Coconut: Lecture notes

Josh Felmeden

November 26, 2019

Contents

1 Hamming Codes

For any $m \geq 1$, the generalised Hamming code encodes up to $m - 1$ data bits with m parity bits for a total of $2^m - 1$ respectively bits.

- The minimum distance $d = 3$ (respectively 4 for SECDED) for any value of m
- The i-th bit is a parity bit if i is a power of 2, otherwise a data bit.
- The parity bits are set such that the sum of all bits whose index i has a 1 at the j-th position when written out in binary equals 0.
- For the SECDED version, an additional initial bit 0 is set such that the sum of all the bits in the codeword is zero.

The parity bits in the generalized scheme is computed such that all of the 1s in the rows must sum to zero. For example:

0	1	2	3	4	5	6	7
0	0	0	0	1	1	1	1
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
Bit name	P_1	P_2	m_3	P_4	m_5	m_6	m_7

So, $P_4 + m_5 + m_6 + m_7 = 0$ where P_4 is a parity bit, and the rest are data bits.

We use the following rules:

1. $P_4 + m_5 + m_6 + m_7 = 0$
2. $P_2 + m_3 + m_6 + m_7 = 0$
3. $P_1 + m_3 + m_5 + m_7 = 0$

Let's revisit the table. If we have the sequence:

0	1	2	3	4	5	6	7
Ex 1	0	1	1	0	0	0	1
Bit name	P_1	P_2	m_3	P_4	m_5	m_6	m_7

Then, rules **1** and **2** are violated, but **3** is not. So, with the GHC decoding rule, we sum the indexes of violating parity bits. Since we know that P_1s bits are okay, but P_2, P_4 are not, we can therefore tell that m_6 is the issue, so if we flip this, then we'll probably be okay.

1.1 Reed-Solomon codes

The idea behind these boys are that we have something called the singleton bound. Now, if we have an encoding function $E : \sigma^k \to \sigma^n$, the singleton d is going to be $d \leq n-k+1$. So this is the singleton bound. A code with $d = n - k + 1$ is called the MDS (maximum distance separable).

The **Reed-Solomon** code is:

- Pick some points.
- Find the polynomial.
- Evaluate as much as you need it to be.

If we have a message that's not all 0, we are evaluating at n different values. Also, over a field, at most $k - 1$ values are zeroes. Therefore, at least $n - (k - 1)$ points in a non zero code word are nonzero.

2 Information Theory

Firstly, let's lay some ground rules, shall we?

- Σ = Alphabet (finite, non empty set)
- $X =$ Alphabet (with only 0s and 1s)
- $\Sigma^+ =$ Non empty words over Σ
- $\Sigma^* =$ Words over Σ including ϵ (empty)

We want to be able to send some message $\underline{m} \in \Sigma^+$ to some channel with some noise.

Here are some more definitions:

- $f : X \to y$ is injective if $x \neq x' \to f(x) \neq f(x)$ and $f(x) = f(x) \to x = x'$.
- $f : X \to y$ is surjective if every $y \leq y$ has some $x \in X$ such that $f(x) = y$.
- If both are true, then it is bijective.

Finally, Image$(f) = \{y \in Y | y = f(x)$ for some $x \in X\}$.

Now, let's look at an example:

σ	A	B	C	D
$f(\sigma)$	0	1	00	01

This is a problem because does $00 =$ c or AA? Instead, we could do something like:

σ	A	B	C
$f_1(\sigma)$	1	10	100

or

σ	A	B	C
$f_2(\sigma)$	1	01	001

Now, f_2 is actually better because we know when something ends by the 1. The first one leaves us waiting for additional 0s. We call the second function **prefix-free**.

Prefix free

An encoding E is prefix free if for all $\underline{m}, \underline{m}'$, $E(\underline{m})$ is not a prefix of $E(\underline{m}')$

Theorem: If E is injective and prefix-free, then E^+ is injective.

Let's now assume that $\underline{m} \neq \underline{m}'$ but $E^+(\underline{m}) = E^+(\underline{m}')$. We can now work backwards and prove that this is a contradiction:

- Assume $|\underline{m}| \leq |\underline{m}'| (w \log)$
- If $\underline{m}$ is prefix of $\underline{m}'$, then $\exists \underline{s} : \underline{m}' - \underline{ms}$
- $E(\underline{m}') = E^+(\underline{m})E^+(\underline{s}) = E^+(\underline{m}) \rightarrow E^+(\underline{s}) = \epsilon$
  - This is a contradiction, because we said that they should evaluate to the same thing before but now one is empty.