Information Theory
Paul Trow

The information revolution is one of the most important developments in human history. It is to twentieth century what the industrial revolution was to the nineteenth. Information in all its many forms, including writing, voice, music, pictures and data, can now be sent from one place to another almost instantaneously. Volumes can be stored on a pocket-sized disk and rapidly searched for key words or phrases. And, of course, the World Wide Web links a myriad of sources of information in a global network. An essential ingredient in this revolution has been a branch of mathematics called information theory, which is used to encode and transmit data from one place to another.

The field of information theory was created by the mathematician Claude Shannon, working at Bell Laboratories, in a paper published in 1948, entitled "A Mathematical Theory of Communication." Shannon considered the general problem in which a source of information is to be encoded, sent over a channel, and finally decoded by the recipient. To the mathematician, information means anything that can be stored on a computer as strings of 0's and 1's, called bits. It may be music on a compact disk, an e-mail message to a friend, or simply random bits of data. The channel may be a radio signal from the ground to a satellite, or it may be the hard drive on a computer.

Whenever information passes through a channel, errors can occur in transmission. For example, they might be caused by physical imperfections on a disk or by sunspots interfering with radio signals. One of the goals of information theory is to encode data so that errors which occur in transmission can be corrected by the receiver. Such encoding occurs every time you make a telephone call or save a file on your computer. The mathematical codes that make this possible are called *error-correcting codes*. If you scratch a compact disk, error-correcting codes prevent the scratch from adding unwanted sounds to the music. They also enable spacecraft to send accurate pictures back to Earth despite errors that are caused by the Sun's radiation.

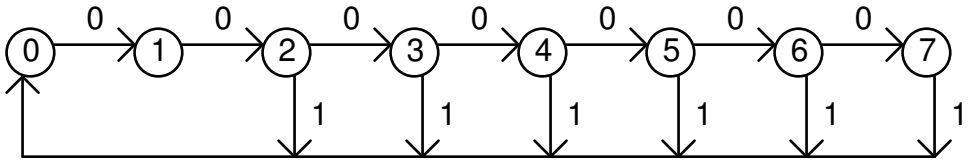
To get an idea of how such codes work, consider a simple scheme for error correction. Whenever you want to send a 0, you send a code word of five 0's, and when you want to send a 1, you send five 1's. This repetition is called redundancy. The English language has a built in redundancy, which enables the listener to understand the meaning of a sentence even when he has missed some of the words. When the receiver gets the message, each code word of five bits is checked. If an error has been introduced, some code word will contain both 0's and 1's. To correct the received code word, the computer counts the number of 0's and the number of 1's, and takes whichever has a majority as the received bit. This method is not perfect, since if three or more errors occur in a code word, then it will be incorrectly decoded. But if the probability of an error occurring in an individual bit is small, then it is very likely that a bit will be received correctly. Furthermore, if the length of a code word is increased from five to a larger odd number, then the probability of an error can be made as small as desired.

This method has the obvious disadvantage that to achieve greater accuracy, many bits must be sent for each bit received. This can slow the rate of transmission of the message and increase the amount of space required to store it. In our example, since a five-bit code word is sent for each bit of source message, the rate of transmission is 1:5, or $\frac{1}{5}$. In general, the rate is p:q, or $\frac{p}{q}$, if q bits are sent for every p bits of source message. In designing a code, there are several important factors to consider: it should have a high rate of transmission, a low probability of receiving errors, and it should not be too complex to implement. Improving one of these properties may require giving up something for the others. Furthermore, certain applications may require that greater weight be placed on one property than the others. Balancing these considerations is one of the challenges of information theory.

In his groundbreaking paper of 1948, Shannon proved a very important theorem, called the Noisy Coding Theorem, which relates these properties. It says roughly that for every channel, there is a number C, called the capacity of the channel, with the property that if the rate of transmission is slightly less than C, then there is a code that reduces the probability of an error in decoding to a value as small as you wish. This remarkable theorem, which is the theoretical basis for information theory, gives researchers the confidence to look for codes with low error rates. While Shannon's theorem is of great theoretical importance, it does not explicitly give codes which can be used in practical applications - it just guarantees that they exist. Actually creating the codes which are used in new technology has been the work of many mathematicians and engineers over the fifty years since Shannon's original paper.

How does mathematics contribute to the design of error-correcting codes? It turns out that a variety of algebraic structures, most of which were originally studied without any applications in mind, can be used to design codes. For example, one of the earliest classes of codes, called Hamming codes (after the mathematician Richard Hamming, one of the pioneers of information theory, who died earlier this year), are vector spaces over finite fields. These are analogous to the usual n-dimensional Euclidean spaces familiar to calculus students, except that they have a finite number of elements.
The simplest example of a finite field is arithmetic modulo 2, in which 0+0=0 and 0+1=1, but 1+1=0. A binary Hamming code is a collection of vectors $(x_1, x_2, ..., x_n)$, where each x_i is 0 or 1, which is closed under coordinate-wise addition and multiplication by the numbers 0 and 1. Encoding a message is carried out by multiplying source vectors by a matrix, where all arithmetic is done modulo 2. Decoding can be done by the minimum distance method. The Hamming distance between two vectors is just the number of positions in which they disagree. For example, the distance between (0,1,1,0) and (0,1,1,1) is 1, since they disagree only in the last position. When an error is introduced into a code word, the received word can be decoded by choosing the code word that is closest to it. Hamming codes are superior to the simple repetition codes, described previously, in that they give higher rates of transmission with lower probability of error.

Besides error correction, information theory also addresses problems of data storage. These include encoding data as compactly as possible, and coding issues related to the physical nature of the storage medium. For example, when data is recorded on magnetic disks, such as the hard drive in a computer, it is stored as magnetic charges, positive or negative. As the disk spins rapidly, data is read or written to the disk by a head. A change in polarity, which causes a voltage pulse, represents a 1, while no change represents a 0. If two 1's are stored next to each other, the pulses may interfere with each other, preventing the read head from detecting the changes in polarity. So it is desirable to separate 1's by a few 0's. Another coding consideration is that each computer has a kind of internal clock which keeps track of where the head is on the disk. This clock is kept synchronized by occurrences of 1's. If there is too long a sequence of 0's, the clock may lose synchronization, causing errors in the data read. These considerations lead to so called run-length constraints. A (d,k) constraint means that data is stored so that between every two 1's, there must be at least d 0's, and so that one never sees a sequence of more than k consecutive 0's. As an example, the (2,7) constraint can be represented by the graph below.



The vertices are labeled 0 through 7. A path in the graph is a sequence of the vertices, which follows the directions of the arrows on the edges - for example, 12340. Observe that if you follow any path in the graph, and read off the corresponding sequence of 0's and 1's from the arrows, they will obey the (2,7) constraint. You can think of the source as being arbitrary sequences of bits, and the channel as being those sequences which obey the (2,7) constraint. The problem for the coding theorist is to encode arbitrary data so that it obeys the (2,7) constraint. It turns out to be impossible to construct a code which does this at a rate of 1:1.

The reason for this has to do with a quantity called entropy, which measures the amount of information that can be stored in a system. Entropy is closely related to Shannon's channel capacity (and in fact is the same for noiseless channels). A system is a collection of sequences of symbols obeying some constraint, such as the (2,7) constraint. If B_n denotes the number of sequences of symbols of length n which can occur in the system, then the entropy of a system is defined to be $\lim_{n\to\infty} \frac{1}{n} \log_2 B_n$. For example, for arbitrary source data (in which there are no constraints), there are 2^n possible sequences of length n, so the entropy is $\frac{1}{n} \log_2 2^n = 1$ (the limit not being necessary in this case). On the other hand, for the (2,7) system, the entropy is $\log_2 \lambda$ where λ is the largest positive eigenvalue of the transition matrix of the graph. The transition matrix in this case is an 8×8 matrix, indexed by the vertices of the graph, having a 1 in position (i,j) if there is an edge from

vertex i to vertex j, and 0 otherwise. The number $\log_2 \lambda$ can be calculated, and is approximately equal to .52, which is less than 1. Since the entropy of the channel is less than the entropy of the source, it is not possible to encode source data into the (2,7) system at a rate 1:1.

However it is possible to encode at a rate 1:2. By using two bits of channel for each bit of source, you are taking the "square" of the (2,7) system - that is, taking it two bits at a time - which doubles the entropy. Since the entropy of the squared system is greater than 1, it is possible to encode at this rate. In general, it is possible to encode at a rate p:q provided that $\frac{p}{q} < \log_2 \lambda$. To achieve the highest rate of transmission, you would like $\frac{p}{q}$ to be as close to $\log_2 \lambda$ as possible, which you can do by finding good rational approximations to $\log_2 \lambda$. However, to do so you must make p and q larger, which increases the complexity of the code.

Entropy tells us when it is possible to encode at a certain rate, but it does not give actual codes. To construct codes satisfying various run-length constraints, mathematicians at I.B.M. Research have applied a branch of mathematics called symbolic dynamics, which deals with systems consisting of infinite sequences of symbols, such as the (2,7) system. Although symbolic dynamics was originally created to study purely mathematical problems in dynamical systems, long before the invention of computers or magnetic disks, its techniques have turned out to be surprisingly well-suited to applied coding problems.

Information theory provides an excellent example of how branches of pure mathematics, such as algebra and dynamical systems, have been applied to problems in the real world. As the information revolution progresses, information theorists will continue to invent the codes necessary for accurate and rapid communication.