# Thresholding Quantizer Design for Mutual Information Maximization Under Output Constraint

Thuan Nguyen
School of Electrical and
Computer Engineering
Oregon State University
Corvallis, OR, 97331
Email: nguyeth9@oregonstate.edu

Thinh Nguyen
School of Electrical and
Computer Engineering
Oregon State University
Corvallis, 97331
Email: thinhq@eecs.oregonstate.edu

***Abstract*—We consider a channel with discrete input $X$, a continuous noise that corrupts the input $X$ to produce the continuous-valued output $U$. A thresholding quantizer is then used to quantize the continuous-valued output $U$ to the final discrete output $V$. The goal is to jointly design a thresholding quantizer that maximizes the mutual information between input and quantized output $I(X;V)$ while minimizing a pre-specified function of the quantized output $F(p_V)$. A general dynamic programming algorithm is proposed having the time complexity $O(KNM^2)$ where $N$, $M$ and $K$ are the sizes of input $X$, output $U$ and quantized output $V$, respectively. Moreover, we show that if $F(p_V) = \sum_{i=1}^{K} g_i(p_{v_i})$ where $g_i(.)$ is a convex function, $p_{v_i} \in p_V = \{p_{v_1}, \dots, p_{v_K}\}$ is the probability mass function of output $v_i \in V$ and the channel conditional density $p(u|x)$ satisfies the dominated condition (often true in practice), then the existing SMAWK algorithm can be applied to reduce the time complexity of the dynamic programming algorithm from $O(KNM^2)$ to $O(KNM)$. Both theoretical and numerical results are provided to verify our contributions.**

Keyword: channel quantization, mutual information, constraints, threshold, partition, optimization.

## I. Introduction

A communication system can be modeled by an abstract channel with a set of inputs at the transmitter and a set of corresponding outputs at the receiver. Often times the transmitted symbols (inputs) are different from the receiving symbols (outputs), i.e., errors occur due to many factors such as the physics of signal propagation through a medium or thermal noise. Thus, the goal of a communication system is to transmit the information reliably at the fastest rate. The fastest achievable rate with the error approaching zero for a given channel is the channel capacity which is the maximum mutual information between the input and output random variables. In the case of discrete memoryless channels (DMC), a channel matrix is used to specify the property of the transmissions. Furthermore, for a given channel matrix, the mutual information is a concave function of the input probability mass function (pmf), thus there are efficient convex algorithms to find the channel capacity [1], [2], [3]. On the other hand, in many real-world scenarios, the channel matrix is not given. Rather, the channel matrix is a result of the engineering design under many considerations such as complexity of circuit implementations, power consumption, encoding/decoding speeds, and so on. In this case, the entries in the channel matrix are also the variables to be optimized. Consequently, the mutual information is no longer a concave function of the input pmf, but is a possibly non-concave/convex function in both input pmf and the entries of the channel matrix. Thus, the problem becomes more challenging.

A particular class of channel matrix design is the quantizer design. Specifically, many real-world communication scenarios can be modeled as a channel with discrete inputs, additive continuous noise, and the discrete outputs as a result of quantizing the sum of continuous noise and discrete inputs. In such cases, each quantization scheme produces a different channel matrix which ultimately determines the channel capacity. Thus, designing an optimal quantizer is critical. Many quantizers are based on some intuitive objectives such as minimizing the MSE distortion and error rate or maximizing the mutual information (capacity) between the inputs and outputs [4]–[7]. Recently, designing quantizer that maximizes the mutual information [8]–[13] is very important because of their applications in designing Polar code and LDPC code decoders [14], [15].

Our paper is focused on an important class of quantizer called the thresholding quantizer shown in Fig. 1. The transmitted signal $X$ is discrete, the noise is continuous, thus the received signal $U$ is continuous and the output $V$ is discrete due to the quantization of $U$ via a thresholding scheme. The thresholding scheme (Section II) is designed to maximize the mutual information $I(X;V)$ while minimizing a pre-specified function of the quantized output pmf $F(p_V)$. Unlike many existing works, introducing $F(p_V)$ into the problem formulation allows one to shape the quantized output $V$ for different applications. Several application examples using $F(p_V)$ are presented in Sec. II. To that end, this paper makes the following contributions: (1) we propose a general and efficient dynamic programming algorithm having the time complexity $O(KNM^2)$ to determine the optimal thresholding values of the quantizer that maximizes $I(X;V)$ while minimizing $F(p_V)$; (2) we show that if $F(p_V) = \sum_{i=1}^{K} g_i(p_{v_i})$ where $g_i(.)$ is a convex function, $p_{v_i} \in p_V = \{p_{v_1}, \dots, p_{v_K}\}$ is the probability of output $v_i \in V$ and the channel conditional density $p(u|x)$ satisfies the dominated condition (often true

978-1-7281-5207-3/20/$31.00 ©2020 IEEE

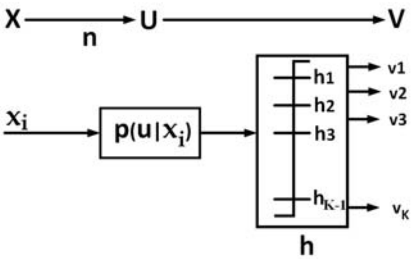

Figure 1: A discrete memoryless channel having $N$ inputs and $K$ quantized outputs using $K-1$ thresholds.

in practice), then the existing SMAWK algorithm [16] can be applied to reduce the time complexity of the dynamic programming algorithm from $O(KNM^2)$ to $O(KNM)$.

## II. Problem Formulation

Fig. 1 illustrates a thresholding quantizer for a discrete memoryless channel. The input set consists of $N$ discrete transmitted symbols $x_i \in \mathbb{R}$, $x_1 < x_2 < \cdots < x_N$ with a given pmf $p_X = \{p_1, p_2, \ldots, p_N\}$. Due to a continuous noise, the received signal $u \in \mathbb{R}$ is modeled via the conditional density $p_{U|X}(u|x_i)$. We note that $p_{U|X}(u|x_i)$ can have different statistics associated with each transmitted signal $x_i$. The quantized outputs $V$ is obtained by quantizing $u$ into $K$ discrete outputs $v_i \in V = \{v_1, \ldots, v_K\}$ using $K-1$ thresholds $h_1 \leq h_2 \leq \cdots \leq h_{K-1}$ with the following mapping:

$$Q(u) = v_i, \text{ if } h_{i-1} \leq u < h_i. \tag{1}$$

Let $p_X$ be the input pmf, $p_V = (p_{v_1}, p_{v_2}, \ldots, p_{v_K})$ be the pmf of the quantized output, $F(p_V)$ be a given function of $p_V$ of the form:

$$F(p_V) = g_1(p_{v_1}) + g_2(p_{v_2}) + \cdots + g_K(p_{v_K}), \tag{2}$$

for some functions $g_i : \mathbb{R} \rightarrow \mathbb{R}$. Since both $I(X;V)$ and $p_V$ depend on the quantizer, i.e., they are functions of the threshold vector $h = (h_1, h_2, \ldots, h_{K-1})$, we are interested in solving the following optimization problem:

$$\max_h \beta I(X;V) - F(p_V), \tag{3}$$

where $\beta$ is pre-specified parameter to control a given trade-off between $I(X;V)$ and $F(p_V)$. Since we assume a fixed input pmf $p_X$, this problem is equivalent to the problem:

$$\min_h [\beta H(X|V) + F(p_V)]. \tag{4}$$

The problem setup above can be used in many scenarios that involve subsequent storage or transmission of $V$ as shown in Fig. 2. We list a few possible candidates below.

**Compression.** Suppose we want to compress data $U$ to $V$ and then transmit/store $V$ as the intermediate representation of $U$ over a low bandwidth channel or in a smaller storage. Since the goal is to maintain the mutual information between $X$ and $V$ as much as possible while reducing the information in $V$ (thus compression), $F(p_V)$ can be used to represent the entropy function

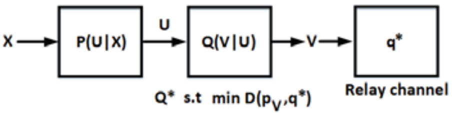

Figure 2: Quantized output $V$ becomes the input of a relay channel having optimal input distribution $q^*$. One wants to find an optimal quantizer which maximizes $I(X;V)$ while minimizing the distance $D(p_V, q^*)$ between $p_V$ and $q^*$.

$$H(p_V) = H(p_{v_1}, p_{v_2}, \ldots, p_{v_K}) = -\sum_{i=1}^{K} p_{v_i} \log(p_{v_i}),$$

which satisfies the form in (2), where $g_i(x) = -x\log(x), \forall i$.

**Power consumption.** Suppose we want to minimize the power to transmit $V$ over the relay channel using PAM while keeping the mutual information between $X$ and $V$ as much as possible. Each value $v_1, v_2, \ldots, v_K$ are transmitted using pulses of different magnitudes associated with the varying power levels $w_1, w_2, \ldots, w_K$. Thus, the average transmit power consumption can be represented as

$$F(p_V) = F(p_{v_1}, p_{v_2}, \ldots, p_{v_K}) = \sum_{i=1}^{K} w_i p_{v_i},$$

which satisfies the form in (2), where $g_i(x) = w_i x$.

**Matching pmf.** Suppose we want to match the output pmf $p_V$ to some given pmf $q^* = (q_1^*, q_2^*, \ldots, q_K^*)$, the input to the relay channel (Fig. 2). The motivation for matching is that $q^*$ was determined a priori to be optimal/good in some context for the relay channel. In this scenario, we want to minimize the difference between $p_V$ and $q^*$. Two popular methods to measure the differences between any two pmfs are the $l_2$ and Kullback Leibler (KL) distances. The KL and $l_2$ distances are defined respectively as:

$$D_{KL}(p_V||q^*) = \sum_{i=1}^{K} p_{v_i} \log \frac{p_{v_i}}{q_i^*},$$

$$D_{l_2}(p_V, q^*) = \sum_{i=1}^{K} (p_{v_i} - q_i^*)^2.$$

Both distances can be easily verified to satisfy the form in (2).

**Deterministic Information Bottleneck (DIB).** It can be seen that the existing DIB method [17] which solves the following problem

$$Q^* = \min_Q [H(p_V) - \beta I(X;V)], \tag{5}$$

is an instance of our problem. We also note that DIB method finds the local solution (5) for a general quantizer. In contrast, our algorithm finds a global solution within the space of all possible thresholding quantizers.

## III. Algorithms

In this section, we first present a dynamic programming algorithm that can determine an optimal solution with the time complexity of $O(KNM^2)$ where $N$ is the size of input alphabet $X$, $K$ is the size of quantized output set $V$, and $M$ is the parameter to control the solution precision. In particular, since the conditional pmf $p_{U|X}(u|x)$ is a continuous function, $u$ is a continuous value, and to perform numerical computations, we need to quantize the range of $U$ into one of the $M$ disjoint bins $(u_i, u_{i+1}]$ of equal width of $\epsilon$, $i = 1, 2, \dots M$. Note that $M >> K$, and a larger $M$ results in a smaller $\epsilon$. The algorithms will find each $h_i$ that is no more than $\epsilon/2$ away from the true $h_i^*$. We then show that under some certain conditions, the dynamic programming algorithm (4) can be augmented to find the optimal solution in linear time complexity $O(KNM)$.

### A. Dynamic Programming Algorithm

The key to the proposed dynamic programming algorithm is the following observation. First, our problem is a 1-dimensional clustering problem where the values of $u$ are clustered into $K$ bins using the thresholding vector $h = (h_1, h_2, \dots, h_{K-1})$ as the boundaries with $h_1 \leq h_2 \leq \cdots \leq h_{K-1}$. Specifically, by definition of a thresholding quantizer, if $h_{l-1} \leq u < h_l$ then $Q(u) = v_l$. Therefore, $h_{l-1}$ is $u_i$ and $h_l$ is $u_j$ for some $i < j$. Thus the clustering problem is to determine the $K-1$ indices of $u$ that form the boundaries of the $K$ clusters.

Now, let us define $D(i, j, k)$ as the minimum (optimal) value of $\beta H(X|V) + F(p_V)$ by clustering $u$ in the range $(u_i, u_j]$ into $k$ clusters where $0 \leq i \leq j \leq M$ and $0 \leq k \leq K$. Each $D(i, j, k)$ is the result of using an optimal quantizer $Q^*(i, j, k)$ which separates points in $(u_i, u_j]$ to $k$ clusters using $k-1$ thresholds. For a given $Q^*(i, j, k)$, define $w(i, j, k)$ and $t(i, j, k)$ as the values of the conditional entropy $\beta H(X|V)$ and cost function $F(p_V)$ associated with the optimal quantizer $Q^*(i, j, k)$, then

$$D(i, j, k) = w(i, j, k) + t(i, j, k). \tag{6}$$

Now, the key of the dynamic programming algorithm is based on the following recursion:

$$D(i, j, k) = \min_{0 \leq q \leq j-1} \{D(i, q, k-1) + D(q+1, j, 1)\}. \tag{7}$$

The recursion can be briefly explained as follows. First, we can show that the value of the $k$ partitions is equal to the sum of values of each of its partitions. This is the property of $H(X|V)$ and $F(p_V)$ as each can be written as functions of sum of their partitions, i.e.,

$$\begin{aligned} \beta H(X|V) + F(p_V) &= \beta[\sum_{i=1}^{K} p_{v_i} H(X|v_i)] + \sum_{i=1}^{K} g_i(p_{v_i}) \\ &= \sum_{i=1}^{K} [\beta p_{v_i} H(X|v_i) + g_i(p_{v_i})]. \end{aligned} \tag{8}$$

In the above recursion, the value of $k$ partitions with a total of $j$ elements can be written as the sum of $k-1$ partitions with $q$ elements and one additional partition with $j-q$ elements. Thus, minimum value can be found by searching for the right index $q$, and the recursion follows. Again, we note that this dynamic programming approach works because the value of the large partition equals the sum of the values of its smaller sub-partitions.

Now, consider initial values $D(i, j, k) = 0$ if $j = 0$ or $k = 0$. From this initial values, using (7), one can compute all of $D(i, j, k)$. The optimal solution is $D(1, M, K)$. After finding the optimal solution, one can use the backtracking method to find all the optimal thresholds. The backtracking step is performed by storing the indices that result in the minimum values. Specifically,

$$H_k(j) = \underset{q}{\operatorname{argmin}} \{D(i, q, k-1) + D(q+1, j, 1)\}. \tag{9}$$

Then, $H_k(j)$ saves the position of $k - 1^{th}$ threshold. Finally, let $h_K^* = M$, for each $i = \{K-1, K-2, \dots, 1\}$, all of other optimal thresholds can be found by backtracking.

$$h_i^* = H_{i+1}(h_{i+1}^*). \tag{10}$$

We note that given $p_X$, $p_{U|X}(u|x)$, and $h$, it is straightforward to compute $\beta H(X|V) + F(p_V)$, and therefore $D(i, j, k)$. However, we omit these derivations due to limited space. Rather, we present the Algorithm 1 that shows the proposed dynamic programming approach.

**Algorithm 1** Dynamic programming for finding $D(1, M, K)$

```
1:  Input: $p_X$, $p_U$, $p_{U|X}$, $N$, $M$, $K$.
2:  Initialization: $D(i, j, k) = 0$ for $\forall$ $j = 0$ or $k = 0$.
3:  Recursion step:
4:      For $k = 1, 2, \dots, K$
5:          For $j = 1, 2, \dots, M$
                $D(i, j, k) = \min_{0 \leq q \leq j-1} \{D(i, q, k-1) + D(q+1, j, 1)\}.$
6:          End For
7:          Store the local decision:
                $H_k(j) = \operatorname{argmin}_q \{D(i, q, k-1) + D(q+1, j, 1)\}.$
8:      End For
9:  Backtracking step: Let $h_K^* = M$, for each $i = \{K-1, K-2, \dots, 1\}$
                $h_i^* = H_{i+1}(h_{i+1}^*).$
10: Output: $D(1, M, K)$, $h^* = \{h_1^*, h_2^*, \dots, h_{K-1}^*\}$.
```

**Complexity.** Algorithm 1 requires $O(KM^2)$ memory space to store $D(i, j, k)$ and $O(KM)$ memory space to store $H_i(j)$, the total auxiliary space is $O(KM^2 + KM)$. We also note that except step 5 in Algorithm 1 takes the time complexity of $O(KNM^2)$, other steps can be done in a linear time. Thus, the total time complexity of Algorithm 1 is $O(KNM^2)$.

### *B. Speedup Dynamic Programming Using SMAWK algorithm*

SMAWK algorithm [16] use the special property of totally monotone matrix (to be defined shortly) to find the maximum (or minimum) value in each row of $M \times M$ matrix in linear time $O(M)$. As will be seen shortly, SMAWK can be used to speed up the proposed dynamic programming algorithm under a certain condition. We first begin with some definitions.

**Definition 1. Dominated conditional distribution channel.** A channel is a dominated conditional distribution channel if all the densities $\phi_i(u) = p_{U|X}(u|x_i)$ satisfies:

$$\frac{\phi_i(u)}{\phi_j(u)} \geq \frac{\phi_i(u')}{\phi_j(u')}, \tag{11}$$

for $\forall\ i \leq j$ and $u \leq u'$.

In practice, the inequality (11) is not too restricted. For example, in typical communication scenarios [7], [18] where the noise is additive, i.e., $u = x_i + n$, the inequality (11) holds for a variety of common noise distributions such as normal distribution, exponential distribution, gamma distribution, uniform distribution, and more generally, all log-concave distributions (Corollary 2 [7]).

**Definition 2. Quadrangle inequality (Monge matrix).** For $i \leq j \leq k \leq l$. The function $w(.)$ satisfies quadrangle inequality if

$$w(i,k) + w(j,l) \leq w(i,l) + w(j,k). \tag{12}$$

For more detailed about quadrangle inequality or Monge matrix, please see [19].

**Definition 3. Totally monotone matrix**. A $2 \times 2$ matrix

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

is monotone if $A_{11} > A_{12}$ implies that $A_{21} > A_{22}$. A matrix $B$ is totally monotone matrix if any $2 \times 2$ submatrix of $B$ is monotone matrix.

Reader can view [16] for the original definition and applications of the totally monotone matrices.

**Theorem 1.** If any $2 \times 2$ submatrix of $B$ satisfies the Quadrangle inequality or Monge matrix, i.e.,

$$B[i_1, j_1] + B[i_2, j_2] \leq B[i_1, j_2] + B[i_2, j_1], \tag{13}$$

for $i_1 \leq i_2$ and $j_1 \leq j_2$, then $B$ is a totally monotone matrix.

*Proof.* The proof can be viewed in [20], Lemma 2.4. □

Now, we reformulate the recursion step in Algorithm 1. For convenience, we define the matrix $D_k$, $k = 1, 2, \dots, K-1$ as follows:

$$D_k[i,j] = \begin{cases} D(0, j-1, k) + D(j, i, 1), & i \geq j, \\ +\infty & i < j. \end{cases}$$

Noting that $D(j, i, 1)$ corresponds to the cost of clustering the interval $(u_j, u_i)$ to $k + 1^{th}$ cluster $(v_{k+1})$. Thus, from (6) and (8)

$$\begin{aligned} D(j,i,1) &= w(j,i,1) + t(j,i,1) \\ &= \beta p_{v_{k+1}} H(X|v_{k+1}) + g_{k+1}(p_{v_{k+1}}). \end{aligned} \tag{14}$$

Thus, solving (7) (step 5, Algorithm 1) is equivalent to find all the minimum in each row of $D_k$. To show that matrix $D_k$ is totally monotone, for $\forall\ i_1 \leq i_2$ and $j_1 \leq j_2$ one has to show that

$$D_k[i_1, j_1] + D_k[i_2, j_2] \leq D_k[i_1, j_2] + D_k[i_2, j_1]. \tag{15}$$

From the definition of $D_k[i,j]$, (15) is equivalent to

$$D(i_1, j_1, 1) + D(i_2, j_2, 1) \leq D(i_1, j_2, 1) + D(i_2, j_1, 1). \tag{16}$$

**Theorem 2.** For a dominated conditional distribution channel and if the output constraint function $g_i(.)$ is convex $\forall\ i$, then:

$$D(r, s, 1) + D(r', s', 1) \leq D(r, s', 1) + D(r', s, 1) \tag{17}$$

for all $1 \leq r \leq r' \leq s \leq s' \leq M$.

*Proof.* Please see our extension version. □

Theorem 2 implies (16). Thus, $D_k$ is a totally monotone matrix for any dominated conditional distribution channel if the output constraints function $g_i(.)$ is convex.

**Corollary 1.** For any dominated conditional distribution channel, if the output constraint function $g_i(.)$ is convex $\forall\ i$, the global solution of problem (4) can be found in $O(KNM)$.

*Proof.* We begin with the recursion step in Algorithm 1. Due to $D_k$ is totally monotone matrix, the SMAWK algorithm [16] can be applied to find the minimum value in each row of $D_k$ for $\forall\ k = 1, 2, \dots, K$ in $O(M)$ time complexity while each comparison is in $N$-dimension space. Therefore, step 5 in Algorithm 1 which is the most time consuming step, can be solved in a linear time complexity $O(KNM)$ which finally reduces the time complexity of Algorithm 1 to $O(KNM)$. □

## IV. Numerical results

To illustrate the performance of the Algorithm 1, we provide the following example. Consider a communication system which transmits input $X = \{x_1 = -1, x_2 = 1\}$ having $p_1 = 0.2$, $p_2 = 0.8$ over an additive noise channel with i.i.d Gaussian noise $N(\mu = 0, \sigma = 1)$. The output signal is $U = X + N$. Due to the additive property, the conditional density of output $u$ given input $x_1$ is $p_{U|X}(u|x_1 = -1) = N(-1, 1)$ while the conditional density of output $u$ given input $x_2$ is $p_{U|X}(u|x_2 = 1) = N(1, 1)$. Note that $u$ is continuous, $u \in U = \mathbb{R}$.

The continuous output $u$ then is quantized to 4 output levels $V = \{v_1, v_2, v_3, v_4\}$ using a quantizer $Q$ having 3 thresholds $Q = \{h_1, h_2, h_3\}$. Quantized output $V$ is transmitted over a relay channel $C$ with the optimal input pmf $q^* = [q_1^*, q_2^*, q_3^*, q_4^*]$. We have to find an optimal quantizer $Q^*$ such that the mutual information $I(X;V)$ is maximized

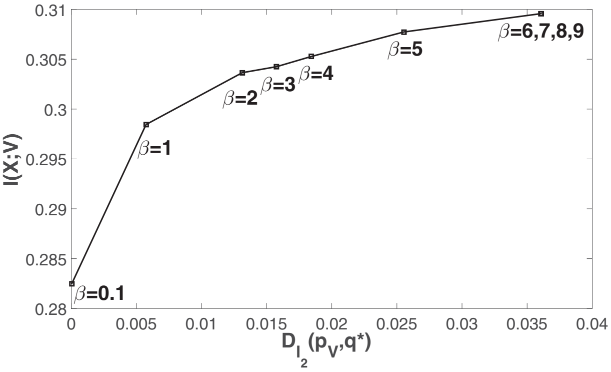

Figure 3: Distance $D_{l_2}(p_V, q^*)$ and mutual information $I(X;V)$ with multiple values of $\beta$.

while the distance $D(p_V, q^*)$ is minimized. In this example, we suppose that $q^*$ is uniform distribution i.e., $q_1^* = q_2^* = q_3^* = q_4^* = 1/4$ and the distance is the $l_2$ distance $D_{l_2}(p_V, q^*)$. Now, we first discrete $U$ to $M = 200$ bins from $[-10, 10]$ with the same width $\epsilon = 0.1$. Thus, $U = \{u_1, u_2, \ldots, u_{200}\}$ with the conditional density $p_{U|X}(x_i|u_j)$ and $p_{u_j}$, $i = 1, 2$, $j = 1, 2, \ldots, 200$ can be determined by using two given conditional densities $p_{U|X}(u|x_1 = -1) = N(-1, 1)$ and $p_{U|X}(u|x_2 = 1) = N(1, 1)$.

Next, to find the optimal quantizer $Q^*$, we scan all the possible value of $\beta \geq 0$. For each value of $\beta$, we run Algorithm 1 to find the optimal quantizer. The simulation results are provided in Fig. 3. As seen, a larger value of $\beta$ results in a larger mutual information $I(X;V)$ at the expense of increasing the distance $D_{l_2}(p_V, q^*)$. On the other hand, a smaller value of $\beta$ produces the opposite effect. We also note that for a relatively large $\beta$ i.e., $\beta \geq 6$, the optimal quantizer $Q^*$ yields the same value of $I(X;V)$ and $D_{l_2}(p_V, q^*)$. That is because with a large enough $\beta$, the optimal quantizer actually only finds the optimal of $I(X;V)$ without paying attention to the constraint on $D_{l_2}(p_V, q^*)$.

To compare the actual running times of Algorithm 1 (with and without SMAWK algorithm) and the exhaustive search, we also run exhaustive search algorithm for all possible thresholds triplets $\{h_1, h_2, h_3\} \in [-10; 10]$. We note that the time complexity of an exhaustive search algorithm is $O(M^{K-1})$. The average running time of exhaustive search algorithm is $t_e = 611.89337$ seconds while the average running times of Algorithm 1 with and without SMAWK algorithm are $t_w = 29.37562$ and $t_{wo} = 93.25663$ seconds, respectively. All algorithms produce the same optimal values.

## V. Conclusion

In this paper, we consider a problem of jointly designing a thresholding quantizer that maximizes the mutual information between input and quantized output $I(X;V)$ while minimizing a pre-specified function of the quantized output $F(p_V)$. This problem has a number of interesting applications. A general dynamic programming algorithm is proposed to significantly reduce the time complexity over the naive exhaustive search. Moreover, we show that if $F(p_V) = \sum_{i=1}^{K} g_i(p_{v_i})$ where $g_i(.)$ is a convex function, $p_{v_i} \in p_V = \{p_{v_1}, \ldots, p_{v_K}\}$ is the probability mass function of output $v_i \in V$ and the channel conditional density $p(u|x)$ satisfies the dominated condition (often true in practice), then the existing SMAWK algorithm can be applied to reduce the time complexity of the dynamic programming algorithm from $O(KNM^2)$ to $O(KNM)$. Both theoretical and numerical results are provided to verify our contributions.

## References

[1] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[2] Thuan Nguyen and Thinh Nguyen. On closed form capacities of discrete memoryless channels. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2018.

[3] Thuan Nguyen and Thinh Nguyen. Single-bit quantization capacity of binary-input continuous-output channels. *arXiv preprint arXiv:2001.01842*, 2020.

[4] Jiuyang Alan Zhang and Brian M Kurkoski. Low-complexity quantization of discrete memoryless channels. In *2016 International Symposium on Information Theory and Its Applications (ISITA)*, pages 448–452. IEEE, 2016.

[5] Thuan Nguyen, Yu-Jung Chu, and Thinh Nguyen. On the capacities of discrete memoryless thresholding channels. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2018.

[6] Brian M Kurkoski and Hideki Yagi. Quantization of binary-input discrete memoryless channels. *IEEE Transactions on Information Theory*, 60(8):4544–4552, 2014.

[7] Thuan Nguyen and Thinh Nguyen. On the uniqueness of binary quantizers for maximizing mutual information. *arXiv preprint arXiv:2001.01836*, 2020.

[8] Yuta Sakai and Ken-ichi Iwata. Suboptimal quantizer design for outputs of discrete memoryless channels with a finite-input alphabet. In *Information Theory and its Applications (ISITA), 2014 International Symposium on*, pages 120–124. IEEE, 2014.

[9] Thuan Nguyen and Thinh Nguyen. Communication-channel optimized partition. *arXiv preprint arXiv:2001.01708*, 2020.

[10] Xuan He, Kui Cai, Wentu Song, and Zhen Mei. Dynamic programming for discrete memoryless channel quantization. *arXiv preprint arXiv:1901.01659*, 2019.

[11] Thuan Nguyen and Thinh Nguyen. Minimizing impurity partition under constraints. *arXiv preprint arXiv:1912.13141*, 2019.

[12] Ken-ichi Iwata and Shin-ya Ozawa. Quantizer design for outputs of binary-input discrete memoryless channels using smawk algorithm. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 191–195. IEEE, 2014.

[13] Thuan Nguyen and Thinh Nguyen. Optimal quantizer structure for binary discrete input continuous output channels under an arbitrary quantized-output constraint. *arXiv preprint arXiv:2001.02999*, 2020.

[14] Francisco Javier Cuadros Romero and Brian M Kurkoski. Decoding ldpc codes with mutual information-maximizing lookup tables. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 426–430. IEEE, 2015.

[15] Ido Tal and Alexander Vardy. How to construct polar codes. *arXiv preprint arXiv:1105.6164*, 2011.

[16] Alok Aggarwal, Maria M Klawe, Shlomo Moran, Peter Shor, and Robert Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(1-4):195–208, 1987.

[17] DJ Strouse and David J Schwab. The deterministic information bottleneck. *Neural computation*, 29(6):1611–1630, 2017.

[18] Brian M Kurkoski and Hideki Yagi. Single-bit quantization of binary-input, continuous-output channels. In *Information Theory (ISIT), 2017 IEEE International Symposium on*, pages 2088–2092. IEEE, 2017.

[19] F Frances Yao. Efficient dynamic programming using quadrangle inequalities. In *Proceedings of the twelfth annual ACM symposium on Theory of computing*, pages 429–435. ACM, 1980.

[20] Rainer E Burkard, Bettina Klinz, and Rüdiger Rudolf. Perspectives of monge properties in optimization. *Discrete Applied Mathematics*, 70(2):95–161, 1996.