Two-Dimensional Information Theory and Coding

WITH APPLICATIONS TO GRAPHICS DATA AND HIGH-DENSITY STORAGE MEDIA

Jørn Justesen and Søren Forchhammer

CAMBRIDGE

Two-Dimensional Information Theory and Coding

With Application to Graphics and High-Density Storage Media

Jørn Justesen and Søren Forchhammer
Technical University of Denmark

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS

Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo, Delhi

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521888608

First published 2010

Printed in the United Kingdom at the University Press, Cambridge

A catalogue record for this publication is available from the British Library

Library of Congress Cataloguing in Publication data
Justesen, Jørn.
Two-dimensional information theory and coding : with application to graphics and high-density storage media / Jørn Justesen and Søren Forchhammer.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-521-88860-8 (hardback)
1. Coding theory. 2. Information storage and retrieval systems – Mathematics. 3. Bipartite graphs. 4. Computer storage devices – Mathematical models. 5. Computer graphics – Mathematics. 6. Data visualization. 7. Binary system (Mathematics) I. Forchhammer, Søren. II. Title.
QA268.J87 2010
005.7′2 – dc22 2009020679

ISBN 978-0-521-88860-8 Hardback

Two-Dimensional Information Theory and Coding

This self-contained introduction to two-dimensional (2-D) theory and coding provides the key techniques for modelling data and estimating their information content. Throughout, special emphasis is placed on applications to transmission, storage, compression, and error protection of graphic information.

The book begins with a self-contained introduction to information theory, including concepts of entropy and channel capacity, which requires minimal mathematical background knowledge. It then introduces error-correcting codes, particularly Reed–Solomon codes, the basic methods for error-correction, and codes applicable to data organized in 2-D arrays. Common techniques for data compression, including compression of 2-D data based on application of the basic source coding, are also covered, together with an advanced chapter dedicated to 2-D constrained coding for storage applications.

Numerous worked examples illustrate the theory, whilst end-of-chapter exercises test the reader's understanding, making this an ideal book for graduate students and also for practitioners in the telecommunications and data-storage industries.

JØRN JUSTESEN is a Professor in the Department of Photonics Engineering at the Technical University of Denmark (DTU), a position he has held since 1976. He has previously held visiting positions at the Institute for Information Transmission Problems, Moscow, and the University of Maryland, College Park.

SØREN FORCHHAMMER is an Associate Professor in the Department of Photonics Engineering at DTU. He has previously held visiting positions at IBM Almaden Research Center, California, and at McMaster University, Ontario.

Preface

Shannon's paper from 1948, which presented information theory in a way that already included most of the fundamental concepts, helped bring about a fundamental change in electronic communication. Today digital formats have almost entirely replaced earlier forms of signaling, and coding has become a universal feature of communication and data storage.

Information theory has developed into a sophisticated mathematical discipline, and at the same time it has almost disappeared from textbooks on digital communication and coding methods. This book is a result of the authors' desire to teach information theory to students of electrical engineering and computer science, and to demonstrate its continued relevance. We have also chosen to mix source coding and error-correcting codes, since both are components of the systems we focus on.

Early attempts to apply information-theory concepts to a broad range of subjects met with limited success. The development of the subject has mostly been fuelled by the advances in design of transmitters and receivers for digital transmission such as modem design and other related applications. However, more recently the extensive use of digitized graphics has made possible a vast range of applications, and we have chosen to draw most of our examples from this area.

The first five chapters of the book can be used for a one-semester course at the advanced-undergraduate or beginning-graduate level. Chapter 6 serves as a transition from the basic subjects to the more complex environments covered by current standards. The last chapters introduce problems related to two-dimensional storage and other current applications.

We would like to thank the graduate students and post-docs who have participated in our work and contributed to figures in the book.

Søren Forchhammer and Jørn Justesen

Contents

1 Introduction to information theory

1.1 Introduction

In this chapter we present some of the basic concepts of information theory. The situations we have in mind involve the exchange of information through transmission and storage media designed for that purpose. The information is represented in digital formats using symbols and alphabets taken in a very broad sense. The deliberate choice of the way information is represented, often referred to as coding, is an essential aspect of the theory, and for many results it will be assumed that the user is free to choose a suitable code.

We present the classical results of information theory, which were originally developed as a model of communication as it takes place over a telephone circuit or a similar connection. However, we shall pay particular attention to two-dimensional (2-D) applications, and many examples are chosen from these areas. A huge amount of information is exchanged in formats that are essentially 2-D, namely web-pages, graphic material, etc. Such forms of communication typically have an extremely complex structure. The term media is often used to indicate the structure of the message as well as the surrounding organization. Information theory is relevant for understanding the possibilities and limitation of many aspects of 2-D media and one should not expect to be able to model and analyze all aspects within a single approach.

1.2 Entropy of discrete sources

A *discrete information source* is a device or process that outputs symbols at discrete instances from some finite alphabet $\mathcal{A} = \{x_1, x_2, \ldots, x_r\}$. We usually assume that a large number of such symbols will have to be processed, and that they may be produced at regular instances in time or read from various positions in the plane in some fashion. The use of a large number of symbols allows us to describe the frequencies in terms of a probability distribution.

A single symbol can be thought of as a random variable, X, with values from the finite alphabet $\mathcal{A}$. We assume that all variables have the same probability distribution, and the probability of x_i is written as $P(x_i)$ or $P[x = x_i]$. It is often convenient to refer to the probability distribution, $P(X)$, of the variable as a vector of probabilities $\mathbf{p}_X$, where the values are listed in some fixed order.

As a measure of the amount of information provided by X we introduce the entropy of a random variable.

Definition. The *entropy* of the variable X is

$$H(X) = E[\log(1/P(X))] = -\sum P(x)\log P(x), \tag{1.1}$$

where $E[\,]$ indicates the expected value.

The word entropy (as well as the use of the letter H) has its historic origin in statistical mechanics. The choice of the base for the logarithms is a matter of convention, but we shall always use binary logarithms and refer to the amount of information as measured in *bits*.

We note that

$$0 \le H(X) \le \log r.$$

The entropy is always positive since each term is positive. The maximal value of $H(X)$ is reached when each value of X has probability $1/r$ and the minimal when one outcome has probability 1 (See Exercise 1.1). When $r = 2^m$, the variable provides at most m bits of information, which shows that the word bit in information theory is used in a way that differs slightly from common technical terminology. If X^N represents a string of N symbols, each with the same alphabet and probability distribution, it follows from (1.1) that the maximal entropy is NH, and that this maximum is reached when the variables are independent. We refer to a source for which the outputs are independent as a *memoryless source*, and if the probability distributions also are identical we describe it as i.i.d. (independent identically distributed). In this case we say that the entropy of the source is H bits/symbol.

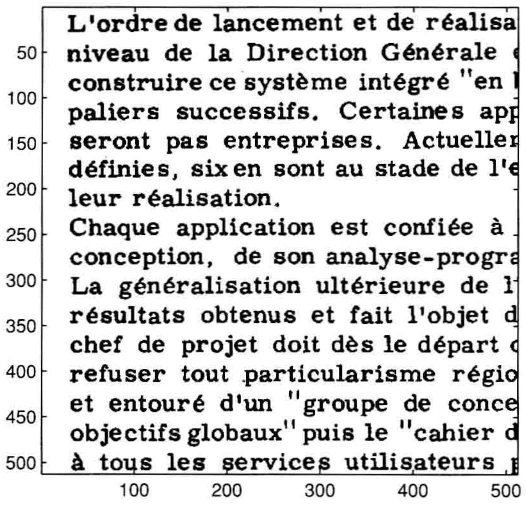
Example 1.1 (Entropy of a binary source). *For a binary memoryless source with probability distribution $P = (p, 1-p)$ we get the entropy (1.1) as*

$$H(p) = -p\log p - (1-p)\log(1-p), \tag{1.2}$$

where $H(p)$ is called the (binary) entropy function. The entropy function is symmetric with respect to $p = 1/2$, and reaches its maximal value, 1, there. For small p, H increases quickly with increasing p, and $H(0.11) = 0.5$.

We note that the same letter, H, is used to indicate the entropy of a random variable and the value of the entropy function for a particular distribution, $(p, 1-p)$. We shall use the same notation $H(P)$ when P is a general discrete probability distribution.

Example 1.2 (Run-length coding). *For a binary source with a small value of $P(1)$ it may be useful to segment a string of symbols into "runs" of 0s ending with a 1. In this way the source is converted into a memoryless source with alphabet $\{0, 1, 2, \ldots, n\}$, where the symbols indicate the length of the segment. The value k represents the string 0^k1, $k < n$. If runs of more than n 0s are possible, the value $k = n$ represents the string*


50
100
150
200
250
300
350
400
450
500
100
200
300
400
500

L'ordre de lancement et de réalisa
niveau de la Direction Générale
construire ce système intégré "en l
paliers successifs. Certaines app
seront pas entreprises. Actueller
définies, six en sont au stade de l'e
leur réalisation.
Chaque application est confiée à
conception, de son analyse-progra
La généralisation ultérieure de l'
résultats obtenus et fait l'objet d
chef de projet doit dès le départ c
refuser tout particularisme régio
et entouré d'un "groupe de conce
objectifs globaux" puis le "cahier d
à tous les services utilisateurs

Figure 1.1. A binary image of a printed text.

0^n, i.e. a run of 0s that is not terminated. Run-length coding may be used to code binary images of text along the rows (Fig. 1.1).

For a pair of random variables, (X, Y), with values from finite alphabets $\mathcal{A}_x = \{x_1, x_2, \ldots, x_r\}$ and $\mathcal{A}_y = \{y_1, y_2, \ldots, y_s\}$, the conditional probability of y_j given x_i is $P(y_j|x_i) = q_{ij}$. It is often convenient to refer to the conditional probability distribution as the matrix $\mathbf{Q} = P(Y|X) = [q_{ij}]$. The probability distribution of the variable, Y, then becomes

$$\mathbf{p}_Y = \mathbf{p}_X\mathbf{Q},$$

where $\mathbf{p}_X$ and $\mathbf{p}_Y$ are the distributions of X and Y, respectively.

Definition. The *conditional entropy*

$$\begin{aligned} H(Y|X) &= -\sum_{i,j} P(x_i, y_j)\log P(y_j|x_i) \\ &= -\sum_i P(x_i)\sum_j P(y_j|x_i)\log P(y_j|x_i) \end{aligned} \tag{1.3}$$

can be interpreted as the additional information provided by Y when X is known.

For any particular y_j, the average value of $P(y_j|x_i)\log P(y_j|x_i)$ is given by $-P(y_j)\log P(y_j)$. Since the function $-u\log u$ is concave (has negative second derivative), we can use Jensen's inequality, which states that for such a function

$$f(E(u)) \geq E[f(u)].$$

Thus, on applying Jensen's inequality with $u = p(y|x)$ to (1.3), it follows that the entropy is never increased by conditioning, i.e. $H(Y) \geq H(Y|X)$.

For a string of variables, $X_1, X_2, \ldots, X_n$ we have

$$\begin{aligned} &P(X_1, X_2, \ldots, X_n) \\ &\quad = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2) \ldots P(X_n|X_1, X_2, \ldots, X_{n-1}) \end{aligned}$$

and thus, taking the expectation of the logarithm,

$$\begin{aligned} &H(X_1, X_2, \ldots, X_n) \\ &\quad = H(X_1) + H(X_2|X_1) + \cdots + H(X_n|X_1, X_2, \ldots, X_{n-1}). \end{aligned} \tag{1.4}$$

This result is known as the *chain rule*, and expresses the total amount of information as a sum of information contributions from variable X_i when values of the previous variables are known. When the variables are not independent, the chain rule is often a valuable tool with which to decompose the probabilities and entropies of the entire string of data into terms that are easier to analyze and calculate.

Example 1.3 (Entropy by the chain rule). *If X takes values in $\mathcal{A} = \{x_1, x_2, x_3, x_4\}$ with probability distribution $(1/4, 1/4, 1/3, 1/6)$, we can of course calculate the entropy directly from the definition. As an alternative, we can think of the outcome as a result of first choosing a subset of two values, represented by the variable Y. We can then express the entropy by the chain rule:*

$$\begin{aligned} H(X) = H(Y) + H(X|Y) &= H(1/2) + (1/2)H(1/2) + (1/2)H(1/3) \\ &= 3/2 + (\log 3 - 2/3)/2 = 7/6 + (1/2)\log 3. \end{aligned}$$

1.3 Mutual information and discrete memoryless channels

For a pair of random variables, (X, Y), with values from finite alphabets $\mathcal{A}_x = \{x_1, x_2, \ldots, x_r\}$ and $\mathcal{A}_y = \{y_1, y_2, \ldots, y_s\}$, the *mutual information* is defined as

$$I(Y; X) = \sum_{x,y} P(x, y)\log\left(\frac{P(y|x)}{P(y)}\right) = E\left[\log\left(\frac{P(y|x)}{P(y)}\right)\right]. \tag{1.5}$$

The mutual information is a measure of the amount of information about X represented by Y. It follows from the definition that the mutual information can be expressed in terms of entropies as

$$I(X; Y) = H(Y) - H(Y|X) = H(X) - H(X|Y). \tag{1.6}$$

The properties of the conditional entropy imply that the mutual information $I(X; Y) \geq 0$. It is 0 when X and Y are independent, and achieves the maximal value $H(X)$ when each value of X gives a unique value of Y.

It may also be noted that $I(Y; X) = I(X; Y)$. The symmetry follows from rewriting $P(y|x)$ as $P(y, x)/P(x)$ in the definition of I. This is an unexpected property since there is often a causal relation between X and Y, and it is not possible to interchange cause and effect.

From the chain rule for entropy we get

$$\begin{aligned} &I(Y; X_1, X_2, \ldots, X_n) \\ &\quad = I(Y; X_1) + I(Y; X_2|X_1) + \cdots + I(Y; X_n|X_1, X_2, \ldots, X_{n-1}), \end{aligned} \tag{1.7}$$

which we refer to as the *chain rule for mutual information.*

A discrete information channel is a model of communication where X and Y represent the input and output variables, and the channel is defined by the conditional probability matrix, $\mathbf{Q} = P(Y|X) = [q_{ij}]$, which in this context is also called the transition matrix of the channel. When the input distribution $P(X)$ is given, we find the output distribution as

$$\mathbf{p}_Y = \mathbf{p}_X\mathbf{Q}.$$

Example 1.4 (The binary symmetric channel). *The single most important channel model is the binary symmetric channel (BSC). This channel models a situation in which random errors occur with probability p in binary data. For equally distributed inputs, the output has the same distribution, and the mutual information may be found from (1.6) as*

$$C = I(Y; X) = 1 - H(p), \tag{1.8}$$

where H again indicates the binary entropy function. Thus $C = 1$ for $p = 0$ or 1, C has minimal value 0 for $p = 1/2$, and $C = 1/2$ for $p = 0.11$.

For two variables we can combine (1.4) and (1.6) to get

$$H(X, Y) = H(X) + H(Y) - I(X; Y).$$

Mutual information cannot be increased by data processing. If Z is a variable that is obtained by processing Y, but depends on X only through Y, we have from (1.7)

$$I(X; Z, Y) = I(X; Y) + I(X; Z|Y) = I(X; Z) + I(X; Y|Z).$$

Here $I(X; Z|Y) = 0$, since X and Z are independent given Y, and $I(X; Y|Z) \geq 0$. Thus

$$I(X; Z) \leq I(X; Y). \tag{1.9}$$

1.4 Source coding for discrete sources

In this section we demonstrate that the entropy of a message has a more operational significance: if the entropy of the source is H, a message in the form of a string of N symbols can, by suitable coding, be represented by about NH binary symbols. We refer to such a process as *source coding.*

1.4.1 The Kraft inequality

We shall prove the existence of efficient source codes by actually constructing some codes that are important in applications. However, getting to these results requires some intermediate steps.

A binary variable-length source code is described as a mapping from the source alphabet $\mathcal{A}$ to a set of finite strings, C from the binary code alphabet, which we always denote $\{0, 1\}$. Since we allow the strings in the code to have different lengths, it is important that we can carry out the reverse mapping in a unique way. A simple way of ensuring this property is to use a *prefix code*, a set of strings chosen in such a way that no string is also the beginning (prefix) of another string. Thus, when the current string belongs to C, we know that we have reached the end, and we can start processing the following symbols as a new code string. In Example 1.5 an example of a simple prefix code is given.

If c_i is a string in C and $l(c_i)$ its length in binary symbols, the *expected length* of the source code per source symbol is

$$L(C) = \sum_{i=1}^{N} P(c_i)l(c_i).$$

If the set of lengths of the code is $\{l(c_i)\}$, any prefix code must satisfy the following important condition, known as the *Kraft inequality*:

$$\sum_i 2^{-l(c_i)} \leq 1. \tag{1.10}$$

The code can be described as a binary search tree: starting from the root, two branches are labelled 0 and 1, and each node is either a leaf that corresponds to the end of a string, or a node that can be assumed to have two continuing branches. Let l_m be the maximal length of a string. If a string has length $l(c)$, it follows from the prefix condition that none of the $2^{l_m-l(c)}$ extensions of this string are in the code. Also, two extensions of different code strings are never equal, since this would violate the prefix condition. Thus by summing over all codewords we get

$$\sum_i 2^{l_m-l(c_i)} \leq 2^{l_m}$$

and the inequality follows. It may further be proven that any uniquely decodable code must satisfy (1.10) and that if this is the case there exists a prefix code with the same set of code lengths. Thus restriction to prefix codes imposes no loss in coding performance.

Example 1.5 (A simple code). *The code* $\{0, 10, 110, 111\}$ *is a prefix code for an alphabet of four symbols. If the probability distribution of the source is (1/2, 1/4, 1/8, 1/8), the average length of the code strings is* $1 \times 1/2 + 2 \times 1/4 + 3 \times 1/4 = 7/4$, *which is also the entropy of the source.*

If all the numbers $-\log P(c_i)$ were integers, we could choose these as the lengths $l(c_i)$. In this way the Kraft inequality would be satisfied with equality, and furthermore

$$L = \sum_i P(c_i) l(c_i) = -\sum_i P(c_i) \log P(c_i) = H(X)$$

and thus the expected code length would equal the entropy. Such a case is shown in Example 1.5. However, in general we have to select code strings that only approximate the optimal values. If we round $-\log P(c_i)$ to the nearest larger integer $\lceil -\log P(c_i) \rceil$, the lengths satisfy the Kraft inequality, and by summing we get an upper bound on the code lengths

$$l(c_i) = \lceil -\log P(c_i) \rceil \leq -\log P(c_i) + 1. \tag{1.11}$$

The difference between the entropy and the average code length may be evaluated from

$$\begin{aligned} H(X) - L &= \sum_i P(c_i) \left[\log\left(\frac{1}{P(c_i)} \right) - l_i \right] = \sum_i P(c_i) \log\left(\frac{2^{-l_i}}{P(c_i)} \right) \\ &\leq \log \sum_i 2^{-l_i} \leq 0, \end{aligned}$$

where the inequalities are those established by Jensen and Kraft, respectively. This gives

$$H(X) \leq L \leq H(X) + 1, \tag{1.12}$$

where the right-hand side is given by taking the average of (1.11).

The loss due to the integer rounding may give a disappointing result when the coding is done on single source symbols. However, if we apply the result to strings of N symbols, we find an expected code length of at most $NH + 1$, and the result per source symbol becomes at most $H + 1/N$. Thus, for sources with independent symbols, we can get an expected code length close to the entropy by encoding sufficiently long strings of source symbols.

1.4.2 Huffman coding

In this section we present a construction of a variable-length source code with minimal expected code length. This method, *Huffman coding*, is a commonly used coding technique. It is based on the following simple recursive procedure for binary codewords.

(1) Input: a list of symbols, S, and their probability distribution.
(2) Let the two smallest probabilities be p_i and p_j (maybe not unique).
(3) Assign these two symbols the same code string, α, followed by 0 and 1, respectively.
(4) Merge them into a single symbol, a_{ij}, with probability $p(a_{ij}) = p_i + p_j$.
(5) Repeat from Step 1 until only a single symbol is left.

The result of the procedure can be interpreted as a decision tree, where the code string is obtained as the labels on the branches that lead to a leaf representing the symbol. It