Capacity-Achieving Probabilistic Shaping for Noisy and Noiseless Channels

Von der Fakultät für Elektrotechnik und Informationstechnik
der Rheinisch-Westfälischen Technischen Hochschule Aachen
zur Erlangung des akademischen Grades
eines Doktors der Ingenieurwissenschaften
genehmigte Dissertation

vorgelegt von
Dipl. El.-Ing. ETH Georg Böcherer
aus Freiburg im Breisgau

Berichter: Prof. Dr. rer. nat. Rudolf Mathar
Prof. Dr. sc. techn. Gerhard Kramer

Tag der mündlichen Prüfung
13.02.2012

Diese Dissertation ist auf den Internetseiten
der Hochschulbibliothek online verfügbar.

Acknowledgments

I want to thank Prof. Rudolf Mathar for the freedom to pursue my ideas during my time at his institute. The TI group was my second home for four years and a half, thank you all. Thanks to Daniel and Gernot, Chunhui and Milan, Fabian and Steven, Andreas and Martijn for collaboration, trips around the world, coffee, and friendship. Special thanks to Prof. Valdemar Cardoso da Rocha Junior and Prof. Cecilio Pimentel for the continuous support. I am grateful to my father Prof. Siegfried Böcherer for all the telephone calls that helped me to get the math at least partially right. Prof. Gerhard Kramer read my dissertation cover to cover, which is the best reward I can think of. Finally, I thank my wife Noêmia and our children Izabel and Rafael for reminding me on a daily basis that work is not the only thing that matters.

Contents

1 Introduction

1.1 Motivation

In Shannon theory, the key step in calculating capacity of a communication channel is to determine the capacity-achieving input probability mass function (pmf). Unequal transition probabilities between input and output symbols, input power constraints, or input symbols of unequal durations can lead to non-uniform capacity-achieving input pmfs. A key result in information theory was an efficient algorithm to *calculate* capacity-achieving pmfs, published independently in 1972 by Blahut [6] and Arimoto [5].

In digital communication systems, there is a binary interface that separates the system into a source-related part and a channel-related part. A natural question is how the bit stream at the binary interface can be mapped to channel input symbols in such a way that the resulting pmf is close to capacity-achieving. The topic of this thesis is to answer this question.

In literature, for noisy channels, the generation of channel input pmfs is referred to as *signal shaping*. There is a vast amount of literature on this topic, see Fischer [33, Chapter 4] and references therein.

In this thesis, we consider signal shaping where the channel input pmf is generated by parsing the bit stream at the binary interface by a prefix-free code. We call this approach *prefix-free matching* and a device that implements this procedure a *prefix-free matcher*. This idea is old and occurred both in the context of noisy and noiseless channels. To use prefix-free matchers for noisy channels was first proposed by Forney *et al* [35, Section IV.A]. Kschischang and Pasupathy proposed in [52] to use the Huffman source code of the pmf that maximizes entropy at the channel input as a prefix-free matcher. In [73], Ungerböck called this approach *Huffman shaping* and pointed out how it can be incorporated into a digital communication system. For some specific noiseless channels, prefix-free matchers were proposed in the literature, for example by Franaszek [36]. Kerpez suggested in [48] to use the Huffman source code of the capacity-achieving pmf as prefix-free matcher for runlength constrained noiseless channels. Although the prefix-free matchers proposed in literature perform well in the considered examples, no information theoretic justifications are given. A remarkable exception is the work of Lempel *et al* [53], which derives an efficient algorithm that finds the optimal prefix-free matcher for noiseless channels with unequal symbol durations. Furthermore, Lempel *et al* provide an example that shows that Huffman shaping is in general sub-optimal. However, no achievability results are given in [53], i.e., it remains an open question if capacity can be achieved by prefix-free matchers when the channel symbols are generated blockwise and the blocklength goes to infinity. Kerpez claims in [48] that Huffman shaping is asymptotically capacity-achieving for runlength constraints, however, while I

believe that this is in fact true, the given proof is rough and I did not succeed to fill in the intermediate steps.

In summary, the two simple questions "how can optimal prefix-free matchers be constructed?" and "are prefix-free matchers asymptotically capacity-achieving?" have only been addressed for specific cases in the literature. This is surprising, since the corresponding prefix-free source coding results have been known for a long time: asymptotic achievability has been shown by Shannon in his 1948 paper [69, Section 1.9] and Huffman showed in 1952 [45] how optimal prefix-free source codes can be constructed. The main theme of this thesis is to answer these two questions for prefix-free matching.

1.2 Contributions

We introduce some basic concepts in Chapter 2. Our contributions can then be divided into three parts.

Memoryless channels (Chapter 3, 4, 5). We consider the three information theoretic functionals *entropy* $\mathbb{H}$, *relative entropy* (often referred to as *Kullback-Leibler distance*) $\mathbb{D}$ as a function of the first argument, and *mutual information* as a function of the input pmf. We show the following:

- Discrete input channels: we propose the algorithm *geometric Huffman coding*(GHC). GHC finds the optimal prefix-free matcher for $\mathbb{D}$ and $\mathbb{H}$ and is asymptotically capacity-achieving for $\mathbb{D}$, $\mathbb{H}$, and $\mathbb{I}$. In particular, GHC minimizes $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x})$ over all dyadic pmfs $\boldsymbol{d}$. This is in contrast to the pmf induced by Huffman coding, which minimizes $\mathbb{D}(\boldsymbol{x}\|\boldsymbol{d})$ over all dyadic pmfs $\boldsymbol{d}$. Our proof of the optimality of GHC solves a question raised in [3].

- Discrete input channels with unequal symbol durations: we propose the algorithm *normalized geometric Huffman coding* (nGHC). This algorithm finds the optimal prefix-free matcher for $\mathbb{D}$ and $\mathbb{H}$, and asymptotically achieves capacity for $\mathbb{D}$, $\mathbb{H}$, and $\mathbb{I}$.

- Discrete input channels subject to a constraint on the average symbol cost: we propose the algorithm *cost constrained geometric Huffman coding* (ccGHC). We show that ccGHC is asymptotically capacity-achieving for $\mathbb{D}$, $\mathbb{H}$, and $\mathbb{I}$.

Noiseless channels with memory (Chapter 6). We extend our results from the first part to general noiseless channels, which are specified by an infinite set of strings of positive length. We show the following.

- General noiseless channels: to assess the fundamental relation between the combinatorial and the probabilistic notion of capacity, we use the concept of *general sources* as defined by Han [42, Remark 1.3.2]. We prove that the maximum entropy rate of general sources is exactly equal to the combinatorial capacity. With

the help of this result, we show that the maximum entropy rate of a conventional source is upper-bounded by the combinatorial capacity.

- Channels generated by finite state graphs: we show that any finite state channel has a memoryless representation, i.e., that it can be generated by a graph with only one state. The maximum entropy rate of a finite state channel is equal to the combinatorial capacity. This extends the results of Shannon [69, Theorem 8] and Khandekar *et al* [49, Theorem 5.1] to finite state channels with periodic adjacency matrices. Our proof uses memoryless representations to construct capacity-achieving sources and is not based on *Perron-Frobenius theory*.

- Based on nGhc we define *variable length memoryless* (vlm) *codes*. vlm codes are capacity-achieving for finite state channels. In the literature, the state of the art for noiseless coding is the *State Splitting Algorithm*, which allows to construct capacity-achieving codes for finite state noiseless channels with integer valued symbol lengths, see Marcus *et al* [59, Chapter 4 & 5] and Lind and Marcus [54, Chapter 5]. vlm codes work for any positive symbol lengths. Furthermore, the construction of vlm codes is simple while the State Splitting Algorithm is involved.

Prefix-free matchers and systematic block codes (Chapter 7, 8). In the third part, we show how prefix-free matchers can be combined with systematic block codes.

- For the general class of not necessarily binary systematic block codes, the concepts of shaping gain and coding gain are developed. The shaping gain determines the performance loss that results from using a pmf different from the capacity-achieving one, and the coding gain quantifies the loss due to sub-optimal codes.

- Sparse-dense transmission as introduced by Ratzer [65, Chapter 5] is considered. Here, only the data symbols are matched to the channel and check symbols are not. We derive an algorithm to calculate sparse-dense capacity and prove that prefix-free matchers asymptotically achieve sparse-dense capacity. Formulas for shaping and coding gain of sparse-dense transmission are provided.

- We propose matched transmission, a scheme that allows to operate a systematic block code in such a way that all symbols are matched to the channel. We derive analytical formulas for shaping and coding gain. Capacity of matched transmission is equal to channel capacity and we show for matched transmission that prefix-free matchers are asymptotically capacity-achieving.

- The results are evaluated for a binary symmetric channel where the input symbols zero and one are of unequal duration. The numerical results reveal that prefix-free matchers in practice allow to operate systematic low-density parity-check codes with a shaping gain that does not degrade capacity, i.e., the gap between achieved transmission rate and capacity is only due to the applied error-correcting code.

2 Preliminaries

2.1 Convex optimization

2.1.1 Basic definitions

Sets The following table summarizes the notation for some common sets.

$\mathbf{R}$	set of *real numbers*
$\mathbf{R}_{\geq 0}$	set of *non-negative real numbers*
$\mathbf{R}_{>0}$	set of *positive real numbers*
$\mathbf{N}$	set of *natural numbers* $\{1, 2, 3, \dots\}$
$\mathbf{N}_0$	set of *natural numbers including zero* $\{0, 1, 2, \dots\}$
$\mathbf{Z}$	set of *integers*
$\mathbf{C}$	set of *complex numbers*

In general, we use calligraphic letters for sets. Denote by $\mathcal{S}$ some set. Then $\mathcal{S}^n$ denotes the *Cartesian product* of n copies of $\mathcal{S}$.

Convex set A set $\mathcal{S}$ is *convex* if the line between any two points in $\mathcal{S}$ lies in $\mathcal{S}$, i.e., if for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}$ and any θ with $0 \leq \theta \leq 1$, we have

$$\theta \boldsymbol{x}_1 + (1 - \theta)\boldsymbol{x}_2 \in \mathcal{S}. \tag{2.1}$$

Convex functions A function $f : \mathbf{R}^n \to \mathbf{R}$ is *convex* if its domain $\mathbf{dom}\, f$ is a convex set and if for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbf{dom}\, f$, and θ with $0 \leq \theta \leq 1$, we have

$$f[\theta \boldsymbol{x}_1 + (1 - \theta)\boldsymbol{x}_2] \leq \theta f(\boldsymbol{x}_1) + (1 - \theta) f(\boldsymbol{x}_2). \tag{2.2}$$

If for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbf{dom}\, f$, $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ equality only holds for $\theta = 0$ and $\theta = 1$, then f is *strictly convex*.

Concave functions A function f is *concave* if its negative is convex and *strictly concave* if its negative is strictly convex.

Convex optimization problem A *convex optimization problem* is of the form

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & f_0(\boldsymbol{x}) \\ \text{subject to} & f_i(\boldsymbol{x}) \leq 0, \quad i = 1, \dots, p \\ & h_i(\boldsymbol{x}) = 0, \quad i = 1, \dots, q \end{array} \tag{2.3}$$

with variable $\boldsymbol{x} \in \mathbf{R}^n$. The objective f_0 is a convex function, the inequality constraints f_i are convex functions, and the equality constraints h_i are affine functions. In the context of duality, (2.3) is referred to as the *primal problem*.

Domain The *domain* $\mathcal{D}$ of Problem (2.3) is defined as

$$\mathcal{D} := \mathbf{dom}\, f_0 \cap \bigcap_{i=1}^{p} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{q} \mathbf{dom}\, h_i. \tag{2.4}$$

Feasible The set of *feasible points* is

$$\mathcal{F} := \{\boldsymbol{x} \in \mathcal{D} | f_i(\boldsymbol{x}) \leq 0,\ i = 1, \ldots, p \text{ and } h_i(\boldsymbol{x}) = 0,\ i = 1, \ldots, q\}. \tag{2.5}$$

Note that the domain $\mathcal{D}$ is not the same as the feasible set, i.e., some points in $\mathcal{D}$ may be infeasible. A problem is *feasible* if there is at least one feasible point. Otherwise the problem is called *infeasible*.

Optimality If $\boldsymbol{x}^*$ minimizes $f_0(\boldsymbol{x})$ among all feasible points $\boldsymbol{x}$, then $\boldsymbol{x}^*$ is called an *optimal point* and $f_0(\boldsymbol{x}^*)$ is called the *optimal value*.

Affine hull The *affine hull* of a set $\mathcal{S}$ is defined as the set of all affine combinations of points in $\mathcal{S}$, i.e.,

$$\mathbf{aff}\, \mathcal{S} := \{\theta_1 \boldsymbol{x}_1 + \cdots + \theta_k \boldsymbol{x}_k | \boldsymbol{x}_1, \ldots, \boldsymbol{x}_k \in \mathcal{S}, \theta_1 + \cdots + \theta_k = 1, k \in \mathbf{N}\}. \tag{2.6}$$

Relative interior The *relative interior* of a set $\mathcal{S}$ is defined as

$$\mathbf{relint}\, \mathcal{S} := \{\boldsymbol{x} \in \mathcal{S} | B(\boldsymbol{x}, r) \cap \mathbf{aff}\, \mathcal{S} \subseteq \mathcal{S} \text{ for some } r > 0\} \tag{2.7}$$

where $B(\boldsymbol{x}, r)$ is a ball in $\mathbf{R}^n$ around $\boldsymbol{x}$ with radius r.

2.1.2 Duality

Lagrangian The *Lagrangian* is defined as

$$L : \mathbf{R}^n \times \mathbf{R}^p \times \mathbf{R}^q \to \mathbf{R}, \quad L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) := f_0(\boldsymbol{x}) + \sum_{i=1}^{p} \lambda_i f_i(\boldsymbol{x}) + \sum_{i=1}^{q} \nu_i h_i(\boldsymbol{x}). \tag{2.8}$$

The domain of L is $\mathbf{dom}\, L = \mathcal{D} \times \mathbf{R}^p \times \mathbf{R}^q$.

Dual function The *dual function* is defined as

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) := \inf_{\boldsymbol{x} \in \mathcal{D}} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}). \tag{2.9}$$

Dual problem The *dual problem* is

$$\begin{array}{ll} \underset{\boldsymbol{\lambda},\boldsymbol{\nu}}{\text{maximize}} & g(\boldsymbol{\lambda},\boldsymbol{\nu}) \\ \text{subject to} & \lambda_i \geq 0, \qquad i = 1,\ldots,p. \end{array} \tag{2.10}$$

The feasible points of the dual problem are called *dual feasible* and the optimal points $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ are called *dual optimal*.

Strong duality For any primal feasible point $\boldsymbol{x}$ and dual feasible point $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, the dual function lower-bounds the objective function, i.e., $f_0(\boldsymbol{x}) \geq g(\boldsymbol{\lambda}, \boldsymbol{\nu})$, see [19, Section 5.1.3]. If the optimal primal value is equal to the optimal dual value, we say that *strong duality* holds. If $f_0(\boldsymbol{x}) = g(\boldsymbol{\lambda}, \boldsymbol{\nu})$, then $\boldsymbol{x}$ minimizes f_0 and is primal optimal and $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ maximizes g and is dual optimal.

Karush-Kuhn-Tucker conditions The following proposition states the *Karush-Kuhn-Tucker (KKT) conditions* for optimality.

Proposition 2.1. *Denote by f_0 a function that is convex on its domain* $\mathbf{dom}\, f_0 \subseteq \mathbf{R}^n$. *Assume further that strong duality holds. Then the following conditions are necessary and sufficient for a feasible point $\boldsymbol{x}$ and the point $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ to be primal and dual optimal.*

$$\lambda_i \geq 0, \qquad i = 1,\ldots,p \tag{2.11}$$

$$\lambda_i f_i(\boldsymbol{x}) = 0, \qquad i = 1,\ldots,p \tag{2.12}$$

$$\boldsymbol{x} \text{ minimizes } L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \text{ over } \mathbf{dom}\, f_0. \tag{2.13}$$

Proof. Necessity. Let $\boldsymbol{x}^*$ be primal optimal and let $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ be dual optimal. This directly implies that $\boldsymbol{\lambda}^*$ is feasible and that condition (2.11) must hold. To see the necessity of conditions (2.12) and (2.13), consider the following chain of inequalities.

$$f_0(\boldsymbol{x}^*) = g(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) \tag{2.14}$$

$$= \inf_{\boldsymbol{x} \in \mathbf{dom}\, f_0} \left(f_0(\boldsymbol{x}) + \sum_{i=1}^{p} \lambda_i^* f_i(\boldsymbol{x}) + \sum_{i=1}^{q} \nu_i^* h_i(\boldsymbol{x}) \right) \tag{2.15}$$

$$\leq f_0(\boldsymbol{x}^*) + \sum_{i=1}^{p} \lambda_i^* f_i(\boldsymbol{x}^*) + \sum_{i=1}^{q} \nu_i^* h_i(\boldsymbol{x}^*) \tag{2.16}$$

$$\leq f_0(\boldsymbol{x}^*). \tag{2.17}$$

The equality in the first line follows from strong duality, the second line follows from the definition of the dual function, the third line follows since the infimum of the Lagrangian over $\boldsymbol{x}$ is less than or equal to its value at $\boldsymbol{x} = \boldsymbol{x}^*$. The last line follows since by primal and dual feasibility, $\lambda_i \geq 0$ and $f_i(\boldsymbol{x}^*) \leq 0$, $i = 1,\ldots,p$ and $h_i(\boldsymbol{x}) = 0$, $i = 1,\ldots,q$. Thus, we have equality in all lines. Condition (2.12) now follows from equality in the last line and condition (2.13) follows from equality in the third line.

Sufficiency. Assume conditions (2.11)–(2.13) hold for $\boldsymbol{x}$ and $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ and assume further that $\boldsymbol{x}$ is feasible. Then

$$\begin{aligned} g(\boldsymbol{\lambda}, \boldsymbol{\nu}) &= L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) && (2.18) \\ &= f_0(\boldsymbol{x}) + \sum_{i=1}^{p} \lambda_i f_i(\boldsymbol{x}) + \sum_{i=1}^{q} \nu_i h_i(\boldsymbol{x}) && (2.19) \\ &= f_0(\boldsymbol{x}). && (2.20) \end{aligned}$$

The first line follows from condition (2.13) and the last line follows from condition (2.12) and feasibility of $\boldsymbol{x}$. Condition (2.11) implies that $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ is dual feasible. In summary, $\boldsymbol{x}$ and $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ are primal and dual feasible and $f_0(\boldsymbol{x}) = g(\boldsymbol{\lambda}, \boldsymbol{\nu})$. Therefore, they are primal and dual optimal. This concludes the proof. □

2.1.3 Objective functions on $\mathbf{R}^n_{\geq 0}$

Continuity Let a function $f : \mathbf{R}^n \to \mathbf{R}$ be defined on a set $\mathcal{S} \subseteq \mathbf{R}^n$. The function f is *continuous* in $\boldsymbol{x} \in \mathcal{S}$ if for each $\epsilon > 0$, there is a $\delta > 0$ such that the following implication holds for all $\boldsymbol{y} \in \mathcal{S}$:

$$|\boldsymbol{x} - \boldsymbol{y}| < \delta \Rightarrow |f(\boldsymbol{x}) - f(\boldsymbol{y})| < \epsilon. \tag{2.21}$$

The function f is continuous on $\mathcal{S}$ if it is continuous in each point $\boldsymbol{x} \in \mathcal{S}$.

Partial derivative Denote by f a function that is defined on $\mathbf{R}^n_{\geq 0}$. Let $\boldsymbol{e}_i$ denote a vector with a one in the ith position and zeros elsewhere. The *partial derivatives* are defined as

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} := \begin{cases} \lim\limits_{\epsilon \to 0} \frac{f(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) - f(\boldsymbol{x})}{\epsilon}, & \text{if } x_i > 0 \\ \lim\limits_{\epsilon \downarrow 0} \frac{f(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) - f(\boldsymbol{x})}{\epsilon}, & \text{if } x_i = 0 \end{cases} \tag{2.22}$$

provided that the limit exists. The notation $\epsilon \downarrow 0$ means *limit from above*, i.e., "$\epsilon > 0$ and $\epsilon \to 0$".

Proposition 2.2. *Consider a function f that is defined on $\mathbf{R}^n_{\geq 0}$. Assume the partial derivatives of f are defined and continuous in $\boldsymbol{x}$. Let $\epsilon > 0$ and $\boldsymbol{v} \in \mathbf{R}^n \setminus \mathbf{0}$ be such that $\boldsymbol{x} + \epsilon \boldsymbol{v} \in \mathbf{R}_{\geq 0}$. Then the* directional derivative *of f in $\boldsymbol{x}$ along $\boldsymbol{v}$ is given by*

$$\lim_{\epsilon \downarrow 0} \frac{f(\boldsymbol{x} + \epsilon \boldsymbol{v}) - f(\boldsymbol{x})}{\epsilon} = \sum_{k=1}^{n} \frac{\partial f(\boldsymbol{x})}{\partial x_k} v_k. \tag{2.23}$$

Proof. For $\epsilon > 0$, we have

$$\frac{f(\boldsymbol{x}+\epsilon\boldsymbol{v})-f(\boldsymbol{x})}{\epsilon} = \frac{\sum_{k=1}^{n}[f(\boldsymbol{x}+\epsilon\sum_{i=k}^{n}v_i\boldsymbol{e}_i)-f(\boldsymbol{x}+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)]}{\epsilon} \tag{2.24}$$

$$= \frac{\sum_{k=1}^{n}[f(\boldsymbol{x}+\epsilon v_k\boldsymbol{e}_k+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)-f(\boldsymbol{x}+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)]}{\epsilon} \tag{2.25}$$

$$= \sum_{k=1}^{n}\frac{f(\boldsymbol{x}+\epsilon v_k\boldsymbol{e}_k+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)-f(\boldsymbol{x}+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)}{\epsilon v_k}v_k. \tag{2.26}$$

Consider now the kth summand. By the mean value theorem for functions of one variable, there exists for each $\epsilon > 0$ an ϵ_k, $0 < \epsilon_k < \epsilon v_k$, such that

$$\frac{f(\boldsymbol{x}+\epsilon v_k\boldsymbol{e}_k+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)-f(\boldsymbol{x}+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)}{\epsilon v_k}v_k$$

$$= \frac{\partial f(\boldsymbol{x}+\epsilon_k\boldsymbol{e}_k+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)}{\partial x_k}v_k. \tag{2.27}$$

This holds for any $\epsilon > 0$. Thus, we have in the limit for the argument

$$\lim_{\epsilon\downarrow 0}\left|\epsilon_k\boldsymbol{e}_k+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i\right| \le \lim_{\epsilon\downarrow 0}\left|\epsilon v_k\boldsymbol{e}_k+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i\right| \tag{2.28}$$

$$= 0. \tag{2.29}$$

Thus, by the continuity of the partial derivatives,

$$\lim_{\epsilon\downarrow 0}\frac{f(\boldsymbol{x}+\epsilon v_k\boldsymbol{e}_k+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)-f(\boldsymbol{x}+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)}{\epsilon v_k}v_k \tag{2.30}$$

$$= \lim_{\epsilon\downarrow 0}\frac{\partial f(\boldsymbol{x}+\epsilon_k\boldsymbol{e}_k+\epsilon\sum_{i=k+1}^{n}v_i\boldsymbol{e}_i)}{\partial x_k}v_k \tag{2.31}$$

$$= \frac{\partial f(\boldsymbol{x})}{\partial x_k}v_k. \tag{2.32}$$

Summing over k finally yields the statement of the proposition. □

Proposition 2.3. *Denote by f a convex function that is defined on $\mathbf{R}^n_{\geq 0}$. Assume the partial derivatives of f are defined and continuous on $\mathbf{R}^n_{\geq 0}$ with the possible exception that if $x_i = 0$,*

$$\frac{f(\boldsymbol{x}+\epsilon\boldsymbol{e}_i)-f(\boldsymbol{x})}{\epsilon} \xrightarrow{\epsilon\downarrow 0} -\infty. \tag{2.33}$$

Then

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} = 0, \qquad \forall i : x_i > 0 \tag{2.34}$$

$$\frac{\partial f(\boldsymbol{x})}{\partial x_i} \geq 0, \qquad \forall i : x_i = 0 \tag{2.35}$$

are necessary and sufficient conditions for $\boldsymbol{x}$ to minimize f over $\mathbf{R}^n_{\geq 0}$.

Proof. Sufficiency. Assume $\boldsymbol{x} \in \mathbf{R}^n_{\geq 0}$ fulfills the conditions (2.34) and (2.35). This implies by assumption that the partial derivatives of f are defined and continuous in $\boldsymbol{x}$. Denote by $\boldsymbol{y} \in \mathbf{R}_{\geq 0}$ an arbitrary point. By convexity of f,

$$\theta f(\boldsymbol{y}) + (1-\theta) f(\boldsymbol{x}) \geq f[\theta \boldsymbol{y} + (1-\theta)\boldsymbol{x}], \qquad 0 < \theta < 1. \tag{2.36}$$

By rearranging the terms we get

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \geq \frac{f[\theta \boldsymbol{y} + (1-\theta)\boldsymbol{x}] - f(\boldsymbol{x})}{\theta}. \tag{2.37}$$

This inequality holds for any $0 < \theta < 0$ and in particular when θ approaches zero. Passing to the limit gives

$$\lim_{\theta \downarrow 0} \frac{f[\theta \boldsymbol{y} + (1-\theta)\boldsymbol{x}] - f(\boldsymbol{x})}{\theta} = \lim_{\theta \downarrow 0} \frac{f[\boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})] - f(\boldsymbol{x})}{\theta} \tag{2.38}$$

$$= \sum_i \frac{\partial f(\boldsymbol{x})}{\partial x_i}(y_i - x_i) \tag{2.39}$$

where the last line follows since all partial derivatives of f are by assumption defined and continuous in $\boldsymbol{x}$ and consequently, Proposition 2.2 applies. Substituting the right-hand side of (2.37) by (2.39) yields

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \geq \sum_i \frac{\partial f(\boldsymbol{x})}{\partial x_i}(y_i - x_i). \tag{2.40}$$

Consider now the summands on the right-hand side. For $x_i > 0$, $\frac{\partial f(\boldsymbol{x})}{\partial x_i} = 0$ by condition (2.34). For $x_i = 0$, since $\boldsymbol{y} \in \mathbf{R}_{\geq 0}$, $y_i - x_i = y_i > 0$ and by condition (2.35), $\frac{\partial f(\boldsymbol{x})}{\partial x_i} \geq 0$. Thus, each summand and thereby the sum is greater or equal to zero and consequently

$$f(\boldsymbol{y}) - f(\boldsymbol{x}) \geq 0. \tag{2.41}$$

Since this holds for any $\boldsymbol{y} \in \mathbf{R}_{\geq 0}$, $\boldsymbol{x}$ minimizes f.

Necessity. Denote by $\boldsymbol{y} \in \mathbf{R}_{\geq 0}$ an arbitrary point. Assume that $\boldsymbol{x}$ minimizes f and assume for now that the partial derivatives of f are defined in $\boldsymbol{x}$. This implies

$$0 \leq f[\theta \boldsymbol{y} + (1-\theta)\boldsymbol{x}] - f(\boldsymbol{x}) \tag{2.42}$$

$$= f[\boldsymbol{x} + \theta(\boldsymbol{y} - \boldsymbol{x})] - f(\boldsymbol{x}), \qquad 0 < \theta < 1 \tag{2.43}$$

Dividing by θ and passing to the limit, we get

$$0 \leq \sum_i \frac{\partial f(\boldsymbol{x})}{\partial x_i}(y_i - x_i). \tag{2.44}$$

This holds for any $\boldsymbol{y} \in \mathbf{R}^n_{\geq 0}$, in particular for $\boldsymbol{y} = \boldsymbol{x} + \epsilon \boldsymbol{e}_i$, $\epsilon > 0$. With $\boldsymbol{y}$ so defined, the last inequality becomes

$$0 \leq \epsilon \frac{\partial f(\boldsymbol{x})}{\partial x_i} \tag{2.45}$$

which implies since $\epsilon > 0$

$$0 \leq \frac{\partial f(\boldsymbol{x})}{\partial x_i}. \tag{2.46}$$

Thus for all i, the partial derivative of f in $\boldsymbol{x}$ has to be greater or equal to zero, which shows the necessity of condition (2.35). If $x_i > 0$, for a small enough positive ϵ, $\boldsymbol{y} = \boldsymbol{x} - \epsilon \boldsymbol{e}_i$ is in $\mathbf{R}^n_{\geq 0}$. For $\boldsymbol{y}$ so defined, (2.44) becomes

$$0 \leq -\epsilon \frac{\partial f(\boldsymbol{x})}{\partial x_i}. \tag{2.47}$$

Since ϵ is positive, this implies

$$0 \geq \frac{\partial f(\boldsymbol{x})}{\partial x_i}. \tag{2.48}$$

Thus, for $x_i > 0$, (2.46) and (2.48) hold simultaneously, which is only possible if $\frac{\partial f(\boldsymbol{x})}{\partial x_i} = 0$. This shows the necessity of condition (2.34).

It remains to show that if $\boldsymbol{x}$ minimizes f, then the partial derivative of f is indeed defined in $\boldsymbol{x}$. To this end, assume the contrary, i.e., for some i, $x_i = 0$ and

$$\frac{f(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) - f(\boldsymbol{x})}{\epsilon} \xrightarrow{\epsilon \downarrow 0} -\infty. \tag{2.49}$$

For small enough positive ϵ, this implies

$$\frac{f(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) - f(\boldsymbol{x})}{\epsilon} < 0 \tag{2.50}$$

thus $f(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) < f(\boldsymbol{x})$, and $\boldsymbol{x}$ cannot minimize f. This concludes the proof. □

Proposition 2.4. *Consider a convex optimization problem with the objective function f_0 being defined on $\mathbf{R}^n_{\geq 0}$. Assume $\frac{\partial f_0(\boldsymbol{x})}{\partial x_i}$ is defined and continuous on $\mathbf{R}^n_{\geq 0}$ with the possible exception that for $x_i = 0$,*

$$\frac{f_0(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) - f_0(\boldsymbol{x})}{\epsilon} \xrightarrow{\epsilon \downarrow 0} -\infty. \tag{2.51}$$

Assume further that strong duality holds. Then the following conditions are necessary and sufficient for a feasible point $\boldsymbol{x}$ to be optimal. There exist $\boldsymbol{\lambda}, \boldsymbol{\nu}$ such that

$$\lambda_i \geq 0, \qquad i = 1, \ldots, p \tag{2.52}$$

$$\lambda_i f_i(\boldsymbol{x}) = 0, \qquad i = 1, \ldots, p \tag{2.53}$$

$$\frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})}{\partial x_i} = 0, \qquad \forall i : x_i > 0 \tag{2.54}$$

$$\frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})}{\partial x_i} \geq 0, \qquad \forall i : x_i = 0. \tag{2.55}$$

In particular, if $\boldsymbol{x}$ is optimal, then all partial derivations of f_0 are defined and continuous in $\boldsymbol{x}$.

Proof. Conditions (2.54) and (2.55) guarantee by Proposition 2.3 that $\boldsymbol{x}$ minimizes the Lagrangian $L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$. The statement of the proposition now follows from Proposition 2.1. □

2.1.4 Convex problem with affine constraints

The optimization problems that we consider in this work are of the form

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_i(\boldsymbol{x}) \leq 0, \quad i = 1, \ldots, p \\ & h_i(\boldsymbol{x}) = 0, \quad i = 1, \ldots, q \end{aligned} \tag{2.56}$$

where the objective function f_0 is a convex function and where both the functions f_i in the inequality constraints and the functions h_i in the equality constraints are affine functions.

Slater's condition [19, Section 5.2.3]. *Slater's condition* for convex optimization problems with affine constraints is as follows:

> There exists a feasible point $\boldsymbol{x}$ in $\mathbf{relint}\,\mathcal{D} = \mathbf{relint}\,\mathbf{dom}\,f_0$.

If Slater's condition is fulfilled, strong duality hold. The equality in the condition holds since the constraints are affine and have therefore the whole $\mathbf{R}^n$ as domain. Thus

$$\mathcal{D} = \mathbf{dom}\,f_0 \cap \bigcap_{i=1}^{p} \mathbf{dom}\,f_i \cap \bigcap_{i=1}^{q} \mathbf{dom}\,h_i \tag{2.57}$$

$$= \mathbf{dom}\,f_0. \tag{2.58}$$

Proposition 2.5. *Denote by f_0 a convex function that is defined on $\mathbf{R}^n_{\geq 0}$. Consider the optimization problem*

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{w}^T\boldsymbol{x} - E \leq 0 \\ & \mathbf{1}^T\boldsymbol{x} - 1 = 0 \end{aligned} \tag{2.59}$$

where $\boldsymbol{w}$ is a vector with positive entries and where $(\cdot)^T$ denotes transposition. *E is a positive real number. Denote by $w_{\min}$ the smallest entry of $\boldsymbol{w}$. Assume $E > w_{\min}$. Then strong duality holds.*

Proof. Strong duality holds if Slater's condition is fulfilled. Since the constraints are affine, Slater's condition is fulfilled if there is a feasible point in the relative interior of the domain of the problem. It is given by

$$\mathbf{relint}\,\mathcal{D} = \mathbf{relint}\,\mathbf{dom}\,f_0 \tag{2.60}$$

$$= \mathbf{relint}\,\mathbf{R}^n_{\geq 0} \tag{2.61}$$

$$= \mathbf{R}^n_{>0}. \tag{2.62}$$

Denote by m the index of $w_{\min}$, i.e., $w_m = w_{\min}$ and denote by $w_{\max}$ the greatest entry of $\boldsymbol{w}$. For some $\theta > 0$ define

$$\boldsymbol{x} := (1-\theta)\boldsymbol{e}_m + \theta\frac{1}{n}\mathbf{1}. \tag{2.63}$$

Clearly, for any $0 < \theta < 1$, $\boldsymbol{x} \in \mathbf{R}^n_{>0} = \mathbf{relint}\, \mathcal{D}$, furthermore,

$$\sum_i x_i = (1-\theta) \cdot 1 + \theta n \frac{1}{n} = 1, \tag{2.64}$$

i.e., the equality constraint is fulfilled. Define $\epsilon = E - w_{\min}$. By assumption, $\epsilon > 0$. Assign $\theta = \frac{\epsilon}{w_{\max}}$. Note that $\theta > 0$. Now,

$$\begin{aligned} \boldsymbol{w}^T \boldsymbol{x} - E &= (1-\theta) w_{\min} + \theta \frac{\boldsymbol{w}^T \mathbf{1}}{n} - E && (2.65)\\ &\leq w_{\min} + \theta w_{\max} - E && (2.66)\\ &= w_{\min} + \epsilon - E && (2.67)\\ &= w_{\min} + E - w_{\min} - E && (2.68)\\ &= 0. && (2.69) \end{aligned}$$

Thus, the inequality constraint is fulfilled. Altogether, $\boldsymbol{x}$ is feasible and lies in the relative interior of the domain of the problem. Since the constraints are affine, Slater's condition is therefore fulfilled and since the problem is convex, this shows that strong duality holds. This concludes the proof. □

2.2 Information theory

Probability mass function A vector $\boldsymbol{p} \in \mathbf{R}^n$ is a *probability mass function* (pmf), if

$$p_i \geq 0,\ i = 1, \ldots, n \quad \text{and} \quad \sum_i p_i = 1. \tag{2.70}$$

Shorthand, we write for the first condition $\boldsymbol{p} \geq \mathbf{0}$ and for the second condition $\mathbf{1}^T \boldsymbol{p} = 1$.

Logarithm The logarithm of a non-negative real number x to the base e is called the *natural logarithm* and we denote it by $\log x$. The logarithm of x to the base 2 is called the *binary logarithm* and we denote it by $\log_2 x$.

Relative entropy Denote by $\boldsymbol{x}$ and $\boldsymbol{y}$ two non-negative vectors with n entries, i.e., $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{R}^n_{\geq 0}$. We define the *relative entropy* of $\boldsymbol{x}$ and $\boldsymbol{y}$ as

$$\mathbb{D}(\boldsymbol{x} \| \boldsymbol{y}) := \sum_i x_i \log \frac{x_i}{y_i} \tag{2.71}$$

where for any non-negative real number a,

$$0 \log \frac{0}{a} := 0 \tag{2.72}$$

and for any positive real number a,

$$a \log \frac{a}{0} := \infty. \tag{2.73}$$

With this definition, the domain of $\mathbb{D}$ is $\mathbf{R}^n \times \mathbf{R}^n$. Relative entropy $\mathbb{D}(\boldsymbol{x}\|\boldsymbol{y})$ is convex in $\boldsymbol{x}$ and convex in $\boldsymbol{y}$. This can be shown along the lines of the proof of [24, Theorem 2.7.2].

Proposition 2.6. *Assume all entries of $\boldsymbol{y}$ are positive. Then the partial derivative of $\mathbb{D}(\boldsymbol{x}\|\boldsymbol{y})$ is for $x_i > 0$ given by*

$$\frac{\partial\, \mathbb{D}(\boldsymbol{x}\|\boldsymbol{y})}{\partial x_i} = \log \frac{x_i}{y_i} + 1 \tag{2.74}$$

and for $x_i = 0$, we have

$$\frac{\mathbb{D}(\boldsymbol{x} + \epsilon \boldsymbol{e}_k\|\boldsymbol{y}) - \mathbb{D}(\boldsymbol{x}\|\boldsymbol{y})}{\epsilon} \xrightarrow{\epsilon \downarrow 0} -\infty. \tag{2.75}$$

In particular, $\mathbb{D}(\boldsymbol{x}\|\boldsymbol{y})$ fulfills as a function of $\boldsymbol{x}$ the conditions on the objective function f in Proposition 2.3 and the conditions on the objective function f_0 in Proposition 2.4.

Proof. By applying basic differentiation rules, we get for $x_i > 0$

$$\frac{\partial\, \mathbb{D}(\boldsymbol{x}\|\boldsymbol{y})}{\partial x_i} = \log \frac{x_i}{y_i} + 1. \tag{2.76}$$

Since by assumption $y_i > 0$, the right-hand side is well-defined.

For $x_i = 0$, we have

$$\frac{(0+\epsilon) \log \frac{0+\epsilon}{y_i} - 0 \log \frac{0}{y_i}}{\epsilon} = \frac{\epsilon \log \frac{\epsilon}{y_i}}{\epsilon} \tag{2.77}$$

$$= \log \frac{\epsilon}{y_i} \tag{2.78}$$

$$\xrightarrow{\epsilon \downarrow 0} -\infty. \tag{2.79}$$

□

Information inequality Denote by $\boldsymbol{p}$ and $\boldsymbol{q}$ two pmfs. Then

$$\mathbb{D}(\boldsymbol{p}\|\boldsymbol{q}) \geq 0, \qquad \text{with equality if and only if } \boldsymbol{p} = \boldsymbol{q}. \tag{2.80}$$

This property of relative entropy is called the *information inequality*, see [24, Theorem 2.6.3].

Log sum inequality Denote by $\boldsymbol{x}$ and $\boldsymbol{y}$ two non-negative vectors. Then

$$\sum_i x_i \log \frac{x_i}{y_i} \geq \Big(\sum_i x_i\Big) \log \frac{\sum_i x_i}{\sum_i y_i},$$

$$\text{with equality if and only if } \frac{x_i}{y_i} = K, \quad i = 1, \ldots, n \tag{2.81}$$

where K is some constant. This property is called the *log sum inequality*. See [24, Theorem 2.7.1].

Entropy Denote by $\boldsymbol{p}$ a pmf. *Entropy* is defined as

$$\mathbb{H}(\boldsymbol{p}) := -\sum_i p_i \log p_i \tag{2.82}$$

where

$$0 \log 0 := 0. \tag{2.83}$$

Entropy $\mathbb{H}(\boldsymbol{p})$ is concave in $\boldsymbol{p}$ [24, Theorem 2.7.3]. Note that, since

$$\mathbb{H}(\boldsymbol{p}) = -\,\mathbb{D}(\boldsymbol{p} \| \mathbf{1}) \tag{2.84}$$

Proposition 2.6 also applies to the negative of the entropy function.

Mutual information A *discrete memoryless channel* (dmc) is described by a *transition matrix* $\boldsymbol{H}$

$$\boldsymbol{H} = \begin{pmatrix} h_{11} & \cdots & h_{1n} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mn} \end{pmatrix} = (\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n). \tag{2.85}$$

Each column vector $\boldsymbol{h}_i$ is a pmf. The *input pmf* $\boldsymbol{p}$ and the *output pmf* $\boldsymbol{r}$ of the dmc relate as

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{p}. \tag{2.86}$$

The *mutual information* between input and output of the channel is given by

$$\mathbb{I}_{\boldsymbol{H}}(\boldsymbol{p}) = \mathbb{H}(\boldsymbol{r}) - \sum_i p_i \, \mathbb{H}(\boldsymbol{h}_i) \tag{2.87}$$

If the transition matrix $\boldsymbol{H}$ is clear from the context, we will simply write $\mathbb{I}(\boldsymbol{p})$. In the form of (2.87), mutual information is well-defined for any input pmf $\boldsymbol{p}$. Using the definition of entropy, mutual information can be rewritten as follows.

$$\begin{aligned} \mathbb{I}(\boldsymbol{p}) &= \mathbb{H}(\boldsymbol{r}) - \sum_i p_i \, \mathbb{H}(\boldsymbol{h}_i) && (2.88) \\ &= -\sum_j r_j \log r_j + \sum_i p_i \sum_j h_{ji} \log h_{ji} && (2.89) \\ &= -\sum_i p_i \sum_j h_{ji} \log r_j + \sum_i p_i \sum_j h_{ji} \log h_{ji} && (2.90) \\ &= \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j}. && (2.91) \end{aligned}$$

Since

$$r_j = \sum_{i=1}^{n} p_i h_{ji} \tag{2.92}$$

we conclude that if $r_j = 0$ then $p_i h_{ji} = 0$. Thus, (2.91) is well-defined for any transition matrix $\boldsymbol{H}$ and any input pmf $\boldsymbol{p} \in \mathbf{R}^n_{\geq 0}$. Mutual information $\mathbb{I}(\boldsymbol{p})$ is concave in $\boldsymbol{p}$ [24, Theorem 2.7.4].

Proposition 2.7. *The partial derivatives of $\mathbb{I}(\boldsymbol{p})$ on $\mathbf{R}^n_{\geq 0}$ are defined and given by*

$$\frac{\partial\, \mathbb{I}(\boldsymbol{p})}{\partial p_k} = \sum_j h_{jk} \log \frac{h_{jk}}{r_j} - 1 \tag{2.93}$$

with the possible exception that for $p_k = 0$,

$$\frac{\mathbb{I}(\boldsymbol{p} + \epsilon \boldsymbol{e}_k) - \mathbb{I}(\boldsymbol{p})}{\epsilon} \xrightarrow{\epsilon \downarrow 0} \infty \tag{2.94}$$

In particular, the negative mutual information fulfills the conditions on the objective function f in Proposition 2.3 and the conditions on the objective function f_0 in Proposition 2.4.

Proof.

Case 1: $p_k > 0$. For $p_k > 0$, we get by applying basic differentiation rules to $\mathbb{I}(\boldsymbol{p})$

$$\frac{\partial\, \mathbb{I}(\boldsymbol{p})}{\partial p_k} = \sum_j h_{jk} \log \frac{h_{jk}}{r_j} - 1. \tag{2.95}$$

Since $r_j = 0$ implies $p_k h_{jk} = 0$, and since $p_k > 0$, $r_j = 0$ implies $h_{jk} = 0$. Thus, for $p_k > 0$, the partial derivatives are well-defined and given by (2.95) for any dmc.

Case 2: $p_k = 0$. First, we note that

$$r_j = \sum_i h_{ji} p_i = \sum_{i \neq k} h_{ji} p_i. \tag{2.96}$$

We now have

$$\begin{aligned}
&\frac{\mathbb{I}(\boldsymbol{p} + \epsilon \boldsymbol{e}_k) - \mathbb{I}(\boldsymbol{p})}{\epsilon} \\
&= \frac{1}{\epsilon} \left[\sum_{i \neq k} p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j + \epsilon h_{jk}} + \epsilon \sum_j h_{jk} \log \frac{h_{jk}}{r_j + \epsilon h_{jk}} - \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j} \right] \qquad (2.97) \\
&= \sum_j h_{jk} \log \frac{h_{jk}}{r_j + \epsilon h_{jk}} - \frac{1}{\epsilon} \sum_i p_i \sum_j h_{ji} \log \frac{r_j + \epsilon h_{jk}}{r_j}. \qquad (2.98)
\end{aligned}$$

We have for the second term

$$\frac{1}{\epsilon}\sum_i p_i \sum_j h_{ji} \log \frac{r_j + \epsilon h_{jk}}{r_j} = \frac{1}{\epsilon}\sum_i p_i \sum_{j:r_j>0} h_{ji} \log \frac{r_j + \epsilon h_{jk}}{r_j} \tag{2.99}$$

$$= \frac{1}{\epsilon}\sum_i p_i \sum_{j:r_j>0} h_{ji} \log\Big(1 + \frac{\epsilon h_{jk}}{r_j}\Big) \tag{2.100}$$

$$= \frac{1}{\epsilon}\sum_i p_i \sum_{j:r_j>0} h_{ji} \Big(\frac{\epsilon h_{jk}}{r_j} - \frac{1}{2}\Big(\frac{\epsilon h_{jk}}{r_j}\Big)^2 + \cdots\Big) \tag{2.101}$$

$$= \sum_i p_i \sum_{j:r_j>0} h_{ji} \frac{h_{jk}}{r_j} + \mathcal{O}(1) \tag{2.102}$$

$$= \sum_{j:r_j>0} h_{jk} \frac{\sum_i p_i h_{ji}}{r_j} + \mathcal{O}(1) \tag{2.103}$$

$$= \sum_{j:r_j>0} h_{jk} + \mathcal{O}(1) \tag{2.104}$$

where we used in (2.101) the Taylor expansion of $\log(1+x)$ in $x = 0$. We now have to distinguish between two subcases.

Case 2.1: $r_j = 0 \Rightarrow h_{jk} = 0$. If for each j with $r_j = 0$ we also have $h_{jk} = 0$, we have

$$\frac{\mathbb{I}(\boldsymbol{p} + \epsilon \boldsymbol{e}_k) - \mathbb{I}(\boldsymbol{p})}{\epsilon} = \sum_j h_{jk} \log \frac{h_{jk}}{r_j + \epsilon h_{jk}} - \sum_{j:r_j>0} h_{jk} + \mathcal{O}(1) \tag{2.105}$$

$$= \sum_j h_{jk} \log \frac{h_{jk}}{r_j + \epsilon h_{jk}} - \sum_j h_{jk} + \mathcal{O}(1) \tag{2.106}$$

$$= \sum_j h_{jk} \log \frac{h_{jk}}{r_j + \epsilon h_{jk}} - 1 + \mathcal{O}(1) \tag{2.107}$$

$$\xrightarrow{\epsilon \downarrow 0} \sum_j h_{jk} \log \frac{h_{jk}}{r_j} - 1 \tag{2.108}$$

i.e., in this case, the partial derivative is well-defined and given by (2.95).

Case 2.2: $\exists j : r_j = 0, h_{jk} \neq 0$. Assume there is a j with $r_j = 0$ and $h_{jk} > 0$. In this case, we have

$$\frac{\mathbb{I}(\boldsymbol{p} + \epsilon \boldsymbol{e}_k) - \mathbb{I}(\boldsymbol{p})}{\epsilon} = \sum_j h_{jk} \log \frac{h_{jk}}{r_j + \epsilon h_{jk}} - \sum_{j:r_j>0} h_{jk} + \mathcal{O}(1) \tag{2.109}$$

$$\xrightarrow{\epsilon \downarrow 0} \infty. \tag{2.110}$$

This concludes the proof. □

2.3 References

Definitions and notation in Section 2.1 are taken from Boyd and Vandenberghe [19]. Proposition 2.1 is stated in El Gamal and Kim [31, Appendix E]. A version of Proposition 2.1 for differentiable objective functions with open domain is stated in [19, Section 5.5.3]. Proposition 2.2 is stated for differentiable functions with open domain in any standard textbook on advanced analysis, see for example Edwards [30, Theorem 2.1] or Munkres [63, Theorem 5.1]. In [39, Theorem 4.4.1], Gallager states Proposition 2.3 for functions with the probability simplex as domain. The definitions in Section 2.2 are from Cover and Thomas [24]. Gallager uses mutual information in the form of (2.91) in [40, Equation (8.73)]. The properties of mutual information as stated in Proposition 2.7 are claimed without proof by Gallager in [39, Page 92].

3 Matching channels

In Shannon theory, a common task is to maximize or minimize functionals of pmfs. For example, to calculate the capacity C of a dmc, we have to maximize the mutual information $\mathbb{I}(\boldsymbol{p})$ between input and output of the channel over all input pmfs $\boldsymbol{p}$, i.e.,

$$\mathsf{C} = \max_{\text{pmf } \boldsymbol{p}} \mathbb{I}(\boldsymbol{p}). \tag{3.1}$$

The pmf that maximizes the mutual information is called the *capacity-achieving pmf* and we denote it by $\boldsymbol{p}^*$. To be able to communicate at maximum rate over the channel, the sequence of channel input symbols has to resemble a sequence of symbols that are *independent and identically distributed (iid)* according to the capacity achieving pmf $\boldsymbol{p}^*$. Thus, in a digital communication system, such a sequence has to be generated somehow. How this can optimally be done is the topic of this thesis. In this chapter, we first show how the class of dyadic pmfs can be generated by prefix-free codes. We then solve the problem of minimizing the relative entropy $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x})$ for a given target vector $\boldsymbol{x}$ over all dyadic pmfs $\boldsymbol{d}$. After that, we show how this result can be used to maximize over all dyadic pmfs the entropy rate of a noiseless channel and the mutual information of a dmc.

3.1 Prefix-free matchers and dyadic pmfs

In a digital communication system, the interface between source and channel coding is a stream of iid bits. The bits are *equiprobable*, i.e., both the probability that a certain bit takes the value zero and the probability that it takes the value one is one half. We call such a sequence of bits a *fair bit stream*. We will now introduce a technique that allows us to reversibly transform a fair bit stream into a sequence of symbols that are iid according to a non-uniform pmf. To this end, we use the concept of prefix-free codes.

Full prefix-free codes

Prefix-free codes are a well-studied subject, we will only give a very short introduction. Details can be found for example in Cover and Thomas [24, Chapter 5] or Gallager [40, Section 2.3]. A *prefix-free code* is a set of codewords over a finite alphabet where no codeword is a prefix of another codeword. In this work, we exclusively consider binary prefix-free codes. A *full prefix-free code* is a prefix-free code where no codeword can be added to the set without destroying the prefix-free property. Equivalently, a prefix-free code is full if no codeword in the set can be shortened without destroying the prefix-free property. We denote by ℓ_i the length (the number of bits) of the ith codeword in a prefix-free code. The following proposition characterizes prefix-free codes by their codeword lengths. It is called the *Kraft inequality* after Kraft [50].

Proposition 3.1. *Every prefix-free code satisfies the following inequality:*

$$\sum_i 2^{-\ell_i} \leq 1 \tag{3.2}$$

with equality if and only if the code is full. Conversely, if a set of lengths $\{\ell_i\}_{i=1}^n$ fulfills the inequality, then there exists a prefix-free code with lengths ℓ_i and if it fulfills the inequality with equality, then there exists a full prefix-free code with lengths ℓ_i.

Prefix-free matchers

According to the Kraft inequality, for a full prefix-free code, the vector

$$\boldsymbol{d} = (2^{-\ell_1}, \ldots, 2^{-\ell_n})^T \tag{3.3}$$

is a pmf. We call it the *induced pmf* of the code. By parsing the stream by a full prefix-free code, a non-uniform pmf can be generated. For example, consider the set of symbols $\{\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}\}$. Then the mapping

$$\begin{aligned} 1 &\mapsto \mathtt{0} \\ 01 &\mapsto \mathtt{1} \\ 001 &\mapsto \mathtt{2} \\ 000 &\mapsto \mathtt{3} \end{aligned} \tag{3.4}$$

generates the pmf $(2^{-1}, 2^{-2}, 2^{-3}, 2^{-3})^T$ over the set $\{\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}\}$ when a fair bit stream is parsed by the full prefix-free code $\{1, 01, 001, 000\}$. We call a device that implements this procedure a *prefix-free matcher*. The pmf generated by a full prefix-free code is exactly the corresponding induced pmf.

Dyadic pmfs

Different full prefix-free codes can induce the same pmf, e.g., by swapping 0 and 1 in the binary codewords of the example above, we get a different full prefix-free code, but the induced pmf remains the same. Furthermore, by adding virtual codewords of length infinity to a full prefix-free code, we can add entries with value zero to the induced pmf. The full prefix-free code from our example can also be used to generate the pmf

$$(2^{-1}, 2^{-2}, 2^{-3}, 2^{-3}, 2^{-\infty})^T. \tag{3.5}$$

Since $2^{-\infty} = 0$, the Kraft inequality continues to hold with equality. In addition, we also allow trivial pmfs where one symbol is generated with probability one and all other symbols with probability zero. The corresponding codeword lengths are zero and infinity, respectively. Note that the Kraft inequality is also fulfilled with equality for trivial pmfs. We are mainly interested in the pmfs that can be generated by full prefix-free codes. By the Kraft inequality, the set of such pmfs is given by

$$\Big\{\boldsymbol{d} \;\Big|\; \sum_i d_i = 1 \text{ and } d_i = 2^{-\ell_i}, \ell_i \in \mathbf{N}_0,\ i = 1, \ldots, n\Big\}. \tag{3.6}$$

We call it the set of *dyadic pmfs*. Most of the time, we will not specify a full prefix-free code that generates a particular dyadic pmf, however, we should keep in mind that such a full prefix-free code can easily be constructed.

3.2 Geometric Huffman coding

We define a *non-negative vector* as a vector with non-negative real entries and at least one positive entry. Consider now a non-negative vector $\boldsymbol{x}$. The objective is to approximate $\boldsymbol{x}$ by a dyadic pmf by solving the matching problem

$$\underset{\text{dyadic } \boldsymbol{d}}{\text{minimize}} \quad \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}). \tag{3.7}$$

We now propose an algorithm that constructs a dyadic pmf based on the non-negative vector $\boldsymbol{x}$. The reader should be familiar with *Huffman coding* [45], see for example [40, Section 2.5.3] for a detailed explanation. Our algorithm consists in constructing a prefix-free tree similar to the Huffman procedure, but with different updating rules. The solution of Problem (3.7) is then the dyadic pmf induced by the constructed tree. The updating rules are as follows. Suppose in some step in the algorithm, $\boldsymbol{x}$ has n entries and is sorted, i.e., $x_1 \geq x_2 \geq \cdots \geq x_n$. The first rule states how this target vector with n entries can be turned into a target vector with $n-1$ entries.

Rule 1: Updating $\boldsymbol{x}$

The two smallest entries x_n and x_{n-1} are replaced by x' with the following rule.

$$x' = \begin{cases} x_{n-1}, & \text{if } x_{n-1} \geq 4x_n \\ 2\sqrt{x_{n-1}x_n}, & \text{if } x_{n-1} < 4x_n. \end{cases} \tag{3.8}$$

The second rule states how the prefix-free tree is constructed.

Rule 2: Updating the binary tree

The entries 1 to n correspond to n root nodes of n trees.

if $x_{n-1} \geq 4x_n$: remove the whole tree that ends at node n and associate probability zero with the leafs.

if $x_{n-1} < 4x_n$: join node n and node $n-1$ in a parent node.

Since it involves a *geometric mean*, we call this method *geometric Huffman coding* (GHC) and write $\boldsymbol{d} = \text{GHC}(\boldsymbol{x})$, where $\boldsymbol{d}$ is the dyadic pmf that is induced by the prefix-free tree constructed by GHC. In Subsection 3.2.3, we will show that GHC actually finds the optimal dyadic pmf of Problem (3.7). GHC has the same complexity as Huffman coding, which is $\mathcal{O}(n \log n)$ [23, Chapter 16.3]. An implementation of GHC in MATLAB is displayed in Algorithm 1 and can be downloaded at our website [8].

Algorithm 1.(GHC)

```
function d = ghc(x)

n = length(x);
indices = zeros(n-1,2);
cut_tree = false(n-1,1);

for k=1:n-1
    [~,index] = sort(x,'descend');
    m = index(end-1:end);
    if(4*(x(m(2)))<=x(m(1)))
        x = [x(1:m(2)-1);x(m(2)+1:end)];
        cut_tree(k)=true;
    else
        m = sort(m,'ascend');
        x = [x(1:m(1)-1);x(m(1)+1:m(2)-1);x(m(2)+1:end);2*sqrt(x(m(1))*x(m(2)))];
    end
    indices(k,:)=m;
end


L = zeros(n,1);
for k=1:n-1
    m = indices(n-k,:);
    if(cut_tree(n-k))
        L = [L(1:m(2)-1);Inf;L(m(2):k);L(k+2:end)];
    else
        L = [L(1:m(1)-1);L(k)+1;L(m(1):m(2)-2);L(k)+1;L(m(2)-1:k-1);L(k+2:end)];
    end
end

d = 2.^(-L);
```

3.2.1 Example

We now illustrate how GHC works by an example. Consider 5 symbols with the target pmf

$$\boldsymbol{t} = (0.328,\ 0.32,\ 0.22,\ 0.11,\ 0.022)^T. \tag{3.9}$$

Throughout this chapter, we will denote non-negative target vectors by $\boldsymbol{x}$ and target pmfs by $\boldsymbol{t}$. Our aim is now to use GHC to construct a dyadic approximation of the pmf $\boldsymbol{t}$. See Figure 3.1 for an illustration.

Step 1 First, we compare the two smallest entries of $\boldsymbol{t}$. We have

$$\frac{t_4}{t_5} = 5 \tag{3.10}$$

i.e., $t_4 > 4t_5$. Thus, according to Rule 1, we remove t_5 from $\boldsymbol{t}$ to continue with

$$\boldsymbol{t}' = (0.328,\ 0.32,\ 0.22,\ 0.11)^T. \tag{3.11}$$

According to Rule 2, we remove the tree ending at t_5 and associate probability zero, or equivalently, codeword length infinity with its leafs. There is only one leaf, namely t_5 itself.

Step 2 Now, the two smallest entries of $\boldsymbol{t}'$ relate as

$$\frac{t'_3}{t'_4} = 2 \tag{3.12}$$

i.e, $t'_3 < 4t'_4$. Thus, according to Rule 1, we replace them by twice their geometric mean, i.e.,

$$t'' = 2\sqrt{t'_3 t'_4} = 0.311. \tag{3.13}$$

We continue with

$$\boldsymbol{t}'' = (0.328,\ 0.32,\ 0.311). \tag{3.14}$$

In the tree, following Rule 2, we join node t'_3 and node t'_4 in the parent node t''_3.

Step 3 & 4 Steps 3 & 4 work exactly as Step 2. The resulting prefix-free tree is displayed in Figure 3.1.

Comparison to Huffman coding

Huffman coding uses the updating rule $x' = x_n + x_{n-1}$. For comparison, Huffman coding is applied to $\boldsymbol{t}$. The steps are illustrated in the right column of Figure 3.1. We write $\boldsymbol{d} = \textsc{Hc}(\boldsymbol{t})$ for the dyadic pmf induced by the Huffman code of $\boldsymbol{t}$. By reading off the codeword lengths, the induced dyadic pmfs are

$$\boldsymbol{d}_{\textsc{Ghc}} = (2^{-1},\ 2^{-2},\ 2^{-3},\ 2^{-3},\ 2^{-\infty})^T \text{ and } \boldsymbol{d}_{\textsc{Hc}} = (2^{-2},\ 2^{-2},\ 2^{-2},\ 2^{-3},\ 2^{-3})^T. \tag{3.15}$$

The relative entropies are

$$\mathbb{D}(\boldsymbol{d}_{\textsc{Ghc}} \| \boldsymbol{t}) \approx 0.0944 \quad \text{and} \quad \mathbb{D}(\boldsymbol{d}_{\textsc{Hc}} \| \boldsymbol{t}) \approx 0.1355. \tag{3.16}$$

The relative entropy resulting from Ghc is smaller than the one that results from Huffman coding. Since Ghc assigns zero to t_5, one may want to manually assign probability

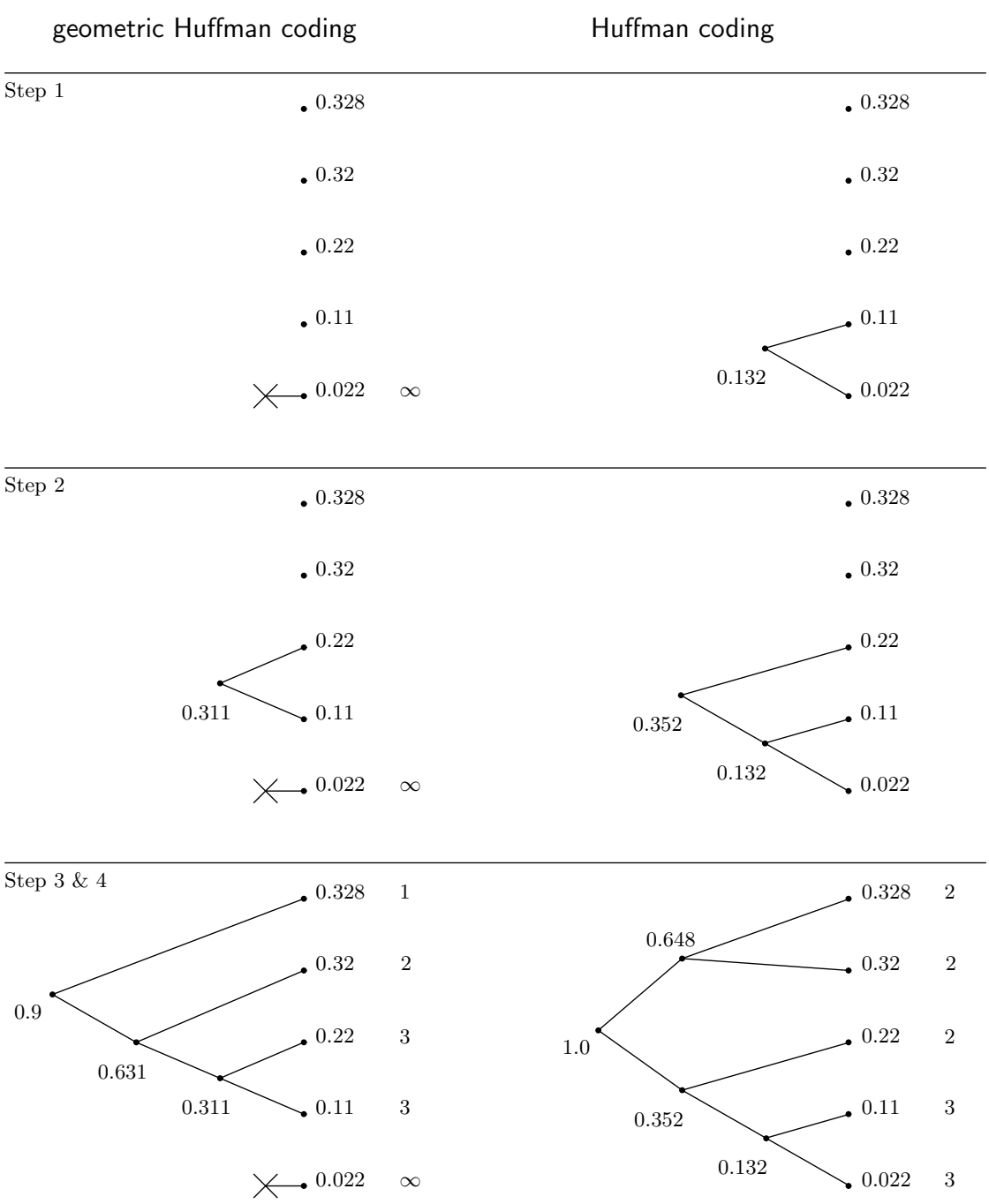


Figure 3.1: For $\boldsymbol{t} = (0.328,\ 0.32,\ 0.22,\ 0.11,\ 0.022)^T$, the left column displays GHC. The right column illustrates Huffman coding.

zero to t_5 and then apply Huffman coding to $(t_1, \dots, t_4)^T$. The resulting pmf and the resulting relative entropy are respectively given by

$$\boldsymbol{d}_{\text{HC}'} = (2^{-2},\ 2^{-2},\ 2^{-2},\ 2^{-2},\ 2^{-\infty})^T, \quad \mathbb{D}(\boldsymbol{d}_{\text{HC}'} \| \boldsymbol{t}) \approx 0.1076. \tag{3.17}$$

While $\boldsymbol{d}_{\text{HC}'}$ slightly improves upon $\boldsymbol{d}_{\text{HC}}$, the relative entropy is still larger than the one resulting from GHC. It can be shown that Huffman coding minimizes the relative entropy $\mathbb{D}(\boldsymbol{x}\|\boldsymbol{d})$ over all dyadic pmfs $\boldsymbol{d}$. Note that this is not equivalent to minimizing $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x})$ because the relative entropy is not symmetric in its arguments [24, Section 2.3]. The corresponding relative entropies of GHC and Huffman coding are

$$\mathbb{D}(\boldsymbol{t}\|\boldsymbol{d}_{\text{GHC}}) = \infty \text{ and } \mathbb{D}(\boldsymbol{t}\|\boldsymbol{d}_{\text{HC}}) = 0.087660 \tag{3.18}$$

where in the case of GHC, the value infinity is a consequence of the probability zero assigned to entry 5. Huffman coding achieves a finite value close to zero, which illustrates that for $\mathbb{D}(\boldsymbol{t}\|\boldsymbol{d})$, Huffman coding outperforms GHC.

3.2.2 GHC assigns probability zero to values of zero

The algorithm GHC has the following property, which will be of great importance in the rest of this work.

Proposition 3.2. *Denote by $\boldsymbol{x}$ a non-negative target vector and define the dyadic pmf $\boldsymbol{d} = \text{GHC}(\boldsymbol{x})$. Then*

$$d_i = 0, \qquad \textit{whenever } x_i = 0. \tag{3.19}$$

Proof. Assume $\boldsymbol{x}$ is ordered, i.e., $x_1 \geq x_2 \geq \cdots \geq x_n$. Then, GHC assigns $d_n = 0$ if $x_{n-1} \geq 4x_n$. Since $\boldsymbol{x}$ is by assumption non-negative, for $x_n = 0$, this condition is fulfilled for any value of x_{n-1} and the statement of the proposition follows. □

3.2.3 Optimality of GHC

Proposition 3.3. *Denote by $\boldsymbol{x}$ a non-negative vector. The dyadic pmf $\boldsymbol{d} = \text{GHC}(\boldsymbol{x})$ is the optimal dyadic pmf of Problem (3.7), i.e., it minimizes $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x})$ over all dyadic pmfs $\boldsymbol{d}$.*

Proof. Denote by $\boldsymbol{x}$ some non-negative vector with n entries. The pmf $\boldsymbol{d}$ is dyadic if and only if there exist numbers $\ell_i \in \mathbf{N}_0$, such that $d_i = 2^{-\ell_i}$, $i = 1, \dots, n$, and $\sum_i 2^{-\ell_i} = 1$. Using this, we can write

$$\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}) = \sum_i d_i \log \frac{d_i}{x_i} \tag{3.20}$$

$$= \log(2) \sum_i d_i \log_2 \frac{d_i}{x_i} \tag{3.21}$$

$$= \log(2) \sum_i 2^{-\ell_i} (-\log_2 x_i - \ell_i). \tag{3.22}$$

We define $\boldsymbol{u}$ by $u_i = -\log_2 x_i$, $i = 1, \ldots, n$. Omitting the constant factor $\log 2$, our aim is thus to minimize

$$\sum_i 2^{-\ell_i}(u_i - \ell_i) \tag{3.23}$$

subject to $\ell_1, \ldots, \ell_n$ being the codeword lengths of a full prefix-free code. Based on (3.23), we now prove the optimality of GHC in a way similar to the proof given in [40, Sec. 2.5.3] for the optimality of Huffman coding.

Lemma 1. *For an optimal algorithm, $u_i > u_j$ implies $\ell_i \geq \ell_j$.*

Proof. Assume the contrary, i.e., $u_i > u_j$ and $\ell_i < \ell_j$. Consider the ith and jth terms in (3.23), i.e.,

$$2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell_j}(u_j - \ell_j). \tag{3.24}$$

By interchanging ℓ_i and ℓ_j, the term decreases:

$$\begin{aligned}
&[2^{-\ell_j}(u_i - \ell_j) + 2^{-\ell_i}(u_j - \ell_i)] - [2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell_j}(u_j - \ell_j)] && (3.25)\\
&\quad = 2^{-\ell_j}(u_i - u_j) + 2^{-\ell_i}(u_j - u_i) && (3.26)\\
&\quad = (\underbrace{2^{-\ell_i} - 2^{-\ell_j}}_{>0})(\underbrace{u_j - u_i}_{<0}) < 0 && (3.27)
\end{aligned}$$

so any code with $u_i > u_j$ and $\ell_i < \ell_j$ is not optimal. □

We now assume that the entries of $\boldsymbol{x}$ are in descending order, i.e., $x_1 \geq x_2 \geq \cdots \geq x_n$. Correspondingly, the entries of $\boldsymbol{u}$ are in ascending order, i.e., $u_1 \leq u_2 \leq \cdots \leq u_n$.

Case 1: $\ell_{n-1}, \ell_n < \infty$**.** We first consider the case when an optimal algorithm assigns finite values to the codeword lengths ℓ_n and ℓ_{n-1} of the two smallest entries x_{n-1} and x_n, which correspond to the greatest entries u_{n-1} and u_n of $\boldsymbol{u}$. We show that in this case, there is an optimal algorithm for which $\ell_n = \ell_{n-1}$.

Lemma 2. *If an optimal algorithm assigns finite values to the codeword lengths ℓ_n and ℓ_{n-1} of the two greatest entries u_{n-1} and u_n, then there is an optimal algorithm for which ℓ_n and ℓ_{n-1} are siblings, i.e., $\ell_n = \ell_{n-1}$, and in addition, no other codeword is longer than ℓ_n and ℓ_{n-1}.*

Proof. In a full prefix-free code, the sibling of the longest finite codeword is also a longest codeword. According to Lemma 1, if $u_n > u_{n-1} > u_{n-2} \geq \cdots$, an optimal algorithm assigns the two longest codewords to u_n and u_{n-1}. If only $u_n \geq u_{n-1} \geq u_{n-2} \geq \cdots$, assigning the two longest codewords to u_n and u_{n-1} does not change optimality. □

We can now use $\ell_n = \ell_{n-1}$ to rewrite (3.23):

$$\sum_{i=1}^{n} 2^{-\ell_i}(u_i - \ell_i) = \sum_{i=1}^{n-2} 2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell_{n-1}}(u_{n-1} - \ell_{n-1}) + 2^{-\ell_n}(u_n - \ell_n) \tag{3.28}$$

$$= \sum_{i=1}^{n-2} 2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell_n}(u_{n-1} + u_n - 2\ell_n) \tag{3.29}$$

$$= \sum_{i=1}^{n-2} 2^{-\ell_i}(u_i - \ell_i) + 2^{-(\ell_n - 1)}\Big[\underbrace{\Big(\frac{u_{n-1} + u_n}{2} - 1\Big)}_{=:u'} - \underbrace{(\ell_n - 1)}_{=:\ell'}\Big] \tag{3.30}$$

$$= \sum_{i=1}^{n-2} 2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell'}(u' - \ell'). \tag{3.31}$$

Thus, by combining u_n and u_{n-1} through

$$u' = \frac{u_{n-1} + u_n}{2} - 1 \tag{3.32}$$

the size n problem is reduced to a size $n-1$ problem. The updating rule for the codeword lengths $\ell' = \ell_n - 1$ corresponds to joining the nodes of u_{n-1} and u_n in a parent node.

Case 2: $\ell_n = \infty$. The optimal algorithm may assign probability zero to the greatest entry u_n, which corresponds to $\ell_n = \infty$. We thus have

$$\sum_{i=1}^{n} 2^{-\ell_i}(u_i - \ell_i) = \sum_{i=1}^{n-1} 2^{-\ell_i}(u_i - \ell_i) + 2^{-\infty}(u_n - \infty) \tag{3.33}$$

$$= \sum_{i=1}^{n-1} 2^{-\ell_i}(u_i - \ell_i) \tag{3.34}$$

where we used the convention $-0 \log 0 = 0$ and equivalently $2^{-\infty}\infty = 0$. Thus, if we assign $\ell_n = \infty$, the size n problem is reduced to a size $n-1$ problem. The corresponding updating rule for the prefix-free tree is to remove the node of u_n from the tree.

Choosing between the cases. It remains to check if it is better to assign probability zero to u_n or to combine u_n and u_{n-1}. First, assume the algorithm combines u_n and u_{n-1}. Then the contribution to the sum (3.23) is $2^{-\ell'}(u' - \ell')$. We can now assign probability zero to u_n and use the codeword of u' for u_{n-1}. The contribution of u_n to (3.23) is then zero and the contribution of u_{n-1} is $2^{-\ell'}(u_{n-1} - \ell')$. Thus, since our aim

is to minimize (3.23), doing the former is better if and only if

$$2^{-\ell'}(u_{n-1} - \ell') > 2^{-\ell'}(u' - \ell') \tag{3.35}$$
$$\Leftrightarrow (u_{n-1} - \ell') > \left(\frac{u_{n-1} + u_n}{2} - 1\right) - \ell' \tag{3.36}$$
$$\Leftrightarrow u_{n-1} > \frac{u_{n-1} + u_n}{2} - 1 \tag{3.37}$$
$$\Leftrightarrow u_{n-1} > u_n - 2. \tag{3.38}$$

We now express this condition in terms of $\boldsymbol{x}$. Recalling that $u_i = -\log_2 x_i$, the condition (3.38) becomes

$$u_{n-1} > u_n - 2 \tag{3.39}$$
$$\Leftrightarrow -\log_2 x_{n-1} > -\log_2 x_n - 2 \tag{3.40}$$
$$\Leftrightarrow \log_2 x_{n-1} < \log_2 x_n + 2 \tag{3.41}$$
$$\Leftrightarrow x_{n-1} < 4x_n. \tag{3.42}$$

The updating rule (3.32) becomes

$$u' = \frac{u_{n-1} + u_n}{2} - 1 \tag{3.43}$$
$$\Leftrightarrow -\log_2 x' = \frac{-\log_2 x_{n-1} - \log_2 x_n}{2} - 1 \tag{3.44}$$
$$\Leftrightarrow \log_2 x' = \frac{\log_2 x_{n-1} + \log_2 x_n}{2} + 1 \tag{3.45}$$
$$\Leftrightarrow x' = 2\sqrt{x_{n-1}x_n}. \tag{3.46}$$

Altogether, the optimal algorithm updates as follows.

$$x' = \begin{cases} x_{n-1}, & \text{if } x_{n-1} \geq 4x_n \\ 2\sqrt{x_{n-1}x_n}, & \text{if } x_{n-1} < 4x_n. \end{cases} \tag{3.47}$$

This is exactly the updating rule of Ghc. □

3.2.4 Optimal pmf

The set of dyadic pmfs with n entries is only a finite subset of the infinite set of pmfs with n entries. Thus, by restricting ourselves to dyadic pmfs, we get in general worse results than we would obtain when we could use any pmf. For clarity, we denote dyadic pmfs by $\boldsymbol{d}$ and arbitrary (possibly non-dyadic) pmfs by $\boldsymbol{p}$. The objective is now to quantify the penalty that results from the restriction to dyadic pmfs. To this end, we start by calculating the optimal, possibly non-dyadic pmf. Denote again by $\boldsymbol{x}$ a non-negative vector and consider the optimization problem

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{minimize}} \quad & \mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{p} \text{ is a pmf.} \end{aligned} \tag{3.48}$$

Denote by $\boldsymbol{p}^*$ an optimal pmf of (3.48) and denote by D the optimal value. The topic of this subsection is to characterize $\boldsymbol{p}^*$ and D.

Proposition 3.4. *Denote by $\boldsymbol{x}$ a non-negative vector. For the optimization problem (3.48), the optimal value D and the optimal pmf $\boldsymbol{p}^*$ are given by*

$$\mathsf{D} = -\log \sum_i x_i \tag{3.49}$$

$$\boldsymbol{p}^* = \frac{1}{\sum x_i}\boldsymbol{x} = e^{\mathsf{D}}\boldsymbol{x}. \tag{3.50}$$

Proof. We prove the statement by using the information inequality. We write $\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})$ as

$$\begin{aligned}
\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) &= \sum_i p_i \log \frac{p_i}{x_i} && (3.51)\\
&= \sum_i p_i \log \frac{p_i}{\frac{x_i}{\sum_j x_j}\sum_j x_j} && (3.52)\\
&= \sum_i p_i \log \frac{p_i}{\frac{x_i}{\sum_j x_j}} - \log \sum_j x_j. && (3.53)
\end{aligned}$$

The vector $\boldsymbol{x}/\sum_j x_j$ is a pmf, thus, by the information inequality,

$$\begin{aligned}
\sum_i p_i \log \frac{p_i}{\frac{x_i}{\sum_j x_j}} &= \mathbb{D}(\boldsymbol{p}\|\frac{\boldsymbol{x}}{\sum_j x_j}) && (3.54)\\
&\geq 0 && (3.55)\\
\Rightarrow \sum_i p_i \log \frac{p_i}{\frac{x_i}{\sum_j x_j}} - \log \sum_j x_j &\geq -\log \sum_j x_j && (3.56)
\end{aligned}$$

with equality if and only if $\boldsymbol{p} = \boldsymbol{x}/\sum_j x_j$. Thus, the optimal value is the right-hand side of the last inequality and given by $\mathsf{D} = -\log \sum_j x_j$. This optimal value is achieved by $\boldsymbol{p}^* = \boldsymbol{x}/\sum_j x_j$. This concludes the proof. □

Note that Proposition 3.4 is a generalization of the information inequality: if $\boldsymbol{x}$ is itself a pmf, then according to Proposition 3.4, $\mathsf{D} = 0$ and $\boldsymbol{p}^* = \boldsymbol{x}$, and this is exactly what the information inequality states.

3.2.5 Using a 'wrong' pmf

We are now in the position to express the relative entropy achieved by some pmf $\boldsymbol{p}$ in terms of the optimal value D and the optimal pmf $\boldsymbol{p}^*$.

Proposition 3.5. *Denote by $\boldsymbol{x}$ a non-negative vector and by D and $\boldsymbol{p}^*$ the optimal value and optimal pmf of Problem (3.48), respectively. Denote by $\boldsymbol{p}$ some pmf. Then*

$$\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) = \mathsf{D} + \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*). \tag{3.57}$$

Proof. We have

$$\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) - \mathsf{D} = \mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) + \log \sum_j x_j \tag{3.58}$$

$$= \sum_i p_i \log \frac{p_i}{x_i} + \log \sum_j x_j \tag{3.59}$$

$$= \sum_i p_i \log \frac{p_i \sum_j x_j}{x_i} \tag{3.60}$$

$$= \sum_i p_i \log \frac{p_i}{\frac{x_i}{\sum_j x_j}} \tag{3.61}$$

$$= \sum_i p_i \log \frac{p_i}{p_i^*} \tag{3.62}$$

$$= \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) \tag{3.63}$$

where we used Proposition 3.4 in the first and the last line. □

3.2.6 Asymptotic achievability

We will now show that the optimal value D can be achieved by dyadic pmfs if we jointly consider consecutive symbols. Denote by $\boldsymbol{x}$ a non-negative vector with n entries and define

$$\boldsymbol{x}^k := \underbrace{\boldsymbol{x} \otimes \boldsymbol{x} \otimes \cdots \otimes \boldsymbol{x}}_{k \text{ times}} \tag{3.64}$$

where $\otimes$ denotes the *Kronecker product*, i.e.,

$$\boldsymbol{x} \otimes \boldsymbol{x} = (x_1x_1, \ldots, x_1x_n, x_2x_1, \ldots, x_2x_n, \ldots, x_nx_1, \ldots, x_nx_n)^T. \tag{3.65}$$

For instance, if $\boldsymbol{x}$ is a pmf, then $\boldsymbol{x}^k$ is the joint pmf of k symbols that are iid according to $\boldsymbol{x}$. The topic of this subsection is to show that for the dyadic pmf $\boldsymbol{d}_k = \textsc{Ghc}(\boldsymbol{x}^k)$, the per symbol relative entropy

$$\frac{\mathbb{D}(\boldsymbol{d}_k\|\boldsymbol{x}^k)}{k} \tag{3.66}$$

converges to the optimal value $\mathsf{D} = \mathbb{D}(\boldsymbol{p}^*\|\boldsymbol{x})$ for k to infinity. We call this property *asymptotic achievability*. To show asymptotic achievability of Ghc, we need an auxiliary result that we will detail next.

Greedy channel matching

Denote by $\boldsymbol{t}$ a target pmf. By Proposition 3.4, the optimal value of $\mathbb{D}(\boldsymbol{p}\|\boldsymbol{t})$ is $\mathsf{D} = 0$. We will show that $\mathsf{D} = 0$ is asymptotically achieved by a simple sub-optimal algorithm. Asymptotic achievability of Ghc then follows by optimality. Consider the following algorithm.

Algorithm 2.

$t_1 \geq t_2 \geq \cdots \geq t_n$
$\boldsymbol{d} = 0$, $i = 1$
repeat
  $d_i = 2^{-\lfloor -\log_2 t_i \rfloor}$
  $i \leftarrow i + 1$
until $\sum_i d_i = 1$

In the algorithm, $\lfloor \cdot \rfloor$ denotes the *floor function*, which is defined as

$$\lfloor x \rfloor := \max\{z \in \mathbf{Z} \,|\, z \leq x\}. \tag{3.67}$$

We call this algorithm *greedy channel matching* (GCM) and write $\boldsymbol{d} = \text{GCM}(\boldsymbol{t})$. What the algorithm basically does is to assign for some $m \leq n$ the values $d_i = 2^{-\lfloor -\log_2 t_i \rfloor}$ to the first m entries of $\boldsymbol{d}$ and the value 0 to the remaining $n - m$ entries of $\boldsymbol{d}$. Note that since $\lfloor -\log_2 t_i \rfloor \in \mathbf{N}$ and since the terminating condition of the algorithm is $\sum_i d_i = 1$, the vector $\boldsymbol{d}$ constructed in this way is a dyadic pmf.

First, we show that Algorithm 2 is well-defined, i.e., that the terminating condition is actually fulfilled for some $m \leq n$.

Proposition 3.6. *Assume $t_1 \geq t_2 \geq \cdots \geq t_n$. Define $d_i = 2^{-\lfloor -\log_2 t_i \rfloor}$ for $i = 1, \ldots, n$. Then there is an $m' \leq n$ such that $\sum_{i=1}^{m'} d_i = 1$. With other words, Algorithm 2 is well-defined.*

Proof. Define $\ell_i = \lfloor -\log_2 t_i \rfloor$ and $L_m = \sum_{i=1}^{m} 2^{-\ell_i}$. We have to show that $L_m = 1$ for some $m \leq n$. First, it holds that

$$\sum_{i=1}^{n} 2^{-\ell_i} \geq \sum_{i=1}^{n} 2^{-\log_2 t_i} \tag{3.68}$$

$$= \sum_{i=1}^{n} t_i = 1 \tag{3.69}$$

and therefore,

$$\exists m \leq n : L_m \geq 1. \tag{3.70}$$

Next, we show that L_m is a multiple of $2^{-\ell_{m+1}}$. Since $\ell_{m+1} \geq \ell_m$ and since both ℓ_{m+1} and ℓ_m are integers, we have $\ell_{m+1} = \ell_m + u$ for some integer $u \geq 0$. Thus, L_m is a multiple of $2^{-\ell_{m+1}}$ if it is a multiple of $2^{-\ell_m} = 2^{-\ell_{m+1}} 2^u$. We show the latter by induction in m.
Induction basis: For $m = 1$, $L_1 = 2^{-\ell_1}$.
Induction step: Assume $L_m = s \cdot 2^{-\ell_m}$ for some positive integer s. For some integer $u \geq 0$, it holds that $\ell_{m+1} = \ell_m + u$. Now

$$L_{m+1} = L_m + 2^{-\ell_{m+1}} \tag{3.71}$$

$$= s \cdot 2^{-\ell_m} + 2^{-\ell_{m+1}} \tag{3.72}$$

$$= s \cdot 2^{-(\ell_{m+1} - u)} + 2^{-\ell_{m+1}} \tag{3.73}$$

$$= (s 2^u + 1) 2^{-\ell_{m+1}}. \tag{3.74}$$

Thus,

$$\exists s \in \mathbf{N} : L_m = s2^{-\ell_{m+1}}. \tag{3.75}$$

Assume now $L_m < 1$, i.e., there exists an integer $s < 2^{\ell_{m+1}}$ and equivalently, $s \leq 2^{\ell_{m+1}} - 1$ such that $L_m = s2^{-\ell_{m+1}}$. We now have

$$\begin{aligned} 1 &= 2^{\ell_{m+1}} 2^{-\ell_{m+1}} && (3.76)\\ &= (2^{\ell_{m+1}} - 1 + 1)2^{-\ell_{m+1}} && (3.77)\\ &\geq (s+1)2^{-\ell_{m+1}} && (3.78)\\ &= L_m + 2^{-\ell_{m+1}} && (3.79)\\ &= L_{m+1}. && (3.80) \end{aligned}$$

Thus, we have shown the following implication:

$$L_m < 1 \Rightarrow L_{m+1} \leq 1. \tag{3.81}$$

Since $\ell_1 \geq 0$, $L_1 \leq 1$. Thus, the statements (3.70) and (3.81) can only both be true if there exists an $m' \leq n$ with $L_{m'} = 1$. This concludes the proof. □

Having shown that Gcm is well-defined, we next show that for $\boldsymbol{d} = \textsc{Gcm}(\boldsymbol{t})$, the relative entropy $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{t})$ is bounded.

Proposition 3.7. *For any target pmf $\boldsymbol{t}$ and $\boldsymbol{d} = \textsc{Gcm}(\boldsymbol{t})$, we have*

$$\mathbb{D}(\boldsymbol{d}\|\boldsymbol{t}) \leq \log 2. \tag{3.82}$$

Proof. Assume $\boldsymbol{t}$ has n entries and is sorted, i.e., $t_1 \geq t_2 \geq \cdots \geq t_n$. Denote by m the number of non-zero entries that Gcm assigns to $\boldsymbol{d}$. Then

$$\begin{aligned} \mathbb{D}(\boldsymbol{d}\|\boldsymbol{t}) &= \sum_{i=1}^{n} d_i \log \frac{d_i}{t_i} && (3.83)\\ &= \sum_{i=1}^{m} d_i \log \frac{2^{-\lfloor -\log_2 t_i \rfloor}}{t_i} && (3.84)\\ &\leq \sum_{i=1}^{n} d_i \log \frac{2^{-\lfloor -\log_2 t_i \rfloor}}{t_i} && (3.85)\\ &\leq \sum_{i} d_i \log \frac{2^{-(-\log_2 t_i - 1)}}{t_i} && (3.86)\\ &= \sum_{i} d_i \log \frac{2t_i}{t_i} = \log 2. && (3.87) \end{aligned}$$

Notice how the range of the sum changes from m to n in (3.85). The inequality in (3.86) follows since

$$\lfloor x \rfloor \geq x - 1. \tag{3.88}$$

This concludes the proof. □

Asymptotic achievability of Ghc

With the results for the sub-optimal algorithm Gcm, we are now in the position to show asymptotic achievability for Ghc.

Proposition 3.8. *Denote by $\boldsymbol{x}$ a non-negative vector. For $\boldsymbol{d}_k = \textsc{Ghc}(\boldsymbol{x}^k)$,*

$$\frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{x}^k)}{k} \overset{k\to\infty}{\longrightarrow} \mathsf{D} = \mathbb{D}(\boldsymbol{p}^* \| \boldsymbol{x}). \tag{3.89}$$

In particular, for a target pmf $\boldsymbol{t}$ and $\boldsymbol{d}_k = \textsc{Ghc}(\boldsymbol{t}^k)$,

$$\frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{t}^k)}{k} \overset{k\to\infty}{\longrightarrow} 0. \tag{3.90}$$

Proof. Define $\tilde{\boldsymbol{d}}_k = \textsc{Gcm}(\boldsymbol{p}^{*k})$. We now have

$$\begin{aligned} \frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{x}^k)}{k} &\leq \frac{\mathbb{D}(\tilde{\boldsymbol{d}}_k \| \boldsymbol{x}^k)}{k} && (3.91)\\ &= \mathsf{D} + \frac{\mathbb{D}(\tilde{\boldsymbol{d}}_k \| \boldsymbol{p}^{*k})}{k} && (3.92)\\ &\leq \mathsf{D} + \frac{\log 2}{k} && (3.93) \end{aligned}$$

where the first inequality follows via Proposition 3.3 from the optimality of Ghc. The second line follows from Proposition 3.5 and the inequality in the last line follows from Proposition 3.7. The term $\log 2/k$ goes to zero for $k \to \infty$ and the statement of the proposition follows. □

3.3 Noiseless channel

We now apply the results from the previous section to a *noiseless channel*. In this section, we consider a very simple model and we will extend the results to other models in later chapters. A noiseless channel is given by n input symbols. These symbols are transmitted noiselessly, i.e., at the output of the channel, exactly the symbol at the input is recovered. We assume that each symbol is of duration 1. Then, the maximum rate at which we can transmit information over the channel is the maximum entropy rate H, which is given by the optimal value of the optimization problem

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{maximize}} \quad & \mathbb{H}(\boldsymbol{p}) \\ \text{subject to} \quad & \boldsymbol{p} \text{ is a pmf.} \end{aligned} \tag{3.94}$$

If the input pmf has to be generated by a prefix-free matcher, the corresponding matching problem is

$$\underset{\text{dyadic } \boldsymbol{d}}{\text{maximize}} \quad \mathbb{H}(\boldsymbol{d}). \tag{3.95}$$

By observing that

$$\mathbb{H}(\boldsymbol{p}) = -\,\mathbb{D}(\boldsymbol{p}\|\mathbf{1}) \tag{3.96}$$

it follows that all results from the previous section apply to the problem of maximizing the entropy rate of a noiseless channel. We detail this in the following. For the problem of minimizing $\mathbb{D}(\boldsymbol{p}\|\mathbf{1})$, we denote in the following by D the optimal value.

3.3.1 Matching

Because of (3.96), the matching problem (3.95) is according to Proposition 3.3 optimally solved by $\boldsymbol{d} = \textsc{Ghc}(\mathbf{1})$.

3.3.2 Optimal pmf

Because of (3.96), the maximum entropy rate is by Proposition 3.4 given by

$$\begin{aligned} \mathsf{H} &= -\mathsf{D} && (3.97)\\ &= \log \sum_i 1 && (3.98)\\ &= \log n. && (3.99) \end{aligned}$$

Again by Proposition 3.4, the entries of the optimal pmf $\boldsymbol{p}^*$ are given by

$$\begin{aligned} p_i^* &= e^{\mathsf{D}} 1 && (3.100)\\ &= e^{-\mathsf{H}} && (3.101)\\ &= e^{-\log n} && (3.102)\\ &= \frac{1}{n}, \qquad i = 1, \ldots, n && (3.103) \end{aligned}$$

i.e., the uniform pmf is optimal.

3.3.3 Using a 'wrong' pmf

Denote by $\boldsymbol{p}$ an arbitrary pmf. The entropy rate achieved by an arbitrary pmf $\boldsymbol{p}$ can be expressed in terms of the optimal value H as

$$\begin{aligned} \mathbb{H}(\boldsymbol{p}) &= \mathbb{H}(\boldsymbol{p}) + \mathsf{H} - \mathsf{H} && (3.104)\\ &= \mathsf{H} - \sum_i p_i \log p_i - \sum_i p_i \mathsf{H} && (3.105)\\ &= \mathsf{H} - \Big(\sum_i p_i \log p_i - \sum_i p_i \log e^{-\mathsf{H}}\Big) && (3.106)\\ &= \mathsf{H} - \Big(\sum_i p_i \log p_i - \sum_i p_i \log p_i^*\Big) && (3.107)\\ &= \mathsf{H} - \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) && (3.108) \end{aligned}$$

i.e., the penalty of using $\boldsymbol{p}$ instead of the optimal pmf is $\mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*)$. This shows that the matching problem (3.95) can alternatively be solved by minimizing $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{p}^*)$ over all dyadic pmfs $\boldsymbol{d}$.

3.3.4 Asymptotic Achievability

Because of (3.96) and by Proposition 3.8, we have for $\boldsymbol{d}_k = \textsc{Ghc}(\mathbf{1}^k)$

$$\frac{\mathbb{H}(\boldsymbol{d}_k)}{k} = -\frac{\mathbb{D}(\boldsymbol{d}_k\|\mathbf{1}^k)}{k} \tag{3.109}$$

$$\overset{k\to\infty}{\longrightarrow} -\mathbb{D}(\boldsymbol{p}^*\|\mathbf{1}) \tag{3.110}$$

$$= -\mathsf{D} \tag{3.111}$$

$$= \mathsf{H} \tag{3.112}$$

i.e., Ghc asymptotically achieves the maximum entropy rate.

3.4 Discrete memoryless channel

We will now show how Ghc can be used to find dyadic pmfs that achieve the capacity of dmcs. Recall that a dmc is specified by a matrix $\boldsymbol{H}$ of transition probabilities from n input symbols to m output symbols. An input pmf $\boldsymbol{p}$ relates to its corresponding output pmf $\boldsymbol{r}$ as

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{p}. \tag{3.113}$$

By (2.91), the mutual information between input and output can be written as

$$\mathbb{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j}. \tag{3.114}$$

The capacity of a dmc is the maximum mutual information between input and output, where the maximum is taken over all input pmfs. To find the best dyadic input pmf, we need to solve the matching problem

$$\underset{\text{dyadic } \boldsymbol{d}}{\text{maximize}} \quad \mathbb{I}(\boldsymbol{d}). \tag{3.115}$$

In contrast to the matching problems for relative entropy (3.7) and entropy rate (3.95), we do not know an efficient method to directly solve the matching problem for mutual information. In order to overcome this difficulty, we proceed as follows. First, we drop the restriction to dyadic pmfs and characterize the capacity-achieving pmf $\boldsymbol{p}^*$. Then, we derive the penalty that results from using a pmf $\boldsymbol{p}$ different from $\boldsymbol{p}^*$. Finally, we minimize an upper-bound on this penalty over all dyadic pmfs and show asymptotic achievability.

3.4.1 Capacity

Capacity of a dmc is the optimal value of the following optimization problem.

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{minimize}} \quad & -\mathbb{I}(\boldsymbol{p}) \\ \text{subject to} \quad & \mathbf{1}^T\boldsymbol{p} - 1 = 0. \end{aligned} \tag{3.116}$$

The domain of $\mathbb{I}$ is $\mathbf{R}^n_{\geq 0}$. This is a convex optimization problem and the solution can efficiently be found by numerical methods as provided for example by the software package CVX [41]. The topic of this subsection is to analytically characterize the entries of $\boldsymbol{p}^*$.

Proposition 3.9. *The following conditions are necessary and sufficient for a pmf $\boldsymbol{p}$ to be capacity-achieving.*

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} = \mathsf{C}, \qquad \forall i : p_i > 0 \tag{3.117}$$

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq \mathsf{C}, \qquad \forall i : p_i = 0. \tag{3.118}$$

Proof. By Proposition 2.5 strong duality holds. By Proposition 2.7 the partial derivatives of $-\mathbb{I}$ are well-defined with the possible exception of taking the value $-\infty$ on the boundary. Thus, Proposition 2.4 applies. The Lagrangian of Problem (3.116) is

$$L(\boldsymbol{p}, \nu) = -\mathbb{I}(\boldsymbol{p}) + \nu(\mathbf{1}^T\boldsymbol{p} - 1). \tag{3.119}$$

Note that any pmf is feasible. Thus, by Proposition 2.4, a pmf $\boldsymbol{p}$ is optimal if and only if the KKT conditions are fulfilled, i.e.,

$$\frac{\partial L(\boldsymbol{p}, \nu)}{\partial p_i} = -\frac{\partial \mathbb{I}(\boldsymbol{p})}{\partial p_i} + \nu = 0, \qquad \forall i : p_i > 0 \tag{3.120}$$

$$-\frac{\partial \mathbb{I}(\boldsymbol{p})}{\partial p_i} + \nu \geq 0, \qquad \forall i : p_i = 0. \tag{3.121}$$

By Proposition 2.7, these conditions imply that all partial derivatives of $\mathbb{I}$ in $\boldsymbol{p}$ are well-defined and given by

$$\frac{\partial \mathbb{I}(\boldsymbol{p})}{\partial p_i} = \sum_j h_{ji} \log \frac{h_{ji}}{r_j} - 1, \qquad i = 1, \ldots, n. \tag{3.122}$$

Plugging this into the KKT conditions, we get

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq 1 + \nu, \qquad \text{with equality if } p_i > 0. \tag{3.123}$$

For a capacity-achieving pmf $\boldsymbol{p}^*$, we have

$$\begin{aligned} \mathsf{C} &= \mathbb{I}(\boldsymbol{p}^*) && (3.124)\\ &= \sum_i p_i^* \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} && (3.125)\\ &= \sum_i p_i^* (1+\nu) && (3.126)\\ &= 1+\nu && (3.127) \end{aligned}$$

where we used (3.123) in the third line. Note that we have equality in (3.123) for all i with $p_i^* > 0$. We conclude that $1+\nu = \mathsf{C}$. Plugging this into (3.123), we finally get

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq \mathsf{C}, \qquad \text{with equality if } p_i > 0. \tag{3.128}$$

This concludes the proof. □

3.4.2 Using a 'wrong' pmf

We now use Proposition 3.9 to express the mutual information $\mathbb{I}(\boldsymbol{p})$ achieved by some pmf $\boldsymbol{p}$ in terms of capacity C and capacity-achieving pmf $\boldsymbol{p}^*$. We will need the following property of mutual information.

Proposition 3.10. *For a dmc, denote by $\boldsymbol{p}'$ an input pmf and by $\boldsymbol{p}$ some other input pmf with the only restriction that*

$$p_i = 0, \quad \text{whenever } p_i' = 0. \tag{3.129}$$

Then the mutual information achieved by $\boldsymbol{p}$ is given by

$$\mathbb{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j'} - \mathbb{D}(\boldsymbol{r} \| \boldsymbol{r}') \tag{3.130}$$

where $\boldsymbol{r}$ and $\boldsymbol{r}'$ are the output pmfs resulting from $\boldsymbol{p}$ and $\boldsymbol{p}'$, respectively.

Proof.

$$\mathbb{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j} \tag{3.131}$$

$$= \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji} r'_j}{r_j r'_j} \tag{3.132}$$

$$= \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r'_j} + \sum_i p_i \sum_j h_{ji} \log \frac{r'_j}{r_j} \tag{3.133}$$

$$= \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r'_j} - \sum_j \Big(\sum_i p_i h_{ji} \Big) \log \frac{r_j}{r'_j} \tag{3.134}$$

$$= \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r'_j} - \mathbb{D}(\boldsymbol{r} \| \boldsymbol{r}'). \tag{3.135}$$

The term in (3.132) is well-defined even if $r'_j = 0$. This can be seen as follows. Since $r'_j = \sum_i h_{ji} p'_i$, $r'_j = 0$ implies $p'_i h_{ji} = 0$ for each i. But according to the assumption (3.129), this also implies $p_i h_{ji} = 0$ for each i. Thus, because of $0 \log \frac{0}{0} = 0$, (3.132) is well-defined. □

We are now in the position to quantify the penalty that results when we do not use the capacity-achieving pmf of a dmc.

Proposition 3.11. *Consider a dmc with capacity-achieving pmf $\boldsymbol{p}^*$ and capacity C. Denote by $\boldsymbol{p}$ an arbitrary input pmf with the only restriction that*

$$p_i = 0, \quad \text{whenever } p_i^* = 0. \tag{3.136}$$

The mutual information that is achieved by $\boldsymbol{p}$ is given by

$$\mathbb{I}(\boldsymbol{p}) = \mathbb{I}(\boldsymbol{p}^*) - \mathbb{D}(\boldsymbol{r} \| \boldsymbol{r}^*) \tag{3.137}$$

$$= \mathsf{C} - \mathbb{D}(\boldsymbol{r} \| \boldsymbol{r}^*). \tag{3.138}$$

Proof. We have

$$\mathbb{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} - \mathbb{D}(\boldsymbol{r} \| \boldsymbol{r}^*) \tag{3.139}$$

$$= \sum_i p_i \mathsf{C} - \mathbb{D}(\boldsymbol{r} \| \boldsymbol{r}^*) \tag{3.140}$$

$$= \mathsf{C} - \mathbb{D}(\boldsymbol{r} \| \boldsymbol{r}^*) \tag{3.141}$$

$$= \mathbb{I}(\boldsymbol{p}^*) - \mathbb{D}(\boldsymbol{r} \| \boldsymbol{r}^*). \tag{3.142}$$

where equality in the first line follows from assumption (3.136) and Proposition 3.10 and where equality in the second line follows from assumption (3.136) and Proposition 3.9. □

3.4.3 Matching

Proposition 3.11 suggests to match a dmc by minimizing the relative entropy $\mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)$ of the output pmfs over all dyadic input pmfs. However, we do not know an efficient algorithm to do so. We will therefore use the fact that the relative entropy at the output is upper-bounded by the relative entropy at the input.

Proposition 3.12. *For a dmc the relative entropy between the output pmfs is upper bounded by the relative entropy between the input pmfs, i.e., for two input pmfs $\boldsymbol{p}$ and $\boldsymbol{p}'$ and the corresponding output pmfs $\boldsymbol{r}$ and $\boldsymbol{r}'$,*

$$\mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}') \le \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}'). \tag{3.143}$$

Proof. This follows from the log sum inequality (2.81):

$$\begin{aligned}
\mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}') &= \sum_j r_j \log \frac{r_j}{r'_j} && (3.144)\\
&= \sum_j \Big(\sum_i h_{ji} p_i\Big) \log \frac{\sum_i h_{ji} p_i}{\sum_i h_{ji} p'_i} && (3.145)\\
&\le \sum_j \sum_i h_{ji} p_i \log \frac{h_{ji} p_i}{h_{ji} p'_i} && (3.146)\\
&= \sum_j \sum_i h_{ji} p_i \log \frac{p_i}{p'_i} && (3.147)\\
&= \sum_i p_i \log \frac{p_i}{p'_i} \Big(\sum_j h_{ji}\Big) && (3.148)\\
&= \sum_i p_i \log \frac{p_i}{p'_i} && (3.149)\\
&= \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}'). && (3.150)
\end{aligned}$$

□

Thus, according to Proposition 3.11, if $p_i = 0$ whenever $p^*_i = 0$, we have the following lower bound on $\mathbb{I}(\boldsymbol{p})$:

$$\mathbb{I}(\boldsymbol{p}) \ge \mathsf{C} - \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*). \tag{3.151}$$

According to Proposition 3.2, $\boldsymbol{d} = \textsc{Ghc}(\boldsymbol{p}^*)$ guarantees $d_i = 0$ whenever $p^*_i = 0$. Thus,

$$\mathbb{I}(\boldsymbol{d}) \ge \mathsf{C} - \mathbb{D}(\boldsymbol{d}\|\boldsymbol{p}^*) \tag{3.152}$$

and by Proposition 3.3, $\boldsymbol{d} = \textsc{Ghc}(\boldsymbol{p}^*)$ minimizes $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{p}^*)$ over all dyadic pmfs, i.e., it minimizes our upper-bound on the penalty term.

3.4.4 Asymptotic achievability

We now show that dyadic pmfs asymptotically achieve capacity. To this end, we consider k consecutive input symbols. Since the channel is memoryless, the capacity-achieving joint pmf of k symbols is $\boldsymbol{p}^{*k}$ and the mutual information achieved by $\boldsymbol{p}^{*k}$ is $\mathbb{I}(\boldsymbol{p}^{*k}) = k\mathsf{C}$. We now have the following result.

Proposition 3.13. *Denote by $\boldsymbol{p}^*$ the capacity-achieving pmf of a dmc with capacity C. Then for $\boldsymbol{d}_k = \textsc{Ghc}(\boldsymbol{p}^{*k})$, it holds that*

$$\frac{\mathbb{I}(\boldsymbol{d}_k)}{k} \overset{k\to\infty}{\longrightarrow} \mathsf{C} \tag{3.153}$$

i.e., Ghc *asymptotically achieves the capacity of the dmc.*

Proof. Denote by $\boldsymbol{r}_k$ and $\boldsymbol{r}^{*k}$ the output pmfs resulting from the input pmfs $\boldsymbol{d}_k$ and $\boldsymbol{p}^{*k}$, respectively. We have

$$\begin{aligned}
\mathbb{I}(\boldsymbol{d}_k) &= \mathbb{I}(\boldsymbol{d}^{*k}) - \mathbb{D}(\boldsymbol{r}_k \| \boldsymbol{r}^{*k}) && (3.154)\\
&= k\mathsf{C} - \mathbb{D}(\boldsymbol{r}_k \| \boldsymbol{r}^{*k}) && (3.155)\\
&\geq k\mathsf{C} - \mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{p}^{*k}) && (3.156)
\end{aligned}$$

where the first line follows from Proposition 3.11 and where the third line follows from Proposition 3.12, which applies because of Proposition 3.2. Dividing by k, we get

$$\begin{aligned}
\frac{\mathbb{I}(\boldsymbol{d}_k)}{k} &\geq \mathsf{C} - \frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{p}^{*k})}{k} && (3.157)\\
&\overset{k\to\infty}{\longrightarrow} \mathsf{C} && (3.158)
\end{aligned}$$

where the last line follows from Proposition 3.8. This concludes the proof. □

3.5 References

Parts of this chapter were published in [17].

A fair bit stream at the binary interface in a digital communication system can be obtained in several ways. In [43], Han proves that optimal source coding achieves this. In practice, source encoders are seldom perfect. Here, pseudorandomness can help. Vasić *et al* propose in [74] to use a scrambler to generate fair bit streams. A similar approach is proposed by Ungerböck in [73]. In [4], we show how Huffman source codes can be optimized to make the source encoder output resemble a fair bit stream.

The problem of minimizing $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{t})$ for a pmf $\boldsymbol{t}$ over dyadic pmfs $\boldsymbol{d}$ is stated by Abrahams in [3], but no solution is provided. In [71], Stubley and Blake propose a sub-optimal algorithm. An algorithm in terms of codeword lengths that is equivalent to Ghc was independently found by Dubé and Beaudoin [29]. Lemma 1 and Lemma 2 are stated and proved by Gallager in [40, Lemma 2.5.1] and [40, Lemma 2.5.3], respectively. Asymptotic optimality of Huffman source coding is shown in [24, Section 5.4] by proving this

property for Shannon coding. The role of GCM for prefix-free matching corresponds to the role of Shannon coding for prefix-free source coding.

The maximum entropy (3.99) and the entropy maximizing pmf (3.103) of a random variable over a finite set are well-known, see for example [24, Theorem 2.6.4].

Gallager states Proposition 3.9 in [39, Theorem 4.5.1]. An efficient algorithm to numerically solve Problem (3.116) was proposed by Blahut [6] and Arimoto [5]. Under condition (3.129), the identity (3.130) in Proposition 3.10 is equivalent to [25, Equation (8.7)] stated by Csiszár and Körner. The latter result is also stated by Topsøe in [72, Theorem 9.1] and referred to as *the compensation identity*. Proposition 3.11 is stated for *additive white Gaussian noise* (AWGN) channels in [76, Equation (5)] by Wu and Verdú. The Gaussian case follows directly from [24, Theorem 8.6.5]. Proposition 3.12 and proof are stated as part of the *data processing lemma* by Csiszár and Körner in [25, Lemma 3.11]. Under a certain condition, the bound can be strengthened, see [25, Problem 3.19]. Proposition 3.12 can also be shown following [24, Section 4.4].

4 Matching channels with unequal symbol durations

In the previous chapter, we implicitly assumed that the symbols generated by the pmf of interest are all of equal duration. We then minimized relative entropy and maximized entropy and mutual information. In this chapter, we assume that each symbol has an associated duration and that different symbols have possibly different durations. We now want to optimize performance per average duration. For example for a dmc, the capacity is now

$$\mathsf{C} = \max_{\text{pmf } \boldsymbol{p}} \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T \boldsymbol{p}} \tag{4.1}$$

where the ith entry of $\boldsymbol{w}$ is the duration of the ith input symbol and where $\boldsymbol{w}^T \boldsymbol{p}$ is the average duration. Our goal is to find methods that yield the best dyadic pmfs with respect to the new objective function. We start with the problem of minimizing the relative entropy per average duration for a given non-negative target vector over all dyadic pmfs. We then show how the obtained results can be applied to noiseless channels and dmcs with unequal symbol durations. The main new ingredients of this chapter are iterative algorithms.

4.1 Normalized geometric Huffman coding

Consider an arbitrary non-negative target vector $\boldsymbol{x}$ with n entries and a vector of positive durations $\boldsymbol{w}$. The objective is to approximate $\boldsymbol{x}$ by a dyadic pmf $\boldsymbol{d}$ such that the relative entropy $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x})$ normalized by $\boldsymbol{w}^T \boldsymbol{d}$ is minimized, i.e., the objective is to solve the matching problem

$$\underset{\boldsymbol{d} \text{ dyadic}}{\text{minimize}} \quad \frac{\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x})}{\boldsymbol{w}^T \boldsymbol{d}}. \tag{4.2}$$

We can solve Problem (4.2) iteratively. The intuition is as follows. Denote by D the optimal value of Problem (4.2) and by $\boldsymbol{d}^*$ the optimal dyadic pmf. Then, for an arbitrary dyadic pmf $\boldsymbol{d}$,

$$\frac{\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x})}{\boldsymbol{w}^T \boldsymbol{d}} \geq \mathsf{D}, \qquad \text{with equality if } \boldsymbol{d} = \boldsymbol{d}^* \tag{4.3}$$

$$\Rightarrow \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}) - \mathsf{D}\boldsymbol{w}^T \boldsymbol{d} \geq 0, \qquad \text{with equality if } \boldsymbol{d} = \boldsymbol{d}^*. \tag{4.4}$$

Thus, $\boldsymbol{d}^*$ can be found by minimizing the left-hand side of the last line. We can write this as

$$\begin{aligned}
\boldsymbol{d}^* &= \operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}) - \mathsf{D}\boldsymbol{w}^T\boldsymbol{d} && (4.5)\\
&= \operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \sum_i p_i \Big(\log \frac{p_i}{x_i} - \mathsf{D}w_i\Big) && (4.6)\\
&= \operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \sum_i p_i \log \frac{p_i}{x_i e^{\mathsf{D}w_i}} && (4.7)\\
&= \operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x} \circ e^{\mathsf{D}\boldsymbol{w}}) && (4.8)
\end{aligned}$$

where we used the symbol $\circ$ to denote *elementwise multiplication* (often called *Hadamard product*), i.e., for two vectors $\boldsymbol{x}, \boldsymbol{y}$ with n entries,

$$\boldsymbol{x} \circ \boldsymbol{y} := (x_1 y_1, \ldots, x_n y_n)^T \tag{4.9}$$

and where we used the notation $e^{\mathsf{D}\boldsymbol{w}}$ to denote *exponentiation by a vector*, i.e.,

$$e^{\mathsf{D}\boldsymbol{w}} := (e^{\mathsf{D}w_1}, \ldots, e^{\mathsf{D}w_n})^T. \tag{4.10}$$

According to Proposition 3.3, the minimization problem in (4.8) is solved by GHC, i.e.,

$$\operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x} \circ e^{\mathsf{D}\boldsymbol{w}}) = \text{GHC}(\boldsymbol{x} \circ e^{\mathsf{D}\boldsymbol{w}}). \tag{4.11}$$

The remaining problem is that we in most cases do not know the optimal value D in advance. We therefore substitute it by a guess Δ and we update Δ iteratively. The resulting algorithm is as follows.

Algorithm 3.(NGHC)

$\boldsymbol{d}' = \text{GHC}(\boldsymbol{x})$
repeat
1. $\Delta = \frac{\mathbb{D}(\boldsymbol{d}'\|\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{d}'}$
2. $\boldsymbol{d}' = \text{GHC}(\boldsymbol{x} \circ e^{\Delta\boldsymbol{w}})$

until $\mathbb{D}(\boldsymbol{d}'\|\boldsymbol{x}) - \Delta\boldsymbol{w}^T\boldsymbol{d}' = 0$

We call this algorithm *normalized geometric Huffman coding* (NGHC) and write $\boldsymbol{d} = \text{NGHC}(\boldsymbol{x}, \boldsymbol{w})$.

4.1.1 Optimality of normalized geometric Huffman coding

Proposition 4.1. *The algorithm* NGHC *finds the optimal solution of Problem* (4.2) *in finitely many steps.*

Proof. Δ *is strictly monotonically decreasing.* Denote by Δ_i the value that is assigned to Δ in step 1. of the ith iteration and denote by $\boldsymbol{d}'_i$ the value that is assigned to $\boldsymbol{d}'$ in step 2. of the ith iteration. Assume that the algorithm does not terminate in the ith iteration. We have

$$\Delta_i = \frac{\mathbb{D}(\boldsymbol{d}'_{i-1}\|\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{d}'_{i-1}} \tag{4.12}$$

$$\Rightarrow \mathbb{D}(\boldsymbol{d}'_{i-1}\|\boldsymbol{x}) - \Delta_i \boldsymbol{w}^T\boldsymbol{d}'_{i-1} = 0. \tag{4.13}$$

Thus, since by (4.11)

$$\boldsymbol{d}'_i = \operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}) - \Delta_i \boldsymbol{w}^T\boldsymbol{d} \tag{4.14}$$

and since according to our assumption, the algorithm does not terminate in the ith iteration, we have

$$\mathbb{D}(\boldsymbol{d}'_i\|\boldsymbol{x}) - \Delta \boldsymbol{w}^T\boldsymbol{d}'_i < 0 \tag{4.15}$$

$$\Rightarrow \frac{\mathbb{D}(\boldsymbol{d}'_i\|\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{d}'_i} < \Delta_i \tag{4.16}$$

$$\Rightarrow \Delta_{i+1} < \Delta_i \tag{4.17}$$

which shows that Δ is strictly monotonically decreasing until termination.

Optimality after termination. Assume the algorithm terminated, and denote by $\boldsymbol{d}'$ the pmf after termination. Because of the assignments in step 1. and step 2., the terminating condition implies for any dyadic pmf $\boldsymbol{d}$

$$\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}) - \Delta \boldsymbol{w}^T\boldsymbol{d} \geq 0, \quad \text{with equality if } \boldsymbol{d} = \boldsymbol{d}'. \tag{4.18}$$

Consequently,

$$\frac{\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{d}} \geq \Delta, \quad \text{with equality if } \boldsymbol{d} = \boldsymbol{d}' \tag{4.19}$$

and we conclude that after termination, $\boldsymbol{d}'$ is the solution of (4.2).

Termination in finitely many steps. It remains to show that the algorithm terminates after finitely many steps. This can be seen as follows. First, as we have shown, the algorithm is strictly monotonically decreasing in Δ. In particular, until termination, $\boldsymbol{d}'_i \neq \boldsymbol{d}_j$ for all $j < i$. Second, there are only finitely many distinct dyadic pmfs with n entries. Thus, the algorithm has to terminate after finitely many steps. □

4.1.2 Optimal pmf

We now want to quantify the penalty that results from the restriction to dyadic pmfs. To this end, we first determine what can be achieved without this restriction, i.e., we consider the optimization problem

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{minimize}} \quad & \frac{\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{p}} \\ \text{subject to} \quad & \boldsymbol{p} \text{ is a pmf.} \end{aligned} \tag{4.20}$$

Denote by D the optimal value of this problem and denote by $\boldsymbol{p}^*$ the corresponding optimal pmf. We next characterize D and $\boldsymbol{p}^*$.

Proposition 4.2. *The optimal pmf of Problem (4.20) is given by*

$$\boldsymbol{p}^* = \boldsymbol{x} \circ e^{\mathsf{D}\boldsymbol{w}} \tag{4.21}$$

and the optimal value is given by the solution of

$$\sum_i x_i e^{sw_i} = 1, \qquad s \in \mathbf{R}. \tag{4.22}$$

Proof. Denote by $\boldsymbol{p}$ some pmf. Then

$$\frac{\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{p}} \geq \mathsf{D}, \qquad \text{with equality if } \boldsymbol{p} = \boldsymbol{p}^* \tag{4.23}$$

which is equivalent to

$$\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) - \mathsf{D}\boldsymbol{w}^T\boldsymbol{p} \geq 0, \qquad \text{with equality if } \boldsymbol{p} = \boldsymbol{p}^*. \tag{4.24}$$

The left-hand side can be written as

$$\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) - \mathsf{D}\boldsymbol{w}^T\boldsymbol{p} = \mathbb{D}(\boldsymbol{p}\|\boldsymbol{x} \circ e^{\mathsf{D}\boldsymbol{w}}). \tag{4.25}$$

Thus, $\boldsymbol{p}^*$ minimizes the objective of (4.20) if it minimizes

$$\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x} \circ e^{\mathsf{D}\boldsymbol{w}}) \tag{4.26}$$

and furthermore,

$$\min_{\text{pmf } \boldsymbol{p}} \mathbb{D}(\boldsymbol{p}\|\boldsymbol{x} \circ e^{\mathsf{D}\boldsymbol{w}}) = 0. \tag{4.27}$$

Thus, by Proposition 3.4,

$$p_i^* = e^0 x_i e^{\mathsf{D}w_i} \tag{4.28}$$

$$= x_i e^{\mathsf{D}w_i}, \qquad i = 1, \ldots, n. \tag{4.29}$$

Since $\boldsymbol{p}^*$ is a pmf, D is given by the solution of

$$\sum_i x_i e^{sw_i} = 1, \qquad s \in \mathbf{R}. \tag{4.30}$$

This concludes the proof. □

Note that the sum

$$\sum_i x_i e^{sw_i} \tag{4.31}$$

is monotonically increasing in s, which follows from

$$\frac{\partial}{\partial s} \sum_i x_i e^{sw_i} = \sum_i x_i w_i e^{sw_i} > 0. \tag{4.32}$$

Thus, D can be found via bisection.

Using a 'wrong' pmf

We are now in the position to express the relative entropy per average duration that is achieved by some pmf $\boldsymbol{p}$ in terms of the optimal value D and the optimal pmf $\boldsymbol{p}^*$.

Proposition 4.3. *Denote by $\boldsymbol{x}$ a non-negative target vector and by D and $\boldsymbol{p}^*$ the optimal value and the optimal pmf of Problem (4.20), respectively. Denote by $\boldsymbol{p}$ some arbitrary pmf with the only restriction that*

$$p_i = 0, \quad \text{whenever } x_i = 0. \tag{4.33}$$

Then

$$\frac{\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{p}} = \mathsf{D} + \frac{\mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*)}{\boldsymbol{w}^T\boldsymbol{p}}. \tag{4.34}$$

Proof. We write

$$\begin{aligned}
\frac{\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})}{\boldsymbol{w}^T\boldsymbol{p}} &= \frac{\sum_i p_i \log \frac{p_i}{x_i}}{\boldsymbol{w}^T\boldsymbol{p}} && (4.35)\\
&= \frac{\sum_i p_i \log \frac{p_i p_i^*}{x_i p_i^*}}{\boldsymbol{w}^T\boldsymbol{p}} && (4.36)\\
&= \frac{\sum_i p_i \log \frac{p_i^*}{x_i} + \sum_i p_i \log \frac{p_i}{p_i^*}}{\boldsymbol{w}^T\boldsymbol{p}} && (4.37)\\
&= \frac{\mathsf{D}\boldsymbol{w}^T\boldsymbol{p} + \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*)}{\boldsymbol{w}^T\boldsymbol{p}} && (4.38)\\
&= \mathsf{D} + \frac{\mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*)}{\boldsymbol{w}^T\boldsymbol{p}} && (4.39)
\end{aligned}$$

which is what we wanted to show. The term in the second line is well-defined even if $p_i^* = 0$ for some i. This can be seen as follows. According to Proposition 4.2, $p_i^* = x_i e^{\mathsf{D}w_i}$, therefore (4.33) implies $p_i = 0$ whenever $p_i^* = 0$. Thus, because of $0 \log \frac{0}{0} = 0$, the second line is well-defined. This concludes the proof. □

4.1.3 Asymptotic achievability

We finally show that the optimal value D can be achieved by dyadic pmfs if we jointly consider consecutive symbols. We will need the following notation. The vector $\boldsymbol{v}_k$ denotes the durations of k consecutive symbols. It can be calculated by

$$\begin{aligned}
\boldsymbol{v}_k &= \boldsymbol{w} \oplus \boldsymbol{w} \oplus \cdots \oplus \boldsymbol{w} && (4.40)\\
&=: \oplus^k \boldsymbol{w} && (4.41)
\end{aligned}$$

where we call $\boldsymbol{w} \oplus \boldsymbol{w}$ the *cost sum*. The cost sum $\boldsymbol{w} \oplus \boldsymbol{w}$ has n^2 entries and is given by

$$\boldsymbol{w} \oplus \boldsymbol{w} = (w_1 + w_1, \ldots, w_1 + w_n, w_2 + w_1, \ldots, w_2 + w_n, \ldots, w_n + w_1, \ldots, w_n + w_n)^T$$

i.e., the $[(i-1)n + j]$th entry is given by $w_i + w_j$.

Proposition 4.4. *Denote by $\boldsymbol{x}$ a given non-negative target vector. Then for $\boldsymbol{d}_k = \textsc{nGhc}(\boldsymbol{x}^k, \boldsymbol{v}_k = \oplus^k \boldsymbol{w})$,*

$$\frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{x}^k)}{\boldsymbol{v}_k^T \boldsymbol{d}_k} \overset{k\to\infty}{\longrightarrow} \mathsf{D} = \frac{\mathbb{D}(\boldsymbol{p}^* \| \boldsymbol{x})}{\boldsymbol{w}^T \boldsymbol{p}^*}. \tag{4.42}$$

In particular, for a target pmf $\boldsymbol{t}$ and $\boldsymbol{d}_k = \textsc{nGhc}(\boldsymbol{t}^k, \boldsymbol{v}_k = \oplus^k \boldsymbol{w})$,

$$\frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{t}^k)}{\boldsymbol{v}_k^T \boldsymbol{d}_k} \overset{k\to\infty}{\longrightarrow} 0. \tag{4.43}$$

Proof. Define

$$w_{\min} = \min\{w_1, \ldots, w_m\}. \tag{4.44}$$

Furthermore, define $\tilde{d}_k = \textsc{Ghc}(\boldsymbol{p}^{*k})$. We have

$$\begin{aligned}
\frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{x}^k)}{\boldsymbol{v}_k^T \boldsymbol{d}_k} &= \mathsf{D} + \frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{p}^{*k})}{\boldsymbol{v}_k^T \boldsymbol{d}_k} && (4.45)\\
&\le \mathsf{D} + \frac{\mathbb{D}(\tilde{\boldsymbol{d}}_k \| \boldsymbol{p}^{*k})}{\boldsymbol{v}_k^T \tilde{\boldsymbol{d}}_k} && (4.46)\\
&\le \mathsf{D} + \frac{\mathbb{D}(\tilde{\boldsymbol{d}}_k \| \boldsymbol{p}^{*k})}{k w_{\min}} && (4.47)\\
&= \mathsf{D} + w_{\min}^{-1} \frac{\mathbb{D}(\tilde{\boldsymbol{d}}_k \| \boldsymbol{p}^{*k})}{k}. && (4.48)
\end{aligned}$$

Equality in the first line follows from Proposition 4.3, which applies because of Proposition 3.2. The inequality in the second line follows since by Proposition 4.1, the pmf $\boldsymbol{d}_k = \textsc{nGhc}(\boldsymbol{x}^k, \boldsymbol{v}_k)$ minimizes the left-hand side of the first line over all dyadic pmfs. Thus, it also minimizes the right-hand side of the first line and $\tilde{\boldsymbol{d}}_k = \textsc{Ghc}(\boldsymbol{p}^{*k})$ in the second line can only do worse. The inequality in the third line follows from $\boldsymbol{v}_k^T \boldsymbol{d}_k \ge k w_{\min}$. For $k \to \infty$, the normalized relative entropy

$$\frac{\mathbb{D}(\tilde{\boldsymbol{d}}_k \| \boldsymbol{p}^{*k})}{k} \tag{4.49}$$

in (4.48) goes by Proposition 3.8 to zero. Since nGhc can only do better, the proposition follows. □

4.2 Noiseless channel

In this section, we apply the results from the previous section to a *noiseless channel with unequal symbol durations*. Such a channel is specified by a vector of positive durations $\boldsymbol{w}$. Capacity is given by the maximum entropy per average duration, i.e., by

$$\mathsf{H} = \max_{\text{pmf } \boldsymbol{p}} \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T \boldsymbol{p}}. \tag{4.50}$$

If we generate the input pmf by a prefix-free matcher, the best entropy rate that can be achieved is given by the optimal value of the corresponding matching problem

$$\underset{\text{dyadic } \boldsymbol{d}}{\text{maximize}} \quad \frac{\mathbb{H}(\boldsymbol{d})}{\boldsymbol{w}^T \boldsymbol{d}}. \tag{4.51}$$

Similar to Section 3.3, we can write the entropy rate achieved by an input pmf $\boldsymbol{p}$ as

$$\frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T \boldsymbol{p}} = -\frac{\sum_i p_i \log \frac{p_i}{1}}{\boldsymbol{w}^T \boldsymbol{p}} \tag{4.52}$$

$$= -\frac{\mathbb{D}(\boldsymbol{p} \| \mathbf{1})}{\boldsymbol{w}^T \boldsymbol{p}}. \tag{4.53}$$

Thus, all results from the previous section directly apply to noiseless channels with unequal symbols durations. We detail them for noiseless channels in the following. For the problem of minimizing $\mathbb{D}(\boldsymbol{p} \| \mathbf{1})/(\boldsymbol{w}^T \boldsymbol{p})$, we denote the optimal value by D.

4.2.1 Matching

Because of (4.53), the matching problem (4.51) is according to Proposition 4.1 optimally solved by $\boldsymbol{d} = \textsc{nGhc}(\mathbf{1}, \boldsymbol{w})$.

4.2.2 Optimal pmf

By Proposition 4.2 and because of (4.53), $-\mathsf{H} = \mathsf{D}$ is given by the solution of

$$\sum_i 1 \cdot e^{s w_i} = 1. \tag{4.54}$$

Consequently, the maximum entropy can directly be obtained by solving

$$\sum_i e^{-s w_i} = 1. \tag{4.55}$$

The solution can be found by Algorithm 6, which we derive for a more general setting in Subsection 6.3.1. Furthermore, the capacity-achieving pmf is given by

$$\boldsymbol{p}^* = \mathbf{1} \circ e^{\mathsf{D}\boldsymbol{w}} \tag{4.56}$$

$$= e^{-\mathsf{H}\boldsymbol{w}}. \tag{4.57}$$

4.2.3 Using a 'wrong' pmf

Denote by $\boldsymbol{p}$ an arbitrary input pmf. The achieved entropy rate can be written as

$$\frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T \boldsymbol{p}} = -\frac{\mathbb{D}(\boldsymbol{p} \| \mathbf{1})}{\boldsymbol{w}^T \boldsymbol{p}} \tag{4.58}$$

$$= -\mathsf{D} - \frac{\mathbb{D}(\boldsymbol{p} \| \boldsymbol{p}^*)}{\boldsymbol{w}^T \boldsymbol{p}} \tag{4.59}$$

$$= \mathsf{H} - \frac{\mathbb{D}(\boldsymbol{p} \| \boldsymbol{p}^*)}{\boldsymbol{w}^T \boldsymbol{p}} \tag{4.60}$$

where the first line follows from (4.53), the second line follows from Proposition 4.3 and the last line again follows from (4.53). The second term in the last line gives the penalty that results from using $\boldsymbol{p}$ instead of $\boldsymbol{p}^*$.

4.2.4 Asymptotic achievability

For $\boldsymbol{d}_k = \textsc{nGhc}(\mathbf{1}^k, \boldsymbol{v}_k = \oplus^k \boldsymbol{w})$, we have

$$\frac{\mathbb{H}(\boldsymbol{d}_k)}{\boldsymbol{v}_k^T \boldsymbol{d}_k} = -\frac{\mathbb{D}(\boldsymbol{d}_k \| \mathbf{1}^k)}{\boldsymbol{v}_k^T \boldsymbol{d}_k} \tag{4.61}$$

$$\stackrel{k\to\infty}{\longrightarrow} -\mathsf{D} \tag{4.62}$$

$$= \mathsf{H}. \tag{4.63}$$

where we used (4.53) in the first and last line and where we used Proposition 4.1 in the second line. We conclude that nGhc asymptotically achieves the capacity of a noiseless channel with durations.

4.3 Discrete memoryless channels

In this section, we consider dmcs where the input symbols are of possibly unequal durations and we will show how nGhc can be used to generate asymptotically capacity-achieving dyadic pmfs. Recall that a dmc is specified by a matrix $\boldsymbol{H}$ of transition probabilities from n input symbols to m output symbols. An input pmf $\boldsymbol{p}$ relates to its corresponding output pmf $\boldsymbol{r}$ as

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{p}. \tag{4.64}$$

By (2.91), the mutual information between input and output can be written as

$$\mathbb{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j}. \tag{4.65}$$

The input symbols have possibly unequal positive durations $\boldsymbol{w}$. The capacity of such a dmc is given by the maximum mutual information per average weight $\mathbb{I}(\boldsymbol{p})/(\boldsymbol{w}^T\boldsymbol{p})$. To find the best dyadic input pmf, we need to solve the matching problem

$$\underset{\text{dyadic } \boldsymbol{d}}{\text{maximize}} \quad \frac{\mathbb{I}(\boldsymbol{d})}{\boldsymbol{w}^T \boldsymbol{d}}. \tag{4.66}$$

We do not know an efficient algorithm that solves this problem directly. We therefore proceed as follows. First, we drop the restriction to dyadic pmfs. After deriving an algorithm for calculating capacity C and capacity-achieving pmf $\boldsymbol{p}^*$, we analytically characterize C and $\boldsymbol{p}^*$. Based on this characterization, we derive the penalty that results from using a pmf $\boldsymbol{p}$ different from $\boldsymbol{p}^*$. Finally, we minimize an upper-bound on this penalty over all dyadic pmfs and show asymptotic achievability.

4.3.1 Capacity

We consider the problem

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{maximize}} \quad & \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}} \\ \text{subject to} \quad & \boldsymbol{p} \text{ is a pmf.} \end{aligned} \tag{4.67}$$

The following variation of nGHC solves this problem iteratively. See Section 4.1 for the intuition behind the algorithm.

Algorithm 4.

$\boldsymbol{p}' = (\frac{1}{n}, \ldots, \frac{1}{n})^T$
$\epsilon > 0$
repeat
1. $C = \frac{\mathbb{I}(\boldsymbol{p}')}{\boldsymbol{w}^T\boldsymbol{p}'}$
2. $\boldsymbol{p}' = \underset{\text{pmf } \boldsymbol{p}}{\operatorname{argmax}} \mathbb{I}(\boldsymbol{p}') - C\boldsymbol{w}^T\boldsymbol{p}'$

until $\mathbb{I}(\boldsymbol{p}') - C\boldsymbol{w}^T\boldsymbol{p}' < \epsilon$

Operation 1. is a simple assignment and operation 2. consists in solving a convex optimization problem, which can efficiently be done by convex optimization software as for example CVX [41].

Proposition 4.5. *Algorithm 4 solves Problem (4.67) in the following sense. Denote by C_i the capacity estimate in the ith step.*

1. *For $\epsilon = 0$, $\{C_i\}_{i=1}^{\infty}$ converges to capacity C from below.*
2. *For $\epsilon > 0$, the error after terminating is bounded as*

$$C \geq \mathsf{C} - \frac{\epsilon}{w_{\min}}. \tag{4.68}$$

Proof. Convergence: The capacity estimates $\{C_i\}_{i=1}^{\infty}$ form a strictly monotonically increasing sequence. This can be seen as follows. Assume the algorithm does not terminate in the ith step. Thus,

$$\mathbb{I}(\boldsymbol{p}') - C_i\boldsymbol{w}^T\boldsymbol{p}' \geq \epsilon \tag{4.69}$$

$$\Rightarrow C_{i+1} = \frac{\mathbb{I}(\boldsymbol{p}')}{\boldsymbol{w}^T\boldsymbol{p}'} \geq C_i + \frac{\epsilon}{\boldsymbol{w}^T\boldsymbol{p}'} \tag{4.70}$$

$$\geq C_i + \frac{\epsilon}{w_{\max}} \tag{4.71}$$

which shows that $C_{i+1} > C_i$. The sequence is bounded from above by the capacity C. Thus, the sequence converges to some limit C_∞. We now show in two steps that the limit is equal to capacity.

First, assume C_i is a fixed point, i.e., $C_{i+1} = C_i$. Then C_i is equal to capacity, i.e., $C_i = \mathsf{C}$. This can be seen as follows.

$$\mathbb{I}(\boldsymbol{p}') - C_i \boldsymbol{w}^T \boldsymbol{p}' = \mathbb{I}(\boldsymbol{p}') - C_{i+1} \boldsymbol{w}^T \boldsymbol{p}' \tag{4.72}$$

$$= 0 \tag{4.73}$$

where equality in the first line follows from the fixed point assumption and where equality in the second line follows from the assignment in step 1. of the algorithm in the $(i+1)$th iteration. Thus, because of step 2. of the algorithm in the ith iteration, for any pmf $\boldsymbol{p}$,

$$\mathbb{I}(\boldsymbol{p}) - C_i \boldsymbol{w}^T \boldsymbol{p} \leq 0, \quad \text{with equality if } \boldsymbol{p} = \boldsymbol{p}' \tag{4.74}$$

$$\Rightarrow \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T \boldsymbol{p}} \leq C_i, \quad \text{with equality if } \boldsymbol{p} = \boldsymbol{p}'. \tag{4.75}$$

Thus $C_i = \mathsf{C}$.

Next, we show that the limit C_∞ is a fixed point. To this end, define the function

$$f(C) = \frac{\mathbb{I}(\boldsymbol{p}')}{\boldsymbol{w}^T \boldsymbol{p}'} : \boldsymbol{p}' = \operatorname*{argmax}_{\text{pmf } \boldsymbol{p}} \mathbb{I}(\boldsymbol{p}) - C \boldsymbol{w}^T \boldsymbol{p}. \tag{4.76}$$

Note that $C_{i+1} = f(C_i)$. The value C_∞ is the limit of the sequence and the function f is continuous in C, so for each $\epsilon > 0$, there exists an i_0 such that for each i with $i \geq i_0$, it holds that $|C_\infty - C_i| < \frac{\epsilon}{2}$ and $|f(C_\infty) - f(C_i)| < \frac{\epsilon}{2}$. Thus

$$|f(C_\infty) - C_\infty| = |f(C_\infty) - C_\infty + C_{i+1} - C_{i+1}| \tag{4.77}$$

$$\leq |f(C_\infty) - C_{i+1}| + |C_{i+1} - C_\infty| \tag{4.78}$$

$$= |f(C_\infty) - f(C_i)| + |C_{i+1} - C_\infty| \tag{4.79}$$

$$\leq \frac{\epsilon}{2} + \frac{\epsilon}{2} \tag{4.80}$$

$$= \epsilon \tag{4.81}$$

where the inequality in the second line follows from the triangular inequality. This holds for any $\epsilon > 0$, thus $f(C_\infty) = C_\infty$, i.e., the limit C_∞ is a fixed point and equal to capacity.

Error bound: Assume the algorithm terminates. Then, from step 2. and the terminating condition,

$$\forall \boldsymbol{p} : \mathbb{I}(\boldsymbol{p}) - C \boldsymbol{w}^T \boldsymbol{p} < \epsilon \tag{4.82}$$

$$\Rightarrow \forall \boldsymbol{p} : \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T \boldsymbol{p}} < C + \frac{\epsilon}{\boldsymbol{w}^T \boldsymbol{p}} \leq C + \frac{\epsilon}{w_{\min}} \tag{4.83}$$

this holds for any pmf, and in particular, the capacity-achieving pmf $\boldsymbol{p}^*$. Consequently,

$$\frac{\mathbb{I}(\boldsymbol{p}^*)}{\boldsymbol{w}^T \boldsymbol{p}^*} = \mathsf{C} \tag{4.84}$$

$$\leq C + \frac{\epsilon}{w_{\min}}. \tag{4.85}$$

Solving for C, we get

$$C \geq \mathsf{C} - \frac{\epsilon}{w_{\min}} \tag{4.86}$$

which is the statement in the proposition. □

4.3.2 Capacity-achieving pmf

As we have shown, capacity C and capacity-achieving pmf $\boldsymbol{p}^*$ can efficiently be found by Algorithm 4. The topic of this subsection is to analytically characterize the entries of $\boldsymbol{p}^*$.

Proposition 4.6. *For a dmc with possibly unequal symbol durations $\boldsymbol{w}$ and capacity C, the following conditions are necessary and sufficient for an input pmf $\boldsymbol{p}$ to be capacity-achieving:*

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} = w_i \mathsf{C}, \qquad \forall i : p_i > 0 \tag{4.87}$$

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq w_i \mathsf{C}, \qquad \forall i : p_i = 0. \tag{4.88}$$

Proof. Denote by $\boldsymbol{p}$ an arbitrary pmf and by $\boldsymbol{p}^*$ a capacity-achieving pmf. We have

$$\frac{-\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}} \geq -\mathsf{C}, \qquad \text{with equality if } \boldsymbol{p} = \boldsymbol{p}^* \tag{4.89}$$

$$\Rightarrow -\mathbb{I}(\boldsymbol{p}) + \mathsf{C}\boldsymbol{w}^T\boldsymbol{p} \geq 0, \qquad \text{with equality if } \boldsymbol{p} = \boldsymbol{p}^*. \tag{4.90}$$

Thus, the capacity-achieving pmf $\boldsymbol{p}^*$ is also a solution of minimizing the left-hand side in the last line and it is therefore given by the optimal point of

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{minimize}} \quad & -\mathbb{I}(\boldsymbol{p}) + \mathsf{C}\boldsymbol{w}^T\boldsymbol{p} \\ \text{subject to} \quad & \mathbf{1}^T\boldsymbol{p} - 1 = 0. \end{aligned} \tag{4.91}$$

where the domain of the objective function is $\mathbf{R}^n_{\geq 0}$. For this modified problem, by Proposition 2.5, strong duality holds. By Proposition 2.7, the partial derivatives of the objective function are well-defined with the possible exception of taking the value $-\infty$ on the boundary. Thus, Proposition 2.4 applies. The Lagrangian is

$$L(\boldsymbol{p}, \nu) = -\mathbb{I}(\boldsymbol{p}) + \mathsf{C}\boldsymbol{w}^T\boldsymbol{p} + \nu(\mathbf{1}^T\boldsymbol{p} - 1) \tag{4.92}$$

Note that any pmf $\boldsymbol{p}$ is feasible. Thus, by Proposition 2.4, a pmf $\boldsymbol{p}$ is optimal if and only if the KKT conditions are fulfilled, i.e.,

$$\frac{\partial L(\boldsymbol{p}, \nu)}{\partial p_i} = -\frac{\partial \mathbb{I}(\boldsymbol{p})}{\partial p_i} + \mathsf{C}w_i + \nu = 0, \qquad \forall i : p_i > 0 \tag{4.93}$$

$$-\frac{\partial \mathbb{I}(\boldsymbol{p})}{\partial p_i} + \mathsf{C}w_i + \nu \geq 0, \qquad \forall i : p_i = 0. \tag{4.94}$$

If $\boldsymbol{p}$ fulfills these conditions, then all partial derivatives of $\mathbb{I}$ in $\boldsymbol{p}$ are well-defined and given by

$$\frac{\partial \mathbb{I}(\boldsymbol{p})}{\partial p_i} = \sum_j h_{ji} \log \frac{h_{ji}}{r_j} - 1, \qquad i = 1, \dots, n. \tag{4.95}$$

Plugging this into the KKT conditions, we get

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq 1 + \mathsf{C} w_i + \nu, \qquad \text{with equality if } p_i > 0. \tag{4.96}$$

For a capacity-achieving pmf $\boldsymbol{p}^*$, we have

$$\begin{aligned} \mathsf{C} \boldsymbol{w}^T \boldsymbol{p}^* &= \mathbb{I}(\boldsymbol{p}^*) && (4.97)\\ &= \sum_i p_i^* \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} && (4.98)\\ &= \sum_i p_i^* (1 + \mathsf{C} w_i + \nu) && (4.99)\\ &= 1 + \mathsf{C} \boldsymbol{w}^T \boldsymbol{p}^* + \nu && (4.100) \end{aligned}$$

where we used (4.96) in the third line. Note that (4.96) holds with equality for all i with $p_i^* > 0$. Comparing first and last line, we conclude $1 + \nu = 0$ and (4.96) becomes

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq \mathsf{C} w_i, \qquad \text{with equality if } p_i > 0. \tag{4.101}$$

This concludes the proof. □

4.3.3 Using a 'wrong' pmf

We now quantify the penalty that results from using an input pmf different from the capacity-achieving one.

Proposition 4.7. *Consider a dmc with possibly unequal symbol durations $\boldsymbol{w}$. Denote by C and $\boldsymbol{p}^*$ capacity and capacity-achieving pmf, respectively. Let $\boldsymbol{p}$ be some pmf with the only restriction that*

$$p_i = 0, \qquad \text{whenever } p_i^* = 0. \tag{4.102}$$

Then the mutual information per average duration achieved by $\boldsymbol{p}$ can be expressed as

$$\frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T \boldsymbol{p}} = \mathsf{C} - \frac{\mathbb{D}(\boldsymbol{r} \| \boldsymbol{r}^*)}{\boldsymbol{w}^T \boldsymbol{p}}. \tag{4.103}$$

Proof. We write the mutual information achieved by $\boldsymbol{p}$ as

$$\mathbb{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*) \tag{4.104}$$

$$= \sum_i p_i \mathsf{C} w_i - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*) \tag{4.105}$$

$$= \mathsf{C}\boldsymbol{w}^T\boldsymbol{p} - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*) \tag{4.106}$$

where we used Proposition 3.10 in the first line and Proposition 4.6 and assumption (4.102) in the second line. Dividing by $\boldsymbol{w}^T\boldsymbol{p}$ yields

$$\frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}} = \mathsf{C} - \frac{\mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)}{\boldsymbol{w}^T\boldsymbol{p}} \tag{4.107}$$

which is the statement of the proposition. □

4.3.4 Matching

Denote by $\boldsymbol{p}^*$ the capacity-achieving pmf of a dmc with capacity C. Define the dyadic pmf $\boldsymbol{d} = \textsc{nGhc}(\boldsymbol{p}^*, \boldsymbol{w})$ and denote by $\boldsymbol{r}$ and $\boldsymbol{r}^*$ the output pmfs resulting from $\boldsymbol{d}$ and $\boldsymbol{p}^*$, respectively. The achieved mutual information per average duration is lower-bounded as

$$\frac{\mathbb{I}(\boldsymbol{d})}{\boldsymbol{w}^T\boldsymbol{d}} = \mathsf{C} - \frac{\mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)}{\boldsymbol{w}^T\boldsymbol{d}} \tag{4.108}$$

$$\geq \mathsf{C} - \frac{\mathbb{D}(\boldsymbol{d}\|\boldsymbol{p}^*)}{\boldsymbol{w}^T\boldsymbol{d}} \tag{4.109}$$

where equality in the first line follows from Proposition 4.7, which applies because of Proposition 3.2. In the second line we used Proposition 3.12. According to Proposition 4.1, nGhc maximizes the lower-bound on $\mathbb{I}(\boldsymbol{d})/(\boldsymbol{w}^T\boldsymbol{d})$ in the last line over all dyadic pmfs.

4.3.5 Asymptotic achievability

We conclude this section by showing that nGhc asymptotically achieves capacity. We use the notation from Subsection 4.1.3.

Proposition 4.8. *Denote by* C *and* $\boldsymbol{p}^*$ *the capacity and the capacity-achieving pmf, respectively, of a dmc with possibly unequal symbol durations* $\boldsymbol{w}$*. For the dyadic pmf* $\boldsymbol{d}_k = \textsc{nGhc}(\boldsymbol{p}^{*k}, \boldsymbol{v}_k = \oplus^k \boldsymbol{w})$*, we have*

$$\frac{\mathbb{I}(\boldsymbol{d}_k)}{\boldsymbol{v}_k^T\boldsymbol{d}_k} \overset{k\to\infty}{\longrightarrow} \mathsf{C} \tag{4.110}$$

i.e., nGhc *is asymptotically capacity-achieving.*

Proof. Denote by $\boldsymbol{r}_k$ and $\boldsymbol{r}^{*k}$ the output pmfs that result from the input pmfs $\boldsymbol{d}_k$ and $\boldsymbol{p}^{*k}$, respectively. We have

$$\frac{\mathbb{I}(\boldsymbol{d}_k)}{\boldsymbol{v}_k^T \boldsymbol{d}_k} = \mathsf{C} - \frac{\mathbb{D}(\boldsymbol{r}_k \| \boldsymbol{r}^{*k})}{\boldsymbol{v}_k^T \boldsymbol{d}_k} \tag{4.111}$$

$$\geq \mathsf{C} - \frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{p}^{*k})}{\boldsymbol{v}_k^T \boldsymbol{d}_k} \tag{4.112}$$

$$\overset{k \to \infty}{\longrightarrow} \mathsf{C} \tag{4.113}$$

where the first line follows from Proposition 4.7, which applies because of Proposition 3.2. The second line follows from Proposition 3.12 and the last line follows from Proposition 4.4. This concludes the proof. □

4.4 References

Lempel *et al* proposed in [53] an algorithm that finds the optimal dyadic pmf for a noiseless channel with unequal symbol durations. In the literature, their algorithm is usually called the *Lempel-Even-Cohn* (LEC) algorithm, see for example Abrahams [2, 3]. It can be shown that the LEC algorithm is basically equivalent to nGHC, however, it misses an efficient way to determine where the dyadic pmf should have entries of value zero. Abrahams used in [2] the LEC Algorithm to determine a pmf of signal points that maximizes the energy efficiency in terms of bits/energy.

Capacity and capacity-achieving pmf of noiseless channels with unequal symbol durations are derived using different methods by Marcus [60, Page 8–9], by Krause [51, Theorem 1], and by ourselves [16, Lemma 1].

Kerpez suggests in [48] to use the Huffman source code of the capacity-achieving pmf as a prefix-free matcher for a class of noiseless channels with unequal symbol durations. In [48, Theorem 1], he also claims asymptotic achievability for this approach. However, the proof is rough and difficult to follow. There is a vast amount of literature on source coding and matching for source symbols with non-uniform pmfs and channel symbols with unequal durations, respectively. See Abrahams [1] and references therein.

Verdú called in [75, Theorem 2] the capacity of a dmc with unequal symbol durations *capacity per unit cost*. He showed that the operational capacity per unit cost is actually given by the solution of (4.67). In contrast to our work, he allowed one symbol to be of duration zero. Jimbo and Kunisawa provide in [47] an algorithm that solves (4.67). They also state without proof Proposition 4.6 in [47, Lemma 2]. An additional alternative to solve Problem (4.67) is to observe that the objective function is quasiconvex and then to solve the problem via the bisection method as Boyd and Vandenberghe propose in [19, Section 4.2.5]. An example of a dmc with unequal symbol durations is formulated by MacKay in [56].

5 Matching channels with cost constraints

In this chapter, we consider another variation of the basic setup from Chapter 3. We let all symbols be of equal duration, but we now associate with each symbol a cost. Different symbols can have different costs. In addition, we introduce a cost constraint E. All optimization problems in this chapter are subject to the constraint that the average cost cannot exceed E. For example, the capacity of a dmc with symbol costs $\boldsymbol{w}$ is now given by

$$\mathsf{C} = \max_{\text{pmf } \boldsymbol{p}:\boldsymbol{w}^T\boldsymbol{p}\le E} \mathbb{I}(\boldsymbol{p}). \tag{5.1}$$

The goal of this chapter is to find good dyadic pmfs under the new constraint. We start with the problem of minimizing the relative entropy for a given non-negative target vector over all dyadic pmfs subject to the cost constraint. We then apply the results to noiseless channels and dmcs and show that capacity can asymptotically be achieved by prefix-free matchers.

5.1 Cost constrained geometric Huffman coding

Consider a non-negative vector $\boldsymbol{x}$ and a positive cost vector $\boldsymbol{w}$. The objective is to approximate $\boldsymbol{x}$ by a dyadic pmf $\boldsymbol{d}$ such that the relative entropy $\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x})$ is minimized subject to an average cost constraint E, i.e., the objective is to solve

$$\begin{aligned} \underset{\boldsymbol{d} \text{ dyadic}}{\text{minimize}} \quad & \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{w}^T\boldsymbol{d} \le E. \end{aligned} \tag{5.2}$$

We do not know an efficient algorithm that directly solves this problem. To tackle this problem, we include the cost constraint into the objective function by adding a scaled version $\xi\boldsymbol{w}^T\boldsymbol{d}$ of the cost constraint to the objective function. This can be written as

$$\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}) + \xi\boldsymbol{w}^T\boldsymbol{d} = \sum_i d_i \log\frac{d_i}{x_i} + \sum_i d_i\xi w_i \tag{5.3}$$

$$= \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}\circ e^{-\xi\boldsymbol{w}}). \tag{5.4}$$

By Proposition 3.3, the term in the last line is minimized over all dyadic pmfs by $\boldsymbol{d} = \textsc{Ghc}(\boldsymbol{x}\circ e^{-\xi\boldsymbol{w}})$. The cost constraint can be guaranteed by iteratively adapting ξ: if for the resulting $\boldsymbol{d}$, $\boldsymbol{w}^T\boldsymbol{d} > E$, increase ξ and repeat, if $\boldsymbol{w}^T\boldsymbol{d} < E$, decrease ξ and repeat. Thus, the solution can be found by bisection. In summary, we have the following algorithm, which we call *cost constrained geometric Huffman coding* (ccGhc).

Algorithm 5.(ccGhc)

$\ell < u$
$\epsilon > 0$
repeat
1. $\xi = \frac{\ell+u}{2}$
2. $\boldsymbol{d} = \textsc{Ghc}(\boldsymbol{x} \circ e^{-\xi \boldsymbol{w}})$
3. **if** $\boldsymbol{w}^T \boldsymbol{d} \leq E$, $u \leftarrow \xi$; **else** $\ell \leftarrow \xi$

until $u - \ell < \epsilon$
$\xi = u$
$\boldsymbol{d} = \textsc{Ghc}(\boldsymbol{x} \circ e^{-\xi \boldsymbol{w}})$

We will argue in Subsection 5.3 why ccGhc not always finds the optimal dyadic pmf for Problem (5.2), but we will show in Subsection 5.1.4 that ccGhc asymptotically achieves the optimal value.

5.1.1 Optimal pmf

To be able to quantify the performance of ccGhc, we first need to derive what can be achieved without the restriction to dyadic pmfs. We start by characterizing the optimal possibly non-dyadic pmf. We define the *distance-cost function* $\mathsf{D}(E)$ pointwise by the solution of

$$\begin{array}{ll} \underset{\boldsymbol{p}}{\text{minimize}} & \mathbb{D}(\boldsymbol{p} \| \boldsymbol{x}) \\ \text{subject to} & \boldsymbol{w}^T \boldsymbol{p} - E \leq 0 \\ & \mathbf{1}^T \boldsymbol{p} - 1 = 0 \end{array} \tag{5.5}$$

where the domain of the problem is $\mathbf{R}^n_{\geq 0}$. That is, if $\boldsymbol{p}^*$ is the optimal pmf for a specific E, we define

$$\mathsf{D}(E) := \mathbb{D}(\boldsymbol{p}^* \| \boldsymbol{x}) \tag{5.6}$$

Define

$$w_{\min} := \min_{i : x_i > 0} w_i. \tag{5.7}$$

The objective can only take finite values if $E \geq w_{\min}$. The unconstrained solution is by Proposition 3.4 given by

$$\boldsymbol{t} := \frac{\boldsymbol{x}}{\sum_i x_i}. \tag{5.8}$$

For the rest of this section, we assume $w_{\min} < \boldsymbol{w}^T \boldsymbol{t}$ and we say the constraint is *active* if $w_{\min} < E < \boldsymbol{w}^T \boldsymbol{t}$.

Proposition 5.1. *Given is a non-negative target vector $\boldsymbol{x}$, symbol costs $\boldsymbol{w}$, and an active cost constraint E. Then necessary and sufficient conditions for a feasible pmf $\boldsymbol{p}$ to solve Problem (5.5) are*

$$p_i = 0, \qquad \forall i : x_i = 0 \tag{5.9}$$

$$\log p_i = \log x_i - 1 - \nu - \lambda w_i, \qquad \forall i : x_i > 0 \tag{5.10}$$

where ν is a finite real value and $\lambda > 0$. From these conditions, the capacity-achieving pmf $\boldsymbol{p}^*$ can be calculated as

$$p_i^* = \frac{x_i e^{-\lambda w_i}}{\sum_j x_j e^{-\lambda w_j}}, \qquad i = 1, \ldots, n \tag{5.11}$$

where λ is given by the solution of

$$\frac{\sum_i w_i x_i e^{-\lambda w_i}}{\sum_i x_i e^{-\lambda w_i}} = E. \tag{5.12}$$

Proof. The problem we want to solve is

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{minimize}} \quad & \mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{w}^T\boldsymbol{p} - E \leq 0 \\ & \mathbf{1}^T\boldsymbol{p} - 1 = 0 \end{aligned} \tag{5.13}$$

where the domain is $\mathbf{R}^n_{\geq 0}$. By Proposition 2.5 strong duality holds for this problem. By the convention $\log 0 = -\infty$, clearly, whenever $x_i = 0$, we have to assign $p_i = 0$, since otherwise, the objective function would take the value infinity. Therefore, without loss of generality, we assume for now that $x_i > 0$ for all i. Under this assumption, according to Proposition 2.6, the partial derivatives of $\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})$ are well-defined with the exception of taking the value $-\infty$ on the boundary of $\mathbf{R}^n_{\geq 0}$. All together, Proposition 2.4 applies. The Lagrangian is

$$L(\boldsymbol{p}, \lambda, \nu) = \mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) + \lambda(\boldsymbol{w}^T\boldsymbol{p} - E) + \nu(\mathbf{1}^T\boldsymbol{p} - 1). \tag{5.14}$$

According to Proposition 2.4, a feasible $\boldsymbol{p}$ is optimal if and only if the KKT conditions are fulfilled, i.e.,

$$\lambda \geq 0 \tag{5.15}$$

$$\lambda(\boldsymbol{w}^T\boldsymbol{p} - E) = 0 \tag{5.16}$$

$$\frac{\partial L(\boldsymbol{p}, \nu, \lambda)}{\partial p_i} = \frac{\partial\, \mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})}{\partial p_i} + \nu + \lambda w_i = 0, \qquad \forall i : p_i > 0 \tag{5.17}$$

$$\frac{\partial\, \mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})}{\partial p_i} + \nu + \lambda w_i \geq 0, \qquad \forall i : p_i = 0. \tag{5.18}$$

The two last conditions imply together with Proposition 2.6 that all partial derivatives of $\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})$ are well-defined in $\boldsymbol{p}$. In particular, $p_i > 0$, $i = 1, \ldots, n$, and by Proposition 2.6,

$$\frac{\partial\, \mathbb{D}(\boldsymbol{p}\|\boldsymbol{x})}{\partial p_i} = \log \frac{p_i}{x_i} + 1, \qquad i = 1, \ldots, n. \tag{5.19}$$

Thus

$$\log p_i = \log x_i - 1 - \nu - \lambda w_i, \qquad i = 1, \ldots, n. \tag{5.20}$$

Since $\boldsymbol{p}$ is a pmf, its entries have to sum up to one. Therefore,

$$p_i = \frac{p_i}{\sum_j p_j} = \frac{x_i e^{-1-\nu-\lambda w_i}}{\sum_j x_j e^{-1-\nu-\lambda w_j}} \tag{5.21}$$

$$= \frac{x_i e^{-\lambda w_i}}{\sum_j x_j e^{-\lambda w_j}}. \tag{5.22}$$

Furthermore, since by assumption the constraint is active, $\lambda > 0$ and condition (5.16) holds if and only if

$$\boldsymbol{w}^T \boldsymbol{p} = E \tag{5.23}$$

$$\Leftrightarrow \frac{x_i w_i e^{-\lambda w_i}}{\sum_j x_j e^{-\lambda w_j}} = E. \tag{5.24}$$

Note that the resulting pmf $\boldsymbol{p}$ is actually feasible.

In the case when $x_i = 0$ for some i, as argued above, an optimal pmf has to assign $p_i = 0$. This assignment is provided by (5.22), so this formula yields the optimal pmf for any non-negative vector $\boldsymbol{x}$. This concludes the proof. □

5.1.2 Strict convexity of distance-cost function

Next, we show that the distance-cost function is a strictly convex function of the cost constraint when the constraint is active. We need this result to show asymptotic achievability for CCGHC.

Proposition 5.2. *For an active cost constraint E, the distance-cost function $\mathsf{D}(E)$ is strictly convex in E.*

Proof. Denote by $\boldsymbol{p}^*$ an optimal pmf for cost constraint E. Since by assumption the cost constraint is active, $\lambda > 0$. Thus, by (5.16), $\boldsymbol{w}^T \boldsymbol{p}^* = E$, i.e,

$$\boldsymbol{w}^T \boldsymbol{p}^* = \frac{\sum_i w_i x_i e^{-\lambda w_i}}{\sum_j x_j e^{-\lambda w_j}} \tag{5.25}$$

$$=: f(\lambda) \tag{5.26}$$

$$= E. \tag{5.27}$$

We differentiate $f(\lambda)$ and get

$$\frac{\partial f(\lambda)}{\partial \lambda} = \frac{\sum_i \sum_j (w_i w_j - w_i^2) x_i x_j e^{-\lambda(w_i+w_j)}}{\left[\sum_j x_j e^{-\lambda w_j}\right]^2} \tag{5.28}$$

We now want to show that $\frac{\partial f(\lambda)}{\partial \lambda} < 0$. Since the denominator is positive, we only need to consider the numerator. We have

$$\begin{aligned}
&\sum_i \sum_j (w_i w_j - w_i^2) x_i x_j e^{-\lambda(w_i+w_j)} \\
&\qquad = \sum_i \sum_{j>i} (w_i w_j - w_i^2 + w_j w_i - w_j^2) x_i x_j e^{-\lambda(w_i+w_j)} && (5.29)\\
&\qquad = \sum_i \sum_{j>i} [-(w_i - w_j)^2] x_i x_j e^{-\lambda(w_i+w_j)} && (5.30)\\
&\qquad \leq 0 && (5.31)
\end{aligned}$$

where the inequality follows because each summand is smaller or equal to 0. We now show that the inequality in the last line can be replaced by a strict inequality. By assumption, E is active, and in particular, $w_{\min} < E$. Denote by m the index of $w_{\min}$ in $\boldsymbol{w}$. By definition of $w_{\min}$, $x_m > 0$. An active cost constraint is fulfilled with equality, therefore, there has to be another index k with $w_k > w_{\min}$ and $x_k > 0$. Consequently, the corresponding summand has to be negative, i.e.,

$$[-(w_m - w_k)^2] x_m x_k e^{-\lambda(w_m+w_k)} < 0. \tag{5.32}$$

As a result, the sum in (5.30) is also strictly smaller than zero and the inequality in (5.31) can be replaced by a strict inequality. Thus, f is strictly monotonically decreasing and thereby invertible on its image, i.e., on $(w_{\min}, \boldsymbol{w}^T \boldsymbol{t})$. Recall that we defined $\boldsymbol{t}$ as the optimal pmf of the unconstrained problem. Consequently, $\lambda = f^{-1}(E)$ is strictly monotonically decreasing. By [19, Section 5.6.3],

$$\lambda = -\frac{\partial \mathsf{D}(E)}{\partial E} \tag{5.33}$$

thus,

$$\frac{\partial^2 \mathsf{D}(E)}{\partial E^2} = -\frac{\partial f^{-1}(E)}{\partial E} > 0 \tag{5.34}$$

which is by [24, Theorem 2.6.1] a sufficient condition for the strict convexity of $\mathsf{D}(E)$ in E. □

5.1.3 Using a 'wrong' pmf

We now express the relative entropy achieved by a pmf different from the optimal one in terms of the optimal pmf and the cost constraint.

Proposition 5.3. *Given is a non-negative target vector $\boldsymbol{x}$ and a positive cost vector $\boldsymbol{w}$. For a given active cost constraint E, denote by $\boldsymbol{p}^*$ an optimal pmf. Denote by $\boldsymbol{p}$ an arbitrary pmf with the only restriction that*

$$p_i = 0, \qquad \text{whenever } p_i^* = 0. \tag{5.35}$$

Then

$$\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) = \mathsf{D}(E) - \lambda(\boldsymbol{w}^T\boldsymbol{p} - E) + \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*). \tag{5.36}$$

where $-\lambda$ is the slope of the tangent of D in $[E, \mathsf{D}(E)]$, i.e.,

$$-\lambda = \frac{\partial \mathsf{D}(E)}{\partial E}. \tag{5.37}$$

Proof. We write the relative entropy of $\boldsymbol{p}$ and $\boldsymbol{x}$ as

$$\begin{aligned}
\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) &= \sum_i p_i \log \frac{p_i}{x_i} && (5.38)\\
&= \sum_i p_i \log \frac{p_i p_i^*}{x_i p_i^*} && (5.39)\\
&= \sum_i p_i \log \frac{p_i^*}{x_i} + \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) && (5.40)\\
&= \sum_i p_i \log p_i^* - \sum_i p_i \log x_i + \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) && (5.41)
\end{aligned}$$

where the second line is well-defined even if $p_i^* = 0$ for some i because of (5.35) and $0 \log \frac{0}{0} = 0$. We further develop the first term:

$$\begin{aligned}
\sum_i p_i \log p_i^* &= \sum_i (p_i + p_i^* - p_i^*) \log p_i^* && (5.42)\\
&= -\mathbb{H}(\boldsymbol{p}^*) + \sum_i (p_i - p_i^*) \log p_i^* && (5.43)\\
&= -\mathbb{H}(\boldsymbol{p}^*) + \sum_i (p_i - p_i^*)(\log x_i + 1 - \nu - \lambda w_i) && (5.44)\\
&= -\mathbb{H}(\boldsymbol{p}^*) - \lambda(\boldsymbol{w}^T\boldsymbol{p} - \boldsymbol{w}^T\boldsymbol{p}^*) + \sum_i p_i \log x_i - \sum_i p_i^* \log x_i && (5.45)\\
&= \mathbb{D}(\boldsymbol{p}^*\|\boldsymbol{x}) - \lambda(\boldsymbol{w}^T\boldsymbol{p} - \boldsymbol{w}^T\boldsymbol{p}^*) + \sum_i p_i \log x_i && (5.46)
\end{aligned}$$

where we used Proposition 5.1 in the third line. Using (5.46) in (5.41), we get

$$\begin{aligned}
\mathbb{D}(\boldsymbol{p}\|\boldsymbol{x}) &= \mathbb{D}(\boldsymbol{p}^*\|\boldsymbol{x}) - \lambda(\boldsymbol{w}^T\boldsymbol{p} - \boldsymbol{w}^T\boldsymbol{p}^*) + \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) && (5.47)\\
&= \mathsf{D}(E) - \lambda(\boldsymbol{w}^T\boldsymbol{p} - E) + \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) && (5.48)
\end{aligned}$$

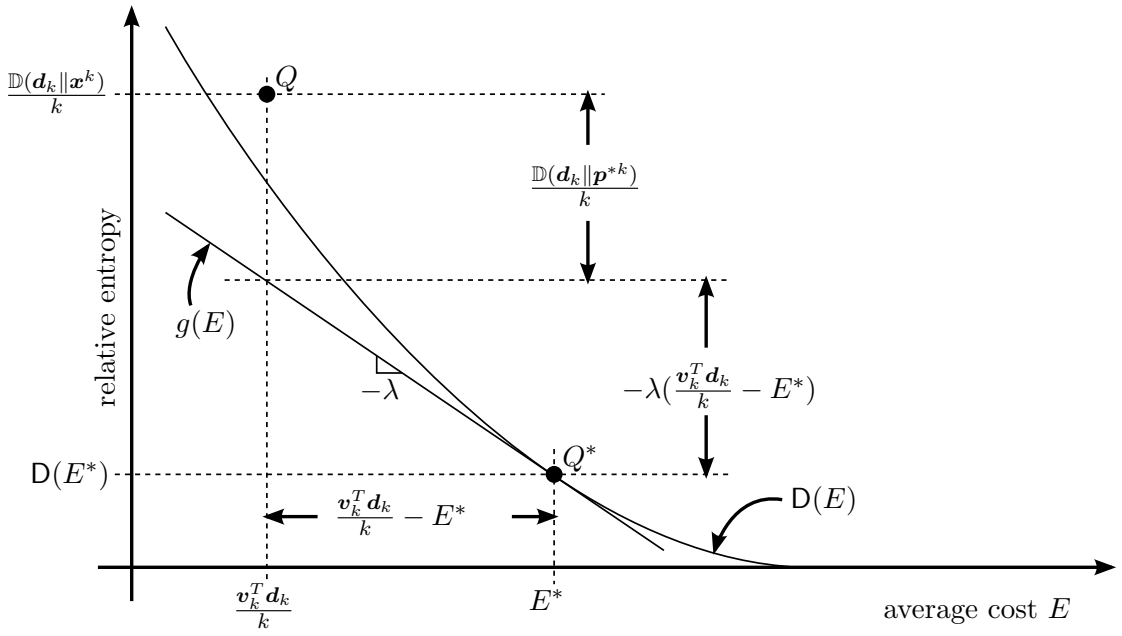


Figure 5.1: Existence of good dyadic operating points.

where $\boldsymbol{w}^T\boldsymbol{p}^* = E$ in the second line since the constraint is by assumption active. According to [19, Section 5.6.3], for an active constraint E, λ is given by

$$\lambda = -\frac{\partial \mathsf{D}(E)}{\partial E}. \tag{5.49}$$

This concludes the proof. □

5.1.4 Asymptotic Achievability

We finally show that CCGHC is asymptotically capacity-achieving. Given is a non-negative target vector $\boldsymbol{x}$, positive costs $\boldsymbol{w}$, and an active cost constraint E^*. We consider k consecutive symbols. The corresponding target vector is given by the Kronecker product $\boldsymbol{x}^k$ of k copies of $\boldsymbol{x}$. The cost vector is given by the cost sum $\boldsymbol{v}_k = \oplus^k \boldsymbol{w}$ where we used the notation from Subsection 4.1.3. The cost constraint becomes kE^* and for an active cost constraint, the optimal pmf $\boldsymbol{p}^{*k}$ fulfills $\boldsymbol{v}_k\boldsymbol{p}^{*k} = kE^*$. For k to infinity, we have the following result.

Proposition 5.4. *Given is a non-negative target vector $\boldsymbol{x}$, positive costs $\boldsymbol{w}$, and an active cost constraint E^*. Define $\boldsymbol{d}_k = \text{CCGHC}(\boldsymbol{x}^k, \boldsymbol{v}_k = \oplus^k\boldsymbol{w}, kE^*)$. Then,*

$$\frac{\mathbb{D}(\boldsymbol{d}_k\|\boldsymbol{x}^k)}{k} \to \mathbb{D}(\boldsymbol{p}^*\|\boldsymbol{x}) \tag{5.50}$$

$$\text{and } \frac{\boldsymbol{v}_k^T\boldsymbol{d}_k}{k} \to E^*,\ \frac{\boldsymbol{v}_k^T\boldsymbol{d}_k}{k} \le E^* \tag{5.51}$$

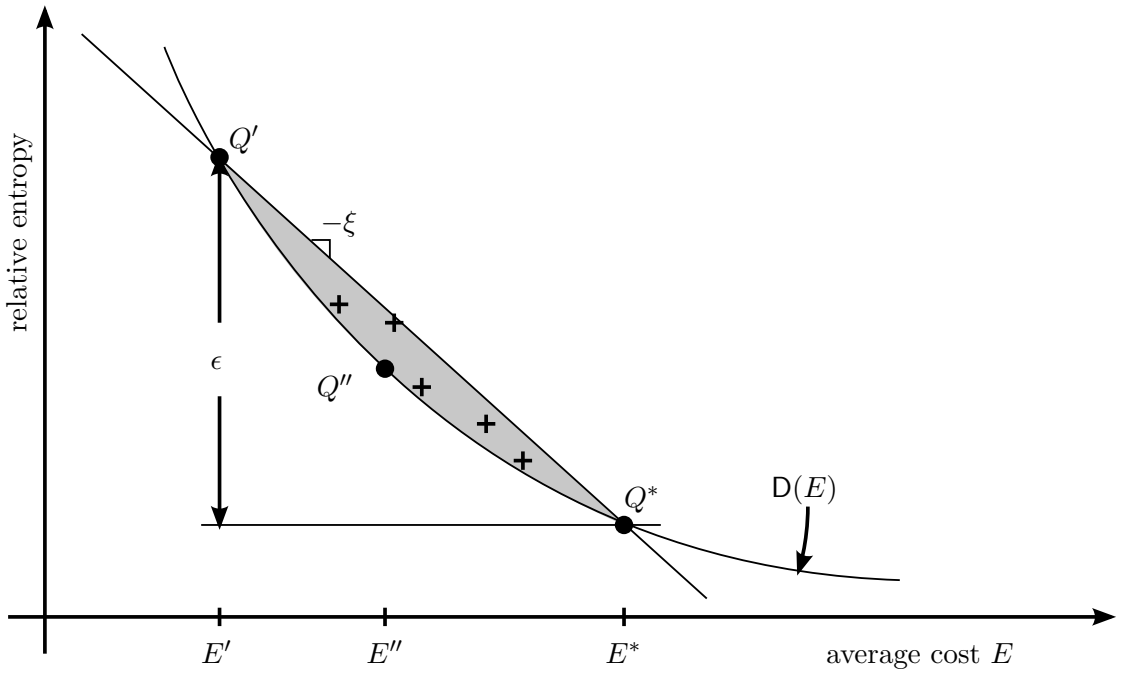


Figure 5.2: Finding good dyadic operating points.

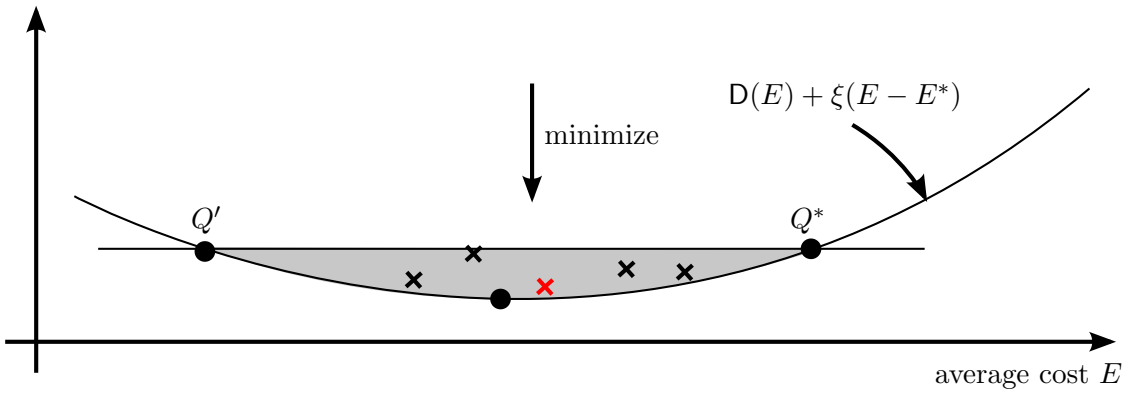


Figure 5.3: Rotated relative entropy geometry.

i.e., the relative entropy per symbol converges to the optimal value and the average cost per symbol converges to the cost constraint E^*, while the cost constraint is always fulfilled.

Proof. We now show that for any active cost constraint E^*, the target operating point $Q^* = [E^*, \mathsf{D}(E^*)]$ can be achieved by a dyadic pmf. We do this in two steps. First, we show the existence of dyadic operating points close to the target operating point, and then we show that ccGhc actually finds them.

Existence of good dyadic points. Consider the optimal pmf $\boldsymbol{p}^{*k}$ of k consecutive symbols. Define $\boldsymbol{v}_k = \oplus^k \boldsymbol{w}$. Furthermore, define $\boldsymbol{d}_k = \textsc{Ghc}(\boldsymbol{p}^{*k})$. Proposition 5.3 applies for $\boldsymbol{d}_k$ because of Proposition 3.2 and the relative entropy achieved by $\boldsymbol{d}_k$ can be written as

$$\frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{x}^k)}{k} = \mathsf{D}(E^*) - \lambda\Big(\frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k} - E^*\Big) + \frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{p}^{*k})}{k}. \tag{5.52}$$

By Proposition 3.8, since $\boldsymbol{d}_k = \textsc{Ghc}(\boldsymbol{p}^{*k})$, the normalized relative entropy on the right-hand side goes to zero as $k \to \infty$. Consider now Figure 5.1. The tangent of $\mathsf{D}(E)$ in Q^* is given by

$$g(E) := \mathsf{D}(E^*) - \lambda(E - E^*). \tag{5.53}$$

As the normalized relative entropy of $\boldsymbol{d}_k$ and $\boldsymbol{p}^{*k}$ gets smaller, the normalized relative entropy of $\boldsymbol{d}_k$ and $\boldsymbol{x}^k$ on the left-hand side of (5.52) is approaching the tangent g from above. However, because the tangent is linear in E and D is according to Proposition 5.2 strictly convex in E and lower-bounds $\frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{x}^k)}{k}$, the dyadic operating point $Q = (\frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k}, \frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{x}^k)}{k})$ has to approach Q^* both in terms of distance and cost. However, Q may approach Q^* also from the right, i.e., there is no guarantee that for $\boldsymbol{d}_k = \textsc{Ghc}(\boldsymbol{p}^{*k})$, the cost constraint is fulfilled.

Finding good dyadic points. The algorithm ccGhc guarantees that the cost constraint is fulfilled. It remains to show that ccGhc finds good dyadic points. This can best be seen in Figure 5.2. Suppose we want to find a dyadic pmf $\boldsymbol{d}_k$ such that for a given $\epsilon > 0$,

$$\frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{x})}{k} \leq \mathsf{D}(E^*) + \epsilon \text{ and } \frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k} \leq E^*. \tag{5.54}$$

Define

$$E' : \mathsf{D}(E') = \mathsf{D}(E^*) + \epsilon \tag{5.55}$$

and define further

$$E'' = \frac{E' + E^*}{2}. \tag{5.56}$$

The chord from $Q^* = [E^*, \mathsf{D}(E^*)]$ to $Q' = [E', \mathsf{D}(E')]$ cuts a segment from the area above D. Because of the strict convexity of D, this segment is nonempty. Note that all operating points in the segment fulfill the requirements (5.54). As shown in the first part

of this proof, for a large enough k, there exist dyadic operating points approximating $Q'' = [E'', \mathsf{D}(E'')]$ that lie within this segment. Define now ξ as the negative slope of the chord, i.e.,

$$\xi = -\frac{\mathsf{D}(E^*) - \mathsf{D}(E')}{E^* - E'}. \tag{5.57}$$

By adding $\xi \cdot (E - E^*)$ to the points in Figure 5.2, all points rotate around the target point Q^* such that Q' and Q^* are on the same height, i.e.,

$$\mathsf{D}(E') + \xi(E' - E^*) = \mathsf{D}(E') - \frac{\mathsf{D}(E^*) - \mathsf{D}(E')}{E^* - E'}(E' - E^*) \tag{5.58}$$

$$= \mathsf{D}(E^*). \tag{5.59}$$

See Figure 5.3 for an illustration. Since all dyadic points in the segment lie below Q^*, we can find them by minimization in the direction of the vertical axis. Thus, we assign

$$\boldsymbol{d}_k = \operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \frac{\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}^k)}{k} + \xi\Big(\frac{\boldsymbol{v}_k^T \boldsymbol{d}}{k} - E^*\Big) \tag{5.60}$$

$$= \operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \frac{\mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}^k)}{k} + \xi\frac{\boldsymbol{v}_k^T \boldsymbol{d}}{k} \tag{5.61}$$

$$= \operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}^k) + \xi\boldsymbol{v}_k^T \boldsymbol{d} \tag{5.62}$$

$$= \operatorname*{argmin}_{\text{dyadic } \boldsymbol{d}} \mathbb{D}(\boldsymbol{d}\|\boldsymbol{x}^k \circ e^{-\xi \boldsymbol{v}_k}) \tag{5.63}$$

$$= \textsc{Ghc}(\boldsymbol{x}^k \circ e^{-\xi \boldsymbol{v}_k}) \tag{5.64}$$

where the last line follows from Proposition 3.3. The value of ξ will also be evaluated by ccGhc, thus $\boldsymbol{d}_k = \textsc{ccGhc}(\boldsymbol{x}^k, \boldsymbol{v}_k, kE^*)$ will give a dyadic operating point at least as good as $\boldsymbol{d}_k = \textsc{Ghc}(\boldsymbol{x}^k \circ e^{-\xi \boldsymbol{v}_k})$. This concludes the proof. □

5.2 Noiseless channel

We now apply the results from the previous section to *noiseless channels with average cost constraint*. The input symbols are all of duration 1, but each symbol has an associated positive cost and for different symbols, the cost can be different. The average cost cannot exceed a prescribed value E. The capacity of such a channel is given by the maximum entropy rate subject to the cost constraint, i.e., by

$$\mathsf{H} = \max_{\text{pmf } \boldsymbol{p}: \boldsymbol{w}^T \boldsymbol{p} \le E} \mathbb{H}(\boldsymbol{p}) \tag{5.65}$$

where the entries of $\boldsymbol{w}$ are the costs of the symbols. When we generate the pmf by a prefix-free matcher, the maximum rate that we can achieve is given by the optimal value of the matching problem

$$\begin{aligned} &\operatorname*{maximize}_{\text{dyadic } \boldsymbol{d}} && \mathbb{H}(\boldsymbol{d}) \\ &\text{subject to} && \boldsymbol{w}^T \boldsymbol{d} \le E. \end{aligned} \tag{5.66}$$

As we have seen in Section 3.3 and Section 4.2, we can express entropy as a relative entropy, i.e.,

$$\mathbb{H}(\boldsymbol{p}) = -\,\mathbb{D}(\boldsymbol{p}\|\mathbf{1}). \tag{5.67}$$

Thus, all results that we derived for relative entropy in the previous section also apply to noiseless channels. We detail this in the following.

5.2.1 Matching

Because of (5.67), $\boldsymbol{d} = \textsc{ccGhc}(\mathbf{1}, \boldsymbol{w}, E)$ yields according to Section 5.1 a good feasible dyadic pmf for the matching problem (5.66).

5.2.2 Optimal pmf

Capacity H and capacity-achieving pmf $\boldsymbol{p}^*$ are given by the solution of the optimization problem

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{minimize}} \quad & -\,\mathbb{H}(\boldsymbol{p}) \\ \text{subject to} \quad & \boldsymbol{w}^T\boldsymbol{p} - E \leq 0 \\ & \mathbf{1}^T\boldsymbol{p} - 1 = 0 \end{aligned} \tag{5.68}$$

where the domain of the problem is $\mathbf{R}^n_{\geq 0}$. We define the *entropy-cost function* $\mathsf{H}(E)$ pointwise by the solution of (5.68), i.e., for the cost constraint E, $\mathsf{H}(E)$ is the optimal value of (5.68). Because of (5.67), Proposition 5.1 directly gives us the following result.

Proposition 5.5. *Given are symbol costs $\boldsymbol{w}$ and an active cost constraint E. Then necessary and sufficient conditions for a feasible pmf $\boldsymbol{p}$ to solve Problem (5.68) are*

$$\log p_i = 1 - \nu - \lambda w_i, \quad i = 1, \ldots, n \tag{5.69}$$

where ν is a finite real number and $\lambda > 0$. From these conditions, the capacity-achieving pmf $\boldsymbol{p}^*$ can be calculated as

$$p_i^* = \frac{e^{-\lambda w_i}}{\sum_j e^{-\lambda w_j}}, \quad i = 1, \ldots, n \tag{5.70}$$

where λ is given by the solution of

$$\frac{\sum_i w_i e^{-\lambda w_i}}{\sum_i e^{-\lambda w_i}} = E. \tag{5.71}$$

5.2.3 Strict concavity of entropy-cost function

Because of the relation (5.67), Proposition 5.2 gives us the following result for the entropy-cost function.

Proposition 5.6. *For an active cost constraint E, the entropy-cost function $\mathsf{H}(E)$ is strictly concave in E.*

5.2.4 Using a 'wrong' pmf

We now express the entropy rate achieved by some pmf $\boldsymbol{p}$ in terms of the cost constraint E and the capacity-achieving pmf $\boldsymbol{p}^*$. Because of (5.67), we get from Proposition 5.3 the following result.

Proposition 5.7. *Let $\boldsymbol{w}$ denote positive symbol costs. Denote by E an active cost constraint and by $\boldsymbol{p}^*$ the corresponding capacity-achieving pmf. Let $\boldsymbol{p}$ be an arbitrary input pmf with the only restriction that*

$$p_i = 0, \qquad \text{whenever } p_i^* = 0. \tag{5.72}$$

The entropy rate achieved by $\boldsymbol{p}$ can then be written as

$$\mathsf{H}(\boldsymbol{p}) = \mathsf{H}(E) + \lambda(\boldsymbol{w}^T\boldsymbol{p} - E) - \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) \tag{5.73}$$

where λ is the slope of the tangent of H in $[E, \mathsf{H}(E)]$, i.e.,

$$\lambda = \frac{\partial \mathsf{H}(E)}{\partial E}. \tag{5.74}$$

5.2.5 Asymptotic achievability

From Proposition 5.4, because of (5.67), we can conclude the asymptotic achievability of ccGhc for noiseless channels with cost constraints.

Proposition 5.8. *Given are positive symbol costs $\boldsymbol{w}$ and an active cost constraint E. Then $\boldsymbol{d}_k = \text{ccGhc}(\mathbf{1}^k, \boldsymbol{v}_k = \oplus^k \boldsymbol{w}, kE)$ achieves capacity, i.e.,*

$$\frac{\mathbb{H}(\boldsymbol{d}_k)}{k} \overset{k\to\infty}{\longrightarrow} \mathsf{H}(E) \tag{5.75}$$

$$\text{and } \frac{\boldsymbol{v}_k^T\boldsymbol{d}_k}{k} \le E, \quad \frac{\boldsymbol{v}_k^T\boldsymbol{d}_k}{k} \overset{k\to\infty}{\longrightarrow} E. \tag{5.76}$$

5.3 ccGhc is not necessarily optimal: an example

We consider the signal constellation of *quadrature amplitude modulation* with 16 signal points (16-QAM). The cost of a signal point x is given by $w = |x|^2$. We normalize by the greatest cost, and the resulting costs are given by

$$\boldsymbol{w} = \tfrac{1}{9}\begin{pmatrix} 9 & 5 & 5 & 9 \\ 5 & 1 & 1 & 5 \\ 5 & 1 & 1 & 5 \\ 9 & 5 & 5 & 9 \end{pmatrix}^T \tag{5.77}$$

where we wrote the 16 entries of $\boldsymbol{w}$ in four rows to mimic the signal constellation of 16-QAM. We want to maximize entropy subject to an average cost constraint E. For $E < w_{\min} = \frac{1}{9}$, the problem is infeasible. For $E = \frac{1}{9}$, only the inner four points can be

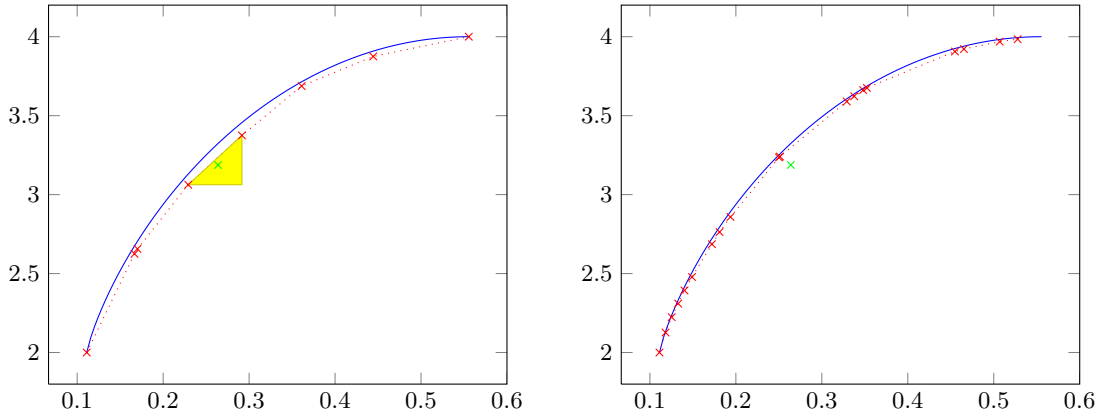


Figure 5.4: In both plots, the horizontal position gives the average cost and the vertical position the entropy. The blue lines mark the entropy-cost function. In the left plot, the red crosses mark the dyadic operating points found by ccGhc$(\mathbf{1}, \boldsymbol{w}, E)$. The red crosses in the right plot show the dyadic operating points found by ccGhc$(\mathbf{1}^2, \oplus^2 \boldsymbol{w}, 2E)$. The dotted line indicates the convex hull of the set of dyadic operating points. A yellow triangle marks a region that may contain optimal dyadic operating points that ccGhc cannot find. The dyadic pmf that generates the point marked by a green cross was found by random search. Its codeword lengths are given in (5.81).

used and entropy is according to Subsection 3.3.2 maximized by the uniform pmf over these four points, i.e., by

$$\boldsymbol{p}_{\min} = \tfrac{1}{4}(\begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{matrix})^T \tag{5.78}$$

and the entropy achieved by $\boldsymbol{p}_{\min}$ is 2 bits. According to Subsection 3.3.2, without the constraint, entropy is maximized by the uniform pmf over all 16 points, i.e., by

$$\boldsymbol{p}_{\max} = \tfrac{1}{16}(\begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{matrix})^T \tag{5.79}$$

and the entropy achieved by $\boldsymbol{p}_{\max}$ is 4 bits. The constraint is active for

$$\boldsymbol{w}^T \boldsymbol{p}_{\min} < E < \boldsymbol{w}^T \boldsymbol{p}_{\max} \tag{5.80}$$

and we use the formula from Proposition 5.5 to calculate the entropy-cost function. The resulting curve is displayed in Figure 5.4. We now use ccGhc to calculate dyadic pmfs. We call ccGhc$(\mathbf{1}, \boldsymbol{w}, E)$ for 100 values of E equally spaced in the active interval.

We get 8 distinct dyadic pmfs. The resulting dyadic operating points are displayed by red crosses in Figure 5.4. This small number is surprising. If for each value of E, the rightmost red cross to the left of E yields the optimal dyadic operating point under the constraint E, then this would in particular imply that there is no dyadic operating point in the yellow triangular in the figure. However, by randomly generating dyadic pmfs, we find the dyadic pmf

$$\boldsymbol{d}_{\text{rand}} = 2^{-(7\ 5\ 5\ 7\ 5\ 2\ 2\ 5\ 5\ 3\ 4\ 4\ 7\ 5\ 5\ 7)^T}. \tag{5.81}$$

The corresponding dyadic operating point is displayed by a green cross and lies inside the yellow triangular. So, for $E = \boldsymbol{w}^T\boldsymbol{d}_{\text{rand}}$, ccGhc could not find the optimal dyadic pmf. The reason is the following: ccGhc optimizes the sum of the objective function and a scaled version of the cost function. Consequently, ccGhc *can only find dyadic operating points that lie on the boundary of the convex hull of the set of all dyadic operating points.* See [19, Section 4.7.4] for a related discussion. All red crosses in Figure 5.4 lie on this boundary, but the green cross does not. We can compensate for the non-optimality of ccGhc by invoking Proposition 5.8, which states that ccGhc is asymptotically capacity achieving for noiseless channels with cost constraints. In Figure 5.4, the dyadic operating points that result from jointly considering two consecutive symbols are displayed. They were calculated by ccGhc$(\boldsymbol{1}^2, \oplus^2\boldsymbol{w}, 2E)$ again for 100 equally spaced values of E. As we can see, ccGhc now outperforms the randomly found operating point of $\boldsymbol{d}_{\text{rand}}$, i.e., there is a dyadic operating point found by ccGhc that lies left and above of the random operating point.

We discussed the possible non-optimality of ccGhc for noiseless channels with costs, but the same occurs when ccGhc is applied to relative entropy and dmcs.

5.4 Discrete memoryless channel

In this section, we consider the important class of dmcs with average cost constraint. The main result of this section is that ccGhc is asymptotically capacity-achieving for this class of dmcs. Recall that a dmc is specified by a matrix $\boldsymbol{H}$ of transition probabilities from n input symbols to m output symbols. An input pmf $\boldsymbol{p}$ relates to its corresponding output pmf $\boldsymbol{r}$ as

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{p}. \tag{5.82}$$

By (2.91), the mutual information between input and output can be written as

$$\mathbb{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j}. \tag{5.83}$$

All input symbols are now of equal duration 1. Each symbol has an associated positive cost and different symbols can have different costs. The input pmf is subject to an average cost constraint E. Capacity is now given by

$$\mathsf{C} = \max_{\text{pmf } \boldsymbol{p}:\boldsymbol{w}^T\boldsymbol{p}\le E} \mathbb{I}(\boldsymbol{p}) \tag{5.84}$$

where the entries of $\boldsymbol{w}$ are the costs of the symbols. To find the best dyadic input pmf, we need to solve the matching problem

$$\begin{aligned} \underset{\text{dyadic } \boldsymbol{d}}{\text{maximize}} \quad & \mathbb{I}(\boldsymbol{d}) \\ \text{subject to} \quad & \boldsymbol{w}^T \boldsymbol{d} \leq E. \end{aligned} \tag{5.85}$$

As for the other variations of the dmc, we do not know an algorithm that directly solves this problem efficiently. To tackle this problem, we use the same approach as we did for the simple dmc in Section 3.4 and the dmc with unequal symbol durations in Section 4.3. First, we drop the restriction to dyadic pmfs and analytically characterize the capacity-achieving pmf $\boldsymbol{p}^*$. Based on this characterization, we derive the penalty that results from using a pmf $\boldsymbol{p}$ different from $\boldsymbol{p}^*$. Then, we upper-bound this penalty and minimize this bound over all dyadic pmfs. Finally, we show that ccGhc finds dyadic input pmfs that are asymptotically capacity-achieving.

5.4.1 Capacity-achieving pmf

Capacity of a cost constrained dmc is given by the optimal value of the following optimization problem.

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{minimize}} \quad & -\mathbb{I}(\boldsymbol{p}) \\ \text{subject to} \quad & \boldsymbol{w}^T \boldsymbol{p} - E \leq 0 \\ & \mathbf{1}^T \boldsymbol{p} - 1 = 0 \end{aligned} \tag{5.86}$$

where the domain of the problem is $\mathbf{R}^n_{\geq 0}$. This is a convex optimization problem and both optimal pmf $\boldsymbol{p}^*$ and optimal value $-\mathsf{C}$ are efficiently found by numerical methods as provided for example by the software package CVX [41]. The goal is to analytically characterize the optimal pmf $\boldsymbol{p}^*$.

Proposition 5.9. *For a dmc with positive symbol costs $\boldsymbol{w}$ and an active cost constraint E, the following conditions are necessary and sufficient for a feasible $\boldsymbol{p}$ to be capacity-achieving.*

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} = \lambda(w_i - E) + \mathsf{C}, \qquad \forall i : p_i > 0 \tag{5.87}$$

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq \lambda(w_i - E) + \mathsf{C}, \qquad \forall i : p_i = 0 \tag{5.88}$$

where $\lambda > 0$.

Proof. By Proposition 2.5 strong duality holds. By Proposition 2.7, the partial derivatives of the objective function $-\mathbb{I}$ are well-defined with the possible exception of taking the value $-\infty$ on the boundary of the domain $\mathbf{R}^n_{\geq 0}$. Thus, Proposition 2.4 applies. The Lagrangian of Problem (5.86) is

$$L(\boldsymbol{p}, \lambda, \nu) = -\mathbb{I}(\boldsymbol{p}) + \lambda(\boldsymbol{w}^T \boldsymbol{p} - E) + \nu(\mathbf{1}^T \boldsymbol{p} - 1). \tag{5.89}$$

By Proposition 2.4, a feasible pmf $\boldsymbol{p}$ is optimal if and only if the KKT conditions are fulfilled, i.e.,

$$\lambda \geq 0 \tag{5.90}$$

$$\lambda(\boldsymbol{w}^T\boldsymbol{p} - E) = 0 \tag{5.91}$$

$$\frac{\partial L(\boldsymbol{p}, \lambda, \nu)}{\partial p_i} = -\frac{\partial \mathbb{I}(\boldsymbol{p})}{\partial p_i} + \lambda w_i + \nu = 0 \qquad \forall i : p_i > 0 \tag{5.92}$$

$$-\frac{\partial \mathbb{I}(\boldsymbol{p})}{\partial p_i} + \lambda w_i + \nu = 0 \qquad \forall i : p_i = 0. \tag{5.93}$$

If these conditions are fulfilled, then according to Proposition 2.7, all partial derivatives of $\mathbb{I}$ in $\boldsymbol{p}$ are well-defined and given by

$$\frac{\partial \mathbb{I}(\boldsymbol{p})}{\partial p_i} = \sum_j h_{ji} \log \frac{h_{ji}}{r_j} - 1 \tag{5.94}$$

Plugging this into the KKT conditions, we get

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq 1 + \lambda w_i + \nu, \qquad \text{with equality if } p_i > 0. \tag{5.95}$$

The unknown ν can be expressed in terms of capacity C. For a capacity-achieving pmf $\boldsymbol{p}^*$, we have

$$\mathsf{C} = \mathbb{I}(\boldsymbol{p}^*) \tag{5.96}$$

$$= \sum_i p_i^* \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} \tag{5.97}$$

$$= \sum_i p_i^* (1 + \lambda w_i + \nu) \tag{5.98}$$

$$= 1 + \nu + \lambda \boldsymbol{w}^T \boldsymbol{p}^* \tag{5.99}$$

$$= 1 + \nu + \lambda E \tag{5.100}$$

where we used (5.95) in the third line and where the last line follows since the constraint is by assumption active. Solving for ν gives

$$\nu = -1 - \lambda E + \mathsf{C}. \tag{5.101}$$

Plugging this into (5.95), we finally get

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq \lambda(w_i - E) + \mathsf{C}, \qquad \text{with equality if } p_i > 0. \tag{5.102}$$

This concludes the proof. □

5.4.2 Using a 'wrong' pmf

We define the *capacity-cost function* $\mathsf{C}(E)$ pointwise by the solution of (5.86), i.e., if $\boldsymbol{p}^*$ is the optimal pmf for the cost constraint E, we define $\mathsf{C}(E) = \mathbb{I}(\boldsymbol{p}^*)$. This allows us to express the mutual information achieved by an arbitrary pmf $\boldsymbol{p}$ in terms of a cost constraint E, the corresponding optimal pmf $\boldsymbol{p}^*$, and the value of the capacity-cost function $\mathsf{C}(E)$.

Proposition 5.10. *Given is a dmc with positive symbol costs $\boldsymbol{w}$, an active cost constraint E, and the corresponding capacity-achieving pmf $\boldsymbol{p}^*$. Denote by $\boldsymbol{p}$ an arbitrary pmf with the only restriction that*

$$p_i = 0, \qquad \text{whenever } p_i^* = 0. \tag{5.103}$$

Then the mutual information achieved by $\boldsymbol{p}$ can be written as

$$\mathbb{I}(\boldsymbol{p}) = \mathsf{C}(E) + \lambda(\boldsymbol{w}^T\boldsymbol{p} - E) - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*) \tag{5.104}$$

where λ is given by the slope of the tangent of the capacity-cost function in $[E, \mathsf{C}(E)]$, i.e.,

$$\lambda = \frac{\partial \mathsf{C}(E)}{\partial E}. \tag{5.105}$$

Proof. Because of our assumption (5.103), we can write the mutual information achieved by $\boldsymbol{p}$ according to Proposition 3.10 as

$$\mathbb{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*). \tag{5.106}$$

We write the first term further as

$$\begin{aligned}
\sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} &= \sum_i (p_i^* + p_i - p_i^*) \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} && (5.107)\\
&= \mathbb{I}(\boldsymbol{p}^*) + \sum_i (p_i - p_i^*) \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} && (5.108)\\
&= \mathsf{C}(E) + \sum_i (p_i - p_i^*)[\lambda(w_i - E) + \mathsf{C}] && (5.109)\\
&= \mathsf{C}(E) + \lambda(\boldsymbol{w}^T\boldsymbol{p} - \boldsymbol{w}^T\boldsymbol{p}^*) && (5.110)\\
&= \mathsf{C}(E) + \lambda(\boldsymbol{w}^T\boldsymbol{p} - E) && (5.111)
\end{aligned}$$

where we used Proposition 5.9 in the third line. Plugging (5.111) into (5.106), we get

$$\mathbb{I}(\boldsymbol{p}) = \mathsf{C}(E) + \lambda(\boldsymbol{w}^T\boldsymbol{p} - E) - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*). \tag{5.112}$$

Since the constraint is active, the value of λ is according to [19, Section 5.6.3] given by

$$\lambda = \frac{\partial \mathbb{I}(E)}{\partial E}. \tag{5.113}$$

This concludes the proof. □

5.4.3 Strictly concave lower bound on capacity-cost function

An important issue for the design of algorithms that find capacity-achieving pmfs of dmcs with average cost constraints is the question if the capacity-cost function is strictly concave in the cost constraint E. While this is for many dmcs of practical interest indeed the case, we do not know if this property holds in general. This motivates us to present a lower bound on the capacity-cost function that is provable strictly concave for any dmc. Denote by E an active cost constraint and by $\boldsymbol{p}^*$ the corresponding capacity-achieving pmf. Denote further by $\boldsymbol{p}$ an arbitrary pmf with the restriction that

$$p_i = 0, \quad \text{whenever } p_i^* = 0. \tag{5.114}$$

The mutual information achieved by $\boldsymbol{p}$ can now be bounded as

$$\begin{aligned} \mathbb{I}(\boldsymbol{p}) &= \mathsf{C}(E) + \lambda(\boldsymbol{w}^T\boldsymbol{p} - E) - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*) && (5.115)\\ &\geq \mathsf{C}(E) + \lambda(\boldsymbol{w}^T\boldsymbol{p} - E) - \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) && (5.116) \end{aligned}$$

where we used Proposition 5.10 in the first line and where we used Proposition 3.12 in the second line. Denote now by $E' \leq E$ another active cost constraint and by $\boldsymbol{p}'$ the optimal pmf of

$$\begin{aligned} \underset{\text{pmf } \boldsymbol{p}}{\text{minimize}} \quad & \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) \\ \text{subject to} \quad & \boldsymbol{w}^T\boldsymbol{p} \leq E'. \end{aligned} \tag{5.117}$$

Denote the optimal value by $\mathsf{D}(E')$. This problem is an instance of the problem solved in Proposition 5.1. Note that according to this proposition, $\boldsymbol{p}'$ fulfills the condition (5.114). Thus, we can set $\boldsymbol{p} = \boldsymbol{p}'$ in (5.116) and get

$$\begin{aligned} \mathsf{C}(E) + \lambda(\boldsymbol{w}^T\boldsymbol{p}' - E) - \mathbb{D}(\boldsymbol{p}'\|\boldsymbol{p}^*) &= \mathsf{C}(E) + \lambda(E' - E) - \mathsf{D}(E') && (5.118)\\ &=: \mathsf{C}_{\text{-}}(E', E). && (5.119) \end{aligned}$$

This function has some interesting properties.

- For $E' \leq E$, it lower bounds the capacity cost-function $\mathsf{C}(E')$, i.e.,

$$\mathsf{C}_{\text{-}}(E', E) \leq \mathsf{C}(E') \tag{5.120}$$

- For $E' = E$, the bound is tight, i.e.,

$$\mathsf{C}_{\text{-}}(E, E) = \mathsf{C}(E). \tag{5.121}$$

- According to Proposition 5.2, $\mathsf{D}(E')$ is strictly convex in E' and consequently, $\mathsf{C}_{\text{-}}(E', E)$ is strictly concave in E'.

The concavity of this bound gives the intuition why ccGhc is capacity-achieving for dmcs with average cost constraints. We will give a formal proof for this property in the next subsection.

5.4.4 Matching

For a dmc with positive symbol costs $\boldsymbol{w}$, an active cost constraint E, and a capacity-achieving pmf $\boldsymbol{p}^*$, the dyadic pmf $\boldsymbol{d} = \textsc{ccGhc}(\boldsymbol{p}^*, \boldsymbol{w}, E)$ yields a good feasible dyadic pmf. This is a consequence of the proposition on asymptotic achievability, which we will state in the next subsection.

5.4.5 Asymptotic achievability

We finally show how ccGhc can be used to generate capacity-achieving dyadic pmfs.

Proposition 5.11. *Given is a dmc with positive symbol costs $\boldsymbol{w}$ and an active cost constraint E. The corresponding capacity-achieving pmf is denoted by $\boldsymbol{p}^*$. Denote the capacity by C and assign $\boldsymbol{d}_k = \textsc{ccGhc}(\boldsymbol{p}^{*k}, \boldsymbol{v}_k = \oplus^k \boldsymbol{w}, kE)$. Then*

$$\frac{\mathbb{I}(\boldsymbol{d}_k)}{k} \overset{k\to\infty}{\longrightarrow} \mathsf{C} \tag{5.122}$$

$$\text{and } \frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k} \leq E, \quad \frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k} \overset{k\to\infty}{\longrightarrow} E. \tag{5.123}$$

Proof. For the mutual information achieved by $\boldsymbol{d}_k$, we have

$$\frac{\mathbb{I}(\boldsymbol{d}_k)}{k} = \mathsf{C}(E) + \lambda\left(\frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k} - E\right) - \frac{\mathbb{D}(\boldsymbol{r}_k \| \boldsymbol{r}^{*k})}{k} \tag{5.124}$$

$$\geq \mathsf{C}(E) + \lambda\left(\frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k} - E\right) - \frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{p}^{*k})}{k} \tag{5.125}$$

where we used Proposition 5.3 in the first line and Proposition 3.12 in the second line. According to Proposition 5.4, we have

$$\frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{p}^{*k})}{k} \to 0 \tag{5.126}$$

$$\text{and } \frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k} \leq E, \quad \frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k} \to E. \tag{5.127}$$

Thus, as k goes to infinity, we have for the lower bound (5.125)

$$\mathsf{C}(E) + \lambda\left(\frac{\boldsymbol{v}_k^T \boldsymbol{d}_k}{k} - E\right) - \frac{\mathbb{D}(\boldsymbol{d}_k \| \boldsymbol{p}^{*k})}{k} \overset{k\to\infty}{\longrightarrow} \mathsf{C}(E) + \lambda(E - E) - 0 \tag{5.128}$$

$$= \mathsf{C}. \tag{5.129}$$

Thus, since the lower bound on $\mathbb{I}(\boldsymbol{d}_k)/k$ converges to capacity, so does $\mathbb{I}(\boldsymbol{d}_k)/k$. This concludes the proof. □

5.5 References

We presented ccGhc and the proof of asymptotic achievability for the first time in [9].

The pmf that maximizes entropy subject to a cost constraint is stated by, e.g., Kschischang and Pasupathy in [52, Section IV], Ungerböck in [73, Section 2] and Fischer in [33, Section 4.1.2].

The problem (5.66) of maximizing entropy over all dyadic pmfs subject to a cost constraint was formulated by Kschischang and Pasupathy in [52, Section VII.B]. They proposed to use the dyadic pmf induced by the Huffman source code of the optimal pmf, see also Fischer [33, Example 4.1]. The same approach was proposed by Ungerböck in [73] and he called it *Huffman shaping*.

For noisy channels with cost constraint, the common approach in literature is to use as a prefix-free matcher the Huffman source code of the input pmf that maximizes entropy [52, 73]. In [10], we formulate a version of Proposition 5.10 for additive noise channels and propose to use the prefix-free matcher obtained by applying GHC to the capacity-achieving pmf of the channel.

6 Noiseless channels with memory

This chapter is about capacity, maximum entropy rate, and capacity-achieving coding for noiseless channels with memory. This generalizes the results we obtained in Section 4.2 for noiseless channels with unequal symbol durations. The chapter is organized as follows. In Section 6.2 we define general noiseless channels. We introduce the two different concepts of general sources and sources. We then show that for any noiseless channel, the maximum entropy rate of general sources is equal to the combinatorial capacity and that the maximum entropy rate of sources is upper-bounded by the combinatorial capacity. This precisely establishes the relation between the combinatorial and probabilistic notion of capacity in the most general setting. In Section 6.3, we consider noiseless channels that are generated by walks on graphs with finitely many states. For finite state channels, we show that there is no loss in terms of capacity when the set of strings is restricted to strings that start and end at the same state. We then impose the restriction that all cycles in the graph form uniquely decodable codes. Under this restriction, we state an explicit formula for the combinatorial capacity. We then give an explicit formula for the pmf of a capacity-achieving source. We show that coding can directly be done by matching a dyadic pmf to the capacity-achieving pmf using the techniques introduced in the previous chapters. Finally, in Section 6.4, we apply our framework for noiseless channels with memory to four examples from the literature.

6.1 Preliminaries

We start by introducing some notations and definitions that we will need in the following.

Regular operations

Denote by $\mathcal{A}$ and $\mathcal{B}$ two sets of strings. According to Sipser [70, Definition 1.23], the *regular operations union*, *concatenation*, and *star* are defined as follows.

$$\text{Union: } \mathcal{A} \cup \mathcal{B} = \{\mathtt{a} \in \mathcal{A} \text{ or } \mathtt{a} \in \mathcal{B}\} \tag{6.1}$$

$$\text{Concatenation: } \mathcal{A}\mathcal{B} = \{\mathtt{ab} | \mathtt{a} \in \mathcal{A} \text{ and } \mathtt{b} \in \mathcal{B}\} \tag{6.2}$$

$$\text{Star: } \mathcal{A}^\star = \{\mathtt{a}_1\mathtt{a}_2 \cdots \mathtt{a}_k | k \geq 0 \text{ and each } \mathtt{a}_i \in \mathcal{A}\}. \tag{6.3}$$

We shortly discuss the implication of concatenation on the respective cardinalities of the involved sets. This is important when calculating the capacity of noiseless channels. For $\mathcal{A} = \{\mathtt{aa}, \mathtt{a}\}$ and $\mathcal{B} = \{\mathtt{b}, \mathtt{ab}\}$, the concatenation of $\mathcal{A}$ and $\mathcal{B}$ is

$$\mathcal{A}\mathcal{B} = \{\mathtt{aab}, \mathtt{aaab}, \mathtt{ab}\}. \tag{6.4}$$

In particular, $|\mathcal{AB}| < |\mathcal{A}||\mathcal{B}|$, where we use $|\mathcal{S}|$ to denote the cardinality of set $\mathcal{S}$. This illustrates that concatenation and Cartesian product are two different set operations. For a set $\mathcal{A}$, we denote by $\mathcal{A}^k$ the concatenation of k copies of $\mathcal{A}$, i.e.,

$$\mathcal{A}^k = \underbrace{\mathcal{A}\mathcal{A}\cdots\mathcal{A}}_{k \text{ times}}. \tag{6.5}$$

For the cardinality, it holds that

$$|\mathcal{A}^k| \leq |\mathcal{A}|^k \tag{6.6}$$

with equality if $\mathcal{A}$ is a *uniquely decodable code*. This property can be checked by the *Sardinas-Patterson test*. This test was original published by Sardinas and Patterson [68]. A working description is given by Cover and Thomas [24, Problem 5.27]. A formal proof of the correctness of the test is for instance given by Salomaa [67].

Limit superior

Denote by x_k a sequence of real numbers. The *limit superior* is defined as

$$\limsup_{k\to\infty} x_k := \lim_{k\to\infty} (\sup_{\ell\geq k} x_\ell) \tag{6.7}$$

If the limit superior is a finite number x', then x' is equivalently characterized by

$$\text{for any } \epsilon > 0 : x_k > x' - \epsilon \text{ infinitely often and } x_k < x' + \epsilon \text{ almost everywhere.} \tag{6.8}$$

In our derivations, we will make use of the latter characterization of the limit superior.

General Dirichlet series

We briefly state two results from [44] that we will need in the following. A *general Dirichlet series* is a series of the form

$$f(s) = \sum_{k=1}^{\infty} a_k e^{-\lambda_k s}, \qquad s \in \mathbf{C} \tag{6.9}$$

where $\mathbf{C}$ denotes the set of complex numbers, the a_k are complex numbers, and where $\{\lambda_k\}_{k\in\mathbf{N}}$ is a sequence of increasing real numbers whose limit is infinity. Denote by $\mathrm{Re}(s)$ the real part of s.

Proposition 6.1. [44, Theorem 3] *The series may converge for all values of s, or for none, or for some only. In the last case, there is a number Q such that the series is convergent for $\mathrm{Re}(s) > Q$ and divergent for $\mathrm{Re}(s) < Q$.*

By *divergent*, we mean non-convergent. The number Q is called the *abscissa of convergence*. Define

$$A(k) = \sum_{\ell=1}^{k} a_\ell. \tag{6.10}$$

Proposition 6.2. [44, Theorem 7] *If the abscissa of convergence of the series is positive, it is given by the formula*

$$Q = \limsup_{k\to\infty} \frac{\log |A(k)|}{\lambda_k}. \tag{6.11}$$

6.2 General noiseless channels

In this section, we establish two fundamental results for the most general setting of noiseless channels. First, we define combinatorial capacity and characterize it analytically. We then show that the maximum entropy rate is equal to or smaller than the combinatorial capacity.

6.2.1 Combinatorial capacity

A *discrete noiseless channel* $(\mathcal{A}, w)$ consists of a countably infinite set $\mathcal{A}$ of strings and a *weight function* $w\colon \mathcal{A} \to \mathbf{R}_{>0}$ that associates with each element $\mathtt{a} \in \mathcal{A}$ a positive length $w(\mathtt{a})$. For memoryless channels, the weight function corresponds to the symbol durations $\boldsymbol{w}$ that we defined in Section 4.2. The length of a string $\mathtt{a} \in \mathcal{A}$ can be any positive real value and is in particular not restricted to the positive integers. The weight function has the following recursive property: if $\mathtt{a} \in \mathcal{A}$ can be written as the concatenation of two elements $\mathtt{b}, \mathtt{c} \in \mathcal{A}$, i.e., if $\mathtt{a} = \mathtt{bc}$, then $w(\mathtt{a}) = w(\mathtt{b}) + w(\mathtt{c})$.

In this chapter, discrete noiseless channel, noiseless channel, and channel are all synonyms. For notational convenience, we will refer to a channel $(\mathcal{A}, w)$ simply by its set $\mathcal{A}$, the corresponding weight function is either generic or clear from the context. Let Ω denote the set of distinct string lengths of elements in $\mathcal{A}$. We order and index the set Ω such that $\Omega = \{\nu_k\}_{k=1}^{\infty}$ with $\nu_1 < \nu_2 < \cdots$. For every $\nu_k \in \Omega$, $N(\nu_k)$ denotes the number of distinct strings of length ν_k in $\mathcal{A}$. We define the *combinatorial capacity* of a channel $\mathcal{A}$ as

$$\mathsf{C} = \limsup_{k\to\infty} \frac{\log\big[\sum_{\ell=1}^{k} N(\nu_\ell)\big]}{\nu_k}. \tag{6.12}$$

This definition subsumes the definitions given in the literature. In particular, consider the case when Ω is *not too dense*, i.e., there exists some constant $L \geq 0$ and some constant $K \geq 0$ such that for any integer $k \geq 0$

$$\max_{\nu_\ell < k} \ell \leq L k^K. \tag{6.13}$$

In other words, the number of distinct string lengths in Ω that are smaller or equal to k increases only polynomial with k. Then our definition (6.12) coincides with the original definition

$$\mathsf{C} = \limsup_{k\to\infty} \frac{\log N(\nu_k)}{\nu_k} \tag{6.14}$$

that was given by Shannon in [69]. This follows from [16, Theorem 1 (ii)] and an alternative proof is given in [15, Appendix B.3].

Here, we do not assume that Ω is not too dense. The only restriction that we make on the channels under consideration is that the combinatorial capacity is well-defined, i.e., that the limit in (6.12) exists and is finite. This implies in particular that $N(\nu)$ is finite for any finite ν. We define the *generating function* of $\mathcal{A}$ as

$$\mathrm{G}_{\mathcal{A}}(s) := \sum_{\mathtt{a}\in\mathcal{A}} e^{-w(\mathtt{a})s}, \qquad s \in \mathbf{C}. \tag{6.15}$$

Note that the series expansion on the right-hand side is a general Dirichlet series. The generating function of $\mathcal{A}$ can be written as

$$\mathrm{G}_{\mathcal{A}}(s) = \sum_{\mathtt{a}\in\mathcal{A}} e^{-w(\mathtt{a})s} \tag{6.16}$$

$$= \sum_{k=1}^{\infty} \sum_{\mathtt{a}\in\mathcal{A}:w(\mathtt{a})=\nu_k} e^{-\nu_k s} \tag{6.17}$$

$$= \sum_{k=1}^{\infty} N(\nu_k) e^{-\nu_k s}. \tag{6.18}$$

The abscissa of convergence of $\mathrm{G}_{\mathcal{A}}$ identifies capacity:

Proposition 6.3. *The combinatorial capacity of a channel $\mathcal{A}$ is equal to the abscissa of convergence Q of its generating function $\mathrm{G}_{\mathcal{A}}$, i.e.,*

$$\mathsf{C} = \limsup_{k\to\infty} \frac{\log\left[\sum_{\ell=1}^{k} N(\nu_\ell)\right]}{\nu_k} = Q. \tag{6.19}$$

Proof. Since $N(\nu_\ell)$ is a non-negative integer,

$$\sum_{\ell=1}^{k} N(\nu_\ell) = \left|\sum_{\ell=1}^{k} N(\nu_\ell)\right|. \tag{6.20}$$

With

$$|A(k)| = \sum_{\ell=1}^{k} N(\nu_\ell) \tag{6.21}$$

he statement of the proposition now follows directly from Proposition 6.2. □

6.2.2 Maximum entropy rate

For the general setting of noiseless channels as introduced in the previous subsection, we now want to relate the combinatorial notion of capacity to a probabilistic notion of capacity. We do this by introducing two distinct concepts, general sources and sources. For both types, we define entropy rate. We show for general sources that the maximum entropy rate is equal to combinatorial capacity and we show for sources that the maximum entropy rate is upper-bounded by the combinatorial capacity.

Entropy rate of general sources

The objective is to relate the combinatorial characterization of channels as given by the combinatorial capacity (6.12) to a probabilistic measure in terms of the maximum entropy per average length. Ultimately, we have a process in mind that generates at each time instant a substring, which is then appended to the string that has been generated so far, such that at each time instant, the generated string is element of $\mathcal{A}$. In magnetic recording, such a process would generate a substring, write it to the tape, generate another substring, write it to the tape, and so forth, without ever rewinding the tape. The difficulty of analyzing such a process is the following: fix two time instants k and k', $k < k'$. The probability that the process writes a specific string to the tape until time instant k' depends in general on the probabilities of the strings that it can write until time instant k. This dependency can become arbitrarily complicated depending on the composition of the elements of $\mathcal{A}$. Because of these interdependencies, it is difficult to bound the entropy rate of such a process. We solve these interdependencies by decoupling the time instants k and k': each time the recording system wants to write to the tape, it first rewinds the tape completely and then overwrites everything that has been written before. We call such a system a *general source*. In accordance with the definition by Han [42, Remark 1.3.2], a general source is a sequence of random variables

$$X = \{X_k\}_{k=1}^{\infty} \tag{6.22}$$

where X_k takes values in a countable (possibly infinite) set $\mathcal{X}_k$. Each random variable X_k is distributed according to a pmf p_{X_k}. In particular, the X_k are stochastically independent and for each index k, p_{X_k} can be chosen independent from the pmfs chosen for other indices. Similar to [42, Equation (1.7.1)], we define the *entropy rate of a general source* X by

$$\bar{\mathbb{H}}(X) = \limsup_{k\to\infty} \frac{\mathbb{H}(X_k)}{\mathbb{E}[w(X_k)]} \tag{6.23}$$

where $\mathbb{E}[w(X_k)]$ is the expected length of X_k according to p_{X_k}. Clearly, for a sequence X to be a general source of a channel $\mathcal{A}$, the X_k need to generate elements from $\mathcal{A}$, i.e., $\mathcal{X}_k \subseteq \mathcal{A}$. However, this is not enough to guarantee that (6.23) is a meaningful measure for the entropy rate of X. This can best be seen in an example.

Example 1. Let $\mathcal{A}$ be the set of all binary strings and assume $w(\mathtt{1}) = w(\mathtt{0}) = 1$. Because of the recursive property of w, this defines the length of each string in $\mathcal{A}$, e.g., $w(\mathtt{01}) = w(\mathtt{1}) + w(\mathtt{0}) = 2$. Define $\mathcal{X}_k = \{\mathtt{a} \in \mathcal{A} \mid w(\mathtt{a}) \leq k\}$, i.e., X_k can generate any string of length smaller or equal to k. Then, the maximum entropy per average length of X_k is according to (4.55) given by the solution of

$$\sum_{\mathtt{a}\in\mathcal{X}_k} e^{-w(\mathtt{a})s} = 1. \tag{6.24}$$

Assume X_k is distributed according to the optimal pmf (4.57) for each k. Then, since $\mathcal{X}_k \to \mathcal{A}$ for $k \to \infty$, the entropy rate of $X = \{X_k\}_{k=1}^{\infty}$ is according to the definition

(6.23) given by the solution of

$$\sum_{\mathtt{a}\in\mathcal{A}} e^{-w(\mathtt{a})s} = 1. \tag{6.25}$$

For $\mathcal{A}$, the set of distinct lengths is $\Omega = \{k\}_{k=1}^{\infty}$ and the number of distinct strings of length k is $N(k) = 2^k$. Thus

$$\begin{aligned}
\sum_{\mathtt{a}\in\mathcal{A}} e^{-w(\mathtt{a})s} &= \sum_{k=1}^{\infty} 2^k e^{-ks} && (6.26)\\
&= 2e^{-s}\sum_{k=0}^{\infty}(2e^{-s})^k && (6.27)\\
&= \frac{2e^{-s}}{1-2e^{-s}}. && (6.28)
\end{aligned}$$

Setting the last line equal to one, we get

$$\begin{aligned}
\frac{2e^{-s}}{1-2e^{-s}} &= 1 && (6.29)\\
\Leftrightarrow 2e^{-s} &= 1-2e^{-s} && (6.30)\\
\Leftrightarrow s &= \log 4. && (6.31)
\end{aligned}$$

Thus, the entropy rate of the general source X is two bits per bit length, but a binary process can at most transmit one bit per bit length. The reason for this is that the elements of $\mathcal{A}$ are generated ambiguously, in our example, $\mathcal{X}_1 \subset \mathcal{X}_2 \subset \cdots$, e.g., the string $\mathtt{0}$ can be generated by all X_k, $k = 1, 2, \ldots$ and thus contributes multiple times when calculating the entropy rate (6.23).

To guarantee that (6.23) correctly measures the entropy rate, we define $X = \{X_k\}_{k=1}^{\infty}$ is a *general source of* $\mathcal{A}$ if and only if

(i) For each k: $\mathcal{X}_k \subseteq \mathcal{A}$

(ii) $\mathcal{X}_\ell \cap \mathcal{X}_k = \emptyset$ if $\ell \neq k$.

That is, a general source can only generate elements of $\mathcal{A}$ and in addition, each element of $\mathcal{A}$ can be element of $\mathcal{X}_k$ for at most one value of k. With this definition, we have the following result.

Proposition 6.4. *The maximum entropy rate of a general source of a channel $\mathcal{A}$ is equal to its combinatorial capacity, i.e.,*

$$\max_X \bar{\mathbb{H}}(X) = \mathsf{C}. \tag{6.32}$$

Proof. The proof has two parts. First, we show achievability, i.e., we define a particular general source that has an entropy rate larger than or equal to the combinatorial capacity. We then show the converse, i.e., that no general source can have an entropy rate larger than the combinatorial capacity.

Achievability: For a general source $X = \{X_k\}_{k=1}^{\infty}$, define the sets $\mathcal{X}_k$ as

$$\mathcal{X}_k = \{\mathsf{a} \in \mathcal{A} \mid k \le w(\mathsf{a}) < k+1\} \tag{6.33}$$

and define the pmf of X_k as

$$p_{X_k}(\mathsf{a}) = \frac{1}{|\mathcal{X}_k|}, \quad \forall \mathsf{a} \in \mathcal{X}_k \tag{6.34}$$

i.e., for each k, X_k is uniformly distributed over $\mathcal{X}_k$. Note that $|\mathcal{X}_k| < \infty$ follows automatically from C being well-defined, see (6.12). We now show that $\bar{\mathbb{H}}(X) \ge \mathsf{C}$. Clearly, $\mathbb{E}[w(X_k)] < k+1$ and

$$\mathbb{H}(X_k) = \log |\mathcal{X}_k| = \log \sum_{k \le \nu < k+1} N(\nu). \tag{6.35}$$

Thus, we have for the entropy rate

$$\bar{\mathbb{H}}(X) \ge \limsup_{k\to\infty} \frac{\log \sum\limits_{k\le\nu<k+1} N(\nu)}{k+1} \tag{6.36}$$

$$= \limsup_{k\to\infty} \frac{\log \sum_{\ell\le k} N(\nu_\ell)}{\nu_k} \tag{6.37}$$

$$= \mathsf{C} \tag{6.38}$$

where the second line can be shown by combinatorial arguments following the proof of [12, Theorem 5].

Converse: We now show that the entropy rate of any general source is smaller or equal to the combinatorial capacity, i.e.,

$$\mathsf{R} := \max_{\{\mathcal{X}_k, p_{X_k}\}_{k=1}^{\infty}} \bar{\mathbb{H}}(X) \le \mathsf{C} \tag{6.39}$$

where the maximization is taken over all partitions $\{\mathcal{X}_k\}_{k=1}^{\infty}$ of $\mathcal{A}$ and for each partition and each k, over all pmfs p_{X_k}. To show this, we pick an arbitrary partition and show that the maximum entropy rate for this partition is upper-bounded by the combinatorial capacity.

Denote by $\{\mathcal{X}_k\}_{k=1}^{\infty}$ an arbitrary partition of an arbitrary subset of $\mathcal{A}$ and consider the general source $X = \{X_k\}_{k=1}^{\infty}$ where X_k takes values in $\mathcal{X}_k$. The maximization of the entropy rate can be written as

$$R_X := \max_{\{p_{X_\ell}\}_{\ell=1}^{\infty}} \bar{\mathbb{H}}(X) \tag{6.40}$$

$$= \max_{\{p_{X_\ell}\}_{\ell=1}^{\infty}} \limsup_{k\to\infty} \frac{\mathbb{H}(X_k)}{\mathbb{E}[w(X_k)]} \tag{6.41}$$

$$= \limsup_{k\to\infty} \max_{p_{X_k}} \frac{\mathbb{H}(X_k)}{\mathbb{E}[w(X_k)]} \tag{6.42}$$

where equality in the last line follows because the X_k are by definition stochastically independent. Thus, the maximization can be done elementwise. For each k, the maximum entropy per average length of X_k

$$R_{X_k} := \max_{p_{X_k}} \frac{\mathbb{H}(X_k)}{\mathbb{E}[w(X_k)]} \tag{6.43}$$

is according to (4.55) given by the solution of

$$\sum_{\mathtt{a}\in\mathcal{X}_k} e^{-w(\mathtt{a})s} = 1. \tag{6.44}$$

With each X_k being distributed according to the optimal pmf (4.57), the entropy rate of X is given by

$$R_X = \limsup_{k\to\infty} R_{X_k} \tag{6.45}$$

which implies in particular that for any $\epsilon > 0$

$$R_{X_k} \geq R_X - \epsilon \quad \text{infinitely often with respect to } k. \tag{6.46}$$

We now use this to show that the generating function $\mathrm{G}_{\mathcal{A}}$ diverges for $\mathrm{Re}(s) < R_X$. Because of (6.46),

$$\sum_{\mathtt{a}\in\mathcal{X}_k} e^{-w(\mathtt{a})(R_X-\epsilon)} \geq \sum_{\mathtt{a}\in\mathcal{X}_k} e^{-w(x)R_{X_k}} = 1 \quad \text{infinitely often with respect to } k. \tag{6.47}$$

Because of $\bigcup_{k=1}^{\infty} \mathcal{X}_k \subseteq \mathcal{A}$, the generating function is for any ℓ bounded by

$$\mathrm{G}_{\mathcal{A}}(s) = \sum_{\mathtt{a}\in\mathcal{A}} e^{-w(\mathtt{a})s} \geq \sum_{k=1}^{\ell} \sum_{\mathtt{a}\in\mathcal{X}_k} e^{-w(\mathtt{a})s}. \tag{6.48}$$

Because of (6.47), it holds that

$$\sum_{k=1}^{\ell} \sum_{\mathtt{a}\in\mathcal{X}_k} e^{-w(\mathtt{a})(R_X-\epsilon)} \overset{\ell\to\infty}{\longrightarrow} \infty \tag{6.49}$$

and we conclude that for any $\epsilon > 0$, $\mathrm{G}_{\mathcal{A}}$ diverges in $s = R_X - \epsilon$. Thus, by Proposition 6.1, $\mathrm{G}_{\mathcal{A}}$ diverges for all $s \in \mathbf{C}$ with $\mathrm{Re}(s) < R_X$. Since G_A converges for $\mathrm{Re}(s) > Q$, it must hold that $R_X \leq Q$ for any general source X. Since $\mathsf{R} = R_X$ for some general source X, $\mathsf{R} \leq Q$. Consequently, by Proposition 6.3, $\mathsf{R} \leq \mathsf{C}$. This concludes the proof. □

Entropy rate of sources

In most cases, general sources are a poor model for transmission over a channel. More practical is a source that generates at each time instant a substring, which is then appended to the string that has been generated so far. At each time instant, the complete string has to be element of $\mathcal{A}$. The notion of a source differs fundamentally from what we defined as a general source: a general source generates a complete new string at each time instant.

The formal definition is as follows. Denote by cat the *operation of concatenation*: $\mathrm{cat}(\mathtt{a},\mathtt{b}) = \mathtt{ab}$. The random process $\{Y_k\}_{k=1}^{\infty}$, $Y_k \in \mathcal{Y}$ is a *source of the channel* $\mathcal{A}$ if the sequence of random variables $X_k = \mathrm{cat}(Y_1,\ldots,Y_k)$ with the supports

$$\mathcal{X}_k = \Big\{\mathrm{cat}(y_1,\ldots,y_k) \mid (y_1,\ldots,y_k) \in \mathcal{Y}^k,\ p_Y(y_1,\ldots,y_k) > 0\Big\} \tag{6.50}$$

is a general source of $\mathcal{A}$, i.e., for each k, $\mathcal{X}_k \subseteq \mathcal{A}$ and $\mathcal{X}_k \cap \mathcal{X}_\ell = \emptyset$ if $k \neq \ell$. The sequence X is called the *general source induced by the source* Y. Following [79, 49], we define the *entropy rate of a source* Y as

$$\bar{\mathbb{H}}(Y) = \limsup_{k\to\infty} \frac{\mathbb{H}(Y_1,\ldots,Y_k)}{\mathbb{E}[w(Y_1,\ldots,Y_k)]}. \tag{6.51}$$

We next show that the entropy rate of Y is upper-bounded by the entropy rate of the induced general source X. The next example gives the intuition behind the key step in the proof.

Example 2. Consider a channel with the set of strings $\mathcal{A}$ and assume

$$\{\mathtt{00},\mathtt{11},\mathtt{0000},\mathtt{1101},\mathtt{1110},\mathtt{1111}\} \subset \mathcal{A}. \tag{6.52}$$

Assume for the induced general source

$$\mathcal{X}_1 = \{\mathtt{00},\mathtt{11}\} \tag{6.53}$$

$$\mathcal{X}_2 = \{\mathtt{0000},\mathtt{1101},\mathtt{1110},\mathtt{1111}\}. \tag{6.54}$$

Thus, Y_1 takes values in $\{\mathtt{00},\mathtt{11}\}$ and Y_2 takes values in $\{\mathtt{00},\mathtt{01},\mathtt{10},\mathtt{11}\}$. A possible joint pmf of p_Y is

$$p_{(Y_1,Y_2)}(\mathtt{00},\mathtt{00}) = p_{(Y_1,Y_2)}(\mathtt{11},\mathtt{01}) = p_{(Y_1,Y_2)}(\mathtt{11},\mathtt{10}) = p_{(Y_1,Y_2)}(\mathtt{11},\mathtt{11}) = \frac{1}{4} \tag{6.55}$$

which implies the marginal pmf

$$p_{Y_1}(\mathtt{00}) = \frac{1}{4}, \qquad p_{Y_1}(\mathtt{11}) = \frac{3}{4}. \tag{6.56}$$

Thus,

$$\frac{\mathbb{H}(Y_1)}{\mathbb{E}[w(Y_1)]} = \frac{-\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4}}{2} \approx 0.2812, \qquad \frac{\mathbb{H}(Y_1,Y_2)}{\mathbb{E}[w(Y_1,Y_2)]} \approx 0.3466. \tag{6.57}$$

For the general source X, the same values can be achieved by assigning $p_{X_1} = p_{Y_1}$ and $p_{X_2} = p_{(Y_1,Y_2)}$. However, X can do better, since p_{X_1} and p_{X_2} can be chosen independently. By choosing $p_{X_1}(\mathtt{00}) = p_{X_1}(\mathtt{11}) = \frac{1}{2}$ and $p_{X_2} = p_{(Y_1,Y_2)}$, the entropies per average length are

$$\frac{\mathbb{H}(X_1)}{\mathbb{E}[w(X_1)]} \approx 0.3466 > 0.2812, \qquad \frac{\mathbb{H}(X_2)}{\mathbb{E}[w(X_2)]} \approx 0.3466. \tag{6.58}$$

The same holds for any joint pmf p_Y of Y. Thus, the intuition is: in terms of entropy per average length, X_k can always do as good as $(Y_1, \dots, Y_k)$ by using $p_{X_k} = p_{(Y_1,\dots,Y_k)}$, but in many cases, it can do better.

Proposition 6.5. *Denote by Y a source of a channel $\mathcal{A}$. Then the maximum entropy rate is upper-bounded by the combinatorial capacity, i.e.,*

$$\bar{\mathbb{H}}(Y) \leq \mathsf{C}. \tag{6.59}$$

Proof. Denote by $\{X_k\}_{k=1}^{\infty}$ the general source induced by the source Y. The entropy rate of Y can now be bounded as

$$\max_{p_Y} \bar{\mathbb{H}}(Y) = \max_{p_Y} \limsup_{k\to\infty} \frac{\mathbb{H}(Y_1, \dots, Y_k)}{\mathbb{E}[w(Y_1, \dots, Y_k)]} \tag{6.60}$$

$$\leq \max_{\{p_{X_\ell}\}_{\ell=1}^{\infty}} \limsup_{k\to\infty} \frac{\mathbb{H}(X_k)}{\mathbb{E}[w(X_k)]} \tag{6.61}$$

$$\leq \mathsf{C} \tag{6.62}$$

where the inequality in (6.61) holds because for a general source, the maximization can be done in each step k independent from the other steps, see also Example 2. The inequality in the last line follows from Proposition 6.4. □

6.3 Finite state channels

In the last section, we have shown that for general noiseless channels, the combinatorial capacity is given by the abscissa of convergence of the generating function, the maximum entropy rate of general sources is equal to the combinatorial capacity, and the maximum entropy rate of sources is upper-bounded by the combinatorial capacity. However, the derivations were not constructive, i.e., we didn't actually calculate the concrete value of capacity or specify a capacity-achieving source. This is the topic of this section.

The idea is that, to be able to calculate the capacity of a channel $\mathcal{A}$, the set $\mathcal{A}$ should have some repetitive structure. We therefore consider channels where the set $\mathcal{A}$ can be generated by a *directed graph* D with finitely many states. The transitions between the states are labelled by substrings that are elements of $\mathcal{A}$. We allow infinitely many transitions between two states.

6.3.1 Combinatorial capacity

We start by characterizing the combinatorial capacity of finite state channels.

Strongly connected graph

A directed graph consists of strongly connected components. A *strongly connected component* of a directed graph is a set of states $\mathcal{S}$ such that for each ordered pair (i, j), $i, j \in \mathcal{S}$, there is a path from state i to state j. Denote by $\mathcal{A}$ a finite state channel that is generated by a directed graph D. Denote by $\mathsf{C}(\mathcal{S})$ the capacity of the set that is generated by walks between the states in the strongly connected component $\mathcal{S}$ of D. In the spirit of [12, Section 2.2], it can be shown that

$$\mathsf{C}(\mathcal{A}) = \max_{\mathcal{S}} \mathsf{C}(\mathcal{S}) \tag{6.63}$$

where the maximization is over all strongly connected components $\mathcal{S}$ of the graph D that generates $\mathcal{A}$. We therefore assume from now on without loss of generality that $\mathcal{A}$ is generated by a *strongly connected graph* D, i.e., a graph that consists of one strongly connected component. Denote by $\mathcal{A}_{i,j}$ the set of strings that start at state i and end at state j. We now show that strong connectivity implies that the capacity of $\mathcal{A}_{i,j}$ is equal to the capacity of $\mathcal{A}$. The existence of capacity-achieving memoryless codes is founded on this result.

Proposition 6.6. *Assume a channel $\mathcal{A}$ is generated by a strongly connected graph with n states. Let C denote the capacity of $\mathcal{A}$ and let $\mathsf{C}_{i,j}$ denote the capacity of $\mathcal{A}_{i,j}$. Then*

$$\mathsf{C} = \mathsf{C}_{i,j}, \qquad 1 \le i, j \le n. \tag{6.64}$$

Proof. Denote by G and $\mathrm{G}_{i,j}$ the generating functions of $\mathcal{A}$ and $\mathcal{A}_{i,j}$, respectively. The generating function G is bounded by

$$\begin{aligned} \max_{i,j} \mathrm{G}_{i,j}(s) &\le \mathrm{G}(s) && (6.65)\\ &\le \sum_{i,j} \mathrm{G}_{i,j}(s) && (6.66)\\ &\le n^2 \max_{i,j} \mathrm{G}_{i,j}(s). && (6.67) \end{aligned}$$

Since the upper bound differs from the lower bound only by a factor, we conclude that G and $\max_{i,j} \mathrm{G}_{i,j}$ have the same abscissa of convergence and therefore, by Proposition 6.3,

$$\mathsf{C} = \max_{i,j} \mathsf{C}_{i,j}. \tag{6.68}$$

From now on, denote by a, b two states for which the maximum is achieved. Denote by $N_{i,j}(\nu)$ the number of paths from state i to state j of length ν. Since the channel is strongly connected, there exist $\sigma, \tau \in \Omega$ with $\sigma, \tau < \infty$ such that $N_{i,a}(\sigma)$ and $N_{b,j}(\tau)$ are positive, i.e., there exists at least one path from i to a of length σ and there exists at least one path from b to j of length τ. Now, the following bounds on the generating function $\mathrm{G}_{i,j}$ hold:

$$\begin{aligned} \mathrm{G}_{a,b}(s) &\ge \mathrm{G}_{i,j}(s) && (6.69)\\ &\ge N_{i,a}(\sigma) e^{-\sigma s} \mathrm{G}_{a,b}(s) N_{b,j}(\tau) e^{-\tau s} && (6.70)\\ &\ge \mathrm{G}_{a,b}(s) e^{-(\sigma+\tau)s}. && (6.71) \end{aligned}$$

Upper and lower bound only differ by a factor, therefore, $\mathsf{G}_{a,b}$ and $\mathsf{G}_{i,j}$ have the same abscissa of convergence, for all pairs (i,j), $1 \leq i,j \leq n$. Thus, by Proposition 6.3,

$$\mathsf{C}_{i,j} = \mathsf{C}_{a,b}, \qquad 1 \leq i,j \leq n \tag{6.72}$$

which concludes the proof. □

Uniquely decodable graph

For a channel $\mathcal{A}$, we now identify the convergence behavior of the generating function $\mathsf{G}_{\mathcal{A}}$ with the convergence behavior of a matrix series. The *spectral radius* $\rho(A)$ of a matrix A is defined as

$$\rho(A) = \max_i |\lambda_i| \tag{6.73}$$

where the λ_i are the eigenvalues of A. The spectral radius will be useful, since it determines if a matrix series converges or not.

Proposition 6.7. *Denote by A a square matrix. Then the series*

$$\sum_{k=0}^{\infty} A^k \tag{6.74}$$

converges if $\rho(A) < 1$ and it diverges if $\rho(A) > 1$.

Proof. Convergence. Consider the power series

$$\sum_{k=0}^{\infty} z^k, \quad z \in \mathbf{C}. \tag{6.75}$$

Its radius of convergence is 1. Thus, by [20, Satz 3.64], the matrix series in (6.74) converges if the absolute value of each eigenvalue of A is smaller than 1, but this is equivalent to requiring $\rho(A) < 1$.

Divergence. If $\rho(A) > 1$, then for some eigenvalue λ of A, we have $|\lambda| > 1$. According to [20, Satz 3.51], we can write A in its *Jordan normal form*, i.e., there exists an invertible matrix T such that

$$A = T^{-1}JT \tag{6.76}$$

where J is an upper-triangular matrix with the eigenvalues of A on its diagonal. The series (6.74) can now be written as

$$\sum_{k=0}^{\infty} A^k = \sum_{k=0}^{\infty} (T^{-1}JT)^k \tag{6.77}$$

$$= T^{-1}\Big(\sum_{k=0}^{\infty} J^k\Big)T \tag{6.78}$$

and we conclude that the series (6.74) diverges if $\sum_k J^k$ diverges. Since J is a triangular matrix, J^k has λ^k on its diagonal. Thus, since $|\lambda| > 1$, this diagonal entry diverges for $k \to \infty$. Consequently, the matrix series diverges. This concludes the proof. □

The objective is now to use this proposition for calculating the capacity of channels. Define the *generating matrix* of $\mathcal{A}$ as

$$\mathbf{G}_{\mathcal{A}}(s) = \begin{bmatrix} \mathrm{G}_{1,1}(s) & \cdots & \mathrm{G}_{1,n} \\ \vdots & \ddots & \vdots \\ \mathrm{G}_{n,1}(s) & \cdots & \mathrm{G}_{n,n} \end{bmatrix} \tag{6.79}$$

where the $\mathrm{G}_{i,j}$ are, as in the proof of Proposition 6.6, the generating functions of strings that start at state i and end at state j. Define by $\mathrm{T}_{i,j}(s)$ the *transition generating function*, i.e., the generating function of the set of transitions that go directly from state i to state j. Define further the *transition generating matrix*

$$\mathbf{T}_{\mathcal{A}}(s) = \begin{bmatrix} \mathrm{T}_{1,1}(s) & \cdots & \mathrm{T}_{1,n} \\ \vdots & \ddots & \vdots \\ \mathrm{T}_{n,1}(s) & \cdots & \mathrm{T}_{n,n} \end{bmatrix} \tag{6.80}$$

We want to express $\mathbf{G}_{\mathcal{A}}$ in terms of $\mathbf{T}_{\mathcal{A}}$. The condition for being able to do so is unique decodability. We call a graph D *uniquely decodable* if each set of cycles $\mathcal{A}_{i,i}$, $i = 1, \ldots, n$ in the graph is uniquely decodable. It is enough to check this property for cycle-free cycles. If D is uniquely decodable, then

$$\mathbf{G}_{\mathcal{A}}(s) = \sum_{k=1}^{\infty} [\mathbf{T}_{\mathcal{A}}(s)]^k. \tag{6.81}$$

Thus, the following holds.

Proposition 6.8. *If a uniquely decodable graph generates $\mathcal{A}$, then capacity C is given by the solution of the equation*

$$\rho[\mathbf{T}_{\mathcal{A}}(s)] = 1. \tag{6.82}$$

Proof. The generating function $\mathrm{G}_{\mathcal{A}}$ converges (diverges) if $\mathbf{G}_{\mathcal{A}}$ converges (diverges). Since the graph is uniquely decodable, $\mathbf{G}_{\mathcal{A}}$ converges (diverges) if the matrix series

$$\sum_{k=1}^{\infty} [\mathbf{T}_{\mathcal{A}}(s)]^k \tag{6.83}$$

converges (diverges). By Proposition 6.7, the series converges (diverges) if the spectral radius of $\mathbf{T}_{\mathcal{A}}(s)$ is smaller (greater) than 1. Thus, the abscissa of convergence Q of $\mathrm{G}_{\mathcal{A}}$ is given by the solution of

$$\rho[\mathbf{T}_{\mathcal{A}}(s)] = 1. \tag{6.84}$$

By Proposition 6.3, $\mathsf{C} = Q$ and the proposition follows. □

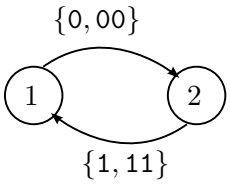


Figure 6.1: An example of a channel defined by a uniquely decodable graph.

Thus, if a channel $\mathcal{A}$ is generated by a strongly connected and uniquely decodable graph, capacity is given by the solution of (6.82). The solution can efficiently be found as follows. By [49, Page 7], $\rho[\mathbf{T}_{\mathcal{A}}(s)]$ is strictly decreasing and convex in s. Thus, assuming that an efficient method is available to calculate the spectral radius of a fixed matrix, the solution can efficiently be found via the bisection method. Algorithm 6 summarizes this method.

Algorithm 6.

$u > \mathsf{C} > \ell$
$\epsilon > 0$
repeat
1. $s = \frac{\ell+u}{2}$
2. $q = \rho[\mathbf{T}_{\mathcal{A}}(s)]$

if $q > 1$, $\ell = s$; **else** $u = s$
until $|u - \ell| < \epsilon$
$\mathsf{C} = \ell$

The following example illustrates this method.

Example 3. Consider the graph in Figure 6.1. The weights of the symbols 0 and 1 are both equal to 1, i.e., $w(0) = w(1) = 1$. The transition generating matrix is given by

$$\mathbf{T}(s) = \begin{pmatrix} 0 & e^{-s} + e^{-2s} \\ e^{-s} + e^{-2s} & 0 \end{pmatrix}. \tag{6.85}$$

We first have to check if the graph is uniquely decodable: the cycles starting and ending at state 1 are given by

$$\mathcal{M} = \{01,\ 011,\ 001,\ 0011\}. \tag{6.86}$$

This set is neither prefix- nor suffix-free. We therefore apply the Sardinas-Patterson test [68] as described in [24, Problem 5.27]. The set of dangling suffixes is $\{1\}$, and $1 \notin \mathcal{M}$. Thus $\mathcal{M}$ is uniquely decodable. The same can be shown for the cycles starting and ending at state 2, and consequently, the graph is uniquely decodable. The capacity of the channel is now given by solving

$$\rho[\mathbf{T}(s)] = \rho\left[\begin{pmatrix} 0 & e^{-s} + e^{-2s} \\ e^{-s} + e^{-2s} & 0 \end{pmatrix}\right] = 1. \tag{6.87}$$

We apply Algorithm 6 and find $\mathsf{C} \approx 0.4812$.

6.3.2 Maximum entropy rate

We now relate the maximum entropy rate of sources to the combinatorial capacity of finite state channels that are generated by uniquely decodable graphs.

Proposition 6.9. *For a channel $\mathcal{A}$ generated by a uniquely decodable graph D, the maximum entropy rate of a source is equal to the combinatorial capacity.*

Proof. Choose some state i, denote by $\mathcal{M}_i$ the set of strings that start at state i, end at state i, but do not pass through state i in-between. Since D is uniquely decodable by assumption,

$$\mathrm{G}_{i,i}(s) = \sum_{k=1}^{\infty} [\mathrm{G}_{\mathcal{M}_i}(s)]^k \tag{6.88}$$

$$= \mathrm{G}_{\mathcal{M}_i^+}(s) \tag{6.89}$$

where $\mathcal{M}_i^+$ denotes the *plus operation* defined as $\mathcal{M}_i^+ = \mathcal{M}_i \mathcal{M}_i^\star$. By Proposition 6.6, $\mathcal{A}$ and $\mathcal{A}_{i,i}$ have the same capacity and consequently, the channel $\mathcal{M}_i^+$ has the same capacity as $\mathcal{A}$. Since $\mathcal{M}_i$ is uniquely decodable by assumption, a source Y with entropy rate equal to C is by (4.57) given by a sequence of random variables $\{Y_k\}_{k=1}^{\infty}$ that are iid according to

$$\text{for each } \mathtt{a} \in \mathcal{M}_i : p_Y(\mathtt{a}) = e^{-w(\mathtt{a})\mathsf{C}}. \tag{6.90}$$

By Proposition 6.5, the entropy rate cannot be larger than C and the proposition follows. □

In the proof of Proposition 6.9, we defined $\mathcal{M}_i$ as the set of strings that start at state i, end at state i and do not pass through state i in-between. If the graph D of $\mathcal{A}$ is uniquely decodable, then the capacity of $\mathcal{M}_i^+$ is equal to the capacity of $\mathcal{A}$. Since $\mathcal{M}_i^+$ can be generated by a graph with only one state, we call it a *memoryless representation of* $\mathcal{A}$. A memoryless representation is useful in two ways: first it allows for an alternative calculation of capacity, and second, it allows for the definition of a capacity-achieving source. We illustrate this in the following example.

Example 4. Consider again the graph in Figure 6.1. Recall that the cycles starting and ending at state 1 are given by

$$\mathcal{M} = \{\mathtt{01}, \mathtt{011}, \mathtt{001}, \mathtt{0011}\}. \tag{6.91}$$

By Proposition 6.6, the capacity of the channel is given by the abscissa of convergence of the generating function $\mathrm{G}_{1,1}$, which is given by

$$\mathrm{G}_{1,1}(s) = \sum_{k=1}^{\infty} [\mathrm{G}_{\mathcal{M}}(s)]^k. \tag{6.92}$$

By Proposition 6.7, the abscissa of convergence is given by the solution of the equation

$$\rho[\mathrm{G}_{\mathcal{M}}(s)] = \rho[e^{-2s} + e^{-3s} + e^{-3s}e^{-4s}] \tag{6.93}$$
$$= e^{-2s} + e^{-3s} + e^{-3s}e^{-4s} \tag{6.94}$$
$$= 1. \tag{6.95}$$

We again use Algorithm 6 to find $\mathsf{C} \approx 0.4812$, which is in accordance with the value found in Example 3. A capacity-achieving source $Y = \{Y_k\}_{k=1}^{\infty}$ is directly given: let the Y_k take values in $\mathcal{M}$ and be iid according to

$$p_Y(\mathtt{a}) = e^{-w(\mathtt{a})\mathsf{C}}, \qquad \forall \mathtt{a} \in \mathcal{M}. \tag{6.96}$$

Note that the method proposed in the proof of [49, Theorem 5.1] cannot be used for the channel in Figure 6.1. The reason is that the adjacency matrix

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{6.97}$$

is *periodic*, for instance, for all $k \in \mathbb{N}$, $(A^{2k})_{1,1} = 1$ and $(A^{2k+1})_{1,1} = 0$. Thus, Perron-Frobenius theory cannot be applied and in particular, there is no stationary distribution of the states for a random walk on the graph.

6.3.3 Coding

As we have shown, channels that are generated by a uniquely decodable graph have a memoryless representation $\mathcal{M}$. A source that takes values in $\mathcal{M}$ iid according to a certain pmf is capacity-achieving. In many cases, $\mathcal{M}$ is an infinite set. Denote by $\mathcal{M}_\ell = \{\mathtt{a}_1, \ldots, \mathtt{a}_\ell\}$ the subset of $\mathcal{M}$ with the ℓ elements of shortest length. Then, according to Proposition 6.8, the capacity C_ℓ of $\mathcal{M}_\ell^+$ is given by the solution of

$$\sum_{i=1}^{\ell} e^{-w(\mathtt{a}_i)s} = 1. \tag{6.98}$$

As ℓ increases, C_ℓ approaches capacity, i.e.,

$$\lim_{\ell \to \infty} \mathsf{C}_\ell = \mathsf{C}. \tag{6.99}$$

For a finite ℓ, denote by $\boldsymbol{w}_\ell$ the string lengths of the elements in $\mathcal{M}_\ell$. Then, according to Subsection 4.2.1, the dyadic pmf $\boldsymbol{d}_\ell = \textsc{nGhc}(\mathbf{1}, \boldsymbol{w}_\ell)$ maximizes entropy per average length over all dyadic pmfs. According to Subsection 4.2.4, the capacity C_ℓ can be arbitrarily well approximated by a dyadic pmf in terms of achieved entropy rate by jointly considering blocks of consecutive symbols. We thus have the following result.

Proposition 6.10. *Let $\mathcal{A}$ be a channel with capacity C that is generated by a uniquely decodable graph. Denote by $\mathcal{M}$ a memoryless representation of $\mathcal{A}$. Denote by $\boldsymbol{d}_{\ell,k} = \textsc{nGhc}(\mathbf{1}, \oplus^k \boldsymbol{w}_\ell)$ the optimal dyadic blocklength-k pmf of $\mathcal{M}_\ell$. Then*

$$\lim_{\ell,k \to \infty} \bar{\mathbb{H}}(\boldsymbol{d}_{\ell,k}) = \mathsf{C}. \tag{6.100}$$

	$\ell = 1$	$\ell = 2$	$\ell = 3$	$\ell = 4$
$k = 2$	0.4	0.4	0.4	0.4026
$k = 3$	0.6	0.6	0.6016	0.6024
$k = 4$	0.6667	0.6856	0.69	0.6919

Table 6.1: Capacity is 0.6942. For $k = 4$ and $\ell = 4$, i.e., for a code with 256 codewords, the achieved rate is within -0.3457% of capacity.

We call the coding strategy described in the proposition *variable length memoryless* (VLM) *coding*. For a memoryless representation $\mathcal{M}$, $\text{VLM}(\mathcal{M}_\ell, k)$ denotes the dyadic pmf that maximizes the entropy rate when jointly generating by a prefix-free matcher blocks of length k of the ℓ symbols of shortest length from $\mathcal{M}$.

Example 5. Consider again the graph in Figure 6.1. The memoryless representation is

$$\mathcal{M}^+ = \{\texttt{01}, \texttt{011}, \texttt{001}, \texttt{0011}\}^+. \tag{6.101}$$

Note that already for $\ell = 4$, $\mathcal{M}_\ell = \mathcal{M}$. The entropy rates achieved by $\text{VLM}(\mathcal{M}_\ell, k)$ coding for $\ell = 2, 3, 4$ and $k = 1, 2, 3, 4$ are displayed in Table 6.1. For $\ell = 4$ and $k = 4$, i.e., for a code with 256 codewords, the achieved rate is within -0.3457% of capacity.

6.4 Applications

6.4.1 Capacity of asynchronous channel

The *asynchronous channel* was introduced by Cai and Yeung in [22]. By [22, Theorem 1], the asynchronous channel can be specified by a set $\mathcal{W}$ of *run-lengths* and a set $\mathcal{L}$ of *labels*. The set of run-lengths $\mathcal{W}$ is a non-empty and countable subset of the positive real numbers $\mathbb{R}_{>0}$. The set of labels $\mathcal{L}$ is non-empty and finite. One *run* is the substring of a string during which the label does not change. We refer to the length of a run r by the weight function $w(r)$ and we refer to the label of a run r by the *label function* $\ell(r)$. The set of an asynchronous channel is given by

$$\begin{aligned}\langle \mathcal{W}, \mathcal{L} \rangle := \big\{ \operatorname{cat}(r_1, \ldots, r_k) \;\big|\; & k \in \mathbf{N}, \\ & \text{for } i = 1, \ldots, k \colon w(r_i) \in \mathcal{W} \text{ and } \ell(r_i) \in \mathcal{L}, \\ & \text{for } i = 1, \ldots, k-1 \colon \ell(r_{i+1}) \neq \ell(r_i) \big\}.\end{aligned} \tag{6.102}$$

Note that the graph in Figure 6.1 can be interpreted as an asynchronous channel with

$$\mathcal{W} = \{1, 2\}, \quad \mathcal{L} = \{\texttt{0}, \texttt{1}\}. \tag{6.103}$$

If the cardinality of $\mathcal{L}$ is equal to one, then each string in $\langle \mathcal{W}, \mathcal{L} \rangle$ consists of only one run with its length in $\mathcal{W}$ and with its label equal to the unique label from $\mathcal{L}$. From now on, we therefore assume $|\mathcal{L}| \geq 2$. The asynchronous channel can be represented by a graph of $|\mathcal{L}|$ states as follows. Index the labels in $\mathcal{L}$, i.e., $\mathcal{L} = \{\ell_1, \ldots, \ell_{|\mathcal{L}|}\}$. Then,

between any ordered pair (i, j) of states, there are $|\mathcal{W}|$ transitions, one for each $w \in \mathcal{W}$. Furthermore, each transition to state i has label ℓ_i. This completely specifies the graph. We now choose as memoryless representation $\mathcal{M}$ the set of strings that start at state 1, end at state 1, but do not pass through state 1 in-between. Denote by $r = r_1 \cdots r_k$ a string in $\mathcal{M}$ where each r_i is a run. For the labels, the following holds:

$$\ell(r_1) \in \mathcal{L} \setminus \ell_1 \tag{6.104}$$

$$\text{for } i = 2, \ldots, k-1 : \ell(r_i) \in \mathcal{L} \setminus \{\ell_1, \ell(r_{i-1})\} \tag{6.105}$$

$$\ell(r_k) = \ell_1. \tag{6.106}$$

The weight of each run can be chosen arbitrarily from $\mathcal{W}$. Denote by $\mathrm{G}_{\mathcal{W}}$ the generating function of $\mathcal{W}$. The generating function of the first run of strings in $\mathcal{M}$ is

$$\mathrm{G}_{\mathcal{R}_1}(s) = (|\mathcal{L}| - 1)\, \mathrm{G}_{\mathcal{W}}(s). \tag{6.107}$$

For the intermediate runs, the generating function is

$$\mathrm{G}_{\mathcal{R}}(s) = (|\mathcal{L}| - 2)\, \mathrm{G}_{\mathcal{W}}(s) \tag{6.108}$$

and for the last run, the generating function is

$$\mathrm{G}_{\mathcal{R}_{\text{last}}}(s) = \mathrm{G}_{\mathcal{W}}(s). \tag{6.109}$$

The number of intermediate runs can be any integer between 0 and ∞. Thus, the generating function of $\mathcal{M}$ is

$$\mathrm{G}_{\mathcal{M}}(s) = \mathrm{G}_{\mathcal{R}_1}(s) \sum_{k=0}^{\infty} [\mathrm{G}_{\mathcal{R}}(s)]^k\, \mathrm{G}_{\mathcal{R}_{\text{last}}}(s) \tag{6.110}$$

$$= (|\mathcal{L}| - 1)\, \mathrm{G}_{\mathcal{W}}(s) \sum_{k=0}^{\infty} \Big[(|\mathcal{L}| - 2)\, \mathrm{G}_{\mathcal{W}}(s) \Big]^k\, \mathrm{G}_{\mathcal{W}}(s) \tag{6.111}$$

$$= (|\mathcal{L}| - 1)\, \mathrm{G}_{\mathcal{W}}(s) \frac{\mathrm{G}_{\mathcal{W}}(s)}{1 - (|\mathcal{L}| - 2)\, \mathrm{G}_{\mathcal{W}}(s)}. \tag{6.112}$$

The capacity is given by the solution of $\mathrm{G}_{\mathcal{M}}(s) = 1$. This equation can be simplified as follows.

$$\mathrm{G}_{\mathcal{M}}(s) = (|\mathcal{L}| - 1)\, \mathrm{G}_{\mathcal{W}}(s) \frac{\mathrm{G}_{\mathcal{W}}(s)}{1 - (|\mathcal{L}| - 2)\, \mathrm{G}_{\mathcal{W}}(s)} = 1 \tag{6.113}$$

$$\Leftrightarrow (|\mathcal{L}| - 1)\, \mathrm{G}_{\mathcal{W}}(s)\, \mathrm{G}_{\mathcal{W}}(s) = 1 - (|\mathcal{L}| - 2)\, \mathrm{G}_{\mathcal{W}}(s) \tag{6.114}$$

$$\Leftrightarrow (|\mathcal{L}| - 1)\, \mathrm{G}_{\mathcal{W}}(s)\, \mathrm{G}_{\mathcal{W}}(s) = 1 + \mathrm{G}_{\mathcal{W}}(s) - (|\mathcal{L}| - 1)\, \mathrm{G}_{\mathcal{W}}(s) \tag{6.115}$$

$$\Leftrightarrow (|\mathcal{L}| - 1)\, \mathrm{G}_{\mathcal{W}}(s)[1 + \mathrm{G}_{\mathcal{W}}(s)] = 1 + \mathrm{G}_{\mathcal{W}}(s) \tag{6.116}$$

$$\Leftrightarrow (|\mathcal{L}| - 1)\, \mathrm{G}_{\mathcal{W}}(s) = 1. \tag{6.117}$$

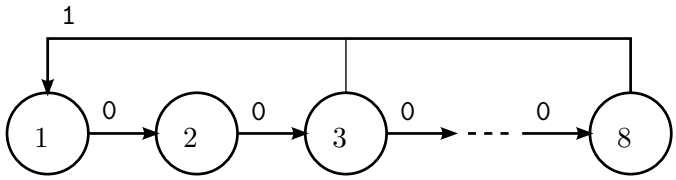


Figure 6.2: Directed graph generating the $(2, 7)$-constrained channel.

VLM code		Franaszek's code [36]	
00	001	11	0100
01	0001	10	1000
10	00001	011	000100
110	000001	010	001000
1110	0000001	000	1001000
1111	00000001	0011	00100100
		0010	00001000

Table 6.2: VLM code and Franaszek's code for the $(2, 7)$ constraint.

The last line coincides with the formula given by Yeung *et al* [77, Theorem 2]. Denote by C the capacity of the channel. Then, the source $Y = \{Y_k\}_{k=1}^{\infty}$ where the Y_k take values in $\mathcal{M}$ and are iid according to

$$p_Y(y) = e^{-w(y)\mathsf{C}}, \qquad \forall y \in \mathcal{M} \tag{6.118}$$

has an entropy rate equal to C. Furthermore, by Proposition 6.5, the entropy rate cannot be larger. These two statements prove [77, Theorem 4], i.e., that the maximum entropy rate is equal to the combinatorial capacity for asynchronous channels. Furthermore, VLM coding is directly applicable to the memoryless representation $\mathcal{M}$ and asymptotically capacity-achieving. Thus, VLM coding provides an alternative to the arithmetic coding approach chosen by Cai *et al* in [21].

6.4.2 Coding for $(2, 7)$ constraint

A (d, k)-constrained channel allows the transmission of binary strings where two consecutive 1s are separated by at least d and at most k 0s. (d, k) constraints can be generated by directed graphs. Figure 6.2 shows a graph that generates $(2, 7)$-constrained binary strings. We choose as a memoryless representation the set of strings that start and end at state one. It is given by

$$\mathcal{M}^+ = \{001, 0001, 00001, 000001, 0000001, 00000001\}^+. \tag{6.119}$$

We apply the VLM code to the whole set $\mathcal{M}$ with blocklength 1. The resulting code is displayed in Table 6.2. For comparison, Franaszek's $(2, 7)$-constraint code [36] is displayed. The capacity of the $(2, 7)$-constraint is $\mathsf{C} = 0.5174$. The VLM code achieves a rate of 0.50667 with 6 codewords and is within -2.11% of capacity, while Franaszek's code achieves a rate of 0.5 with 7 codewords and is within -3.47% of capacity.

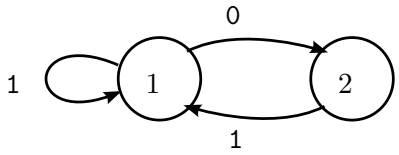
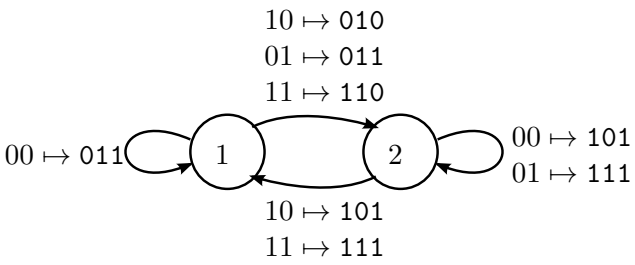


Figure 6.3: Directed graph generating the $(0,1)$-constrained channel.

State splitting algorithm



VLM$(\mathcal{M},3)$ code

00	↦ 111
010	↦ 1101
011	↦ 1011
1000	↦ 10101
101	↦ 0111
110	↦ 01101
111	↦ 01011
1001	↦ 010101

Figure 6.4: On the left-hand side, the code from [54, Example 5.4.5] for the $(0,1)$ constraint is displayed. A memoryless representation of the $(0,1)$ constraint is $\mathcal{M} = \{1, 01\}^+$. On the right-hand side, the VLM code for $\ell = 2$ and blocklength $k = 3$ is shown.

6.4.3 Coding for $(0,1)$ constraint

We now consider the $(0,1)$ constraint. A generating graph is displayed in Figure 6.3. As memoryless representation, we choose the set of strings that start and end at state 1, i.e,

$$\mathcal{M}^+ = \{1, 01\}^+. \tag{6.120}$$

The VLM code for the whole set $\mathcal{M}$ and blocklength 3 is displayed in Figure 6.4. For comparison, we list the fixed-rate 2/3 code that was obtained by the *State Splitting Algorithm* in [54, Example 5.4.5]. The capacity of the $(0,1)$ constraint is $\mathsf{C} = 0.6942$. The rate achieved by the VLM code is 0.6866 and -1.1% within capacity, while the fixed-rate code has rate 2/3 and is -2.9% within capacity.

6.4.4 Huffman source coding is not optimal

Kerpez proposed in [48] to use the Huffman code [45] of the capacity-achieving pmf as matching code for (d,k) constraints. We now provide an example that shows that doing so is in general suboptimal. For $(d,k) = (10,19)$, the memoryless representation $\mathcal{M}^+$ is given by

$$\mathcal{M}^+ = \{\underbrace{0\cdots0}_{k \text{ times}}1 \mid 10 \le k \le 19\}^+. \tag{6.121}$$

Capacity C of $\mathcal{M}^+$ is calculated by Algorithm 6 and the capacity-achieving pmf $\boldsymbol{p}^*$ is calculated according to (4.57). We calculate the Huffman code of $\boldsymbol{p}^*$ and we also calculate $\textsc{vlm}(\mathcal{M}, 1)$. The codeword lengths found by VLM and Huffman coding are respectively given by

$$-\log_2 \boldsymbol{d}_{\text{VLM}} = (3\ 3\ 3\ 3\ 3\ 3\ 4\ 4\ 4\ 4)^T \tag{6.122}$$

$$-\log_2 \boldsymbol{d}_{\text{HC}} = (2\ 3\ 3\ 3\ 3\ 4\ 4\ 4\ 5\ 5)^T \tag{6.123}$$

Capacity and the rates achieved by VLM and Huffman coding are respectively given by

$$\mathsf{C} = 0.22334, \qquad \mathbb{R}_{\text{VLM}} = 0.22034, \qquad \mathbb{R}_{\text{HC}} = 0.22022. \tag{6.124}$$

Thus, VLM lies within -1.3622% of capacity, which is slightly better than Huffman coding, which lies within -1.4148% of capacity.

6.5 References

Parts of this chapter were published in [15, 14, 16].

The investigation of the exponential growth of combinatorial structures by generating functions is put forward by Flajolet and Sedgewick [34]. The authors restrict their investigations to integer valued string lengths.

Finite state noiseless channels were introduced by Shannon in [69, Chapter 1]. Shannon's results on capacity and entropy rate were made precise and extended to arbitrary (in particular non-integer valued) symbol lengths by Khandekar *et al* [49]. They state explicit formulas for the combinatorial capacity and the transition probabilities that achieve capacity. The resulting sources are Markov chains on the generating graph. Their results apply for finite state channels with primitive adjacency matrices and are based on Perron-Frobenius theory. Khandekar *et al* observed (6.68) in the proof of [49, Corollary 4.3].

The construction of fixed rate capacity-achieving codes is based on the *State Splitting Algorithm*. The surrounding theory is developed in detail by Lind and Marcus [54] and by Marcus *et al* [59]. This technique applies for integer valued symbols and the resulting codes have memory.

7 Matching for systematic block codes

The result of this work so far is that when the input pmf of a communication channel is generated by a prefix-free matcher, the resulting mutual information between input and output can be made arbitrary close to capacity. An important operational meaning of capacity is that, according to the channel coding theorem [24, Theorem 7.7.1], there exist error-correcting codes that allow for any rate smaller than capacity the transmission of data with a probability of error that vanishes when the length of the codes goes to infinity. This theorem was originally stated by Shannon in 1948 [69, Theorem 11] and it was a long standing problem to find such codes. To a certain extend, this problem is solved now. To quote MacKay [57, Section 50.7],

> *The best solution to the communication problem is: Combine a simple, pseudo-random code with a message-passing decoder.*

The topic of this chapter is the question how prefix-free matchers can be combined with existing error-correcting codes. We consider this question for a family of codes that form a part of MacKay's solution, namely *systematic linear block codes*. We will define them precisely in Subsection 7.1.2. The two main results of this chapter are as follows:

1. We develop analytical formulas for shaping gain and coding gain. The shaping gain quantifies how much rate is lost compared to capacity because of mismatched channel input pmfs. The coding gain quantifies how much rate is lost because of imperfections of the applied systematic block code. Shaping and coding gain allow to determine if rather the imperfection of the prefix-free matcher or the imperfection of the error-correcting code form the bottleneck of the considered system.

2. We develop a scheme that allows to combine a prefix-free matcher with existing systematic block codes such that the resulting shaping gain is one, i.e., no loss occurs because of mismatched channel input pmfs.

Our results apply to any systematic block code. In Chapter 8, we apply the results to the low-density parity-check codes used in the DVB-S2 standard. We derive our results for dmcs with unequal symbol durations. One reason for this is that, by choosing unequal symbol durations, it is easy to create examples where rather the shaping gain than the coding gain is the bottleneck, even in the case of channels with binary input. For equal symbol durations, the shaping gain of a uniform input pmf for channels with binary input is lower bounded by 0.942 [58], with equality for the Z channel when the transition probability ϵ approaches 1 [66, Section 5.2]. This implies that the shaping gain will hardly become the bottleneck for a practical binary channel with equal symbol

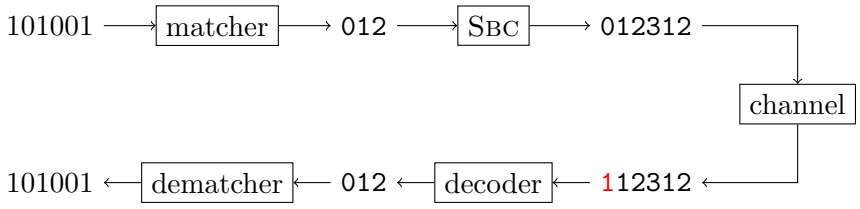


Figure 7.1: Reverse concatenation. The channel input alphabet is $\{0, 1, 2, 3\}$ and the matcher code is $1 \mapsto 0, 01 \mapsto 1, 001 \mapsto 2, 000 \mapsto 3$. The encoder adds parity checks. The parity check matrix is $\boldsymbol{H} = \left[\begin{smallmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{smallmatrix}\right]$. The vector of information symbols that corresponds to the data in the figure is $\boldsymbol{s} = (0\ 1\ 2)^T$ and the vector of parity check symbols is $(3\ 1\ 2)^T$. The transmitted symbols are $\boldsymbol{t} = \left[\begin{smallmatrix} \boldsymbol{s} \\ \boldsymbol{r} \end{smallmatrix}\right]$. It can be checked that the "parity check" is fulfilled, i.e., $\boldsymbol{Ht} = \boldsymbol{0}$ mod 4. After the channel, one symbol is corrupted. This is indicated by the color red. The decoder corrects this error and as a consequence, the bit sequence at the output of the dematcher is identical to the original sequence. This illustrates that reverse concatenation allows the combination of prefix-free matchers with error-correcting codes without resulting in a disastrous symbol-error propagation.

durations. Of course, all our results of this chapter apply as a special case to channels with equal symbol durations.

7.1 Matching and error-correction

In this section, we first introduce the problem of combining prefix-free matchers with error-correcting codes. We then formally define systematic block codes and develop the notions of shaping and coding gain. For clarity, we postpone the consideration of unequal symbol durations to the following sections.

7.1.1 Reverse concatenation

Lets recall how channel matching is incorporated into a digital communication system. The digital interface between source and channel coding is a stream of iid equiprobable bits. By parsing the stream by a full prefix-free code, a dyadic pmf can be generated. For example, consider the set of symbols $\{0, 1, 2, 3\}$. Then the mapping

$$\begin{aligned} 1 &\mapsto 0 \\ 01 &\mapsto 1 \\ 001 &\mapsto 2 \\ 000 &\mapsto 3 \end{aligned} \tag{7.1}$$

generates the pmf $(2^{-1}, 2^{-2}, 2^{-3}, 2^{-3})^T$ over the set $\{0, 1, 2, 3\}$ when the stream of iid equiprobable bits is parsed by the prefix-free code $\{1, 01, 001, 000\}$. A device that implements this procedure is called a prefix-free matcher. When prefix-free matchers are used for noisy channels, a severe problem occurs: one single bit error can lead to a complete loss of a block, since the binary input and output streams are out of sync, e.g., suppose the data bits 101001 were mapped by the matcher (3.4) to the block 012 and then transmitted over the channel, and suppose 112 was detected at the channel output. Then,

$$101001 \mapsto 012 \tag{7.2}$$

$$112 \mapsto 0101001 \tag{7.3}$$

i.e., matcher input and dematcher output are of different length and aligning the first two bits leads to 5 bit errors in the overlapping strings. Error-correction based on matcher input and dematcher output needs the capability of correcting insertion and deletion errors, which is difficult [62]. The above problem can be solved by interchanging the order of matching and error-correction. This is illustrated in the generic block-diagram in Figure 7.1. We refer to this scheme by *reverse concatenation*, a term that was coined in [7].

7.1.2 Systematic linear block codes

We consider *systematic linear block codes* as described in [26, Appendix A]. Both the information symbols, the check symbols, and the entries of the parity check matrix are elements from $\mathbf{Z}_n = \{0, 1, \ldots, n-1\}$. Addition is taken modulo n. Denote by $\boldsymbol{s}$ a vector of information symbols with K entries. Denote the number of check symbols by M and the codeword length by $N = K + M$. The information symbols $\boldsymbol{s}$ are encoded by multiplying $\boldsymbol{s}$ by a matrix $\boldsymbol{G}$. The matrix is called a *systematic generator matrix* if it is of the form

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{P} \end{bmatrix} \tag{7.4}$$

where $\boldsymbol{I}$ is the identity matrix of dimension K and where $\boldsymbol{P} \in \mathbf{Z}_n^M \times \mathbf{Z}_n^K$. The codeword is then given by

$$\boldsymbol{t} = \boldsymbol{G}\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{r} \end{bmatrix} \mod n. \tag{7.5}$$

Thus, the upper K entries of $\boldsymbol{t}$ reproduce the information symbols and the lower M entries of $\boldsymbol{t}$ contain the check symbols $\boldsymbol{r}$. The *systematic parity-check matrix* $\boldsymbol{H}$ is given by

$$\boldsymbol{H} = \begin{bmatrix} -\boldsymbol{P} & \boldsymbol{I} \end{bmatrix} \mod n. \tag{7.6}$$

Thus, for any information vector $\boldsymbol{s} \in \mathbf{Z}_n^K$, the parity-check for the corresponding uncorrupted codeword

$$\boldsymbol{H}\boldsymbol{t} = \boldsymbol{H}\boldsymbol{G}\boldsymbol{s} = (-\boldsymbol{P} + \boldsymbol{P})\boldsymbol{s} = \boldsymbol{0}\boldsymbol{s} = \boldsymbol{0} \mod n \tag{7.7}$$

is fulfilled. We define the *coding rate* by

$$c := \frac{K}{M+K} = \frac{K}{N}. \tag{7.8}$$

See Figure 7.1 for a simple example of a systematic block code.

7.1.3 Shaping gain, coding gain, and capacity

A systematic block code replicates the information symbols $\boldsymbol{s}$ and generates the check symbols $\boldsymbol{r}$ as a deterministic function of $\boldsymbol{s}$. The concatenation of $\boldsymbol{s}$ and $\boldsymbol{r}$ is then transmitted over the channel. As a consequence, the block code influences the pmf of the channel input. To evaluate the performance of a block code, we compare the achieved transmission rate at a certain target block error rate to the capacity C of the considered channel. In our theoretical derivations, we leave the target block error rate implicit, but we make it explicit in Chapter 8 where we give numeric results. Assume the information symbols at the output of the matcher are iid according to the pmf $\boldsymbol{p}$ and denote by c the employed coding rate. We define the *overall gain* of the considered code as the ratio of the achieved transmission rate $\mathbb{R}(\boldsymbol{p}, c)$ and the capacity C of the considered channel, i.e.,

$$\frac{\mathbb{R}(\boldsymbol{p}, c)}{\mathsf{C}}. \tag{7.9}$$

For example, consider a dmc where all input symbol durations are equal to one. Since there are K information symbols and N symbols in total in one block, the transmission rate is given by

$$\mathbb{R}(\boldsymbol{p}, c) = \frac{K\,\mathbb{H}(\boldsymbol{p})}{N} = c\,\mathbb{H}(\boldsymbol{p}) \tag{7.10}$$

and the overall gain is

$$\frac{c\,\mathbb{H}(\boldsymbol{p})}{\mathsf{C}}. \tag{7.11}$$

We now want to split the overall gain into two factors, the *shaping gain* and the *coding gain*. The shaping gain should characterize the influence of the channel input pmf on the transmission rate and the coding gain should characterize the effect of the error-correcting code onto the transmission rate. The goal of this subsection is to formally define shaping and coding gain. Our definition is based on two assumptions, which we detail next.

Uniform check symbol assumption

Consider an infinite sequence of symbols $\{S_i\}_{i=1}^{\infty}$. Assume the symbols take values in $\mathbf{Z}_n$ and are iid according to some pmf $\boldsymbol{p}$. Then, under mild restrictions on $\boldsymbol{p}$ (the Toeplitz matrix with cyclic shifts of $\boldsymbol{p}$ as rows has to be aperiodic) the sum of all symbols modulo n takes values in $\mathbf{Z}_n$ and is uniformly distributed, i.e.

$$\sum_{i=1}^{\infty} S_i \bmod n \sim \boldsymbol{u}. \tag{7.12}$$

Because of this, we assume the following.

Assumption 1. *For any pmf of the information symbols, the check symbols of a systematic block code are iid according to the uniform pmf.*

Ideal systematic block code assumption

Denote by S the random vector of information symbols that are iid according to $\boldsymbol{p}$. Denote by R the random vector of check symbols and by T the random codeword that results from stacking S and R. Then

$$\begin{aligned} \mathbb{H}(T) &= \mathbb{H}(S) + \mathbb{H}(R|S) && (7.13)\\ &= \mathbb{H}(S) && (7.14)\\ &= K\,\mathbb{H}(\boldsymbol{p}) && (7.15) \end{aligned}$$

where the second line follows since the check symbols R depend deterministically on the information symbols S. Denote by Y the channel output that results from the input T. We now have

$$\begin{aligned} \mathbb{I}(T;Y) &\leq \mathbb{I}(S;Y_1^K) + \mathbb{I}(R;Y_{K+1}^N) && (7.16)\\ &= K\,\mathbb{I}(\boldsymbol{p}) + M\,\mathbb{I}(\boldsymbol{u}) && (7.17) \end{aligned}$$

where $Y_1^K = (Y_1, \ldots, Y_K)^T$ and where Y_{K+1}^N is defined accordingly. The inequality in the first line follows from [24, Lemma 7.9.2] and the equality in the second line follows from the assumption that the information symbols are iid according to $\boldsymbol{p}$ and the uniform check symbol assumption. We now assume the following.

Assumption 2. *For each information symbol pmf $\boldsymbol{p}$, there exists an* ideal systematic block code *with block error rate zero for which the transmission rate is equal to the mutual information, i.e.,*

$$\mathbb{H}(S) = \mathbb{I}(T;Y). \tag{7.18}$$

Furthermore, we have equality in (7.16), i.e.

$$\mathbb{I}(T;Y) = K\,\mathbb{I}(\boldsymbol{p}) + M\,\mathbb{I}(\boldsymbol{u}). \tag{7.19}$$

Normalizing by $N = K + M$ yields

$$\frac{\mathbb{I}(T;Y)}{N} = c\,\mathbb{I}(\boldsymbol{p}) + (1-c)\,\mathbb{I}(\boldsymbol{u}). \tag{7.20}$$

We call the coding rate c for which $\mathbb{H}(S) = \mathbb{I}(T;Y)$ the ideal coding rate *and denote it by $c^*(\boldsymbol{p})$.*

We are now in the position to define shaping and coding gain.

Shaping gain

For an information symbol pmf $\boldsymbol{p}$, we define the shaping gain as the ratio of the transmission rate achieved by an ideal systematic block code and the capacity of the channel, i.e.,

$$\frac{\mathbb{R}[\boldsymbol{p}, c^*(\boldsymbol{p})]}{\mathsf{C}}. \tag{7.21}$$

Coding gain

For an information symbol pmf $\boldsymbol{p}$ and an employed coding rate c, we define the coding gain as the ratio of the achieved transmission rate and the transmission rate of an ideal systematic block code, i.e.,

$$\frac{\mathbb{R}(\boldsymbol{p}, c)}{\mathbb{R}[\boldsymbol{p}, c^*(\boldsymbol{p})]}. \tag{7.22}$$

Overall gain

The overall gain is given by the product of shaping gain and coding gain, i.e.,

$$\frac{\mathbb{R}[\boldsymbol{p}, c^*(\boldsymbol{p})]}{\mathsf{C}} \cdot \frac{\mathbb{R}(\boldsymbol{p}, c)}{\mathbb{R}[\boldsymbol{p}, c^*(\boldsymbol{p})]} = \frac{\mathbb{R}(\boldsymbol{p}, c)}{\mathsf{C}}. \tag{7.23}$$

This coincides with the definition of overall gain that we originally started with. Thus, although our definitions of shaping and coding gain are based on two idealizing assumptions, their product is well-defined in practice for any systematic block code.

Capacity of a transmission scheme

We will in the following evaluate three schemes to operate systematic block codes, namely *uniform transmission*, *sparse-dense transmission*, and *matched transmission*. Denote by $\boldsymbol{p}$ the pmf of the information symbols and denote by $\mathbb{R}_{\text{scheme}}[\boldsymbol{p}, c^*(\boldsymbol{p})]$ the transmission rate that is achieved when an ideal systematic block code is operated by the considered scheme. We define the capacity of the scheme as

$$\mathsf{C}_{\text{scheme}} := \max_{\boldsymbol{p}} \mathbb{R}_{\text{scheme}}[\boldsymbol{p}, c^*(\boldsymbol{p})] \tag{7.24}$$

i.e., the capacity of a transmission scheme is the highest rate that can be achieved within the possible configurations of the scheme by an ideal systematic block code.

7.2 Uniform transmission

We start by considering the case when the matcher generates a uniform pmf by using a prefix-free code where all codewords are of equal length. This is only possible if the number n of channel input symbols is a power of two. Uniform transmission is often found in practice, since in this configuration, the matcher maps blocks of a fixed length of $\log_2 n$ bits to each channel symbol, matcher input and dematcher output are always in sync and a corrupted bit does not lead to a disastrous error propagation as we observed in Subsection 7.1.1. We call this scheme *uniform transmission*.

7.2.1 Uniform capacity

For uniform transmission, the information symbols at the matcher output are distributed according to the uniform pmf $\boldsymbol{u}$. Thus, the only remaining parameter is the employed coding rate c. Define

$$\mathbb{I}_{\mathrm{u}}(c) := \frac{c\,\mathbb{I}(\boldsymbol{u}) + (1-c)\,\mathbb{I}(\boldsymbol{u})}{c\boldsymbol{w}^T\boldsymbol{u} + (1-c)\boldsymbol{w}^T\boldsymbol{u}} \tag{7.25}$$

$$= \frac{\mathbb{I}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}}. \tag{7.26}$$

To ensure that an ideal systematic block code achieves this mutual information per average cost, it has to be equal to the transmission rate, i.e.,

$$\mathbb{R}_{\mathrm{u}}(c) := \frac{c\,\mathbb{H}(\boldsymbol{u})}{c\boldsymbol{w}^T\boldsymbol{u} + (1-c)\boldsymbol{w}^T\boldsymbol{u}} \tag{7.27}$$

$$= \frac{c\,\mathbb{H}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}} \tag{7.28}$$

$$\overset{!}{=} \frac{\mathbb{I}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}}. \tag{7.29}$$

Thus, the ideal coding rate is given by

$$c^* = \frac{\mathbb{I}(\boldsymbol{u})}{\mathbb{H}(\boldsymbol{u})} \tag{7.30}$$

and the ideal transmission rate is

$$\mathbb{R}_{\mathrm{u}} = \frac{\mathbb{I}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}}. \tag{7.31}$$

We conclude that *uniform capacity* is given by

$$\mathsf{C}_{\mathrm{u}} = \mathbb{R}_{\mathrm{u}} \tag{7.32}$$

$$= \frac{\mathbb{I}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}}. \tag{7.33}$$

7.2.2 Uniform gains

Shaping gain

The shaping gain of uniform transmission is given by

$$\frac{\mathbb{R}_{\mathrm{u}}}{\mathsf{C}} = \frac{\mathbb{I}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}}\frac{1}{\mathsf{C}} \tag{7.34}$$

$$= \frac{\mathsf{C}_{\mathrm{u}}}{\mathsf{C}}. \tag{7.35}$$

Coding gain

For an employed coding rate c, the coding gain is

$$\frac{\mathbb{R}_{\mathrm{u}}(c)}{\mathbb{R}_{\mathrm{u}}} = \frac{\frac{c\,\mathbb{H}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}}}{\frac{\mathbb{I}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}}} \tag{7.36}$$

$$= c\frac{\mathbb{H}(\boldsymbol{u})}{\mathbb{I}(\boldsymbol{u})}. \tag{7.37}$$

Note that the average duration $\boldsymbol{w}^T\boldsymbol{u}$ cancels out.

Overall gain

The overall gain is

$$\frac{\mathbb{R}_{\mathrm{u}}}{\mathsf{C}} \cdot \frac{\mathbb{R}_{\mathrm{u}}(c)}{\mathbb{R}_{\mathrm{u}}} = \frac{\mathbb{R}_{\mathrm{u}}(c)}{\mathsf{C}} \tag{7.38}$$

$$= \frac{c\,\mathbb{H}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}} \cdot \frac{1}{\mathsf{C}}. \tag{7.39}$$

7.3 Sparse-dense transmission

We can improve upon uniform transmission by allowing the matcher to use any full prefix-free code. In addition, we now drop the restriction that the number of channel inputs n has to be a power of two. Thus, we can now optimize over the pmf of the information symbols. We start by allowing any (possibly non-dyadic) pmf $\boldsymbol{p}$ and show then that the obtained results can also be achieved by dyadic pmfs $\boldsymbol{d}$. Assume the information symbols are iid according to $\boldsymbol{p}$. Because of Assumption 1, the check symbols continue to be uniformly distributed. This configuration was introduced in [65, Chapter 5] and called *sparse-dense transmission*. Using a non-uniform pmf $\boldsymbol{p}$ for the information symbols reduces the per symbol entropy. The term sparse refers to this reduction. The uniform pmf of the check symbols corresponds to the maximum entropy per symbol and the term dense refers to this property.

7.3.1 Sparse-dense capacity

We start by allowing the information symbols to be distributed according to any pmf from the probability simplex. We will show in Subsection 7.3.5 that the same performance can be achieved by dyadic pmfs. The two parameters we have to choose for sparse-dense transmission are the information symbol pmf $\boldsymbol{p}$ and the coding rate c. Define

$$\mathbb{I}_{\text{sd}}(\boldsymbol{p}, c) := \frac{c\,\mathbb{I}(\boldsymbol{p}) + (1-c)\,\mathbb{I}(\boldsymbol{u})}{c\boldsymbol{w}^T\boldsymbol{p} + (1-c)\boldsymbol{w}^T\boldsymbol{u}}. \tag{7.40}$$

Note that this expression depends on the pmf $\boldsymbol{p}$ and on the coding rate c. The transmission rate achieved by coding rate c is given by

$$\mathbb{R}_{\text{sd}}(\boldsymbol{p}, c) = \frac{c\,\mathbb{H}(\boldsymbol{p})}{c\boldsymbol{w}^T\boldsymbol{p} + (1-c)\boldsymbol{w}^T\boldsymbol{u}}. \tag{7.41}$$

An ideal systematic block code achieves the mutual information $\mathbb{I}_{\text{sd}}(\boldsymbol{p}, c)$ if it is equal to the transmission rate. Thus, *sparse-dense capacity* is given by the solution of the following optimization problem.

$$\begin{aligned} \underset{\boldsymbol{p},c}{\text{maximize}} \quad & \mathbb{R}_{\text{sd}}(\boldsymbol{p}, c) \\ \text{subject to} \quad & \mathbb{R}_{\text{sd}}(\boldsymbol{p}, c) = \mathbb{I}_{\text{sd}}(\boldsymbol{p}, c) \\ & 0 \le c \le 1 \\ & \boldsymbol{p} \text{ is a pmf.} \end{aligned} \tag{7.42}$$

For a fixed input pmf $\boldsymbol{p}$, we solve the equality constraint for the code rate c, i.e,

$$\mathbb{R}_{\text{sd}}(\boldsymbol{p}, c) = \mathbb{I}_{\text{sd}}(\boldsymbol{p}, c) \tag{7.43}$$

$$\Leftrightarrow c\,\mathbb{H}(\boldsymbol{p}) = c\,\mathbb{I}(\boldsymbol{p}) + (1-c)\,\mathbb{I}(\boldsymbol{u}) \tag{7.44}$$

$$\Leftrightarrow c = \frac{\mathbb{I}(\boldsymbol{u})}{\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p}) + \mathbb{I}(\boldsymbol{u})} \tag{7.45}$$

and we conclude that the ideal coding rate is given by

$$c^*(\boldsymbol{p}) = \frac{\mathbb{I}(\boldsymbol{u})}{\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p}) + \mathbb{I}(\boldsymbol{u})}. \tag{7.46}$$

Using this expression for the coding rate in (7.41), we get for the transmission rate achieved by an ideal systematic block code

$$\begin{aligned}
\mathbb{R}_{\text{sd}}(\boldsymbol{p}) &:= \mathbb{R}_{\text{sd}}[\boldsymbol{p}, c^*(\boldsymbol{p})] && (7.47)\\
&= \frac{c\,\mathbb{H}(\boldsymbol{p})}{c\boldsymbol{w}^T\boldsymbol{p} + (1-c)\boldsymbol{w}^T\boldsymbol{u}}\bigg|_{c=c^*(\boldsymbol{p})} && (7.48)\\
&= \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p} + (\frac{1}{c}-1)\boldsymbol{w}^T\boldsymbol{u}}\bigg|_{c=c^*(\boldsymbol{p})} && (7.49)\\
&= \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p} + (\frac{\mathbb{H}(\boldsymbol{p})-\mathbb{I}(\boldsymbol{p})+\mathbb{I}(\boldsymbol{u})}{\mathbb{I}(\boldsymbol{u})}-1)\boldsymbol{w}^T\boldsymbol{u}} && (7.50)\\
&= \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\text{u}}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]} && (7.51)
\end{aligned}$$

We define

$$\mathbb{I}_{\text{sd}}(\boldsymbol{p}) := \mathbb{I}_{\text{sd}}[\boldsymbol{p}, c^*(\boldsymbol{p})]. \tag{7.52}$$

Note that $\mathbb{I}_{\text{sd}}(\boldsymbol{p}) = \mathbb{R}_{\text{sd}}(\boldsymbol{p})$, i.e., the transmission rate achieved by an ideal block code is identical to the mutual information rate achieved by an ideal block code. We will use in the following the notion of mutual information rate rather than the notion of transmission rate. In summary, we have shown the following.

Proposition 7.1. Sparse-dense capacity C_{sd} *is given by*

$$\mathsf{C}_{\text{sd}} = \max_{\boldsymbol{p}} \mathbb{I}_{\text{sd}}(\boldsymbol{p}) = \max_{\boldsymbol{p}} \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\text{u}}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]}. \tag{7.53}$$

Denote by $\boldsymbol{p}^*$ *the capacity-achieving pmf. The corresponding ideal code rate is given by*

$$c^*(\boldsymbol{p}^*) = \frac{\mathbb{I}(\boldsymbol{u})}{\mathbb{H}(\boldsymbol{p}^*) - \mathbb{I}(\boldsymbol{p}^*) + \mathbb{I}(\boldsymbol{u})}. \tag{7.54}$$

7.3.2 Calculating sparse-dense capacity

The objective of this subsection is to solve the optimization problem (7.53), i.e., we want to derive an algorithm that finds the pmf $\boldsymbol{p}^*$ that maximizes the mutual information per average duration $\mathbb{I}_{\text{sd}}(\boldsymbol{p})$. Recall that $\mathbb{I}(\boldsymbol{p})$ and $\mathbb{H}(\boldsymbol{p})$ are concave in $\boldsymbol{p}$. Thus, $\mathbb{I}_{\text{sd}}(\boldsymbol{p})$ is a fraction with a concave numerator and a denominator that is the difference of two concave functions. There is no algorithm known to us that directly maximizes this kind of functions. Therefore, we first remove the fraction from the objective function by the same technique we applied before in Algorithm 4. Denote by $\boldsymbol{p}^*$ a capacity-achieving pmf, i.e., $\mathsf{C}_{\text{sd}} = \mathbb{I}_{\text{sd}}(\boldsymbol{p}^*)$. Then

$$\mathbb{I}_{\text{sd}}(\boldsymbol{p}) = \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\text{u}}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]} \leq \mathsf{C}_{\text{sd}}, \quad \text{with equality if } \boldsymbol{p} = \boldsymbol{p}^*. \tag{7.55}$$

Multiplying by the denominator and moving all terms to the left-hand side, we get

$$\mathbb{H}(\boldsymbol{p}) - \mathsf{C}_{\mathrm{sd}}\Big\{\boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}\big[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})\big]\Big\} \leq 0, \quad \text{with equality if } \boldsymbol{p} = \boldsymbol{p}^*. \tag{7.56}$$

Thus, $\boldsymbol{p}^*$ can be found by maximizing the left-hand side of the last inequality. Sparse-dense capacity C_{sd} is in general not known, so we replace it by an estimate C. By iteratively maximizing (7.56) and using the resulting pmf $\boldsymbol{p}'$ to update C via $C = \mathbb{I}_{\mathrm{sd}}(\boldsymbol{p}')$, the estimate C converges to the maximum C_{sd}. This can be shown along the lines of the proof of Proposition 4.5. It remains to find a way to maximize the left-hand side of (7.56) over the pmf $\boldsymbol{p}$. Replacing C_{sd} by its estimate C, the negative of the left-hand side of (7.56) can be written as

$$\begin{aligned} -\mathbb{H}(\boldsymbol{p}) + C\Big\{\boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}\big[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})\big]\Big\} &= \underbrace{C\big[\boldsymbol{w}^T\boldsymbol{p} - \frac{\mathbb{I}(\boldsymbol{p})}{\mathsf{C}_{\mathrm{u}}}\big]}_{=:f(\boldsymbol{p},C)} - \underbrace{\mathbb{H}(\boldsymbol{p})\big(1 - \frac{C}{\mathsf{C}_{\mathrm{u}}}\big)}_{=:g(\boldsymbol{p},C)} && (7.57)\\ &= f(\boldsymbol{p},C) - g(\boldsymbol{p},C) && (7.58)\\ &=: h(\boldsymbol{p},C). && (7.59) \end{aligned}$$

Since $-\mathbb{I}(\boldsymbol{p})$ is convex in $\boldsymbol{p}$ and $\boldsymbol{w}^T\boldsymbol{p}$ is linear in $\boldsymbol{p}$, the first function f is convex in $\boldsymbol{p}$. The sparse-dense capacity is greater than or equal to the uniform capacity, so for $\mathsf{C}_{\mathrm{u}} \leq C \leq \mathsf{C}_{\mathrm{sd}}$, we have

$$1 - \frac{C}{\mathsf{C}_{\mathrm{u}}} \leq 0. \tag{7.60}$$

Consequently, for $\mathsf{C}_{\mathrm{u}} \leq C \leq \mathsf{C}_{\mathrm{sd}}$, also the second function g is convex in $\boldsymbol{p}$ and we conclude that $h(\boldsymbol{p})$ is the difference of two convex functions. This says little about h. For example, according to [78, Theorem 2], any function with bounded Hessian can be written as the difference of two convex functions. However, a local minimum can be found by the *convex-concave procedure* as defined in [18, slide 26]. The convex-concave procedure is an iterative method and works as follows. In some iteration step, denote the result from the previous step by $\boldsymbol{p}'$. Then, replace g by its first order Taylor approximation in $\boldsymbol{p}'$. Without the factor $(1 - \frac{C}{\mathsf{C}_{\mathrm{u}}})$, this approximation is given by

$$\begin{aligned} \mathbb{H}(\boldsymbol{p}') + \sum_i \frac{\partial\,\mathbb{H}(\boldsymbol{p}')}{\partial p_i'}(p_i - p_i') &= \mathbb{H}(\boldsymbol{p}') + \sum_i (-\log p_i' - 1)(p_i - p_i') && (7.61)\\ &= -\sum_i p_i \log p_i' - \sum_i (p_i - p_i') && (7.62)\\ &= -\sum_i p_i \log p_i'. && (7.63) \end{aligned}$$

Thus, including the factor $(1 - \frac{C}{\mathsf{C}_{\mathrm{u}}})$, the first order Taylor approximation of g in $\boldsymbol{p}'$ is given by

$$\hat{g}(\boldsymbol{p},\boldsymbol{p}',C) = \big(1 - \frac{C}{\mathsf{C}_{\mathrm{u}}}\big)\Big[-\sum_i p_i \log p_i'\Big]. \tag{7.64}$$

Note that $\hat{g}(\boldsymbol{p}, \boldsymbol{p}', C)$ is affine in $\boldsymbol{p}$. The new objective becomes

$$\hat{h}(\boldsymbol{p}, \boldsymbol{p}', C) = f(\boldsymbol{p}, C) - \hat{g}(\boldsymbol{p}, \boldsymbol{p}', C). \tag{7.65}$$

The function $\hat{h}$ is now the difference of a convex and an affine function and thus convex. Thus, it can efficiently be minimized over $\boldsymbol{p}$. The whole algorithm now is as follows.

Algorithm 7.

$\boldsymbol{p}' = \boldsymbol{u}$
repeat
  1. $C = \mathbb{I}_{\text{sc}}(\boldsymbol{p}')$
  repeat
    2. $\boldsymbol{p}'' = \underset{\boldsymbol{p} \text{ pmf}}{\operatorname{argmin}} \hat{h}(\boldsymbol{p}, \boldsymbol{p}', C)$
    3. $\boldsymbol{p}' = \boldsymbol{p}''$
  until convergence
until convergence

Operations 1. and 3. are simple assignments and operation 2. consists in solving a convex optimization problem, which can efficiently be done by convex optimization software as for example CVX [41]. We have the following observations about Algorithm 7:

- Algorithm 7 finds a local maximum of $\mathbb{I}_{\text{sd}}(\boldsymbol{p})$.
- If the channel is noiseless, i.e., if $\mathbb{I}(\boldsymbol{p}) = \mathbb{H}(\boldsymbol{p})$, then the objective function h is given by

  $$C\boldsymbol{w}^T\boldsymbol{p} - \mathbb{H}(\boldsymbol{p}) \tag{7.66}$$

  and convex in $\boldsymbol{p}$. Thus, in this case, Algorithm 7 globally solves the optimization problem and finds capacity and capacity-achieving pmf.
- If Algorithm 7 always converges to the true sparse-dense capacity remains an open question.

In the following, we assume that sparse-dense capacity was actually found by Algorithm 7. This can be achieved by using an alternative way (e.g., exhaustive search) to find capacity and capacity-achieving pmf and then by initializing Algorithm 7 with an appropriate starting point. The key observation is that sparse-dense capacity C_{sd} and capacity-achieving pmf $\boldsymbol{p}^*$ are optimal value and optimal point of a convex optimization problem, namely the optimization problem that is solved in the inner loop of Algorithm 7 after convergence, i.e., with the parameters $C = \mathsf{C}_{\text{sd}}$ and $\boldsymbol{p}' = \boldsymbol{p}^*$. Thus, we can use KKT conditions to characterize C_{sd} and $\boldsymbol{p}^*$. We will need this for showing asymptotic achievability of prefix-free matchers.

7.3.3 Capacity-achieving pmf

Proposition 7.2. *The pmf $\boldsymbol{p}^*$ that achieves sparse-dense capacity fulfills*

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} \leq \mathsf{C}_{\mathrm{u}} w_i + \Big(1 - \frac{\mathsf{C}_{\mathrm{u}}}{\mathsf{C}_{\mathrm{sd}}}\Big)(-\log p_i^*), \quad \textit{with equality if } p_i^* > 0. \tag{7.67}$$

Proof. The optimization problem is

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{minimize}} \quad & \hat{h}(\boldsymbol{p}, \boldsymbol{p}^*, \mathsf{C}_{\mathrm{sd}}) \\ & = \mathsf{C}_{\mathrm{sd}}\big[\boldsymbol{w}^T\boldsymbol{p} - \frac{\mathbb{I}(\boldsymbol{p})}{\mathsf{C}_{\mathrm{u}}}\big] - \big(1 - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}}\big)\Big[-\sum_i p_i \log p_i^*\Big] \qquad (7.68) \\ \text{subject to} \quad & \mathbf{1}^T\boldsymbol{p} - 1 = 0. \end{aligned}$$

If $p_i^* = 0$, then $p_i = 0$, since otherwise, the objective function takes the value ∞. Without loss of generality, we assume $p_i^* > 0$ for all i. Under this assumption, the objective function is defined on $\mathbf{R}^n_{\geq 0}$ and by Proposition 2.5, strong duality holds. By Proposition 2.7, the partial derivatives of the objective function are defined with the possible exception of taking the value $-\infty$ on the boundary. Thus, Proposition 2.4 applies. The Lagrangian is

$$L(\boldsymbol{p}, \nu) = \hat{h}(\boldsymbol{p}, \boldsymbol{p}^*, \mathsf{C}_{\mathrm{sd}}) + \nu(\mathbf{1}^T\boldsymbol{p} - 1). \tag{7.69}$$

Note that any pmf is feasible. Thus, according to Proposition 2.4, a pmf $\boldsymbol{p}$ is optimal if and only if there exists a ν such that the following conditions hold.

$$\begin{aligned} \frac{\partial L(\boldsymbol{p}, \nu)}{\partial p_i} &= \mathsf{C}_{\mathrm{sd}} w_i - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}} \frac{\partial\, \mathbb{I}(\boldsymbol{p})}{\partial p_i} - \Big(1 - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}}\Big)(-\log p_i^*) + \nu \\ &= 0, \qquad \forall i : p_i > 0 \qquad (7.70) \\ \frac{\partial L(\boldsymbol{p}, \nu)}{\partial p_i} &\geq 0, \qquad \forall i : p_i = 0. \qquad (7.71) \end{aligned}$$

If these conditions hold, then all partial derivatives of $\mathbb{I}$ in $\boldsymbol{p}$ are well-defined and given by

$$\frac{\partial\, \mathbb{I}(\boldsymbol{p})}{\partial p_i} = \sum_j h_{ji} \log \frac{h_{ji}}{r_j} - 1. \tag{7.72}$$

Plugging this into the KKT conditions, we get

$$\begin{aligned} \frac{\partial L(\boldsymbol{p}, \nu)}{\partial p_i} &= \mathsf{C}_{\mathrm{sd}} w_i - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}}\Big(\sum_j h_{ji} \log \frac{h_{ji}}{r_j} - 1\Big) - \Big(1 - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}}\Big)(-\log p_i^*) + \nu \qquad (7.73) \\ &\geq 0, \quad \text{with equality if } p_i > 0. \qquad (7.74) \end{aligned}$$

We multiply the inequality by $\mathsf{C}_\mathrm{u}/\mathsf{C}_\mathrm{sd}$ and solve for the sum to get

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j} \leq \mathsf{C}_\mathrm{u} w_i + \left(1 - \frac{\mathsf{C}_\mathrm{u}}{\mathsf{C}_\mathrm{sd}}\right)(-\log p_i^*) + 1 + \frac{\mathsf{C}_\mathrm{u}}{\mathsf{C}_\mathrm{sd}}\nu, \qquad \text{with equality if } p_i > 0. \tag{7.75}$$

We take the expectation with respect to $\boldsymbol{p}^*$ and get

$$\mathbb{I}(\boldsymbol{p}^*) = \mathsf{C}_\mathrm{u} \boldsymbol{w}^T \boldsymbol{p}^* + \left(1 - \frac{\mathsf{C}_\mathrm{u}}{\mathsf{C}_\mathrm{sd}}\right) \mathbb{H}(\boldsymbol{p}^*) + 1 + \frac{\mathsf{C}_\mathrm{u}}{\mathsf{C}_\mathrm{sd}}\nu. \tag{7.76}$$

On the other hand, we can evaluate (7.53) in $\boldsymbol{p}^*$. By noting that $\mathbb{I}_\mathrm{sd}(\boldsymbol{p}^*) = \mathsf{C}_\mathrm{sd}$ and by solving (7.53) for $\mathbb{I}(\boldsymbol{p}^*)$, we get

$$\mathbb{I}(\boldsymbol{p}^*) = \mathsf{C}_\mathrm{u} \boldsymbol{w}^T \boldsymbol{p}^* + \left(1 - \frac{\mathsf{C}_\mathrm{u}}{\mathsf{C}_\mathrm{sd}}\right) \mathbb{H}(\boldsymbol{p}^*). \tag{7.77}$$

By comparing the two expressions for $\mathbb{I}(\boldsymbol{p}^*)$, we conclude that

$$1 + \frac{\mathsf{C}_\mathrm{u}}{\mathsf{C}_\mathrm{sd}}\nu = 0. \tag{7.78}$$

Using this in (7.75) yields the statement of the proposition. □

7.3.4 Using a 'wrong' pmf

We now derive an expression for the penalty that results from using a pmf different from the capacity-achieving one.

Proposition 7.3. *Denote by $\boldsymbol{p}^*$ a pmf that achieves sparse-dense capacity. Denote by $\boldsymbol{p}$ a pmf with the only restriction that*

$$p_i = 0 \quad \textit{whenever } p_i^* = 0. \tag{7.79}$$

Then the mutual information per average cost achieved by $\boldsymbol{p}$ is given by

$$\mathbb{I}_\mathrm{sd}(\boldsymbol{p}) = \mathsf{C}_\mathrm{sd} \frac{\mathbb{H}(\boldsymbol{p})}{-\sum_i p_i \log p_i^* - \frac{\mathsf{C}_\mathrm{sd}}{\mathsf{C}_\mathrm{u}}\left[\mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\right]}. \tag{7.80}$$

Proof. Because of our assumption (7.79), Proposition 3.10 applies and the mutual information $\mathbb{I}(\boldsymbol{p})$ can be written as

$$\mathbb{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*). \tag{7.81}$$

By (7.53), the mutual information per average duration that results from $\boldsymbol{p}$ is given by

$$\mathbb{I}_\mathrm{sd}(\boldsymbol{p}) = \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_\mathrm{u}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]}. \tag{7.82}$$

The denominator can now be written as

$$
\begin{aligned}
&\boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})] \\
&= \boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}\left\{\mathbb{H}(\boldsymbol{p}) - \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} + \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\right\} && (7.83)\\
&= \boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}\left\{\mathbb{H}(\boldsymbol{p}) - \sum_i p_i \left(\mathsf{C}_{\mathrm{u}} w_i + \Big(1 - \frac{\mathsf{C}_{\mathrm{u}}}{\mathsf{C}_{\mathrm{sd}}}\Big)(-\log p_i^*)\right) + \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\right\} && (7.84)\\
&= \boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}\left\{\mathbb{H}(\boldsymbol{p}) - \left[\mathsf{C}_{\mathrm{u}}\boldsymbol{w}^T\boldsymbol{p} + \Big(1 - \frac{\mathsf{C}_{\mathrm{u}}}{\mathsf{C}_{\mathrm{sd}}}\Big)(-\sum_i p_i \log p_i^*)\right] + \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\right\} && (7.85)\\
&= \frac{1}{\mathsf{C}_{\mathrm{u}}}\left\{\mathbb{H}(\boldsymbol{p}) - \Big(1 - \frac{\mathsf{C}_{\mathrm{u}}}{\mathsf{C}_{\mathrm{sd}}}\Big)(-\sum_i p_i \log p_i^*) + \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\right\} && (7.86)\\
&= \frac{1}{\mathsf{C}_{\mathrm{u}}}\left\{\mathbb{H}(\boldsymbol{p}) + \sum_i p_i \log p_i^* - \frac{\mathsf{C}_{\mathrm{u}}}{\mathsf{C}_{\mathrm{sd}}}\sum_i p_i \log p_i^* + \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\right\} && (7.87)\\
&= \frac{1}{\mathsf{C}_{\mathrm{u}}}\left\{-\frac{\mathsf{C}_{\mathrm{u}}}{\mathsf{C}_{\mathrm{sd}}}\sum_i p_i \log p_i^* - \mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) + \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\right\} && (7.88)\\
&= \frac{1}{\mathsf{C}_{\mathrm{sd}}}\left\{-\sum_i p_i \log p_i^* - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}}\Big[\mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\Big]\right\}. && (7.89)
\end{aligned}
$$

Thus, we get

$$
\begin{aligned}
\mathbb{I}_{\mathrm{sd}}(\boldsymbol{p}) &= \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]} && (7.90)\\
&= \frac{\mathbb{H}(\boldsymbol{p})}{\frac{1}{\mathsf{C}_{\mathrm{sd}}}\left\{-\sum_i p_i \log p_i^* - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}}\big[\mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\big]\right\}} && (7.91)\\
&= \mathsf{C}_{\mathrm{sd}}\frac{\mathbb{H}(\boldsymbol{p})}{-\sum_i p_i \log p_i^* - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}}\big[\mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\big]} && (7.92)
\end{aligned}
$$

which is the statement of the proposition. □

7.3.5 Matching

We now show that sparse-dense capacity can be approximated arbitrarily well by a prefix-free matcher.

Proposition 7.4. *Denote by $\boldsymbol{p}^*$ a pmf that achieves sparse-dense capacity. Denote by $\boldsymbol{p}$ an arbitrary pmf with the only restriction that*

$$p_i = 0 \quad \textit{whenever } p_i^* = 0. \tag{7.93}$$

The following holds.

1. *The mutual information per average duration* $\mathbb{I}(\boldsymbol{p})$ *is lower-bounded by*

$$\mathbb{I}_{\mathrm{sd}}(\boldsymbol{p}) \geq \mathsf{C}_{\mathrm{sd}} \frac{\mathbb{H}(\boldsymbol{p})}{-\sum_i p_i \log p_i^*}. \tag{7.94}$$

2. *The lower bound is maximized over all dyadic pmfs by* $\boldsymbol{d} = \textsc{nGhc}(\mathbf{1}, \boldsymbol{\ell})$ *where* $\ell_i = -\log p_i^*$, $i = 1, \ldots, n$.

3. $\boldsymbol{d}_k = \textsc{nGhc}(\mathbf{1}, \boldsymbol{\ell}^{\oplus k})$ *achieves capacity for blocklength* $k \to \infty$.

Proof. Because of (7.93), Proposition 7.3 applies and we have

$$\mathbb{I}_{\mathrm{sd}}(\boldsymbol{p}) = \mathsf{C}_{\mathrm{sd}} \frac{\mathbb{H}(\boldsymbol{p})}{-\sum_i p_i \log p_i^* - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}} \left[\mathbb{D}(\boldsymbol{p}\|\boldsymbol{p}^*) - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\right]} \tag{7.95}$$

$$\geq \mathsf{C}_{\mathrm{sd}} \frac{\mathbb{H}(\boldsymbol{p})}{-\sum_i p_i \log p_i^* - \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}_{\mathrm{u}}} \left[\mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*) - \mathbb{D}(\boldsymbol{r}\|\boldsymbol{r}^*)\right]} \tag{7.96}$$

$$= \mathsf{C}_{\mathrm{sd}} \frac{\mathbb{H}(\boldsymbol{p})}{-\sum_i p_i \log p_i^*} \tag{7.97}$$

where we used Proposition 3.12 in the second line. This shows part 1. of the proposition.

Define $\boldsymbol{\ell}$ by $\ell_i = -\log p_i^*$, $i = 1, \ldots, n$. The fraction in the lower bound can be written as

$$\frac{\mathbb{H}(\boldsymbol{p})}{-\sum_i p_i \log p_i^*} = -\frac{\mathbb{D}(\boldsymbol{p}\|\mathbf{1})}{\sum_i p_i \ell_i}. \tag{7.98}$$

Note that, by the information inequality,

$$-\sum_i p_i \log p_i^* \geq -\sum_i p_i \log p_i \tag{7.99}$$

$$= \mathbb{H}(\boldsymbol{p}) \tag{7.100}$$

$$\geq 0. \tag{7.101}$$

Consequently, the fractions in (7.98) are always positive and

$$\frac{\mathbb{D}(\boldsymbol{p}\|\mathbf{1})}{\sum_i p_i \ell_i} < 0. \tag{7.102}$$

Define now $\boldsymbol{d} = \textsc{nGhc}(\mathbf{1}, \boldsymbol{\ell})$. nGhc calculates in each step

$$\boldsymbol{d}' = \textsc{Ghc}(\mathbf{1} \circ e^{\Delta \ell}) \tag{7.103}$$

$$= \textsc{Ghc}[(e^{-\ell})^{-\Delta}] \tag{7.104}$$

$$= \textsc{Ghc}(\boldsymbol{p}^{*-\Delta}) \tag{7.105}$$

for some Δ. The exponentiation is elementwise, i.e.,

$$(\boldsymbol{p}^*)^{-\Delta} = (p_1^{*-\Delta}, \ldots, p_n^{*-\Delta})^T. \tag{7.106}$$

Because of (7.102), $\Delta < 0$ and consequently $-\Delta > 0$. Thus, whenever $p_i^* = 0$, $p_1^{*-\Delta} = 0$. This implies because of Proposition 3.2 that $d_i = 0$ whenever $p_i^* = 0$, i.e., $\textsc{nGhc}(\mathbf{1}, \boldsymbol{\ell})$ assigns $d_i = 0$ whenever $p_i^* = 0$. Thus, the bound from statement 1. of this proposition applies for $\mathbb{I}_{\mathrm{sd}}(\boldsymbol{d})$. We have

$$\mathbb{I}_{\mathrm{sd}}(\boldsymbol{d}) \geq \mathsf{C}_{\mathrm{sd}} \frac{\mathbb{H}(\boldsymbol{d})}{-\sum_i d_i \log p_i^*} \tag{7.107}$$

$$= \mathsf{C}_{\mathrm{sd}} \frac{-\mathbb{D}(\boldsymbol{d} \| \mathbf{1})}{\sum_i d_i \ell_i} \tag{7.108}$$

which shows that $\boldsymbol{d} = \textsc{nGhc}(\mathbf{1}, \boldsymbol{\ell})$ maximizes the lower bound. This shows part 2 of this proposition.

Statement 3 follows from the asymptotic achievability of nGhc as stated in Proposition 4.1. This concludes the proof. □

7.3.6 Sparse-dense gains

Shaping gain

Suppose the applied input pmf is $\boldsymbol{p}$. The shaping gain is given by

$$\frac{\mathbb{R}_{\mathrm{sd}}[\boldsymbol{p}, c^*(\boldsymbol{p})]}{\mathsf{C}} = \frac{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{p})}{\mathsf{C}} \tag{7.109}$$

$$= \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T \boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]} \frac{1}{\mathsf{C}}. \tag{7.110}$$

The shaping gain is upper-bounded by

$$\frac{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{p})}{\mathsf{C}} \leq \frac{\mathsf{C}_{\mathrm{sd}}}{\mathsf{C}} \leq 1. \tag{7.111}$$

Coding Gain

For an applied coding rate c, the coding gain is given by

$$\frac{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{p}, c)}{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{p})} = \frac{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{p}, c)}{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{p}, c^*)} \tag{7.112}$$

$$= \frac{c\, \mathbb{H}(\boldsymbol{p})}{c \boldsymbol{w}^T \boldsymbol{p} + (1-c) \boldsymbol{w}^T \boldsymbol{u}} \cdot \frac{\boldsymbol{w}^T \boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]}{\mathbb{H}(\boldsymbol{p})} \tag{7.113}$$

$$= \frac{\boldsymbol{w}^T \boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]}{\boldsymbol{w}^T \boldsymbol{p} + (\frac{1}{c} - 1) \boldsymbol{w}^T \boldsymbol{u}}. \tag{7.114}$$

Overall gain

The overall gain is

$$\frac{\mathbb{I}_{\mathrm{sd}}(\boldsymbol{p})}{\mathsf{C}} \cdot \frac{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{p}, c)}{\mathbb{I}_{\mathrm{sd}}(\boldsymbol{p})} = \frac{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{p}, c)}{\mathsf{C}} \tag{7.115}$$

$$= \frac{c\,\mathbb{H}(\boldsymbol{p})}{c\boldsymbol{w}^T\boldsymbol{p} + (1-c)\boldsymbol{w}^T\boldsymbol{u}} \cdot \frac{1}{\mathsf{C}}. \tag{7.116}$$

7.4 Matched transmission

We now introduce *matched transmission*. The key part of matched transmission is the *bootstrap scheme*, which allows to match the pmf of the check symbols to the capacity achieving pmf of the considered channel. As before, we consider a systematic block code that generates M check symbols per K information symbols, i.e., a rate $c = K/(K+M)$-code. Assume further that the number n of symbols is a power of two and denote the exponent by b, i.e., $n = 2^b$.

7.4.1 Bootstrapping the check symbols

At the transmitter side, B blocks are sequentially encoded and then transmitted in reverse order. For the first block, the matcher generates K matched data symbols from a fair bit stream. These K matched data symbols form the first block of K symbols to be transmitted over the channel. The very same K matched symbols are also passed through the encoder. The encoder calculates M check symbols. In a binary representation, this corresponds to bM check bits. According to the uniform check symbol assumption, the check symbols are iid and equiprobable. Consequently, the binary representation of the check symbols forms a vector of fair bits. This binary vector is passed through the matcher to generate matched check symbols. These matched check symbols are kept back. In the next round, the matcher again generates matched data symbols from the bit stream. These matched data symbols are concatenated with the matched check symbols from the first round. The number of matched data symbols is chosen such that the total number of matched data symbols and matched check symbols is equal to K. This concatenation of matched data symbols and matched check symbols from the previous round again forms a block of K symbols to be transmitted over the channel. This procedure continuous until the last but second round. In the last round, a (M', K') sparse-dense code is applied. Matched check symbols from round $B-1$ plus matched data symbols are concatenated to form a block of K' matched symbols. The encoder calculates M' check symbols from the K' matched symbols and appends them (unmatched) to the K' matched symbols. These M' check symbols of the last round are the *ur-symbols* from which the decoder will start to bootstrap all B blocks. All B blocks are transmitted in reverse order such that the decoder can immediately start decoding. See Figure 7.4 for an illustration.

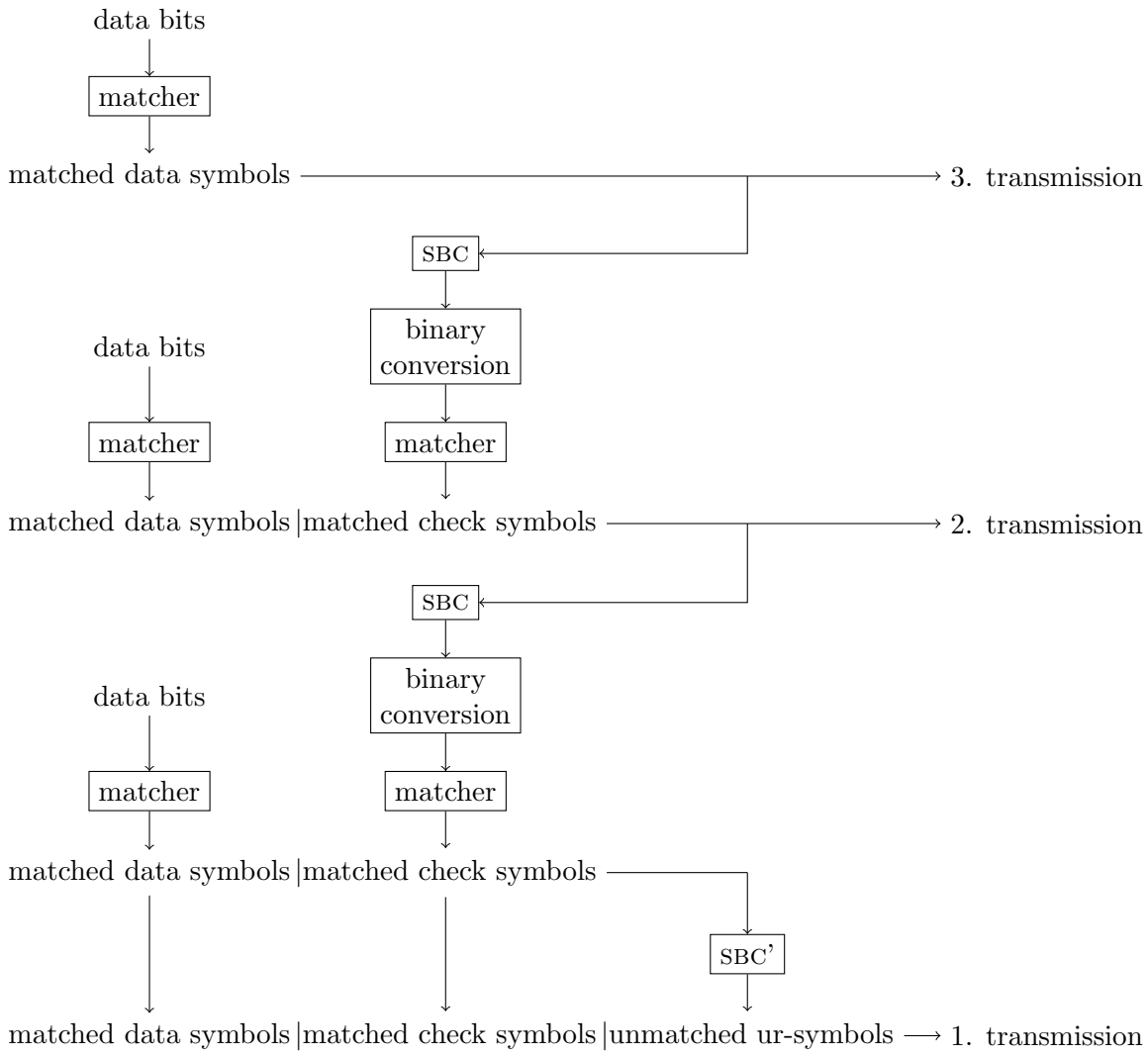


Figure 7.2: Bootstrap scheme. The encoder SBC takes K symbols as input and generates M check symbols as output. The corresponding values for the encoder SBC$'$ are K' and M', respectively. The symbol | denotes concatenation. The diagram displays the bootstrap scheme for $B = 3$ rounds. The second round can be repeated various times to jointly process any number B of blocks. Increasing the number B increases the fraction of matched symbols and consequently increases the effective transmission rate.

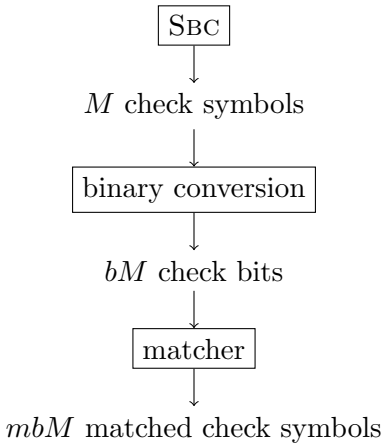


Figure 7.3: Number of matched check symbols.

Number of matched check symbols

We now explicitly calculate the number of matched check symbols that are embedded in the K matched symbols in an intermediate round of the bootstrap scheme. Assume the matched symbols are iid according to $\boldsymbol{p}$. Because we have a fair bit stream at the binary interface, each bit contains the information of $\log 2$ nats. Thus, before matching, the information is $\sharp\{\text{unmatched bits}\}\log 2$. After matching, the information is $\sharp\{\text{matched symbols}\}\,\mathbb{H}(\boldsymbol{p})$. Because the matcher applies a deterministic bijective mapping, the information before and after matching is the same, i.e.,

$$\sharp\{\text{matched symbols}\}\,\mathbb{H}(\boldsymbol{p}) = \sharp\{\text{unmatched bits}\}\log 2. \tag{7.117}$$

We define the *matching rate* as

$$m := \frac{\sharp\{\text{matched symbols}\}}{\sharp\{\text{unmatched bits}\}} = \frac{\log 2}{\mathbb{H}(\boldsymbol{p})} \tag{7.118}$$

The M unmatched check symbols are converted into bM unmatched bits and then matched to the channel, thus, the number of matched check symbols is

$$\sharp\{\text{matched check symbols}\} = mbM = \frac{b\log(2)M}{\mathbb{H}(\boldsymbol{p})}. \tag{7.119}$$

7.4.2 Matched capacity

The number M' of unmatched ur-symbols is independent of the number of packets B. Thus as B grows, the fraction of matched symbols approaches 1. Therefore, we will focus now on the building block of the bootstrap scheme, which consists in the transmission of K matched symbols over a channel and decoding conditioned on M perfectly known check symbols. The M perfectly known check symbols are not for free: the M check

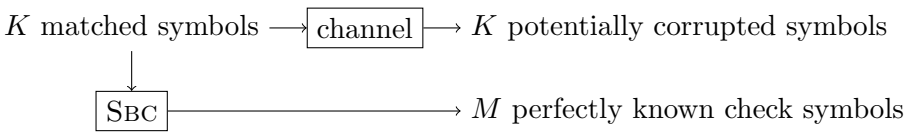


Figure 7.4: Building block of the bootstrap scheme.

symbols of the next packet to be transmitted are embedded in matched form in the K matched symbols. We will take this into account when calculating the transmission rate.

One block contains mbM matched check symbols and $K - mbM$ matched information symbols. Since the matched check symbols belong to the matched information symbols of the previous block, in one block, check symbols and information symbols are stochastically independent. To take the dependencies into account, we now consider $K - mbM$ information symbols from one block together with the corresponding mbM check symbols, which are transmitted as part of the next block. Define

$$\mathbb{I}_{\mathrm{bs}}(\boldsymbol{p}, c) := \frac{(K - mbM)\,\mathbb{I}(\boldsymbol{p}) + mbM\,\mathbb{I}(\boldsymbol{p})}{(K - mbM)\boldsymbol{w}^T\boldsymbol{p} + mbM\boldsymbol{w}^T\boldsymbol{p}} \tag{7.120}$$

$$= \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}}. \tag{7.121}$$

To ensure that an ideal systematic block code achieves this mutual information per average duration, it has to be equal to the transmission rate. The transmission rate is information per average duration, i.e.,

$$\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p}, c) = \frac{(K - mbM)\,\mathbb{H}(\boldsymbol{p})}{(K - mbM)\boldsymbol{w}^T\boldsymbol{p} + mbM\boldsymbol{w}^T\boldsymbol{p}} \tag{7.122}$$

$$= \frac{(K - mbM)\,\mathbb{H}(\boldsymbol{p})}{K\boldsymbol{w}^T\boldsymbol{p}} \tag{7.123}$$

$$= \frac{\mathbb{H}(\boldsymbol{p}) - \frac{mbM}{K}\,\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}} \tag{7.124}$$

$$= \frac{\mathbb{H}(\boldsymbol{p}) - \frac{b\log(2)M}{K}}{\boldsymbol{w}^T\boldsymbol{p}} \tag{7.125}$$

$$= \frac{\mathbb{H}(\boldsymbol{p}) - b\log(2)\frac{M+K-K}{K}}{\boldsymbol{w}^T\boldsymbol{p}} \tag{7.126}$$

$$= \frac{\mathbb{H}(\boldsymbol{p}) - b\log(2)(\frac{1}{c} - 1)}{\boldsymbol{w}^T\boldsymbol{p}} \tag{7.127}$$

where we used (7.118) in the fourth line. We calculate the ideal coding rate by

$$\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p}, c) = \mathbb{I}_{\mathrm{bs}}(\boldsymbol{p}, c) \tag{7.128}$$

$$\Leftrightarrow \frac{\mathbb{H}(\boldsymbol{p}) - b\log(2)(\frac{1}{c} - 1)}{\boldsymbol{w}^T\boldsymbol{p}} = \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}} \tag{7.129}$$

$$\Leftrightarrow \mathbb{H}(\boldsymbol{p}) - b\log(2)\Big(\frac{1}{c} - 1\Big) = \mathbb{I}(\boldsymbol{p}) \tag{7.130}$$

$$\Leftrightarrow c = \frac{1}{1 + \frac{1}{b\log(2)}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]}. \tag{7.131}$$

Thus, the ideal coding rate is given by

$$c^*(\boldsymbol{p}) = \frac{1}{1 + \frac{1}{b\log(2)}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]} \tag{7.132}$$

and the ideal transmission rate is given by

$$\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p}) = \mathbb{R}_{\mathrm{bs}}[\boldsymbol{p}, c^*(\boldsymbol{p})] \tag{7.133}$$

$$= \mathbb{I}_{\mathrm{bs}}[\boldsymbol{p}, c^*(\boldsymbol{p})] \tag{7.134}$$

$$= \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}}. \tag{7.135}$$

Matched capacity is given by the maximum rate that can be achieved by an ideal systematic block code, i.e.,

$$\mathsf{C}_{\mathrm{bs}} = \max_{\boldsymbol{p}} \mathbb{R}_{\mathrm{bs}}(\boldsymbol{p}) \tag{7.136}$$

$$= \max_{\boldsymbol{p}} \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}} \tag{7.137}$$

$$= \mathsf{C}. \tag{7.138}$$

Thus, matched capacity is equal to capacity.

7.4.3 Matching

The matching problem consists in maximizing $\mathbb{I}(\boldsymbol{d})/(\boldsymbol{w}^T\boldsymbol{d})$ where $\boldsymbol{d}$ is a dyadic pmf. This problem is solved in Section 4.3. The dyadic pmf $\boldsymbol{d} = \textsc{nGhc}(\boldsymbol{p}^*, \boldsymbol{w})$ maximizes a lower bound on the achieved mutual information per average duration and $\boldsymbol{d}_k = \textsc{nGhc}(\boldsymbol{p}^{*k}, \boldsymbol{w}^{\oplus k})$ achieves capacity for $k \to \infty$.

7.4.4 Matched gains

Shaping gain

The shaping gain of matched transmission is

$$\frac{\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p})}{\mathsf{C}} = \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}} \cdot \frac{1}{\mathsf{C}}. \tag{7.139}$$

As we showed in the previous subsection, for matched transmission, dyadic pmfs are asymptotically capacity-achieving. This implies that prefix-free matchers can achieve a shaping gain of one.

Coding gain

The coding gain of matched transmission is

$$\frac{\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p}, c)}{\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p})} = \frac{\mathbb{H}(\boldsymbol{p}) - b\log(2)(\frac{1}{c} - 1)}{\boldsymbol{w}^T\boldsymbol{p}} \cdot \frac{\boldsymbol{w}^T\boldsymbol{p}}{\mathbb{I}(\boldsymbol{p})} \tag{7.140}$$

$$= \frac{\mathbb{H}(\boldsymbol{p}) - b\log(2)(\frac{1}{c} - 1)}{\mathbb{I}(\boldsymbol{p})}. \tag{7.141}$$

Note that the average cost cancels out.

Overall gain

For the overall gain of matched transmission, we get

$$\frac{\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p})}{\mathsf{C}} \cdot \frac{\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p}, c)}{\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p})} = \frac{\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p}, c)}{\mathsf{C}} \tag{7.142}$$

$$= \frac{1}{\mathsf{C}} \frac{\mathbb{H}(\boldsymbol{p}) - b\log(2)(\frac{1}{c} - 1)}{\boldsymbol{w}^T\boldsymbol{p}}. \tag{7.143}$$

7.5 References

Parts of this chapter were published in [11].

The combination of a systematic block code with a prefix-free matcher was considered by Vasić *et al* [74]. Related to the combination of error correcting codes and matchers are the *MacKay-Neal* (MN) *codes* proposed by MacKay in [55, Section VI]. For MN codes, arithmetic coding is used to generate a sequence of bits distributed according to $(p,\ 1-p)^T$. The same MN code can then be used to reliably transmit over bsc with different crossover probability ϵ by adapting p.

Gallager invented ldpc codes in [38]. Ldpc codes can be put into the form of systematic block codes. Non-binary ldpc codes were for example designed by Davey and MacKay [27] and Declercq and Fossorier [28].

Sparse-dense codes were introduced by Ratzer [65, Section 5.3] for binary *crosstalk channels* where the crossover probability depends on the frequency of transmitted `1`s. Ratzer used both arithmetic coding and Huffman source coding as matchers [65, Section 5.2]. We use the techniques introduced in Section 7.3.2 in [13] to calculate *bit-interleaved coded modulation* (BICM) capacity.

Another approach to achieve non-uniform pmfs on the channel is to do error-correction encoding, matching, and jointly dematching and decoding in this order. Gallager proposed this approach in [39, Page 208] and it is nicely explained in McEliece [61, Section 5].

Jiang and Narayanan [46] use *multilevel coding* to achieve non-uniform pmfs on the channel. Yet another approach is presented by Ratzer and MacKay in [64]. All approaches have in common that decoding and dematching cannot be performed separately. This stands in contrast to our proposed schemes.

8 Case study: error-correction for a bsc with unequal symbol durations

We consider a *binary symmetric channel* (bsc) with unequal symbol durations. The crossover probability is ϵ, i.e., the transition matrix is given by

$$\boldsymbol{H} = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}. \tag{8.1}$$

The durations of the binary input symbols `0` and `1` are given by

$$w(\mathtt{0}) = 1, \quad w(\mathtt{1}) = 5 \tag{8.2}$$

i.e., in vector form, $\boldsymbol{w} = (1, 5)^T$. We vary the crossover probability ϵ between 0.00725 and 0.05775, see Table 8.2. We evaluate the performance around block-error rates of 10^{-2}. The reason of this value is the following. Assume a one-bit feedback from the receiver that tells the transmitter for each block if the block could be decoded correctly or not. Then a block needs to be transmitted in the average $1/(1-10^{-2})$ times until error-free reception. This corresponds to a reduction of the rate by one percent and is thus negligible. For each considered setup, we transmit 5000 blocks.

8.1 Setup

8.1.1 Systematic block codes

As systematic block codes, we use the *low-density parity-check* (ldpc) codes as used in the DVB-S2 standard. We consider codes with the rates 8/9, 5/6, 4/5, and 3/4. The blocklength is for all code rates given by $N = 64800$. Thus, the number of information bits K and the number of check bits M can easily be calculated. For example, for the coding rate $c = 3/4$, we have

$$K = cN = 3/4 \cdot 64800 = 48600, \qquad M = N - K = 16200. \tag{8.3}$$

The DVB-S2 codes are systematic and amenable for simulation, since an implementation is readily available in the Communications System Toolbox of MATLAB. It is important to note that for uniform, sparse-dense, and matched transmission, we use exactly the same code, encoders, and decoders, namely the unaltered implementation in MATLAB. The differences between uniform, sparse-dense, and matched transmission are detailed in the next section.

$$
\begin{array}{cccc}
00 \mapsto \texttt{0000} & 010 \mapsto \texttt{0001} & 011 \mapsto \texttt{0010} & 10111 \mapsto \texttt{0011} \\
100 \mapsto \texttt{0100} & 10110 \mapsto \texttt{0101} & 10100 \mapsto \texttt{0110} & 1010111 \mapsto \texttt{0111} \\
110 \mapsto \texttt{1000} & 11101 \mapsto \texttt{1001} & 11100 \mapsto \texttt{1010} & 101011 \mapsto \texttt{1011} \\
11110 \mapsto \texttt{1100} & 111110 \mapsto \texttt{1101} & 111111 \mapsto \texttt{1110} & 1010100 \mapsto \texttt{1111}
\end{array}
$$

$$
\boldsymbol{d}_4 = (2^{-2}\ \ 2^{-3}\ \ 2^{-3}\ \ 2^{-5}\ \ 2^{-3}\ \ 2^{-5}\ \ 2^{-5}\ \ 2^{-7}\ \ 2^{-3}\ \ 2^{-5}\ \ 2^{-5}\ \ 2^{-6}\ \ 2^{-5}\ \ 2^{-6}\ \ 2^{-6}\ \ 2^{-7})^T
$$

Table 8.1: The vector $\boldsymbol{d}_4$ is the dyadic pmf for blocks of $k = 4$ symbols for sparse-dense and matched transmission. The table above displays a prefix-free matcher that generates $\boldsymbol{d}_4$.

8.1.2 Prefix-free matcher

We first discuss the prefix-free matcher used for sparse-dense and matched transmission. The crossover probabilities ϵ that we evaluate for sparse-dense and matched transmission are listed in Table 8.2. To generate dyadic pmfs, we jointly consider blocks of $k = 4$ symbols on the channel. For sparse-dense transmission, we first use Algorithm 7 to calculate the pmf $\boldsymbol{p}^{\mathrm{sd}}$ that achieves sparse-dense capacity. We then calculate the dyadic pmf according to Proposition 7.4 as follows. We define $\boldsymbol{\ell} = (-\log_2 p_1^{\mathrm{sd}},\ -\log_2 p_2^{\mathrm{sd}})^T$ and define

$$
\boldsymbol{d}_4^{\mathrm{sd}} = \textsc{nGhc}(\mathbf{1}, \oplus^4 \boldsymbol{\ell}) \tag{8.4}
$$

where $\mathbf{1}$ denotes an all-one column vector with $2^4 = 16$ entries and where $\oplus^4 \boldsymbol{\ell}$ denotes the cost sum of 4 copies of $\boldsymbol{\ell}$. For matched transmission, according to Subsection 7.4.2, the pmf that achieves matched capacity is equal to the capacity-achieving pmf $\boldsymbol{p}^*$. We calculate $\boldsymbol{p}^*$ by Algorithm 4. For example, for $\epsilon = 0.028$, $\boldsymbol{p}^*$ is given by $(0.7349,\ 0.2651)^T$. We then calculate the dyadic pmf according to Subsection 7.4.3 by

$$
\boldsymbol{d}_4^{\mathrm{bs}} = \textsc{nGhc}(\boldsymbol{p}^{*4}, \oplus^4 \boldsymbol{w}) \tag{8.5}
$$

where $\boldsymbol{p}^{*4}$ denotes the Kronecker product of 4 copies of $\boldsymbol{p}^*$ and where $\oplus^4 \boldsymbol{w}$ denotes the cost sum of 4 copies of $\boldsymbol{w}$. For all considered values of ϵ, $\boldsymbol{d}_4^{\mathrm{sd}} = \boldsymbol{d}_4^{\mathrm{bs}}$, i.e., in the considered setup, sparse-dense and matched transmission use the same dyadic pmf. We denote it by $\boldsymbol{d}_4$. Furthermore, $\boldsymbol{d}_4$ remains unchanged over the whole range of the considered ϵ. The dyadic pmf $\boldsymbol{d}_4$ and a prefix-free matcher that generates it are displayed in Table 8.1.

8.1.3 Effective transmission rate

For calculating the coding gain of the considered transmission schemes, we use the effective transmission rate. The effective transmission rate of one block is given by

$$
\hat{\mathbb{R}} = \frac{\text{information}}{\text{block duration}}. \tag{8.6}
$$

Since we assume that the data to be transmitted comes in form of a fair bit stream, the information in one block is in bits given by the number of data bits that were mapped

to the transmitted block, and in nats, the information is

$$\text{information} = \sharp\{\text{data bits mapped to the block}\} \cdot \log 2. \tag{8.7}$$

The duration of one block is given by

$$\text{block duration} = \sharp\{\texttt{0}\text{s in the block}\} \cdot 1 + \sharp\{\texttt{1}\text{s in the block}\} \cdot 5. \tag{8.8}$$

Thus, the effective transmission rate is

$$\hat{\mathbb{R}} = \frac{\sharp\{\text{data bits mapped to the block}\} \cdot \log 2}{\sharp\{\texttt{0}\text{s in the block}\} \cdot 1 + \sharp\{\texttt{1}\text{s in the block}\} \cdot 5}. \tag{8.9}$$

For all three transmission schemes, the block duration is random, and while the number of data bits mapped to one block is constant for uniform transmission, it is random for sparse-dense and matched transmission. We evaluated the variation of the effective transmission rate $\hat{\mathbb{R}}$ for 5000 transmissions. The statistics show that for each considered configuration, $\hat{\mathbb{R}}$ can be modelled as Gaussian distributed with the mean given by the expected transmission rate $\mathbb{R}_{\text{scheme}}(\boldsymbol{p}, c)$ as derived for each scheme in Chapter 7 and a very small variance. We therefore use in our evaluations for each configuration the mean $\bar{\mathbb{R}}$ of the 5000 realizations of the effective transmission rate $\hat{\mathbb{R}}$, but because the variance is vanishingly small, we omit to provide confidence intervals.

8.2 Transmission schemes

We now detail how we operate the ldpc codes for the different transmission schemes.

8.2.1 Uniform transmission

Matching

K iid equiprobable data bits are mapped one-to-one to K binary channel symbols. The ldpc encoder is applied to these K channel symbols and M binary check symbols are generated and appended to the K channel symbols. No matching is performed.

Transmission

All $K + M$ binary channel symbols are transmitted over a bsc with crossover probability ϵ.

Decoder parameters

Since we have uniform priors, we pass to the decoder for each of the $N = K + M$ received binary symbols the *log-likelihood ratios* (llrs)

$$\text{llr}(0) = \log \frac{(1-\epsilon)}{\epsilon}, \text{ if received symbol} = \texttt{0} \tag{8.10}$$

$$\text{llr}(1) = \log \frac{\epsilon}{(1-\epsilon)}, \text{ if received symbol} = \texttt{1}. \tag{8.11}$$

Evaluation

For each ϵ, the shaping gain is according to Subsection 7.2.2 calculated as

$$\frac{\mathbb{R}_{\mathrm{u}}}{\mathsf{C}} = \frac{\mathbb{I}(\boldsymbol{u})}{\boldsymbol{w}^T\boldsymbol{u}}\frac{1}{\mathsf{C}} \tag{8.12}$$

where $\boldsymbol{w} = (1,\ 5)^T$, $\boldsymbol{u} = (1/2,\ 1/2)^T$ and where C is calculated by Algorithm 4. The coding gain is calculated as

$$\frac{\bar{\mathbb{R}}}{\mathbb{R}_{\mathrm{u}}} = \bar{\mathbb{R}}\frac{\boldsymbol{w}^T\boldsymbol{u}}{\mathbb{I}(\boldsymbol{u})} \tag{8.13}$$

where $\bar{\mathbb{R}}$ is the mean of the observed effective transmission rates. Note that the overall gain is given by

$$\frac{\mathbb{R}_{\mathrm{u}}}{\mathsf{C}} \cdot \frac{\bar{\mathbb{R}}}{\mathbb{R}_{\mathrm{u}}} = \frac{\bar{\mathbb{R}}}{\mathsf{C}} \tag{8.14}$$

that is, the overall gain only depends on capacity and effectively observed values.

8.2.2 Sparse-dense transmission

Matching

K matched binary channel symbols are generated by parsing the fair data bit stream by the prefix-free matcher displayed in Table 8.1. The ldpc encoder is applied to these K binary channel symbols and M unmatched binary check symbols are generated and appended to the K matched symbols.

Transmission

All $K+M$ binary channel symbols are transmitted over a bsc with crossover probability ϵ.

Decoding

The decoder assumes that the matcher is perfect, i.e., that it generates matched binary symbols that are iid according to the pmf $\boldsymbol{p}^{\mathrm{sd}}$ that achieves sparse-dense capacity. We calculate $\boldsymbol{p}^{\mathrm{sd}}$ by Algorithm 7. For the matched K symbols, we pass to the decoder

$$\mathrm{llr}(0) + \log\frac{p_1^{\mathrm{sd}}}{p_2^{\mathrm{sd}}} = \log\frac{(1-\epsilon)p_1^{\mathrm{sd}}}{\epsilon p_2^{\mathrm{sd}}},\ \text{if received symbol} = \mathtt{0} \tag{8.15}$$

$$\mathrm{llr}(1) + \log\frac{p_1^{\mathrm{sd}}}{p_2^{\mathrm{sd}}} = \log\frac{\epsilon p_1^{\mathrm{sd}}}{(1-\epsilon)p_2^{\mathrm{sd}}},\ \text{if received symbol} = \mathtt{1}. \tag{8.16}$$

For the unmatched M check symbols, we pass to the decoder

$$\mathrm{llr}(0) = \log\frac{(1-\epsilon)}{\epsilon},\ \text{if received symbol} = \mathtt{0} \tag{8.17}$$

$$\mathrm{llr}(1) = \log\frac{\epsilon}{(1-\epsilon)},\ \text{if received symbol} = \mathtt{1}. \tag{8.18}$$

Evaluation

The ideal transmission rate of sparse-dense transmission is according to Proposition 7.1 given by

$$\mathbb{R}_{\mathrm{sd}}(\boldsymbol{p}) = \frac{\mathbb{H}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p} + \frac{1}{\mathsf{C}_{\mathrm{u}}}[\mathbb{H}(\boldsymbol{p}) - \mathbb{I}(\boldsymbol{p})]} \tag{8.19}$$

where $\boldsymbol{p}$ is the pmf generated by the matcher. Since our matcher jointly generated blocks of 4 symbols according to the dyadic pmf $\boldsymbol{d}_4$, see Table 8.1, we normalize entropy, mutual information, and average duration by 4. The ideal transmission rate for $\boldsymbol{d}_4$ is thus given by

$$\mathbb{R}_{\mathrm{sd}}(\boldsymbol{d}_4) = \frac{\frac{\mathbb{H}(\boldsymbol{d}_4)}{4}}{\frac{\boldsymbol{v}_4^T\boldsymbol{d}_4}{4} + \frac{1}{\mathsf{C}_{\mathrm{u}}}\left[\frac{\mathbb{H}(\boldsymbol{d}_4)}{4} - \frac{\mathbb{I}(\boldsymbol{d}_4)}{4}\right]} \tag{8.20}$$

where $\boldsymbol{v}_4 = \oplus^4\boldsymbol{w}$ is the cost sum of 4 copies of $\boldsymbol{w}$. The shaping gain is now calculated as

$$\frac{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{d}_4)}{\mathsf{C}} \tag{8.21}$$

and the coding gain is calculated as

$$\frac{\bar{\mathbb{R}}}{\mathbb{R}_{\mathrm{sd}}(\boldsymbol{d}_4)}. \tag{8.22}$$

8.2.3 Matched Transmission

To allow comparison between the schemes, we evaluate matched transmission by simulating its building block as displayed in Figure 7.4.

Matching

The matcher generates K matched binary channel symbols by parsing a fair bit stream. The ldpc encoder generates M unmatched binary check symbols, but they are not appended to the K matched symbols.

Transmission

The K matched symbols are transmitted over a bsc with crossover probability ϵ. The M check symbols remain unchanged.

Decoder

The decoder assumes a perfect matcher, i.e., it assumes that the received matched symbols are iid according to the pmf that achieves matched capacity. According to Subsection 7.4.2, matched capacity is equal to capacity, and the same holds for the corresponding pmfs. Thus, the decoder assumes that the matched symbols are iid according to $\boldsymbol{p}^*$,

which we calculate by Algorithm 4. For the K matched symbols, we pass to the decoder

$$\begin{aligned}\mathrm{llr}(0) + \log\frac{p_1^*}{p_2^*} &= \log\frac{(1-\epsilon)p_1^*}{\epsilon p_2^*}, \text{ if received symbol} = \mathtt{0}\\ \mathrm{llr}(1) + \log\frac{p_1^*}{p_2^*} &= \log\frac{\epsilon p_1^*}{(1-\epsilon)p_2^*}, \text{ if received symbol} = \mathtt{1}.\end{aligned}$$

Since the check symbols are perfectly known, we pass

$$\begin{aligned}\infty, &\text{ if check symbol} = \mathtt{0}\\ -\infty, &\text{ if check symbol} = \mathtt{1}.\end{aligned}$$

Evaluation

According to Subsection 7.4.4, the ideal transmission rate for matched transmission is given by

$$\mathbb{R}_{\mathrm{bs}}(\boldsymbol{p}) = \frac{\mathbb{I}(\boldsymbol{p})}{\boldsymbol{w}^T\boldsymbol{p}} \tag{8.23}$$

where $\boldsymbol{p}$ denotes the pmf generated by the matcher. Thus, the ideal transmission rate is given by

$$\mathbb{R}_{\mathrm{bs}}(\boldsymbol{d}_4) = \frac{\mathbb{I}(\boldsymbol{d}_4)}{\boldsymbol{v}_4^T\boldsymbol{d}_4}. \tag{8.24}$$

We can now calculate the shaping gain as

$$\frac{\mathbb{R}_{\mathrm{bs}}(\boldsymbol{d}_4)}{\mathsf{C}}. \tag{8.25}$$

To calculate the effective transmission rate, we need to take into account that for matched transmission, some of the bits in the fair bit stream stem from check symbols, see Subsection 7.4.1. We know that the first M bits of the fair bit stream are check bits. Thus, the number of information bits mapped to the K matched symbols is given by the number of parsed bits minus M. Note that the number of information bits can vary from block to block. With the effective transmission rates calculated that way, the coding gain is given by

$$\frac{\bar{\mathbb{R}}}{\mathbb{R}_{\mathrm{bs}}(\boldsymbol{d}_4)}. \tag{8.26}$$

8.3 Discussion

The numerical results are displayed in Figure 8.1. Shaping and coding gain are displayed in horizontal direction. For the coding gains, the corresponding block error probabilities are displayed in vertical direction. The error bars mark 95% confidence intervals, which

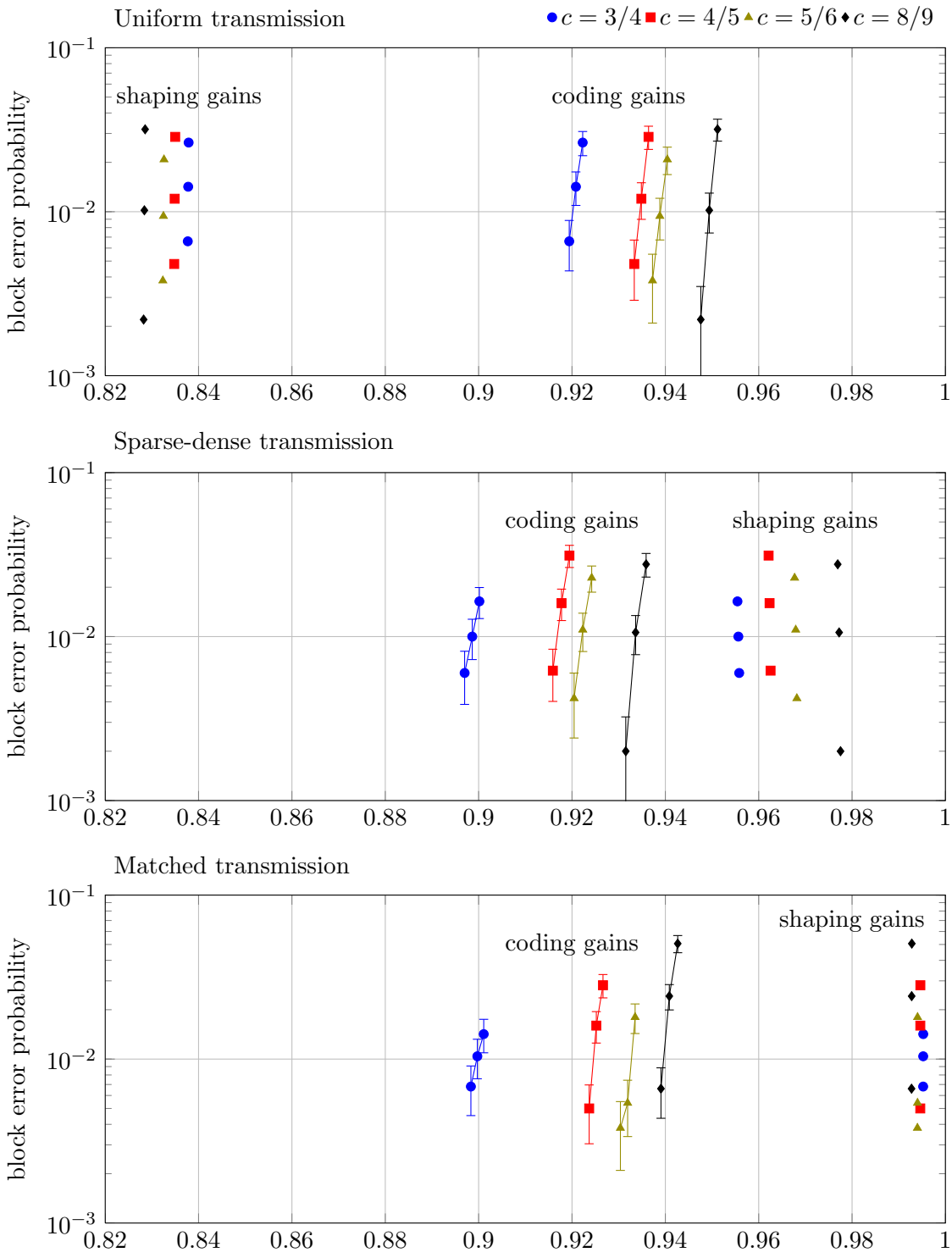


Figure 8.1: Coding and shaping gains.

Uniform transmission

$c = 3/4$	$c = 4/5$	$c = 5/6$	$c = 8/9$
0.02850	0.02075	0.01525	0.00775
0.02825	0.02050	0.01500	0.00750
0.02800	0.02025	0.01475	0.00725

Sparse-dense transmission

$c = 3/4$	$c = 4/5$	$c = 5/6$	$c = 8/9$
0.03250	0.02400	0.01775	0.00900
0.03225	0.02375	0.01750	0.00875
0.03200	0.02350	0.01725	0.00850

Matched transmission

$c = 3/4$	$c = 4/5$	$c = 5/6$	$c = 8/9$
0.05775	0.03750	0.02600	0.01150
0.05750	0.03725	0.02575	0.01125
0.05725	0.03700	0.02550	0.01100

Table 8.2: Values for the channel parameter ϵ. The locations of the values in the table correspond to the location of the corresponding numerical result in Figure 8.1.

were calculated according to [32, Section 9.4.2]. To facilitate the orientation in the graph, the shaping gains are displayed at the same vertical position as the corresponding coding gains. The upper plot displays the results for uniform transmission, the plot in the middle shows the results for sparse-dense transmission, and the lower plot displays the results for matched transmission. As we can see, for uniform transmission, the shaping gains are smaller than the coding gains, i.e., for the considered channel, the bottleneck is shaping rather than coding. For sparse-dense transmission, the picture changes. Now the coding gains are smaller than the shaping gains and the bottleneck is now coding rather than shaping. The huge improvement of the shaping gain compared to uniform transmission is because the considered codes are high-rate codes, i.e., $K \gg M$. Since sparse-dense transmission can match the K information symbols but not the M check symbols, and since $K \gg M$, most of the transmitted symbols are matched. As the code rate decreases, so does the shaping gain: we observe the greatest shaping gain for code rate 9/8 and the smallest shaping gain for code rate 3/4. For low-rate codes, we can expect that the shaping gain remains the bottleneck even when using sparse-dense transmission. For matched transmission, the situation is different. Here, we have a shaping gain of almost 1 and this remains unchanged for all considered coding rates. We would observe the same shaping gains for low-rate codes, so with matched transmission, the coding gain is always the bottleneck.

The coding gain changes only slightly for the different transmission schemes although each scheme operates the codes in a different way. This is in accordance with the general

observation that ldpc codes have universal properties, see for example [37].

9 Conclusions

In Chapter 3, 4, and 5, we developed algorithms to find prefix-free matchers that minimize relative entropy, normalized relative entropy, and relative entropy subject to a cost constraint, respectively. In the first two cases, we proved optimality and in all three cases, we proved asymptotic achievability. These algorithms solve the prefix-free matching problem and form thereby a counterpart to Huffman coding, which solves the prefix-free source coding problem. We showed that our algorithms can directly be used to maximize the entropy rate for noiseless channels. For dmcs, with the capacity-achieving pmf as an additional parameter, the algorithms can be used to maximize mutual information. For this part of our work, two main problems remain open.

- As we have illustrated at two points in this work, to use the Huffman source code of the capacity-achieving pmf as a prefix-free matcher is sub-optimal. However, in many practical examples, the Huffman source code is a good prefix-free matcher, and in some cases, it is even equal to the optimal prefix-free matcher obtained by our algorithms. The conjecture is that Huffman source codes are asymptotically capacity-achieving prefix-free matchers and vice versa, that GHC asymptotically achieves the maximum compression ratio when used as a source code. To prove this conjecture is an open problem.

- Our algorithm CCGHC not necessarily finds the optimal prefix-free matcher subject to a cost constraint. The problem is that CCGHC only finds points on the convex hull of all points that can be achieved by prefix-free matchers. It is a challenging question how an algorithm could be designed that finds optimal points that lie inside of the convex hull.

In Chapter 6, we established the fundamental relation between combinatorial and probabilistic capacity of general noiseless channels. We then considered finite state channels and showed how memoryless codes can be constructed and we showed that these codes are asymptotically capacity-achieving. In all examples that we considered, our codes achieve a higher entropy rate than existing techniques. Further research could be performed in the following directions.

- Our results for finite state channels on combinatorial capacity, maximum entropy rate, and coding could be extended to more general settings, including e.g. context-free grammars.

- Our VLM codes are not only variable length but also variable rate. In contrast, most existing techniques construct fixed-rate codes for noiseless channels. The advantages and disadvantages of VLM in more practical settings could be assessed.

In the Chapter 7 and 8, we considered how prefix-free matchers can be combined with systematic block codes. We introduced the concept of shaping and coding gain. For dmcs with unequal symbol duration, we showed how the shaping gain of systematic block codes can be improved by sparse-dense transmission. By applying a prefix-free matcher both to data symbols and to check symbols, we showed that the shaping gain can be made equal to one. These are possible directions for future research:

- Any existing systematic block code can be operated by our schemes and decoding and dematching is performed sequentially. Existing approaches require to jointly perform decoding and dematching, see for example [39, page 208], [61, Section 5], [64], [46]. It would be interesting to compare the different schemes in terms of achieved shaping gain, coding gain, delay, and system complexity.

- For low rate codes, the degree of the check symbols can be very low, i.e., they are calculated as the sum of very few data symbols. In this situation, our uniform check symbol assumption becomes questionable and our results for capacity, shaping and coding gain, and matching have to be rethought. The work by Vasić *et al* [74] may serve as a starting point.

- Our results could be extended to channels with average cost constraints and to channels with crosstalk [65, Section 4.2].

- Our schemes may be useful for reliable communication with a shaping gain of one over additive noise channels in the bandwidth-limited regime. To this end, systematic block codes over $\mathbf{Z}_n$ with n equal to the number of signalling points could be considered.

An open question is the following.

- Does Algorithm 7 always find sparse-dense capacity? More specifically, are there channels where the mutual information $\mathbb{I}_{\text{sd}}$ in (7.53) has more than one local maximum?

Bibliography

[1] J. Abrahams, "Code and parse trees for lossless source encoding," in *Proc. Compression and Complexity of Sequences 1997*, 1997, pp. 145–171.

[2] ——, "Variable-length unequal cost parsing and coding for shaping," *IEEE Trans. Inf. Theory*, vol. 44, no. 4, pp. 1648–1650, 1998.

[3] ——, "Correspondence between variable length parsing and coding," in *The mathematics of information coding, extraction and distribution*, G. Cybenko, D. P. O'Leary, and J. Rissanen, Eds. Springer, 1999, ch. 1, pp. 1–7.

[4] F. Altenbach, G. Böcherer, and R. Mathar, "Short Huffman codes producing 1s half of the time," in *Proc. Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, 2011.

[5] S. Arimoto, "An algorithm for computing the capacity of arbitrary discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 14–20, 1972.

[6] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 460–473, 1972.

[7] M. Blaum, R. D. Cideciyan, E. Eleftheriou, R. Galbraith, K. Lakovic, T. Mittelholzer, T. Oenning, and B. Wilson, "High-rate modulation codes for reverse concatenation," *IEEE Trans. Magn.*, vol. 43, no. 2, pp. 740–743, 2007.

[8] G. Böcherer, "Geometric Huffman coding," http://www.georg-boecherer.de/ghc, Dec. 2010.

[9] G. Böcherer, F. Altenbach, M. Malsbender, and R. Mathar, "Writing on the facade of RWTH ICT Cubes: Cost constrained geometric Huffman coding," in *Proc. IEEE Int. Symp. Wireless Commun. Syst. (ISWCS)*, 2011.

[10] G. Böcherer, F. Altenbach, and R. Mathar, "Capacity achieving modulation for fixed constellations with average power constraint," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2011.

[11] G. Böcherer and R. Mathar, "Operating LDPC codes with zero shaping gap," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2011.

[12] G. Böcherer, "Analytic asymptotics of discrete noiseless channels," Master's thesis, ETH Zurich, 2007. [Online]. Available: http://www.georg-boecherer.de/repository/analyticAsymptotics.pdf

[13] G. Böcherer, F. Altenbach, A. Alvarado, S. Corroy, and R. Mathar, "An efficient algorithm to calculate BICM capacity," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2012.

[14] G. Böcherer, V. C. da Rocha Jr., C. Pimentel, and R. Mathar, "Maximum entropy rate of Markov sources for systems with non-regular constraints," in *Proc. Int. Symp. Inf. Theory and its Applicat. (ISITA)*, 2008.

[15] G. Böcherer, V. C. da Rocha Jr., and C. Pimentel, "Capacity of general discrete noiseless channels," in *Proc. Int. Symp. Commun. Applicat. (ISCTA)*, 2007.

[16] G. Böcherer, V. C. da Rocha Jr., C. Pimentel, and R. Mathar, "On the capacity of constrained systems," in *Proc. Int. ITG Conf. Source Channel Coding*, 2010.

[17] G. Böcherer and R. Mathar, "Matching dyadic distributions to channels," in *Proc. Data Compression Conf.*, 2011, pp. 23–32.

[18] S. Boyd, "Convex optimization II, lecture 14: Sequential convex programming," lecture notes, 2008. [Online]. Available: http://www.stanford.edu/class/ee364b/lectures/seq_slides.pdf

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[20] K. Burg, H. Haf, F. Wille, and A. Meister, *Höhere Mathematik für Ingenieure Band II: Lineare Algebra*. B. G. Teubner, Wiesbaden, 2007.

[21] N. Cai, S.-W. Ho, and R. Yeung, "Probabilistic capacity and optimal coding for asynchronous channel," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2007, pp. 54–59.

[22] N. Cai and R. W. Yeung, "Self-synchronizable codes for asynchronous communication," in *Proc. IEEE Int Information Theory Symp*, 2002.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.

[24] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., 2006.

[25] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Cambridge University Press, 2011.

[26] M. C. Davey, "Error-correction using low-density parity-check codes," Ph.D. dissertation, University of Cambridge, Dec. 1999.

[27] M. C. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165–167, 1998.

[28] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, 2007.

[29] D. Dubé and V. Beaudoin, "Constructing optimal whole-bit recycling codes," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Jun. 2009, pp. 27–31.

[30] C. H. Edwards, *Advanced Calculus of Several Variables*. Academic Press, 1973.

[31] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.

[32] L. Fahrmeir, R. Künstler, I. Pigeot, and G. Tutz, *Statistik: Der Weg zur Datenanalyse*. Springer-Verlag, 2004.

[33] R. F. H. Fischer, *Precoding and Signal Shaping for Digital Transmission*. John Wiley & Sons, Inc., 2002.

[34] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*. Cambridge University Press, 2008.

[35] J. Forney, G., R. Gallager, G. Lang, F. Longstaff, and S. Qureshi, "Efficient modulation for band-limited channels," *IEEE J. Sel. Areas Commun.*, vol. 2, no. 5, pp. 632–647, 1984.

[36] P. A. Franaszek, "Run-length limited variable length coding with error propagation limitation," US Patent 3689899, Sep. 1972.

[37] M. Franceschini, G. Ferrari, and R. Raheli, "Does the performance of LDPC codes depend on the channel?" *IEEE Trans. Commun.*, vol. 54, no. 12, pp. 2129–2132, 2006.

[38] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[39] ——, *Information Theory and Reliable Communication*. John Wiley & Sons, Inc., 1968.

[40] ——, *Principles of Digital Communication*. Cambridge University Press, 2008.

[41] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 1.21," http://cvxr.com/cvx, Jul. 2010.

[42] T. S. Han, *Information-Spectrum Methods in Information Theory*. Springer, 2003.

[43] ——, "Folklore in source coding: information-spectrum approach," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 747–753, 2005.

[44] G. H. Hardy and M. Riesz, *The General Theory of Dirichlet's Series*. Cambridge: at the University Press, 1915.

[45] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.

[46] J. Jiang and K. R. Narayanan, "Multilevel coding for channels with non-uniform inputs and rateless transmission over the BSC," in *Proc. IEEE Int Information Theory Symp*, 2006, pp. 518–522.

[47] M. Jimbo and K. Kunisawa, "An iteration method for calculating the relative capacity," *Inf. Contr.*, vol. 43, no. 2, pp. 216–223, Nov. 1979.

[48] K. J. Kerpez, "Runlength codes from source codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pp. 682–687, 1991.

[49] A. Khandekar, R. McEliece, and E. Rodemich, "The discrete noiseless channel revisited," in *Coding, Communications, and Broadcasting*. Research Studies Press Ltd., 2000, pp. 115–137.

[50] L. G. Kraft, "A device for quantizing, grouping, and coding amplitude modulated pulses," Master's thesis, Departement of Electrical Engineering MIT, 1949.

[51] R. M. Krause, "Channels which transmit letters of unequal duration," *Inf. Contr.*, vol. 5, pp. 3–24, 1962.

[52] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 913–929, 1993.

[53] A. Lempel, S. Even, and M. Cohn, "An algorithm for optimal prefix parsing of a noiseless and memoryless channel," *IEEE Trans. Inf. Theory*, vol. 19, no. 2, pp. 208–214, 1973.

[54] D. Lind and B. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1995.

[55] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[56] ——, "An alternative to runlength-limited codes: Turn timing errors into substitution errors," Available at www.inference.phy.cam.ac.uk/mackay, Sep. 2000.

[57] ——, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2004.

[58] E. E. Majani and H. Rumsey, "Two results on binary-input discrete memoryless channels," in *Proc. IEEE Int Information Theory (papers in summary form ly received) Symp. (Cat. No.91CH3003-1)*, 1991.

[59] B. H. Marcus, R. M. Roth, and P. H. Siegel, "An introduction to coding for constrained systems," Oct. 2001.

[60] R. S. Marcus, "Discrete noiseless coding," Master's thesis, MIT, 1957.

[61] R. McEliece, "Are turbo-like codes effective on nonstandard channels?" *IEEE Inf. Theo. Society Newsletter*, vol. 51, no. 4, pp. 1,3–8, Dec. 2001. [Online]. Available: http://backup.itsoc.org/publications/nltr/01_dec/dec01.pdf

[62] M. Mitzenmacher. (2008) A survey of results for deletion channels and related synchronization channels. [Online]. Available: http://www.eecs.harvard.edu/~michaelm/postscripts/DelSurvey.pdf

[63] J. R. Munkres, *Analysis on Manifolds*. Addison-Wesley Publishing Company, 1991.

[64] E. A. Ratzer and D. J. C. MacKay, "Sparse low-density parity-check codes for channels with cross-talk," in *Proc. IEEE Information Theory Workshop*, 2003, pp. 127–130.

[65] E. A. Ratzer, "Error-correction on non-standard communication channels," Ph.D. dissertation, University of Cambridge, 2003.

[66] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[67] A. Salomaa, *Jewels of Formal Language Theory*. Computer Science Press, Inc., 1981.

[68] A. Sardinas and G. W. Patterson, "A necessary and sufficient condition for the unique decomposition of coded messages," in *Convention Record of the I.R.E.*, 1953.

[69] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and 623–656, Jul. and Oct. 1948.

[70] M. Sipser, *Introduction to the Theory of Computation*, 2nd ed. Thomson Course Technology, 2006.

[71] P. R. Stubley and I. F. Blake, "On a discrete probability distribution matching problem," Jun. 1991, preprint.

[72] F. Topsøe, "Basic concepts, identities and inequalities—the toolkit of information theory," *Entropy*, vol. 3, pp. 162–190, 2001.

[73] G. Ungerböck, "Huffman shaping," in *Codes, Graphs, and Systems*, R. Blahut and R. Koetter, Eds. Springer, 2002, ch. 17, pp. 299–313.

[74] B. Vasic, O. Milenkovic, and S. McLaughlin, "Scrambling for nonequiprobable signalling," *Electronics Letters*, vol. 32, no. 17, pp. 1551–1552, 1996.

[75] S. Verdu, "On channel capacity per unit cost," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 1019–1030, 1990.

[76] Y. Wu and S. Verdú, "The impact of constellation cardinality on Gaussian channel capacity," in *Proc. 48th Annual Allerton Conf. Communication, Control, and Computing (Allerton)*, 2010, pp. 620–628.

[77] R. Yeung, N. Cai, S.-W. Ho, and A. Wagner, "Reliable communication in the absence of a common clock," *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 700–712, 2009.

[78] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Computation*, vol. 15, pp. 915–936, 2003.

[79] E. Zehavi and J. K. Wolf, "On runlength codes," *IEEE Trans. Inf. Theory*, vol. 34, no. 1, pp. 45–54, 1988.

Index