Matching Dyadic Distributions to Channels

G. Böcherer and R. Mathar
Institute for Theoretical Information Technology
RWTH Aachen University, 52056 Aachen, Germany
Email: {boecherer,mathar}@ti.rwth-aachen.de

Abstract

Many communication channels with discrete input have non-uniform capacity achieving probability mass functions (PMF). By parsing a stream of independent and equiprobable bits according to a full prefix-free code, a modulator can generate dyadic PMFs at the channel input. In this work, we show that for discrete memoryless channels and for memoryless discrete noiseless channels, searching for good dyadic input PMFs is equivalent to minimizing the Kullback-Leibler distance between a dyadic PMF and a weighted version of the capacity achieving PMF. We define a new algorithm called Geometric Huffman Coding (GHC) and prove that GHC finds the optimal dyadic PMF in $\mathcal{O}(m \log m)$ steps where m is the number of input symbols of the considered channel. Furthermore, we prove that by generating dyadic PMFs of blocks of consecutive input symbols, GHC achieves capacity when the block length goes to infinity.

I. INTRODUCTION

For many communication channels, the ultimate rate for reliable data transmission is given by the maximum information per cost. For discrete memoryless channels (DMC) and for additive noise channels with finite input alphabet, the ultimate rate is the maximum mutual information between input and output per channel use. For memoryless discrete noiseless channels (DNC), the ultimate rate is the maximum entropy of the input per average weight. In both cases, the maximum is achieved by an input that is distributed according to a capacity achieving probability mass function (PMF). To use non-uniform input PMFs in a digital communication system, a modulator has to generate this PMF by mapping independent equiprobable data bits to the channel input symbols. One way to do this is to parse the data bits by a full prefix-free code and to map each codeword to an input symbol [1, Sec. VII]. PMFs that can be generated in this way are dyadic, i.e., the probability of each point is of the form $2^{-\ell}, \ell \in \mathbb{N}$. The capacity achieving PMFs are in general not dyadic, which raises two questions. First, what is an optimal dyadic PMF that maximizes information per cost, and second, if we jointly generate blocks of consecutive input symbols by a dyadic PMF, can we asymptotically achieve capacity by letting the block length go to infinity.

For noiseless channels, an efficient algorithm to find the optimal dyadic PMF that maximizes entropy per average weight was found in [2]. In general, a common approach in the literature is to use the dyadic PMF that results from the optimal *source code* of the capacity achieving PMF. Dyadic PMFs resulting from source codes are in general

This work has been supported by the UMIC Research Centre, RWTH Aachen University.

not optimal. For the (d, k) constrained noiseless channel, it was claimed in [3] that a source code *asymptotically* achieves capacity. To the best of our knowledge, for DMCs, there exist no results in the literature on optimality and asymptotic behavior of dyadic PMFs. In [1], [4], the authors use source codes for additive noise channels. While good numerical results are observed, optimality and asymptotic behavior are not assessed. In [5], input entropy per average weight is maximized for additive noise channels. This is in general not equivalent to the maximization of mutual information per channel use.

Denote the capacity achieving PMF of a channel by $\boldsymbol{p}^*$. In this work, we show for DMCs that minimizing the Kullback-Leibler distance (KL) $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{p}^*)$ over all dyadic PMFs $\boldsymbol{p}$ maximizes a lower bound on the achieved mutual information per channel use. For DNCs, we show that searching for the optimal dyadic input PMF is equivalent to minimizing the weighted KL-distance $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{p}^{*\mathsf{R}}) \triangleq \sum_i p_i \log(p_i/p_i^{*\mathsf{R}})$ over all dyadic PMFs $\boldsymbol{p}$. The value of R is given by the fraction of the channel capacity that is achievable by dyadic PMFs. We introduce an algorithm called Geometric Huffman Coding (GHC) and prove that GHC minimizes $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{x})$ over all dyadic PMFs $\boldsymbol{p}$, for any given vector $\boldsymbol{x}$ with non-negative entries. In particular, for $\boldsymbol{x} = \boldsymbol{p}^*$, GHC minimizes $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{p}^*)$ and for $\boldsymbol{x} = \boldsymbol{p}^{*\mathsf{R}}$, GHC minimizes $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{p}^{*\mathsf{R}})$. The complexity of GHC is $\mathcal{O}(m \log m)$, where m is the number of input symbols of the considered channel. Furthermore, we show that, to asymptotically achieve capacity for DMCs and DNCs, the normalized KL-distance $\mathrm{D}(\boldsymbol{p}^{(k)}\|\boldsymbol{p}^{(k)*})/k$ has to vanish for block length $k \to \infty$. This is achieved by GHC. Based on the present work, we show in [6] that for finite signal constellations with average power constraint, GHC achieves capacity. GHC is as handy as Huffman coding and an implementation of GHC in MATLAB is readily available at our website [7].

The remainder of this work is organized as follows. In Section II, we define GHC. In Section III, we show optimality and asymptotic optimality of GHC for DMCs. We show optimality and asymptotic optimality of GHC for DNCs in Section IV.

II. Geometric Huffman Coding

For a PMF $\boldsymbol{p}$ and a vector $\boldsymbol{x}$ with non-negative entries, the KL-distance is given by

$$\mathrm{D}(\boldsymbol{p}\|\boldsymbol{x}) = \sum_i p_i \log \frac{p_i}{x_i}. \tag{1}$$

Note that $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{x})$ can be equal to infinity. The dyadic PMF $\boldsymbol{p}$ that minimizes the KL-distance is directly given by the full prefix-free code that is constructed by the algorithm of the following proposition. A prefix-free code is full if it fulfills the Kraft inequality [8, Theorem 5.2.2] with equality.

Proposition 1. *Without loss of generality, we assume $x_1 \geq x_2 \geq \cdots \geq x_m$. The dyadic PMF $\boldsymbol{p}$ that minimizes $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{x})$ is obtained by constructing a Huffman tree with the updating rule*

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

Since it involves a geometric mean, we call this method Geometric Huffman Coding. *We write $\boldsymbol{p} = \mathrm{GHC}(\boldsymbol{x})$.*

Proof: The proof is given in the appendix. ■

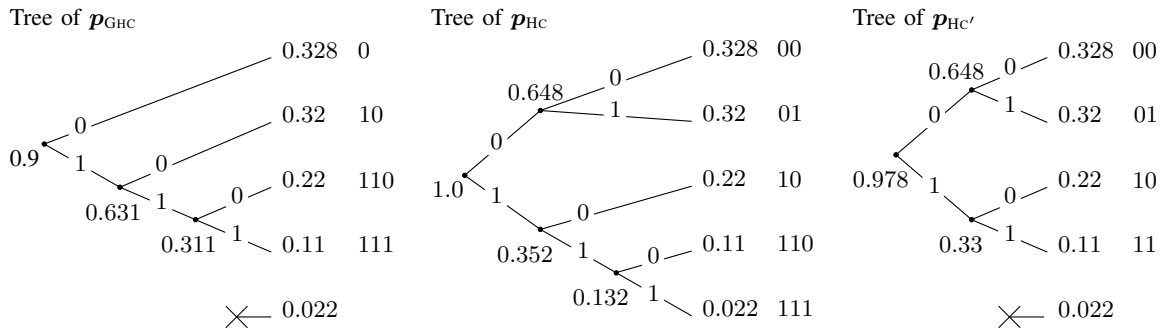

Fig. 1: For $\boldsymbol{q} = (0.328, 0.32, 0.22, 0.11, 0.022)^T$, the left figure displays the code tree of GHC. The figure in the middle shows the code tree of Huffman coding. The right figure displays the code tree of Huffman coding applied to $(q_1, \ldots, q_4)^T$.

An implementation of GHC in MATLAB can be found at our website [7]. In comparison to GHC, Huffman coding uses the updating rule $x' = x_m + x_{m-1}$. Furthermore, it can be shown that Huffman coding minimizes the KL-distance $\mathrm{D}(\boldsymbol{x}\|\boldsymbol{p})$ over all dyadic PMFs $\boldsymbol{p}$. Note that this is not equivalent to minimizing (1) because the KL-distance is not symmetric in its arguments. GHC has the same complexity as Huffman coding, which is $\mathcal{O}(m \log m)$ [9, Chap. 16.3].

For illustration purpose, we apply GHC and Huffman coding to the PMF

$$\boldsymbol{q} = (0.328, 0.32, 0.22, 0.11, 0.022)^T \tag{3}$$

where $(\cdot)^T$ denotes the transpose. The resulting code trees are displayed in Figure 1. By reading off the codeword lengths, the corresponding dyadic PMFs are

$$\boldsymbol{p}_{\text{GHC}} = (2^{-1}, 2^{-2}, 2^{-3}, 2^{-3}, 0)^T \text{ and } \boldsymbol{p}_{\text{HC}} = (2^{-2}, 2^{-2}, 2^{-2}, 2^{-3}, 2^{-3})^T \tag{4}$$

and the KL-distances to $\boldsymbol{q}$ are

$$\mathrm{D}(\boldsymbol{p}_{\text{GHC}}\|\boldsymbol{q}) = 0.13619 \text{ and } \mathrm{D}(\boldsymbol{p}_{\text{HC}}\|\boldsymbol{q}) = 0.19548 \tag{5}$$

where we used the dual logarithm. As expected, the KL-distance resulting from GHC is smaller than the one that results from Huffman coding. Since GHC assigns zero to q_5, one may want to manually assign probability zero to q_5 and then apply Huffman coding to $(q_1, \ldots, q_4)^T$. The corresponding code tree is displayed in Figure 1. The corresponding PMF and the resulting KL-distance to $\boldsymbol{q}$ are respectively

$$\boldsymbol{p}_{\text{HC}'} = (2^{-2}, 2^{-2}, 2^{-2}, 2^{-2}, 0)^T, \quad \mathrm{D}(\boldsymbol{p}_{\text{HC}'}\|\boldsymbol{q}) = 0.15523. \tag{6}$$

While $\boldsymbol{p}_{\text{HC}'}$ slightly improves upon $\boldsymbol{p}_{\text{HC}}$, the KL-distance is still larger than the one resulting from GHC.

Let $\boldsymbol{q}$ now denote some arbitrary PMF. We consider k subsequent symbols that are independent and identically distributed according to $\boldsymbol{q}$. We denote the joint PMF of these symbols by $\boldsymbol{q}^{(k)}$. Our aim is to show that for $\boldsymbol{p}^{(k)} = \text{GHC}(\boldsymbol{q}^{(k)})$, the normalized KL-distance $\mathrm{D}(\boldsymbol{p}^{(k)}\|\boldsymbol{q}^{(k)})/k$ vanishes for $k \to \infty$. To show this, we will need the following lemma, which shows the existence of dyadic PMFs with a bounded KL-distance for any PMF $\boldsymbol{q}$.

Lemma 1. *Without loss of generality, $q_1 \geq q_2 \geq \cdots \geq q_m$. Assign then $p_i = 2^{-\lfloor -\log_2 q_i \rfloor}$ for $i \leq k$, and $p_i = 0$ for $i > k$, where $k \leq m$ is chosen*[1] *such that $\sum_{i=1}^m p_i = 1$. Then*

[1]It can actually be shown that such k always exists, so GCC is well-defined.

$\boldsymbol{p}$ is a dyadic PMF and $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{q}) \leq \log 2$. We call this method Greedy Channel Coding *(*GCC*) and write $\boldsymbol{p} = \mathrm{GCC}(\boldsymbol{q})$.*

Proof:

$$\mathrm{D}(\boldsymbol{p}\|\boldsymbol{q}) = \sum_{i=1}^{m} p_i \log \frac{p_i}{q_i} \leq \sum_{i=1}^{m} p_i \log \frac{2^{-\lfloor -\log_2 q_i \rfloor}}{q_i} \tag{7}$$

$$\leq \sum_{i=1}^{m} p_i \log \frac{2^{-(-\log_2 q_i - 1)}}{q_i} \tag{8}$$

$$= \sum_{i=1}^{m} p_i \log \frac{2q_i}{q_i} = \log 2 \tag{9}$$

where the inequality in (7) follows from the values that GCC assigns to the p_i. ■

An implementation of GCC in MATLAB is available at [7]. It is now easy to show the asymptotic behavior of GHC.

Proposition 2. *For $\boldsymbol{p}^{(k)} = \mathrm{GHC}(\boldsymbol{q}^{(k)})$ it holds that*

$$\frac{\mathrm{D}(\boldsymbol{p}^{(k)}\|\boldsymbol{q}^{(k)})}{k} \xrightarrow{k\to\infty} 0. \tag{10}$$

Proof: Define $\tilde{\boldsymbol{p}}^{(k)} = \mathrm{GCC}(\boldsymbol{q}^{(k)})$. Then

$$\frac{\mathrm{D}(\boldsymbol{p}^{(k)}\|\boldsymbol{q}^{(k)})}{k} \leq \frac{\mathrm{D}(\tilde{\boldsymbol{p}}^{(k)}\|\boldsymbol{q}^{(k)})}{k} \leq \frac{\log 2}{k} \tag{11}$$

where the first inequality follows via Proposition 1 from the optimality of GHC and where the second inequality follows from Lemma 1. $\log 2/k$ goes to zero for $k \to \infty$ and the statement of the proposition follows. ■

III. Discrete Memoryless Channel

We now show how GHC can be used to find dyadic PMFs that well approximate the capacity of DMCs. A DMC is specified by a set of m input symbols, a set of n output symbols and a matrix of transition probabilities (h_{ji}). An input PMF $\boldsymbol{p}$ relates to its corresponding output PMF $\boldsymbol{r}$ as

$$\boldsymbol{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix} = \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{n1} & \cdots & h_{nm} \end{pmatrix} \begin{pmatrix} p_1 \\ \vdots \\ p_m \end{pmatrix}. \tag{12}$$

The mutual information between input and output is given by [10, Eq. (8.73)]

$$\mathcal{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j}. \tag{13}$$

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 optimization problem

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{maximize}} \quad & \mathcal{I}(\boldsymbol{p}) \\ \text{subject to} \quad & \boldsymbol{p} \text{ is a PMF} \\ & p_i = 2^{-\ell_i}, \ell_i \in \mathbb{N}, i = 1, \ldots, m. \end{aligned} \tag{14}$$

This is a nonlinear optimization problem with integer constraints and therefore intractable for practical purposes. In order to overcome this difficulty, we proceed as follows. First, we will drop the restriction to dyadic PMFs and characterize the capacity achieving PMF $\boldsymbol{p}^*$. Then, we will derive the penalty that results from using a PMF $\boldsymbol{p}$ different from $\boldsymbol{p}^*$. Finally, we will minimize this penalty over all dyadic PMFs.

Capacity and capacity achieving PMF are respectively defined as

$$\mathsf{C} = \max_{\boldsymbol{p}} \mathcal{I}(\boldsymbol{p}), \qquad \boldsymbol{p}^* = \operatorname*{argmax}_{\boldsymbol{p}} \mathcal{I}(\boldsymbol{p}). \tag{15}$$

Denote by $\boldsymbol{r}$ and $\boldsymbol{r}^*$ the output PMFs that result from using the input PMFs $\boldsymbol{p}$ and $\boldsymbol{p}^*$, respectively. According to [11, Eq. (4.5.1)], the output PMF $\boldsymbol{r}^*$ resulting from the capacity achieving PMF $\boldsymbol{p}^*$ has the important property that

$$\sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} = \mathsf{C}, \quad \text{whenever } p_i^* > 0. \tag{16}$$

We now use this property to express the mutual information $\mathcal{I}(\boldsymbol{p})$ achieved by some PMF $\boldsymbol{p}$ in terms of capacity C and capacity achieving PMF $\boldsymbol{p}^*$. The only assumption that we make about $\boldsymbol{p}$ is that

$$p_i = 0, \quad \text{whenever } p_i^* = 0. \tag{17}$$

Under this assumption, we have for $\mathcal{I}(\boldsymbol{p})$

$$\begin{aligned}
\mathcal{I}(\boldsymbol{p}) = \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j} &= \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji} r_j^*}{r_j r_j^*} && (18)\\
&= \sum_i p_i \sum_j h_{ji} \log \frac{h_{ji}}{r_j^*} + \sum_i p_i \sum_j h_{ji} \log \frac{r_j^*}{r_j} && (19)\\
&= \mathsf{C} - \sum_j \Big(\sum_i p_i h_{ji} \Big) \log \frac{r_j}{r_j^*} && (20)\\
&= \mathsf{C} - \sum_j r_j \log \frac{r_j}{r_j^*} && (21)\\
&= \mathsf{C} - \mathrm{D}(\boldsymbol{r} \| \boldsymbol{r}^*) && (22)
\end{aligned}$$

where equality in (20) follows from (16) and (17). From the last line, we see that the penalty of using $\boldsymbol{p}$ instead of $\boldsymbol{p}^*$ is exactly the KL-distance between the corresponding output PMFs $\boldsymbol{r}$ and $\boldsymbol{r}^*$. To get a simple expression that directly depends on $\boldsymbol{p}$ and $\boldsymbol{p}^*$, we lower bound the last line. According to [8, Eq. (4.45)] the KL-distance between the output PMFs is upper-bounded by the KL-distance between the input PMFs, i.e, $\mathrm{D}(\boldsymbol{r} \| \boldsymbol{r}^*) \leq \mathrm{D}(\boldsymbol{p} \| \boldsymbol{p}^*)$. Thus,

$$\mathcal{I}(\boldsymbol{p}) \geq \mathsf{C} - \mathrm{D}(\boldsymbol{p} \| \boldsymbol{p}^*). \tag{23}$$

We conclude that for DMCs, the penalty that results from using $\boldsymbol{p}$ instead of $\boldsymbol{p}^*$ is upper bounded by $\mathrm{D}(\boldsymbol{p} \| \boldsymbol{p}^*)$. According to Proposition 1, we can now efficiently minimize the penalty bound over all dyadic input PMFs $\boldsymbol{p}$ by using $\boldsymbol{p} = \textsc{Ghc}(\boldsymbol{p}^*)$. Note that $\textsc{Ghc}$ guarantees (17): assume $\boldsymbol{p}^*$ is ordered and $p_m^* = 0$, $p_{m-1}^* > 0$. Then $p_{m-1}^* > 4p_m^*$ and $\textsc{Ghc}$ assigns $p_m = 0$.

We now jointly consider the PMF of k consecutive channel inputs. Denote by $\boldsymbol{p}^{(k)*}$ the capacity achieving joint PMF. Since the channel is memoryless, $\boldsymbol{p}^{(k)*}$ is the product of k marginal PMFs $\boldsymbol{p}^*$ and we have $\mathcal{I}(\boldsymbol{p}^{(k)*}) = k\,\mathcal{I}(\boldsymbol{p}^*) = k\mathsf{C}$. Thus, for a joint PMF $\boldsymbol{p}^{(k)}$ we have

$$\mathcal{I}(\boldsymbol{p}^{(k)}) \geq k\mathsf{C} - \mathrm{D}(\boldsymbol{p}^{(k)} \| \boldsymbol{p}^{(k)*}). \tag{24}$$

The mutual information per channel use $\bar{\mathcal{I}}(\boldsymbol{p}^{(k)}) \triangleq \mathcal{I}(\boldsymbol{p}^{(k)})/k$ is thus given by

$$\bar{\mathcal{I}}(\boldsymbol{p}^{(k)}) \geq \mathsf{C} - \frac{\mathrm{D}(\boldsymbol{p}^{(k)} \| \boldsymbol{p}^{(k)*})}{k}. \tag{25}$$

By using $\boldsymbol{p}^{(k)} = \textsc{Ghc}(\boldsymbol{p}^{(k)*})$, according to Proposition 2, $\bar{\mathcal{I}}(\boldsymbol{p}^{(k)}) \to \mathsf{C}$ for $k \to \infty$ and we conclude that GHC is asymptotically capacity achieving.

IV. Memoryless Discrete Noiseless Channel

Following [12], a memoryless DNC is given by a finite alphabet $\mathcal{A} = (a_1, \ldots, a_m)$ of atomic symbols and an associated weight function $w\colon \mathcal{A} \to \mathbb{R}_{>0}$, $a_i \mapsto w_i > 0$. The information rate $\bar{\mathcal{H}}$ that is transmitted over the channel is given by the entropy of the input PMF divided by the average weight, i.e.,

$$\bar{\mathcal{H}}(\boldsymbol{p}) = \frac{\mathcal{H}(\boldsymbol{p})}{\sum_i p_i w_i}, \qquad \text{with } \mathcal{H}(\boldsymbol{p}) = -\sum_i p_i \log p_i. \tag{26}$$

To find the dyadic PMF that maximizes $\bar{\mathcal{H}}(\boldsymbol{p})$, we need to solve the optimization problem

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{maximize}} \quad & \bar{\mathcal{H}}(\boldsymbol{p}) \\ \text{subject to} \quad & \boldsymbol{p} \text{ is a PMF} \\ & p_i = 2^{-\ell_i}, \ell_i \in \mathbb{N}, i = 1, \ldots, m. \end{aligned} \tag{27}$$

As in the case of DMCs, this is an intractable nonlinear optimization problem with integer constraints. We will therefore proceed in the same way as we did for the DMC in Section III. We start by calculating the capacity and the capacity achieving PMF $\boldsymbol{p}^*$. This can be done by Lagrange Multipliers, see, e.g., [13]. Denote by b the base of the logarithm $\log$. The capacity is achieved by the input PMF

$$p_i^* = b^{-\mathsf{C} w_i}, \quad i = 1, \ldots, m \tag{28}$$

where C denotes capacity and is given by the greatest positive real solution of the equation

$$\sum_i b^{-s w_i} = 1. \tag{29}$$

From (28), we have the relation $w_i = -\frac{1}{\mathsf{C}} \log p_i^*$. We can thus write

$$\sum_i p_i w_i = -\frac{1}{\mathsf{C}} \sum_i p_i \log p_i^*. \tag{30}$$

Denote by R the fraction of C that can be achieved by the best dyadic PMF $\tilde{\boldsymbol{p}}$, i.e.,

$$\tilde{\boldsymbol{p}} \triangleq \underset{\boldsymbol{p} \text{ dyadic}}{\operatorname{argmax}} \bar{\mathcal{H}}(\boldsymbol{p}), \qquad \mathsf{R} \triangleq \frac{\bar{\mathcal{H}}(\tilde{\boldsymbol{p}})}{\mathsf{C}}. \tag{31}$$

In general, R is not known beforehand, but we will show in Subsection IV-A how it can be found. Suppose for now that we know R. Assume further that

$$p_i = 0, \quad \text{whenever } p_i^* = 0. \tag{32}$$

Furthermore, we use the convention $0 \log 0 = 0$. With these assumptions, we can now write $\bar{\mathcal{H}}(\boldsymbol{p})$ as

$$\bar{\mathcal{H}}(\boldsymbol{p}) = \frac{-\mathsf{R}\sum_i p_i \log p_i^* + \mathsf{R}\sum_i p_i \log p_i^* + \mathcal{H}(\boldsymbol{p})}{\sum\limits_i p_i w_i} \tag{33}$$

$$= \mathsf{RC} - \frac{\sum_i p_i \log p_i - \mathsf{R}\sum_i p_i \log p_i^*}{\sum\limits_i p_i w_i} \tag{34}$$

$$= \mathsf{RC} - \frac{\sum_i p_i \log \frac{p_i}{p_i^{*\mathsf{R}}}}{\sum\limits_i p_i w_i} = \mathsf{RC} - \frac{\mathrm{D}(\boldsymbol{p}\|\boldsymbol{p}^{*\mathsf{R}})}{\sum_i p_i w_i} \tag{35}$$

where in (34), we used (30) and the definition of entropy. By (31), for the best dyadic PMF $\tilde{\boldsymbol{p}}$ we have $\bar{\mathcal{H}}(\tilde{\boldsymbol{p}}) = \mathsf{RC}$. It follows that for any dyadic PMF $\boldsymbol{p}$, we have $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{p}^{*\mathsf{R}}) \geq 0$ and for the best dyadic PMF $\tilde{\boldsymbol{p}}$, we have $\mathrm{D}(\tilde{\boldsymbol{p}}\|\boldsymbol{p}^{*\mathsf{R}}) = 0$. We conclude that for DNCs, the best dyadic PMF is found by minimizing $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{p}^{*\mathsf{R}})$ over all dyadic PMFs $\boldsymbol{p}$ and by Proposition 1, this PMF is given by $\boldsymbol{p} = \textsc{Ghc}(\boldsymbol{p}^{*\mathsf{R}})$. Recall that, as we argued in Section III, GHC guarantees (32).

We now consider the PMF of k consecutive symbols. We denote the corresponding weights by $\boldsymbol{w}^{(k)}$. The capacity achieving joint PMF is the product of k copies of $\boldsymbol{p}^*$ and we denote it by $\boldsymbol{p}^{(k)*}$. Clearly, $w_i^{(k)} \geq k w_{\min}$ for $i = 1, \ldots, m^k$ where $w_{\min} = \min\{w_1, \ldots, w_m\}$. Using this, we get for $\bar{\mathcal{H}}(\boldsymbol{p}^{(k)})$ the lower bound

$$\bar{\mathcal{H}}(\boldsymbol{p}^{(k)}) = \frac{\mathcal{H}(\boldsymbol{p}^{(k)})}{\sum_i p_i^{(k)} w_i^{(k)}} = \frac{\mathcal{H}(\boldsymbol{p}^{(k)}) + \sum_i p_i^{(k)} \log p_i^{(k)*} - \sum_i p_i^{(k)} \log p_i^{(k)*}}{-\frac{1}{\mathsf{C}}\sum_i p_i^{(k)} \log p_i^{(k)*}} \tag{36}$$

$$= \mathsf{C} - \frac{\mathrm{D}(\boldsymbol{p}^{(k)}\|\boldsymbol{p}^{(k)*})}{\sum_i p_i^{(k)} w_i^{(k)}} \tag{37}$$

$$\geq \mathsf{C} - w_{\min}^{-1} \frac{\mathrm{D}(\boldsymbol{p}^{(k)}\|\boldsymbol{p}^{(k)*})}{k}. \tag{38}$$

For $\boldsymbol{p}^{(k)} = \textsc{Ghc}(\boldsymbol{p}^{(k)*})$, according to Proposition 2, the last term in the last line vanishes for $k \to \infty$ and we have $\bar{\mathcal{H}}(\boldsymbol{p}^{(k)}) \to \mathsf{C}$, thus GHC is asymptotically capacity achieving.

A. *Finding* R

The exact value of R is in general not known beforehand. However, R and the best dyadic PMF can be found iteratively by the Lempel-Even-Cohn (LEC) algorithm [2]. The idea of the algorithm is to start with some R, then find the best dyadic PMF for this R, and then update the value of R. The best PMF for a given R is in the original formulation of the LEC algorithm found as follows. A subset of ℓ nonzero entries of $\boldsymbol{p}$ is chosen. A Huffman-like procedure of complexity $\mathcal{O}(m \log m)$ then finds the best dyadic PMF with ℓ nonzero entries. There are $m-1$ values for ℓ that have to be evaluated, the complexity of the overall procedure is thus roughly $\mathcal{O}(m^2 \log m)$.

Algorithm 1 Finding R and the optimal dyadic PMF for DNCs

```
procedure LEC(p*)
    R ← 1
    while D(p‖p*^R) ≠ 0 do
        p ← GHC(p*^R)
        R ← H̄(p)/C
    end while
end procedure
```

From (35) and a careful study of the original formulation in [2, Sec. III,V], it can be shown that the iteration step is equivalent to minimizing the weighted KL-distance $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{p}^{*R})$. This can be done with complexity $\mathcal{O}(m \log m)$ by GHC. A formulation of the complete LEC algorithm with GHC as the iteration step is provided in Algorithm 1. Besides improving the complexity of the iteration step from $\mathcal{O}(m^2 \log m)$ to $\mathcal{O}(m \log m)$, our formulation answers a question that was raised in [14], namely how the LEC algorithm could be used to find the dyadic PMF that minimizes the KL-distance $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{p}^*)$. The simple answer is to perform the iteration step once with $R = 1$. An implementation in MATLAB of our formulation of the LEC algorithm can be found at [7].

APPENDIX

Denote by $\boldsymbol{x}$ some non-negative vector with m entries. Assume $\boldsymbol{x}$ is ordered, i.e., $x_1 \geq x_2 \geq \cdots \geq x_m$. We now show that GHC minimizes $\mathrm{D}(\boldsymbol{p}\|\boldsymbol{x})$ over all dyadic PMFs $\boldsymbol{p}$. The PMF $\boldsymbol{p}$ is dyadic if and only if there exist numbers $\ell_i \in \mathbb{N}$, $i = 1, \ldots, m$, such that $p_i = 2^{-\ell_i}, \forall i$ and $\sum_i 2^{-\ell_i} = 1$. This is equivalent to $\ell_1, \ldots, \ell_m$ being the codeword lengths of a full prefix-free code [10, Sec. 2.3.2]. Using this, we can write

$$\mathrm{D}(\boldsymbol{p}\|\boldsymbol{x}) = \sum_i p_i \log \frac{p_i}{x_i} = \log(2) \sum_i p_i \log_2 \frac{p_i}{x_i} \tag{39}$$

$$= \log(2) \sum_i 2^{-\ell_i}(-\log_2 x_i - \ell_i). \tag{40}$$

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

$$\sum_i 2^{-\ell_i}(u_i - \ell_i) \tag{41}$$

subject to $\ell_1, \ldots, \ell_m$ are the codeword lengths of a full prefix-free code. Based on (41), we now prove the optimality of GHC in a way similar to the proof given in [10, Sec. 2.5.3] for the optimality of Huffman coding.

Assume for now that an optimal algorithm assigns finite values to the codeword lengths ℓ_m and ℓ_{m-1} of the two least likely symbols, which correspond to the greatest entries u_m and u_{m-1} of $\boldsymbol{u}$. We now show that in this case, there is an optimal algorithm for which $\ell_m = \ell_{m-1}$.

Lemma 2. *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 term

$$2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell_j}(u_j - \ell_j). \tag{42}$$

By interchanging ℓ_i and ℓ_j, the term decreases:

$$\begin{aligned}
&[2^{-\ell_j}(u_i - \ell_j) + 2^{-\ell_i}(u_j - \ell_i)] - [2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell_j}(u_j - \ell_j)] && (43)\\
&= 2^{-\ell_j}(u_i - u_j) + 2^{-\ell_i}(u_j - u_i) && (44)\\
&= (\underbrace{2^{-\ell_i} - 2^{-\ell_j}}_{>0})(\underbrace{u_j - u_i}_{<0}) < 0 && (45)
\end{aligned}$$

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

Lemma 3. *There is an optimal algorithm for which the codewords of the two greatest entries u_m and u_{m-1} are siblings, i.e., $\ell_m = \ell_{m-1}$, and in addition, no other codeword is longer than ℓ_m and ℓ_{m-1}.*

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

We can now use $\ell_m = \ell_{m-1}$ to rewrite (41):

$$\begin{aligned}
\sum_{i=1}^{m} 2^{-\ell_i}(u_i - \ell_i) &= \sum_{i=1}^{m-2} 2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell_{m-1}}(u_{m-1} - \ell_{m-1}) + 2^{-\ell_m}(u_m - \ell_m) && (46)\\
&= \sum_{i=1}^{m-2} 2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell_m}(u_{m-1} + u_m - 2\ell_m) && (47)\\
&= \sum_{i=1}^{m-2} 2^{-\ell_i}(u_i - \ell_i) + 2^{-(\ell_m - 1)}\Big[\Big(\underbrace{\frac{u_{m-1} + u_m}{2} - 1}_{\triangleq u'}\Big) - (\underbrace{\ell_m - 1}_{\triangleq \ell'})\Big] && (48)\\
&= \sum_{i=1}^{m-2} 2^{-\ell_i}(u_i - \ell_i) + 2^{-\ell'}(u' - \ell'). && (49)
\end{aligned}$$

Thus, by combining u_m and u_{m-1} through

$$u' = \frac{u_{m-1} + u_m}{2} - 1 \tag{50}$$

the size m problem is reduced to a size $m - 1$ problem.

The optimal algorithm may assign probability zero to the greatest entry u_m, which corresponds to $\ell_m = \infty$. We thus have

$$\sum_{i=1}^{m} 2^{-\ell_i}(u_i - \ell_i) = \sum_{i=1}^{m-1} 2^{-\ell_i}(u_i - \ell_i) + 2^{-\infty}(u_i - \infty) = \sum_{i=1}^{m-1} 2^{-\ell_i}(u_i - \ell_i) \tag{51}$$

where we used the convention $-0 \log 0 = 0$ and equivalently $2^{-\infty}\infty = 0$. Thus, if $\ell_m = \infty$, the size m problem reduces to a size $m - 1$ problem.

It remains to check if it is better to assign probability zero to u_m or to combine u_m and u_{m-1}. First, assume the algorithm combines u_m and u_{m-1}. Then the contribution

to the sum (41) is $2^{-\ell'}(u' - \ell')$. We can now assign probability zero to u_m and use the codeword of u' for u_{m-1}. The contribution of u_m to (41) is then zero and the contribution of u_{m-1} is $2^{-\ell'}(u_{m-1} - \ell')$. Thus, since our aim is to minimize (41), doing the former is better if and only if

$$2^{-\ell'}(u_{m-1} - \ell') > 2^{-\ell'}(u' - \ell') \tag{52}$$

$$\Leftrightarrow (u_{m-1} - \ell') > \left(\frac{u_{m-1} + u_m}{2} - 1\right) - \ell' \tag{53}$$

$$\Leftrightarrow u_{m-1} > \frac{u_{m-1} + u_m}{2} - 1 \tag{54}$$

$$\Leftrightarrow u_{m-1} > u_m - 2. \tag{55}$$

Recalling $x_i = 2^{-u_i}$, the updating rule (50) and the condition (55) can be expressed in terms of $\boldsymbol{x}$ as

$$x' = \begin{cases} x_{m-1}, & \text{if } x_{m-1} \geq 4x_m \\ 2\sqrt{x_{m-1}x_m}, & \text{if } x_{m-1} < 4x_m. \end{cases} \tag{56}$$

References

[1] F. R. Kschischang and S. Pasupathy, "Optimal nonuniform signaling for Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 913–929, 1993.

[2] 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.

[3] K. J. Kerpez, "Runlength codes from source codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pp. 682–687, 1991.

[4] G. Ungerboeck, "Huffman shaping," in *Codes, Graphs, and Systems*, R. Blahut and R. Koetter, Eds. Springer, 2002, ch. 17, pp. 299–313.

[5] J. Abrahams, "Variable-length unequal cost parsing and coding for shaping," *IEEE Trans. Inf. Theory*, vol. 44, no. 4, pp. 1648–1650, 1998.

[6] 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.

[7] G. Böcherer, "Geometric huffman coding," http://www.georg-boecherer.de/ghc, Dec. 2010.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, Inc., 2006.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.

[10] R. G. Gallager, *Principles of Digital Communication*. Cambridge University Press, 2008.

[11] ——, *Information Theory and Reliable Communication*. John Wiley & Sons, Inc., 1968.

[12] R. M. Krause, "Channels which transmit letters of unequal duration," *Inf. Contr.*, vol. 5, pp. 3–24, 1962.

[13] R. S. Marcus, "Discrete noiseless coding," Master's thesis, MIT, 1957.

[14] J. Abrahams, "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.