1}{2} \rfloor$ bits, we stay within the same ball.

Example

Example. Design a C(5,2) code with $d_{\min} = 3$.

For a C(5,2) code, the codewords are $n = 5$ bits long.

The first codeword could be (00000).

The weight of every other codeword must be 3 or more. (00111) is a natural choice.

The next codeword also needs to have weight 3 or more, but it also has to differ from the previous codeword in at least 3 digits. (11100) is a suitable choice.

(11111) is not suitable for the final codeword, because its Hamming distance from the previous two codewords is just 2.But if we change the third bit to 0, it works: (11011).

$d_{\min} = 3$, so the code can detect $d_{\min} - 1 = 2$ errors and correct $\lfloor \frac{d_{\min}-1}{2} \rfloor = 1$ error.

Example

So the coding function is

ψ :

u	c
00	00000
01	00111
10	11100
11	11011

We receive the vector 10111. Which should be our guess?
The Hamming-distance from each of the 4 codewords is

$$d(10111, 00000) = 4,$$
$$d(10111, 00111) = 1,$$
$$d(10111, 11100) = 3,$$
$$d(10111, 11011) = 2.$$

Block coding

For ψ^{-1}, we can use the table for ψ, just reversed. Example:

ψ :

u	c
00	00000
01	00111
10	11100
11	11011

$\rightarrow$ ψ^{-1} :

c'	u'
00000	00
00111	01
11100	10
11011	11

We can make computing ψ^{-1} even easier with a little trick.

We say that a coding is systematic if the first k bits of each codeword c are the corresponding message u. The remaining $n-k$ bits are called parity bits.

For systematic codes, ψ^{-1} is simply truncation: to get u', we truncate c' and keep only the first k bits.

Is the above code systematic?

No, because for example $\psi(01) = 00111$.

Systematic codes

Can we change this code to make it systematic?

ψ :

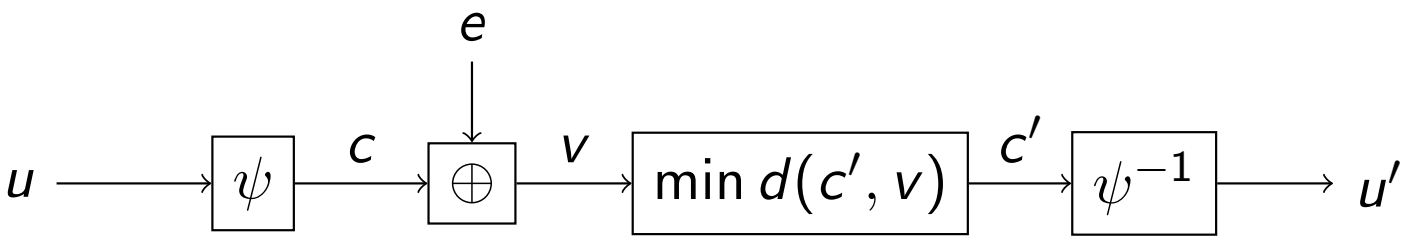
u	c
00	00000
01	00111
10	11100
11	11011

Yes, we can! In several ways, actually.

- We can rearrange how the codewords are assigned to the messages. This operation will not change any of the key parameters of the code (n, k, d_{min}).
- We can flip all bits in the same position in all of the codewords.
- We may permute the bits of the codewords (same permutation for all codewords).

Summary (so far)

The coding scheme:



For a $C(n, k)$ code, n is the codeword length, k is the message length. $n > k$ always holds.

The redundancy of a $C(n, k)$ code is the value $n - k$.

The code rate of a $C(n, k)$ code is $\frac{k}{n}$; this describes the reduction in channel capacity due to using an error correction code.

A code can detect $d_{\min} - 1$ errors and correct $\lfloor \frac{d_{\min}-1}{2} \rfloor$ errors, where $d_{\min}$ is the minimal Hamming-distance between codewords.

For a systematic code, ψ^{-1} is truncation.

Computational cost

What is the computational cost of the scheme?

On the transmitter side, there are two main approaches to computing the code $\psi(u)$:

- For small k, all $(u, \psi(u))$ pairs can be computed in advance and stored in a table (size 2×2^k).
- If ψ is easy to compute, then $\psi(u)$ can be computed online, even if k is large.

On the receiver side, computing $\min d(c', v)$ takes 2^k steps – as this is exponential in k, it is only feasible for relatively small values of k.

(Truncation for systematic codes is fast.)

Parity check bit

Parity check bit: we add a single parity check bit to a message of length k; the value of the bit is equal to the sum of all bits in the message.

This will result in a $C(k+1, k)$ code with $d_{\text{min}} = 2$, so the code can detect 1 error but cannot correct it.

Decoding the codeword is not possible for this code, but the error detection is still relevant for some applications:

- if resending the message is possible;
- another example is promotional codes, where we only want to check if a codeword is correct, but if not, correcting it is not required.

E.g. the TCP/IP protocol stack contains two error detection codes in different layers; one is a 16-bit parity checksum, the other is a CRC code. Neither is used for error correction; instead, erroneous packets are retransmitted.

Theoretical upper bounds

How good can a code be?

Generally, we would want $\frac{k}{n}$ to be as close to 1 as possible, while we want d_{min} to be as high as possible simultaneously.

A high d_{min} corresponds to the codewords being far away from each other (in Hamming-distance).

But the space $\{0,1\}^n$ is finite; it has 2^n points total, and the biggest distance between points is n.

If we want many codewords (high value of k), then some of them will be close to each other (low d_{min}), so there are limitations.

Singleton bound

Theorem (Singleton bound)

$$d_{\min} \leq n - k + 1.$$

Proof. Take all 2^k codewords, and delete the last $d_{\min} - 1$ bits from each codeword. The resulting truncated codewords must still be all different since originally their Hamming-distance was at least $d_{\min}$.

The 2^k truncated codewords have length $n - d_{\min} + 1$, and there are $2^{n-d_{\min}+1}$ different vectors in $\{0,1\}^{n-d_{\min}+1}$, so

$$2^k \leq 2^{n-d_{\min}+1}$$

must hold, and $d_{\min} \leq n - k + 1$ is obtained by rearranging.

Singleton bound

According to the Singleton bound, $d_{\min} \leq n - k + 1$ always holds.

Codes where $d_{\min} = n - k + 1$ holds with equality are called Maximum Distance Separable (MDS) codes.

Is the previous $C(5, 2)$ code MDS?

$$d_{\min} = 3 < 5 - 2 + 1 = 4,$$

so no, it is not an MDS code.

Are repeater codes MDS?

For the $n\times$ repeater code, $k = 1, d_{\min} = n$, and

$$d_{\min} = n = n - k + 1 = n - 1 + 1 = n,$$

so yes, repeater codes are MDS.

Hamming bound

Theorem (Hamming bound)

Assume a $C(n,k)$ code can correct $t = \lfloor \frac{d_{\min}-1}{2} \rfloor$ errors. Then

$$\sum_{i=0}^{t} \binom{n}{i} \leq 2^{n-k}.$$

Proof. As before, consider all 2^k codewords and balls of radius t around them.

Each ball contains $\sum_{i=0}^{t} \binom{n}{i}$ points, and the balls must be disjoint, so they contain $2^k \sum_{i=0}^{t} \binom{n}{i}$ points total, while there are 2^n points in the entire space $\{0,1\}^n$, so

$$2^k \sum_{i=0}^{t} \binom{n}{i} \leq 2^n$$

must hold.

Hamming bound

Codes where the Hamming bound has equality are called perfect codes. (As the balls cover the entire space perfectly.)

Is the earlier $C(5,2)$ code with $t = 1$ perfect?

$$\sum_{i=0}^{1} \binom{5}{i} = 1 + 5 = 6 < 2^{5-2} = 8,$$

so it is not a perfect code.

Is the $5\times$ repeater code perfect?

For the $5\times$ repeater code, $n = 5, k = 1, t = \lfloor \frac{n-1}{2} \rfloor = 2$, and

$$\sum_{i=0}^{2} \binom{5}{i} = 1 + 5 + 10 = 2^{5-1} = 16,$$

so it is a perfect code. (Actually, the $n\times$ repeater code is perfect for every odd value of n.)