The Chronicles of Alice and Bob

Navin Kashyap

Queen’s University
Introducing Alice and Bob

- Two people, partners in crime, in whose mysterious lives coding theorists show an inordinate amount of interest.
- Alice must communicate to Bob an important piece of information.
- Alice is concerned that the information may be intercepted by her enemies.
- Alice is short on time and money. She needs to send her information across cheaply and quickly.
- The communications channel she uses is noisy and unreliable. She needs to ensure that the information she sends is received correctly and reliably by Bob.
Coding Theory to the Rescue

What is Coding?

The representation of information using symbols, often 0’s and 1’s.

Comes in three flavours —

Secrecy Coding (aka Cryptography): Enables Alice to encrypt her message to foil her enemies.

Source Coding: Enables Alice to compress her message to save on cost of transmission.

Channel Coding: Enables Alice to send her message reliably to Bob, by introducing some mechanism to counter channel noise.
Channel noise follows a model known to both Alice and Bob, who intend to fully make use of this knowledge to encode and decode.

Noise may be deterministic or random; if random, it follows a probabilistic model.

The effect of noise is to introduce errors in the transmitted message.

The goal of channel coding is to reduce or eliminate the effects of channel noise.
Three Types of Channel Codes

Error-Detecting Codes: Allows Bob to detect errors in received message; useful in random noise situations.

Error-Correcting Codes: Allows Bob to correct errors in received message; useful in random noise situations.

Constrained Codes: Alice encodes the message in such a way as to prevent errors from corrupting the message; useful in deterministic noise situations.

All coding schemes work by adding redundancy to the message to compensate for errors: more symbols are transmitted than are in the original message.
A Simple Error-Detecting Code

Channel Noise: Flips bits at random (0 → 1, 1 → 0).

Alice’s Message: 7-bit binary sequence \((b_1, b_2, b_3, b_4, b_5, b_6, b_7)\)

Encoder: Adds 8th bit, called a parity bit, \(b_8\), such that

\[
\sum_{i=1}^{8} b_i \equiv 0 \pmod{2}
\]

This code detects an odd number of errors.
Applications of Error-Detecting Codes

Computer network communication protocols such as TCP/IP use error-detecting codes extensively to determine if data packets sent across a network have been corrupted or not.

Usual remedy for receipt of a corrupted data packet is to request retransmission.
Applications of Error-Detecting Codes


ISBN numbers are 10 symbols in length, and of the form x-xxx-xxxxx-x. The symbols used are \{0,1,2,\ldots,9,X\}, where X represents the number 10.


The tenth symbol is a check symbol.

Example : 0-444-85193-3.

\[
1 \times 0 \ + \ 2 \times 4 \ + \ 3 \times 4 \ + \ 4 \times 4 \ + \ 5 \times 8 \ + \\
6 \times 5 \ + \ 7 \times 1 \ + \ 8 \times 9 \ + \ 9 \times 3 \ + \ 10 \times 3 \ = \ 242
\]

\[
\equiv 0 \pmod{11}
\]
A Simple Error-Correcting Code

Channel Noise: Flips bits at random ($0 \rightarrow 1$, $1 \rightarrow 0$).

Alice’s Message: Single bit, $b = 0$ or $1$.

Encoder: $0 \rightarrow 000$, $1 \rightarrow 111$.

Decoder: Majority rule — if more 0’s than 1’s received, then decode as 0; else decode as 1.

This code can correct single errors, and detect double errors.
Repetition Codes

Encoder (Alice): Repeats each message bit \( r \) times.

Decoder (Bob): Uses majority rule for each block of \( r \) received bits.

Coding Rate: \( R = 1/r \)

(1 message bit per \( r \) coded bits).

Error-Correction Capability: Up to \((r - 1)/2\) errors in each block.

Error-Detection Capability: Up to \( r - 1 \) errors in each block.
Repetition Codes: A Probabilistic Analysis

Channel Model: Each bit gets flipped with probability $p$, independent of other bits.

The probability that a single message bit is received correctly is

$$P_{b,C} = \sum_{j=0}^{\left\lfloor \frac{r-1}{2} \right\rfloor} \binom{r}{j} p^j (1 - p)^{r-j}$$

So, the probability that entire $k$-bit message is received correctly is

$$P_C = (P_{b,C})^k$$
Repetition Codes: A Probabilistic Analysis

Say $p = 0.1$ is the probability of a channel error.

Alice needs to be 99.9% sure that her $k$-bit message will be received correctly by Bob, i.e.,

$$P_C = (P_{b,C})^k \geq 0.999.$$

- $k = 1$ needs $r = 7$ repetitions: $R = 1/7$.
- $k = 2$ needs $r = 11$ repetitions: $R = 1/11$. 
**Shannon’s Noisy Channel Coding Theorem**

**Channel Model:** Each bit gets flipped with probability $p$, independent of other bits.

**Channel Capacity:** $C(p) = 1 + p \log_2 p + (1 - p) \log_2 (1 - p)$.

**Theorem [Claude Shannon (1948)]:**

For any $\epsilon > 0$, if $R < C(p)$ and $n$ is sufficiently large, there exists a code which can send $k = nR$ message bits across the channel in such a way that all $k$ bits are received correctly with probability greater than $1 - \epsilon$.

- Note that $C(0.1) = 0.5310$. 


Hamming Codes

In 1950, Richard Hamming discovered a family of single-error-correcting codes, parametrized by an integer \( r \).

**Codeword length**: \( n = 2^r - 1 \).

**# Message bits encoded**: \( k = 2^r - 1 - r \).

**Rate**: \( R = \frac{k}{n} = \frac{2^r - 1 - r}{2^r - 1} \).
The [7,4] Hamming Code

Parity-check matrix:

\[
H = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 \\
\end{bmatrix}
\]

(the \(i\)th column of \(H\) is the binary representation of the integer \(i\))

The Hamming code

\[
C = \{ \mathbf{x} \in \{0, 1\}^7 : H \mathbf{x}^T = 0 \}
\]

is the nullspace of the matrix \(H\), hence has dimension \(7 - 3 = 4\).

Thus, \(C\) has \(2^4 = 16\) codewords, of length 7 each; rate \(R = 4/7\).
The [7,4] Hamming Code

\[
H = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}
\]

**Encoder (Alice):** Bijection between the binary message words of length 4 and the Hamming codewords of length 7.

**Channel Noise:** Flips bits at random (0 → 1, 1 → 0).

Suppose that Alice sends 1000011, but Bob receives 1000001:

\[
1000011 \longrightarrow 10000001
\]

Set \( y = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1] \), and compute \( s = Hy^T \).
The [7,4] Hamming Code

\[ 1 0 0 0 0 1 1 \longrightarrow 1 0 0 0 0 0 1 \]

\[ s = H y^T = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \]

\[ s = [1 \ 1 \ 0]^T \] tells Bob that the 6th bit has been flipped.
Aside: A Hat Trick

Team of three people.

A hat, red or blue, is placed randomly on each person’s head.

Each person can see the others’ hats, but not his/her own.

The team members are asked to write down the colour of their own hats on separate slips of paper.

They are allowed to write ‘red’, ‘blue’ or ‘don’t know’.

If at least one of them gives the right answer, AND none of them gives the wrong answer, the team wins; else the team loses.

What is their best strategy?
The Best Strategy

Each person acts as follows:

if the hats he/she sees are of the same colour,
then he/she writes down the name of the other colour.

if the hats he/she sees are of different colours,
then he/she puts down ‘don’t know’.

This strategy gives the team a 75% chance of winning.

Now, find the best strategy for a seven-person team.

[Hint: This involves the [7,4] Hamming code,
and gives the team a 7/8 chance of winning.]
Other Families of Error-Correcting Codes

Algebraic Codes

BCH and Reed-Solomon codes
Reed-Muller code
Golay code
Algebraic geometry codes, such as Goppa codes.

Sparse Graph Codes

Convolutional codes
Turbo codes
Low-density parity-check (LDPC) codes
Applications of Error-Correcting Codes

Deep-Space Communications

The North polar cap of Mars, taken by Mariner 9 in 1972. (Source: NASA.)

The NASA Mariner probes (1969–1973) used a powerful Reed-Muller code capable of correcting 7 errors out of 32 bits transmitted, consisting of 6 data bits and 26 parity check bits.
Applications of Error-Correcting Codes

Compact Discs

To guard against scratches, cracks and similar damage, CD’s use Cross-Interleaved Reed-Solomon Coding (CIRC), which involves a (28,24) Reed-Solomon code and a (32,28) Reed-Solomon code, separated by a 28-way convolutional interleaver.

Can correct error bursts of up to 4000 bits (∼ 2.5mm of track).
While on the Topic of CD’s . . .

On a CD, 1’s and 0’s are stored as pits and lands.

Each high area of the CD (viewed from the polycarbonate side) is called a pit and the flat section is called a land.

These pits are about 0.5 microns wide, and the tracks in which the pits lie are 1.6 microns apart.
How a CD Player Works

A CD drive consists of three fundamental components:

A **drive motor** to spin the disk. This drive motor is precisely controlled to rotate between 200 and 500 RPMs, depending on which track is currently being read.

A **laser** and a **lens system** to focus in on the pits and lands and read them.

A **tracking mechanism** that can move the laser assembly so that the laser’s beam can follow the spiral track.
How a CD Player Works

The fundamental job of the CD player is to **focus the laser** on the track of pits.

The laser beam passes through the polycarbonate layer, reflects off the aluminum layer and returns to an **opto-electronic device** that detects changes in light.

The pits reflect light differently than the lands, and the opto-electronic sensor can detect that change in reflectivity.

The electronics in the drive interpret the changes in reflectivity to read the bits.
Because the laser is tracking the spiral of data using the pits, there cannot be extended gaps in the data track where there are no pits.

To solve this problem, data is encoded using a constrained code (aka modulation code) in which codewords do not contain long runs of zeros.
The “No Adjacent Zeros” Constraint

Consider the constrained system of binary sequences none of which contain a pair of adjacent zeros.

An example of such a sequence is

\[ 1110110111111101 \]

**Encoder (Alice):** Insert (“stuff”) a ‘1’ after every ‘0’ in the message (data) sequence.

\[ 010001101 \rightarrow 01101010111011 \]

**Decoder (Bob):** Remove (“destuff”) the first ‘1’ after every ‘0’ in the received sequence.

**Rate:** Average code rate of 2/3.

(on an average, one “stuffed” bit for each pair of message bits)
Capacity of a Constrained System

Theorem [Claude Shannon (1948)]:

Given a constrained system, \( S \), there exists a real number, \( C(S) \), such that

(i) for any \( R < C(S) \), there exists a code of rate \( R \), which converts unconstrained binary sequences to constrained sequences in \( S \); and

(ii) there exists no such code with rate \( R > C(S) \).

The number \( C(S) \) is called the capacity of \( S \). It can be computed as

\[
C(S) = \lim_{n \to \infty} \frac{\log_2 q_n(S)}{n}
\]

where \( q_n(S) \) is the number of sequences of length \( n \) in \( S \).
Capacity

The “no adjacent zeros” constrained system has a capacity of

$$\log_2 \left( \frac{1 + \sqrt{5}}{2} \right) = 0.6942$$

.
Some Fun Facts

Let $f_n$ be the number of sequences of length $n$ that do not contain a pair of adjacent zeros.

The first few numbers in this sequence are:

$$1, 2, 3, 5, 8, 13, 21, 34, 55, 89, \ldots$$

This is the famous Fibonacci sequence generated by the recursion:

$$f_n = f_{n-1} + f_{n-2}$$
Some Fun Facts

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

The Fibonacci representation of an integer:

\[ 30 = 1 \times 21 + 0 \times 13 + 1 \times 8 + 0 \times 5 + 0 \times 3 + 0 \times 2 + 1 \times 1 \]

\[ 30 = (1\,0\,1\,0\,0\,0\,1)_F \]

This always results in a sequence with no adjacent 1’s.

This yields a nice way of converting between unconstrained binary sequences and sequences satisfying the “no adjacent zeros” constraint:

\[(1\,1\,1\,1\,0) \rightarrow (1\,1\,1\,1\,0)_2 \rightarrow 30 \rightarrow (1\,0\,1\,0\,0\,1)_F \rightarrow (0\,1\,0\,1\,1\,1\,0)\]
Other Families of Constrained Codes

\((d, k)\)-RLL constraints: every pair of successive 1’s must have at least \(d\) and at most \(k\) 0’s between them.

For example, the (0,1)-RLL constraint is simply the “no adjacent zeros” constraint.

Forbidden subsequence constraints: no sequence can contain as a subsequence any of a pre-specified forbidden list of patterns.

For example, the “no adjacent zeros” constraint is obtained by forbidding the pattern \texttt{00}.

Spectral null constraints: The discrete Fourier transform of each sequence must have nulls at pre-specified frequencies.
Applications of Constrained Codes

**Digital data storage:** CD’s, DVD’s, magnetic tapes, magneto-optic devices.

(0,1)-RLL, (1,3)-RLL and (2,7)-RLL constrained codes are in various magnetic recording industry standards.

**High-rate optical communications:** Special constrained codes, called *ghostbusting codes*, are being developed to combat inter-symbol interference effects in very high bitrate communications over an optical fibre.

**DNA computing:** Constrained codes to prevent “bad” secondary structures of DNA strands used in DNA computing.
To Learn More About Codes

Take MATH 478