An MCMC Approach to Universal Lossy Compression of Analog Sources

Dror Baron
Department of Electrical and Computer Engineering
North Carolina State University; Raleigh, NC
Email: barondror@ncsu.edu
Tsachy Weissman
Department of Electrical Engineering
Stanford University; Stanford, CA
Email: tsachy@stanford.edu

Abstract

Motivated by the Markov Chain Monte Carlo (MCMC) approach to the compression of discrete sources developed by Jalali and Weissman, we propose a lossy compression algorithm for analog sources that relies on a finite reproduction alphabet, which grows with the input length. The algorithm achieves, in an appropriate asymptotic sense, the optimum Shannon theoretic tradeoff between rate and distortion, universally for stationary ergodic continuous amplitude sources. We further propose an MCMC-based algorithm that resorts to a reduced reproduction alphabet when such reduction does not prevent achieving the Shannon limit. The latter algorithm is advantageous due to its reduced complexity and improved rates of convergence when employed on sources with a finite and small optimum reproduction alphabet.

I. Introduction

Lossy compression of analog sources is a pillar of modern communication systems. Despite numerous applications such as image compression [2, 3], video compression [4], and speech coding [5–7], there is a significant gap between theory and practice.

Parts of the research were performed when the first author was with the Electrical Engineering Department at the Technion, Israel. Subsets of the work appeared in [1].

A. *Entropy coding*

Many practical lossy compression algorithms employ *entropy coding*, where *scalar quantization* is followed by lossless compression (ECSQ). ECSQ has motivated much work into optimization of scalar quantizers [8–10], whereas the translation to bits can use Huffman [11] or arithmetic [12, 13] codes. Despite its simplicity and elegance, even for independent and identically distributed (iid) sources, ECSQ operates far from the *rate distortion* (RD) function [13, 14], the fundamental limit of lossy compression (cf. Figure 1 for an example). For non-iid sources, ECSQ may compare even less favorably with the fundamental RD limit.

In order to bridge the gap between ECSQ and the RD function, *vector quantization* (VQ) converts an entire vector to a codeword [7, 15, 16], in contrast to scalar quantization, which compresses individual scalar input elements. VQ provides a better trade-off between rate and distortion as the vector dimension increases, but increased complexity is required [17]. The significant computation required by VQ necessitates developing computationally feasible alternatives.

B. *Related work*

For *finite alphabet* sources, recent advances have demonstrated that the RD limit can be approached asymptotically [18–20] by *partitioning an input into sub-blocks*, where a Shannon-style random codebook [13, 14] is applied to each sub-block. Some of these schemes can compress *universally* without knowing the source statistics beforehand, but it is challenging to generate a codebook distribution whose statistics differ from those of the input statistics [21].

Lossy compression over a finite alphabet can also be performed by *directly mapping the entire input to an output sequence* while accounting for the trade-off between the compressibility of the output and the distortion between the input and output sequences. This optimization can be deterministic [22] or stochastic [23] in nature. Directly mapping to the output sequence effectively quantizes the entire input – a long sequence – into a large output codebook, and achieves the RD limit for stationary ergodic finite alphabet sources universally. Another promising recent approach to (non-universal) lossy compression relies on algebraic codes [24].

For *analog sources*, less progress has been made in developing theoretically-justified compression algorithms. Some results have been derived specifically for the high-rate regime, where the Shannon lower bound is asymptotically tight [25] under appropriate technical conditions imposed on the probability density function of the source and the distortion measure. In particular, in the limit of low distortions (high-rate) the RD limit has been characterized for mixtures of *probability distribution functions* (pdf's)

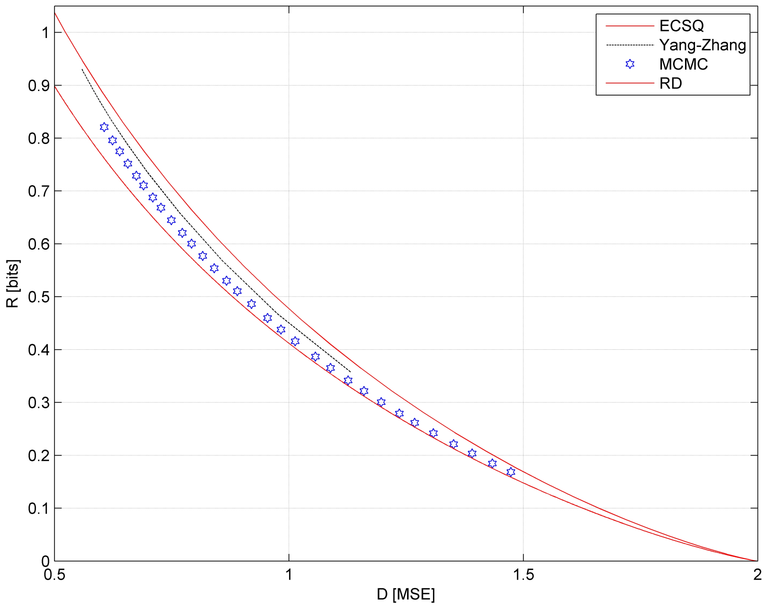

Fig. 1. **Laplace source**: *Comparison of entropy coding (ECSQ), results by Yang and Zhang [33], average rate and distortion of Algorithm 2 (MCMC) over 10 simulations, and the RD function.* ($n = 1.5 \cdot 10^4$, $|\mathcal{Z}| = 9$, $r = 50$, $k \approx \frac{1}{2}\log_{|\mathcal{Z}|}(n)$.)

where one distribution is discrete and the other continuous [26, 27]. For example, the sparse Gaussian source is a mixture pdf; bounds on its RD function have been provided [28–31].

Despite the theoretical insights in the high-rate regime, compression of analog sources at *low-to-medium rates* is of interest in many applications [2–5]. There do exist special input pdf's for which entropy coding approaches the RD function [32] in the low-rate limit, but the low-rate regime is challenging in general. We aspire to develop results of general applicability and not be limited to specific pdf's with fortuitous properties.

C. Contributions

The key point in our approach to fixed-to-variable length compression of analog sources is to quantize to discrete reproduction levels, and then apply a compression algorithm similar to that of Jalali and Weissman [23], which uses the stochastic optimization approach of *Markov chain Monte Carlo* (MCMC) simulated annealing, as pioneered in [34]. A careful choice of the set of reproduction levels, growing appropriately with input length both in size and in resolution, achieves the RD function despite the analog nature of the source. A somewhat similar approach was suggested by Yang et al. [22, 33] using deterministic optimization techniques. Note, however, that Yang and Zhang [33] require availability of a training sequence, and so their algorithm is not universal. Although it is possible to apply deterministic optimization in a universal setting by partitioning the input into blocks, Yang and Zhang mention that this approach results in a performance loss of 0.2–0.3 dB [33].

Our first contribution is a lossy compression algorithm for analog sources that relies on a *data-independent reproduction alphabet* that grows with the input length. This algorithm asymptotically achieves the RD function universally for stationary ergodic continuous amplitude sources. However, the reproduction alphabet grows with the input length, slowing down the convergence to the RD function, and is thus an impediment in practice.

To address this issue, we next propose an MCMC-based algorithm that uses an *adaptive reproduction alphabet*. The ground-breaking work by Rose on the discrete nature of the reproduction alphabet for iid sources when the Shannon lower bound is not tight [35] suggests that, for most sources of practical interest, restriction of the reconstruction to a rather small fixed-size alphabet does not stand in the way of attaining the fundamental compression limits. Indeed, at low rates even a binary reproduction alphabet is often optimal [32]. When employed on such sources, our latter algorithm zeroes in on the same finite reproduction alphabet, and thus enjoys rates of convergence commensurate with the finite-alphabet setting.

In order to render this adaptive algorithm computationally feasible, we develop a method to update the optimal reproduction levels rapidly. Utilizing this computational feature, our adaptive algorithm provides faster computation, achieves the RD function universally, and in some cases the smaller reproduction alphabet accelerates convergence to the RD function. Consequently, the adaptive algorithm is more suitable in practice. We emphasize that our algorithms are both universal, requiring no knowledge of the source statistics.

The remainder of the paper is organized as follows. We provide background information in Section II. Our first, brute force algorithm is described in Section III, followed by the adaptive reproduction alphabet algorithm in Section IV. Numerical results are reported in Section V. We complete the paper with a discussion in Section VI. Proofs appear in appendices, in order to make the main portion of the manuscript easily accessible.

II. Background

A. Notation and definitions

Consider a stationary ergodic real-valued source $X = \{X_i, i \geq 1\}$. The input to our algorithms is $x^n = x_1x_2 \ldots x_n$, which is an individual realization of the random vector X^n. The input x^n is compressed using an *encoder* $e : \mathcal{X}^n \to \{0,1\}^+$ 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 output y^n over the reproduction alphabet $\mathcal{Y}$, which may be a continuous or discrete subset of the real line. The output y^n is the lossy approximation of x^n.

We assess the performance of an encoder-decoder pair relative to the trade-off between rate and distortion [13, 14]. The *rate* of such a pair is defined as $R = E[\frac{1}{n}|e(X^n)|]$, the expected number of bits per description of a source symbol, where $|\cdot|$ denotes length, size, or cardinality, and $E[\cdot]$ is expectation. The *distortion* $D = E[d_n(X^n, y^n)]$ quantifies the expected per-symbol distortion,

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

where $d : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ measures the distortion. For concreteness in what follows, we assume the distortion is the square of the error $d(x_i, y_i) = (x_i - y_i)^2$, but our approach readily carries over to accommodate ℓ_p distortion measures.

We define $R(X, D)$, the RD function of a stationary ergodic source X, in an operational manner as follows. Let $C^i(X, D)$ be the smallest-cardinality codebook for input blocks of length i generated by X, such that the expected distortion between the input block x^i and the nearest codeword in $C^i(X, D)$ is at most D. The rate $R^i(X, D)$ is defined as the normalized log-cardinality of the codebook, $R^i(X, D) = \frac{1}{i}\log_2(|C^i(X, D)|)$. Finally, $R(X, D)$ is the limit of $R^i(X, D)$ over increasingly long blocks,

$$R(X, D) = \lim_{i\to\infty} \frac{1}{i}\log_2(|C^i(X, D)|). \tag{2}$$

B. Lossy compression using MCMC

We describe a variant of the scheme in [23] that compresses an input x^n to an output y^n over a finite alphabet $\mathcal{Y} \subseteq \mathbb{R}$, whose cardinality depends on n. This algorithm will later be employed as the main building block for compressing an analog source. The encoder approximates x^n by y^n, which is compressed using the *context tree weighting* (CTW) universal lossless compression algorithm.[1] The approximation y^n is chosen to provide a good trade-off between the coding length required for y^n and the distortion with respect to x^n. The decoding procedure is straightforward; the output bits are passed through the CTW decompressor to retrieve y^n.

[1] We prefer CTW [36], because for context tree sources it has lower expected redundancy than Lempel-Ziv based schemes [37] or adaptive arithmetic coding based on full-tree Markov models [33].

Denote the empirical symbol counts by $m_k(y^n, u^k)[a]$, i.e.,

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

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

$$H_k(y^n) \triangleq -\frac{1}{n} \sum_{a,u^k} m_k(y^n, u^k)[a] \log \left(\frac{m_k(y^n, u^k)[a]}{\sum_{a'} m_k(y^n, u^k)[a']} \right), \tag{3}$$

where $\log(\cdot)$ is the base-two logarithm, and we use the convention wherein $0 \log(0) = 0$. For $k = o(\log(n))$, the difference between the CTW coding length and the empirical conditional entropy is $o(1)$ uniformly over $y^n \in \mathcal{Y}^n$ [36]. (Seeing that the input x^n and output y^n might have different statistics, k must grow with n even if x^n is iid or is generated by a known-depth Markov source.) We define the energy $\varepsilon(y^n)$ corresponding to y^n by

$$\varepsilon(y^n) \triangleq n[H_k(y^n) - \beta d_n(x^n, y^n)], \tag{4}$$

where $\beta < 0$ is the slope of the RD function at the point we want to attain. The Boltzmann probability mass function (pmf) is

$$f_s(y^n) \triangleq \frac{1}{Z_s} \exp\{-s\varepsilon(y^n)\}, \tag{5}$$

where $s > 0$ is inversely related to temperature in simulated annealing [34], and Z_s is the normalization constant, which does not need to be computed.

Ideally, our goal is to compute the globally minimum energy solution $\widehat{x^n}$,

$$\widehat{x^n} \triangleq \arg\min_{w^n \in \mathcal{Y}^n} \varepsilon(w^n) = \arg\min_{w^n \in \mathcal{Y}^n} [H_k(w^n) - \beta d_n(x^n, w^n)]. \tag{6}$$

Computation of $\widehat{x^n}$ involves an exhaustive search over exponentially many sequences and is thus infeasible. We use the stochastic optimization approach of *Markov chain Monte Carlo* (MCMC) simulated annealing [34] to approximate the globally minimum solution, in contrast to the deterministic approach of Yang et al. [22]. We denote the resulting approximation by y^n.

Because it is difficult to sample from the Boltzmann pmf (5) directly, we instead use a Gibbs sampler, which computes the marginal distributions at all n locations conditioned on the rest of y^n being kept fixed. For each location, the Gibbs sampler resamples from the distribution of y_i conditioned on $y^{n \backslash i} \triangleq \{y_n : \ n \neq i\}$ as induced by the joint pmf in (5), readily computed to be

$$f_s(y_i = a | y^{n \backslash i}) = \frac{1}{\sum_b \exp\left\{-s \left[n \Delta H_k(y^{i-1} b y_{i+1}^n, a) - \beta \Delta d(b, a, x_i)\right]\right\}}, \tag{7}$$

where $\Delta H_k(y^{i-1}by_{i+1}^n, a)$ is the change in $H_k(y^n)$ (3) when $y_i = a$ is replaced by b, and $\Delta d(b, a, x_i) = d(b, x_i) - d(a, x_i) = (b - x_i)^2 - (a - x_i)^2$ is the change in distortion. We refer to the resampling from 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 $s = O(\log(t))$ [23, 34]. In each iteration, the Gibbs sampler modifies y^n in a random manner that resembles heat bath concepts in statistical physics. Although MCMC could sink into a local minimum, we decrease the temperature slowly enough that the randomness of Gibbs sampling eventually drives MCMC out of the local minimum toward the set of minimal energy solutions, which includes $\widehat{x^n}$ (6), because large s favors low-energy y^n. Pseudo-code for our encoder appears in Algorithm 1 below.

```
ALGORITHM 1: LOSSY ENCODER WITH FIXED REPRODUCTION ALPHABET

INPUT: $x^n \in \mathbb{R}^n$, $\mathcal{Y}$, $\beta$, $c$, $r$

OUTPUT: bit-stream

PROCEDURE:
Initialize $y$ by quantizing $x$ with $\mathcal{Y}$
Initialize $m_k(\cdot,\cdot)$ using $y$
for $t = 1$ to $r$ do // super-iteration
    $s \leftarrow c\log(t)$ for some $c > 0$ // inverse temperature
    Draw permutation of numbers $\{1, \ldots, n\}$ at random
    for $t' = 1$ to $n$ do
        Let $i$ be component $t'$ in permutation
        Generate new $y_i$ using $f_s(y_i = \cdot | y^{n\backslash i})$ given in (7) // Gibbs sampling
        Update $m_k(\cdot,\cdot)[\cdot]$
Apply CTW to $y^n$ // compress outcome
```

III. UNIVERSAL ALGORITHM WITH DATA-INDEPENDENT REPRODUCTION ALPHABET

Let us consider how Algorithm 1 can be used to compress analog sources. We will see that choosing the reproduction alphabet $\mathcal{Y}$ to be a finite subset of $\mathbb{R}$ (but growing with the input length n in a data-independent way) achieves the RD function.

Let us assume that the variance of source symbols emitted by X is finite, and consider the following data-independent reproduction alphabet,

$$\overline{\mathcal{Y}} \triangleq \left\{ -\frac{\gamma^2}{\gamma}, -\frac{\gamma^2 - 1}{\gamma}, \ldots, \frac{\gamma^2}{\gamma} \right\}, \quad \gamma = \lceil \log(n) \rceil, \tag{8}$$

where $\lceil\cdot\rceil$ denotes rounding up. In words, $\overline{\mathcal{Y}}$ is a quantization of the interval $[-\gamma, \gamma]$ to resolution $1/\gamma$. Other choices of $\overline{\mathcal{Y}}$ also allow to demonstrate various RD results; an examination of (25) indicates that

[2] We recommend an ordering where each super-iteration scans a permutation of all n locations of the input, because in this manner each location is scanned fairly often. Other orderings are possible, including a completely random order as prescribed by Jalali and Weissman [23].

slower-growing $\gamma(n)$ also achieves the RD function. The essential point is that $\overline{\mathcal{Y}}$ quantizes a wider interval with finer resolution as n is increased, and $|\overline{\mathcal{Y}}|$ increases sufficiently slowly with n.

To prove achievability of the RD function asymptotically, we first prove that a global optimization (6) that determines $\widehat{x^n}$ followed by lossless compression with CTW [36] achieves the RD function. Yang et al. [22, 33] proved a similar result for their deterministic algorithm while relying on a different reproduction alphabet; our contribution is to prove achievability using the data-independent reproduction alphabet $\overline{\mathcal{Y}}$.

Theorem 1: Consider square error distortion (1), let X be a finite variance stationary and ergodic source with RD function $R(X,D)$ (2), and use the data-independent reproduction alphabet $\overline{\mathcal{Y}}$ (8) to approximate x^n by the globally minimum energy solution $\widehat{x^n}$ (6). Then the length of context tree weighting (CTW) [36] applied to $\widehat{x^n}$ converges as follows,

$$\limsup_{n\to\infty} E\left[\frac{1}{n}|CTW(\widehat{x^n})| - \beta d_n(x^n,\widehat{x^n})\right] \le \min_{D\ge 0}[R(X,D)-\beta D]. \tag{9}$$

Note that the $\limsup$ in (9) is actually a limit since the expectation on the left hand side is lower bounded by the right hand side for any scheme and any n, cf., e.g., [22, 23]. The detailed proof appears in Appendix A, and we feature some highlights here. In order to prove achievability for the continuous alphabet source X, we construct a near-optimal codebook for a given input length n [14], and then quantize the components of every codeword in the codebook to $\overline{\mathcal{Y}}$. As n is increased, $\overline{\mathcal{Y}}$ quantizes a wider interval of values more finely. The wider interval ensures that outlier source symbols have a vanishing effect on the distortion, and finer quantization provides near-optimal distortion within the interval. Therefore, we have achievability for the continuous amplitude source X via the finite alphabet $\overline{\mathcal{Y}}$.

What is the RD performance of $\widehat{x^n}$? Keeping in mind that the resolution of the quantizer $\overline{\mathcal{Y}}$ is $1/\gamma$, for a rate $R=\frac{1}{n}CTW(\widehat{x^n})$ we expect the distortion between x^n and $\widehat{x^n}$ (1) to obey

$$d_n(x^n,\widehat{x^n}) \ge D(R)+O(\gamma^{-2}) = D(R)+O(\log(n)^{-2}).$$

(We could decrease the excess distortion $O(\gamma^{-2})$ by choosing a larger data-independent reproduction alphabet. But this approach can only go so far to improve RD performance, because the redundancy or excess coding length above the entropy rate of CTW is $O(\log(n)/n)$ [36], and even in a non-universal setting the RD performance approaches the RD function gradually as n is increased [38].) In terms of rate, for a distortion $D=d_n(x^n,\widehat{x^n})$ the slope β of the RD function yields

$$\frac{1}{n}CTW(\widehat{x^n}) \ge R(D) - \beta O(\gamma^{-2}).$$

Now consider running Algorithm 1 instead of the global energy minimization (6) using the data-independent reproduction alphabet $\overline{\mathcal{Y}}$. The constant c used in Line 4 of Algorithm 1 plays a crucial role. If c is large, then the Boltzmann pmf (5) favors low-energy sequences too greedily, and the algorithm might get stuck in local minima. On the other hand, there exists a universal constant c_1 that does not depend on n such that for $c < c_1$ we obtain universal performance. To understand why this happens, observe that Algorithm 1 optimizes over $|\overline{\mathcal{Y}}|^n$ possible outputs. As long as $c < c_1$, there is a sufficiently large probability to transition between any two outputs, and the algorithm cannot get bogged down in a local mimimum. Therefore, in the limit of many iterations Algorithm 1 converges in distribution to the set of minimal energy solutions, and we enjoy the same RD performance as in Theorem 1. We refer the reader to Geman and Geman [34] for further discussions relating to the choice of c_1. The proof appears in Appendix B.

Theorem 2: Consider square error distortion (1), let X be a finite variance stationary and ergodic source with RD function $R(X, D)$ (2), and use Algorithm 1 with the data-independent reproduction alphabet $\overline{\mathcal{Y}}$ (8) and sufficiently small $c < c_1$. Let y_r^n be the MCMC approximation to x^n after r super-iterations. Then the length of context tree weighting (CTW) [36] applied to y_r^n converges as follows,

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

Theorem 2 is an information theoretic result saying that for sufficiently large block length n and number of super-iterations r, we come arbitrarily close to achieving the fundamental compression limits of the source. In practice, one can only take as many iterations as the computational power affords.

It is also important to note that Theorems 1 and 2 are stated in terms of achieving the $(R(\beta), D(\beta))$ pair at a certain slope β that we want to attain. However, in practice the user will often be interested in achieving a prescribed rate R or distortion D. In this case, it is possible to compute the requisite $D(R)$ or $R(D)$ using a simple line search algorithm [39], and there is no need to compute the entire $(R(\beta), D(\beta))$ curve.

An important feature of the algorithm is that each iteration of Lines 7–9 requires computation that is proportional to the context depth k and alphabet size $|\overline{\mathcal{Y}}|$ [23]. Because the alphabet grows slowly in n, the per-iteration computational costs are modest. Each super-iteration contains n iterations, and so its computation is $O(nk|\overline{\mathcal{Y}}|) = o(n \log^3(n))$. Decoding is also fast. We first decompress CTW [36], and the finite alphabet is then mapped to our data-independent reproduction alphabet $\overline{\mathcal{Y}}$.

It is also 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 outlier values in x^n with

$|x_i| > \gamma$ do not increase the distortion by much. The results also apply for some other distortion metrics but are not fully general, because of the possible presence of outliers in the data.

Although promising from a theoretical perspective, Algorithm 1 is of limited practical interest. In order to approach the RD function closely, $\overline{\mathcal{Y}}$ may need to be large, which slows down the algorithm. We focus on using an adaptive reproduction alphabet to improve the algorithm.

IV. Adaptive reproduction alphabet algorithm

Our approach to overcome the disadvantages of large alphabets (Section III) is inspired by the groundbreaking work by Rose on the discrete nature of the reproduction alphabet for iid sources when the Shannon lower bound is not tight [35]. In many cases of interest, a small reproduction alphabet achieves the RD function of an analog source. Indeed, at sufficiently low rates even a binary reproduction alphabet is sometimes optimal [32]. 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. The possibility of having a large alphabet is conducive to our theoretical statements, but we show numerically in Section V that in practice a small adaptive alphabet will suffice at low rates.

A. Adaptive reproduction levels

Following the approach of Yang and Zhang [33], 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, the optimal $a^*(\cdot)$ is the conditional expectation [33],

$$a^*(\alpha) = E[x_i|z_i = \alpha] = \frac{\sum_{i:z_i=\alpha} x_i}{\sum_{i:z_i=\alpha} 1},\ \forall\alpha \in \mathcal{Z}. \tag{10}$$

Note that $a^*(\alpha)$ can be computed universally without knowing the input source X.

The encoder knows x^n and can compute $a^*(\cdot)$, but the decoder does not have access to x^n. Therefore, the encoder describes a quantized version of $a^*(\alpha)$ to the decoder for each $\alpha \in \mathcal{Z}$,

$$a_q^*(\alpha) \triangleq \frac{\lceil a^*(\alpha)\Delta \rfloor}{\Delta}. \tag{11}$$

Theorem 1 indicates that the data-independent reproduction alphabet $\overline{\mathcal{Y}}$ (8) quantizes a sufficiently wide interval $[-\gamma, \gamma]$ with sufficiently fine resolution $1/\gamma$, and so a quantizer resolution $\frac{1}{\Delta} \approx \gamma$ need only select from $2\gamma^2 + 1 = 2\lceil \log(n) \rceil^2 + 1$ levels. Therefore, each such quantization level can be encoded using $\approx \log((\log(n))^2)+1$ bits, and it suffices to allocate $\mu\log(\log(n))$ bits, where $\mu > 2$. 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 its numerical value. We expend one flag bit per character of $\mathcal{Z}$ to describe the *effective alphabet* $\mathcal{Z}_e = \mathcal{Z}_e(z^n)$, where $\mathcal{Z}_e \subseteq \mathcal{Z}$ is the subset of the reproduction alphabet $\mathcal{Z}$ that appears in z^n. Because $|\mathcal{Z}| = |\overline{\mathcal{Y}}| = 2\lceil\log(n)\rceil^2+1$, only $O(\log^2(n))$ flag bits are needed. In fact, because z^n can be described using any $|\mathcal{Z}_e|$ symbols out of $|\overline{\mathcal{Y}}|$, it suffices for the encoder to describe the cardinality of $\mathcal{Z}_e$ using $O(\log(\log(n)))$ bits, which is insignificant.

The energy function (4) must be modified to support adaptive alphabets as follows,

$$\varepsilon_a(z^n) \triangleq n[H_k(z^n) - \beta d_a(x^n, z^n)] + \mu\log(\log(n))|\mathcal{Z}_e(z^n)|, \tag{12}$$

where $\mu\log(\log(n))|\mathcal{Z}_e|$ bits are used to encode the reproduction levels that appear in the effective alphabet $\mathcal{Z}_e$, $d_a(x^n, z^n)$ is distortion with the adaptive alphabet,

$$d_a(x^n, z^n) = d_n(x^n, a_q^*(z^n)) = \frac{1}{n}\sum_{i=1}^{n}(x_i - a_q^*(z_i))^2, \tag{13}$$

$a_q^*(z^n)$ is shorthand for the n-tuple obtained by applying a_q^* to the components of z^n, and $a_q^*(\cdot)$ is computed using (10) and (11). These definitions require to modify the previous Gibbs sampler (7) as follows,

$$\begin{aligned} & f_s(z_i = a|z^{n\backslash i}) \\ = \; & \frac{1}{\sum_b \exp\left\{-s\left[n\Delta H_k(z^{i-1}bz_{i+1}^n, a) - \beta\Delta d_a(b,a,z^n) + \mu\log(\log(n))\Delta\mathcal{Z}_e(b,a)\right]\right\}}, \end{aligned} \tag{14}$$

where

$$\Delta d_a(b,a,z^n) \triangleq n\left[d_a(x^n, z^{i-1}bz_{i+1}^n) - d_a(x^n, z^{i-1}az_{i+1}^n)\right] \tag{15}$$

is the change in distortion using the adaptive alphabet (13), and $\Delta\mathcal{Z}_e(b,a)$ is the change in the size of the effective alphabet when $z_i = a$ is replaced by b. Alternately, the optimization routine can loop over different alphabet sizes $|\mathcal{Z}|$ without accounting for $|\mathcal{Z}|$ in the energy (12); this latter approach was used in our simulations (Section V).

The key point is that if a reduced alphabet yields similar distortion results without increasing the coding length, then the modified energy function (12) induces a smaller effective $\mathcal{Z}_e$. Motivated by the theoretical results by Rose [35] and our numerical results (Section V), for many analog sources of practical interest a small alphabet offers good and in some cases optimum RD performance. In such cases, the adaptive alphabet algorithm is advantageous.

Even if the entire alphabet is used, i.e., $\mathcal{Z}_e = \mathcal{Z}$, then the location of the reproduction levels is optimized via $a^*(\cdot)$ in lieu of the uniform quantization used in $\overline{\mathcal{Y}}$ (8). Consequently, if we allow the adaptive alphabet algorithm to use $\mathcal{Z}$ with the same cardinality of $\overline{\mathcal{Y}}$ as in Algorithm 1, then the RD performance can only improve.

We now state formally that the adaptive alphabet algorithm achieves the RD function asymptotically without prior knowledge of the source statistics. As before, our result relies on the existence of a universal constant c_2 such that for $c < c_2$ the transition probabilities between the $|\mathcal{Z}|^n$ possible outputs are sufficiently large.

Theorem 3: Consider square error distortion (1), let X be a finite variance stationary and ergodic source with RD function $R(X,D)$ (2), use $\mathcal{Z}$ with cardinality $|\mathcal{Z}| = 2\lceil \log(n) \rceil^2 + 1$ and sufficiently small $c < c_2$ in Algorithm 2, and let $a_q^*(z_r^n)$ be the MCMC approximation to x^n after r super-iterations. Then the length of context tree weighting (CTW) [36] applied to z_r^n converges as follows,

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

The formal proof appears in Appendix C. The key point is that adaptive reproduction levels offer pointwise improvement over the data-independent reproduction alphabet $\overline{\mathcal{Y}}$ from Section III, per the same alphabet size.

B. Fast computation

An important contribution by Jalali and Weissman [23] was to show how to compute $\Delta H_k(y^{i-1} b y_{i+1}^n, a)$ and $\Delta d(b,a,x_i)$ rapidly. Without this computational contribution, the encoder would be impractical. The adaptive algorithm updates $\Delta H_k(z^{i-1} b z_{i+1}^n, a)$ in an analogous manner. However, whereas $\Delta d(b,a,x_i) = (b - x_i)^2 - (a - x_i)^2$ is trivial to compute for the data-independent reproduction alphabet $\overline{\mathcal{Y}}$ (8), in our case $\Delta d_a(b,a,z^n)$ (15) requires to re-compute $d_a(\cdot,\cdot)$, which depends on $a_q^*(\cdot)$. Unfortunately, modifying a single location in z^n may change the distortion for numerous symbols.

We now show how to compute $\Delta d_a(b,a,z^n)$ rapidly for the adaptive reproduction alphabet algorithm. To do so, we evaluate $d_a(x^n, z^n)$,

$$\begin{aligned}
d_a(x^n, z^n) &= \frac{1}{n}\sum_{i=1}^{n} d(x_i, a_q^*(z_i)) && (16)\\
&= \frac{1}{n}\sum_{\alpha\in\mathcal{Z}} \sum_{\{i:\ z_i=\alpha\}} \left(x_i - a_q^*(\alpha)\right)^2 && (17)\\
&= \frac{1}{n}\sum_{\alpha\in\mathcal{Z}} \left\{ \sum_{\{i:\ z_i=\alpha\}} (x_i)^2 - 2a_q^*(\alpha)x_i + (a_q^*(\alpha))^2 \right\}, && (18)
\end{aligned}$$

where (16) uses the definitions of $d_n(\cdot,\cdot)$ and $d_a(\cdot,\cdot)$ in (1) and (13), respectively, and (17) 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 (10) and (11),

$$a_q^*(\alpha) = \frac{\lceil E[x_i|z_i=\alpha]\Delta\rceil}{\Delta} = \frac{\left\lceil \frac{\sum_{\{i:\ z_i=\alpha\}} x_i}{\sum_{\{i:\ z_i=\alpha\}} 1}\Delta \right\rceil}{\Delta}. \tag{19}$$

We see that (18) and (19) rely extensively on

$$X_\alpha^m \triangleq \sum_{\{i:\ z_i=\alpha\}} (x_i)^m, \quad m \in \{0,1,2\}, \alpha \in \mathcal{Z}, \tag{20}$$

the m'th moments of the portion of x where $z_i = \alpha$. We now have

$$a_q^*(\alpha) = \frac{\left\lceil \frac{X_\alpha^1}{X_\alpha^0}\Delta \right\rceil}{\Delta}, \tag{21}$$

$$d_a(x^n, z^n) = \frac{1}{n}\sum_{\alpha\in\mathcal{Z}} \left\{ X_\alpha^2 - 2a_q^*(\alpha)X_\alpha^1 + (a_q^*(\alpha))^2 X_\alpha^0 \right\}. \tag{22}$$

With these definitions in place, the update of $\Delta d_a(b,a,z^n)$ in each iteration becomes rapid. During an iteration, symbol $z_i = \alpha$ changes to $\overline{z}_i = \overline{\alpha}$. We subtract $(x_i)^m$ from X_α^m and add $(x_i)^m$ to $X_{\alpha'}^m$, i.e.,

$$\overline{X}_\alpha^m = X_\alpha^m - (x_i)^m \quad \text{and} \quad \overline{X}_{\alpha'}^m = X_{\alpha'}^m + (x_i)^m, \quad m \in \{0,1,2\}.$$

Given these updated values of $\overline{X}_\alpha^m$ and $\overline{X}_{\alpha'}^m$, computation of $\Delta d_a(b,a,z^n) = d_a(b,a,\overline{z^n}) - d_a(b,a,z^n)$ per (21) and (22) requires constant time per iteration. Pseudo-code for the adaptive alphabet Algorithm 2 appears below.

```
ALGORITHM 2: LOSSY ENCODER WITH ADAPTIVE REPRODUCTION ALPHABET

INPUT: $x^n \in \mathbb{R}^n$, $\mathcal{Z}$, $\beta$, $c$, $r$, $\mu$

OUTPUT: bit-stream

PROCEDURE:
Initialize $z$ by quantizing $x$ // can quantize with data-independent $\overline{\mathcal{Y}}$
Initialize $m_k(\cdot,\cdot)$ and other data structures using $z$
for $t = 1$ to $r$ do super-iteration
    $s \leftarrow c\log(t)$ for some $c > 0$ // inverse temperature
    Draw permutation of numbers $\{1,\ldots,n\}$ at random
    for $t' = 1$ to $n$ do
        Let $i$ be component $t'$ in permutation
        for all $\alpha$ in $\mathcal{Z}$ do // evaluate possible changes to $z_i$
            Compute $\Delta d_a(b,a,z^n)$ via (15), (20), (21), and (22)
            Compute $f_s(z_i = \alpha|z^{n\backslash i})$ given in (14) // modified Gibbs distribution
        Generate new $z_i$ using $f_s(z_i = \cdot|z^{n\backslash i})$ // Gibbs sampling
        Update $m_k(\cdot,\cdot)[\cdot]$ and $X_{z_i}^m$, $m \in \{0,1,2\}$ // previous and new $z_i$
Encode effective alphabet $\mathcal{Z}_e$
Encode $a_q^*(\alpha)$ using $\mu\log(\log(n))|\mathcal{Z}_e|$ bits
Apply CTW to $z^n$
```

As for the data-independent reproduction alphabet case, Algorithm 2 requires $O(nk|\mathcal{Z}|)$ time to compute $\Delta H_k(z^{i-1}bz_{i+1}^n, a)$. Utilizing the computational techniques specified above, $\Delta d_a(b, a, z^n)$ can be computed in constant time per inner loop of each iteration (Line 9), which requires $O(n|\mathcal{Z}|) = O(n|\overline{\mathcal{Y}}|)$ computation per super-iteration. We see that computing $\Delta H_k(z^{i-1}bz_{i+1}^n, a)$ should require more time than computing $\Delta d_a(b, a, z^n)$; this was confirmed in our implementation.

We have also noticed empirically that Algorithm 2 often comes quite close to optimum RD performance after a few dozen super-iterations, resulting in reasonable overall computational demands. Additionally, in practice the effective alphabet $\mathcal{Z}_e$ is often modest. CTW [36] converges to the empirical entropy as long as $k_n = \log(n)/\log(|\mathcal{Z}_e|) - \Omega_n(1)$, where the $\Omega_n(1)$ term decays to 0 as n is increased, and for finite n a smaller alphabet $|\mathcal{Z}_e|$ allows CTW to converge to the empirical entropy for larger context depths k_n. Therefore, Algorithm 2 can optimize over deeper context trees, leading to improved compression and faster convergence to the RD function.

The decoder of the adaptive reproduction alphabet Algorithm 2 resembles the decoder in [23]. First, the bit-stream generated by CTW is decompressed to reconstruct z^n. The actual real-valued reproduction sequence is obtained by mapping from z^n to y^n via the adaptive quantizer $a_q^*(\alpha)$, since the mapping a_q^* has been described to the decoder.

V. Numerical results

To demonstrate the potential of our approach, we implemented the adaptive alphabet Algorithm 2 in Matlab; our code is available for download at `http://people.engr.ncsu.edu/dzbaron/software/RD_BaronWeissman/`. Results for Laplace and autoregressive sources are provided.

Implementation details: We ran Algorithm 2 for sequences of length $n = 1.5 \cdot 10^4$ using $r = 50$ super-iterations, and $k \approx \frac{1}{2}\log_{|\mathcal{Z}|}(n)$. We found two heuristics to be useful. First, for each individual compression problem and RD slope β the specific temperature evolution $s = O(\log(t))$ may vary. Therefore, for each point we ran four temperature evolution sequences and allowed each one to improve over the energy $\varepsilon_a(z^n)$ computed with previous evolution sequences. Second, using a good starting point helps Algorithm 2 converge. Therefore, we began running low rate problems with small β, and each solution was used as a starting point for the next larger β.

Below we plot results averaging over 10 simulations. Each plot compares the MCMC approach over a range of β values to entropy coding (ECSQ), results by Yang and Zhang [33], and the RD function. We note in passing that the RD function of the Laplace source was computed numerically using the mapping approach of Rose [35], and the RD function of the autoregressive source was computed analytically via water pouring [40].

Laplace source: We first evaluated an iid Laplace source with pdf $f(x) = \frac{1}{2}e^{-|x|}$ such that $E[X] = 0$ and var$(X) = 2$. For this source, entropy coding performs rather well. However, Yang and Zhang [33] give better RD performance (Figure 1). Algorithm 2 improves further over the deterministic minimization by Yang and Zhang [33], which requires availability of a training sequence. Although their algorithm can be used in a universal setting by partitioning the input into blocks, Yang and Zhang mention that this approach results in a performance loss of 0.2–0.3 dB [33].

Relying on the mapping approach of Rose [35], it can be shown that for low-to-medium rates a small odd number of reproduction levels suffices to approach the fundamental RD limit of the Laplace source. This approach maps the unit interval with Lebesgue measure to the reproduction space, and the codebook optimization is over the mapping. When the Shannon lower bound is not tight, the mapping approach boils down to an annealing process that tracks the location of the reproduction alphabet at different β. In contrast, the Blahut-Arimoto algorithm [41, 42] optimizes the output distribution, and it approaches the correct reproduction distribution only in the limit of high resolution. We have observed numerically that the optimal mapping $a^*(\alpha)$ is similar to the reproduction alphabet computed by the mapping approach of Rose [35]. This similarity suggests that applying Algorithm 1 to the "correct" finite alphabet would not improve results by much.

Autoregressive source: Figure 2 illustrates the RD performance of the different algorithms for an *autoregressive* (AR) source, where

$$x_n = \rho x_{n-1} + w_n,$$

$\rho = 0.9$, and the innovation sequence $w_n \sim \mathcal{N}(0, 1)$ is zero mean unit norm iid Gaussian.

Entropy coding (ECSQ) is not well suited for non-iid sources; vector quantization [17], the deterministic minimization algorithm by Yang and Zhang [33], and MCMC can be used instead. Note that ECSQ appears in the upper right hand side of the figure; its RD performance is poor.

We plotted the RD performance of Algorithm 2 using a small reproduction alphabet ($|\mathcal{Z}| = 3$) and a moderately sized one ($|\mathcal{Z}| = 9$). At low rates, the smaller alphabet offers better RD performance; as the rate is increased, larger alphabets quantize the source more precisely. Although the compression results of Yang and Zhang are better for the AR source than those of Algorithm 2, the algorithm of Yang and Zhang is not universal.

It is interesting to note that the results of Figure 2 may seem a bit messy, in particular for $|\mathcal{Z}| = 9$ the results are not monotonic. This is explained by realizing that our second heuristic chooses as a starting

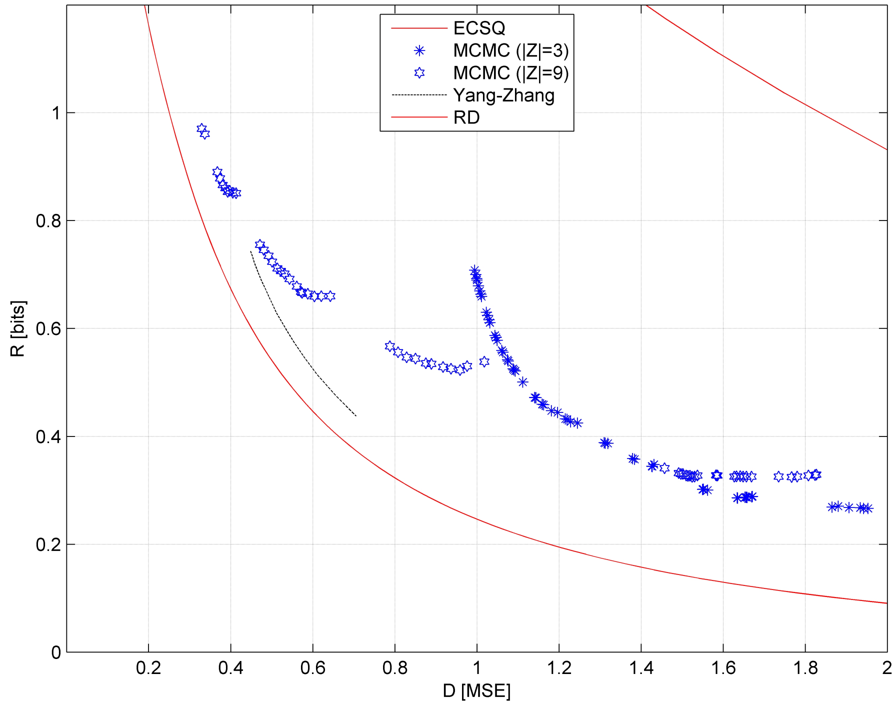

Fig. 2. **AR source**: *Comparison of entropy coding (ECSQ), average rate and distortion of Algorithm 2 (MCMC) over 10 simulations, results by Yang and Zhang [33], and the RD function.* ($n = 1.5 \cdot 10^4$, $|\mathcal{Z}| \in \{3, 9\}$, $r = 50$, $k \approx \frac{1}{2} \log_{|\mathcal{Z}|}(n)$.)

point for each RD computation the previous RD point, and therefore while increasing β (decreasing the distortion D) it is possible for both R and D to improve. In some cases the algorithm can improve significantly over the heuristic starting point, yielding big reductions in distortion between adjacent points (see the gap in D between 0.66 and 0.78) at the expense of extra rate.

VI. Discussion

In this paper, we extended the MCMC simulated annealing approach of Jalali and Weissman [23] to fixed-to-variable length compression of analog sources. We described two lossy compression algorithms that asymptotically achieve the RD function universally for stationary ergodic continuous amplitude sources. The first algorithm relies on a data-independent reproduction alphabet that samples a wider interval with finer resolution as the input length is increased. However, the large alphabet slows down the convergence to the RD function, and is an impediment in practice. Our second algorithm therefore uses a (potentially smaller) adaptive reproduction alphabet. Not only is the adaptive algorithm theoretically motivated for iid sources by the discrete nature of the reproduction alphabet when the Shannon lower bound is not tight [35], but our numerical results suggest that even for non-iid sources it works well using a small alphabet. Additionally, the smaller alphabet accelerates the computation.

We opened the paper by mentioning that there is a significant gap between theory and practice, and close the paper by asking where this paper is located on the theory–practice spectrum. The message of the paper is that there is hope to go beyond entropy coding in a completely universal way and thus bypass the vector quantizer paradigm, which requires training data. As mentioned after Theorem 2, in practice one can only take as many iterations as the computational power affords. That said, for sufficiently large block length n and number of super-iterations r, we come arbitrarily close to achieving the fundamental compression limits of the source.

Applications: In applications such as image compression [2, 3], video compression [4], and speech coding [5–7], our algorithms can process a vector of real-valued numbers whose statistics are either completely unknown, or perhaps only known approximately. As an example, consider image coding. The EQ coder [3] processes each sub-band of wavelets sequentially, going from low-frequency sub-bands and proceeding toward high-frequency sub-bands. The EQ coder classifies wavelet coefficients in each sub-band based on the magnitudes of parent coefficients, relying on the insight that the magnitudes of children coefficients are correlated with the magnitudes of parents [43]. In a similar manner, our algorithms can utilize the parent coefficients as contextual information. This approach should compress better than the EQ coder, which assumes that the wavelet coefficients in each subband are independent conditioned on parent coefficients, whereas our algorithm can account for dependencies between coefficients. We leave the application of our algorithms to image coding for future work.

APPENDIX A. PROOF OF THEOREM 1

Continuous codebook: We begin by constructing a continuous amplitude RD codebook [14]. Given the slope β of the RD function $R(X, D)$, there exists an optimal rate $R(X, \beta)$ and distortion level $D(X, \beta)$. For any $\epsilon_1 > 0$, fix the rate $R = R(X, \beta) + \epsilon_1$. The achievable RD coding theorem [13, 14] demonstrates for the source X that in the limit of large n there exist codebooks whose rates are smaller than R and whose expected per symbol distortions are less than $D(X, \beta)$. We choose such a codebook $\mathcal{C}$ comprised of at most 2^{Rn} codewords, each of length n. The encoder maps x^n to the nearest codeword c_j in $\mathcal{C}$ and transmits its index j. The decoder then maps index j to c_j.

Quantized codebook: Now define a quantized codebook $\overline{\mathcal{C}}$ such that $\overline{c_{ij}}$, the i'th entry of the j'th codeword of $\overline{\mathcal{C}}$, is generated by rounding c_{ij}, the i'th entry of the j'th codeword of $\mathcal{C}$, to the closest value in $\overline{\mathcal{Y}}$. (Recall that $-\gamma$ and γ are the smallest and largest values in $\overline{\mathcal{Y}}$, respectively.) Using $\overline{\mathcal{C}}$, the encoder and decoder are identical, except that $\overline{c_{ij}}$ is used instead of c_{ij}. The quantized codebook $\overline{\mathcal{C}}$ requires the same rate as before.[3] However, the distortion provided by the quantized codebook $\overline{\mathcal{C}}$ is different.

Distortion of quantized codebook: To analyze the change in distortion, we consider three cases. In the first case, the original codebook value is an outlier whereas the signal value x_i is not, i.e., $|c_{ij}| > \gamma$ and $|x_i| \leq \gamma$. The truncation of $|c_{ij}|$ to γ reduces the distortion,

$$d(x_i, \overline{c_{ij}}) = (x_i - \overline{c_{ij}})^2 < (x_i - c_{ij})^2 = d(x_i, c_{ij}).$$

The second case occurs when the original codebook and signal values are both outliers, i.e., $|x_i|, |c_{ij}| > \gamma$. As $n \to \infty$, the amount of variance beyond the increasing $\gamma = \lceil \log(n) \rceil$ vanishes, $E[(x_i \cdot 1_{\{|x_i| > \gamma(n)\}})^2] \overset{n\to\infty}{\longrightarrow} 0$, because the source X has finite variance and γ increases (8). Therefore, for any $\delta_1 > 0$ there exists N_1 such that for all $n > N_1$ the increase in expected distortion $d_n(x^n, y^n)$ (1) due to truncation of outliers is smaller than δ_1. The third case occurs for $|c_{ij}| \leq \gamma$, where rounding changes the square error from $(x_i - c_{ij})^2$ to $(x_i - \overline{c_{ij}})^2$, and the distortion changes by

$$\begin{aligned}(x_i - \overline{c_{ij}})^2 - (x_i - c_{ij})^2 &= (\overline{c_{ij}})^2 - (c_{ij})^2 + 2x_i(c_{ij} - \overline{c_{ij}}) \\ &= (c_{ij} - \overline{c_{ij}})(2x_i - \overline{c_{ij}} - c_{ij}) \\ &= (c_{ij} - \overline{c_{ij}}) \left[2(x_i - c_{ij}) + (c_{ij} - \overline{c_{ij}})\right].\end{aligned}$$

Because $|c_{ij} - \overline{c_{ij}}| \leq \frac{1}{2\gamma}$ (8), the change in distortion is upper bounded as follows,

$$|(x_i - \overline{c_{ij}})^2 - (x_i - c_{ij})^2| \leq \frac{|x_i - c_{ij}|}{\gamma} + \frac{1}{4\gamma^2}.$$

We now define sets of indices that relate to the three cases,

$$\begin{aligned}\mathcal{I}_1 &\triangleq \{i:\ i \in \{1, \ldots, n\}, |c_{ij}| > \gamma, |x_i| \leq \gamma\}, \\ \mathcal{I}_2 &\triangleq \{i:\ i \in \{1, \ldots, n\}, |x_i|, |c_{ij}| > \gamma\}, \\ \mathcal{I}_3 &\triangleq \{i:\ i \in \{1, \ldots, n\}, |c_{ij}| \leq \gamma\}.\end{aligned}$$

[3] By quantizing c_j to $\overline{c_j}$, different c_j could yield identical codewords in $\overline{\mathcal{C}}$; this would allow to reduce the rate.

Summarizing over all $i \in \{1, \ldots, n\}$, and taking expectation over the input x^n and codeword $\overline{c_j}$,

$$\begin{aligned} E\left[nd_n(x^n, \overline{c_j})\right] &= E\left[\sum_{i=1}^{n}(x_i - \overline{c_{ij}})^2\right] \\ &= E\left[\left(\sum_{i\in\mathcal{I}_1}(x_i - \overline{c_{ij}})^2\right) + \left(\sum_{i\in\mathcal{I}_2}(x_i - \overline{c_{ij}})^2\right) + \left(\sum_{i\in\mathcal{I}_3}(x_i - \overline{c_{ij}})^2\right)\right] \\ &\leq E\left[\sum_{i\in\mathcal{I}_1}(x_i - c_{ij})^2\right] + E\left[\sum_{i\in\mathcal{I}_2}(x_i - c_{ij})^2 + n\delta_1\right] \\ &\quad + E\left[\sum_{i\in\mathcal{I}_3}(x_i - c_{ij})^2 + \frac{|x_i - c_{ij}|}{\gamma} + \frac{1}{4\gamma^2}\right] \qquad (23) \\ &\leq nE\left[d_n(x^n, c_j)\right] + n\delta_1 + \frac{E\left[\|x^n - c_j\|_1\right]}{\gamma} + \frac{n}{4\gamma^2}, \qquad (24) \end{aligned}$$

where c_j and $\overline{c_j}$ are the j'th codewords of $\mathcal{C}$ and $\overline{\mathcal{C}}$, respectively, $\|\cdot\|_1$ denotes the ℓ_1 norm, the inequality in (23) relies on the changes in distortion in the three different cases, and the inequality in (24) is due to the γ terms related to $\mathcal{I}_3$ that do not appear for $\mathcal{I}_1$ and $\mathcal{I}_2$. Because $E[d_n(x^n, c_j)] \leq D$ and $\|x^n - c_j\|_1 \leq n\sqrt{d_n(x^n, c_j)}$, we have via Jensen's inequality that $E[\|x^n - c_j\|_1] \leq n\sqrt{D}$. Therefore,

$$E[d_n(x^n, \overline{c_j})] < D + \delta_1 + \frac{\sqrt{D}}{\gamma} + \frac{1}{4\gamma^2} = D + \delta_1 + \frac{\sqrt{D}}{\lceil \log(n) \rceil} + \frac{1}{4\lceil \log(n) \rceil^2}. \qquad (25)$$

Because $\gamma = \lceil \log(n) \rceil$ increases with n,

$$E[d(x^n, \overline{c_j})] \leq D + 2\delta_1. \qquad (26)$$

Therefore, the quantized codebook $\overline{\mathcal{C}}$ approaches the RD function asymptotically for the continuous amplitude source X.

Lossless compression using CTW: Having demonstrated that there exists a codebook based on the finite alphabet $\overline{\mathcal{C}}$ that asymptotically achieves the RD function, we need to prove that the RD performance of $\overline{\mathcal{C}}$ can be approached by compressing $\widehat{x^n}$ losslessly using CTW [36]. The remainder of the proof borrows from the prior art on lossy compression of finite sources [22, 44]. Owing to the linearity of expectation,

$$E\left[\frac{1}{n}|CTW(\widehat{x^n})| - \beta d(x^n, \widehat{x^n})\right] = E\left[\frac{1}{n}|CTW(\widehat{x^n})| - H_k(\widehat{x^n})\right] + E\left[H_k(\widehat{x^n}) - \beta d(x^n, \widehat{x^n})\right]. \qquad (27)$$

Recall that $k = k_n = o(\log(n))$, and so for any $\epsilon_2 > 0$ there exists N_2 such that for all $n > N_2$ CTW converges to the empirical entropy [36],

$$E\left[\frac{1}{n}|CTW(\widehat{x^n})| - H_k(\widehat{x^n})\right] < \epsilon_2, \qquad (28)$$

as long as $k_n = \log(n)/\log(|\mathcal{Y}|) - \Omega_n(1)$. Jalali and Weissman [44] invoke Gray et al. [45] to prove that for any $\delta_2 > 0$ and $\epsilon_3 > 0$ there exists a process $\widetilde{X}$ that is jointly stationary and ergodic with X such that

$$\begin{aligned} E\left[H_k(\widehat{x^n}) - \beta d(x^n, \widehat{x^n})\right] &\leq E\left[H_k(\widetilde{x^n}) - \beta d(x^n, \widetilde{x^n})\right] & (29)\\ &\leq H(\widetilde{X_0}|\widetilde{X_{-k}^{-1}}) + \epsilon_3 - E\left[\beta d(x^n, \widetilde{x^n})\right] & (30)\\ &\leq R(X,D) + \epsilon_4 + \epsilon_3 - \beta(D(\beta) + \delta_2), & (31)\end{aligned}$$

where (29) relies on the definition of $\widehat{x^n}$ (6), (30) is explained by observing that $H_k(\widetilde{x^n})$ converges to $H(\widetilde{X_0}|\widetilde{X_{-k}^{-1}})$ with probability one as k_n is increased, and (31) uses properties of $\widetilde{X}$, i.e., $H(\widetilde{X_0}|\widetilde{X_{-k}^{-1}}) \leq R(X,D) + \epsilon_4$ and $E\left[\beta d(x^n, \widetilde{x^n})\right] \leq D(\beta) + \delta_2$. Note also that $R(X,D)$ relies implicitly on β, and is identical to the $R(\beta)$ mentioned earlier. We complete the proof by combining (26), (27), (28), (31), and the arbitrariness of δ_1, δ_2, ϵ_2, ϵ_3, and ϵ_4 □

Appendix B. Proof of Theorem 2

In light of Theorem 1, we need only prove that MCMC converges in distribution to the set of minimal energy solutions. To prove this, we rely on the closely related proof by by Jalali and Weissman [44, Appendix B], and we only outline the arguments here. While the algorithm is running, y_r^n takes one of $|\mathcal{Y}|^n$ possible values. These values are modeled as states of a Markov chain with $|\mathcal{Y}|^n$ states. There is a sufficiently positive probability to transition between any two states, because $c < c_1$ and each super-iteration of Algorithm 1 processes all n locations of y^n. Therefore, as long as the temperature is reduced slowly enough (because $c < c_1$), the probability to transition between any two states is high enough to prevent getting locked into a local minimum, and the distribution of y_r^n converges toward the stationary distribution of the Markov chain. The proof is completed by noting that at low temperatures the minimal-energy states occupy all the probabilistic mass of the stationary distribution, and the stationary distribution consists of these states. Therefore, y_r^n converges in distribution to the set of minimal energy solutions, and we enjoy the same RD performance as in Theorem 1. □

Appendix C. Proof of Theorem 3

The proof is similar to the proofs of Theorems 1 and 2. Consider the sequence $\widehat{z^n}$ with globally minimal modified energy (12),

$$\widehat{z^n} \triangleq \arg\min_{w^n \in \mathcal{Z}^n} \varepsilon_a(w^n). \qquad (32)$$

We first employ arguments from Appendix A to prove that $\widehat{z^n}$ achieves the RD function asymptotically. Next, we prove that simulated annealing [34] converges to the globally optimal solution asymptotically.

Achievable for global minimum: Recall that the adaptive algorithm uses $\mathcal{Z}$ with cardinality $|\mathcal{Z}| = |\overline{\mathcal{Y}}|$. Consider Appendix A, which proves that the globally optimal data-independent reproduction alphabet solution $\widehat{x^n}$ achieves the RD function asymptotically. Because $|\mathcal{Z}| = |\overline{\mathcal{Y}}|$, there exists a one to one mapping from $\overline{\mathcal{Y}}$ to $\mathcal{Z}$, and $\widehat{x^n}$ is mapped to some $\widetilde{z^n}$. The optimal $a^*(\cdot)$ may reduce the distortion,

$$d_n(x^n, a^*(\widehat{z^n})) \leq d_n(x^n, \widehat{x^n}).$$

Although the quantized version $a_q^*(\cdot)$ may increase the distortion, i.e.,

$$d_a(x^n, \widehat{z^n}) = d_n(x^n, a_q^*(\widehat{z^n})) \geq d_n(x^n, a^*(\widehat{z^n})),$$

allocating $\mu \log(\log(n))$ bits to encode each $a_q^*(\alpha)$ is sufficient to guarantee that the quantization error is smaller than $\frac{1}{\gamma}$ (see the proof of Theorem 1 in Appendix A). Our previous derivations (24), (25), (26) show that for any $\delta > 0$ the overall distortion $d_a(x^n, \widehat{z^n})$ becomes δ-close to $D(\gamma)$ as n is increased. We conclude from the definitions of energy (4) and modified adaptive energy (12) that

$$\varepsilon_a(\widehat{z^n}) \leq \varepsilon(\widehat{x^n}) + n\beta\delta + \mu \log(\log(n))|\overline{\mathcal{Y}}|.$$

Because $|\overline{\mathcal{Y}}| = O(\log^2(n))$, the last term due to encoding the quantized $a_q^*(\cdot)$ vanishes relative to $n\beta\delta$. Taking δ as small as we want enables to approach $\min_{D \geq 0}[R(X, D) - \beta D]$ as closely as needed,

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

Invoking the converse result of Yang et al. [22, 33],

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

Simulated annealing: The proof is similar to that in Appendix B. The only noteworthy point is that for each z^n the quantized $a_q^*(\cdot)$ is a deterministic function of z^n. Therefore, the simulated annealing can again be posed as a Markov chain over $|\mathcal{Z}|^n$ states, where convergence in distribution to the set of minimal energy solutions is obtained by recognizing that for $c < c_2$ there is a sufficiently positive probability to transition between any two states. Therefore, the Markov chain does not get stuck in a local minimum, and instead it converges toward the stationary distribution of the Markov chain, which at low temperatures consists entirely of minimal-energy states. □

ACKNOWLEDGMENTS

This work was partially supported by Israel Science Foundation grant number 2013433 and by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370. The first author thanks the generous hospitality of the Electrical Engineering Department at the Technion, Israel, where parts of this work were performed. We also thank Shirin Jalali for sharing her software implementations of the algorithms in [23], and Ken Rose for enlightening discussions on the discrete nature of the codebook [35].

REFERENCES

[1] D. Baron and T. Weissman, "An MCMC approach to lossy compression of continuous sources," in *Proc. Data Compression Conf. (DCC)*, Mar. 2010, pp. 40–48.

[2] 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, May 1997.

[3] 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)*, Mar. 1997, pp. 221–230.

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

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

[6] A. Buzo, A. Gray Jr, R. Gray, and J. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoustics, Speech and Signal Process.*, vol. 28, no. 5, pp. 562–574, 1980.

[7] M. Sabin and R. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Trans. Acoustics, Speech and Signal Process.*, vol. 32, no. 3, pp. 474–488, 1984.

[8] 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.

[9] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[10] J. Max, "Quantization for minimum distortion," *IRE Trans. Inf. Theory*, vol. 6, no. 1, pp. 7–12, Mar. 1960.

[11] 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.

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

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

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

[15] P. Chou, T. Lookabaugh, and R. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoustics, Speech and Signal Process.*, vol. 37, no. 1, pp. 31–42, 1989.

[16] E. Riskin and R. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizers [image compression]," *IEEE Trans. Signal Process.*, vol. 39, no. 11, pp. 2500–2507, 1991.

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

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

[19] A. Gupta, S. Verdú, and T. Weissman, "Rate-distortion in near-linear time," in *Proc. Int. Symp. Inf. Theory (ISIT2008)*, Jul. 2008.

[20] 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.

[21] 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.

[22] 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.

[23] S. Jalali and T. Weissman, " Block and sliding-block lossy compression via MCMC," *IEEE Trans. Comm.*, 2012, to appear.

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

[25] 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.

[26] 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.

[27] 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.

[28] C. Weidmann and M. Vetterli, "Rate distortion behavior of sparse sources," 2008, submitted.

[29] ——, "Rate-distortion analysis of spike processes," in *Proc. Data Compression Conf. (DCC)*, Mar. 1999, pp. 82–91.

[30] R. Castro, M. B. Wakin, and M. Orchard, "On the problem of simultaneous encoding of magnitude and location," in *Asilomar Conf. Signals, Syst., Comput.*, 2002.

[31] C. Chang, "On the rate distortion function of Bernoulli Gaussian sequences," in *Int. Symp. Inf. Theory (ISIT2010)*, 2010, pp. 66–70.

[32] 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.

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

[34] 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, Nov. 1984.

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

[36] F. M. J. Willems, Y. M. 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.

[37] 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.

[38] J. Wolfowitz, *Coding theorems of information theory*. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1978.

[39] M. Box, D. Davies, and W. Swann, *Non-linear optimization techniques*. Oliver & Boyd, 1969, no. 5.

[40] T. Berger and J. Gibson, "Lossy source coding," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2693–2723, 1998.

[41] R. E. Blahut, “Computation of channel capacity and rate-distortion functions,” *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 460–473, Jul. 1972.

[42] S. Arimoto, “An algorithm for calculating the capacity of an arbitrary discrete memoryless channel,” *IEEE Trans. Inf. Theory*, vol. 18, pp. 14–20, Jan. 1972.

[43] S. Mallat, *A wavelet tour of signal processing*. Academic Press, 1999.

[44] S. Jalali and T. Weissman, “Rate-distortion via Markov chain Monte Carlo,” *Arxiv preprint arXiv:0808.4156*, 2008.

[45] R. Gray, D. Neuhoff, and J. Omura, “Process definitions of distortion-rate functions and source coding theorems,” *Trans. Inf. Theory*, vol. 21, no. 5, pp. 524–532, Sep. 1975.