

Explained: The Shannon limit

January 19 2010, by Larry Hardesty



Claude Shannon's clever electromechanical mouse, which he called Theseus, was one of the earliest attempts to teach a machine to learn and one of the first experiments in artificial intelligence. Photo: Bell Labs

It's the early 1980s, and you're an equipment manufacturer for the fledgling personal-computer market. For years, modems that send data over the telephone lines have been stuck at a maximum rate of 9.6 kilobits per second: if you try to increase the rate, an intolerable number of errors creeps into the data.

Then a group of engineers demonstrates that newly devised error-correcting codes can boost a modem's transmission rate by 25 percent. You scent a business opportunity. Are there codes that can drive the data rate even higher? If so, how much higher? And what are those codes?

In fact, by the early 1980s, the answers to the first two questions were more than 30 years old. They'd been supplied in 1948 by Claude Shannon SM '37, PhD '40 in a groundbreaking paper that essentially created the discipline of [information theory](#). "People who know Shannon's work throughout science think it's just one of the most brilliant things they've ever seen," says David Forney, an adjunct professor in MIT's Laboratory for Information and Decision Systems.

Shannon, who taught at MIT from 1956 until his retirement in 1978, showed that any communications channel — a telephone line, a radio band, a fiber-optic cable — could be characterized by two factors: bandwidth and noise. Bandwidth is the range of electronic, optical or electromagnetic frequencies that can be used to transmit a signal; noise is anything that can disturb that signal.

Given a channel with particular [bandwidth](#) and noise characteristics, Shannon showed how to calculate the maximum rate at which data can be sent over it with zero error. He called that rate the channel capacity, but today, it's just as often called the Shannon limit.

In a noisy channel, the only way to achieve zero error is to add some redundancy to a transmission. For instance, if you were trying to transmit a message with only three bits, like 001, you could send it three times: 001001001. If an error crept in, and the receiver received 001011001 instead, she could be reasonably sure that the correct string was 001.

Any such method of adding extra information to a message so that errors can be corrected is referred to as an error-correcting [code](#). The noisier the channel, the more information you need to add to compensate for errors. As codes get longer, however, the transmission rate goes down: you need more bits to send the same fundamental message. So the ideal code would minimize the number of extra bits while maximizing the

chance of correcting error.

By that standard, sending a message three times is actually a terrible code. It cuts the data [transmission rate](#) by two-thirds, since it requires three times as many bits per message, but it's still very vulnerable to error: two errors in the right places would make the original message unrecoverable.

But Shannon knew that better error-correcting codes were possible. In fact, he was able to prove that for any communications channel, there must be an error-correcting code that enables transmissions to approach the Shannon limit.

His proof, however, didn't explain how to construct such a code. Instead, it relied on probabilities. Say you want to send a single four-bit message over a noisy channel. There are 16 possible four-bit [messages](#). Shannon's proof would assign each of them its own randomly selected code — basically, its own serial number.

Consider the case in which the channel is noisy enough that a four-bit message requires an eight-bit code. The receiver, like the sender, would have a codebook that correlates the 16 possible four-bit messages with 16 eight-bit codes. Since there are 256 possible sequences of eight bits, there are at least 240 that don't appear in the codebook. If the receiver receives one of those 240 sequences, she knows that an error has crept into the data. But of the 16 permitted codes, there's likely to be only one that best fits the received sequence — that differs, say, by only a digit.

Shannon showed that, statistically, if you consider all possible assignments of random codes to messages, there must be at least one that approaches the Shannon limit. The longer the code, the closer you can get: eight-bit codes for four-bit messages wouldn't actually get you very close, but two-thousand-bit codes for thousand-bit messages could.

Of course, the coding scheme Shannon described is totally impractical: a codebook with a separate, randomly assigned code for every possible thousand-bit message wouldn't begin to fit on all the hard drives of all the computers in the world. But Shannon's proof held out the tantalizing possibility that, since capacity-approaching codes must exist, there might be a more efficient way to find them.

The quest for such a code lasted until the 1990s. But that's only because the best-performing code that we now know of, which was invented at MIT, was ignored for more than 30 years. That, however, is a story for the next installment of Explained.

Second part: [Explained: Gallager codes.](#)

Provided by Massachusetts Institute of Technology

Citation: Explained: The Shannon limit (2010, January 19) retrieved 9 April 2024 from <https://phys.org/news/2010-01-shannon-limit.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.
