

Information Theory

Rudy Koot

April 20, 2006

Contents

1	Introduction	2
1.1	Overview	3
2	Information	4
2.1	Self-information	4
2.2	Entropy	6
2.3	Conditional Self-information and Entropy	7
3	Coding	8
3.1	Noiseless coding theorem	9
4	Channels	10
4.1	Noise	10
4.2	Error-correcting codes	11
5	Conclusion	13

Chapter 1

Introduction

Information theory is the study of the mathematical properties of communication systems and the messages transmitted in them. It allows us to answer such questions as: “How do we measure information? How much information can we send through an information carrier or channel? Is it possible to reliably send messages over an unreliable, or noisy, channel?” and “What are the limits of data compression?”

The roots of information theory can be traced back to the end of the 19th century, when several scientists became interested in the transmission of messages through telephone and telegraph wires, devices invented several years earlier. In 1924, Harry Nyquist published his paper *Certain Factors Affecting Telegraph Speed* and in 1928, Ralph V. L. Hartley published his paper *Transmission of Information*. Both were employees of Bell Laboratories. While most scientists, until that time, had only been interested in question which could directly applied Nyquist and Hartley started looking at communication from a mathematical perspective.

Information theory, as is it stands today, was defined by Claude Elwood Shannon (1916–2001), another employee of Bell Laboratories, in his paper *A Mathematical Theory of Communication* (see Shannon [1948]) Shannon based his theory on the model of a communication system as depicted in Figure 1.1. The model contained six elements. A *message source* and a *message receiver* at opposite sides, with a *channel* in the middle connecting them. The message source is the person or object which produces and sends the messages. The message receiver the person or object which receives and consumes them. If our communication system would be a telegraph system, the message source would be the person operating the telegraph, the telegraph wires the channel, and the person reading the message from the strip of paper at the the other end of the wire the message receiver. Less conventional, we could see any file on a computer system as a message source and any program reading such a file as a message receiver. In this case the operating system could be interpreted as the channel.

In information theory two types of channels are distinguished: the *discrete channel* and the *continuous channel*. Over a discrete channel we can send only a finite number of symbols, such as the letters of the alphabet, while we can send any arbitrary waveform over a continuous channel. The telegraph and most digital systems use a discrete channel, while the telephone and other analog

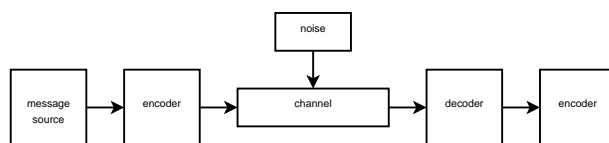


Figure 1.1: Shannon’s model of a communication system.

systems use a continuous channel. As the definitions and mathematics involved with continuous channels is more complex than those of discrete channels, and them being of less interest to computer scientists, this paper will focus on the discrete case.

The other three elements of the model, the *encoder*, the *decoder*, and *noise*, model several of the real-world limitations that communication systems are subject to. The encoder transforms the message into a form that can be send over the channel, and the decoder reconstructs the original message from the symbols exiting the channel. For example, in the telegraph system the person acting as the message source is also acting as the encoder by encoding the message to be send into Morse code. The receiver has to decode the Morse code to be able to read the message. Noise models the effect that in the real-world the symbol that exits the channel will not always be the same symbol that entered the channel. This can be caused by electromagnetic interference when sending messages over metal wires or through the ether, or due to scratches on the medium where a file is stored on.

Starting from this model, Shannon went on to define several properties of the components, such as the *self-information* of the messages transmitted and the *entropy* of the message source. This, in turn, allowed him to proved several theorems answering the question we posed at the beginning of this introduction.

1.1 Overview

In Chapter 2 we will formally introduce the concepts of self-information and entropy and show how they let us quantitatively measure information. In Chapter 3 we will look at why we should encode messages before sending them over a channel, and how efficiently this can be done. In Chapter 4 we will explore the possibilities of sending messages in the presence of noise.

Chapter 2

Information

Imagine Alice rolls a six-sided die and tells Bob whether the outcome was odd or even. How much information did Bob receive from Alice? What if Alice told Bob whether the outcome was six or not? What if Bob could see the die before Alice told him what the outcome was?

In the first case there is little reason to favour odd over even, three of the six faces of the die contain an odd number and three of the six faces an even number. In this case it would be appropriate to say that both the message “odd” and the message “even” convey the same amount of information. In the second case Bob would probably expect the outcome to be “not six” as the probability of the outcome being “six” is five times as small as it being “not six”. Therefore, if Alice told Bob the outcome was “six” this message would reveal more information to Bob than if she merely confirmed Bob’s expectation of the outcome being “not six”. Finally, if Bob already knows the outcome of the roll, nothing what Alice tells him will reveal any new information about the outcome to Bob, as he already knows exactly what Alice is going to tell him. We say that Alice’s message transmits no information.

We will now try to define some of the basic properties of the communication system. We start out with a more simple model than that of Shannon, ignoring the need to encode the messages we send and assuming that there is no noise on the channel which interferes with, and distorts, our message. These difficulties will be treated in later chapters. In this simplified model we can model the message source S as a random variable, where $P(S = m)$ is the probability with which the message receiver expects the message source to send it the message m . For example, in the second case described above we would have $P(S = \text{“six”}) = \frac{1}{6}$ and $P(S = \text{“not six”}) = \frac{5}{6}$.

2.1 Self-information

As we can see from the example above, the amount of information a message transmits is closely tied to the probability with which the receiver expects to receive that message from the message source. More accurately, the amount of information carried by a message is greater if the probability of with which we expect to receive that message is smaller. With this fact in mind we will now

formally define the amount information I a message m transmits or reveals as

$$I(m) = \log_n \left(\frac{1}{P(S = m)} \right) = -\log_n P(S = m). \quad (2.1)$$

The definition of information as defined in (2.1), that is with the logarithm, has several desirable properties. Apart from satisfying our observation that observation that an event with a smaller probability conveys a larger amount of information, it is also *additive*. This means that if we send n (independent) messages $m_1 m_2 \dots m_n$ then the total amount of information will be equal to the sum of the information carried by the individual messages. In formula,

$$I(m_1 m_2 \dots m_n) = \sum_{i=1}^n I(m_i). \quad (2.2)$$

So far, we have not yet specified the base of the logarithm in (2.1) or even where it came. The base we choose will determine the unit in which we express the amount of information. If we choose $n = 10$ the unit will be *Hartleys*, if we choose $n = e$ (the natural logarithm) the unit will be *nats*, and if we choose $n = 2$ the unit will be *bits*. Expressing the amount of information in bits will have several advantages for applications in computer science, so we will assume $n = 2$ throughout the rest of this text.

Example Now let us work out the example given at the beginning of this chapter. In this example Alice is the message source S and Bob the message receiver. The sample space $\Omega = \{1, 2, 3, 4, 5, 6\}$ and under the assumption the die is fair we (and Bob) assume that $p(n) = \frac{1}{6}$ for all $n \in \Omega$. If Alice tells Bob whether the outcome was odd ($\{1, 3, 5\}$) or even ($\{2, 4, 6\}$), Bob will expect that $P(S \in \{1, 3, 5\}) = P(S \in \{2, 4, 6\}) = \frac{1}{2}$ and both messages will therefore transmit

$$I(1, 3, 5) = I(2, 4, 6) = -\log \frac{1}{2} = 1 \text{ bit}$$

of information to Bob. Perhaps the most important aspect of this calculation was that it confirmed our hypothesis that both messages would carry the same amount of information.

Now the second case. Our sample space remains the same, but Bob will now expect that $P(S \in \{6\}) = \frac{1}{6}$ and $P(S \in \{1, 2, 3, 4, 5\}) = \frac{5}{6}$. The message “six” will therefore convey

$$I(6) = -\log \frac{1}{6} \approx 2.58 \text{ bits}$$

of information, while the message “not six” will convey

$$I(1, 2, 3, 4, 5) = -\log \frac{5}{6} \approx 0.26 \text{ bits}$$

of information. This is again in agreement with our hypothesis.

In the final case, where Bob was allowed to see the outcome of throw, and therefore would also be certain of the message that Alice would be going to send him the amount of information Alice’s message conveys would $-\log 1 = 0$ bits, just as expected.

2.2 Entropy

Another important concept in information theory is the entropy $H(S)$ of a message source S , defined as

$$H(S) = \sum_{i=1}^n P(S = m_i) I(m_i) = \sum_{i=1}^n P(S = m_i) \log P(S = m_i). \quad (2.3)$$

This formula shows a striking similarity to the *expected value* function in probability theory. Therefore the entropy of message source can be interpreted as the average amount of information a message send by that message source conveys.

The formula (2.3) satisfies two properties which we will mention here. Let us define a random variable S_p as $P(S_p = m_1) = p$ and $P(S_p = m_2) = 1 - p$. Now the function $H_p(S_p)$ is continuous in p . This can be interpreted to mean that a small change in probability, or information carried by the messages of a message source, will result in an small change of the entropy of the source. The second property is the *grouping axiom*, which Ash [1965] states as

$$\begin{aligned} H(p_1, \dots, p_M) = & H(p_1 + \dots + p_r, p_{r+1}, \dots, p_M) \\ & + (p_1 + \dots + p_r) H\left(\frac{p_1}{\sum_{i=1}^r p_i}, \dots, \frac{p_r}{\sum_{i=1}^r p_i}\right) \\ & + (p_{r+1} + \dots + p_M) H\left(\frac{p_{r+1}}{\sum_{i=r+1}^M p_i}, \dots, \frac{p_M}{\sum_{i=r+1}^M p_i}\right) \end{aligned}$$

and gives

$$H\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right) = H\left(\frac{3}{4}, \frac{1}{4}\right) + \frac{3}{4} H\left(\frac{2}{3}, \frac{1}{3}\right) + \frac{1}{4} H\left(\frac{1}{2}, \frac{1}{2}\right)$$

as an example.

Actually Shannon started out with these two properties of the entropy function and the two properties of the information function mentioned earlier, together called the four *axioms of uncertainty measurement* and proved that (2.1) and (2.3) where the only functions satisfying these four axioms.

Example We will now calculate the entropy of Alice in the various cases given in the example at the start of the chapter. In the first case, Alice's entropy is

$$H(A) = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 = 1.$$

In the second case it will be

$$H(A) = \frac{1}{6} \log 6 + \frac{5}{6} \log \frac{6}{5} \approx 0.65 \quad (2.4)$$

In the final case, Alice will not send any messages, and her entropy will be 0.

2.3 Conditional Self-information and Entropy

Similar to how we can define conditional probabilities in probability theory, we can define conditional self-information of messages and condition entropy of the message source. Where a conditional probability gives the probability of some event will happen, under the assumption some other event will occurs as well, conditional self-information indicates the amount of information a message will convey, given some other piece of information is known.

If we throw a six-sided die and let D denote the outcome of the throw, $P(D = 3|D \text{ is odd}) = \frac{1}{3}$ would be an example of a conditional probability. It states that the chance of the outcome being three would be one-third, given that the outcome was odd.

Conditional self-information and entropy are defined as

$$\begin{aligned} I(m|n) &= -\log p(m|n), \\ H(S|m) &= -\sum_{s \in S} p(s|m) \log p(s|m), \\ H(S|Y) &= -\sum_{y \in Y} p(y) \sum_{s \in S} p(s|y) \log p(s|y). \end{aligned}$$

Example If Alice throws a green, an orange and two purple six-sided dice, adds up the eyes of the green and the purple dice and tells Bob that this number is prime. What would Alice's entropy be if in the next message to Bob she told what the outcome of the green die was?

First we calculate the values of the joint conditional probability function $p(G = g, O = o|g + o \text{ prime})$

	1	2	3	4	5	6
1	$\frac{1}{15}$	$\frac{1}{15}$	0	$\frac{1}{15}$	0	$\frac{1}{15}$
2	$\frac{1}{15}$	0	$\frac{1}{15}$	0	$\frac{1}{15}$	0
3	0	$\frac{1}{15}$	0	$\frac{1}{15}$	0	0
4	$\frac{1}{15}$	0	$\frac{1}{15}$	0	0	0
5	0	$\frac{1}{15}$	0	0	0	$\frac{1}{15}$
6	$\frac{1}{15}$	0	0	0	$\frac{1}{15}$	0

and from this the probabilities and self-information of each message Alice could send

G	1	2	3	4	5	6
$p(G \text{prime})$	$\frac{4}{15}$	$\frac{3}{15}$	$\frac{2}{15}$	$\frac{2}{15}$	$\frac{2}{15}$	$\frac{2}{15}$
$I(G \text{prime})$	1.91	2.32	2.91	2.91	2.91	2.91

This makes Alice's entropy $\frac{1}{6} \log \frac{4}{15} + \frac{1}{6} \log \frac{3}{15} + \frac{4}{6} \log \frac{2}{15} \approx 2.64$.

Chapter 3

Coding

Imagine Alice and Bob would try the same experiment as in the previous chapter, but instead of being able to communicate directly they would only be allowed to communicate through a telegraph. This is where the encoder and decoder stages of Shannon’s model come into play. Instead of Alice just saying whether the outcome of the roll of the die was odd or even, she would have to *encode* the message into a series of dots and dashes, or more eloquently (for a computer scientists, that is) into a serie of ones and zeros.

Let us again start with the first case. One obvious way of encoding the outcomes “odd” and “even” would be

outcome	self-information	code
“odd”	1	0
“even”	1	1

Two things worth noting about this encoding. Firstly, the length of the codes are equal to the information carried by the messages, in both cases 1 bit. Secondly, it is clear that on average be would need to send 1 bit per message, which equals the entropy of the message source. However, if we would choose a similar encoding for the second case, namely

outcome	self-information	code
“six”	1	0
“not six”	1	1

we would this need 1 bit per message and send 1 bit per message on average, which do not match the self-information and entropy of the messages and message source. Instead we could group the messages together in pairs of three and send them in a batch using only the following single codes

outcome	self-information	code
“six”, “six”, “six”	$2.58 + 2.58 + 2.58 \approx 7.75$	11111
“six”, “six”, “not six”	$2.58 + 2.58 + 0.26 \approx 5.43$	11110
“six”, “not six”, “six”	$2.58 + 0.26 + 2.58 \approx 5.43$	11101
“six”, “not six”, “not six”	$2.58 + 0.26 + 0.26 \approx 3.11$	110
“not six”, “six”, “six”	$0.25 + 2.58 + 2.58 \approx 5.43$	11100
“not six”, “six”, “not six”	$0.26 + 2.58 + 0.26 \approx 3.11$	101
“not six”, “not six”, “six”	$2.58 + 2.58 + 0.26 \approx 3.11$	100
“not six”, “not six”, “not six”	$0.26 + 0.26 + 0.26 \approx 0.79$	0

Not only causes the grouping of the message the lengths of the codes to be closer to the self-information of the messages, but a quick calculation shows that we only need to send 1.99 bits per grouped message on average. This translates to 0.66 bits per individual message, which is very close to 0.65, the entropy of the message source we calculated in (2.4).

There are two things worth noting about this grouping of messages. Firstly, by grouping even more would reduce the average number of bits send per message, but never below the entropy. Secondly, grouping messages together in our first example (“odd”, “odd”; “odd”, “even”; ...) would not reduce the the average number of bits send per message.

In fact, algorithms such as Huffman coding, together with the grouping of messages, (Huffman [1952]) or arithmetic coding can generate codes with a length which can come arbitrarily close to the amount of self-information of the message. These facts taken together raise the question whether it is possible to construct a codes, such that the average number of bits send per message would be less than the entropy of the message source. This question was answered in the negative by Shannon, by his proof of the *noiseless coding theorem*.

3.1 Noiseless coding theorem

One of the most important results which can be proved about our simplified communication model is the *noiseless coding theorem*. The noiseless coding theorem states that no code can be constructed where the average length of the codes transmitted is less than the entropy of the source. Furthermore, it can be shown that a code of which the average number of bits needed to send a message is equal to the entropy of the message, is made up of codes of which the length is equal to the information carried by that message.

This has important implications for the field of data compression, as it implies that it would be fruitless to search for coding schemes which are more efficient than Huffman coding with grouping or arithmetic coding.

However, experience learns that different lossless compression schemes can result in significantly different compression ratios. To understand how this is accomplished, one should first realise that the entropy of a message source depends on the probability with which the receiver (or in this case the compression program) expects to receive certain messages from the source (in this case file we wish to compress). Therefore most compression programs do not only consist of a *coder*, but also of a *model* which tries to predict what the next message would likely be based on the previous messages received. To help this prediction the model is, unlike the coder, aware of the semantics of the message source. This better prediction will reduce the entropy of the message source, allowing the coder to generate shorter codes without violating the noiseless coding theorem.

Chapter 4

Channels

4.1 Noise

Until this moment, now we silently assumed that the message receiver would always receive the same message as the message source sent. In reality various physical phenomena, collectively called *noise*, can cause the message to be changed or corrupted during its travel over the channel. We can model this noise by adding a set of probabilities to our channel for each symbol, which indicates the chance with which one symbol is changed into another symbol.

If our channel can transmit two symbols and the chance of one symbol changing into the other are equal for both symbols, we call the channel a *binary symmetric channel*. Figure 4.1 shows such a channel, where the probability of a one or a zero changing into the other is 25%.

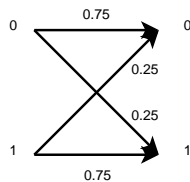


Figure 4.1: A noisy discrete binary symmetric channel.

This noise is usually an unwanted effect, which can make communication difficult or impossible. We would therefore like to add some redundancy to the codes, making it possible to detect which symbols have been changed and reconstruct the original message.

An simple method to achieve this, would be to transmit each symbol several times and have the receiver reconstruct the original symbol by picking the symbol which occurs the most. Unfortunately for us, this method does not work flawlessly. For example, if there was a chance of 0.25 that a symbol would be changed and we would transmit each symbol three times, there would still be a chance of approximately 0.16 that two or three of the symbol would be changed, fooling the receiver when he tries to reconstruct the message.

Shannon proved a very surprising result however, know as the *fundamental theorem of information theory*. This theorem states, that no matter how much

noise there is on the channel, it will always be possible to transmit messages with arbitrary reliability, without lowering the rate of the channel below a certain amount, known as the *channel capacity*.

The channel capacity indicates which percentage of the channel's rate can be used to send useful data, as compared to data necessary for error-correction. The formula to calculate the channel capacity is

$$C = \max_{p(x)} I(X|Y), \quad (4.1)$$

where X is the probability distribution of the input symbol, Y the probability distribution of the output symbol and $I(X|Y)$ the amount of information that Y reveals about X . The quantity $I(X|Y)$ is also called the *information processed by the channel*. It can be shown that $I(X|Y) = H(X) - H(X|Y)$. Ash [1965]

Formula (4.1) is hard to calculate in general, but for a binary symmetric channel it reduces to

$$C = 1 - p \log\left(\frac{1}{p}\right) - (1 - p) \log\left(\frac{1}{1 - p}\right), \quad (4.2)$$

where p is the probability of a symbol changing into another symbol when it is sent over the channel.

For example, if $p = 0.25$ then $E \approx 0.81$ bits need to be reserved for error-correction, leaving approximately 0.19 bits to transmit information. It should be noted that C is always positive, unless $p = 0.5$. In this case $C = 0$, meaning that no information can be sent over the channel. If you think about this for a moment, you will realise that $p = 0.5$ means that the symbols which are output by the channel will be unrelated to the symbols which were put onto the channel.

4.2 Error-correcting codes

Again, information theory only provides us with an upper limit of what error correcting codes can do and does not give a method of constructing them. Unlike the construction of short codes, for which an efficient algorithm was found in 1953, the construction of error correcting codes which approach Shannon's limit remains, an open research question.

One of the first error-correcting codes was the Hamming code, which could only correct single-bit errors. A widely used error-correcting code is the Reed-Solomon coding, which was invented in 1960 and was first widely used in CDs, and is still in heavy use today.

In 1962 Robert G. Gallager developed the low-density parity-check code. This code was for 30 years by far the most efficient code, getting near the Shannon limit. Decoding them is an NP-complete problem, however, so they were of little practical use, until approximate decoder algorithms were developed. Nowadays, low-density parity-check codes are used in the DVB standard for digital television.

In 1993, the three electrical engineers Claude Berrou, Alain Glavieux and Punya Thitimajshima shocked the scientific community with their paper *Near Shannon Limit error-correcting coding and decoding: Turbo-codes*. The Turbo-codes they developed were nearly as efficient as low-density parity-check code,

but could be efficiently encoded and decoded. First used in satellites, Turbo-codes have now found their way into new wireless standards such as UMTS and WiMAX.

Chapter 5

Conclusion

Information theory provides the mathematical foundation, not only for the study of communication systems, but also for the fields of data compression and error correction. While it does not provide many concrete tools to build new compression algorithms or error-correcting codes, it does provide insight in the theoretical limits the efficiency of these codes are bounded by. One may consider this unfortunate, but by revealing us were *not* to look for new breakthroughs it indirectly does point in the direction were we *should* look. This makes information theory not just a mathematical exercise, but also a field which has practical significance.

Bibliography

- R. Ash. *Information Theory*. Interscience, 1965. ISBN 0-470-03445-9.
- M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Addison Wesley, 2002. ISBN 0-321-20473-5.
- S. Goldman. *Information Theory*. Prentice Hall, 1953.
- D. Huffman. A method for the construction of minimum-redundancy codes. In *Proceedings of the I.R.E.*, pages 1098–1102, 1952.
- D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2005.
- G. Markowsky. Information theory. *Encyclopædia Britannica*, 2006.
- R. Meester. *A Natural Introduction to Probability Theory*. Birkhäuser, 2000. ISBN 3-7643-2188-1.
- K. Sayood. *Introduction to Data Compression*. Morgan Kaufmann, 1996. ISBN 1-55860-346-8.
- C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, (27):379–423 & 623–656, July & October 1948.
- Wikipedia. Information theory, 2006. URL http://en.wikipedia.org/w/index.php?title=Information_theory&oldid=4637%9877. [Online; accessed 4-April-2006].