An Algorithm for Quantization of Discrete Probability Distributions

Yuriy A. Reznik
Qualcomm Inc., San Diego, CA
Email: yreznik@ieee.org

Abstract

We study the problem of quantization of discrete probability distributions, arising in universal coding, as well as other applications. We show, that in many situations this problem can be reduced to the covering problem for the unit simplex, yielding precise characterization in the high-rate regime. We present simple and asymptotically optimal algorithm for solving this problem. Performance of this algorithm is studied and compared with several known solutions.

I. Introduction

The problem of coding of probability distributions surfaces many times in the history of source coding. First universal codes, developed in late 1960s, such as *Lynch-Davisson* [9], [21], *combinatorial* [28], and *enumerative* codes [7] used lossless encoding of frequencies of symbols in an input sequence. The *Rice machine* [24], developed in early 1970's, transmitted quantized estimate of variance of source's distribution. *Two-step universal codes*, developed by J. Rissanen in 1980s, explicitly estimate, quantize, and transmit parameters of distribution as a first step of the encoding process [25], [26]. Vector quantization techniques for two-step universal coding were proposed in [4], [30]. In practice, two-step coding was often implemented by constructing a Huffman tree, then encoding and transmitting this code tree, and then encoding and transmitting the data. Such algorithms become very popular in 1980s and 1990s, and were used, for example, in ZIP archiver [17], and JPEG image compression standard [16].

In recent years, the problem of coding of distributions has attracted a new wave of interest coming from other fields. For example, in computer vision, it is now customary to use histogram-derived descriptions of image features. Examples of such descriptors include SIFT [20], SURF [1], and CHoG [2], differentiating mainly in a way they quantize histograms. Several other uses of coding of distributions are described in [11].

To the best of our knowledge, most related prior studies were motivated by optimal design of two-step universal codes [25], [26], [30], [4], [15]. In such context, quantization of distributions becomes a small sub-problem in a complex rate optimization process, and final solutions yield few insights about it.

In this paper, we treat quantization of distributions as a stand-alone problem. In Section 2, we explain setting of the problem and survey relevant analytic results. In Section 3, we propose and study an algorithm for solving this problem. In Section 4, we provide comparisons with other known techniques. Conclusions are drawn in Section 5.

II. Description of the Problem

Let $A = \{r_1, \ldots, r_m\}$, $m < \infty$, denote a discrete set of events, and let Ω_m denote the set of probability distributions over A:

$$\Omega_m = \left\{ [\omega_1, \ldots, \omega_m] \in \mathbb{R}^m \middle| \, \omega_i \geqslant 0 \, , \textstyle\sum_i \omega_i = 1 \right\} . \tag{1}$$

Let $p \in \Omega_m$ be an input distribution that we need to encode, and let $Q \subset \Omega_m$ be a set of distributions that we will be able to reproduce. We will call elements of Q *reconstruction points* or *centers* in Ω_m.

We will further assume that Q is finite $|Q| < \infty$, and that its elements are enumerated and encoded by using fixed-rate code. The rate of such code is $R(Q) = \log_2 |Q|$ bits. By $d\,(p,q)$ we will denote a *distance measure* between distributions $p, q \in \Omega_m$.

In order to complete traditional setting of the quantization problem, it remains to assume that input quantity $p \in \Omega_m$ is produced by some random process, e.g. a memoryless process with density θ over Ω_m. Then the problem of quantization can be formulated as minimization of the *average distance* to the nearest reconstruction point (cf. [14, Lemma 3.1])

$$\bar{d}(\Omega_m, \theta, R) = \inf_{\substack{Q \subset \Omega_m \\ |Q| \leqslant 2^R}} \mathbf{E}_{\substack{p \in \Omega_m \\ p \sim \theta}} \min_{q \in Q} d(p,q)\,, \tag{2}$$

However, we notice that in most applications, best possible accuracy of the reconstructed distribution is needed *instantaneously*. For example, in the design of a two-step universal code, sample-derived distribution is quantized and immediately used for encoding of this sample [26]. Similarly, in computer vision / image recognition applications, histograms of gradients from an image are extracted, quantized, and used right away to find nearest match for this image.

In all such cases, instead of minimizing the expected distance, it makes more sense to design a quantizer that minimizes the *worst case-* or *maximal distance* to the nearest reconstruction point. In other words, we need to solve the following problem [1]

$$d^*(\Omega_m, R) = \inf_{\substack{Q \subset \Omega_m \\ |Q| \leqslant 2^R}} \max_{p \in \Omega_m} \min_{q \in Q} d(p,q)\,. \tag{3}$$

We next survey some known results about it.

A. Achievable Performance Limits

We note that the problem (3) is purely geometric in nature. It is equivalent to the *problem of covering of Ω_m with at most 2^R balls of equal radius*. Related and immediately applicable results can be found in Graf and Luschgy [14, Chapter 10].

First, observe that Ω_m is a compact set in $\mathbb{R}^{m-1}$ (unit $m-1$-simplex), and that its volume in $\mathbb{R}^{m-1}$ can be computed as follows [29]:

$$\lambda^{m-1}(\Omega_m) = \frac{a^k}{k!}\sqrt{\frac{k+1}{2^k}}\,\Bigg|_{\substack{k=m-1 \\ a=\sqrt{2}}} = \frac{\sqrt{m}}{(m-1)!}\,. \tag{4}$$

Here and below we assume that $m \geqslant 3$.

Next, we bring result for asymptotic covering radius [14, Theorem 10.7]

$$\lim_{R \to \infty} 2^{\frac{R}{m-1}}\, d^*(\Omega_m, R) = C_{m-1} \sqrt[m-1]{\lambda^{m-1}(\Omega_m)}, \tag{5}$$

where $C_{m-1} > 0$ is a constant known as *covering coefficient for the unit cube*

$$C_{m-1} = \inf_{R \geqslant 0} 2^{\frac{R}{m-1}}\, d^*([0,1]^{m-1}, R). \tag{6}$$

[1] The dual problem:

$$R(\varepsilon) = \inf_{Q \subset \Omega_m :\, \max_{p \in \Omega_m} \min_{q \in Q} d(p,q) \leqslant \varepsilon} \log_2 |Q|\,,$$

may also be posed. The quantity $R(\varepsilon)$ can be understood as Kolmogorov's *ε-entropy* for metric space (Ω_m, d) [18].

The exact value of C_{m-1} depends on a distance measure $d(p,q)$. For example, for L_∞ norm

$$d_\infty(p,q) = ||p-q||_\infty = \max_i |p_i - q_i| \,,$$

it is known that

$$C_{m-1,\infty} = \tfrac{1}{2} \,.$$

Hereafter, when we work with specific L_r - norms:

$$d_r(p,q) = ||p-q||_r = \left(\sum_i |p_i - q_i|^r \right)^{1/r} \tag{7}$$

we will attach subscripts r to covering radius $d^*(.)$ and other expressions to indicate the type of norm being used.

By putting all these facts together, we obtain:

Proposition 1. *With $R \to \infty$:*

$$d_r^*(\Omega_m, R) \sim C_{m-1,r} \sqrt[m-1]{\tfrac{\sqrt{m}}{(m-1)!}} \; 2^{-\frac{R}{m-1}}, \tag{8}$$

where $C_{m-1,r}$ are some known constants.

We now note, that with large m the leading term in (8) turns into

$$\sqrt[m-1]{\tfrac{\sqrt{m}}{(m-1)!}} = \frac{e}{m} + O\left(\frac{1}{m^2}\right) \tag{9}$$

which is a decaying function of m. This highlights an interesting property of this quantization problem, distinguishing it from well known cases, such as quantization of the unit $(m-1)$-dimensional cube.

III. An Algorithm for Coding of Distributions

The following algorithm[2] can be viewed as a custom designed lattice quantizer. It is interesting in a sense that its lattice coincides with the concept of *types* in universal coding [8].

A. Choice of Lattice

Given some integer $n \geqslant 1$, we define a set:

$$Q_n = \left\{ [q_1, \ldots, q_m] \in \mathbb{Q}^m \middle| q_i = \tfrac{k_i}{n}, \textstyle\sum_i k_i = n \right\}, \tag{10}$$

where $n, k_1, \ldots, k_m \in \mathbb{Z}^+$. Parameter n serves as a common denominator to all fractions, and can be used to control the density and number of points in Q_n.

By analogy with the concept of *types* in universal coding [8] we will refer to distributions $q \in Q_n$ as types. For same reason we will (somewhat informally) call Q_n a *type lattice*. Several examples of sets Q_n are shown in Figure 1.

[2] An earlier publication related to this technique is [3].

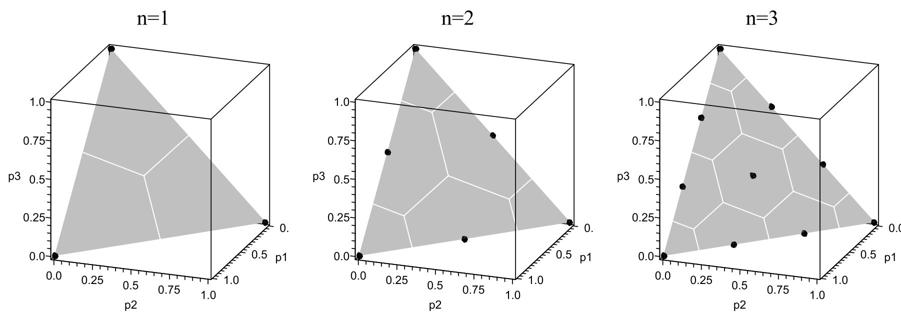

Figure 1. Examples of type lattices and their Voronoi cells (for L-norms) in 3 dimensions.

B. Quantization

The task of finding the nearest type in Q_n can be solved by using the following simple algorithm[3]:

Algorithm 1. *Given p, n, find nearest $q = \left[\frac{k_1}{n}, \ldots, \frac{k_m}{n}\right]$:*

1) *Compute values ($i = 1, \ldots, m$)*

$$k'_i = \left\lfloor np_i + \tfrac{1}{2} \right\rfloor\,, \quad n' = \textstyle\sum_i k'_i\,.$$

2) *If $n' = n$ the nearest type is given by: $k_i = k'_i$. Otherwise, compute errors*

$$\delta_i = k'_i - np_i\,,$$

and sort them such that

$$-\tfrac{1}{2} \leqslant \delta_{j_1} \leqslant \delta_{j_2} \leqslant \ldots \leqslant \delta_{j_m} \leqslant \tfrac{1}{2}\,,$$

3) *Let $\Delta = n' - n$. If $\Delta > 0$ then decrement d values k'_i with largest errors*

$$k_{j_i} = \left[\begin{array}{ll} k'_{j_i} & j = i, \ldots, m - \Delta - 1\,, \\ k'_{j_i} - 1 & i = m - \Delta, \ldots, m\,, \end{array}\right.$$

otherwise, if $\Delta < 0$ increment $|\Delta|$ values k'_i with smallest errors

$$k_{j_i} = \left[\begin{array}{ll} k'_{j_i} + 1 & i = 1, \ldots, |\Delta|\,, \\ k'_{j_i} & i = |\Delta| + 1, \ldots, m\,. \end{array}\right.$$

The correctness of this algorithm in terms of L_∞, L_1, and L_2-norms is self-evident.

C. Enumeration and Encoding

As mentioned earlier, the number of types in lattice Q_n depends on the parameter n. It is essentially the number of partitions of n into m terms $k_1 + \ldots + k_m = n$:

$$|Q_n| = \binom{n+m-1}{m-1}\,. \tag{11}$$

In order to encode a type with parameters $k_1, \ldots, k_m$, we need to obtain its unique index $\xi(k_1, \ldots, k_m)$. We suggest to compute it as follows:

$$\xi(k_1, \ldots, k_n) = \sum_{j=1}^{n-2} \sum_{i=0}^{k_j - 1} \binom{n - i - \sum_{l=1}^{j-1} k_l + m - j - 1}{m - j - 1} + k_{n-1}. \tag{12}$$

[3]This algorithm is similar in concept to Conway and Sloane's quantizer for lattice A_n [5, Chapter 20], but it is naturally scaled and reduced to find solutions within (m-1) simplex.

This formula follows by induction (starting with $m = 2, 3$, etc.), and it implements lexicographic enumeration of types. For example:

$$\begin{aligned} \xi(0,0,\ldots,0,n) &= 0\,, \\ \xi(0,0,\ldots,1,n-1) &= 1\,, \\ \ldots & \\ \xi(n,0,\ldots,0,0) &= \tbinom{n+m-1}{m-1} - 1\,. \end{aligned}$$

Similar combinatorial enumeration techniques were discussed in [7], [27], [28].

Once index is computed, it is transmitted by using direct binary representation at rate:

$$R(n) = \left\lceil \log_2 \tbinom{n+m-1}{m-1} \right\rceil\,. \tag{13}$$

D. *Performance Analysis*

The set of types Q_n is related to so-called *lattice* A_n in lattice theory [5, Chapter 4]. It can be understood as a bounded subset of A_n with $n = m - 1$ dimensions, which is subsequently scaled, and placed in the unit simplex.

Using this analogy, we can show that vertices of Voronoi cells (so called *holes*) in type lattice Q_n are located at:

$$q_i^* = q + v_i, \quad q \in Q_n, \quad i = 1, \ldots, m-1, \tag{14}$$

where v_i are so-called *glue vectors* [5, Chapter 21]:

$$v_i = \tfrac{1}{n}\Big[\underbrace{\tfrac{m-i}{m}, \ldots, \tfrac{m-i}{m}}_{i\ \text{times}},\ \underbrace{\tfrac{-i}{m}, \ldots, \tfrac{-i}{m}}_{m-i\ \text{times}}\Big]\,. \tag{15}$$

We next compute maximum distances (covering radii).

Proposition 2. *Let $a = \lfloor m/2 \rfloor$. The following holds:*

$$\max_{p\in\Omega_m} \min_{q\in Q_n} d_\infty(p,q) = \tfrac{1}{n}\left(1 - \tfrac{1}{m}\right)\,, \tag{16}$$

$$\max_{p\in\Omega_m} \min_{q\in Q_n} d_2(p,q) = \tfrac{1}{n}\sqrt{\tfrac{a(m-a)}{m}}\,, \tag{17}$$

$$\max_{p\in\Omega_m} \min_{q\in Q_n} d_1(p,q) = \tfrac{1}{n}\tfrac{2a(m-a)}{m}\,. \tag{18}$$

Proof: We use vectors (15). The largest component values appear when $i = 1$ or $i = m - 1$. E.g. for $i = 1$:

$$v_1 = \tfrac{1}{n}\left[\tfrac{m-1}{m}, \tfrac{-1}{m}, \ldots, \tfrac{-1}{m}\right]\,.$$

This produces L_∞ - radius. The largest absolute sum is achieved when all components are approximately the same in magnitude. This happens when $i = a$:

$$v_a = \tfrac{1}{n}\Big[\underbrace{\tfrac{m-a}{m}, \ldots, \tfrac{m-a}{m}}_{a\ \text{times}},\ \underbrace{\tfrac{-a}{m}, \ldots, \tfrac{-a}{m}}_{m-a\ \text{times}}\Big]\,.$$

This produces L_1 - radius. L_2 norm is the same for all vectors $v_i, i > 0$. ∎

It remains to evaluate distance / rate characteristics of type-lattice quantizer:

$$d_r^*[Q_n](\Omega_m, R) = \min_{n:|Q_n|\leqslant 2^R} \max_{p\in\Omega_m} \min_{q\in Q_n} d_r(p,q)\,.$$

We report the following.

Theorem 1. *Let $a = \lfloor m/2 \rfloor$. Then, with $R \to \infty$:*

$$d^*_\infty[Q_n](\Omega_m, R) \sim 2^{-\frac{R}{m-1}} \frac{1-\frac{1}{m}}{\sqrt[m-1]{(m-1)!}}, \quad (19)$$

$$d^*_2[Q_n](\Omega_m, R) \sim 2^{-\frac{R}{m-1}} \frac{\sqrt{\frac{a(m-a)}{m}}}{\sqrt[m-1]{(m-1)!}}, \quad (20)$$

$$d^*_1[Q_n](\Omega_m, R) \sim 2^{-\frac{R}{m-1}} \frac{\frac{2a(m-a)}{m}}{\sqrt[m-1]{(m-1)!}}. \quad (21)$$

Proof: We first obtain asymptotic (with $n \to \infty$) expansion for the rate of our code (13):

$$R = (m-1)\log_2 n - \log_2 (m-1)! + O\left(\tfrac{1}{n}\right).$$

This implies that

$$n \sim 2^{\frac{R}{m-1}} \sqrt[m-1]{(m-1)!}. \quad (22)$$

Statements of theorem are obtained by combination of this relation with expressions (16-18). ■

1) Optimality: We now compare the result of Theorem 1 with theoretical asymptotic estimates for covering radius for Ω_m (8). As evident, the maximum distance in our scheme decays with the rate R as:

$$d^*[Q_n](\Omega_m, R) \sim 2^{-\frac{R}{m-1}},$$

which matches the decay rate of theoretical estimates.

The only difference is in a leading constant factor. For example, under L_∞ norm, such factor in expression (8) is

$$\frac{1}{2}\sqrt[m-1]{\sqrt{m}} = \frac{1}{2} + O\left(\frac{\log m}{m}\right).$$

Our algorithm, on the other hand, uses a factor

$$1 - \frac{1}{m},$$

which starts with $\frac{1}{2}$ when $m = 2$, and tends to 1 with $m \to \infty$.

2) Performance in terms of KL-distance: All previous results are obtained using L-norms. Such distance measures are common in computer vision and image recognition applications [1], [20], [22]. In source coding, main interest presents Kullback-Leibler (KL) distance:

$$d_{\mathrm{KL}}(p,q) = D(p||q) = \sum_i p_i \log_2 \frac{p_i}{q_i}. \quad (23)$$

This is not a true distance measure, so precise analysis is complicated. Yet, by observing[4] that for small distances between p and q:

$$d_{\mathrm{KL}}(p,q) \sim \tfrac{1}{2\ln 2} \sum_i \frac{(p_i - q_i)^2}{q_i} \quad (24)$$

[4]For example, by considering Tailor series of $h(p) = -\sum_i p_i \log p_i$ at point q: $h(p) = h(q) + (p-q)^T \nabla h(q) + \frac{1}{2}(p-q)^T \nabla^2 h(q)$ (p-q) + $\ldots$, we note that $D(p||q)$ is just a tail of this series: $D(p||q) = h(q) - h(p) + (p-q)^T \nabla h(q)$.

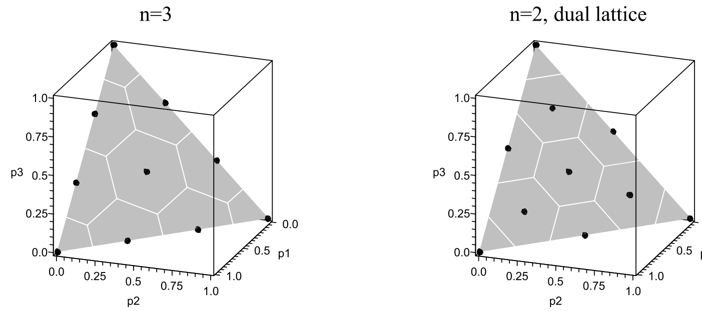

Figure 2. Two 10-point lattices: Q_3 (left), and Q_2^* (right). The second has noticeably smaller cells.

we can at least say, that for a cell in the middle of the simplex $\left\{q_i = \frac{1}{m}\right\}$, KL-distances to the nearest holes q^* will satisfy:

$$d_{\mathrm{KL}}(q^*, q) \sim \frac{m}{2\ln 2}\left[d_2(q^*, q)\right]^2 = \frac{1}{2\ln 2}\frac{a(m-a)}{n^2} = O\left(\frac{1}{n^2}\right). \tag{25}$$

As evident from (24), KL-radii of our cells will increase as we move point q closer to the boundary of the simplex. However, in the asymptotic sense (and excluding boundary points), their magnitude will still decrease at the rate of $O\left(n^{-2}\right)$. This follows, for example, from inequality: $\sum_i \frac{(p_i-q_i)^2}{q_i} \leqslant \sum_i (p_i - q_i)^2 \left(\max_i \frac{1}{q_i}\right)$. By translating n to bitrate (cf. (22)), we obtain

$$d_{\mathrm{KL}}(q^*, q) = O\left(2^{-\frac{2R}{m-1}}\right). \tag{26}$$

Precise lower bounds, supporting this simple estimate, can be obtained by using Pinsker inequality [23], and its recent refinements [10].

E. Additional improvements

1) Introducing bias: As easily observed, type lattice Q_n places reconstruction points with $k_i = 0$ precisely on edges of the probability simplex Ω_m. This is not best placement from quantization standpoint, particularly when n is small. This placement can be improved by using *biased types*:

$$q_i = \frac{k_i + \beta}{n + \beta m}, \quad i = 1, \ldots, m,$$

where $\beta \geqslant 0$ is a constant that defines shift towards the middle of the simplex. In traditional source coding applications, it is customary to use $\beta = 1/2$ [19]. In our case, setting $\beta = 1/m$ appears to work best for L-norms, as it introduces same distance to edges of the simplex as covering radius of the lattice.

2) Using dual type lattice Q_n^*: Another idea for improving performance of our quantization algorithm is to define and use *dual type lattice* Q_n^*. Such a lattice consists of all points:

$$q^* = q + v_i, \quad q \in Q_n, \quad q^* \in \Omega_m \quad i = 0, \ldots, m-1,$$

where v_i are the glue vectors (15). The main advantage of using dual lattice is thinner covering in high dimensions (cf. [5, Chapter 2]). But even at small dimensions, it may sometimes be useful. An example of such a situation for $m = 3$ is shown in Figure. 2.

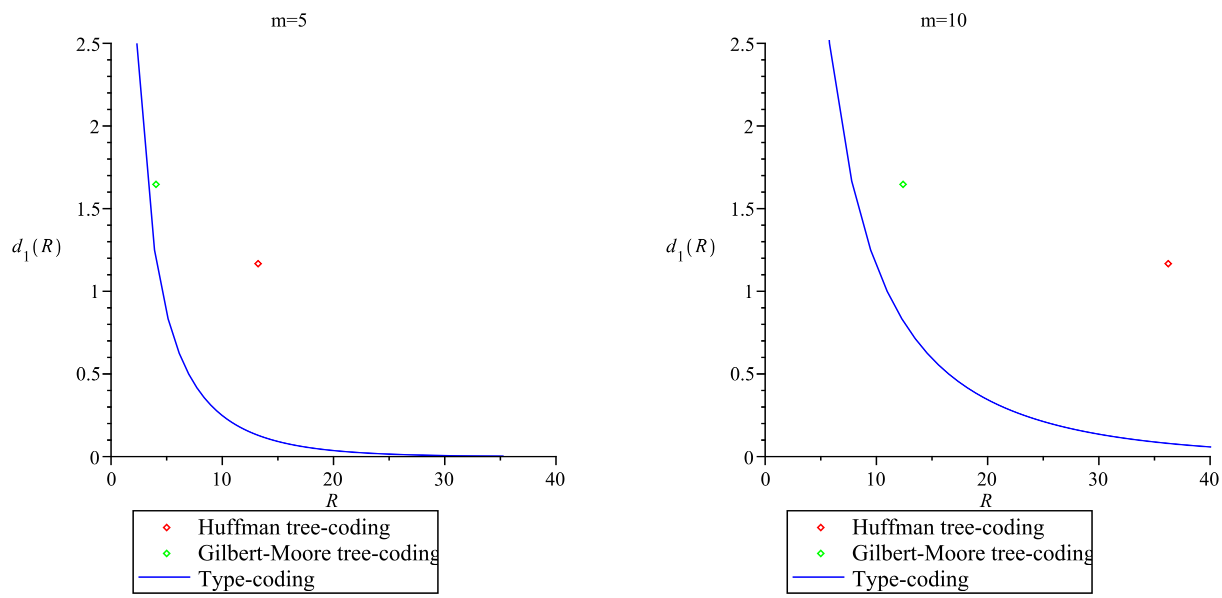

Figure 3. L_1-radius vs rate characteristics $d_1^*[\mathrm{H}](R)$, $d_1^*[\mathrm{GM}](R)$, $d_1^*[\mathrm{Q_n}](R)$ achievable by Huffman-, Gilbert-Moore-, and type-based quantization schemes.

IV. Comparison with Tree-Based Quantization Techniques

Given a probability distribution $p \in \Omega_m$, one popular in practice way of compressing it is to design a prefix code (for example, a Huffman code) for this distribution p first, and then encode the binary tree of such a code. Below we summarize some known results about performance of such schemes.

By denoting by $\ell_1, \ldots, \ell_m$ lengths of prefix codes, recalling that they satisfy Kraft inequality [6], and noting that $2^{-\ell_i}$ can be used to map lengths back to probabilities, we arrive at the following set:

$$Q_{\text{tree}} = \left\{ [q_1, \ldots, q_m] \in \mathbb{Q}^m \middle| q_i = 2^{-\ell_i}, \textstyle\sum_i 2^{-\ell_i} \leqslant 1 \right\}.$$

There are several specific algorithms that one can employ for construction of codes, producing different subsets of Q_{tree}. Below we only consider the use of classic Huffman and Gilbert-Moore [13] codes. Some additional tree-based quantization schemes can be found in [11].

Proposition 3. *There exists a set $Q_{\text{GM}} \subset Q_{\text{tree}}$, such that*

$$d_{\text{KL}}^*[Q_{\text{GM}}](R_{\text{GM}}) \leqslant 2, \tag{27}$$

$$d_1^*[Q_{\text{GM}}](R_{\text{GM}}) \leqslant 2\sqrt{\ln 2}, \tag{28}$$

$$d_\infty^*[Q_{\text{GM}}](R_{\text{GM}}) \leqslant 1, \tag{29}$$

where

$$R_{\text{GM}} = \log_2 |Q_{\text{GM}}| = \log_2 C_{m-1} = 2\,m - \tfrac{3}{2}\log_2 m + O(1), \tag{30}$$

and $C_n = \frac{1}{n+1}\binom{2n}{n}$ is the Catalan number.

Proof: We use Gilbert-Moore code [13]. Upper bound for KL-distance is well known [13]. L_1 bound follows by Pinsker inequality [23]: $d_{\text{KL}}(p,q) \geqslant \frac{1}{2\ln 2}\, d_1(p,q)^2$. L_∞ bound is obvious: $p_i, q_i \in (0,1)$. Gilbert-Moore code uses fixed assignment (e.g. from left to right) of letters to the codewords. Any binary rooted tree with m leaves can serve as a code. The number of such trees is given by the Catalan number C_{m-1}. ■

Proposition 4. *There exists a set $Q_{\mathrm{H}} \subset Q_{\mathrm{tree}}$, such that*

$$\begin{aligned} d^*_{\mathrm{KL}}[Q_{\mathrm{H}}](R_{\mathrm{H}}) &\leqslant 1\,, & (31)\\ d^*_1[Q_{\mathrm{H}})](R_{\mathrm{H}}) &\leqslant \sqrt{2\ln 2}\,, & (32)\\ d^*_\infty[Q_{\mathrm{H}}](R_{\mathrm{H}}) &\leqslant \tfrac{1}{2}\,, & (33)\end{aligned}$$

where

$$R_{\mathrm{H}} = \log_2 |Q_{\mathrm{H}}| = m \log_2 m + O\left(m\right)\,. \quad (34)$$

Proof: We use Huffman code. Its KL-distance bound is well known [6]. L_1 bound follows by Pinsker inequality [23]. L_∞ bound follows from sibling property of Huffman trees [12]. It remains to estimate the number of Huffman trees T_m with m leaves. Consider a skewed tree, with leaves at depths $1, 2, \ldots, m-1, m-1$. The last two leaves can be labeled by $\binom{m}{2}$ combinations of letters, whereas the other leaves - by $(m-2)!$ possible combinations. Hence $T_m \geqslant \binom{m}{2}(m-2)! = \frac{1}{2}m!$. Upper bound is obtained by arbitrary labeling all binary trees with m leaves: $T_m < m!\,C_{m-1}$, where C_{m-1} is the Catalan number. Combining both we obtain: $-\frac{1}{\ln 2}m < \log_2 T_m - m\log_2 m < \left(2 - \frac{1}{\ln 2}\right)m$. ■

We present comparison of maximal L_1 distances achievable by tree-based and type-based quantization schemes in Figure 3. We consider cases of $m = 5$ and $m = 10$ dimensions. It can be observed that the proposed type-based scheme is more efficient and more versatile, allowing a wide range of possible rate/distance tradeoffs.

V. Conclusions

The problem of quantization of discrete probability distributions is studied. It is shown, that in many cases, this problem can be reduced to the covering radius problem for the unit simplex. Precise characterization of this problem in high-rate regime is reported. A simple algorithm for solving this problem is also presented, analyzed, and compared to other known solutions.

Acknowledgements

The author would like to thank Prof. K. Zeger (UCSD) for reviewing and providing very useful comments on the initial draft of this paper. The author also wishes to thank Prof. B. Girod and his students V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai (Stanford University), and Dr. R. Grzeszczuk (Nokia Research Center, Palo Alto) for introduction to the field of computer vision, and collaboration that prompted study of this quantization problem [3].

References

[1] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[2] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, B. Girod, "CHoG: Compressed histogram of gradients A low bit-rate feature descriptor," in *Proc. Computer Vision and Pattern Recognition (CVPR'09)*, pp. 2504–2511, 2009.

[3] V. Chandrasekhar, Y. Reznik, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "Quantization Schemes for Low Bitrate Compressed Histogram of Gradient Descriptors," in *Proc. Computer Vision and Pattern Recognition (CVPR'10)*, 2010.

[4] P. A. Chou, M. Effros, and R. M. Gray, "A vector quantization approach to universal noiseless coding and quantization," *IEEE Trans. Information Theory*, vol. 42, no. 4, pp. 1109–1138, 1996.

[5] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York: Springer-Verlag, 1998.

[6] T. M. Cover and J. M. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, 2006.

[7] T. Cover, "Enumerative source coding," *IEEE Trans. Inform. Theory*, vol. 19, pp. 73–76, Jan. 1973.
[8] I. Csiszár, "The method of types," *IEEE Trans. Inf. Theory*, vol.44, no. 66, pp. 2505–2523, 1998.
[9] L. D. Davisson, "Comments on 'Sequence time coding for data compression," *Proc. IEEE*, vol. 54, p. 2010. Dec. 1966.
[10] A. A. Fedotov, P. Harremoës, and F. Topsøe, "Refinements of Pinsker's inequality," *IEEE Trans. Inf. Theory*, vol. 49, no. 6, pp. 1491–1498, 2003.
[11] T. Gagie, Compressing Probability Distributions, *Information Processing Letters*, vol. 97, no. 4, pp. 133–137, 2006.
[12] R. Gallager, "Variations on a theme by Huffman," *IEEE Trans. Inform. Theory*, vol.24, no.6, pp. 668–674, Nov 1978.
[13] E. N. Gilbert and E. F. Moore, "Variable-Length Binary Encodings," *The Bell System Tech. Journal*, vol. 7, pp. 932–967, 1959.
[14] S. Graf, and H. Luschgy, *Foundations of Quantization for Probability Distributions*. Berlin: Springer-Verlag, 2000.
[15] T. S. Han, and K. Kobayashi, *Mathematics of Information and Coding*. Boston: American Mathematical Society, 2001.
[16] ITU-T and ISO/IEC JTC1, "Digital Compression and Coding of Continuous-Tone Still Images," ISO/IEC 10918-1 — ITU-T Rec. T.81, Sept. 1992.
[17] P. W. Katz, PKZIP. Data compression utility, version 1.1, 1990.
[18] A. N. Kolmogorov and V. M. Tikhomirov, "ε-entropy and ε-capacity of sets in metric spaces," *Uspekhi Math. Nauk*, vol. 14, no. 2, pp. 3–86, 1959. (in Russian)
[19] R. E. Krichevsky and V. K. Trofimov, "The Performance of Universal Encoding," *IEEE Trans. Information Theory*, vol. 27, pp. 199–207, 1981.
[20] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[21] T. J. Lynch, "Sequence time coding for data compression," *Proc. IEEE*, vol. 54, pp. 1490–1491, 1966.
[22] K. Mikolajczyk and C. Schmid, "Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, 2005.
[23] M. S. Pinsker, "Information and Information Stability of Random Variables and Processes," *Problemy Peredachi Informacii*, vol. 7, AN SSSR, Moscow 1960. (in Russian).
[24] R. F. Rice and J. R. Plaunt, "Adaptive variable length coding for efficient compression of spacecraft television data," *IEEE Trans. Comm. Tech.*, vol. 19, no.1, pp. 889–897, 1971.
[25] J. Rissanen, "Universal coding, information, prediction and estimation," *IEEE Trans. Inform. Theory*, vol. 30, pp. 629–636, 1984.
[26] J. Rissanen, "Fisher Information and Stochastic Comprexity," *IEEE Trans. Inform. Theory*, vol. 42, pp. 40–47, 1996.
[27] J. P. M. Schalkwijk, "An algorithm for source coding," *IEEE Trans. Inform. Theory*, vol. 18, pp. 395–399, May 1972.
[28] Yu. M. Shtarkov and V. F. Babkin, "Combinatorial encoding for discrete stationary sources," in *Proc. 2nd Int. Symp. on Information Theory*, Akadémiai Kiadó, 1971, pp. 249–256.
[29] D. Sommerville, *An Introduction to the Geometry of n Dimentions*. New York: Dover, 1958.
[30] K. Zeger, A. Bist, and T. Linder, "Universal Source Coding with Codebook Transmission," *IEEE Trans. Communications*, vol. 42, no. 2, pp. 336–346, 1994.