QUANTIZATION OF DISCRETE PROBABILITY DISTRIBUTIONS

Yuriy A. Reznik

Qualcomm Incorporated
5775 Morehouse Dr., San Diego, CA 92121
yreznik@qualcomm.com

ABSTRACT

We study the problem of quantization of discrete probability distributions. We show that in many cases this problem becomes equivalent to the covering problem for the unit simplex. Such setting yields precise asymptotic characterization of the quantization problem in high-rate regime. Our main contribution is a simple and asymptotically optimal algorithm for solving this problem. Performance of this algorithm is studied and compared with several other known solutions.

1. 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* [20, 8], *combinatorial* [27], and *enumerative* codes [6] used lossless encoding of frequencies of symbols in the input sequence. The *Rice machine* [23], developed in 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 in the encoding process [24, 25]. Vector quantization techniques for two-step universal coding were studied in [29, 3].

A related practical development was the idea of compressing and transmitting Huffman code trees, prior to transmitting data encoded by them. Such algorithms become very popular in 1980s and 1990s, and were used, for example, in ZIP archiver [16], and JPEG image compression standard [15].

In recent years, the problem of coding of distributions has attracted a new wave of interest coming from other fields. For example, modern computer vision algorithms, such as SIFT [19], SURF [1], or CHoG [2] are using histograms of gradients collected from images. Such histograms are usually noisy and redundant, and their quantization becomes an essential step in the design of these algorithms. Several other interesting uses of coding of distributions are described in [10].

This paper is intended to produce a concise definition of this problem, survey applicable mathematical facts, and offer a simple practical algorithm for solving it.

In Section 2, we introduce notation and formulate the problem. In Section 3, we study achievable performance limits. In Section 4, we describe design of our proposed algorithm. In Section 5, we provide comparisons with other known techniques. Conclusions are drawn in Section 6.

2. 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:

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

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 (Shannon's) setting of the quantization problem for distribution $p \in \Omega_m$, it remains to assume that it 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 *expected distance* to the nearest reconstruction point (cf. [13, Lemma 3.1])

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

However, we now notice that in most applications of quantization of probability distributions, *best accuracy of the reconstruction is required instantaneously*! For example, in the design of a two-part universal code, empirical distribution of a sample of data is quantized and used for the purpose of encoding of this particular sample [25]. Similarly, in computer vision / image recognition applications, quantized histograms from a given image are produced and used right away to find its match.

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

following problem [1]

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

We next survey some known results about it.

3. 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 [13, Chapter 10].

First, observe that Ω_m is a compact set in $\mathbb{R}^{m-1}$ (it is a unit $m-1$-simplex), and that its volume in $\mathbb{R}^{m-1}$ can be computed as follows [28]

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

For simplicity, in the above and subsequent formulae we assume that $m \geqslant 3$.

Next, we bring result for asymptotic covering radius [13, Theorem 10.7]

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

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

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

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:

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

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

By putting all these facts together, we obtain:

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

$$d^*_\infty(\Omega_m, R) \sim \tfrac{1}{2}\sqrt[m-1]{\tfrac{\sqrt{m}}{(m-1)!}}\, 2^{-\frac{R}{m-1}} \qquad (8)$$

and more generally (for other L_r-norms, $r \geqslant 1$):

$$d^*_r(\Omega_m, R) \sim C_{m-1,r}\sqrt[m-1]{\tfrac{\sqrt{m}}{(m-1)!}}\, 2^{-\frac{R}{m-1}}, \qquad (9)$$

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

[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 resulting quantity $R(\varepsilon)$ can be understood as Kolmogorov's *ε-entropy* for metric space (Ω_m, d) [17].

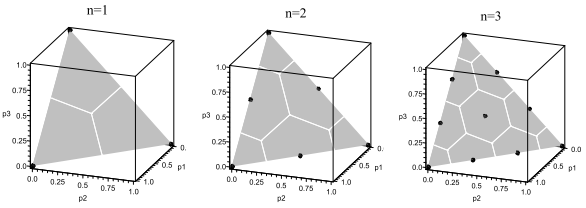

Figure 1. Examples of type lattices ($m = 3, n = 1, 2, 3$).

In other words, we have precise asymptotic characterization of best attainable performance for quantizers of probability distribution.

Our next task is to design such an algorithm.

4. PRACTICAL ALGORITHM FOR CODING OF DISTRIBUTIONS

4.1. Algorithm design

Our algorithm 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.

4.1.1. Type Lattice

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

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

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 [7] we will refer to distributions $q \in Q_n$ as types. For same reason we will call Q_n a *type lattice*. Several examples of type lattices are shown in Figure 1.

4.1.2. Quantization

The task of finding the nearest type in Q_n can be solved by using the following simple algorithm.

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

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

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

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*

$$k_{j_i} = \begin{bmatrix} k'_{j_i} & j = i, \ldots, m-\Delta-1\,, \\ k'_{j_i} - 1 & i = m-\Delta, \ldots, m\,, \end{bmatrix}$$

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

$$k_{j_i} = \begin{cases} k'_{j_i} + 1 & i = 1, \ldots, |\Delta| \,, \\ k'_{j_i} & i = |\Delta| + 1, \ldots, m \,. \end{cases}$$

The correctness of this algorithm is self-evident. By using quick-select instead of full sorting in step 2, its run time can be reduced to $O(m)$.

4.1.3. 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:

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

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 [27, 6, 26]. With precomputed array of binomial coefficients, the computation of index by using this formula requires $O(n)$ operations.

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

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

4.2. Analysis

Type lattice Q_n is related to so-called A_n lattice in quantization theory [4, Chapter 4]. It can be understood as a bounded subset of A_n with $n = m-1$ dimensions, scaled, and placed to fill the space of the unit simplex.

Vertices of Voronoi cells for type lattice Q_n can be defined by using vectors (cf. [4, Chapter 21]):

$$v_i = \frac{1}{n} \Big[\underbrace{\tfrac{m-i}{m}, \ldots, \tfrac{m-i}{m}}_{i \text{ times}}, \underbrace{\tfrac{-i}{m}, \ldots, \tfrac{-i}{m}}_{m-i \text{ times}} \Big] . \tag{14}$$

The actual coordinates of Voronoi cell vertices (so called *holes* in lattice Q_m) are

$$q_i^* = q + v_i, \quad q \in Q_n, \quad i = 1, \ldots, m-1. \tag{15}$$

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 (14). 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 $i = a$:

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

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

It remains to evaluate maximum distance / rate characteristics of type-based 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* $R \to \infty$*:*

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

$$d_2^*[Q_n](\Omega_m, R) \sim 2^{-\frac{R}{m-1}} \frac{\sqrt{\frac{a(m-a)}{m}}}{\sqrt[m-1]{(m-1)!}} , \tag{20}$$

$$d_1^*[Q_n](\Omega_m, R) \sim 2^{-\frac{R}{m-1}} \frac{\frac{2a(m-a)}{m}}{\sqrt[m-1]{(m-1)!}} . \tag{21}$$

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

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

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

4.2.1. Optimality

We now compare the result of Theorem 1 with theoretical asymptotic estimates for covering radius for Ω_m (8, 9). 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 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

$$\frac{1}{2} \leqslant 1 - \frac{1}{m} < 1,$$

which starts with $\frac{1}{2}$ when $m = 2$. This suggests that even in terms of leading constant our algorithm is close to the optimal. It is particularly efficient when the number of dimensions m is small.

4.2.2. Performance in terms of KL-distance

All previous results are obtained using L-norms. Such distance measures are common in computer vision applications [19, 21, 1]. In source coding, main interest presents Kullback-Leibler (KL) distance:

$$d_{\mathrm{KL}}(p,q) = D(p||q) = \sum_i p_i \log_2 \frac{p_i}{q_i}. \qquad (22)$$

It is not a true distance, so the exact analysis is complicated. Yet, by using Pinsker inequality [22]

$$d_{\mathrm{KL}}(p,q) \geqslant \tfrac{1}{2\ln 2}\, d_1(p,q)^2, \qquad (23)$$

we can at least show that for deep holes

$$d_{\mathrm{KL}}(q^*,q) \geqslant \tfrac{1}{2\ln 2}\left(\tfrac{1}{n}\tfrac{2a(m-a)}{m}\right)^2.$$

By translating n to bitrate, we obtain

$$d_{\mathrm{KL}}(q^*,q) \gtrsim \tfrac{1}{2\ln 2}\, 2^{-\frac{2R}{m-1}} \left(\frac{\frac{2a(m-a)}{m}}{\sqrt[m-1]{(m-1)!}}\right)^2. \qquad (24)$$

More precise bounds can be obtained by using inequalities described in [9].

4.3. Additional improvements

4.3.1. 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 most efficient from quantization standpoint, particularly when n is small. This can be fixed 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 $\beta = 1/2$ (cf. [18]). In our case, picking β from the range $[0, 1/m]$ will ensure that edges of the simplex are covered.

Algorithm 1 can be easily adjusted to find nearest points in such modified lattice.

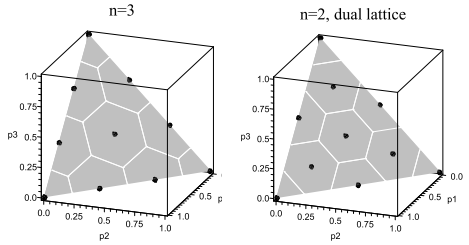

Figure 2. 10-point lattices: Q_3 (left), and Q_2^* (right).

4.3.2. 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:

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

where v_i are the glue vectors (14).

The main advantage of using dual lattice would be thinner covering at high dimensions (cf. [4, Chapter 2]). But even at small dimensions, it may sometimes be useful. An example of this for $m = 3$ is shown in Figure. 2.

5. COMPARISON WITH OTHER 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.

5.1. Performance of tree-based quantizers

By denoting by $\ell_1, \ldots, \ell_m$ lengths of prefix codes, recalling that they satisfy Kraft inequality [5], and noting that $2^{-\ell_i}$ can be used to map lengths back to probabilities, we arrive at the following lattice:

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

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 [12] codes. Some additional tree-based quantization schemes can be found in [10].

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

$$d^*_{\mathrm{KL}}[Q_{\mathrm{GM}}](R_{\mathrm{GM}}) \leqslant 2, \qquad (25)$$
$$d^*_1[Q_{\mathrm{GM}}](R_{\mathrm{GM}}) \leqslant 2\sqrt{\ln 2}, \qquad (26)$$
$$d^*_\infty[Q_{\mathrm{GM}}](R_{\mathrm{GM}}) \leqslant 1, \qquad (27)$$

where

$$R_{\mathrm{GM}} = \log_2 |Q_{\mathrm{GM}}| = \log_2 C_{m-1} = 2m - \tfrac{3}{2}\log_2 m + O(1), \qquad (28)$$

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

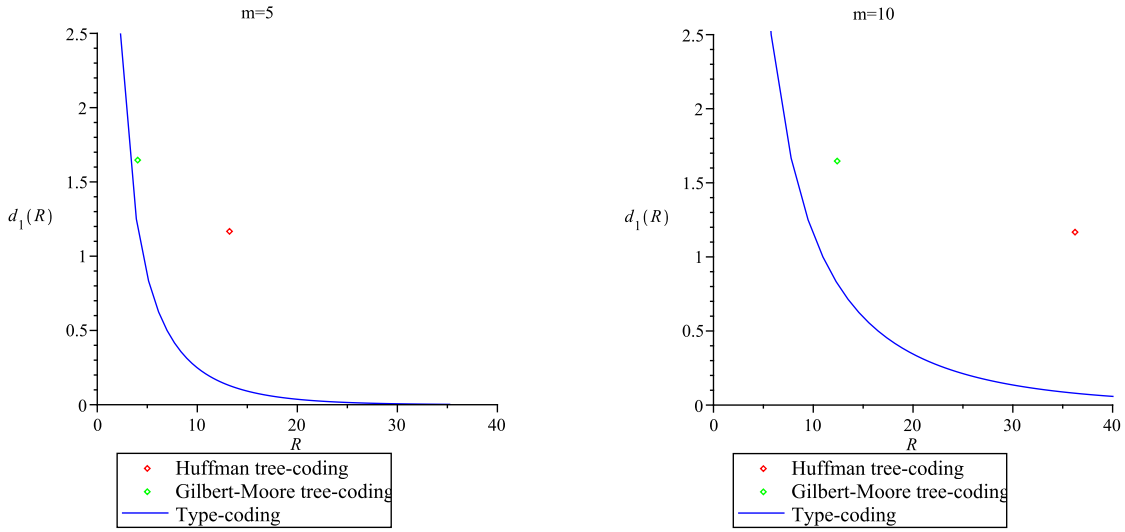

Figure 3. Maximal L_1 distances vs rate characteristics $d_1^*[\mathrm{H}](R)$, $d_1^*[\mathrm{GM}](R)$, $d_1^*[\mathrm{Q_n}](R)$ achievable by Huffman-, Gilbert-Moore-, and type-based quantization schemes.

Proof. We use Gilbert-Moore code [12]. Upper bound for KL-distance is well known [12]. L_1 bound follows by Pinsker's inequality (23). 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 lattice $Q_\mathrm{H} \subset Q_\mathrm{H}$, such that*

$$d^*_{\mathrm{KL}}[Q_\mathrm{H}](R_\mathrm{H}) \leqslant 1\,, \tag{29}$$
$$d^*_1[Q_\mathrm{H})](R_\mathrm{H}) \leqslant \sqrt{2\ln 2}\,, \tag{30}$$
$$d^*_\infty[Q_\mathrm{H}](R_\mathrm{H}) \leqslant \tfrac{1}{2}\,, \tag{31}$$

where

$$R_\mathrm{H} = \log_2 |Q_\mathrm{H}| = m\log_2 m + O(m)\,. \tag{32}$$

Proof. We use Huffman code. Its KL-distance bound is well known [5]. L_1 bound follows by Pinsker's inequality. L_∞ bound follows from sibling property of Huffman trees [11]. 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: $-\frac{1}{\ln 2}m < \log_2 T_m - m\log_2 m < \left(2 - \frac{1}{\ln 2}\right)m$. □

5.2. Comparison

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 much more versatile, allowing a wide range of possible rate/distance tradeoffs.

6. 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.

7. ACKNOWLEDGMENTS

The author would like to thank Prof. K. Zeger (UCSD) for reviewing and providing very useful comments on a draft version of this paper.

8. 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)*, 2009, pp. 2504-2511.

[3] 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.

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

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

[6] T. Cover, "Enumerative source coding," *IEEE Trans. Inform. Theory*, vol. 19, pp. 73–76, Jan. 1973.

[7] I. Csiszár, "The method of types," *IEEE Trans. Inf. Theory*, vol.44, no. 66, pp. 2505–2523, 1998.

[8] L. D. Davisson, "Comments on 'Sequence time coding for data compression," *Proc. IEEE*, vol. 54, p. 2010. Dec. 1966.

[9] 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.

[10] T. Gagie, Compressing Probability Distributions, *Information Processing Letters*, vol. 97, no. 4, pp. 133–137, 2006.

[11] R. Gallager, "Variations on a theme by Huffman," *IEEE Trans. Inform. Theory*, vol.24, no.6, pp. 668-674, Nov 1978.

[12] E. N. Gilbert and E. F. Moore, "Variable-Length Binary Encodings," *The Bell System Tech. Journal*, vol. 7, pp. 932–967, 1959.

[13] S. Graf, and H. Luschgy, *Foundations of Quantization for Probability Distributions*. Berlin: Springer-Verlag, 2000.

[14] T. S. Han, and K. Kobayashi, *Mathematics of Information and Coding*. Boston: American Mathematical Society, 2001.

[15] ITU-T and ISO/IEC JTC1, "Digital Compression and Coding of Continuous-Tone Still Images," ISO/IEC 10918-1 — ITU-T Recommendation T.81 (JPEG), Sept. 1992.

[16] P. W. Katz, PKZIP. Commercial compression system, version 1.1, 1990.

[17] 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)

[18] R. E. Krichevsky and V. K. Trofimov, "The Performance of Universal Encoding," *IEEE Trans. Information Theory*, vol. 27, pp. 199–207, 1981.

[19] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91110, 2004.

[20] T. J. Lynch, "Sequence time coding for data compression," *Proc. IEEE*, vol. 54, pp. 1490-1491, Oct. 1966.

[21] K. Mikolajczyk and C. Schmid, "Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 16151630, 2005.

[22] M. S. Pinsker, "Information and Information Stability of Random Variables and Processes," *Problemy Peredachi Informacii*, vol. 7, AN SSSR, Moscow 1960. (in Russian).

[23] 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.

[24] J. Rissanen, "Universal coding, information, prediction and estimation," *IEEE Trans. Inform. Theory*, vol. 30, pp. 629-636, 1984.

[25] J. Rissanen, "Fisher Information and Stochastic Comprexity," *IEEE Trans. Inform. Theory*, vol. 42, pp. 40-47, 1996.

[26] J. P. M. Schalkwijk, "An algorithm for source coding," *IEEE Trans. Inform. Theory*, vol. 18, pp. 395–399, May 1972.

[27] 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.

[28] D.M.Y. Sommerville, *An Introduction to the Geometry of n Dimentions*. New York: Dover, 1958.

[29] K. Zeger, A. Bist, and T. Linder, "Universal Source Coding with Codebook Transmission," *IEEE Trans. Communications*, vol. 42, no. 2, pp. 336–346, 1994.