A Recursive Quantizer Design Algorithm for Binary-Input Discrete Memoryless Channels

Mehdi Dabirnia, *Member, IEEE*, Alfonso Martinez, *Senior Member, IEEE*, and Albert Guillén i Fàbregas, *Senior Member, IEEE*

***Abstract*— The optimal quantization of output binary-input discrete memoryless channels is considered, whereby the optimal quantizer preserves at least a constant α-fraction of the original mutual information, with the smallest output cardinality. Two recursive methods with top-down and bottom-up approaches are developed; these methods lead to a new necessary condition for the recursive quantizer design. An efficient algorithm with linear complexity, based on dynamic programming and the new necessary optimality condition, is proposed.**

***Index Terms*— Channel quantization, discrete memoryless channel, mutual information preserving quantizer, partitioning and clustering.**

I. Introduction

QUANTIZATION has practical applications in hardware implementations of communication systems, e. g. channel-output quantization [2]–[11], message-passing decoders [12] and polar code construction [13]. In such applications, the number of quantization levels induces a trade-off between performance and system complexity. Therefore, it is of interest to use as few quantization levels as possible while maintaining reliable communication with a given transmission rate. Recently, the authors studied channel-output quantization from an information-theoretic mismatched-decoding perspective [14]. This study revealed that the best mismatched decoder coincides with maximum-likelihood decoding for the channel between the channel input and the quantizer output. This result supports the approach of optimizing the quantizer based on a performance metric for the quantized channel, e. g. mutual information [2]–[8] or error exponent [9].

Manuscript received August 14, 2020; revised February 23, 2021; accepted April 16, 2021. Date of publication April 28, 2021; date of current version August 16, 2021. This work has been funded in part by the European Research Council under grant 725411, and by the Spanish Ministry of Economy and Competitiveness under grant TEC2016-78434-C3-1-R. This article was presented at the 2020 International Zürich Seminar on Communications, Zürich, Switzerland, February 2020 [1]. The associate editor coordinating the review of this article and approving it for publication was C. Tian. *(Corresponding author: Mehdi Dabirnia.)*

Mehdi Dabirnia and Alfonso Martinez are with the Department of Information and Communication Technologies, Universitat Pompeu Fabra, 08018 Barcelona, Spain (e-mail: mehdi.dabirnia@upf.edu; alfonso.martinez@ieee.org).

Albert Guillén i Fàbregas is with the Institució Catalana de Recerca i Estudis Avançats, 08010 Barcelona, Spain, also with the Department of Information and Communication Technologies, Universitat Pompeu Fabra, 08018 Barcelona, Spain, and also with the Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, U.K. (e-mail: guillen@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2021.3076174.

Digital Object Identifier 10.1109/TCOMM.2021.3076174

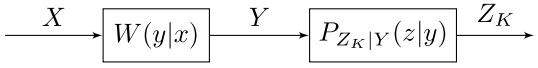

Fig. 1. A discrete memoryless channel followed by a quantizer.

Discrete channel quantization is also related to clustering and partitioning problem in learning theory. An important result by Burshtein *et al.* [15] gives conditions on the existence of an optimal partitioning. Building on this result, Kurkoski and Yagi studied in [2] output quantization of binary-input discrete memoryless channels, described in more detail in Sect. I-B, and developed a dynamic-programing algorithm to find a maximum mutual information quantizer. In this paper, we build on these results and study recursive methods for designing a quantizer that preserves a constant fraction of the mutual information, as formulated in the next section.

A. Problem Formulation

Consider a discrete memoryless channel (DMC) followed by an output quantizer, as shown in Fig. 1. The channel input X takes values in $\mathcal{X} = \{1, \ldots, J\}$, with probability distribution $p_x \triangleq P_X(x)$, and the channel output Y takes values in $\mathcal{Y} = \{1, \ldots, M\}$, with channel transition probabilities $W(y|x)$. Channel output probabilities are denoted by $\pi_y \triangleq P_Y(y)$. The channel output is quantized to Z_K, which takes values in $\mathcal{Z}_K = \{1, \ldots, K\}$, by a possibly stochastic quantizer Q with transition probabilities $P_{Z_K|Y}(z|y)$. The quantizer output probabilities are denoted by $\pi_{z,K} \triangleq P_{Z_K}(z)$.

Let $P(x|y) \triangleq P_{X|Y}(x|y)$ and $P_K(x|z) \triangleq P_{X|Z_K}(x|z)$ denote the conditional probability distribution of the channel input given channel output and quantizer output, respectively. Hence, the mutual information between X and Z_K is

$$I(X; Z_K) = \sum_{z \in \mathcal{Z}_K} \sum_{x \in \mathcal{X}} \pi_{z,K} P_K(x|z) \log \frac{P_K(x|z)}{p_x}. \quad (1)$$

Let $\mathcal{Q}_K$ denote the set of all quantizers Q with K outputs, including stochastic quantizers. In the literature, the quantizer optimization problem is usually formulated as finding an optimal quantizer Q_K^* for fixed cardinality K that maximizes the mutual information of the quantized channel [2]–[7], i. e.

$$Q_K^* = \arg\max_{Q \in \mathcal{Q}_K} I(X; Z_K). \quad (2)$$

We formulate instead the quantizer optimization as follows: for a given $\alpha \in [0, 1]$, find an optimal quantizer Q_α that preserves at least an α-fraction of the original mutual information with

the smallest number of quantization levels. To that order, we define a set $\mathcal{S}_{\alpha,k}$ for $1 \leq k \leq M$ as

$$\mathcal{S}_{\alpha,k} \triangleq \big\{Q \in \mathcal{Q}_k, I(X;Z_k) \geq \alpha I(X;Y)\big\}. \tag{3}$$

We notice that the set $\mathcal{S}_{\alpha,k}$ can be empty for small values of k, e.g. $\mathcal{S}_{\alpha,1}$ is empty for any positive α and $I(X;Y) > 0$. Denote with K^* the smallest value of k for which the set $\mathcal{S}_{\alpha,k}$ is non-empty. Then, the optimal quantizer is given by

$$Q_\alpha = Q^*_{K^*}. \tag{4}$$

B. Previous Work

A deterministic quantizer Q partitions $\mathcal{Y}$ into K non-overlapping subsets $\{\mathcal{A}_1, \ldots, \mathcal{A}_K\}$, mapping each output y to only one quantized output z,

$$Q : \{1, \ldots, M\} \to \{1, \ldots, K\}. \tag{5}$$

For such mapping, we define the pre-image of z as

$$\mathcal{A}_z = \big\{y \in \mathcal{Y} : Q(y) = z\big\}, \tag{6}$$

the set of channel outputs mapped to z. For any DMC and fixed output cardinality K, Kurkoski and Yagi [2] showed that a deterministic quantizer maximizes the mutual information between channel input and quantized output (1); considering only deterministic quantizers is thus sufficient to find the optimal quantizer Q_α.

For each channel output y, we define a vector v_y,

$$v_y = [P(1|y), P(2|y), \ldots, P(J-1|y)], \tag{7}$$

with $v_y \in \mathcal{U} = [0,1]^{J-1}$. We define an equivalent quantizer $\tilde{Q}$ on the vectors $\{v_1, \ldots, v_M\}$ as $\tilde{Q}(v_y) = Q(y) = z$ and the corresponding pre-images as

$$\tilde{\mathcal{A}}_z = \Big\{v_y : \tilde{Q}(v_y) = z\Big\}. \tag{8}$$

Kurkoski and Yagi in [2, Lemma 2], using the results of [15], study a condition for an optimal equivalent quantizer $\tilde{Q}^*$ and show the existence of an optimal equivalent quantizer $\tilde{Q}^*$ for which any two distinct preimages $\tilde{\mathcal{A}}_z$ and $\tilde{\mathcal{A}}_{z'}$ are separated by a hyperplane in the Euclidean space $\mathcal{U}$. Unfortunately, this condition does not offer a practical search method for quantizer design in general; however, as suggested in [2], it simplifies the problem for the binary-input case.

To find an optimal quantizer Q_α as defined in (4), it is not feasible to directly optimize over the output cardinality and find K^*. Nazer et al. in [10] showed that, for binary input case there always exists a K-level quantizer attaining the mutual information of $\Omega\Big(\frac{-K \cdot I(X;Y)}{\log(I(X;Y))}\Big)$ and that there exist pairs of X, Y for which the mutual information attained by any K-level quantizer is $\mathcal{O}\Big(\frac{-K \cdot I(X;Y)}{\log(I(X;Y))}\Big)$. For larger finite input alphabets it is established in [11] that an α-fraction of the mutual information can be preserved using roughly $\big(\log(|\mathcal{X}|/I(X;Y))\big)^{\alpha \cdot (|\mathcal{X}|-1)}$ quantizer levels. While these results give an upper bound on or an approximate number of levels preserving an α-fraction of the mutual information, they do not provide a way to find the value of K^* in (4).

The problem of finding Q_α can be tackled by either a *bottom-up* or *top-down* approach. The former starts with the trivial partition into M subsets $\mathcal{A}_z$, $1 \leq z \leq M$, where each $\mathcal{A}_z$ contains exactly one element of $\mathcal{Y}$. At each step, we decrease the cardinality k by one and design an optimal quantizer Q^*_k with output size k. We stop when the corresponding mutual information goes below the desired threshold. The latter approach starts with the other trivial solution with single partition containing all the elements, i.e. $\mathcal{A}_1 = \mathcal{Y}$. At each step, we increment the cardinality k by one and design an optimal quantizer Q^*_k with output size k. We stop when the corresponding mutual information reaches (or exceeds) the desired threshold. In both approaches, the quantizer design at each step can be performed either *recursively*, namely by starting from the result of previous step, or *independently* of the previous step result.

An example of a recursive bottom-up approach is the *agglomerative information bottleneck* [16] which has been rediscovered multiple times in the literature under names such as *greedy merging* or *greedy combining* [12], [13]. This algorithm iteratively reduces the cardinality by merging two outputs into a new single output. At each iteration, the greedy algorithm evaluates all possible pairwise merges and selects the one that minimizes the mutual information loss. Although the algorithm finds the optimal pairwise merge at each step, it is globally suboptimal, since it fixes all the previously performed merges. This algorithm has complexity $O(M^2)$.

As for the independent approach, several design algorithms from the literature can be utilized. For binary-input DMCs, Kurkoski and Yagi developed an algorithm based on dynamic programming that finds an optimal K-level quantizer with complexity $O(K(M-K)^2)$ [2]. Iwata and Ozawa [3] improved the complexity to $O(K(M-K))$ using the SMAWK algorithm. For non-binary inputs, finding the optimal quantizer is an NP-hard problem [17] and several suboptimal algorithms are proposed in the literature. An example is KL-means quantizer [4], [18], a variation of the K-means clustering algorithm by replacing Euclidean distance metric with Kullback-Leibler divergence. This algorithm has complexity $O(JKMT)$ where T is the number of iterations that the algorithm is run to converge to a local optimum. Another example is a dynamic programming method [6] with complexity $O(JK(M-K)^2)$ to find an optimal sequential deterministic quantizer under a general cost function. The authors also derive a sufficient condition for general optimality of this method and under a condition for the DMC channel, they propose two techniques to reduce the complexity of their algorithm. The complexity of a top-down (or bottom-up) approach with independent design at each step is K^* (or $M - K^*$) times the complexity of a single-step run, respectively. So using the algorithm from [3] with independent top-down approach, one can find Q_α with complexity $O(K^{*2}(M-K^*))$. In Section IV, we propose a recursive algorithm that finds Q_α with complexity $O(K^*M)$.

C. Restriction to Binary Inputs

For the rest of paper we restrict ourselves to binary inputs, for which the posterior conditional probabilities $v_y = P(1|y)$

are in one-dimensional space $\mathcal{U} = [0, 1]$. We also assume that the outputs are labeled to satisfy

$$v_1 < v_2 < \cdots < v_M. \tag{9}$$

There is no loss of generality in assuming (9), as outputs can always be relabeled to satisfy this condition. The inequalities in (9) are strict since in case of equality, the corresponding outputs can be merged without information loss, reducing the output cardinality. Furthermore, we only consider deterministic quantizers as they include the optimal quantizer [2].

D. Contributions

In Section II, we analyze the greedy merging algorithm and derive from the analysis a necessary condition for any optimal quantizer. Furthermore, we characterize and prove some properties of the greedy merging algorithm.

In Section III we propose two new recursive methods for optimal quantizer design: a bottom-up approach inspired by the analysis of greedy merging algorithm in Section II and a top-down approach as dual of the first method. Using these two recursive methods an important necessary condition for recursive quantizer design is given and a concavity property corresponding to the fraction of mutual information versus cardinality plot of optimal quantizers is proven. The proofs of results are given in the Appendix.

Section IV presents the splitting algorithm, a dynamic-programming algorithm for recursive quantizer design. We apply the necessary condition derived in Sect. III to the Quantizer Design Algorithm in [2] to reduce the complexity of recursive design. A complexity analysis for the recursive splitting algorithm shows complexity of $O(K^*M)$ which is obtained by using the SMAWK algorithm [20] for performing the matrix search.

II. Analysis of Greedy Merging Algorithm

The quantizer optimization for a fixed cardinality K formulated in (2) can be rewritten as the minimization of mutual information loss with respect to the original channel as

$$Q_K^* = \arg\min_{Q \in \mathcal{Q}_K} I(X;Y) - I(X;Z_K) \tag{10}$$

as $I(X;Y)$ is fixed for a given input distribution and channel. A quantizer from M channel outputs to K quantized outputs is a combination of $(M-K)$ pairwise merges and its corresponding mutual information loss $I(X;Y) - I(X;Z_K)$ can be decomposed into $M-K$ terms as

$$I(X;Y) - I(X;Z_K) = \sum_{k=K}^{M-1} I(X;Z_{k+1}) - I(X;Z_k) \tag{11}$$

where $Z_M = Y$ and each summation term

$$\Delta I_k = I(X;Z_{k+1}) - I(X;Z_k) \tag{12}$$

is the mutual information loss for a single-step quantizer, i. e. a pairwise merge. Let us define the partial mutual information $I_k(z)$ as the contribution that a quantizer output $z \in \mathcal{Z}_k$ makes to the mutual information given by

$$I_k(z) = \pi_{z,k} \sum_{x \in \mathcal{X}} P_k(x|z) \log \frac{P_k(x|z)}{p_x}. \tag{13}$$

Then, the mutual information $I(X;Z_k)$ is given by

$$I(X;Z_k) = \sum_{z \in \mathcal{Z}_k} I_k(z). \tag{14}$$

As a suboptimal approach, instead of minimizing the total mutual information loss in (11), we can minimize each summation term ΔI_k in a recursive bottom-up manner (from $M-1$ to K). We start from the trivial solution with M outputs and at each step we search for a single-step quantizer (pairwise merge) which minimizes the mutual information loss for that step. The optimal single-step quantizer is given by

$$\hat{Q}_k = \arg\min_{Q \in \mathcal{Q}_{\mathrm{m},k}} I(X;Z_{k+1}) - I(X;Z_k), \tag{15}$$

where $\mathcal{Q}_{\mathrm{m},k}$ is set of all $\binom{k+1}{2}$ possible pairwise merges on $\mathcal{Z}_{k+1}$. This method is called greedy merging [12] since it combines a greedy search over all possible pairwise merges with the selection of the best such merge.

Let us assume that the single level quantizer $\hat{Q}$ merges two outputs $i, j \in \mathcal{Z}_{k+1}$ into $z' \in \mathcal{Z}_k$ and maps the remaining symbols one-to-one, i. e. $\mathcal{Z}_{k+1} \setminus \{i,j\} \mapsto \mathcal{Z}_k \setminus \{z'\}$. We can compute the mutual information loss of the (i,j) merge, denoted by $\Delta I_k(i,j)$, as

$$\Delta I_k(i,j) = I_{k+1}(i) + I_{k+1}(j) - I_k(z') \tag{16}$$

$$= \sum_{x \in \mathcal{X}} \Big(\pi_{i,k+1} \Phi\big(P_{k+1}(x|i)\big) + \pi_{j,k+1} \Phi\big(P_{k+1}(x|j)\big) - \pi_{z',k} \Phi\big(P_k(x|z')\big) \Big), \tag{17}$$

where $\Phi(p) = p \log(p)$, $\pi_{z',k} = \pi_{i,k+1} + \pi_{j,k+1}$ and

$$P_k(x|z') = \frac{\pi_{i,k+1} P_{k+1}(x|i) + \pi_{j,k+1} P_{k+1}(x|j)}{\pi_{z',k}}.$$

According to [2, Lemma 3], there is an optimal quantizer Q_K^* with boundaries satisfying

$$a_0^* = 0 < a_1^* < a_2^* < \cdots < a_{K-1}^* < a_K^* = M, \tag{18}$$

such that the preimages of the quantizer outputs consist of contiguous set of integers,

$$\mathcal{A}_z^* = \{a_{z-1}^* + 1, \ldots, a_z^*\}, \tag{19}$$

for $z \in \mathcal{Z}_K$. We show that this condition must hold for all optimal quantizers.

Lemma 1: For any three channel/quantizer outputs h, i and j satisfying $v_h < v_i < v_j$ at least one of the following is true,

$$\begin{cases} \Delta I(h,i) < \Delta I(h,j) & \text{if } \dfrac{\pi_h}{\pi_j} \le \dfrac{v_j - v_i}{v_i - v_h}, \\ \Delta I(i,j) < \Delta I(h,j) & \text{if } \dfrac{\pi_h}{\pi_j} \ge \dfrac{v_j - v_i}{v_i - v_h}. \end{cases} \tag{20}$$

The proof is in Appendix A. Lemma 1 shows that for any quantizer that does not satisfy the condition in (19), there exists another quantizer satisfying this condition that has a

higher mutual information. Hence, we have the following corollary stating the necessary condition for an optimal quantizer.

Corollary 1: Any optimal quantizer has convex preimages, i.e. the set $\mathcal{A}_z^*$ is a contiguous set of integers for all $z \in \mathcal{Z}_K$.

The necessary condition in Corollary 1 implies the condition (18) on optimal boundaries a_z^* and simplifies quantizer design. Using this necessary condition, an exhaustive search for the optimal boundaries has complexity $\binom{M-1}{K-1}$, i.e. $\mathcal{O}(M^{K-1})$.

Another consequence of Lemma 1 is that greedy merging always combines two adjacent outputs at each step, therefore, the set $\mathcal{Q}_{\mathrm{m},k}$ should only include $(z, z+1)$ merges with $1 \leq z \leq k$ which has k possibilities.

Corollary 2: The greedy merging algorithm results in quantizers with convex preimages.

Proof: We prove this corollary by induction. For the trivial quantizer with $K = M$ outputs, where the preimage of each output contains exactly one element of $\mathcal{Y}$, i.e. $\mathcal{A}_z^{(M)} = \{z\}$, for $1 \leq z \leq M$, this statement clearly holds. Moreover, the quantizer outputs are so labeled that consecutive ones contain contiguous elements of $\mathcal{Y}$, preserving the ordering in (9).

Now assume that at level $k+1$ of the greedy merging algorithm the outputs have convex preimages, i.e. each set $\mathcal{A}_z^{(k+1)}$, $1 \leq z \leq k+1$ contains contiguous elements of $\mathcal{Y}$, and the quantizer outputs are labeled such that

$$v_1^{k+1} < v_2^{k+1} < \cdots < v_{k+1}^{k+1}, \tag{21}$$

where $v_z^{k+1} = P_{k+1}(1|z)$ and the consecutive quantizer outputs have preimages with contiguous elements of $\mathcal{Y}$.

At level k, greedy merging combines two adjacent outputs according to Lemma 1. Without loss of generality assume that the algorithm merges z with $z+1$ from $\mathcal{Z}_{k+1}$ and one-to-one maps the rest of the outputs. Based on the previous assumption, it is clear that each new output $z' \in \mathcal{Z}_k$ has a preimage with contiguous elements. Also,

$$v_z^{k+1} < v_{z'}^k = \frac{\pi_{z,k+1} v_z^{k+1} + \pi_{z+1,k+1} v_{z+1}^{k+1}}{\pi_{z,k+1} + \pi_{z+1,k+1}} < v_{z+1}^{k+1}, \tag{22}$$

hence we have

$$v_1^k < v_2^k < \cdots < v_k^k. \tag{23}$$

Furthermore, it is clear that any two consecutive quantizer outputs have preimages containing contiguous elements of $\mathcal{Y}$. The proof is complete by induction. ∎

Next consider performing greedy merging algorithm for all possible output cardinalities $1 \leq k \leq M$ in a bottom-up manner and looking at the mutual information of the quantized channel $I(X;Z_k)$ as a function of the output cardinality k. In [16], the authors empirically found that $I(X;Z_k)$ is a concave function of k, in other words, the mutual information loss ΔI_k in (12) is increasing with decreasing k. For non-binary inputs ($J > 2$) we found counter-examples for this observation, however, in Appendix B we prove this result for binary inputs:

Theorem 1: The mutual information loss at each step of the greedy merging algorithm can only increase.

III. Recursive Search for Q_α

Corollary 1 gives a necessary condition for a mutual information maximizing quantizer. Based on this condition, the quantizer design for a fixed output cardinality K boils down to searching a set of optimal boundaries a_z^* as in (18). It remains to answer how to use information from previous steps in order to recursively search for Q_α as defined in Section I-A. In this section, we obtain another necessary condition for recursive optimal quantizer design and show that knowing the boundary values of optimal $(k+1)$-level quantizers (or $(k-1)$-level quantizers) simplifies the search for boundary values of optimal k-level quantizers.

A. Modified Greedy Merging

In the following, we propose a new greedy algorithm that starts from a seed quantizer and searches over pairwise merges and another set of single-step quantizers which we denote them as *contractions*. First, let us define splits.

Definition 1 (Splitting an Output): A quantizer output z with preimage $\mathcal{A}_z = \{a_{z-1}+1, \ldots, a_z\}$ of size $b_z = |\mathcal{A}_z| = a_z - a_{z-1} \geq 2$, splits into two non-empty parts z_L (left) and z_R (right) with respective preimages $\mathcal{A}_{z_L} = \{a_{z-1}+1, \ldots, s\}$ and $\mathcal{A}_{z_R} = \{s+1, \ldots, a_z\}$. This split can be done in $b_z - 1$ different ways, $a_{z-1}+1 \leq s \leq a_z - 1$.

Definition 2 (Merging a Split Output): A split output z_k with two non-empty parts z_L (left) and z_R (right) is merged by two actions: first, z_L merges with $z-1$ or $(z-1)_R$ if it has been split too; second, z_R merges with $z+1$ or $(z+1)_L$.

A *contraction from $(k+1)$-level to k-level* is a single-step quantizer that consists of merges and possibly splits, as defined by the following sequence of steps:

1) *Input: a $(k+1)$-level quantizer with output boundaries $\{a_0 = 0, a_1, \ldots, a_k, a_{k+1} = M\}$.*
2) *Select a set of consecutive non-boundary outputs $\mathcal{Z}_s = \{i, i+1, \ldots, j\} \subset \mathcal{Z}_{k+1}$ with $i > 1$, $j < k+1$ and $b_z = |\mathcal{A}_z| \geq 2$ for all $i \leq z \leq j$.*
3) *Split each $z \in \mathcal{Z}_s$ according to Definition 1. This step can be done in $\prod_{z=i}^{j}(b_z - 1)$ different ways.*
4) *Merge z_R with $(z+1)_L$ for all $i \leq z \leq j-1$, also merge $i-1$ with i_L and j_R with $j+1$.*
5) *Output: a k-level quantizer with output boundaries $\{a'_0, \ldots, a'_k\}$ for which*

$$\begin{cases} a'_z = a_z & \text{for } \ 0 \leq z \leq i-2 \\ a_z < a'_z < a_{z+1} & \text{for } \ i-1 \leq z \leq j-1 \\ a'_z = a_{z+1} & \text{for } \ j \leq z \leq k. \end{cases} \tag{24}$$

We denote the set of all quantizers obtained by a *contraction* from all optimal $(k+1)$-level quantizers by $\mathcal{Q}_{\mathrm{c},k}$. Modified greedy merging is a bottom-up approach that starts from the trivial solution with M outputs and at each step decreases the output cardinality by one, then performs a greedy search over all possible *contractions* $\mathcal{Q}_{\mathrm{c},k}$ and all pairwise merges $\mathcal{Q}_{\mathrm{m},k}$, and finally selects the ones with lowest mutual information loss. It stores all the quantizers with highest mutual information in each step to use them as a seed for the next step.

As proved in Appendix C, we have the following optimality property for the modified greedy merging algorithm.

Theorem 2: Modified greedy merging finds all optimal quantizers Q_k^* for every output cardinality $1 \le k \le M$.

As an example to illustrate a *contraction*, consider a quantizer with 3 outputs $\{1,2,3\}$ with preimages $\mathcal{A}_1 = \{1, \ldots, a_1\}$, $\mathcal{A}_2 = \{a_1+1, \ldots, a_2\}$ and $\mathcal{A}_3 = \{a_2+1, \ldots, M\}$. In the second step of the *contraction*, the only possibility for a set of consecutive non-boundary outputs is $\{2\}$ if $b_2 = |\mathcal{A}_2| \ge 2$. In third step, we split the second output into two parts with preimages $\mathcal{A}_{2\mathrm{L}} = \{a_1+1, \ldots, s\}$ and $\mathcal{A}_{2\mathrm{R}} = \{s+1, \ldots, a_2\}$ where $a_1+1 \le s \le a_2-1$. We merge 2_{L} with *1* and 2_{R} with *3* according to the fourth step. The output of this *contraction* is a quantizer with two outputs that has the boundaries $\{a'_0 = 0, a'_1, a'_2 = M\}$ where $a_1 < a'_1 = s < a_2$. The set of all $b_2 - 1$ possible *contractions* for this example are specified by $a_1 + 1 \le s \le a_2 - 1$.

B. Modified Greedy Splitting

Modified greedy splitting is a top-down algorithm, dual to modified greedy merging. It starts from the trivial solution with a single output and increases the output cardinality by one at each step, performing a greedy search over all possible *expansions*, to be defined in the following paragraph. We denote the set of all quantizers obtained by *expansion* from all optimal $(k-1)$-level quantizers by $\mathcal{Q}_{\mathrm{e},k}$. At each step, it keeps all quantizers with the highest mutual information to use them as seed for the next step. In analogy to Theorem 2, we have the following result, which can be proved by showing that an *expansion* is a dual of a *contraction* or a pairwise merge and hence modified greedy splitting is the dual of modified greedy merging,

Theorem 3: Modified greedy splitting finds all optimal quantizers Q_k^* for all output cardinalities $1 \le k \le M$.

An *expansion* consists of splits and merges, as described in the following steps. *Expansion from $(k-1)$-level to k-level:*

1) *Input: a $(k-1)$-level quantizer with boundaries $\{a_0 = 0, a_1, \ldots, a_{k-2}, a_{k-1} = M\}$*
2) *Select a non-empty set of consecutive outputs $\mathcal{Z}_s = \{i, i+1, \ldots, j\} \subseteq \mathcal{Z}_{k-1}$ with $i \ge 1$, $j \le k-1$ and $b_z = |\mathcal{A}_z| \ge 2$ for all $i \le z \le j$.*
3) *Split each $z \in \mathcal{Z}_s$ according to Definition 1. This step can be done in $\prod_{z=i}^{j}(b_z - 1)$ different ways.*
4) *If $|\mathcal{Z}_s| = 1$, omit this step otherwise merge z_R with $(z+1)_L$ for all $i \le z \le j-1$.*
5) *Output: a k-level quantizer with output boundaries $\{a'_0, \ldots, a'_k\}$ for which*

$$\begin{cases} a'_z = a_z & \text{for } \ 0 \le z \le i-1 \\ a_{z-1} < a'_z < a_z & \text{for } \ i \le z \le j \\ a'_z = a_{z-1} & \text{for } \ j+1 \le z \le k. \end{cases} \tag{25}$$

As an example of *expansion*, consider a quantizer with two outputs $\{1,2\}$ with preimages $\mathcal{A}_1 = \{1, \ldots, a_1\}$, $\mathcal{A}_2 = \{a_1+1, \ldots, M\}$. An *expansion* can be obtained in two different ways. One is simply by splitting one of the outputs which can be performed in $b_1 - 1$ and $b_2 - 1$ different ways for first and second output, respectively. Another one is by splitting both outputs and merging 1_{R} with 2_{L} which can be performed in $(b_1-1)(b_2-1)$ different ways. The output of any such *expansion* is a quantizer with three outputs and boundaries $\{a'_0 = 0, a'_1, a'_2, a'_3 = M\}$ with either $0 < a'_1 < a_1 = a'_2 < M$ or $0 < a'_1 = a_1 < a'_2 < M$ or $0 < a'_1 < a_1 < a'_2 < M$.

Theorem 3 implies the following

Corollary 3 (Recursive Necessary Condition): Assuming that the $(k-1)$-level optimal quantizer has boundaries $\{a_z\}_{z=0}^{k-1}$, any k-level optimal quantizer (with boundaries $\{a'_z\}_{z=0}^{k}$) should satisfy (25) *for some $0 < i \le j < k$.*

The complexity of the modified greedy algorithms is $\mathcal{O}\big(\big(\frac{M}{k-1}\big)^{k-1}\big)$ in the worst case. In Section IV we provide a dynamic programming based algorithm incorporating the necessary condition in Corollary 3 for recursive design of the optimal quantizers.

C. Preserved Mutual Information at Level k

The fraction of mutual information preserved by a k-level quantizer is $\alpha_k = \frac{I(X,Z_k)}{I(X;Y)}$ which starts at 0 for $k = 0$ and approaches to 1 as k goes to M. Theorem 1 showed that α_k is concave in k for quantizers obtained by greedy merging. In Appendix D we prove a similar property for optimal quantizers, using modified greedy splitting, which finds optimal quantizers recursively.

Theorem 4: The mutual information difference $\Delta I_k^* = I(X;Z_k) - I(X;Z_{k-1})$ of the optimal quantizers decreases by increasing k.

This theorem shows that in a top-down recursive design, the increase in the fraction of the preserved mutual information by the optimal quantizers, given by $\delta_{\alpha_k} = \frac{I(X;Z_k) - I(X;Z_{k-1})}{I(X;Y)}$, can only decrease with increasing k. Therefore, if at some point δ_{α_k} becomes relatively small, it indicates reaching a meaningful quantizer cardinality. Hence, further runs of the recursive algorithm will not result in significant gains in the terms of mutual information. This also suggests that the termination condition in the recursive algorithm can be based on δ_{α_k} as well.

IV. Dynamic Programming Based Algorithm

This section describes the splitting algorithm, a modified version of the Quantizer Design Algorithm [2] that incorporates the necessary condition of Corollary 3 to reduce the complexity of recursive design. We describe first a single-step version, which takes the boundary values of the optimal $(k-1)$-level quantizer as input and finds optimal k-level quantizers maximizing the mutual information and satisfying the necessary condition of Corollary 3. Then, we provide the recursive version of the algorithm, which finds the optimal quantizer Q_α in (4) recursively.

A. Splitting Algorithm

The algorithm, an instance of dynamic programming, has a state value $S_z(y)$, the maximum partial mutual information when channel outputs 1 to y are quantized to quantizer outputs

1 to z. This value can be computed recursively by conditioning on the state value at time index $z-1$:

$$S_z(y) = \max_{y'} \big\{ (S_{z-1}(y') + I(y' \to y) \big\}, \tag{26}$$

where $I(y' \to y)$ is the contribution of quantizer output z with $\mathcal{A}_z = \{y'+1, \ldots, y\}$ to the mutual information, i.e.

$$I(i \to j) = \sum_{x \in \mathcal{X}} p_x \sum_{y=i+1}^{j} W(y|x) \log \frac{\sum_{\hat{y}=i+1}^{j} W(\hat{y}|x)}{\sum_{\hat{y}=i+1}^{j} P_Y(\hat{y})}. \tag{27}$$

Algorithm 1 Splitting Algorithm (Single Step)

Input: M, k, p_x, $P(y|x)$, $\{a_z\}_{z=0}^{k-1}$
Output: $\{a'_z\}_{z=0}^{k}$

1 **for** $y \leftarrow 1$ *to* a_1 **do**
2 $S_1(y) \leftarrow I(0 \to y)$
3 $h_1(y) \leftarrow 0$
4 $S_2(a_1) \leftarrow \max_{y' \in \{1,\ldots,a_1-1\}} S_1(y') + I(y' \to a_1)$
5 $h_2(a_1) \leftarrow \arg\max_{y' \in \{1,\ldots,a_1-1\}} S_1(y') + I(y' \to a_1)$
6 **for** $z \leftarrow 2$ *to* $k-1$ **do**
7 **for** $y \leftarrow a_{z-1}+1$ *to* $a_z - 1$ **do**
8 $S_z(y) \leftarrow \max_{y' \in \{a_{z-2}+1,\ldots,a_{z-1}\}} S_{z-1}(y') + I(y' \to y)$
9 $h_z(y) \leftarrow \arg\max_{y' \in \{a_{z-2}+1,\ldots,a_{z-1}\}} S_{z-1}(y') + I(y' \to y)$
10 **if** $z < k-1$ **then**
11 $S_z(a_z) \leftarrow S_{z-1}(a_{z-1}) + I(a_{z-1} \to a_z)$
12 $h_z(a_z) \leftarrow a_{z-1}$
13 $S_{z+1}(a_z) \leftarrow \max_{y' \in \{a_{z-1},\ldots,a_z-1\}} S_z(y') + I(y' \to a_z)$
14 $h_{z+1}(a_z) \leftarrow \arg\max_{y' \in \{a_{z-1},\ldots,a_z-1\}} S_z(y') + I(y' \to a_z)$
15 $a'_k \leftarrow M$
16 **for** $z \leftarrow k-1$ *to* 1 **do**
17 $a'_z \leftarrow h_{z+1}(a'_{z+1})$

Thanks to the necessary condition in Corollary 3, in single-step splitting algorithm (Algorithm 1) the state value $S_z(y)$ is only calculated for $y \in \{a_{z-1}, \cdots, a_z\}$ and for each y the maximization is taken over $y' \in \{a_{z-2}, \cdots, a_{z-1}\}$. The value $S_k(M)$ gives the maximum mutual information obtained by optimal k-level quantizer with boundaries $\{a'_z\}_{z=0}^{k}$.

The recursive splitting algorithm (Algorithm 2) is a top-down algorithm that starts with a trivial single level quantizer with boundaries $\{0, M\}$ and at each step increases the quantizer cardinality k by one and designs the optimal quantizer conditioning on the boundary values of the previous step and stops when it reaches an α-fraction of the original mutual information. In this algorithm, the quantizer boundaries for k-th level is denoted by $\{a_z^{(k)}\}_{z=0}^{k}$. In the for loop of line 10, the state value $S_z(y)$ is only calculated for $y \in \{a_{z-1}^{(k-1)}, \cdots, a_{z-1}^{(k-2)} - 1\}$ and for each y the maximization is taken over $y' \in \{a_{z-2}^{(k-1)}, \cdots, h_z(a_{z-1}^{(k-2)})\}$.

Algorithms 1 and 2 can be modified to obtain counterpart algorithms for a bottom-up approach using the boundary conditions in (24).

Algorithm 2 Splitting Algorithm (Recursive)

Input: M, p_x, $P(y|x)$, α
Output: $\{a_z^{(k)}\}_{z=0}^{k}$ for all $2 \le k \le K^*$

1 $I_{XY} \leftarrow \sum_{y=0}^{M-1} I(y \to y+1)$
2 $a_1^{(1)} \leftarrow M$, $a_0^{(1)} \leftarrow 0$
3 **for** $y \leftarrow 1$ *to* $M-1$ **do**
4 $S_1(y) \leftarrow I(0 \to y)$, $h_1(y) \leftarrow 0$
5 $S_2(M) \leftarrow \max_{y' \in \{1,\ldots,M-1\}} S_1(y') + I(y' \to M)$
6 $h_2(M) \leftarrow \arg\max_{y' \in \{1,\ldots,M-1\}} S_1(y') + I(y' \to M)$
7 $a_2^{(2)} \leftarrow M$, $a_1^{(2)} \leftarrow h_2(M)$, $a_0^{(2)} \leftarrow 0$, $k \leftarrow 2$
8 **while** $S_k(M) < \alpha I_{XY}$ **do**
9 $k \leftarrow k+1$
10 **for** $z \leftarrow 2$ *to* $k-1$ **do**
11 $lb \leftarrow \max(z-1, a_{z-2}^{(k-1)})$
12 $ub \leftarrow \min(h_z(a_{z-1}^{(k-2)}), a_{z-1}^{(k-1)} - 1)$
13 $\mathcal{Y}' \leftarrow \{lb, \ldots, ub\}$
14 $S_z(a_{z-1}^{(k-1)}) \leftarrow \max_{y' \in \mathcal{Y}'} S_{z-1}(y') + I(y' \to a_{z-1}^{(k-1)})$
15 $h_z(a_{z-1}^{(k-1)}) \leftarrow \arg\max_{y' \in \mathcal{Y}'} S_{z-1}(y') + I(y' \to a_{z-1}^{(k-1)})$
16 **for** $y \leftarrow a_{z-1}^{(k-1)} + 1$ *to* $a_{z-1}^{(k-2)} - 1$ **do**
17 $\mathcal{Y}' \leftarrow \{a_{z-2}^{(k-1)} + 1, \ldots, h_z(a_{z-1}^{(k-2)})\}$
18 $S_z(y) \leftarrow \max_{y' \in \mathcal{Y}'} S_{z-1}(y') + I(y' \to y)$
19 $h_z(y) \leftarrow \arg\max_{y' \in \mathcal{Y}'} S_{z-1}(y') + I(y' \to y)$
20 $S_k(M) \leftarrow \max_{y' \in \{a_{k-2}^{(k-1)}, \ldots, M-1\}} S_z(y') + I(y' \to M)$
21 $h_k(M) \leftarrow \arg\max_{y' \in \{a_{k-2}^{(k-1)}, \ldots, M-1\}} S_z(y') + I(y' \to M)$
22 $K^* \leftarrow k$, $a_k^{(k)} \leftarrow M$
23 **for** $z \leftarrow k-1$ *to* 1 **do**
24 $a_z^{(k)} \leftarrow h_{z+1}(a_{z+1}^{(k)})$

B. Complexity

First we analyze the complexity of the single-step algorithm. For the case of $k = 2$, the splitting algorithm calculates the maximum of the row vector

$$\mathsf{M}_2 = \Big(S_1(1) + I(1 \to M) \quad S_1(2) + I(2 \to M) \quad \cdots \quad S_1(M-1) + I(M-1 \to M) \Big)$$

which consists of $M-1$ operations. For the case of $k > 2$, for $z = k$ the algorithm finds the maximum of the row vector

$$\mathsf{M}_k = \Big(S_{k-1}(a_{k-2}) + I(a_{k-2} \to M) \quad \cdots \quad S_{k-1}(M-1) + I(M-1 \to M) \Big).$$

For each $2 \le z \le k-1$ the algorithm finds all the row maxima of the matrix M_z on the bottom of Page 5075.

Therefore, since $\sum_{z=1}^{k-1} b_z = M$, the single-step algorithm performs a total of $b_{k-1} + \sum_{z=2}^{k-1} b_z b_{z-1} \le (\frac{M-k+2}{2})^2 + M$ operations, which has a worst-case complexity of $O\big(\frac{M^2}{4}\big)$.

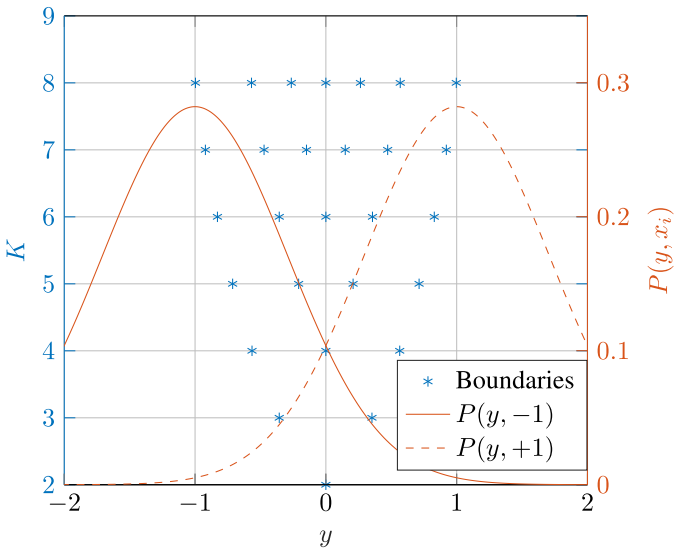

Fig. 2. Optimal quantization of a finely quantized AWGN channel (with uniform ± 1 inputs and $\sigma^2 = 0.5$) to $k = 2$ to $k = 8$ levels using the recursive splitting algorithm.

Iwata and Ozawa in [3] showed that the partial mutual information $I(y' \to y)$ has the following property: for $1 \leq i < r \leq j < s \leq M$

$$I(i \to j) + I(r \to s) \geq I(i \to s) + I(r \to s), \qquad (28)$$

and therefore the matrix M_z is an inverse Monge matrix and hence a totally monotone matrix [19]. This property allows us to use SMAWK algorithm to find all the row maxima of M_z with $c_1 b_{z-1} + c_2 b_z$ operations for some constants c_1 and c_2 [20]. Therefore, using the SMAWK algorithm the number of operations of single-step splitting algorithm reduces to $b_{k-1} + \sum_{z=2}^{k-1}(c_1 b_{z-1} + c_2 b_z) < (c_1 + c_2)M$, which has complexity $O(M)$.

In order to analyze the complexity of the recursive splitting algorithm let us assume that it reaches the α-fraction of the original mutual information at $k = K^*$. Comparing the single-step and the recursive splitting versions one can see that the for loop in line 13 of the recursive algorithm runs for fewer y values (since $a_{z-1}^{(k-2)} \leq a_z^{(k-1)}$), as it avoids recalculating the values $S_z(y)$ obtained in previous recursions. Therefore the number of operations done by the recursive version with the SMAWK algorithm is less than $\sum_{k=2}^{K^*}(c_1 + c_2)M$ and has complexity $O(K^* M)$.

C. Example: Finely Quantized Continuous Output Channel

We consider a binary-input AWGN channel with equally likely ± 1 inputs and Gaussian noise variance $\sigma^2 = 0.5$. We first uniformly quantize the output of the AWGN channel

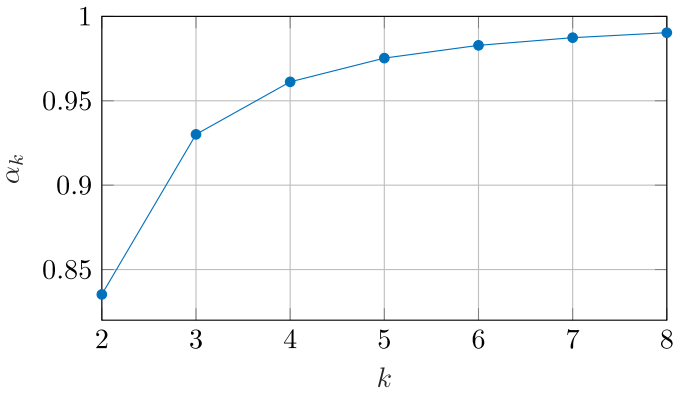

Fig. 3. Mutual information fraction preserved by the optimal quantizers with $k = 2$ to $k = 8$ levels for a finely quantized AWGN channel (with uniform ± 1 inputs and $\sigma^2 = 0.5$).

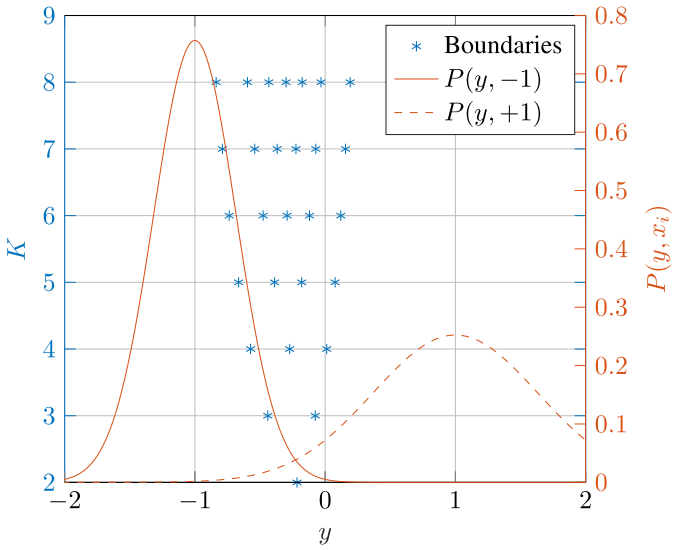

Fig. 4. Optimal recursive quantization of a finely quantized asymmetric Gaussian channel (with $p(-1) = 0.6$, $p(+1) = 0.4$, $\sigma_{-1}^2 = 0.1$ and $\sigma_{+1}^2 = 0.4$) to $k = 2$ to $k = 8$ levels.

y between -2 and 2 with $M = 1000$ levels; the natural order of the outputs of the resulting DMC satisfies (9). Later we apply the recursive splitting algorithm to find a quantizer with minimum output levels which preserves 99% ($\alpha = 0.99$) of the mutual information of the original AWGN. Fig. 2 shows the quantization boundaries for the optimal quantizers (of underlying DMC) with 2 to 8 outputs. The results match the algorithm in [2]. The optimal quantizer with $K^* = 8$ outputs satisfies the mutual information constraint (Fig. 3). As the channel and inputs are symmetric, the optimal quantizers are symmetric around $y = 0$ as well.

Next we consider an asymmetric Gaussian channel with -1 and $+1$ inputs and respective probabilities 0.6 and 0.4 and input-dependent Gaussian noise with variances $\sigma_{-1}^2 = 0.1$ and

$$\mathsf{M}_z = \begin{pmatrix} S_{z-1}(a_{z-2}+1) + I(a_{z-2}+1 \to a_{z-1}) & \cdots & S_{z-1}(a_{z-1}-1) + I(a_{z-1}-1 \to a_{z-1}) & 0 \\ S_{z-1}(a_{z-2}+1) + I(a_{z-2}+1 \to a_{z-1}+1) & \cdots & S_{z-1}(a_{z-1}-1) + I(a_{z-1}-1 \to a_{z-1}+1) & S_{z-1}(a_{z-1}) + I(a_{z-1} \to a_{z-1}+1) \\ \vdots & & \vdots & \vdots \\ S_{z-1}(a_{z-2}+1) + I(a_{z-2}+1 \to a_z - 1) & \cdots & S_{z-1}(a_{z-1}-1) + I(a_{z-1}-1 \to a_z - 1) & S_{z-1}(a_{z-1}) + I(a_{z-1} \to a_z - 1) \end{pmatrix}$$

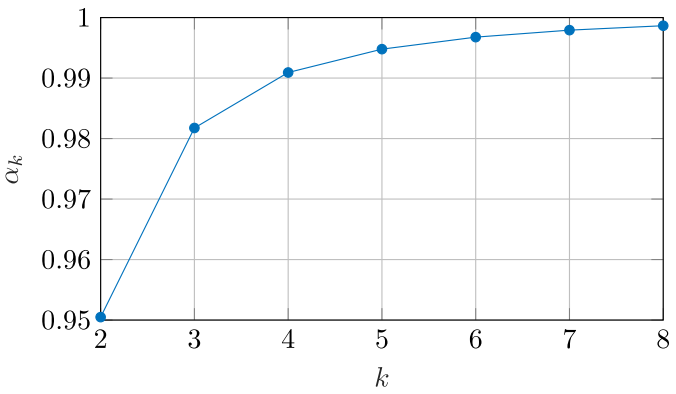

Fig. 5. Mutual information fraction preserved by the optimal quantizers with $k = 2$ to $k = 8$ levels for a finely quantized asymmetric Gaussian channel (with $p(-1) = 0.6$, $p(+1) = 0.4$, $\sigma^2_{-1} = 0.1$ and $\sigma^2_{+1} = 0.4$).

$\sigma^2_{+1} = 0.4$ respectively. The recursive splitting algorithm is run until $\delta_{\alpha_k} = \frac{I(X;Z_k)-I(X;Z_{k-1})}{I(X;Y)} < 0.001$. Fig. 4 shows the quantization boundaries for the optimal quantizers designed recursively. Since the channel is asymmetric, the optimal quantizers are asymmetric as well. Interestingly, as shown in Fig. 4, the central boundary $a_{\frac{k}{2}}$ (for even k) moves further away from zero as k increases. The fraction of preserved mutual information α_k by the optimal quantizers is illustrated in Fig. 5 which shows that δ_{α_k} decreases by increasing k and it goes below 0.001 at $k = 8$.

V. Conclusion

We have studied the problem of finding a quantizer Q_α with smallest cardinality that preserves a constant α-fraction of the mutual information for binary-input discrete memoryless channels. Since direct optimization of the quantizer cardinality is not feasible, two dual bottom-up and top-down approaches to find Q_α are proposed. Based on these approaches, a new necessary optimality condition for the recursive quantizer design is obtained. A recursive splitting algorithm based on dynamic programming as a modification of quantizer design algorithm in [2] is proposed which incorporates the new necessary condition and finds Q_α with complexity $O(K^*M)$ using the acceleration with the SMAWK algorithm. Our results suggest that the recursive quantizer design not only provides a full picture of the preserved fraction of mutual information versus cardinality of optimal quantizers, but also it reduces the complexity of the design process.

Appendix A
Proof of Lemma 1

Let us denote new outputs resulting from (h, j) merge, and (i, j) merge as y'_{hj} and y'_{ij} and their conditional posterior probabilities as v_{hj} and v_{ij}, respectively. We have the following

$$v_{hj} = \frac{(\pi_h v_h + \pi_j v_j)}{\pi_h + \pi_j} \rightarrow \frac{\pi_h}{\pi_j} = \frac{v_j - v_{hj}}{v_{hj} - v_h}, \tag{29}$$

$$v_{ij} = \frac{(\pi_i v_i + \pi_j v_j)}{\pi_i + \pi_j} \rightarrow \frac{\pi_i}{\pi_j} = \frac{v_j - v_{ij}}{v_{ij} - v_i}. \tag{30}$$

Now let us assume that

$$\frac{\pi_h}{\pi_j} = \frac{v_j - v_{hj}}{v_{hj} - v_h} \geq \frac{v_j - v_i}{v_i - v_h}, \tag{31}$$

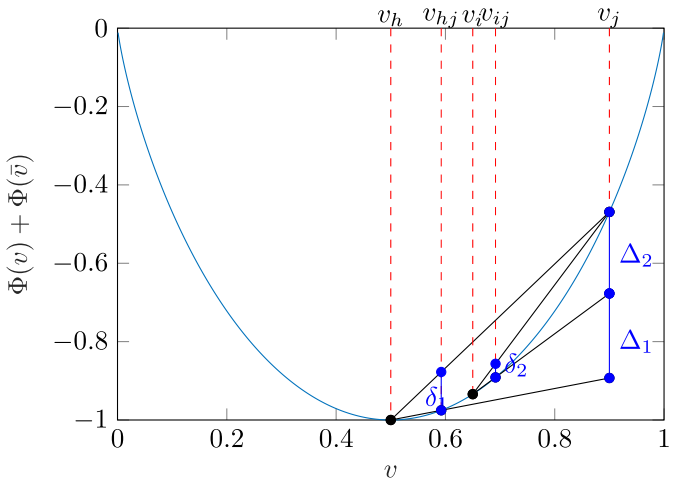

Fig. 6. Illustration of $\Delta I(h, j)$ and $\Delta I(i, j)$.

therefore, $v_{hj} \leq v_i$. With this assumption, we will show that the mutual information loss (17) is larger for a (h, j) merge than for a (i, j) merge. With some algebraic manipulations, we express the mutual information loss for a (h, j) merge as

$$\begin{aligned}\Delta I(h, j) = {} & \pi_h\big(\Phi(v_h) + \Phi(\bar{v}_h)\big) + \pi_j\big(\Phi(v_j) + \Phi(\bar{v}_j)\big) \\ & - (\pi_h + \pi_j)\big(\Phi(v_{hj}) + \Phi(\bar{v}_{hj})\big) > \Delta I(i, j),\end{aligned} \tag{32}$$

where $\bar{v} = 1 - v$.

Fig. 6 illustrates (32) where,

$$\delta_1 = \frac{\Delta I(h, j)}{\pi_h + \pi_j}, \quad \delta_2 = \frac{\Delta I(i, j)}{\pi_i + \pi_j}. \tag{33}$$

We have the following relations on the triangles in Fig. 6,

$$\frac{\delta_1}{\Delta_1 + \Delta_2} = \frac{v_{hj} - v_h}{v_j - v_h} = \frac{\pi_j}{\pi_h + \pi_j}, \tag{34}$$

$$\frac{\delta_2}{\Delta_2} = \frac{v_{ij} - v_i}{v_j - v_i} = \frac{\pi_j}{\pi_i + \pi_j}, \tag{35}$$

where the second equalities come from (29) and (30). Notice that $\Delta_1 > 0$, since $v_{hj} \leq v_i$ and $\Phi(v) + \Phi(\bar{v})$ is a strictly convex function. Using (34) and (35) in (33) we have

$$\Delta I(h, j) = \pi_j(\Delta_1 + \Delta_2) > \pi_j \Delta_2 = \Delta I(i, j), \tag{36}$$

which proves (32).

If we assume the complementary inequality in (31), then $v_{hj} \geq v_i$ and with similar steps we show that $\Delta I(h, j) = \pi_h(\Delta'_1 + \Delta'_2) > \pi_h \Delta'_2 = \Delta I(h, i)$ where Δ'_1 and Δ'_2 will be the base of triangles at v_h. This completes the proof.

Appendix B
Proof of Theorem 1

It is sufficient to prove that the mutual information loss (12) corresponding to the $\hat{Q}_k$ and $\hat{Q}_{k-1}$ (15) satisfies $\Delta I_k \leq \Delta I_{k-1}$. There are two possibilities for the merge $\hat{Q}_{k-1}$. The merge may combine two outputs that were not changed due to $\hat{Q}_k$, for which it is clear that the corresponding mutual information loss ΔI_{k-1} cannot be smaller than ΔI_k, since otherwise that merge should be selected by greedy algorithm for $\hat{Q}_k$. Alternatively, one of the outputs results from $\hat{Q}_k$.

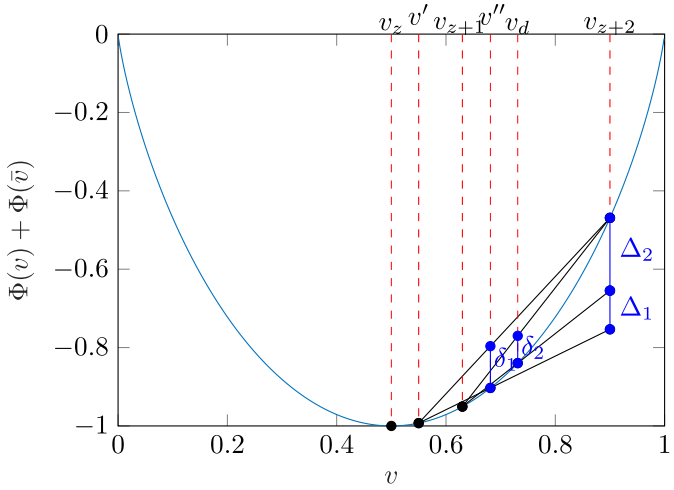

Fig. 7. Illustration of $\Delta I(z', z+2)$ and $\Delta I(z+1, z+2)$.

Let us consider three outputs $z, z+1, z+2 \in \mathcal{Z}_{k+1}$ with $v_z < v_{z+1} < v_{z+2}$. Without loss of generality, we can assume that the greedy merging performs $(z, z+1)$ merge as $\hat{Q}_k$ and later merges the resulting output z' with $z+2$ as $\hat{Q}_{k-1}$. Equivalently, we know that $\Delta I(z, z+1) \leq \Delta I(z+1, z+2)$ and we want to show that $\Delta I(z, z+1) < \Delta I(z', z+2)$. Hence, it is sufficient to show that $\Delta I(z+1, z+2) < \Delta I(z', z+2)$.

Let us denote the new outputs resulting from the $(z', z+2)$ and $(z+1, z+2)$ merges as z'' and z_d respectively and their conditional posterior probabilities as v'' and v_d, respectively. We have the following implications:

$$v'' = \frac{(\pi_{z'} v' + \pi_{z+2} v_{z+2})}{\pi_{z'} + \pi_{z+2}} \rightarrow \frac{\pi_{z'}}{\pi_{z+2}} = \frac{v_{z+2} - v''}{v'' - v'}, \tag{37}$$

$$v_d = \frac{(\pi_{z+1} v_{z+1} + \pi_{z+2} v_{z+2})}{\pi_{z+1} + \pi_{z+2}} \rightarrow \frac{\pi_{z+1}}{\pi_{z+2}} = \frac{v_{z+2} - v_d}{v_d - v_{z+1}}. \tag{38}$$

Using (37) and (38) and since $\pi_{z'} = \pi_z + \pi_{z+1} > \pi_{z+1}$ and $v' = \frac{(\pi_z v_z + \pi_{z+1} v_{z+1})}{\pi_z + \pi_{z+1}} < v_{z+1}$, we have the following strict inequality

$$\begin{aligned} v_{z+2} - v'' &= \frac{\pi_{z'}}{\pi_{z'} + \pi_{z+2}} (v_{z+2} - v') \\ &> \frac{\pi_{z+1}}{\pi_{z+1} + \pi_{z+2}} (v_{z+2} - v_{z+1}) = v_{z+2} - v_d, \end{aligned} \tag{39}$$

and therefore $v'' < v_d$.

Fig. 7 illustrates $\Delta I(z', z+2)$ and $\Delta I(z+1, z+2)$, where

$$\delta_1 = \frac{\Delta I(z', z+2)}{\pi_{z'} + \pi_{z+2}}, \quad \delta_2 = \frac{\Delta I(z+1, z+2)}{\pi_{z+1} + \pi_{z+2}}. \tag{40}$$

We have the following relations on the triangles on Fig. 7

$$\frac{\delta_1}{\Delta_1 + \Delta_2} = \frac{v'' - v'}{v_{z+2} - v'} = \frac{\pi_{z+2}}{\pi_{z'} + \pi_{z+2}}, \tag{41}$$

$$\frac{\delta_2}{\Delta_2} = \frac{v_d - v_{z+1}}{v_{z+2} - v_{z+1}} = \frac{\pi_{z+2}}{\pi_{z+1} + \pi_{z+2}}. \tag{42}$$

where the second equalities come from (37) and (38). Notice that $\Delta_1 > 0$, since $\Phi(v) + \Phi(\bar{v})$ is strictly convex. Therefore, using (40), (41) and (42) we have

$$\begin{aligned} \Delta I(z', z+2) &= \pi_{z+2}(\Delta_1 + \Delta_2) > \pi_{z+2} \Delta_2 \\ &= \Delta I(z+1, z+2). \end{aligned} \tag{43}$$

This completes the proof.

Appendix C
Proof of Theorem 2

Assume that the claim is not true and there is at least one k, $1 < k \leq M$ for which the modified greedy merging algorithm does not find all the optimal k-level quantizers, despite using all the optimal $(k+1)$-level quantizers as a seed. Therefore, there is an optimal k-level quantizer $\tilde{Q}$ such that $\tilde{Q} \notin \{\mathcal{Q}_{\mathrm{c},k} \cup \mathcal{Q}_{\mathrm{m},k}\}$. Since $\tilde{Q}$ cannot be generated by *contraction* or pairwise merge from any optimal $(k+1)$-level quantizer, it is generated by a different single-step quantizer which includes at least one of the following operations:

1. Splitting a boundary output into two parts and merging it from one side: It is clear that merging it from both sides is not possible since it is a boundary output. This operation keeps the same number of quantizer outputs while reducing its mutual information, hence it generates a non-optimal $(k+1)$-level quantizer and does not create new split and merge possibilities for the rest of outputs with lower mutual information loss.

Assume splitting z_1 to two parts, $z_{1L} = \{1, \ldots, s\}$ and $z_{1R} = \{s+1, \ldots, a_1\}$ for $1 \leq s \leq a_1 - 1$, and merging z_{1R} with z_2 resulting in z_2'. This new $(k+1)$-level quantizer is suboptimal and has lower mutual information than the original one. Furthermore, the only new possibilities of split and merge to reduce it to a k-level quantizer are those of splitting z_2' such that $z_{2L}' \subset z_{1R}$, and merging z_{2L}' with z_{1L} and z_{2R}' with z_3 which indeed has higher mutual information loss than simply merging z_2 with z_3 from the original k-level quantizer since $z_2 \subset z_{2R}'$. Therefore, the boundary outputs should not split during the reduction to the k-th level.

2. Splitting a non-boundary output into two parts and merging it only from one side: As the previous item, this operation also keeps the same number of quantizer outputs while reducing its mutual information, hence it generates a non-optimal $(k+1)$-level quantizer and does not create new split and merge possibilities for the rest of outputs with lower mutual information loss. This can be shown with similar arguments as for the previous statement. Therefore, after splitting a non-boundary output, both parts should be merged to their corresponding neighbor outputs (or part of it).

3. Splitting a non-boundary output to three or more than three parts and merging the left part with the left output and right part with the right output only decreases the mutual information of the quantizer keeping same number of outputs and does not create new split and merge possibilities for the rest of outputs with lower mutual information loss. Again, this can be shown with similar arguments as for the first statement. Therefore, a non-boundary output should not be split to more than two parts.

Including at least one of above three operations contradicts the optimality of $\tilde{Q}$ and concludes the proof.

Appendix D
Proof of Theorem 4

It is sufficient to show that $\Delta I_k^* \geq \Delta I_{k+1}^*$. Assume that Q_k^*, Q_{k+1}^* and Q_{k+2}^* are optimal quantizers and that the latter can be obtained from the former with an *expansion*. Since these optimal quantizers should have convex preimages, we can

find two distinct *expansion* that applying both of them on the outputs of Q_k^* would result in Q_{k+2}^* and applying each one of them on the outputs of would result in a $(k+1)$-level quantizer (given by $Q_{1,k+1}$ and $Q_{2,k+1}$). Now assume that $\Delta I_k^* < \Delta I_{k+1}^*$, hence, at least one of the $Q_{1,k+1}$ or $Q_{2,k+1}$ should have a higher mutual information than Q_{k+1}^* which contradicts the optimality of Q_{k+1}^*. Hence, the assumption is not true and $\Delta I_k^* \geq \Delta I_{k+1}^*$.

REFERENCES

[1] M. Dabirnia, A. Martinez, and A. Guillén i Fàbregas, "A recursive algorithm for quantizer design for binary-input discrete memoryless channels," in *Proc. Int. Zürich Seminar Inf. Commun. (IZS)*, Zürich, Switzerland, 2020, pp. 41–45.

[2] B. M. Kurkoski and H. Yagi, "Quantization of binary-input discrete memoryless channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4544–4552, Aug. 2014.

[3] K. Iwata and S. Ozawa, "Quantizer design for outputs of binary-input discrete memoryless channels using SMAWK algorithm," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jun. 2014, pp. 191–195.

[4] J. A. Zhang and B. M. Kurkoski, "Low-complexity quantization of discrete memoryless channels," in *Proc. Int. Symp. Inf. Theory Appl.*, Monterey, CA, USA, 2016, pp. 448–452.

[5] B. M. Kurkoski and H. Yagi, "Single-bit quantization of binary-input, continuous-output channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, p. 2088.

[6] X. He, K. Cai, W. Song, and Z. Mei, "Dynamic programming for sequential deterministic quantization of discrete memoryless channels," *IEEE Trans. Commun.*, early access, Mar. 1, 2021, doi: 10.1109/TCOMM.2021.3062838.

[7] T. D. Nguyen and T. Nguyen, "On binary quantizer for maximizing mutual information," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5435–5445, Sep. 2020.

[8] T. Nguyen and T. Nguyen, "Structure of optimal quantizer for binary-input continuous-output channels with output constraints," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 1450–1455.

[9] H. Yagi and B. M. Kurkoski, "Channel quantizers that maximize random coding exponents for binary-input memoryless channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Ottawa, ON, Canada, Jun. 2012, pp. 2256–2260.

[10] B. Nazer, O. Ordentlich, and Y. Polyanskiy, "Information-distilling quantizers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Aachen, Germany, Jun. 2017, pp. 96–100.

[11] A. Bhatt, B. Nazer, O. Ordentlich, and Y. Polyanskiy, "Information-distilling quantizers," *IEEE Trans. Inf. Theory*, vol. 67, no. 4, pp. 2472–2487, Apr. 2021.

[12] B. M. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Noise thresholds for discrete LDPC decoding mappings," in *Proc. IEEE Global Telecommun. Conf.*, New Orleans, LO, USA, Nov./Dec. 2008, pp. 1–5.

[13] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, Oct. 2013.

[14] M. Dabirnia, A. Martinez, and A. Guillén i Fàbregas, "A mismatched decoding perspective of channel output quantization," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Visby, Sweden, Aug. 2019, pp. 1–4.

[15] D. Burshtein, V. D. Pietra, D. Kanevsky, and A. Nádas, "Minimum impurity partitions," *Ann. Statist.*, vol. 20, no. 3, pp. 1637–1646, Sep. 1992.

[16] N. Slonim and N. Tishby, "Agglomerative information bottleneck," in *Proc. Neural Inf. Process. Syst. (NIPS)*, Denver, CO, USA, 1999, pp. 617–623.

[17] E. Laber, M. Molinaro, and F. M. Pereiram, "Binary partitions with approximate minimum impurity," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 2854–2862.

[18] P. A. Chou, "Optimal partitioning for classification and regression trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 340–354, Apr. 1991.

[19] R. E. Burkard, B. Klinz, and R. Rudolf, "Perspectives of Monge properties in optimization," *Discrete Appl. Math.*, vol. 70, no. 2, pp. 95–161, Sep. 1996.

[20] A. Aggarwal, M. M. Klawe, S. Moran, P. Shor, and R. Wilber, "Geometric applications of a matrix-searching algorithm," *Algorithmica*, vol. 2, no. 2, p. 195–208, Nov. 1987.

Mehdi Dabirnia (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Tehran, in 2007 and 2010, respectively, and the Ph.D. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in December 2017. From 2013 to 2017, he was a Research Assistant with the Communication Theory and Applications Research Laboratory, Bilkent University. Since 2018, he has been a Postdoctoral Researcher with the Information Theory and Coding Group, Universitat Pompeu Fabra (UPF), Barcelona, Spain. His Ph.D. research was focused on the design and analysis of capacity approaching coding solutions for energy harvesting and multiuser communications. He is interested in researching fundamental theoretical limits of communication systems, and developing practical methods and algorithms to approach those limits.

Alfonso Martinez (Senior Member, IEEE) received the degree in telecommunications engineering from the University of Zaragoza in 1997. From 1998 to 2003, he was a Systems Engineer with the Research Centre of the European Space Agency (ESA-ESTEC), Noordwijk, The Netherlands. His work on APSK modulation was instrumental in the definition of the physical layer of DVB-S2. From 2003 to 2007, he was a Research and Teaching Assistant with the Technische Universiteit Eindhoven, The Netherlands, where he conducted research on digital signal processing for MIMO optical systems and on optical communication theory. From 2008 to 2010, he was a Post-Doctoral Fellow with the Information-theoretic Learning Group, Centrum Wiskunde & Informatica (CWI), Amsterdam, The Netherlands. In 2011, he was a Research Associate with the Signal Processing and Communications Laboratory, Department of Engineering, University of Cambridge, Cambridge, U.K. He was a Ramón y Cajal Research Fellow with UPF, from 2012 to 2016, and an Adjunct Associate Professor, from 2017 to 2019. In this period, his research focused on mismatched decoding theory and on the analysis of finite-length communication systems by means of saddlepoint approximations. He is currently a Serra Húnter Associate Professor with Universitat Pompeu Fabra (UPF), Barcelona, Spain. More generally, he is intrigued by the connections between information theory, optical communications, and physics, particularly by the links between classical and quantum information theory. In the following area, he has coauthored a monograph "*Bit-Interleaved Coded Modulation*". His research interests include the fields of information theory and coding, with emphasis on digital modulation and the analysis of mismatched decoding.

Albert Guillén i Fàbregas (Senior Member, IEEE) received the degree in telecommunications engineering from the Universitat Politècnica de Catalunya and in electronics engineering from the Politecnico di Torino, in 1999, and the Ph.D. degree in communication systems from the École Polytechnique Fédérale de Lausanne (EPFL) in 2004.

In 2020, he returned to a full-time faculty position with the Department of Engineering, University of Cambridge, where he had been a full-time faculty and a fellow of Trinity Hall, from 2007 to 2012. Since 2011, he has been an ICREA Research Professor with Universitat Pompeu Fabra (currently on leave). He has held appointments at the New Jersey Institute of Technology, Telecom Italia, the European Space Agency (ESA), the Institut Eurécom, the University of South Australia, Universitat Pompeu Fabra, and the University of Cambridge, and visiting appointments at EPFL, the École Nationale des Télécommunications (Paris), Universitat Pompeu Fabra, the University of South Australia, the Centrum Wiskunde & Informatica, and Texas A&M University, Qatar. His specific research interests include the areas of information theory, communication theory, coding theory, and statistical inference.

Dr. Fàbregas is a member of the Young Academy of Europe. He received the Starting and Consolidator Grants from the European Research Council, the Young Authors Award of the 2004 European Signal Processing Conference, the 2004 Best Doctoral Thesis Award from the Spanish Institution of Telecommunications Engineers, and a Research Fellowship of the Spanish Government to join ESA. Since 2013, he has been an Editor of the *Foundations and Trends in Communications and Information Theory*, where he is a Publisher, and was an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY for the term 2013–2020, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS for the term 2007–2011.