A LINEAR TIME PARTITIONING ALGORITHM FOR FREQUENCY WEIGHTED IMPURITY FUNCTIONS

Thuan Nguyen and Thinh Nguyen

School of EECS, Oregon State University, Corvallis, OR 97331-5501, USA
nguyeth9@oregonstate.edu, thinhq@eecs.oregonstate.edu

ABSTRACT

Partitioning algorithms play a key role in machine learning, signal processing, and communications. They are used in many well-known NP-hard problems such as k-means clustering and vector quantization. The goodness of a partition scheme is measured by a given impurity function over the resulted partitions. The optimal partition is one(s) with the minimum impurity. Practical algorithms for finding an optimal partitioning are approximate, heuristic, and often assume certain properties of the given impurity function such as concavity/convexity. In this paper, we propose a heuristic, efficient (linear time) algorithm for finding the minimum impurity for a broader class of impurity functions which includes popular impurities such as Gini index and entropy. We also make a connection to a well-known result which states that the optimal partitions correspond to the regions separated by hyperplane cuts in the probability space of the posterior distribution.

Index Terms— Partition, quantization, clustering, optimization, minimum impurity.

1. INTRODUCTION

How to partition of a set of M points into K clusters to maximize/minimize a given objective is an important problem. It is key to many algorithms in computer science, signal processing, and communications. In machine learning, many classifiers such as decision tree and k-means clustering employ partitioning as their key components. In signal processing and communication, partitioning is used in vector quantization algorithms for a variety of applications such as compression and error correcting codes. A popular criteria to evaluate the goodness of the partition scheme is the *purity* of data in each partition. Maximizing the *purity* of the partitions is equivalent to minimizing its *impurity*, which is measured by an impurity function over the partitions such as the entropy function and the Gini index [1]. In general, the partition problem is NP-hard. An exhaustive search has the complexity of $O(M^K)$, and it is infeasible for large K and M. Consequently, practical algorithms for finding an optimal partitioning are approximate, heuristic, and often assume certain properties of the given impurity function such as concavity/convexity [2], [3], [4], [5], [6], [7], [8]. In some special cases where data is low dimensional, $K = 2$, and the impurity function is a concave frequency-weighted impurity function, then the optimal partition can be found in $O(M \log M)$ where M is the number of data points [9]. Recently, [10], [11], [12] proposed an exact algorithm that guarantees the performance for binary partition. However, this method is limited to only a binary partition ($K = 2$). For broader cases where $K > 2$ and the impurity function is frequency weighted concave function, there exist theoretical analyses, especially the optimality condition for the partition scheme. Specifically, both [5] and [6] showed that the optimal partitions can be separated by a hyperplane in the probability space of the posterior distribution.

The partitioning problem is also important in recent work on error correcting codes, specifically polar codes [13] and LDPC codes [14]. Many optimal codes depend on optimal quantizers that maximize the mutual information between input and quantized output of a discrete memoryless channel (DMC) [15], [16], [17], [18]. Finding optimal quantizers can be viewed as a partitioning problem where maximizing the mutual information is equivalent to minimizing the conditional entropy [19], [20], [21] which can be shown to be a frequency-weighted impurity function.

In this paper, we propose an efficient (linear time) algorithm that works well for finding the optimal partition for a broader class of frequency-weighted impurity functions. Furthermore, the algorithm can be sped up under the assumption of convexity. Moreover, our analysis showed that the optimal partitions correspond to the regions separated by hyperplane cuts in the probability space of the posterior distribution. This finding is similar to the classic result in [5].

2. PROBLEM FORMULATION

We consider a general setting shown in Fig. 1 [22]. X is a discrete random source consisting of N symbols $x_1, x_2, \dots, x_N$ with a given pmf $p(x_i)$. A discrete random source Y consists of M symbols $y_1, y_2, \dots, y_M$, is generated using a given joint pmf $p(x_i, y_j)$. Due to $p(x_i)$ and $p(x_i, y_j)$ are given, one can easily determine $p(y_j|x_i)$, $p(x_i|y_j)$ and $p(y_j)$. Using a certain

978-1-5090-6631-5/20/$31.00 ©2020 IEEE

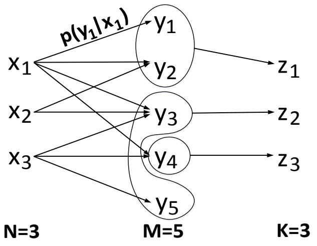

Fig. 1: Quantizer $Q : Y \to Z$.

clustering/partitioning/quantizing operation Q:

$$Q : Y \to Z, \tag{1}$$

Y is mapped into a discrete random source Z consisting of K symbols, $z_1, z_2, \dots, z_K$ with $K < M$. Specifically, each z_i represents a cluster of one or more elements of Y as shown in Fig. 1. An impurity function $F(.)$ is used to measure the goodness of the partitions (clusters) resulted from using Q. The optimal Q^* is one(s) with the lowest value of $F(.)$.

We note that this general setting models many applications in signal processing, communication, and machine learning. For example, in a communication scenario, x_i can be thought as a transmitted symbol, y_j a received symbol, $p(y_j|x_i)$ models the communication channel that introduces noise/distortion, and z_k as a quantized version of y_j. We want z_k to carry most information about x_i in order to decode x_i accurately. In this case, for given $p(x)$, from an information theoretic viewpoint, the conditional entropy function $H(X|Z)$ is the best impurity function. From the machine learning perspective, Gini index is often used in the decision tree as an impurity function. Importantly, we note that a quantizer/mapping Q determines the resulted joint distribution of $p(x_i, z_k)$, and the impurity functions often can be written as a function of $p(z_k)$ and $p(x_i|z_k)$. In particular, both the entropy function and Gini index can be written as a frequency-weighted impurity function which takes the following form:

$$F(X, Z) = \sum_{j=1}^{K} p(z_j) f[p(x_1|z_j), p(x_2|z_j), \dots, p(x_N|z_j)].$$

The factor $p(z_j)$ denotes the weight of set z_j and $f[.]$ measures the impurity in each subset z_j. To find the optimal quantizer Q^*, we want to minimize the total weighted impurity $F(X, Z)$. In this paper, we consider the impurity function $f[.]$ with the following properties:

- $f[.]$ can be written in the following form:

$$f[p(x_1|z_j), \dots, p(x_N|z_j)] = \sum_{i=1}^{N} p(x_i|z_j) g[p(x_i|z_j)] + C.$$

- $g[p(x_i|z_j)]$ is a convex function over $p(x_i|z_j)$.

Popular impurity functions such as entropy [9] and Gini index [1] satisfy the above properties, and thus they are instances of our problem. Also, both entropy and Gini index belong to a class of frequency-weighted impurity concave functions [5], [6] since $f[.]$ is a concave function. However, in our setting, $f[.] = \sum xg[x]$ does not need to be concave, for example, $f[x] = \sum xg[x]$ is convex when $g[x] = x$ is convex. Thus, our impurity function $f[.]$ is more general than the previous impurity functions described in [5], [6], [9].

We also note that our proposed linear time algorithm to find the optimal Q^* does not need to use the convexity of $g[x]$. However, when $g[x]$ is convex, we can speed up the algorithm significantly, depending on the scenario.

3. SOLUTION APPROACH

In this section, we propose an algorithm that finds the local optimal quantizer Q^* which assigns each value of $y_k \in Y$ to only one subset $z_j \in Z$ (hard clustering). We begin by rewriting $F(X, Z)$ as follows:

$$\begin{aligned}
F(X,Z) &= \sum_{j=1}^{K} p(z_j) f[p(x_1|z_j), p(x_2|z_j), \dots, p(x_N|z_j)] && (2)\\
&= \sum_{j=1}^{K} p(z_j)(\sum_{i=1}^{N} p(x_i|z_j) g[p(x_i|z_j)] + C) && (3)\\
&= \sum_{j=1}^{K} p(z_j)(\sum_{i=1}^{N} p(x_i|z_j) g[p(x_i|z_j)]) + C && (4)\\
&= \sum_{j=1}^{K} p(z_j) \sum_{i=1}^{N} \sum_{k=1}^{M} p(x_i|y_k) p(y_k|z_j) g[p(x_i|z_j)] + C && (5)\\
&= \sum_{j=1}^{K} \sum_{k=1}^{M} p(z_j, y_k) \sum_{i=1}^{N} p(x_i|y_k) g[p(x_i|z_j)] + C && (6)\\
&= \sum_{j=1}^{K} \sum_{k=1}^{M} p(z_j|y_k) p(y_k) \sum_{i=1}^{N} p(x_i|y_k) g[p(x_i|z_j)] + C, && (7)
\end{aligned}$$

with (4) due to $[\sum_{j=1}^{K} p(z_j)]C = C$, (5) due to $p(x_i|z_j) = \sum_{k=1}^{M} p(x_i|y_k) p(y_k|z_j)$, (6) due to $p(z_j)p(y_k|z_j) = p(z_j, y_k)$ and (7) due to $p(z_j, y_k) = p(z_j|y_k)p(y_k)$.

Since $p(x_i|y_k)$ and $p(y_k)$ are given $\forall\, i, k$. Thus, $F(X, Z)$ has only two variables: $p(z_j|y_k)$ and $p(x_i|z_j)$. $p(z_j|y_k)$ denotes the assignment of y_k to a subset z_j. For a hard clustering, if $p(z_j|y_k) = 1$, then $y_k \in z_j$, otherwise $y_k \notin z_j$. $p(x_i|z_j)$ denotes the conditional probability of x_i in cluster z_j.

We propose an algorithm similar to the k-means algorithm. It has two alternating steps: **(1) Computing centroids:** fixed $p(z_j|y_k)$, find the optimal $p(x_i|z_j)$ that minimizes $p(y_k) \sum_{i=1}^{N} p(x_i|y_k) g[p(x_i|z_j)]$ and **(2) Clustering assignment:** fixed $p(x_i|z_j)$, find the optimal assignment

$p(z_j|y_k)$ which assigns y_k to cluster z_j. Algorithm 1 shows the pseudo code of the proposed algorithm.

Algorithm 1 Minimize the weighted impurity partitions

```
Input: $p(x), p(x,y), N, K, M$.
Ouput: Optimal partition $Z = \{z_1, \ldots, z_K\}$.
Initialization step: Randomly pick-up $K$ probability vectors $p(x_i|z_j)$ for $i = 1, \ldots, N$ and $j = 1, \ldots, K$ as the initial centroids.
Step 1 (Cluster assignment): Cluster $y_k$ into one of the cluster $z_j$ for a given of $p(x_i|z_j)$ $\forall\ i, j$.
    $z_j = \{y_k | d(y_k, z_j) \leq d(y_k, z_l), \forall l \neq j\}, \forall j,$   (8)
    where
    $d(y_k, z_j) = p(y_k) \sum_{i=1}^{N} p(x_i|y_k) g[p(x_i|z_j)].$   (9)
Step 2 (Computing centroids): Compute centroids $p^*(x_i|z_j)$ for each cluster $z_j$ which is the solution of the following optimal problem.
    $p^*(x_i|z_j) = \min_{p(x_i|z_j)} \sum_{y_k \in z_j} p(y_k) \sum_{i=1}^{N} p(x_i|y_k) g[p(x_i|z_j)], \forall i.$   (10)
Step 3: Go to Step 1 until all clusters stop changing or the maximum number of iterations is reached.
```

Proof of the convergence (outline): Due to the limited space, we describe the outline of the proof. We can show that step 1 of the algorithm always decreases the current value of $F(X, Z)$. Similarly, step 2 will always decrease the current value of $F(X, Z)$. Therefore, by running steps 1 and 2 repeatedly, the algorithm produces a decreasing sequences bounded above by non-zero value, and thus must converge. However, we note that the algorithm might converge to a locally optimal solution.

Speeding up the algorithm using convexity of $g(x)$.

The speed up of the algorithm comes from step 2 of computing the centroids. Since $p(y_k)$, $p(x_i|y_k)$ are given and $g[.]$ is a convex function, the function in Eq. (10) is a linear combination of convex functions, and thus must be convex in $p(x_i|z_j)$. Based on this, we can find the optimal $p^*(x_i|z_j)$ using convex optimization algorithms [23], [24] efficiently. That said, we can propose a faster method for computing the centroids if we can obtain $g'^{-1}[.]$, the inverse function of $g'[.]$, the derivative of $g[.]$. Note that if $g[.]$ is a strictly convex function then $g'[.]$ is a monotonically increasing function and hence $g'^{-1}[.]$ exists. We will show an example to illustrate this point shortly. Furthermore, assuming that $p(x_i, y_k) > 0, \forall i, k$, we can find the global optimal $p^*(x_i|z_j)$ of Eq. (10) in closed-form expression using KKT conditions. We also note that if $g[.]$ is not convex, KKT conditions still can help to find the local optimal of Eq. (10). Thus, the algorithm may converge a bit slower if $g[.]$ is not convex. Now, the optimal $p^*(x_i|z_j)$ can be found using the following lemma:

Lemma 1. Assuming that $g[.]$ is strictly convex and $p(x_i, y_k) > 0, \forall i, k$, then the optimal centroids in Step 2 of Algorithm 1 can be computed by:

$$p^*(x_i|z_j) = w\left[\frac{-\nu^*}{\sum_{y_k \in z_j} p(y_k) p(x_i|y_k)}\right], \tag{11}$$

where ν^* is the root of

$$\sum_{i=1}^{N} w\left[\frac{-\nu^*}{\sum_{y_k \in z_j} p(y_k) p(x_i|y_k)}\right] = 1, \tag{12}$$

where $w^{-1}[.] = g'[.]$.

Proof. Since $p(x_i|z_j)$ is a valid pmf, the optimal $p^*(x_i|z_j)$ has to satisfy the following constraints:

$$\begin{cases} p(x_i|z_j) \succeq 0, \forall i, j \\ \sum_{i=1}^{N} p(x_i|z_j) = 1. \end{cases}$$

Since $p(x_i, y_k) > 0\ \forall\ i, k$, $p(x_i|y_k) = p(x_i, y_k)/p(y_k) > 0$. However, $p(y_k|z_j) > 0$ for at least one y_k since z_j cannot be an empty set. Therefore, $p(x_i|z_j) = \sum_{k=1}^{M} p(x_i|y_k) p(y_k|z_j) > 0$. That said, the above constraints can be reduced to only $\sum_{i=1}^{N} p(x_i|z_j) = 1$.

Consider the Lagrangian function $L[p(x_1|z_j), \ldots, p(x_N|z_j)]$ [24] with the constraint above:

$$\begin{aligned} L[p(x_1|z_j), \ldots, p(x_N|z_j), \nu] &= \sum_{y_k \in z_j} p(y_k) \sum_{i=1}^{N} p(x_i|y_k) g[p(x_i|z_j)] \\ &+ \nu\left(\sum_{i=1}^{N} p(x_i|z_j) - 1\right), \end{aligned} \tag{13}$$

where ν is a dual variable. Using the KKT conditions, the optimal $p^*(x_i|z_j)$ and ν^* must satisfy:

$$\begin{cases} \sum_{i=i}^{N} p^*(x_i|z_j) = 1, \\ \dfrac{\partial L[p(x_1|z_j), \ldots, p(x_N|z_j), \nu]}{\partial p^*(x_i|z_j)} = 0, \forall i. \end{cases} \tag{14}$$

Now, from the second equation of (14),

$$\nu^* = -\sum_{y_k \in z_j} p(y_k) p(x_i|y_k) g'[p^*(x_i|z_j)]]. \tag{15}$$

Let $w[.]$ be the inverse function of $g'[.]$, then

$$p^*(x_i|z_j) = w\left[\frac{-\nu^*}{\sum_{y_k \in z_j} p(y_k) p(x_i|y_k)}\right] \tag{16}$$

Now, ν^* can be found from the first equation of (14),

$$\sum_{i=1}^{N} w\left[\frac{-\nu^*}{\sum_{y_k \in z_j} p(y_k) p(x_i|y_k)}\right] = 1. \tag{17}$$

□

Example 3.1. *Let the impurity function be the conditional entropy function $H(X|Z)$, then the total weighted-frequency impurity $F(X,Z) = \sum_{j=1}^{K} p(z_j)H(X|Z = z_j)$, where $f[x] = -\sum_x x\log x$ and $g[x] = -\log x$ is convex. Thus, $g'[x] = -\frac{1}{x}$ and $w[x] = -\frac{1}{x}$. From Eq. (17),*

$$\sum_{i=1}^{N} \frac{\sum_{y_k \in z_j} p(y_k)p(x_i|y_k)}{\nu^*} = 1. \quad (18)$$

Thus, $\nu^* = \sum_{i=1}^{N}\sum_{y_k\in z_j} p(y_k)p(x_i|y_k) = p(z_j)$ and $p^*(x_i|z_j) = \sum_{y_k\in z_j} p(y_k)p(x_i|y_k)$. We note that this optimal centroids is the same to the centroids of clustering using Bregman divergence (Proposition 1, [25]) due to minimizing conditional entropy is equivalent to minimizing the KL divergence which is a special case of Bregman divergence [25].

Algorithmic complexity: Step 2 of the Algorithm 1 can be obtained in either closed-form expression for convex optimization which is very efficient. The computational bottleneck is step 2 for which, we have to compute the distances from all points in set Y to the K centroids of Z. All computations are performed in N-dimensional space, thus the computational complexity of Algorithm 1 is $O(TKNM)$ where T denotes the number of iterations and K, N and M are the sizes of partition set Z, input source X and the data set Y, respectively.

Hyperplane separation: Algorithm 1 produces a necessary optimal condition that is similar to the previous results [5] and [6] which can be brief stated in the following lemma:

Lemma 2. The optimal partitions that minimize the weighted frequency impurity using Algorithm 1 are separated by a hyperplane in $N-1$ dimensional probability space of the posterior probability $p(x_i|y_k)$.

Proof. (outline) From Algorithm 1, if y_k belongs to an optimal partition z_j, then y_k satisfies the Step 1 of Algorithm 1, or:

$$\sum_{i=1}^{N} p(x_i|y_k)[g[p(x_i|z_j)] - g[p(x_i|z_l)]] \leq 0, \forall l \neq j. \quad (19)$$

Let $a_i = g[p(x_i|z_j)] - g[p(x_i|z_l)], \forall\, i = 1, \ldots, N$. Plug in $p(x_N|y_k) = 1 - \sum_{i=1}^{N-1} p(x_i|y_k)$ into (19) we obtain

$$\sum_{i=1}^{N-1} p(x_i|y_k)[a_i - a_N] + a_N \leq 0. \quad (20)$$

Now at the optimal partitions, $p(x_i|z_j)$ and $p(x_i|z_l)$ are fixed and $[a_i - a_N]$ are constants $\forall\, i$. Therefore, (20) indicates that $p(x_i|y_k)$ is separated by hyperplanes in $N-1$ dimensional space with the parameters $[a_i - a_N], \forall\, 1 \leq i < N$. □

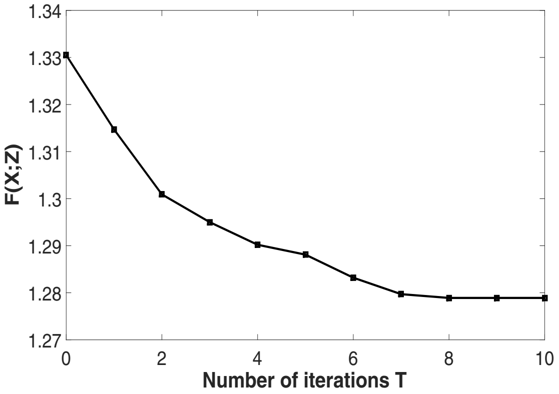

Fig. 2: $F(X,Z)$ as a function of iterations T.

Since the number of hyperplanes in $N-1$ dimensional space is $2\sum_{i=0}^{N}\binom{M-1}{K}$ [5], Lemma 2 reduced the complexity of finding the global solution using an exhaustive searching from K^M to a polynomial time complexity $2\sum_{i=0}^{N}\binom{M-1}{K}$.

4. NUMERICAL EVALUATION

We now provide an example of finding optimal partitions for a Gaussian mixture model consisting of $N = 3$ Gaussian distributions with different means and variances. The conditional entropy is used as the impurity function. Our goal is to classify these points back into $K = 3$ clusters. Specifically,

$$p(y|x_i) = N(y|\mu_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(y-\mu_i)^2}{2\sigma_i^2}}, i = 1, 2, 3,$$

where $\sigma_i = 1\ \forall\ i$ and $\mu_1 = -1, \mu_2 = 0, \mu_3 = 1$.

Since Algorithm 1 is used for discrete dataset, we first uniformly quantize the continuous Gaussian data points to $M = 20$ levels. Next, Algorithm 1 is used to find the optimal partitions. Fig. 2 shows the quick convergence of Algorithm 1 to the exact optimal solution (1.2789) which was computing independently using exhaustive search. The running time of exhaustive search and our Algorithm 1 are 37428.85 and 11.69 seconds, respectively. Note that we can perform an exhaustive searching only with a small value of K, N, and M.

5. CONCLUSION

We propose a heuristic, efficient (linear time) algorithm for finding the minimum impurity for a broad class of impurity functions which includes popular impurities such as Gini index and entropy. We also made a connection to the well-known result which states that the optimal partitions correspond to the regions separated by hyperplane cuts in the probability space of the posterior distribution. Numerical examples are provided to illustrate the proposed algorithm.

6. REFERENCES

[1] J Ross Quinlan, *C4. 5: programs for machine learning*, Elsevier, 2014.

[2] Arthur Nádas, David Nahamoo, Michael A Picheny, and Jeffrey Powell, "An iterative'flip-flop'approximation of the most informative split in the construction of decision trees," in *[Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1991, pp. 565–568.

[3] Philip A. Chou, "Optimal partitioning for classification and regression trees," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , no. 4, pp. 340–354, 1991.

[4] D Burshtein, V Della Pietra, D Kamevsky, and A Nedas, "A splitting theorem for tree construction," in *Proceedings. 1991 IEEE International Symposium on Information Theory*. IEEE, 1991, pp. 284–284.

[5] David Burshtein, Vincent Della Pietra, Dimitri Kanevsky, Arthur Nadas, et al., "Minimum impurity partitions," *The Annals of Statistics*, vol. 20, no. 3, pp. 1637–1646, 1992.

[6] Don Coppersmith, Se June Hong, and Jonathan RM Hosking, "Partitioning nominal attributes in decision trees," *Data Mining and Knowledge Discovery*, vol. 3, no. 2, pp. 197–217, 1999.

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

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

[9] Leo Breiman, *Classification and regression trees*, Routledge, 2017.

[10] Eduardo S Laber, Marco Molinaro, and Felipe A Mello Pereira, "Binary partitions with approximate minimum impurity," in *International Conference on Machine Learning*, 2018, pp. 2860–2868.

[11] Eduardo Laber and Lucas Murtinho, "Minimization of gini impurity: Np-completeness and approximation algorithm via connections with the k-means problem," *Electronic Notes in Theoretical Computer Science*, vol. 346, pp. 567–576, 2019.

[12] Ferdinando Cicalese, Eduardo Laber, and Lucas Murtinho, "New results on information theoretic clustering," in *International Conference on Machine Learning*, 2019, pp. 1242–1251.

[13] Ido Tal and Alexander Vardy, "How to construct polar codes," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582, 2013.

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

[15] Brian M Kurkoski and Hideki Yagi, "Quantization of binary-input discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4544–4552, 2014.

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

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

[18] 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)*. IEEE, 2018, pp. 1–5.

[19] 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)*. IEEE, 2016, pp. 448–452.

[20] 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.

[21] Thuan Nguyen and Thinh Nguyen, "Entropy-constrained maximizing mutual information quantization," *arXiv preprint arXiv:2001.01830*, 2020.

[22] Naftali Tishby, Fernando C Pereira, and William Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[23] Michael Grant and Stephen Boyd, "Cvx: Matlab software for disciplined convex programming, version 2.1," 2014.

[24] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[25] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh, "Clustering with bregman divergences," *Journal of machine learning research*, vol. 6, no. Oct, pp. 1705–1749, 2005.