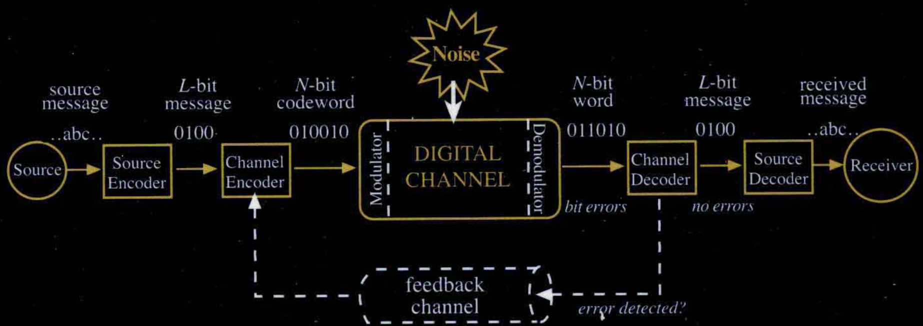
DISCRETE MATHEMATICS AND ITS APPLICATIONS
Series Editor KENNETH H. ROSEN

FUNDAMENTALS of INFORMATION THEORY and CODING DESIGN



Roberto Togneri
Christopher J.S. deSilva

CHAPMAN & HALL/CRC

DISCRETE MATHEMATICS AND ITS APPLICATIONS
Series Editor KENNETH H. ROSEN

FUNDAMENTALS of INFORMATION THEORY and CODING DESIGN

Roberto Togneri
Christopher J.S. deSilva

CHAPMAN & HALL/CRC
A CRC Press Company
Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Togneri, Roberto.
Fundamentals of information theory and coding design / Roberto Togneri and Christopher J.S. deSilva.
p. cm. — (Discrete mathematics and its applications)
Includes bibliographical references and index.
ISBN 1-58488-310-3 (alk. paper)
1.Information theory. 2. Coding theory. I. DeSilva, Christopher J. S. II. Title. III. CRC Press series on discrete mathematics and its applications.

Q360 .T62 2003
003′.54—dc21 2002191160

International Standard Book Number 1-58488-310-3
Library of Congress Card Number 2002191160
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

Preface

What is information? How do we quantify or measure the amount of information that is present in a file of data, or a string of text? How do we encode the information so that it can be stored efficiently, or transmitted reliably?

The main concepts and principles of information theory were developed by Claude E. Shannon in the 1940s. Yet only now, and thanks to the emergence of the information age and digital communication, are the ideas of information theory being looked at again in a new light. Because of information theory and the results arising from coding theory we now know how to quantify information, how we can efficiently encode it and how reliably we can transmit it.

This book introduces the main concepts behind how we model information sources and channels, how we code sources for efficient storage and transmission, and the fundamentals of coding theory and applications to state-of-the-art error correcting and error detecting codes.

This textbook has been written for upper level undergraduate students and graduate students in mathematics, engineering and computer science. Most of the material presented in this text was developed over many years at The University of Western Australia in the unit Information Theory and Coding 314, which was a core unit for students majoring in Communications and Electrical and Electronic Engineering, and was a unit offered to students enrolled in the Master of Engineering by Coursework and Dissertation in the Intelligent Information Processing Systems course.

The number of books on the market dealing with information theory and coding has been on the rise over the past five years. However, very few, if any, of these books have been able to cover the fundamentals of the theory without losing the reader in the complex mathematical abstractions. And fewer books are able to provide the important theoretical framework when discussing the algorithms and implementation details of modern coding systems. This book does not abandon the theoretical foundations of information and coding theory and presents working algorithms and implementations which can be used to fabricate and design real systems. The main emphasis is on the underlying concepts that govern information theory and the necessary mathematical background that describe modern coding systems. One of the strengths of the book are the many worked examples that appear throughout the book that allow the reader to immediately understand the concept being explained, or the algorithm being described. These are backed up by fairly comprehensive exercise sets at the end of each chapter (including exercises identified by an * which are more advanced or challenging).

The material in the book has been selected for completeness and to present a balanced coverage. There is discussion of cascading of information channels and additivity of information which is rarely found in modern texts. Arithmetic coding is fully explained with both worked examples for encoding and decoding. The connection between coding of extensions and Markov modelling is clearly established (this is usually not apparent in other textbooks). Three complete chapters are devoted to block codes for error detection and correction. A large part of these chapters deals with an exposition of the concepts from abstract algebra that underpin the design of these codes. We decided that this material should form part of the main text (rather than be relegated to an appendix) to emphasise the importance of understanding the mathematics of these and other advanced coding strategies.

Chapter 1 introduces the concepts of entropy and information sources and explains how information sources are modelled. In Chapter 2 this analysis is extended to information channels where the concept of mutual information is introduced and channel capacity is discussed. Chapter 3 covers source coding for efficient storage and transmission with an introduction to the theory and main concepts, a discussion of Shannon's Noiseless Coding Theorem and details of the Huffman and arithmetic coding algorithms. Chapter 4 provides the basic principles behind the various compression algorithms including run-length coding and dictionary coders. Chapter 5 introduces the fundamental principles of channel coding, the importance of the Hamming distance in the analysis and design of codes and a statement of what Shannon's Fundamental Coding Theorem tells us we can do with channel codes. Chapter 6 introduces the algebraic concepts of groups, rings, fields and linear spaces over the binary field and introduces binary block codes. Chapter 7 provides the details of the theory of rings of polynomials and cyclic codes and describes how to analyse and design various linear cyclic codes including Hamming codes, Cyclic Redundancy Codes and Reed-Muller codes. Chapter 8 deals with burst-correcting codes and describes the design of Fire codes, BCH codes and Reed-Solomon codes. Chapter 9 completes the discussion on channel coding by describing the convolutional encoder, decoding of convolutional codes, trellis modulation and Turbo codes.

This book can be used as a textbook for a one semester undergraduate course in information theory and source coding (all of Chapters 1 to 4), a one semester graduate course in coding theory (all of Chapters 5 to 9) or as part of a one semester undergraduate course in communications systems covering information theory and coding (selected material from Chapters 1, 2, 3, 5, 6 and 7).

We would like to thank Sean Davey and Nishith Arora for their help with the LaTeX formatting of the manuscript. We would also like to thank Ken Rosen for his review of our draft manuscript and his many helpful suggestions and Sunil Nair from CRC Press for encouraging us to write this book in the first place!

Our examples on arithmetic coding were greatly facilitated by the use of the conversion calculator (which is one of the few that can handle fractions!) made available by www.math.com.

The manuscript was written in LaTeX and we are indebted to the open source software community for developing such a powerful text processing environment. We are especially grateful to the developers of LyX (www.lyx.org) for making writing the document that much more enjoyable and to the makers of xfig (www.xfig.org) for providing such an easy-to-use drawing package.

Roberto Togneri
Chris deSilva

Contents

Chapter 1

Entropy and Information

1.1 Structure

Structure is a concept of which we all have an intuitive understanding. However, it is not easy to articulate that understanding and give a precise definition of what structure is. We might try to explain structure in terms of such things as regularity, predictability, symmetry and permanence. We might also try to describe what structure is not, using terms such as featureless, random, chaotic, transient and aleatory.

Part of the problem of trying to define structure is that there are many different kinds of behaviour and phenomena which might be described as structured, and finding a definition that covers all of them is very difficult.

Consider the distribution of the stars in the night sky. Overall, it would appear that this distribution is random, without any structure. Yet people have found patterns in the stars and imposed a structure on the distribution by naming constellations.

Again, consider what would happen if you took the pixels on the screen of your computer when it was showing a complicated and colourful scene and strung them out in a single row. The distribution of colours in this single row of pixels would appear to be quite arbitrary, yet the complicated pattern of the two-dimensional array of pixels would still be there.

These two examples illustrate the point that we must distinguish between the presence of structure and our perception of structure. In the case of the constellations, the structure is imposed by our brains. In the case of the picture on our computer screen, we can only see the pattern if the pixels are arranged in a certain way.

Structure relates to the way in which things are put together, the way in which the parts make up the whole. Yet there is a difference between the structure of, say, a bridge and that of a piece of music. The parts of the Golden Gate Bridge or the Sydney Harbour Bridge are solid and fixed in relation to one another. Seeing one part of the bridge gives you a good idea of what the rest of it looks like.

The structure of pieces of music is quite different. The notes of a melody can be arranged according to the whim or the genius of the composer. Having heard part of the melody you cannot be sure of what the next note is going to be, leave alone any other part of the melody. In fact, pieces of music often have a complicated, multi-layered structure, which is not obvious to the casual listener.

In this book, we are going to be concerned with things that have structure. The kinds of structure we will be concerned with will be like the structure of pieces of music. They will not be fixed and obvious.

1.2 Structure in Randomness

Structure may be present in phenomena that appear to be random. When it is present, it makes the phenomena more predictable. Nevertheless, the fact that randomness is present means that we have to talk about the phenomena in terms of probabilities.

Let us consider a very simple example of how structure can make a random phenomenon more predictable. Suppose we have a fair die. The probability of any face coming up when the die is thrown is 1/6. In this case, it is not possible to predict which face will come up more than one-sixth of the time, on average.

On the other hand, if we have a die that has been biased, this introduces some structure into the situation. Suppose that the biasing has the effect of making the probability of the face with six spots coming up 55/100, the probability of the face with one spot coming up 5/100 and the probability of any other face coming up 1/10. Then the prediction that the face with six spots will come up will be right more than half the time, on average.

Another example of structure in randomness that facilitates prediction arises from phenomena that are correlated. If we have information about one of the phenomena, we can make predictions about the other. For example, we know that the IQ of identical twins is highly correlated. In general, we cannot make any reliable prediction about the IQ of one of a pair of twins. But if we know the IQ of one twin, we can make a reliable prediction of the IQ of the other.

In order to talk about structure in randomness in quantitative terms, we need to use probability theory.

1.3 First Concepts of Probability Theory

To describe a phenomenon in terms of probability theory, we need to define a *set of outcomes*, which is called the *sample space*. For the present, we will restrict consideration to sample spaces which are finite sets.

DEFINITION 1.1 Probability Distribution *A* probability distribution *on a sample space $S = \{s_1, s_2, \ldots, s_N\}$ is a function P that assigns a probability to each outcome in the sample space. P is a map from S to the unit interval, $P : S \to [0, 1]$, which must satisfy $\sum_{i=1}^{N} P(s_i) = 1$.*

DEFINITION 1.2 Events Events *are subsets of the sample space.*

We can extend a probability distribution P from S to the set of all subsets of S, which we denote by $\mathcal{P}(S)$, by setting $P(E) = \sum_{s \in E} P(s)$ for any $E \in \mathcal{P}(S)$. Note that $P(\emptyset) = 0$.

An event whose probability is 0 is impossible and an event whose probability is 1 is certain to occur.

If E and F are events and $E \cap F = \emptyset$ then $P(E \cup F) = P(E) + P(F)$.

DEFINITION 1.3 Expected Value *If $S = \{s_1, s_2, \ldots, s_N\}$ is a sample space with probability distribution P, and $f : S \to V$ is a function from the sample space to a vector space V, the* expected value *of f is $\bar{f} = \sum_{i=1}^{N} P(s_i) f(s_i)$.*

NOTE We will often have equations that involve summation over the elements of a finite set. In the equations above, the set has been $S = \{s_1, s_2, \ldots, s_N\}$ and the summation has been denoted by $\sum_{i=1}^{N}$. In other places in the text we will denote such summations simply by $\sum_{s \in S}$.

1.4 Surprise and Entropy

In everyday life, events can surprise us. Usually, the more unlikely or unexpected an event is, the more surprising it is. We can quantify this idea using a probability distribution.

DEFINITION 1.4 Surprise *If E is an event in a sample space S, we define the* surprise *of E to be $s(E) = -\log(P(E)) = \log(1/P(E))$.*

Events for which $P(E) = 1$, which are certain to occur, have zero surprise, as we would expect, and events that are impossible, that is, for which $P(E) = 0$, have infinite surprise.

Defining the surprise as the negative logarithm of the probability not only gives us the appropriate limiting values as the probability tends to 0 or 1, it also makes surprise additive. If several independent events occur in succession, the total surprise they generate is the sum of their individual surprises.

DEFINITION 1.5 Entropy *We can restrict the surprise to the sample space and consider it to be a function from the sample space to the real numbers. The expected value of the surprise is the* entropy *of the probability distribution.*

If the sample space is $S = \{s_1, s_2, \ldots, s_N\}$, with probability distribution P, the entropy of the probability distribution is given by

$$H(P) = -\sum_{i=1}^{N} P(s_i)\log(P(s_i)). \tag{1.1}$$

The concept of entropy was introduced into thermodynamics in the nineteenth century. It was considered to be a measure of the extent to which a system was disordered. The tendency of systems to become more disordered over time is described by the *Second Law of Thermodynamics*, which states that the entropy of a system cannot spontaneously decrease. In the 1940's, Shannon [6] introduced the concept into communications theory and founded the subject of information theory. It was then realised that entropy is a property of any stochastic system and the concept is now used widely in many fields. Today, information theory (as described in books such as [1], [2], [3]) is still principally concerned with communications systems, but there are widespread applications in statistics, information processing and computing (see [2], [4], [5]).

Let us consider some examples of probability distributions and see how the entropy is related to predictability. First, let us note the form of the function $s(p) = -p\log(p)$ where $0 < p \leq 1$ and log denotes the logarithm to base 2. (The actual base does not matter, but we shall be using base 2 throughout the rest of this book, so we may as well start here.) The graph of this function is shown in Figure 1.1.

Note that $-p\log(p)$ approaches 0 as p tends to 0 and also as p tends to 1. This means that outcomes that are almost certain to occur and outcomes that are unlikely to occur both contribute little to the entropy. Outcomes whose probability is close to 0.4 make a comparatively large contribution to the entropy.

EXAMPLE 1.1

$S = \{s_1, s_2\}$ with $P(s_1) = 0.5 = P(s_2)$. The entropy is

$$H(P) = -(0.5)(-1) - (0.5)(-1) = 1.$$

In this case, s_1 and s_2 are equally likely to occur and the situation is as unpredictable as it can be. □

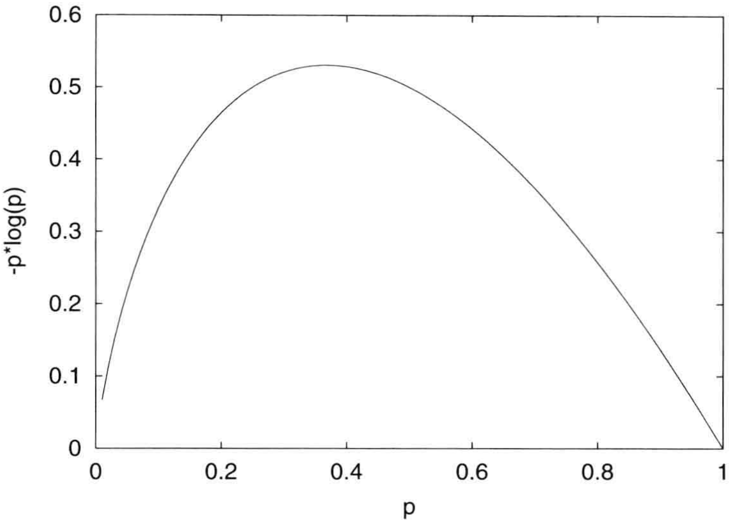


FIGURE 1.1
The graph of $-p\log(p)$.

EXAMPLE 1.2

$S = \{s_1, s_2\}$ with $P(s_1) = 0.96875$, and $P(s_2) = 0.03125$. The entropy is

$$H(P) = -(0.96875)(-0.0444) - (0.03125)(-5) \approx 0.20.$$

In this case, the situation is more predictable, with s_1 more than thirty times more likely to occur than s_2. The entropy is close to zero. □

EXAMPLE 1.3

$S = \{s_1, s_2\}$ with $P(s_1) = 1.0$, and $P(s_2) = 0.0$. Using the convention that $0\log(0) = 0$, the entropy is 0. The situation is entirely predictable, as s_1 always occurs. □

EXAMPLE 1.4

$S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$, with $P(s_i) = 1/6$ for $i = 1, 2, \ldots, 6$. The entropy is 2.585 and the situation is as unpredictable as it can be. □

EXAMPLE 1.5

$S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$, with $P(s_1) = 0.995$ $P(s_i) = 0.001$ for $i = 2, 3, \ldots, 6$.

The entropy is 0.057 and the situation is fairly predictable as s_1 will occur far more frequently than any other outcome. □

EXAMPLE 1.6

$S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$, with $P(s_1) = 0.498 = P(s_2)$ $P(s_i) = 0.001$ for $i = 3, 4, \ldots, 6$. The entropy is 1.042 and the situation is about as predictable as in Example 1.1 above, with outcomes s_1 and s_2 equally likely to occur and the others very unlikely to occur. □

Roughly speaking, a system whose entropy is E is about as unpredictable as a system with 2^E equally likely outcomes.

1.5 Units of Entropy

The units in which entropy is measured depend on the base of the logarithms used to calculate it. If we use logarithms to the base 2, then the unit is the *bit*. If we use natural logarithms (base e), the entropy is measured in *natural units*, sometimes referred to as *nits*. Converting between the different units is simple.

PROPOSITION 1.1
If H_e is the entropy of a probability distribution measured using natural logarithms, and H_r is the entropy of the same probability distribution measured using logarithms to the base r, then

$$H_r = \frac{H_e}{\ln(r)}. \tag{1.2}$$

PROOF Let the sample space be $S = \{s_1, s_2, \ldots, s_N\}$, with probability distribution P. For any positive number x,

$$\ln(x) = \ln(r)\log_r(x). \tag{1.3}$$

It follows that

$$\begin{aligned} H_r(P) &= -\sum_{i=1}^{N} P(s_i)\log_r(P(s_i)) \\ &= -\sum_{i=1}^{N} P(s_i)\frac{\ln(P(s_i))}{\ln(r)} \end{aligned}$$

$$
\begin{aligned}
&= \frac{-\sum_{i=1}^{N} P(s_i)\ln(P(s_i))}{\ln(r)} \\
&= \frac{H_e(P)}{\ln(r)}. \qquad (1.4)
\end{aligned}
$$

$\square$

1.6 The Minimum and Maximum Values of Entropy

If we have a sample space S with N elements, and probability distribution P on S, it is convenient to denote the probability of $s_i \in S$ by p_i. We can construct a vector in R^N consisting of the probabilities:

$$
\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix}.
$$

Because the probabilities have to add up to unity, the set of all probability distributions forms a *simplex* in R^N, namely

$$
K = \left\{ \mathbf{p} \in R^N : \sum_{i=1}^{N} p_i = 1 \right\}.
$$

We can consider the entropy to be a function defined on this simplex. Since it is a continuous function, extreme values will occur at the vertices of this simplex, at points where all except one of the probabilities are zero. If $\mathbf{p}_v$ is a vertex, then the entropy there will be

$$
H(\mathbf{p}_v) = (N-1).0.\log(0) + 1.\log(1).
$$

The logarithm of zero is not defined, but the limit of $x\log(x)$ as x tends to 0 exists and is equal to zero. If we take the limiting values, we see that at any vertex, $H(\mathbf{p}_v) = 0$, as $\log(1) = 0$. This is the minimum value of the entropy function.

The entropy function has a maximum value at an interior point of the simplex. To find it we can use *Lagrange multipliers*.

THEOREM 1.1
If we have a sample space with N elements, the maximum value of the entropy function is $\log(N)$.