

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

Mehdi Dabirnia
Universitat Pompeu Fabra
mehdi.dabirnia@upf.edu

Alfonso Martinez
Universitat Pompeu Fabra
alfonso.martinez@ieee.org

Albert Guillén i Fàbregas
ICREA and Universitat Pompeu Fabra
University of Cambridge
guillen@ieee.org

Abstract—The quantization of the outputs of a binary-input discrete memoryless channel is considered. A new recursive method for finding all optimal quantizers for all output cardinalities is proposed. Two different versions of the newly proposed method for top-down and bottom-up approaches are developed which provide an improved understanding of the quantization problem under consideration. Also, an efficient algorithm based on dynamic programming is proposed and shown to have a comparable complexity with the state of the art.

I. INTRODUCTION

Quantization has practical applications in hardware implementations of communication systems, e.g., from channel output quantization to message passing decoders [1] and polar code construction [2]. In such applications, there is a trade-off between performance and complexity of the system represented by the number of quantization levels. Therefore, it is of interest to use as few quantization levels as possible while maintaining reliable communication with a given transmission rate.

Recently we studied channel output quantization from a mismatched-decoding perspective [3]. This study showed 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 [4] or error exponent [5].

Consider a discrete memoryless channel (DMC) followed by a quantizer at the output, as shown in Fig. 1. The channel input X takes values in $\mathcal{X} = \{1, \dots, J\}$ with probability distribution $p_x = \Pr(X = x)$, and the channel output Y takes values in $\mathcal{Y} = \{1, \dots, M\}$, with channel transition probabilities $W_{y|x} = \Pr(Y = y|X = x)$. The channel output is quantized to $Z^{(K)}$, which takes values in $\mathcal{Z}^{(K)} = \{z_1^{(K)}, \dots, z_K^{(K)}\}$, by a possibly stochastic quantizer $Q_{z|y} = \Pr(Z^{(K)} = z|Y = y)$. The conditional probability distribution of the quantizer output given the channel input is $T_{z|x} = \Pr(Z^{(K)} = z|X = x) = \sum_{y \in \mathcal{Y}} Q_{z|y} W_{y|x}$.

The mutual information between X and $Z^{(K)}$ is

$$I(X; Z^{(K)}) = \sum_{z \in \mathcal{Z}^{(K)}} \sum_{x \in \mathcal{X}} p_x T_{z|x} \log \frac{T_{z|x}}{\sum_{x'} p_{x'} T_{z|x'}}. \quad (1)$$

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.

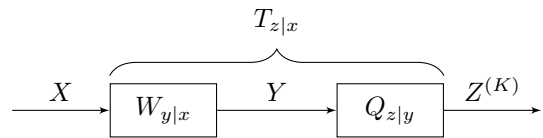


Fig. 1: A discrete memoryless channel followed by a quantizer.

Let us denote the set of all possible quantizers Q with K outputs, including stochastic quantizers, with $\mathcal{Q}^{(K)}$. We formulate the quantizer optimization as follows: for a given constant $0 \leq \alpha \leq 1$, we want to find an optimal quantizer Q_α^* with the smallest cardinality K from the set \mathcal{S} defined as

$$\mathcal{S} \triangleq \{Q \in \mathcal{Q}^{(K)} : 1 \leq K \leq M, I(X; Z^{(K)}) \geq \alpha I(X; Y)\}. \quad (2)$$

The optimal quantizer Q_α^* preserves at least an α -portion of the original mutual information with the smallest number of quantization levels K .

II. BACKGROUND AND CONTRIBUTION

For a fixed output cardinality K , Kurkoski and Yagi showed that there is a deterministic quantizer that maximizes the mutual information (1) between channel input and quantized output [4]. Therefore, considering only deterministic quantizers is sufficient to find the optimal quantizer Q_α^* . A deterministic quantizer Q maps each output y to only one quantized output $z_k^{(K)}$, $Q : \{1, \dots, M\} \rightarrow \{z_1^{(K)}, \dots, z_K^{(K)}\}$, therefore, the corresponding probabilistic map $Q_{z|y}$ takes only values 0 or 1. We define the pre-image of $z_k^{(K)}$ as

$$\mathcal{A}(z_k^{(K)}) = \{y \in \mathcal{Y} : Q^{-1}(z_k^{(K)}) = y\} \quad (3)$$

which is the set of channel outputs that are mapped to $z_k^{(K)}$. Hence, the deterministic quantizer $Q_{z|y}$ partitions \mathcal{Y} to K subsets $\{\mathcal{A}(z_1^{(K)}), \dots, \mathcal{A}(z_K^{(K)})\}$.

Let $P_{x|y} = \Pr(X = x|Y = y)$ be the posterior conditional probability distribution on the channel input which depends on the input distribution p_x and the channel conditional distribution $W_{y|x}$. For each channel output y , we define a vector v_y

$$v_y = [P_{1|y}, P_{2|y}, \dots, P_{J-1|y}] \quad (4)$$

with $v_y \in \mathcal{U} = [0, 1]^{J-1}$. Define an equivalent quantizer \tilde{Q} on the vectors $\{v_1, \dots, v_M\}$ as $\tilde{Q}(v_y) = Q(y) = z$.

Kurkoski and Yagi in [4, Lemma 2], using the results of [6], study a condition to find an optimal quantizer \tilde{Q}^* . They show the existence of an optimal quantizer \tilde{Q}^* for which two distinct preimages $\tilde{Q}^{*-1}(z)$ and $\tilde{Q}^{*-1}(z')$ are separated by a hyperplane in Euclidean space \mathcal{U} . Unfortunately, this condition does not offer a practical search method for quantizer design problem in general; however, as suggested in [4], it simplifies the problem for binary-input case.

The problem of finding Q_α^* can be tackled by either a *bottom-up* or *top-down* approach. The former starts with trivial partition into $K = M$ subsets, where each subset $\mathcal{A}(z_k^{(K)})$, $1 \leq k \leq K$ contains exactly one element of \mathcal{Y} . At each step, we decrease the cardinality K by one and design a quantizer 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., $\mathcal{A}(z_1^{(1)}) = \mathcal{Y}$. At each step, we increment the cardinality K by one and design a quantizer 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*, which means that the design is performed independent of the previous step result.

An example of a recursive bottom-up approach is the *agglomerative information bottleneck* [7] which has been rediscovered multiple times in the literature with different names such as *greedy merging* or *greedy combining* [1], [2]. This algorithm iteratively reduces the cardinality by merging two outputs into a new single output. At each iteration, the greedy algorithm evaluates all the possible pairwise merges and selects the one that minimizes the mutual information loss. Although this 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 $O(M^2)$ for a bottom-up design, resulting in a quantizer for each cardinality $1 \leq K \leq M$.

As for the independent approach, several quantizer 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 quantizer with time complexity $O(K(M-K)^2)$ [4]. Iwata and Ozawa [8] improved the complexity to $O(K(M-K))$ using the SMAWK algorithm. For the non-binary-input case, finding the optimal quantizer is an NP-hard problem [9], however several suboptimal algorithms are proposed in the literature. An example is KL-means quantizer [10] which is a variation of the K-means clustering algorithm by replacing Euclidean distance metric with Kullback-Leibler divergence. This algorithm has complexity $O(KMT)$ where T is the number of iterations that algorithm is run to converge to a local optimum. The complexity of 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.

In this paper, we focus on the binary-input case and we propose a recursive method for quantization of binary-input

DMCs that finds all the optimal quantizers. We develop two versions of the new method, one for top-down and the other for bottom-up approach. In addition, we propose an algorithm based on dynamic programming that has comparable complexity to the best known algorithm from the literature.

III. OPTIMAL RECURSIVE QUANTIZER

For the binary-input case, the posterior conditional probabilities $v_y = P(1|y)$ are in one-dimensional space $\mathcal{U} = [0, 1]$. Denote the output probabilities by $\pi_i = \Pr(Y = i)$. We assume that the outputs are relabelled to satisfy

$$P(1|1) < P(1|2) < \dots < P(1|M). \quad (5)$$

According to the [4, Lemma 3], there is an optimal quantizer Q^* such that preimages of the quantizer outputs consist of contiguous set of integers,

$$\mathcal{A}^*(z_k^{(K)}) = \{a_{k-1}^* + 1, \dots, a_k^*\} \quad (6)$$

for $z_k^{(K)} \in \mathcal{Z}^{(K)}$, with $a_0^* = 0$ and $a_{k-1}^* < a_k^*$ and $a_K^* = M$. The a_k^* 's are optimal quantizer boundaries which satisfy

$$0 < a_1^* < a_2^* < \dots < a_{K-1}^* < M. \quad (7)$$

Here we show that this condition is necessary for any optimal quantizer. Denote the mutual information loss corresponding to merging outputs j and l with $\Delta\iota(j, l)$ which is given by

$$\begin{aligned} \Delta\iota(j, l) = & \sum_{x \in \{1, 2\}} \pi_j \Phi(P(x|j)) + \pi_l \Phi(P(x|l)) \\ & - (\pi_j + \pi_l) \Phi(P(x|y_{jl})), \end{aligned} \quad (8)$$

where $\Phi(x) = x \log(x)$.

Lemma 1. *For binary-input DMC, assuming that the outputs are relabelled to satisfy (5), then for any choice of $1 \leq j < k < l \leq M$ at least one of the following is true,*

$$\begin{cases} \Delta\iota(j, k) < \Delta\iota(j, l) & \text{if } \frac{\pi_j}{\pi_l} \leq \frac{v_l - v_k}{v_k - v_j}, \\ \Delta\iota(k, l) < \Delta\iota(j, l) & \text{if } \frac{\pi_j}{\pi_l} \geq \frac{v_l - v_k}{v_k - v_j}. \end{cases} \quad (9)$$

The proof is in the Appendix. Lemma 1 shows that for any quantizer that does not satisfy the condition in (6), there is another quantizer satisfying this condition that has a higher mutual information. Therefore, based on this necessary condition, the quantizer design reduces to searching for the optimal boundaries a_k^* as in (7).

A. Modified Greedy Merging

The greedy merging algorithm [1], [2] reduces the output cardinality by performing the best pairwise merge at each step. It finds the optimal single-step quantizer by a greedy search, i.e.,

$$Q_m^{(i)} = \arg \min_{Q \in \mathcal{Q}_m^{(i)}} I(X; Z^{(i+1)}) - I(X; Z^{(i)}), \quad (10)$$

where $\mathcal{Q}_m^{(i)}$ is set of all possible single-step deterministic quantizers (pairwise merges) from $Z^{(i+1)}$ to $Z^{(i)}$.

In this section, we propose a new greedy algorithm which considers all pairwise merges and also another set of single-step quantizers which we denote them as *contractions*. A *contraction* is a single-step quantizer that consists of splits and merges. Next, we denote the definitions of split and merge and afterwards we define a *contraction*.

Definition 1 (Splitting an output). A quantizer output z_k with preimage $\mathcal{A}(z_k) = \{a_{k-1} + 1, \dots, a_k\}$ of size $b_k = |\mathcal{A}(z_k)| \geq 2$, splits into two non-empty parts z_{kL} (left) and z_{kR} (right) with preimages $\mathcal{A}(z_{kL}) = \{a_{k-1} + 1, \dots, s\}$ and $\mathcal{A}(z_{kR}) = \{s + 1, \dots, a_k\}$. This split can be done in $b_k - 1$ different ways, $a_{k-1} + 1 \leq s \leq a_k - 1$.

Definition 2 (Merging an split output). An split output z_k with two non-empty parts z_{kL} (left) and z_{kR} (right) is merged as:
1- z_{kL} merges with z_{k-1} (or $z_{(k-1)R}$ if it has been split too)
2- z_{kR} merges with z_{k+1} (or $z_{(k+1)L}$)

Contraction from K -level to $(K - 1)$ -level:

- 1) Input: a K -level quantizer with output boundaries $\{a_1, a_2, \dots, a_{K-1}\}$
- 2) Select a set of consecutive non-boundary outputs $\{z_j, z_{j+1}, \dots, z_l\}$ with $j > 1$, $l < K$ and $b_k = |\mathcal{A}(z_k)| \geq 2$ for all $j \leq k \leq l$.
- 3) Split each z_k according to Definition 1. This step can be done in $\prod_{k=j}^l (b_k - 1)$ different ways.
- 4) Merge z_{kR} with $z_{(k+1)L}$ for all $j \leq k \leq l-1$, also merge z_{j-1} with z_{jL} and z_{lR} with z_{l+1} .
- 5) Output: a $(K-1)$ -level quantizer with output boundaries $\{a'_1, \dots, a'_{K-2}\}$ for which $a_{k-1} < a'_{k-1} < a_k$ for all $j \leq k \leq l-1$.

Let us denote the set of all quantizers obtained by *contraction* as $\mathcal{Q}_c^{(K-1)}$.

As an example to illustrate *contraction*, consider a quantizer with 3 outputs with preimages $\mathcal{A}(z_1) = \{1, \dots, a_1\}$, $\mathcal{A}(z_2) = \{a_1 + 1, \dots, a_2\}$ and $\mathcal{A}(z_3) = \{a_2 + 1, \dots, M\}$. According to step 2 of *contraction*, the only possibility for a set of consecutive non-boundary outputs is $\{z_2\}$ if $b_2 = |\mathcal{A}(z_2)| \geq 2$. In step 3, we split z_2 into two parts $\mathcal{A}(z_{2L}) = \{a_1 + 1, \dots, s\}$ and $\mathcal{A}(z_{2R}) = \{s + 1, \dots, a_2\}$ where $a_1 + 1 \leq s \leq a_2 - 1$. We merge z_{2L} with z_1 and z_{2R} with z_3 according to step 4. The output of this *contraction* is a quantizer with 2 outputs that has the boundary $a'_1 = s$. The set of all $b_2 - 1$ possible *contractions* for this example are specified by $a_1 + 1 \leq s \leq a_2 - 1$.

Modified greedy merging starts from the trivial solution with M outputs and at each step performs a greedy search over all possible *contractions* $\mathcal{Q}_c^{(i)}$ and all pairwise merges $\mathcal{Q}_m^{(i)}$, selecting the one with lowest mutual information loss. At each step it keeps all the quantizers that have the highest mutual information and uses them as a seed for the next step.

Theorem 1. For the binary-input DMC, the modified greedy merging algorithm finds all optimal quantizers Q^* for all output cardinalities $1 \leq K \leq M$.

Due to space limitations, we omit the proof.

B. Modified Greedy Splitting

Modified greedy splitting is a top-down algorithm that is the dual of modified greedy merging. It starts from the trivial solution with a single output and at each step it increases the output cardinality by one, performing a greedy search over all possible *expansions*. It keeps all the quantizers that have the highest mutual information at each step and uses them as a seed for next step. In the following we define an *expansion* which consists of splits and merges.

Assume that we have a K -level quantizer which is specified by its boundaries $\{a_1, a_2, \dots, a_K\}$, we obtain a $(K + 1)$ -level quantizer by set of splits and merges according to following steps.

Expansion from K -level to $(K + 1)$ -level:

- 1) Input: a K -level quantizer with output boundaries $\{a_1, a_2, \dots, a_{K-1}\}$
- 2) Select a set of consecutive outputs $\{z_j, z_{j+1}, \dots, z_l\}$ with $j \geq 1$, $l \leq K$ and $b_k = |\mathcal{A}(z_k)| \geq 2$ for all $j \leq k \leq l$.
- 3) Split each z_k according to Definition 1. This step can be done in $\prod_{k=j}^l (b_k - 1)$ different ways.
- 4) If the size of selected set in Step 2 is one, omit this otherwise merge z_{kR} with $z_{(k+1)L}$ for all $j \leq k \leq l-1$.
- 5) Output: a $(K+1)$ -level quantizer with output boundaries $\{a'_1, \dots, a'_K\}$ for which $a_{k-1} < a'_k < a_k$ for all $j \leq k \leq l$.

Let us denote the set of all quantizers obtained by *expansions* as $\mathcal{Q}_e^{(K+1)}$.

As an example to illustrate *expansion*, consider a quantizer with 2 outputs with preimages $\mathcal{A}(z_1) = \{1, \dots, a_1\}$, $\mathcal{A}(z_2) = \{a_1 + 1, \dots, M\}$. An *expansion* for this example can be obtained in two different ways. The first one is simply by splitting one of the outputs z_1 or z_2 which can be performed in $b_1 - 1$ and $b_2 - 1$ different ways. The second one is by splitting both z_1 and z_2 and merging z_{1R} with z_{2L} . The latter can be performed in $(b_1 - 1)(b_2 - 1)$ different ways. The output of any such *expansion* is a quantizer with 3 outputs that has the boundaries $\{a'_1, a'_2\}$.

Theorem 2. For the binary-input DMC, the modified greedy splitting finds all optimal quantizers Q^* for all output cardinalities $1 \leq K \leq M$.

This theorem can be easily proved by showing the duality between *expansions* and *contractions* plus pairwise merges.

Note that the number of possible *contractions* and *expansions* increases polynomially as the number of outputs with large preimages increase. Therefore, the complexity of the modified greedy algorithms also grows polynomially. In the following we provide an algorithm based on dynamic programming which has quadratic complexity in the worst case.

C. Dynamic Programming Based Algorithm

This algorithm is a modified version of the Quantizer Design Algorithm [4] which is an instance of dynamic programming.

The assumption for this algorithm is that we already know the optimal K -level quantizer (which is specified by its boundaries $\{a_i\}_{i=0}^K$) and we want to find the optimal $(K + 1)$ -level quantizer employing the constraints imposed by *expansion* procedure on the resulting boundaries $\{a'_i\}_{i=0}^{K+1}$. The algorithm has a state value $S_z(y)$, which is the maximum partial mutual information when channel outputs 1 to y are quantized to quantizer outputs 1 to z . This can be computed recursively by conditioning on the state value at time index $z - 1$:

$$S_z(a) = \max_{a'} \left(S_{z-1}(a') + \iota(a' \rightarrow a) \right), \quad (11)$$

where $\iota(a' \rightarrow a)$ is the contribution that the quantizer output $z = \{a' \rightarrow a\}$ makes to the mutual information. It is called partial mutual information and is given by

$$\iota(a' \rightarrow a) = \sum_{x \in \mathcal{X}} P_x \sum_{y=a'+1}^a P_{y|x} \log \frac{\sum_{y'=a'+1}^a P_{y'|x}}{\sum_{x'} \sum_{y'=a'+1}^a P_{y'|x'}}. \quad (12)$$

There are constraints imposed by the *expansion* procedure on the set of states a' that needs to be considered in the maximization in (11). These constraints have a key role in simplifying the original Quantizer Design Algorithm [4].

Splitting Algorithm

1) Inputs

- Binary-input discrete memoryless channel $P_{y|x}$ re-labelled to satisfy (5).
- Input distribution P_x .
- Set of boundaries $\{a_i\}_{i=0}^K$ corresponding to the optimal K -level quantizer.

2) Precompute the partial mutual information. For each $0 \leq i \leq K - 1$,

- For $a' = a_i + 1$ and for each $a \in a_i + 1, \dots, a_{i+1}$, compute $\iota(a' \rightarrow a)$ according to (12).
- For each $a' \in \{a_i + 2, \dots, a_{i+1}\}$ and for each $a \in \{a_{i+1}, \dots, t\}$, (where $t = M$ for $i = K - 1$ and $t = a_{i+2} - 1$ otherwise) compute $\iota(a' \rightarrow a)$ according to (12).

3) Recursion

- $S_1(a) = \iota(1 \rightarrow a)$ for $a \in \{1, \dots, a_1\}$.
- Store the local decision $h_1(a) = 0$ for $a \in \{1, \dots, a_1\}$.
- For each $1 \leq i \leq K - 1$,
 - Compute

$$S_{i+1}(a_i) = \max_{a'} S_i(a') + \iota(a' \rightarrow a_i),$$

$$h_{i+1}(a_i) = \arg \max_{a'} S_i(a') + \iota(a' \rightarrow a_i),$$

where the maximization is over $a' \in \{a_{i-1} + 1, \dots, a_i - 1\}$.

- For each $a \in \{a_i + 1, \dots, a_{i+1} - 1\}$ compute

$$S_{i+1}(a) = \max_{a'} S_i(a') + \iota(a' \rightarrow a),$$

$$h_{i+1}(a) = \arg \max_{a'} S_i(a') + \iota(a' \rightarrow a),$$

where the maximization is over $a' \in \{a_{i-1} + 1, \dots, a_i\}$.

- Compute

$$S_{i+1}(a_{i+1}) = S_i(a_i) + \iota(a_i \rightarrow a_{i+1}),$$

$$h_{i+1}(a_{i+1}) = a_i.$$

- 4) Find the optimal quantizer by traceback. Let $a_{K+1}^* = M$. For each $i \in \{K, K - 1, \dots, 1\}$,

$$a_i^* = h_{i+1}(a_{i+1}^*).$$

Theorem 2 guarantees finding all the optimal quantizers at each step provided that the algorithm is run with all seeds from the previous step and that a tie-preserving implementation collects all locally optimal decisions and tracebacks.

Note that the dual of this algorithm can be developed for the bottom-up approach, based on the *contraction* procedure. Namely, with the assumption of already knowing the optimal K -level quantizer, all the optimal $(K - 1)$ -level quantizers are found using similar dynamic programming approach.

D. Complexity

The splitting algorithm developed here has complexity $O(M^2)$ in the worst case, and more generally it has complexity $O(\sum_{i=1}^K b_i b_{i+1})$ where $\sum_{i=1}^K b_i = M$. The worst case complexity is in the same order as the best known state of the art algorithm in [8].

E. Example: Additive White Gaussian Noise (AWGN) Channel

We consider a binary-input AWGN channel with equally likely ± 1 inputs and noise variance of $\sigma^2 = 0.5$. We first uniformly quantize the output of the AWGN channel y between -2 and 2 with $M = 1000$ levels. The natural order of the outputs of the resulting DMC satisfies (5). Later we apply the splitting algorithm to find a quantizer with minimum output levels which preserves $\alpha = 0.99$ 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 with those obtained by the algorithm in [4]. We observe that the optimal quantizer with $K = 8$ outputs satisfies the mutual information constraint (Fig. 3).

APPENDIX A

PROOF OF LEMMA 1

Let us denote the new output resulting from merging j and l as y'_{jl} and its conditional posterior probability as v_{jl}

$$v_{jl} = P_{1|y'_{jl}} = \frac{(\pi_j v_j + \pi_l v_l)}{\pi_j + \pi_l} \rightarrow \frac{\pi_j}{\pi_l} = \frac{v_l - v_{jl}}{v_{jl} - v_j} \quad (13)$$

$$\bar{v}_{jl} = P_{2|y'_{jl}} = 1 - v_{jl} \rightarrow \frac{\pi_j}{\pi_l} = \frac{\bar{v}_{jl} - \bar{v}_l}{\bar{v}_j - \bar{v}_{jl}}. \quad (14)$$

Now let us assume that

$$\frac{\pi_j}{\pi_l} = \frac{v_l - v_{jl}}{v_{jl} - v_j} \geq \frac{v_l - v_k}{v_k - v_j}, \quad (15)$$

therefore, $v_{jl} \leq v_k$.

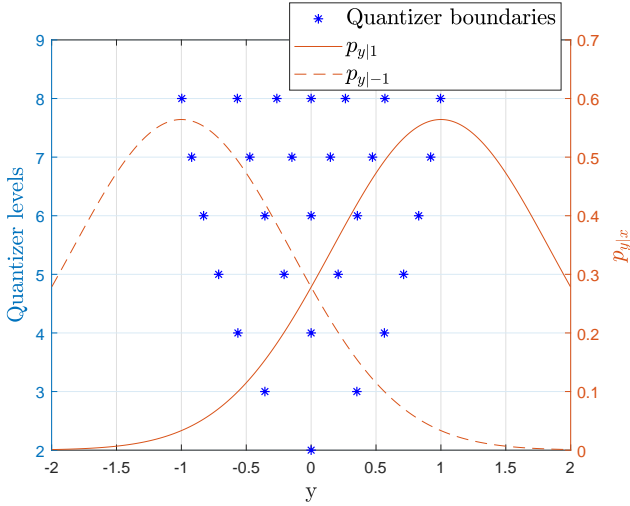


Fig. 2: Optimal quantization of a DMC derived from a finely quantized AWGN channel with $M = 1000$ to $K = 2$ to $K = 8$ levels using the top-down splitting algorithm.

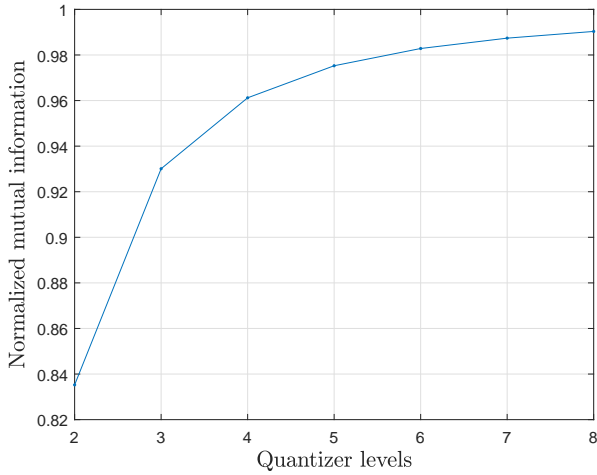


Fig. 3: Normalized mutual information of the Optimal quantizers with $K = 2$ to $K = 8$ levels.

With this assumption, we will show that both terms of the summation in (8) is larger for a (j, l) merge than for a (k, l) merge.

$$\Delta_{\iota_1}(j, l) = \pi_j \Phi(v_j) + \pi_l \Phi(v_l) - (\pi_j + \pi_l) \Phi(v_{jl}) > \Delta_{\iota_1}(k, l), \quad (16)$$

$$\Delta_{\iota_2}(j, l) = \pi_j \Phi(\bar{v}_j) + \pi_l \Phi(\bar{v}_l) - (\pi_j + \pi_l) \Phi(\bar{v}_{jl}) > \Delta_{\iota_2}(k, l). \quad (17)$$

Fig. 4 illustrates (16) where,

$$\delta_1 = \frac{\Delta_{\iota_1}(j, l)}{\pi_j + \pi_l}, \quad \delta_2 = \frac{\Delta_{\iota_1}(k, l)}{\pi_k + \pi_l}. \quad (18)$$

We have the following relations on the triangles in Fig. 4,

$$\frac{\delta_1}{\Delta_1 + \Delta_2} = \frac{v_{jl} - v_j}{v_l - v_j} = \frac{\pi_l}{\pi_j + \pi_l}, \quad (19)$$

$$\frac{\delta_2}{\Delta_2} = \frac{v_{kl} - v_k}{v_l - v_k} = \frac{\pi_l}{\pi_k + \pi_l}, \quad (20)$$

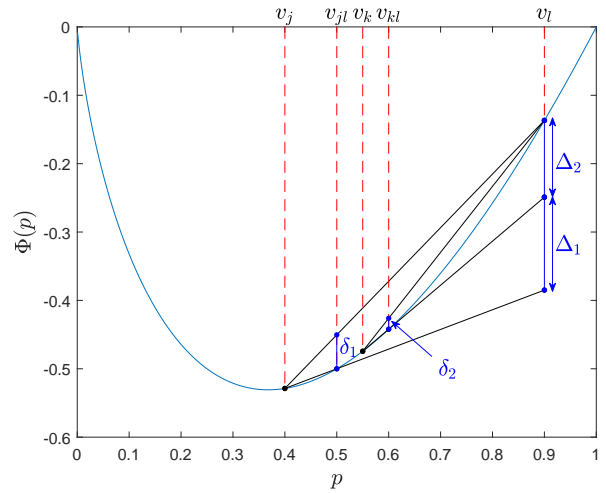


Fig. 4: Illustration of $\Delta_{\iota_1}(j, l)$ and $\Delta_{\iota_1}(k, l)$.

where the second equality comes from (13). Notice that $\Delta_1 > 0$, since $v_{jl} \leq v_k$ and $\Phi(\cdot)$ is a strictly convex function. Using (19) and (20) in (18) we have

$$\Delta_{\iota_1}(j, l) = \pi_l(\Delta_1 + \Delta_2) > \pi_l \Delta_2 = \Delta_{\iota_1}(k, l), \quad (21)$$

which proves (16). We can prove (17) in a similar way since from the assumption in (15) we have $\bar{v}_{jl} \geq \bar{v}_k$.

If we assume other side of inequality from (15), namely

$$\frac{\pi_j}{\pi_l} = \frac{v_l - v_{jl}}{v_{jl} - v_j} \leq \frac{v_l - v_k}{v_k - v_j}, \quad (22)$$

we can similarly prove that $\Delta_{\iota}(j, l) > \Delta_{\iota}(j, k)$. This completes the proof.

REFERENCES

- [1] B. M. Kurkoski, K. Yamaguchi, K. Kobayashi, "Noise thresholds for discrete LDPC decoding mappings", in *Proc. IEEE Global Telecommun. Conf.*, pp. 1–5, Nov./Dec. 2008.
- [2] I. Tal, A. Vardy, "How to construct polar codes", *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6562–6582, 2013.
- [3] M. Dabirnia, A. Martinez, A. Guillén i Fàbregas, "A Mismatched Decoding Perspective of Channel Output Quantization," in *Proc. Inform. Theory Workshop*, Visby, Sweden, Aug. 2019.
- [4] B. M. Kurkoski, H. Yagi, "Quantization of Binary-Input Discrete Memoryless Channels", *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4544–4552, Aug. 2014.
- [5] H. Yagi, B. M. Kurkoski, "Channel Quantizers that Maximize Random Coding Exponents for Binary-Input Memoryless Channels", in *Proc. IEEE Int. Conf. Commun.*, pp. 2256–2260, Jun. 2012.
- [6] D. Burshtein, V. Della Pietra, D. Kanevsky, A. Nádás, "Minimum impurity partitions", *Ann. Statist.*, vol. 20, no. 3, pp. 1637–1646, Sep. 1992.
- [7] N. Slonim and N. Tishby, "Agglomerative Information Bottleneck", in *Proc. of Neural Information Processing Systems (NIPS-99)*, pp. 617–623, 1999.
- [8] K. Iwata and S. Ozawa, "Quantizer design for outputs of binary-input discrete memoryless channels using SMAWK algorithm", in *Proc. IEEE Int. Symp. Information Theory*, Honolulu, HI, 2014, pp. 191–195.
- [9] E. Luber, M. Molinaro, F. M. Pèrreiras, "Binary Partitions with Approximate Minimum Impurity" in *Proc. of Machine Learning Research*, Jul. 2018.
- [10] J. A. Zhang and B. M. Kurkoski, "Low-complexity quantization of discrete memoryless channels," in *Proc. Int. Symp. on Information Theory and Its Applications*, Monterey, CA, 2016, pp. 448–452.