An MCMC Approach
to Lossy Compression of Continuous Sources

Dror Baron
Electrical Engineering Department
Technion – Israel Institute of Technology
Haifa, Israel
Email: drorb@ee.technion.ac.il

Tsachy Weissman
Department of Electrical Engineering
Stanford University
Stanford, CA
Email: tsachy@stanford.edu

Abstract

Motivated by the Markov chain Monte Carlo (MCMC) relaxation method of Jalali and Weissman, we propose a lossy compression algorithm for continuous amplitude sources that relies on a finite reproduction alphabet that grows with the input length. Our algorithm asymptotically achieves the optimum rate distortion (RD) function universally for stationary ergodic continuous amplitude sources. However, the large alphabet slows down the convergence to the RD function, and is thus an impediment in practice. We thus propose an MCMC-based algorithm that uses a (smaller) adaptive reproduction alphabet. In addition to computational advantages, the reduced alphabet accelerates convergence to the RD function, and is thus more suitable in practice.

Keywords

Lossy compression; rate distortion theory; stationary ergodic sources; universal compression

I. Introduction

Lossy compression of continuous amplitude sources has numerous applications and is used to reduce data rates in modern communication systems while relaying the data at the necessary fidelity level. Despite numerous potential applications such as image compression [1, 2], video compression [3], and speech coding [4], there has been a significant gap between theory and practice.

On the theoretical front, recent advances in lossy compression have demonstrated that the *rate distortion* (RD) function can be approached asymptotically for memoryless sources over a finite alphabet [5–7]. These approaches partition an input into sub-blocks, and a Shannon-style codebook [8, 9] is applied to each sub-block. In a sense, both Gioran and Kontoyiannis [5, 7] and Gupta et al. [6] perform vector quantization over sub-blocks. Some of these approaches can compress universally without knowing the source statistics beforehand, but it is challenging to generate a codebook distribution whose statistics differ from the input statistics [10]. Another approach to lossy compression relies on algebraic codes [11].

In contrast to the finite alphabet case, less progress has been made in developing theoretically-motivated compression algorithms for continuous amplitude sources. Some results have been derived specifically for the high-rate regime, where the Shannon lower bound is asymptotically tight [12]. In particular, the low-distortion limit of the RD function has been characterized for mixtures of *probability distribution functions* (pdf's) where one distribution is discrete and the other continuous [13, 14].

Despite the theoretical insights in the high-rate regime, low-to-medium rates are of greater interest in many applications. In practice, many schemes rely on entropy coding, where scalar quantization is followed by lossless coding. Actual translation to bits can use Huffman [15] or arithmetic [8, 16]

The second author is on leave from the Electrical Engineering Department, Technion.

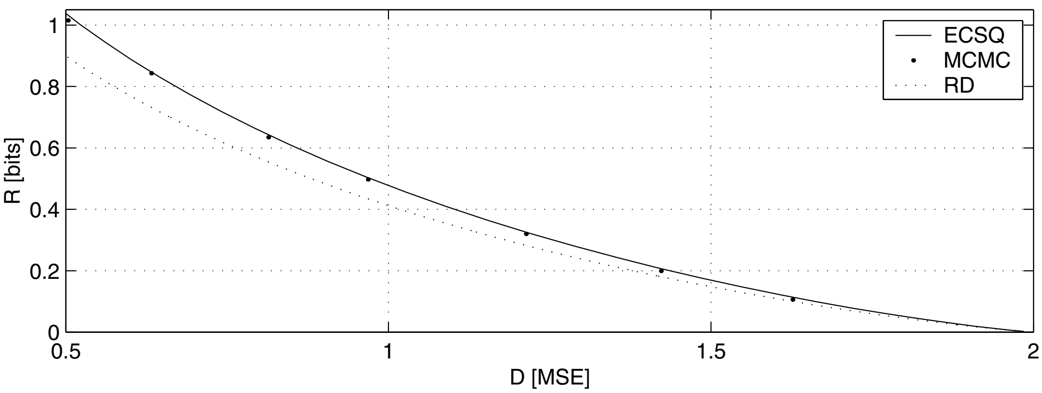

Figure 1. Comparison of entropy coding (ECSQ), Algorithm 1 (MCMC), and the RD function. ($n = 10^5$, $|\mathcal{Z}| \in \{3, 5, 7\}$, $r = 100$, $k \approx \frac{1}{2}\log_{|\mathcal{Z}|}(n)$, and $c = 1.6$.)

codes. *Entropy constrained scalar quantization* (ECSQ) has motivated much work into optimization of scalar quantizers [17]. Using ECSQ, many sources are sub-optimal with respect to (w.r.t.) the RD function [8, 9], sometimes by 10% or more (Figure 1). To bridge part of the gap between ECSQ and the RD function, *vector quantization* (VQ) quantizes an entire vector to a codeword, whereas scalar quantization compresses individual input elements. VQ provides better performance as the vector dimension increases, but increased complexity is required [18]. An interesting approach for compressing continuous amplitude sources universally was offered by Yang and Zhang [19], but their recommended algorithm relies on availability of a training sequence.

Inspired by Jalali and Weissman [20], we use the MCMC simulated annealing approach [21] to obtain the reproduction sequence directly. We first propose a lossy compression algorithm for continuous amplitude sources that relies on a *fixed reproduction alphabet* that grows with the input length. Our algorithm asymptotically achieves the optimum RD function universally for stationary ergodic continuous amplitude sources. However, the large alphabet slows down the convergence to the RD function, and is thus an impediment in practice. Next, we propose a second MCMC-based algorithm that uses a (smaller) *adaptive reproduction alphabet*. The seminal work by Rose on the discrete nature of the reproduction alphabet when the Shannon lower bound is not tight [22] suggests that in many cases of practical interest a small alphabet can be employed without compromising the ability to attain the fundamental compression limits. The smaller reproduction alphabet simplifies the implementation, and accelerates convergence to the RD function. The adaptive algorithm is thus more suitable for practical applications. We emphasize that our algorithms are universal; there is no need to know the source statistics.

The paper is organized as follows. We provide background information in Section II. Our fixed alphabet algorithm is described in Section III, followed by the adaptive algorithm in Section IV. Numerical results appear in Section V. For brevity, proofs have been omitted; extensions are discussed in Baron and Weissman [23, 24].

II. Background

A. Notation

Consider a stationary ergodic source $X = \{X_i, i \geq 1\}$ over a continuous alphabet $\mathcal{X}$. Suppose that the distribution of X is given by some pdf $f(x)$. We process a length-n input block x^n. The input x^n is compressed using an *encoder* $e : \mathcal{X}^n \rightarrow \{0, 1\}^+$ that maps x^n to a finite output string $e(x^n)$. The *decoder* $d : \{0,1\}^+ \to \mathcal{Y}^n$ maps the bit string back to a length-n block y^n over the reproduction alphabet $\mathcal{Y}$, which could be continuous or discrete. The output y^n is an approximation of x^n.

We characterize the performance of an encoder-decoder pair using the trade-off between rate and distortion. The *rate* $R = E[|e(x^n)|]$ quantifies the expected number of bits needed to describe each input symbol, where $|\cdot|$ denotes length or cardinality, and $E[\cdot]$ is expectation. The *distortion* $D = E[d(x^n, y^n)]$ quantifies the error between the input x^n and decoder output y^n,

$$d_n(x^n, y^n) \triangleq \frac{1}{n}\sum_{i=1}^{n} d(x_i, y_i), \tag{1}$$

where $d : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ is a distortion metric. In both definitions for R and D, expectation is taken over the source pdf $f(x)$ for X. It is well-known [8, 9] that, given $f(x)$, there exists an optimal conditional pdf $q_{opt}(y|x)$ that achieves the RD function.

B. MCMC review

We describe a variant of the *Jalali-Weissman encoder* (JW) [20] for finite alphabets $\mathcal{Y}$, which we employ in our algorithms for continuous sources. The distinguishing features of our variant of JW are (*i*) instead of iterations that process individual locations of y^n we use super-iterations that process all n locations, and (*ii*) *context tree weighting* (CTW) [25] is used for lossless compression instead of Lempel-Ziv techniques [26].[1] The encoder approximates x^n by y^n, which is compressed using a universal lossless compressor – in our case CTW [25]. The approximation y^n is chosen to provide a trade-off between coding length and distortion w.r.t. x^n. Note that the JW decoding procedure is straightforward; simply decompress the output bits to retrieve y^n.

Let the empirical symbol counts $m_k(y^n, u^k)[a]$ be

$$m_k(y^n, u^k)[a] \triangleq |\{k < i \leq n : y_{i-k}^{i} = u^k a\}|,$$

where k is the context depth, $a \in \mathcal{Y}$, $u^k \in \mathcal{Y}^k$, and $u^k a$ denotes concatenation of u^k and a. Define the k-depth conditional empirical entropy as

$$H_k(y^n) \triangleq -\frac{1}{n}\sum_{a,u^k} m_k(y^n, u^k)[a] \log\left(\frac{m_k(y^n, u^k)[a]}{\sum_a m_k(y^n, u^k)[a]}\right), \tag{2}$$

where $\log(\cdot)$ is the base-two logarithm, and we employ the convention wherein $0\log(0) = 0$. Using a context depth $k = k_n = o(\log(n))$ guarantees that the CTW coding length asymptotically converges to the conditional empirical entropy [25]. The energy $\varepsilon(y^n)$ [21] corresponding to y^n is now defined as

$$\varepsilon(y^n) \triangleq n[H_k(y^n) - \beta d_n(x^n, y^n)], \tag{3}$$

where β is the slope of the RD function; the optimal $R - \beta D$ implies optimal RD performance [20, p.3] for that slope. The Boltzmann distribution is now defined,

$$f_s(y^n) \triangleq \frac{1}{Z_s}\exp\{-s\varepsilon(y^n)\}, \tag{4}$$

where Z_s is a normalization constant, which is inconsequential in the sequel.

Our goal is for the approximation $\widehat{x^n}$ to be close to the globally minimal energy solution (3),

$$\widehat{x^n} \triangleq \arg\min_{w^n \in \mathcal{Y}^n} \varepsilon(w^n) = \arg\min_{w^n \in \mathcal{Y}^n} [H_k(w^n) - \beta d_n(x^n, w^n)]. \tag{5}$$

[1] We prefer CTW [25], because for context tree sources it has lower redundancy than Lempel-Ziv techniques or full-tree Markov models.

During the minimization process, we refer to the approximation as y^n. Because CTW [25] asymptotically achieves the empirical entropy $H_k(y^n)$, we approximate the coding length $|CTW(y^n)|$ by $H_k(y^n)$. A formulation that would use the exact coding length (e.g., for CTW use $|CTW(y^n)|$) instead of $H_k(y^n)$ is left for future work.

To sample from the Boltzmann distribution (4), we iteratively examine all n locations, $i \in \{1, \dots, n\}$, and for each location use a Gibbs sampler to compute the distribution of y_i conditioned on $y^{n \backslash i} \triangleq \{y_n : n \neq i\}$,

$$f_s(y_i = a | y^{n\backslash i}) = \left(\sum_b \exp\left\{ -s \left[n\Delta H_k(y^{i-1} b y_{i+1}^n, a) - \beta \Delta d(b, a, x_i) \right] \right\} \right)^{-1}, \tag{6}$$

where $\Delta H_k(y^{i-1} b y_{i+1}^n, a)$ is the change in $H_k(y^n)$ (2) when $y_i = a$ is replaced by b, $\Delta d(b, a, x_i) = d(b, x_i) - d(a, x_i)$ is the change in distortion, and $s > 0$ is inversely related to temperature in simulated annealing [21]. We refer to the processing of a single location as an iteration, and group the n possible locations into super-iterations.[2]

During the simulated annealing, the inverse temperature s is gradually increased, where in super-iteration t we use $s = O(\log(t))$ [20, 21]. As s is increased, y^n converges in distribution to the set of minimal energy solutions, which includes $\widehat{x^n}$ (5), because large s favors low-energy y^n. As mentioned above, the JW decoder decompresses the output bits to retrieve y^n.

III. Universal algorithm with constant alphabet

We now discuss an algorithm inspired by JW that is designed specifically for continuous-valued $\mathcal{X}$. In addition to the distinguishing features of our variant of JW, as described in Section II-B, we must use (*i*) real valued reproduction levels, i.e., $\alpha \in \mathbb{R}$, $\forall \alpha \in \mathcal{Y}$, and (*ii*) an appropriate distortion metric for real-valued $\mathcal{X}$ and $\mathcal{Y}$.

Our focus is on square error distortion, which is of interest in many applications [1, 2], and we emphasize that $d_n(x^n, y^n)$ uses $d(x_i, y_i) = (x_i - y_i)^2$, in contrast to the Hamming distortion used by Jalali and Weissman [20]. The approximation y^n cannot do better than $\widehat{x^n}$, the minimal energy solution (5). Of course, even $\widehat{x^n}$ cannot do better than the RD function, no matter what alphabet $\mathcal{Y}$ is used [19, 27].

We now want to argue that the RD function can be achieved. Let us assume that the variance of X is finite, and consider the following reproduction alphabet,

$$\overline{\mathcal{Y}} \triangleq \left\{ -\frac{\gamma^2}{\gamma}, -\frac{\gamma^2 - 1}{\gamma}, \dots, \frac{\gamma^2}{\gamma} \right\},$$

where $\gamma = \lceil \log(n) \rceil$, and $\lceil \cdot \rceil$ denotes rounding up. Other choices of $\overline{\mathcal{Y}}$ would also enable us to demonstrate various RD results; the main point to keep in mind is that, as n is increased, $\overline{\mathcal{Y}}$ samples a wider interval with finer resolution.

To argue that this specific *constant reproduction alphabet* $\overline{\mathcal{Y}}$ combined with our variant of JW achieves the RD function asymptotically, we first show that a global optimization that determines $\widehat{x^n}$ followed by compression with CTW [25] achieves the RD function.

Theorem 1: Let X be any unknown finite variance stationary and ergodic source with RD function $R(X, D)$, consider the square error distortion measure, and let the reproduction alphabet $\overline{\mathcal{Y}}$ be used for

[2] We prefer the implementation where each super-iteration scans a permutation of all n locations of the input, because in this manner each location is scanned fairly often. However, other orderings of how the locations are processed, including a completely random order as prescribed by Jalali and Weissman [20], are also possible.

computing and compressing $\widehat{x^n}$. Then

$$\lim_{n\to\infty} \sup E\left[\frac{1}{n}|CTW(\widehat{x^n})| - \beta d(x^n, \widehat{x^n})\right] \leq \min_{D\geq 0}[R(X,D) - \beta D].$$

We omit the proof for brevity. It is noteworthy that our results could be modified to support other distortion metrics. For example, if we used ℓ_p distortion, then a technical condition $E[|X|^p] < \infty$ ensures that outliers do not increase the distortion by much. Interestingly, Yang et al. [19, 27] proved that a global optimization achieves the RD function; our contribution is to prove achievability using the specific alphabet $\overline{\mathcal{Y}}$. Combining Theorem 1 with the converse result of Yang et al. [19, 27],

$$E\left[\frac{1}{n}|CTW(\widehat{x^n})| - \beta d(x^n, \widehat{x^n})\right] \overset{n\to\infty}{\longrightarrow} \min_{D\geq 0}[R(X,D) - \beta D].$$

Now consider running our variant of JW instead of the global energy minimization (5) using the same alphabet $\overline{\mathcal{Y}}$ and square error distortion metric $d_n(\cdot,\cdot)$ as before. The simulated annealing [20, 21] converges in distribution to the set of minimal energy solutions. Therefore, our variant of JW is universal for continuous amplitude sources.

Theorem 2: Let X be any unknown finite variance stationary and ergodic source with RD function $R(X,D)$, consider the square error distortion measure, let the reproduction alphabet $\overline{\mathcal{Y}}$ be used for computing and compressing $\widehat{x^n}$, and let y_r^n be the approximation to x^n after r super-iterations. Then

$$\lim_{n\to\infty} \lim_{r\to\infty} E\left[\frac{1}{n}|CTW(y_r^n)| - \beta d(x^n, y_r^n)\right] \overset{n\to\infty}{\longrightarrow} \min_{D\geq 0}[R(X,D) - \beta D].$$

An important feature of the algorithm is that each iteration requires computation that is proportional to the context depth k and the alphabet size $|\overline{\mathcal{Y}}|$ [20]. Because the alphabet grows slowly in n, the per-iteration computational costs are modest. Each super-iteration contains n iterations, and we have noticed empirically that the algorithm often offers reasonable RD performance after only a few dozen super-iterations. The decoder is even faster; after decompressing the CTW [25], the finite alphabet is mapped to our constant reproduction alphabet $\overline{\mathcal{Y}}$.

Although promising, our variant of JW is of limited practical interest. In order to approach the RD function closely, $\overline{\mathcal{Y}}$ may need to be large. Not only does the reproduction alphabet become large only for large n, but the large $\overline{\mathcal{Y}}$ slows down the algorithm. One approach to improve the algorithm is to encode outlier source symbols, i.e., $|x_i| > \gamma$, explicitly using $\approx \log(|x_i|/\gamma)$ bits, perhaps using a universal code for integers [28]. This encoder would significantly reduce the distortion caused by outliers, thus allowing to use a narrower interval, yielding a reduction in the alphabet size. We leave the study of special outlier processing for future work, and focus instead on using an adaptive alphabet to improve the algorithm.

IV. Adaptive alphabet algorithm

Our approach to overcome the disadvantages of large alphabets (Section III) is inspired by the seminal work by Rose on the discrete nature of the reproduction alphabet when the Shannon lower bound is not tight [22]. Owing to the discrete reproduction alphabet, in many cases a smaller alphabet could offer reasonable RD performance. Indeed, in some cases a binary reproduction alphabet is optimal in the low rate limit [29]. We thus focus on an algorithm that, while supporting the possibility that the reproduction alphabet must be large, also supports a possible reduction of the alphabet size while allowing the actual reproduction levels to adapt to the input.

Following the approach by Yang and Zhang [19], we map the input x^n to a sequence z^n over a finite alphabet $\mathcal{Z}$, where the actual output y^n is derived via a scalar function $y_i = a(z_i)$. Ideally, the function

$a(\cdot)$ should minimize expected distortion. Because we focus on square error distortion, to find the optimal $a^*(\cdot)$ we compute the conditional expectation [19],

$$a^*(\alpha) = E[x_i | z_i = \alpha], \ \forall \alpha \in \mathcal{Z}. \tag{7}$$

We note in passing that $a^*(\alpha)$ can be interpreted as a form of universal quantization. Unfortunately, the decoder does not know x^n, and cannot compute $a^*(\cdot)$. Therefore, we must encode $a^*(\alpha)$ for each $\alpha \in \mathcal{Z}$. It can be shown that using $|\overline{\mathcal{Y}}| = O(\log^2(n))$ quantization levels suffices to keep the distortion manageable, and so we allocate $b\log(\log(n))$ bits to encode each

$$a_q^*(\alpha) \triangleq \frac{\lceil a^*(\alpha)\Delta \rceil}{\Delta}, \tag{8}$$

where $a_q^*(\alpha)$ is a quantized version of $a^*(\alpha)$, and Δ is a parameter that depends on b and the width of the interval being quantized. We observe that it might be advantageous to allocate more bits to encode $a_q^*(\alpha)$ for symbols $\alpha \in \mathcal{Z}$ that appear more times in z^n, but leave such optimizations for future work. Nonetheless, if some $\alpha \in \mathcal{Z}$ does not appear in z^n, then there is no need to encode it; we expend one flag bit for each $\alpha \in \mathcal{Z}$ to describe whether it is encoded or not. In summary, (3) is modified to support adaptive alphabets as follows,

$$\varepsilon_a(z^n) \triangleq n[H_k(z^n) - \beta d_a(x^n, z^n)] + b\log(\log(n))|\mathcal{Z}_a|, \tag{9}$$

where $\mathcal{Z}_a \subseteq \mathcal{Z}$ is the subset of $\mathcal{Z}$ that appears in z^n, $d_a(x^n, z^n)$ is distortion using the adaptive alphabet,

$$d_a(x^n, z^n) = d_n(x^n, a_q^*(z^n)), \tag{10}$$

and $a_q^*(\cdot)$ is computed using (7) and (8). Alternately, the optimization can loop over different alphabet sizes $|\mathcal{Z}|$ without accounting for $|\mathcal{Z}|$ in the energy (9). Indeed, we looped over several alphabet sizes in our simulations (Section V).

If a reduced alphabet yields similar distortion results without increasing the coding length, then the adaptive alphabet algorithm may choose not to use the entire alphabet $\mathcal{Z}$. Using a reduced alphabet enables us to spend fewer bits encoding $a^*(\cdot)$, and thus decreases the modified energy function (9). The results by Rose [22] (see also Marco and Neuhoff [29]) suggest that for many sources of practical interest a smaller alphabet could nonetheless offer good and in many case optimum RD performance. In such cases, the adaptive alphabet algorithm is advantageous.

Even if the entire alphabet is used, the location of the reproduction levels is optimized via $a^*(\cdot)$. Consequently, if we allow the adaptive alphabet algorithm to use $\mathcal{Z}$ with the same cardinality of $\overline{\mathcal{Y}}$ as in Section III, then the RD performance improves unless $\overline{\mathcal{Y}}$ offers near-optimal RD performance.

We now state formally that the adaptive alphabet algorithm achieves the RD function asymptotically without prior knowledge of the source statistics. Again, the proof is omitted for brevity.

Theorem 3: Let X be any unknown finite variance stationary and ergodic source with RD function $R(X, D)$, consider the square error distortion measure, let an adaptive reproduction alphabet $\mathcal{Z}$ with cardinality $|\mathcal{Z}| = |\overline{\mathcal{Y}}|$ be used for computing and compressing $\widehat{x^n}$, and let y_r^n be the approximation to x^n after r super-iterations. Then

$$\lim_{n\to\infty} \lim_{r\to\infty} E\left[\frac{1}{n}|CTW(y_r^n)| - \beta d_a(x^n, y_r^n)\right] \overset{n\to\infty}{\longrightarrow} \min_{D\geq 0}[R(X, D) - \beta D].$$

In the constant alphabet algorithm (Section III), the Gibbs sampler (6) must compute $\Delta H_k(y^{i-1}by_{i+1}^n, a)$ and $\Delta d(b, a, x_i)$. These values can both be updated rapidly [20]. The adaptive algorithm updates $\Delta H_k(z^{i-1}$

$bz_{i+1}^n, a)$ in an analogous manner. However, whereas $\Delta d(b,a,x_i) = d(b,x_i) - d(a,x_i)$ is trivial to compute in JW, in the adaptive algorithm case

$$\Delta d_a(b,a,z_i) \triangleq d_a(x^n, z^{i-1}bz_{i+1}^n) - d_a(x^n, z^{i-1}az_{i+1}^n),$$

and $d_a(\cdot,\cdot)$ depends on $a_q^*(\cdot)$. In other words, modifying a single location in z^n changes the distortion for numerous symbols. Therefore, in order to compute $\Delta d_a(b,a,z_i)$ efficiently, we must evaluate $d_a(x^n,z^n)$ in detail,

$$\begin{aligned} d_a(x^n,z^n) &= \frac{1}{n}\sum_{i=1}^{n} d(x_i, a_q^*(z_i)) = \frac{1}{n}\sum_{\alpha\in\mathcal{Z}}\sum_{\{i:\ z_i=\alpha\}}\left(x_i - a_q^*(\alpha)\right)^2 && (11)\\ &= \frac{1}{n}\sum_{\alpha\in\mathcal{Z}}\left\{\left[\sum_{\{i:\ z_i=\alpha\}}(x_i)^2\right] - 2a_q^*(\alpha)\left[\sum_{\{i:\ z_i=\alpha\}}x_i\right] + (a_q^*(\alpha))^2\left[\sum_{\{i:\ z_i=\alpha\}}1\right]\right\}, && (12)\end{aligned}$$

where (11) uses the definitions of $d_n(\cdot,\cdot)$ and $d_a(\cdot,\cdot)$ in (1) and (10), respectively, partitions z_i, $i\in\{1,\ldots,n\}$, into the different symbols $\alpha\in\mathcal{Z}$, and invokes the definition of square error distortion. Combining (7) and (8),

$$a_q^*(\alpha) = \frac{\lceil E[x_i|z_i=\alpha]\Delta\rfloor}{\Delta} = \frac{\left\lceil \frac{\sum_{\{i:\ z_i=\alpha\}}x_i}{\sum_{\{i:\ z_i=\alpha\}}1}\Delta\right\rfloor}{\Delta}. \quad (13)$$

We now see that (12) and (13) rely extensively on

$$X_\alpha^m \triangleq \sum_{\{i:\ z_i=\alpha\}}(x_i)^m, \quad m\in\{0,1,2\}, \quad (14)$$

where X_α^m denotes the m'th empirical moment of the portion of x where $z_i=\alpha$. In each iteration of the algorithm, a single z_i may change from α to α'. Consequently, we modify X_α^m and $X_{\alpha'}^m$, $m\in\{0,1,2\}$, by adding and subtracting powers of x_i. Given these updated values, the computation of $\Delta d_a(b,a,z_i)$ is straightforward, as in JW. Pseudo-code for the adaptive alphabet Algorithm 1 appears below.

By utilizing the computational techniques specified above, Algorithm 1 also requires computation that is proportional to the context depth k and alphabet size $|\mathcal{Y}|$. That said, in practice the effective alphabet $\mathcal{Z}_a$ is often smaller than $\mathcal{Z}$. Using a smaller alphabet, CTW [25] converges to the empirical entropy for larger context depths k. Therefore, Algorithm 1 can optimize over deeper context trees, leading to improved compression and faster convergence to the RD function.

As before, the decoder first decompresses the bit-stream generated by CTW to reconstruct $\widehat{z^n}$. The actual real-valued reproduction sequence is obtained by mapping from $\widehat{z^n}$ to $\widehat{y^n}$ via the adaptive quantizer $a_q^*(\alpha)$.

V. Numerical results

To demonstrate the potential of our algorithms, we implemented the adaptive alphabet Algorithm 1 in Matlab. We evaluated our implementation on a Laplace source with pdf $f(x) = \frac{1}{2}e^{-|x|}$ such that $E[X]=0$ and $\text{var}(X)=2$. The discrete nature of the reproduction alphabet [22] suggests that a small odd number of reproduction levels could capture the source well. Therefore, we evaluated alphabets of cardinality $|\mathcal{Z}|\in\{3,5,7\}$. At low rates, the smallest cardinality offers the best RD performance; as the rate is increased, larger alphabets quantize the source more precisely.

We ran Algorithm 1 for sequences of length $n=10^5$ using $r=100$ super-iterations, $k\approx\frac{1}{2}\log_{|\mathcal{Z}|}(n)$, and a temperature constant $c=1.6$. A comparison of our MCMC algorithm to entropy coding (ECSQ) and the RD function appears in Figure 1. Algorithm 1 consistently out-performs entropy coding, although our algorithm is universal and not aware of the source statistics. In contrast, ECSQ was specifically optimized for the Laplace source. Interestingly, the optimal mapping $a^*(\alpha)$ closely resembles the reproduction alphabet computed by the mapping approach of Rose [22], which suggests that applying JW [20] to the "correct" finite alphabet would not improve results by much, if at all. Although the margin of improvement over ECSQ is modest, we hope to improve performance in future work.

```
ALGORITHM 1: LOSSY ENCODER WITH ADAPTIVE REPRODUCTION ALPHABET
INPUT: $x^n \in \mathbb{R}^n$, $\mathcal{Z}$, $\beta$, $r$, $d_a(\cdot,\cdot)$
OUTPUT: bit-stream
PROCEDURE:
$z^n \leftarrow 0$ // initialize
for $t = 1$ to $r$ do // loop over blocks
    $s \leftarrow c\log(t)$ for some $c > 0$ [20, 21] // set inverse temperature for block
    Draw permutation of numbers $\{1, \ldots, n\}$ at random
    for $t' = 1$ to $n$ do // loop within block
        Let $i$ be component $t'$ in the permutation
        for all $\alpha$ in $\mathcal{Z}$ do // evaluate possible changes to $z_i$
            Compute $\Delta d_a(b, a, z_i)$ via (12), (13), and (14)
            Compute $f_s(z_i = \alpha | z^{n \backslash i})$ given in (6) // $\Delta d_a$ instead of $\Delta d$
        Generate $z_i$ using $f_s(\cdot | z^{n \backslash i})$ // Gibbs sampling
        Update $m_k(z^n, u^k)$ and $X^m_{z_i}$, $m \in \{0, 1, 2\}$ // previous and new $z_i$
Encode one flag bit for each $\alpha \in \mathcal{Z}$ to describe whether $a^*_q(\alpha)$ is encoded
Encode all relevant $a^*_q(\alpha)$ using $b\log(\log(n))$ bits each
Apply CTW [25] to $\widehat{z^n}$ // lossless compression of outcome
```

REFERENCES

[1] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Process.*, vol. 6, no. 5, pp. 677–693, 1997.

[2] S. M. Lopresto, K. Ramchandran, and M. T. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Proc. Data Compression Conf. (DCC)*, 1997, pp. 221–230.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A., and Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[4] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, no. 11, pp. 1551–1588, Nov. 1985.

[5] C. Gioran and I. Kontoyiannis, "Lossy compression in near-linear time via efficient random codebooks and databases," *CoRR*, vol. abs/0904.3340, 2009.

[6] A. Gupta, S. Verdu, and T. Weissman, "Rate-distortion in near-linear time," *preprint*, 2008.

[7] I. Kontoyiannis, "An implementable lossy version of the Lempel-Ziv algorithm - Part I: Optimality for memoryless sources," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2293–2305, Nov. 1999.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Wiley-Interscience, 1991.

[9] T. Berger, *Rate distortion theory; a mathematical basis for data compression.* Prentice-Hall Englewood Cliffs, NJ, 1971.

[10] R. Zamir and K. Rose, “Natural type selection in adaptive lossy compression,” *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 99–111, Jan. 2001.

[11] N. Hussami, S. B. Korada, and R. L. Urbanke, “Polar codes for channel and source coding,” *CoRR*, vol. abs/0901.2370, 2009.

[12] T. Linder and R. Zamir, “On the asymptotic tightness of the Shannon lower bound,” *IEEE Trans. Inf. Theory*, vol. 40, no. 6, pp. 2026–2031, Nov. 1994.

[13] A. György, T. Linder, and K. Zeger, “On the rate-distortion function of random vectors and stationary sources with mixed distributions,” *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2110–2115, Sep. 1999.

[14] H. Rosenthal and J. Binia, “On the epsilon entropy of mixed random variables,” *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 1110–1114, Sep. 1988.

[15] D. A. Huffman, “A method for the construction of minimum-redundancy codes,” *Proc. Inst. Radio Eng.*, vol. 9, no. 40, pp. 1098–1101, Sep. 1952.

[16] J. Rissanen and J. G. Langdon, “Universal modeling and coding,” *IEEE Trans. Inf. Theory*, vol. 27, no. 1, pp. 12–23, Jan. 1981.

[17] N. Farvardin and J. W. Modestino, “Optimum quantizer performance for a class of non-Gaussian memoryless sources,” *IEEE Trans. Inf. Theory*, vol. 30, no. 3, pp. 485–496, May 1984.

[18] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Kluwer, 1993.

[19] E. Yang and Z. Zhang, “Variable-rate trellis source encoding,” *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 586–608, Mar. 1999.

[20] S. Jalali and T. Weissman, “Rate-distortion via Markov chain Monte Carlo,” 2008, submitted.

[21] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, pp. 721–741, 1984.

[22] K. Rose, “A mapping approach to rate-distortion computation and analysis,” *IEEE Trans. Inf. Theory*, vol. 40, no. 6, pp. 1939–1952, Nov. 1994.

[23] D. Baron and T. Weissman, “Universal lossy compression of stationary ergodic continuous sources,” 2010, in preparation.

[24] ——, “Universal lossy compression of stationary ergodic continuous amplitude sources,” Nov. 2009, U.S. Provisional Patent Application No. 61/260,xxx.

[25] F. M. J. Willems, Y. Shtarkov, and T. J. Tjalkens, “The context tree weighting method: Basic properties,” *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[26] J. Ziv and A. Lempel, “Compression of individual sequences via variable-rate coding,” *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 530–536, Sep. 1978.

[27] E. Yang, Z. Zhang, and T. Berger, “Fixed-slope universal lossy data compression,” *IEEE Trans. Inf. Theory*, vol. 43, no. 5, pp. 1465–1476, Sep. 1997.

[28] P. Elias, “Universal codeword sets and representations of the integers,” *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 194–203, Mar. 1975.

[29] D. Marco and D. L. Neuhoff, “Low-resolution scalar quantization for Gaussian sources and squared error,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1689–1697, Apr. 2006.