Lecture 14, Tuesday w8, 2014-11-04

Today's lecture was about block codes.

Last time, we saw how a checksum can verify the integrity of a block of data. If we have multiple checks, we can try to infer what went wrong: find the simplest or most probable correction that will make all of the checks happy again.

$[N, K]$ **block code:** Use the channel N times (or the Nth extension of the channel once) to send K bits of information.

Codewords: only S of the $|\mathcal{A}_X|^N$ possible input patterns are ever used. The definitions imply $K = \log S$ (you should know why!).

Rate: number of bits per channel use K/N or $(logS)/N$. (In some texts the log is taken to the base $|\mathcal{A}_X|$, but not by this course, or MacKay.)

Example: Repetition codes have $|\mathcal{A}_X|$ codewords. What's their rate?

Example: Hamming codes correct one bit flip error in a block of N bits. Usually we mean the Hamming $[7, 4]$ code. The Hamming $[3, 1]$ code is just the repetition R_3 code, and the other codes ($[15, 11]$, $[31, 26]$, ...) are rarely used (why?).

Noisy channel coding theorem: Let C be the *capacity*, of the original (non-extended) channel. There exists an $[N, K]$ block code that can communicate at any error probability of your choosing $\epsilon > 0$, at a rate $R < C$. If you communicate at rates beyond the capacity, $R > C$, there is some minimum error probability, meaning you will eventually see errors, no matter how clever your error correcting code.

Example: we argued that the BSC cannot be used to communicate at rates $R > 1 - H_2(f)$. If we could, the argument showed a way to split up a file of $K + NH_2(f)$ bits into two parts, and encode them in a way that uses fewer bits than their information content. There's a similar argument in the notes for the BEC.

Next time, an example of how to achieve communication at rates close to the capacity for the BEC.

Check your progress

We're up to slide 18 in the 'week 7' slides. Mark anything that's unclear or that needs expanding on NB.

A Hamming code will give an incorrect decoding if there are two bit flips. Could we add an extra parity bit to warn of this situation? What if that bit was flipped?

Imagine someone fooled themselves into thinking they had an $[N, K]$ block code for the BSC that allowed reliable communication at rates greater than the capacity. How in detail could they split up a file into K bits that look like uniform

noise, and N bits that looked like independent Bernoulli draws with probability f, in an attempt to create a compressor that can compress any file?

Recommended reading

If you haven't already, read MacKay pp9–15, and Chapter 9.