Block and Sliding-Block Lossy Compression via MCMC

Shirin Jalali* and Tsachy Weissman†,

*Center for Mathematics of Information, CalTech, Pasadena, CA 91125

†Department of Electrical Engineering, Stanford University, Stanford, CA 94305

Abstract

We propose an approach to lossy compression of finite-alphabet sources that utilizes Markov chain Monte Carlo (MCMC) and simulated annealing methods. The idea is to define an energy function over the space of reconstruction sequences. The energy of a candidate reconstruction sequence is defined such that it incorporates its distortion relative to the source sequence, its compressibility, and the point sought on the rate-distortion curve. The proposed algorithm samples from the Boltzmann distribution associated with this energy function using the 'heat-bath' algorithm.

The complexity of each iteration is independent of the sequence length and only linearly dependent on a certain context parameter (which grows sub-logarithmically with the sequence length). We show that the proposed algorithm achieves optimum rate-distortion performance in the limits of large number of iterations, and sequence length, when employed on any stationary ergodic source. Inspired by the proposed block-coding algorithm, we also propose an algorithm for constructing sliding-block (SB) codes using similar ideas.

Index Terms

Rate-distortion coding, Universal lossy compression, Markov chain Monte Carlo, Gibbs sampler, Simulated annealing

I. INTRODUCTION

Consider the problem of lossy compression of a finite-alphabet stationary ergodic source $\mathbf{X} = \{X_i : i \geq 1\}$. Each source output block of length n, X^n, is mapped to $f_n(X^n)$, an index of nr bits, where R can either be constant (fixed-rate coding) or depend on the block that is coded (variable-rate coding). The index $f_n(X^n)$ is then losslessly transmitted to the decoder, and is mapped to a reconstruction block $\hat{X}^n = g_n(f_n(X^n))$, where $\hat{X}^n \in \hat{\mathcal{X}}^n$. The two main measures used to evaluate the performance of a lossy coding scheme $\mathcal{C} = (f_n, g_n, n)$ are the following: i) Distortion D defined as the expected average

distortion between source and reconstruction blocks, i.e., $D \triangleq \mathrm{E}[d_n(X^n, \hat{X}^n)] \triangleq \frac{1}{n}\sum_{i=1}^{n} \mathrm{E}[d(X_i, \hat{X}_i)]$, where $d : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}^+$ is a single-letter distortion measure, and ii) Rate R defined as the average expected number of coded bits per source symbol, i.e., $R \triangleq \mathrm{E}[r]$. For any $D \geq 0$, and stationary process $\mathbf{X}$, its minimum achievable rate at distortion D is characterized as (cf. [1], [2], [3])

$$R(D, \mathbf{X}) = \lim_{n\to\infty} \min_{p(\hat{X}^n|X^n):\mathrm{E}[d_n(X^n,\hat{X}^n)]\leq D} \frac{1}{n} I(X^n; \hat{X}^n). \tag{1}$$

The minimum required rate for lossless compression of a source is its entropy rate defined as $\bar{H}(\mathbf{X}) \triangleq \lim_{k\to\infty} H(X_0|X_{-k}^{-1})$. There are known implementable *universal* lossless codes, such as Lempel-Ziv [4], arithmetic coding [5], etc. that losslessly compress any stationary ergodic source down to its entropy rate. In contrast, neither the explicit solution of (1) is known for a general source (not even for a first-order Markov source [6]), nor are there known practical schemes that universally achieve the rate-distortion curve.

One possible intuitive explanation for this sharp dichotomy is as follows. The essence of universal lossless compression is *learning* the source distribution, and the difference between various coding algorithms is in different efficient methods through which they accomplish this goal. Universal lossy compression, on the other hand, intrinsically consists of two components: quantization and lossless compression. This breakdown can be explained more clearly by the characterization of the rate-distortion function as $R(D, \mathbf{X}) = \inf\{\bar{H}(\mathbf{Z}) : \mathrm{E}[d(X_1, Z_1)] \leq D\}$, where the minimization is over all processes that are jointly stationary ergodic with $\mathbf{X}$ [7]. This representation suggests the following alternative approach to lossy compression: To code a process $\mathbf{X}$, we quantize it, either implicitly or explicitly, to another process $\mathbf{Z}$, which is sufficiently close to $\mathbf{X}$ but more compressible, and then compress $\mathbf{Z}$ via a universal lossless compression algorithm. The quantization step in fact involves a search over jointly stationary ergodic processes, and is another way to understand why universal lossy compression is more difficult than universal lossless compression.

In this paper, we present an approach to implementable lossy source coding of finite-alphabet sources, which borrows tools from statistical physics and computer science: Markov Chain Monte Carlo (MCMC) methods, and simulated annealing [8], [9]. MCMC methods refer to a class of algorithms that are designed to generate samples of a given distribution through generating a Markov chain having the desired distribution as its stationary distribution. MCMC methods include a large number of algorithms. For our application, we use Gibbs sampler [10] also known as the *heat bath* algorithm, which is well-suited to the case where the desired distribution is hard to compute, but the conditional distributions of

each variable given the rest are easy to work out.

The second required tool is simulated annealing. It is a well-known method in discrete optimization for finding the minimizer of a function $f(\mathbf{s})$ over a possibly huge set of states $\mathcal{S}$, i.e., $\mathbf{s}_{\min} = \arg\min_{\mathbf{s}\in\mathcal{S}} f(\mathbf{s})$. Consider a sequence of probability distributions $\{p_i\}_{i=1}^{\infty}$ corresponding to the temperatures $\{T_i\}_{i=1}^{\infty}$, where $T_i \to 0$ as $i \to \infty$. At each time i, the algorithm runs one of the relevant MCMC methods in an attempt to sample from distribution p_i. The sequence of probability distributions is designed such that: 1) their output, with high probability, is the minimizing state $\mathbf{s}_{\min}$, or a state achieving a value close to it, 2) the probability of getting the minimizing state increases as the temperature drops. The probability distribution that satisfies these characteristics, and is almost always used, is the Boltzmann distribution $p_{\beta_i}(\mathbf{s}) \propto e^{-\beta_i f(\mathbf{s})}$, where $\beta_i \propto 1/T_i$. It can be proved that, if the temperature drops slowly enough, the probability of getting the minimizing state as the output of the algorithm approaches one [10].

Simulated annealing and related ideas such as deterministic annealing have been suggested before in the context of lossy compression, either for approximating the rate distortion function, i.e., the optimization problem involving minimization of the mutual information, or as a method for designing the codebook in vector quantization [11], [12], as an alternative to the conventional generalized Lloyd algorithm (GLA) [13]. In contrast, in this paper we use the simulated annealing approach to obtain a particular reconstruction sequence, rather than a codebook.

Let us briefly describe how the new algorithm codes a source sequence x^n. First, to each reconstruction block y^n, it assigns an *energy*, $\mathcal{E}(y^n)$, which is a linear combination of its conditional empirical entropy, to be defined formally in the next section, and its distortion relative to the source sequence x^n. Then, it assumes a Boltzman probability distribution of the form $p(y^n) \propto e^{-\beta\mathcal{E}(y^n)}$ over the space of reconstruction blocks , and generates $\hat{x}^n$ by sampling from this distribution using Gibbs sampler [10]. As we show, for β large enough, with high probability the reconstruction block of our algorithm satisfies $\mathcal{E}(\hat{x}^n) \approx \min \mathcal{E}(y^n)$. The encoder will output $\mathsf{LZ}(\hat{x}^n)$, which is the Lempel-Ziv [4] description of $\hat{x}^n$. The decoder, upon receiving $\mathsf{LZ}(\hat{x}^n)$, reconstructs $\hat{x}^n$ losslessly.

Instead of working at a fixed rate or at a fixed distortion, we are fixing the slope. A fixed-slope rate-distortion scheme working at a fixed slope $s = -\alpha < 0$ is a scheme that minimizes for $R + \alpha D$.

A. Prior work

The literature on universal lossy compression can be divided into two main categories: proofs of existence [14], [15], [16], [17], [18], [19] and algorithm designs. In this section we briefly review some of the work on the latter.

One popular trend in designing universal lossy compression algorithms in extending universal lossless compression algorithms to the case of lossy compression. An example of such attempts is the work by Cheung and Wei [20] who extended the move-to-front transform [21]. Morita and Kobayashi [22] proposed a lossy version of LZW algorithm and Steinberg and Gutman [23] suggested fixed-database lossy compression algorithms based on string-matching. All these extensions, as was later shown by Yang and Kieffer [24], are suboptimal even for memoryless sources. Another suboptimal but practical universal lossy compression algorithm based on approximate pattern matching is the work of Luczak and Szpankowski [25]. In [26], Zamir and Rose proposed an adaptive lossy compression algorithm based on type selection for memoryless sources. More recently, a new lossy version of the LZ algorithm has been proposed by Kontoyiannis [27], which instead of using a fixed database which has the same distribution as the source, employs multiple databases.

Zhang and Wei [28] proposed an online universal lossy data compression algorithm, called 'gold-washing', which involves continuous codebook refinement. The algorithm is called 'online' meaning that the codebook is constructed simultaneously by the encoder and the decoder as the source symbols arrive, and no codebook is shared between the two before the coding starts. Most of the previously mentioned algorithms fall into the class of online algorithms as well.

As mentioned before, the approach we take here is one of fixing the slope. The idea of fixed-slope universal lossy compression was considered by Yang, Zhang and Berger in [29]. In that paper, they first propose an exhaustive search algorithm which is very similar to the algorithm proposed in Section III. After establishing its universality for lossy compression of stationary ergodic sources, they suggest a heuristic approach to approximate its solution. In our case, the special structure of our cost function enables us to employ simulated annealing plus Gibbs sampling to approximate the minimizer.

For the non-universal setting, specifically the case of lossy compression of an i.i.d. source with a known distribution, there is an ongoing progress towards designing codes that approach the optimum performance [30], [31], [32], [33], [34].

B. Paper organization

The organization of the paper is as follows. In Section II, we set up the notation. Section III describes an exhaustive search scheme for fixed-slope lossy compression which universally achieves the rate-distortion curve for any stationary ergodic source. Section IV describes our new universal MCMC-based lossy block coder. Inspired by the proposed block lossy compression algorithm, in Section V we propose an algorithm for constructing SB codes using MCMC and simulated annealing methods. Section VI presents some

simulations results. We conclude in Section VII with a discussion of some future directions.

II. Notation and definitions

Calligraphic letters such as $\mathcal{X}$, $\mathcal{Y}$ represent sets. The size of a set $\mathcal{X}$ is denoted by $|\mathcal{X}|$. For $1 \leq i \leq j \leq n$, $x_i^j = (x_i, x_{i+1}, \ldots, x_j)$. For two vectors x^i and y^j, $x^i y^j$ denotes a vector of length $i+j$ formed by concatenating the two vector as $(x_1, \ldots, x_i, y_1, \ldots, y_j)$. Capital letters represent random variables and capital bold letters represent random vectors. For a random variable X, let $\mathcal{X}$ denote its alphabet set. For an event $\mathcal{A}$, $\mathbb{1}_{\mathcal{A}}$ denotes an indicator function of event $\mathcal{A}$. For $\mathbf{u} \in \mathbb{R}^n$, let $\|\mathbf{u}\|_1 = \sum_{i=1}^n |u_i|$.

Let $\mathbf{X} = \{X_i; \forall\, i \in \mathbb{N}^+\}$ be a stochastic process defined on a probability space $(\mathbf{X}, \Sigma, \mu)$, where Σ denotes the σ-algebra generated by cylinder sets $\mathcal{C}$, and μ is a probability measure defined on it. For a process $\mathbf{X}$, let $\mathcal{X}$ denote the alphabet of X_i. Throughout the paper $\mathcal{X}$ is assumed to be a finite set. For a stationary process $\mathbf{X}$, let $\bar{H}(\mathbf{X})$ denote its entropy rate defined as $\bar{H}(\mathbf{X}) = \lim_{n\to\infty} H(X_{n+1}|X^n)$.

For $y^n \in \mathcal{Y}^n$, define the $|\mathcal{Y}| \times |\mathcal{Y}|^k$ matrix $\mathbf{m}(y^n)$ to denote the $(k+1)^{\text{th}}$ order empirical distribution of y^n. Each column and each row of $\mathbf{m}$ are indexed by a k-tuple $b^k \in \mathcal{Y}^k$ and some $\beta \in \mathcal{Y}$, respectively. The element in row $\beta \in \mathcal{Y}$ and column $b^k \in \mathcal{Y}^k$ of $\mathbf{m}$ is defined as

$$m_{\beta,b^k}(y^n) = \frac{1}{n-k} \left| \left\{ k+1 \leq i \leq n : y_{i-k}^{i-1} = b^k, y_i = \beta \right\} \right|.$$

Let $H_k(y^n)$ denote the k^{th} order conditional empirical entropy induced by y^n, i.e.,

$$H_k(y^n) = \sum_{b^k \in \mathcal{Y}^k} \|\mathbf{m}_{\cdot,b^k}(y^n)\|_1 \mathcal{H}\left(\mathbf{m}_{\cdot,b^k}(y^n)\right), \tag{2}$$

where $\mathbf{m}_{\cdot,b^k}(y^n)$ denotes the column in $\mathbf{m}(y^n)$ corresponding to b^k, and for a vector $\mathbf{v} = (v_1, \ldots, v_\ell)$ with non-negative components, $\mathcal{H}(\mathbf{v})$ denotes the entropy of the random variable whose pmf is proportional to $\mathbf{v}$. Formally,

$$\mathcal{H}(\mathbf{v}) = \begin{cases} \sum_{i=1}^{\ell} \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{\|\mathbf{v}\|_1}{v_i} & \text{if } \mathbf{v} \neq (0, \ldots, 0) \\ 0 & \text{if } \mathbf{v} = (0, \ldots, 0), \end{cases} \tag{3}$$

where $0 \log(0) \triangleq 0$ by convention.[1]

Alternatively, $H_k(y^n) \triangleq H(U_{k+1}|U^k)$, where U^{k+1} is distributed according to the $(k+1)^{\text{th}}$ order empirical distribution induced by y^n, i.e., $\mathrm{P}(U^{k+1} = [b^k, \beta]) = m_{\beta,b^k}(y^n)$.

[1]Here and throughout the paper the base of the logarithms is 2.

III. An exhaustive search scheme for fixed-slope compression

Consider the following scheme for lossy source coding at a fixed slope $\alpha > 0$. For each source sequence x^n, let the reconstruction block $\hat{x}^n$ be defined as $\hat{x}^n = \arg\min_{y^n \in \hat{\mathcal{X}}^n} [H_k(y^n) + \alpha d_n(x^n, y^n)]$. The encoder, after computing $\hat{x}^n$, losslessly conveys it to the decoder using the LZ compressor [4].

Theorem 1: Let $\mathbf{X}$ be a stationary ergodic source, and $R(D, \mathbf{X})$ denote its rate distortion function. Let $\hat{X}^n$ and $\mathsf{LZ}(\hat{X}^n)$ denote the reconstruction sequence generated by the proposed method and its binary representation using the Lempel-Ziv compression, respectively. Then

$$\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n) + \alpha d_n(X^n, \hat{X}^n) \stackrel{n\to\infty}{\longrightarrow} \min_{D\geq 0} \left[R(D, \mathbf{X}) + \alpha D\right], \tag{4}$$

almost surely. (Here $\ell_{\mathsf{LZ}}(\hat{X}^n)$ denotes the length of the binary sequence $\mathsf{LZ}(\hat{X}^n)$.)
In words, the above scheme universally attains the optimum rate-distortion performance at slope α for any stationary ergodic process.

The exhaustive search algorithm described above is very similar to the generic algorithm proposed in [29], which works as follows

$$\hat{x}^n = \arg\min_{y^n \in \hat{\mathcal{X}}^n} \left[\frac{1}{n} l(y^n) + \alpha d_n(x^n, y^n)\right], \tag{5}$$

where $l(y^n)$ represents the length of the binary codeword assigned to y^n by some universal lossless compression algorithm. In other words, a length function $l(y^n)$ denotes a mapping from $\hat{\mathcal{X}}^n \to \mathbb{N}^+$ that satisfies the following two conditions:

1) Kraft inequality: $\sum_{y^n \in \hat{\mathcal{X}}^n} 2^{-l(y^n)} \leq 1$, for each $n \in \mathbb{N}$.
2) For any stationary ergodic process $\mathbf{X}$, $\limsup_{n\to\infty} \frac{1}{n} l(X^n) \leq \bar{H}(\mathbf{X})$, almost surely.

Although the conditional empirical entropy function, $H_k(\cdot)$, is not a length function,[2] it has a close connection to length functions, specifically to $\ell_{\mathrm{LZ}}(\cdot)$. This link is described by the Ziv inequality [35]. Ziv's inequality states that if $k_n \to \infty$ as $n \to \infty$ while $k_n = o(\log n)$, then for any $\epsilon > 0$, there exists $N_\epsilon \in \mathbb{N}$ such that for *any* individual infinite-length sequence $\mathbf{y} = (y_1, y_2, \ldots)$ and any $n \geq N_\epsilon$,

$$\left[\frac{1}{n}\ell_{\mathsf{LZ}}(y^n) - H_{k_n}(y^n)\right] \leq \epsilon.$$

Using this connection and the results of [29], we prove Theorem 1 in Appendix A.

The drawback of the described algorithm is its computational complexity; It involves exhaustive search among the set of all possible reconstructions. The size of this set is $|\hat{\mathcal{X}}|^n$ growing exponentially fast with

[2]Note that $\sum_{y^n} 2^{-nH_k(y^n)} \geq 2^{-nH_k(0,\ldots,0)} + 2^{-nH_k(1,\ldots,1)} = 2$, for any k and n

n.

IV. Universal lossy coding via MCMC

In this section, we will show how simulated annealing and Gibbs sampling enable us to get close to the performance of the impractical exhaustive search coding algorithm described in the previous section. Throughout this section we fix the slope $\alpha > 0$.

Associate with each reconstruction sequence y^n the *energy* $\mathcal{E}(y^n) \triangleq H_k(y^n) + \alpha d_n(x^n, y^n)$, and define the *Boltzmann distribution* as the pmf on $\hat{\mathcal{X}}^n$ given by $p_\beta(y^n) = e^{-\beta \mathcal{E}(y^n)}/Z$, where $Z = \sum_{y^n} e^{-\beta \mathcal{E}(y^n)}$ is the normalization constant (partition function). [3] For large enough values of β, a sample Y^n from p_β, with high probability, satisfies $\mathcal{E}(Y^n) \approx \min_{y^n} \mathcal{E}(y^n)$. Thus, for large β, using a sample from the Boltzmann distribution p_β as the reconstruction sequence yields a performance close to that of an exhaustive search scheme that would use the minimizer of $\mathcal{E}(\cdot)$. Unfortunately, it is hard to sample from the Boltzmann distribution directly. However, as described next, we can get samples close to the optimal point via MCMC.

As mentioned earlier, the Gibbs sampler [10] is a useful sampling method. Its main application is in cases where, although it is prohibitively complex to evaluate the desired probability distribution, its conditional distributions of each variable given the rest are accessible. In our case, the conditional probability distribution under p_β of Y_i given the other variables $Y^{n\setminus i} \triangleq (Y_j : j \neq i)$ can be expressed as

$$\begin{aligned}
\mathrm{P}(Y_i = a | Y^{n\setminus i} = y^{n\setminus i}) &= \frac{p_\beta(y^{i-1}\ a\ y_{i+1}^n)}{\sum_{b\in\hat{\mathcal{X}}} p_\beta(y^{i-1}\ b\ y_{i+1}^n)} \\
&= \frac{e^{-\beta \mathcal{E}(y^{i-1} a y_{i+1}^n)}}{\sum_{b\in\hat{\mathcal{X}}} e^{-\beta \mathcal{E}(y^{i-1} b y_{i+1}^n)}} \\
&= \frac{e^{-\beta (H_k(y^{i-1} a y_{i+1}^n) + \alpha d_n(x^n, y^{i-1} a y_{i+1}^n))}}{\sum_{b\in\hat{\mathcal{X}}} e^{-\beta (H_k(y^{i-1} b y_{i+1}^n) + \alpha d_n(x^n, y^{i-1} b y_{i+1}^n))}} \\
&= \frac{1}{1 + \sum\limits_{b\in\hat{\mathcal{X}}, b\neq a} e^{-\beta(\Delta H_k(a,b,y^{i-1}, y_{i+1}^n) + \alpha \Delta d(a,b,x_i))}},
\end{aligned} \tag{6}$$

where

$$\Delta H_k(a, b, y^{i-1}, y_{i+1}^n) \triangleq H_k(y^{i-1} b y_{i+1}^n) - H_k(y^{i-1} a y_{i+1}^n),$$

[3] Note that, though this dependence is suppressed in the notation for simplicity, $\mathcal{E}(y^n)$, and therefore also p_β and Z depend on x^n and α, which are fixed until further notice.

and

$$\begin{aligned}\Delta d(a,b,x_i) &\triangleq d_n(x^n, y^{i-1}by_{i+1}^n) - d_n(x^n, y^{i-1}ay_{i+1}^n)\\ &= \frac{d(x_i,b) - d(x_i,a)}{n}.\end{aligned}$$

Hence, for $(a,b) \in \hat{\mathcal{X}}^2$, $\mathrm{P}(Y_i = a | Y^{n\setminus i} = y^{n\setminus i})$ depends on y^n only through $\Delta H_k(a,b,y^{i-1},y_{i+1}^n)$ and $\Delta d(a,b,x_i)$. In turn,

$$\{\Delta H_k(y^{i-1}by_{i+1}^n, a)\}_{(a,b)\in\hat{\mathcal{X}}^2}$$

depends on y^n only through $\{\mathbf{m}(y^{i-1}by_{i+1}^n)\}_b$.

Since the number of contexts whose counts are affected by changing one component of y^n is at most $2k+2$, given $\mathbf{m}(y^{i-1}ay_{i+1}^n)$, the number of operations required to obtain $\mathbf{m}(y^{i-1}by_{i+1}^n)$ is linear in k. To be more specific, letting $\mathcal{S}_i(y^n,b)$ denote the set of contexts whose counts are affected when the i^{th} component of y^n is flipped from y_i to b. We have $|\mathcal{S}_i(y^n,b)| \le 2k$. Further, since

$$\begin{aligned}\Delta H_k(a,b,y^{i-1},y_{i+1}^n) = \sum_{u^k\in\mathcal{S}_i(y^{i-1}by_{i+1},a)} \big[&\|\mathbf{m}_{\cdot,u^k}(y^{i-1}by_{i+1}^n)\|_1 \mathcal{H}\left(\mathbf{m}_{\cdot,u^k}(y^{i-1}by_{i+1}^n)\right) -\\ &\|\mathbf{m}_{\cdot,u^k}(y^{i-1}ay_{i+1}^n)\|_1 \mathcal{H}\left(\mathbf{m}_{\cdot,u^k}(y^{i-1}ay_{i+1}^n)\right)\big],\end{aligned} \tag{7}$$

it follows that, given $\mathbf{m}(y^{i-1}ay_{i+1}^n)$ and $H_k(y^{i-1}ay_{i+1}^n)$, the number of operations required to compute $\mathbf{m}(y^{i-1}by_{i+1}^n)$ and $H_k(y^{i-1}by_{i+1}^n)$ is linear in k (and independent of n).

Now consider the following algorithm (Algorithm 1 below) based on the Gibbs sampling for sampling from p_β. Let $\hat{X}_{\alpha,r}^n(X^n)$ denote its (random) outcome when applied to the source sequence X^n,[4] taking $k = k_n$ and $\beta = \{\beta_t\}_t$ to be deterministic sequences satisfying $k_n \to \infty$, $k_n = o(\log n)$, and $\beta_t = \frac{1}{T_0^{(n)}} \log(\lfloor \frac{t}{n} \rfloor + 1)$, for some $T_0^{(n)} > n\Delta$, where

$$\Delta = \max_i \max_{u^{i-1}\in\hat{\mathcal{X}}^{i-1}, u_{i+1}^n\in\hat{\mathcal{X}}^{n-i}, a,b\in\hat{\mathcal{X}}} \left|\mathcal{E}(u^{i-1}au_{i+1}^n) - \mathcal{E}(u^{i-1}bu_{i+1}^n)\right|. \tag{8}$$

By the previous discussion, the computational complexity of the algorithm at each iteration is independent of n and linear in k.

[4]Here and throughout it is implicit that the randomness used in the algorithms is independent of the source, and the randomization variables used at each drawing are independent of each other.

Algorithm 1 Generating the reconstruction sequence

Input: x^n, k, α, $\{\beta_t\}_t$, r
Output: a reconstruction sequence $\hat{x}^n$

1: $y^n \leftarrow x^n$
2: **for** $t = 1$ to r **do**
3: Draw an integer $i \in \{1, \ldots, n\}$ uniformly at random
4: For each $b \in \hat{\mathcal{X}}$ compute $p_{\beta_t}(Y_i = b | Y^{n\setminus i} = y^{n\setminus i})$ given by (6)
5: Update y^n by replacing its i^{th} component y_i by Z, where $Z \sim p_{\beta_t}(Y_i = \cdot | Y^{n\setminus i} = y^{n\setminus i})$
6: Update $\mathbf{m}(y^n)$ and $H_k(y^n)$
7: **end for**
8: $\hat{x}^n \leftarrow y^n$

Theorem 2: Let $\mathbf{X}$ be a stationary ergodic source. Then

$$\lim_{n\to\infty} \lim_{r\to\infty} \left[\frac{1}{n} \ell_{\mathsf{LZ}} \left(\hat{X}^n_{\alpha,r}(X^n) \right) + \alpha d_n(X^n, \hat{X}^n) \right] = \min_{D\geq 0} \left[R(D, \mathbf{X}) + \alpha D \right], \quad \text{a.s.} \tag{9}$$

Theorem 2 is proved by a direct application of the results in [36]. Note that the Markov chain defined by the transition probabilities $\{p_{\beta_t}(Y_i = a | Y^{n\setminus i} = y^{n\setminus i})\}_t$ is a non-homogeneous Markov chain, whose transition probabilities are varying with time. A non-homogeneous Markov chain is called strongly ergodic if there exists a distribution over its state space such that for any distributions $\boldsymbol{\mu}$ and any $n_1 \in \mathbb{N}$, $\limsup_{n_2\to\infty} \|\boldsymbol{\mu}\mathbf{P}^{(n_1,n_2)} - \boldsymbol{\pi}\|_1 = 0$, where $\mathbf{P}^{(n_1,n_2)}$ denotes the transition probabilities matrix between times n_1 and $n_2 > n_1$. The proof's idea is to show the defined Markov chain is strongly ergodic and decreasing the temperature according to $\{\beta_t\}$ implies that its steady state distribution, as the number of iterations converge to infinity, is the uniform distribution over the set of all sequences with minimum energy.

V. Sliding-block rate-distortion coding via MCMC

The conventional method for lossy compression is block coding, where the source symbols are divided into non-overlapping blocks of size $n \in \mathbb{N}$ and each block, independent of the other blocks, is mapped to a reconstruction block of length n [1]. One of the disadvantages of the block codes is that they do not conserve the stationarity of the source process, i.e., a block code maps a stationary process to a non-stationary reconstruction process. The idea of sliding-block coding was introduced by R.M. Gray, D.L. Neuhoff, and D.S. Ornstein in [37], and independently by K. Marton in [38], both in 1975. In this alternative approach to lossy compression, a fixed mapping of a certain window length $2k_f + 1$ slides over the source sequence and generates the reconstruction sequence. The entropy rate of the

reconstruction process is less than the entropy rate of the original process, and can be conveyed to the decoder using a universal lossless compression algorithm. It has been proved that the achievable rate-distortion performances of SB codes and block codes coincide [39].

There are a couple of advantages in using SB codes instead of block codes. One main benefit is that SB codes preserve the stationarity of the source process. This enables the compression algorithms to get rid of the blocking artifacts which result from applying the code to non-overlapping adjacent blocks of data. This issue has been extensively studied in image compression, and one of the reasons wavelet transform is preferred over more traditional image compression schemes like DCT is that it can be implemented as a sliding-window transform, and therefore does not introduce blocking artifacts [40]. The other advantages of SB codes are in terms of speed and memory-efficiency.

Although SB codes seem to serve as a good alternative to block codes, there has been very little progress in desiging such codes since their introduction in 1975, and up to date there is no known practical method for finding even sub-optimal SB codes. In this section, inspired by the lossy compression algorithm proposed in Section IV, we present an iterative algorithm for finding SB codes using simulated annealing and Markov chain Monte Carlo methods. To our knowledge, this is the first implementable algorithm for designing SB codes. The algorithm is an iterative algorithm. It starts from the identity mapping and at each iteration probabilistically decides whether to change one of the mapped values. We show that as the number of iterations grows, if the parameters are chosen appropriately, this algorithm achieves the rate-distortion curve of any stationary ergodic process. Our initial simulations show that in coding a first-order Markov source, even for a small window length of 11, with a moderate number of iterations the algorithm achieves points close to the rate-distortion curve.

A SB code of window length $2k_f+1$ is defined by a mapping $f\colon \mathcal{X}^{2k_f+1} \to \hat{\mathcal{X}}$. Applying the mapping f to the source process $\mathbf{X} = \{X_i\}_{i=-\infty}^{\infty}$ generates the reconstruction process $\hat{\mathbf{X}} = \{\hat{X}_i\}_{i=-\infty}^{\infty}$, where $\hat{X}_i = f(X_{i-k_f}^{i+k_f})$. There exist $K_f = |\mathcal{X}|^{2k_f+1}$ vectors of length $2k_f+1$ with elements taking values in $\mathcal{X}$. Therefore, to specify a SB code of window length $2k_f+1$, there are K_f values to be determined. Hence f can be represented as a vector $\mathbf{f} = [f_0, f_1, \ldots, f_{K_f-1}]$ where $f_i \in \hat{\mathcal{X}}$ is the output of the function f to the input vector $\mathbf{b} = (b_1, b_2, \ldots, b_{2k_f+1}) \in \mathcal{X}^{2k_f+1}$, such that

$$i = \sum_{j=1}^{2k_f+1} r(b_j)|\mathcal{X}|^{j-1},$$

where $r : \mathcal{X} \to \{0, 1, \ldots, |\mathcal{X}|-1\}$ is a fixed arbitrary one-to-one mapping from the source symbols to $\{0, 1, \ldots, |\mathcal{X}|-1\}$.

Example 1: Consider the case of binary source and reconstruction alphabets, i.e., $\mathcal{X} = \hat{\mathcal{X}} = \{0, 1\}$*, and let* $k_f = 1$*. In this case a SB code of window length* 3 *can be specified by 8 values* $(f_0, f_1, \ldots, f_7)$*. For* $x \in \mathcal{X}$*, letting* $r(x) = x$*,* f_i *determines the binary value assigned to the binary expansion of* i*,* $0 \leq i \leq 7$*, in three binary digits. For instance,* $f_3 \in \{0, 1\}$ *denotes the value assigned to the binary vector* $(0, 1, 1)$*.*

To a SB code defined by the vector $\mathbf{f}$, assign the energy $\mathcal{E}(\mathbf{f})$ defined as

$$\mathcal{E}(\mathbf{f}) \triangleq H_k(y^n) + \alpha d_n(x^n, y^n), \tag{10}$$

where $y^n = y^n[x^n, \mathbf{f}]$ is defined by $y_i \triangleq f(x_{i-k_f}^{i+k_f})$, for $k_f + 1 \leq i \leq n - k_f$, and $y_i \triangleq x_i$ otherwise. Consider the SB code $\mathbf{f}_o$ which minimizes the energy function, i.e., $\mathbf{f}_o = \arg\min_{\mathbf{f} \in \hat{\mathcal{X}}^{|\mathcal{X}|^{2k_f+1}}} \mathcal{E}(\mathbf{f})$.

While in block coding at block length n, the search space includes $|\hat{\mathcal{X}}|^n$ possible reconstruction sequences, in SB coding of window length $2k_f + 1$, the search space consists of $|\hat{\mathcal{X}}|^{|\mathcal{X}|^{2k_f+1}}$ possible SB codes. Hence, in this case the search space grows double exponentially in k_f. Note that even for binary source and reconstruction alphabets and $k_f = 2$, the search space is gigantic ($2^{2^5} \approx 10^{9.6}$).

Similar to the case of block lossy compression, we resort to the simulated annealing and Gibbs sampling algorithms to approximate the minimizer of (10). Instead of the space of possible reconstruction sequences, here we define a probability distribution over the space of all possible SB codes. Each SB code is represented by a unique vector $\mathbf{f}$, and $p_\beta(\mathbf{f}) \propto \exp\left(-\beta \mathcal{E}(y^n)\right)$, where $y^n = y^n[x^n, \mathbf{f}]$. The conditional probabilities required at each step of the Gibbs sampler can be written as

$$p_\beta(f_i = \theta | f^{K_f \setminus i}) = \frac{p_\beta(f^{i-1}\theta f_{i+1}^{K_f})}{\sum_\vartheta p_\beta(f^{i-1}\vartheta f_{i+1}^{K_f})} = \frac{1}{\sum_\vartheta e^{-\beta(\mathcal{E}(f^{i-1}\vartheta f_{i+1}^{K_f}) - \mathcal{E}(f^{i-1}\theta f_{i+1}^{K_f}))}}. \tag{11}$$

Therefore, for computing the conditional probabilities we need to be able to compute the energy differences of the form $\mathcal{E}(f^{i-1}\vartheta f_{i+1}^{K_f}) - \mathcal{E}(f^{i-1}\theta f_{i+1}^{K_f})$, $(\vartheta, \theta) \in \hat{\mathcal{X}}^2$ efficiently. Compared to the case of block coding, described in Section IV, computation of these energy differences is somewhat more involved and requires more resourcefulness to be done efficiently. To this end, we first categorize different positions in x^n into $|\mathcal{X}|^{2k_f+1}$ different groups and construct the vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ such that $\alpha_i \triangleq \sum_{j=-k_f}^{k_f} r(x_{i+j})|\mathcal{X}|^{k_f+j}$. In other words, the label of each position is defined to be the symmetric context of length $2k_f + 1$ embracing it, i.e., $x_{i-k_f}^{i+k_f}$. Using this definition, applying a SB code f^{K_f} to a sequence x^n can alternatively be expressed as constructing a sequence y^n where $y_i = f_{\alpha_i}$.

From this representation, changing f_i from θ to ϑ while leaving the other elements of $\mathbf{f}$ unchanged only affects the positions of the y^n sequence that correspond to the label i in $\boldsymbol{\alpha}$. Hence, we can write

the energy difference in (11) as

$$\mathcal{E}(f^{i-1}\vartheta f_{i+1}^{K_f}) - \mathcal{E}(f^{i-1}\theta f_{i+1}^{K_f}) = H_k(y^n) - H_k(\hat{y}^n) + \alpha \sum_{j:\alpha_j=i} \frac{d(x_j,\vartheta) - d(x_j,\theta)}{n}, \tag{12}$$

where y^n and $\hat{y}^n$ represent the results of applying $f^{i-1}\vartheta f_{i+1}^{K_f}$ and $f^{i-1}\theta f_{i+1}^{K_f}$ to x^n, respectively. As noted before y^n and $\hat{y}^n$ differ only at the positions $\{j : \alpha_j = i\}$. Flipping each position in the y^n sequence in turn affects at most $2(k+1)$ columns of the count matrix $\mathbf{m}(y^n)$. Here at each pass of the Gibbs sampler a number of positions in the y^n sequence are flipped simultaneously. Alg. 2 describes how we can keep track of all these changes and update the count matrix. After that, in analogy to Alg. 1, Alg. 3 runs the Gibbs sampling method to find the best SB code of order $2k_f + 1$. At each iteration, Alg. 3 employs Alg. 2.

Algorithm 2 Updating the count matrix of $y^n = f(x^n)$, when f_i changes from θ to ϑ

Input: x^n, k_f, k, $\mathbf{m}(y^n)$, i, ϑ, θ
Output: $\mathbf{m}(\hat{y}^n)$

$a^n \leftarrow \mathbf{0}$
$\hat{y}^n \leftarrow y^n$
for $j = 1$ to n **do**
  if $\alpha_j = i$ **then**
    $\hat{y}_j \leftarrow \theta$
  end if
end for
$\mathbf{m}(\hat{y}^n) \leftarrow \mathbf{m}(y^n)$
for $j = k_f + 1$ to $n - k_f$ **do**
  if $\alpha_j = i$ **then**
    $a_j^{j+k} \leftarrow \mathbf{1}$
  end if
end for
for $j = k + 1$ to $n - k$ **do**
  if $a_j = 1$ **then**
    $m_{y_j, y_{j-k}^{j-1}} \leftarrow m_{y_j, y_{j-k}^{j-1}} - \frac{1}{n-k}$
    $m_{\hat{y}_j, \hat{y}_{j-k}^{j-1}} \leftarrow m_{\hat{y}_j, \hat{y}_{j-k}^{j-1}} + \frac{1}{n-k}$
  end if
end for

Let $\mathbf{f}_{\beta,\alpha,r} = f_{\beta,\alpha,r}^{K_f^{(n)}}$ denote the output of Alg. 3 to the input vector x^n at slope α after r iterations, and annealing process $\boldsymbol{\beta} = \{\beta_t\}_{t=1}^r$. $K_f^{(n)} = 2^{2k_f^{(n)}+1}$ denotes the length of the vector $\mathbf{f}$ representing the SB code. The following theorem states that Alg. 3 is asymptotically optimal for any stationary ergodic source, i.e., coding a source sequence by applying the SB code $\mathbf{f}_{\beta,\alpha,r}$ to the source sequence, and then

Algorithm 3 Universal SB lossy coder based on simulated annealing Gibbs sampler

Input: x^n, k_f, k, α, β, r
Output: f^{K_f}

1: **for** $t = 1$ to r **do**
2: Draw an integer $i \in \{1, \ldots, K_f\}$ uniformly at random
3: For each $\theta \in \hat{\mathcal{X}}$ compute $p_{\beta_t}(f_i = \theta | f^{K_f \setminus i})$ using Algorithm 2, equations (11), and (12)
4: Update f^{K_f} by replacing its i^{th} component f_i by θ drawn from the pmf computed in the previous step
5: **end for**

describing the output to the decoder using the Lempel-Ziv algorithm, asymptotically, as the number of iterations and window length k_f grow to infinity, achieves the rate-distortion curve.

Theorem 3: Consider a sequence $(k_f^{(n)})_n$ such that $k_f^{(n)}$ grows to infinity, $k_n = o(\log n)$, $k_n \to \infty$, and a cooling schedule $\beta_t^{(n)} = \frac{1}{T_0^{(n)}} \log(\lfloor \frac{t}{K_f^{(n)}} \rfloor + 1)$ for some $T_0^{(n)} > K_f^{(n)} \Delta$, where

$$\Delta = \max_i \max_{f^{i-1} \in \hat{\mathcal{X}}^{i-1}, f_{i+1}^n \in \hat{\mathcal{X}}^{K_f - i}, \vartheta, \theta \in \hat{\mathcal{X}}} |\mathcal{E}(f^{i-1} \vartheta f_{i+1}^{K_f}) - \mathcal{E}(f^{i-1} \theta f_{i+1}^{K_f})|, \tag{13}$$

and $K_f^{(n)} = 2^{2k_f^{(n)}+1}$. Then, for any stationary ergodic source $\mathbf{X}$, we have

$$\lim_{n\to\infty} \lim_{r\to\infty} \left[\frac{1}{n} \ell_{\mathsf{LZ}} \left(\hat{X}^n \right) + \alpha d_n(X^n, \hat{X}^n) \right] = \min_{D \geq 0} \left[R(D, \mathbf{X}) + \alpha D \right], \tag{14}$$

almost surely, where $\hat{X}^n$ is the result of applying SB code $f_{\beta,\alpha,r}^{K_f}$ to X^n.

Proof: First, we need to show that a result similar to Theorem 1 holds for SB codes. That is, we need to prove that for the given sequences $(k_f^{(n)})_n$ and $(k_n)_n$, finding a sequence of SB codes according to $\mathbf{f_o} = f_o^{K_f^{(n)}} \triangleq \arg\min_{f^{K_f^{(n)}}} \mathcal{E}(f^{K_f^{(n)}})$, where $\mathcal{E}(f^{K_f^{(n)}})$ is defined in (10), results in a sequence of asymptotically optimal codes for any stationary ergodic source $\mathbf{X}$ at slope α. In other words,

$$\lim_{n\to\infty} \left[\frac{1}{n} \ell_{\mathsf{LZ}}(\hat{X}^n) + \alpha d_n(\hat{X}^n, X^n) \right] = \min_{D \geq 0} \left[R(D, \mathbf{X}) + \alpha D \right], \tag{15}$$

almost surely, where $\hat{X}^n = \hat{X}^n[X^n, f_o^{K_f^{(n)}}]$. After proving this, the rest of the proof follows similar to the proof of Theorem 2. For establishing the equality stated in (15), we prove consistent lower and upper bounds, which in the limit yield the desired result. The lower bound,

$$\liminf_{n\to\infty} \left[\frac{1}{n} \ell_{\mathsf{LZ}}(\hat{X}^n) + \alpha d(X^n, \hat{X}^n) \right] \geq \min_{D \geq 0} \left[R(D, \mathbf{X}) + \alpha D \right], \tag{16}$$

follows from part (1) of Theorem 5 in [29]. For proving the upper bound, we split the cost into two

terms, as

$$\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n)+\alpha d(X^n,\hat{X}^n)\right]=\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n)-H_{k_n}(\hat{X}^n)\right]+\left[H_{k_n}(\hat{X}^n)+\alpha d(X^n,\hat{X}^n).\right]. \tag{17}$$

By Ziv's inequality, the first term on the RHS of (A-2) converges to zero as $n\to\infty$. Hence, we only need to upper bound the second term.

Since, asymptotically, for any stationary ergodic process $\mathbf{X}$, SB codes have the same rate-distortion performance as block codes, for a point $(R(D,\mathbf{X}),D)$ on the rate-distortion curve of the source, and any $\epsilon>0$, there exits a SB code $f^{2\kappa_f^\epsilon+1}$ of some order κ_f^ϵ such that coding the process $\mathbf{X}$ by this SB code results in a process $\tilde{\mathbf{X}}$ which satisfies i) $\bar{H}(\tilde{\mathbf{X}})\leq R(D,\mathbf{X})$, and ii) $\mathrm{E}[d(X_0,\tilde{X}_0)]\leq D+\epsilon$. On the other hand, for a fixed n, $\mathcal{E}(f_o^{K_f})$ is monotonically decreasing in K_f. Therefore, for any process $\mathbf{X}$ and any $\delta>0$, there exists n_δ such that for $n>n_\delta$ and $k_f^{(n)}\geq\kappa_f^\epsilon$

$$\limsup_{n\to\infty}\left[H_{k_n}(\hat{X}^n)+\alpha d_n(X^n,\hat{X}^n)\right]\leq R(D,\mathbf{X})+\alpha(D+\epsilon)+\delta,\quad \text{w.p. } 1. \tag{18}$$

Combining (16) and (18), plus the arbitrariness of ϵ, δ and D yield the desired result. ■

Note that in Algorithm 3, for a fixed k_f, the SB code is a vector of length $K_f=|\mathcal{X}|^{2k_f+1}$. Hence, the size of the search space, $|\hat{\mathcal{X}}|^{K_f}$, is independent of n. Moreover, the transition probabilities defined by (11) depend on the differences of the form presented in (12), which, for a stationary ergodic source and fixed k_f, if n is large enough, linearly scales with n. That is, for a given f^{i-1}, $f_{i+1}^{K_f}$, ϑ and θ,

$$\lim_{n\to\infty}\frac{1}{n}[\mathcal{E}(f^{i-1}\vartheta f_{i+1}^{K_f})-\mathcal{E}(f^{i-1}\theta f_{i+1}^{K_f})]=q, \tag{19}$$

almost surely, where $q\in[0,1]$ is some fixed value depending only on the source distribution. This is an immediate consequence of the ergodicity of the source plus the fact that SB coding of a stationary ergodic process results in another process which is jointly stationary and ergodic with the initial process. On the other hand, similar reasoning proves that Δ defined in (13) scales linearly with n. Therefore, overall, combining these two observations, for large values of n and fixed k_f, the transition probabilities of the non-homogeneous MC defined by the simulated annealing algorithm incorporated in Algorithm 3 are independent of n. This does not mean that the convergence rate of the algorithm is independent of n, because for achieving the rate-distortion function one needs to increase k_f and n simultaneously to infinity.

Remark 1: There is a slight difference between SB codes proposed in [37], and our entropy-constrained SB codes. In [37], it is assumed that after the encoder converts the source process into the coded process,

with no more encryption, it can be directly sent to the decoder via a channel that has capacity of R bits per transmission. Then the decoder, using another SB code, converts the coded process into the reconstruction process. In our setup on the other hand, the encoder directly converts the source process into the reconstruction process, which has lower entropy, and then employs a universal lossless coder to describe the coded sequence to the decoder. The decoder then applies the corresponding universal lossless decoder to retrieve the reconstruction sequence.

VI. Simulation results

In this section we present some experimental results. Sections VI-A and VI-B demonstrate the performance of the proposed algorithms on simulated 1-D and real 2-D data, for block and sliding-block codes, respectively. Section VI-C studies the effects of different parameters on the algorithm's performance.

Since, as described before, $H_k(\cdot)$ is not a length function, in the 1-D simulations, instead of the conditional empirical entropy function, we use the k^{th} order arithmetic codeword length to measure the rate. For a binary sequence y^n, its k^{th} order arithmetic codeword length is defined as

$$L_k(y^n) \triangleq -\log \prod_{b^k \in \{0,1\}^k} \frac{n_{0,b^k}(y^n)! n_{1,b^k}(y^n)!}{\big(n_{0,b^k}(y^n) + n_{1,b^k}(y^n) + 1\big)!}, \tag{20}$$

where, for $\beta \in \{0,1\}$ and $b^k \in \{0,1\}^k$, n_{β,b^k} counts the number of occurrences of the $(k+1)$-tuple $[b^k, \beta]$ in y^n, i.e., $n_{\beta,b^k} \triangleq (n-k)m_{\beta,b^k}$.

A. Block coding

In this section, some of the simulation results obtained by applying Alg. 1 of Section IV to real and simulated data are presented. The algorithm is easy to apply, as is, to both 1-D and 2-D data .

As the first example, consider a $\mathrm{Bern}(p)$ i.i.d source. Fig. 1 and Fig. 2 compare the optimal rate-distortion tradeoff against the rate-distortion performance of Alg. 1 for $p = 0.5$ and $p = 0.1$, respectively. In Fig. 1 the coding parameters are $n = 15 \times 10^3$, $k = 8$, $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, where $\gamma = 0.9$, $r = 10n$, and $\alpha = 4 : -0.5 : 1$. Each point corresponds to the average performance over $N = 50$ simulations, i.e., $N = 50$ input sequences. At each simulation, the algorithm starts from $\alpha = 4$, and gradually decreases the coefficient by 0.5 at each step. Moreover, except for $\alpha = 4$ where $\hat{x}^n$ is initialized by x^n, for other values of α, the algorithm is initialized by the quantized sequence found at the previous step. In Fig. 2, $k = 10$, $n = 5 \times 10^4$, $\alpha = (5, 4.5, 4, 3.5, 2)$, and r, β_t and γ are as before.

As another example, Fig. 3 shows the performance of Alg. 1 when applied to a binary symmetric Markov source (BSMS) with transition probability $p = 0.25$. Here the parameters are: $n = 2 \times 10^4$,

$k = 8$, $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, $\gamma = 0.8$, $r = 10n$ and $\alpha = 5 : -0.5 : 3$. The figure also shows the performance of the heuristic lossy compression algorithm proposed in [29]. The parameters are: $n = 2 \times 10^4$, $k = 8$, and $M = 20$ paths.

The Shannon lower bound, which is shown in the same figure, states that for a BSMS with transition probability p, $R(D) \geq R_{\text{SLB}}(D) \triangleq h(p) - h(D)$. There is no known explicit characterization of the rate-distortion tradeoff for a BSMS except for very low distortions. It has been proven that for $D < D_c$, where $D_c = 0.5 - 0.5\sqrt{1 - p^2/(1-p)^2}$, the SLB holds with equality. For $D > D_c$, a strict inequality holds, and $R(D) > R_{\text{SLB}}$ [41]. In our case where $p = 0.25$, $D_c = 0.0286$. For distortions beyond D_c, an upper bound on the rate-distortion function, derived based on the results of [6], is shown for comparison.

To illustrate the encoding process, Fig. 4 depicts the evolutions of $H_k(\hat{x}^n)$, $d_n(x^n, \hat{x}^n)$, and $\mathcal{E}(\hat{x}^n) = H_k(\hat{x}^n) + \alpha d_n(x^n, \hat{x}^n)$ during coding iterations. It can be observed that, as time proceeds, while the complexity of the sequence has an overall decreasing trend, as expected, its distance with the original sequence increases. Here the source is binary Markov with transition probability $p = 0.2$. The algorithm parameters are $n = 10^4$, $k = 7$, $\alpha = 4$, $r = 10n$, and $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, where $\gamma = 0.8$.

Finally, consider applying the algorithm to the $n \times n$ binary image shown in Fig. 5 , where $n = 252$. Let $N \triangleq n^2$ denote the total number of pixels in the image. Fig. 6(a) and Fig. 6(b) show the coded version after $r = 50N$ iterations for $\alpha = 0.1$ and $\alpha = 3.3$ respectively. The algorithm's cooling process is $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$ with $\gamma = 0.99$. Fig. 7 shows the 2-D context used for constructing the count matrix of the image that is used by the algorithm. In the figure, the solid black square represents the location of the current pixel, and the other marked squares denote its 6^{th} order causal context that are taken into account.

Fig. 6(a), the empirical conditional entropy of the image has decreased from 0.1025 to 0.0600 in the reconstruction image, while an average distortion of $D = 0.0337$ per pixel is introduced. Comparing the required space for storing the original image as a PNG file with the amount required for the coded image reveals that in fact the algorithm not only has reduced the conditional empirical entropy of the image by 41.5%, but also has cut the size of the file by around 39%. Fig. 8 shows the size of the compressed image in terms of the size of the original image when α varies as $0.1 : 0.4 : 3.3$.

B. Sliding block coding

Consider applying Alg. 3 of Section V to the output of a BSMS with $p = 0.2$. Fig. 9 shows the algorithm output along with Shannon lower bound and lower/upper bounds on $R(D)$ from [6]. Here the parameters are: $n = 5 \times 10^4$, $k = 8$, $\beta_t = K_f \alpha \log(t+1)$ and $r = 10K_f$. Red squares correspond to SB

window length of $2k_f+1=9$, and slope values $\alpha = 5.8, 5.6, 5.5, 5.4, 5.2, 5, 4.8$. Blue circles correspond to window length of $2k_f+1=11$, and slope values $\alpha = 5.4, 5.3, 5.25, 5.2, 5.1, 5, 4.7$.

In all of the presented simulation results, it is the empirical conditional entropy of the final reconstruction block that we are comparing to the rate-distortion curve. It should be noted that, though this difference vanishes as the block size grows, for finite values of n there would be an extra (model) cost for losslessly describing the reconstruction block to the decoder.

C. Discussion on the choice of different parameters

1) Context length k and block length n: As stated in Theorem 1, in order to achieve the optimal performance, we need k to grow with n to infinity as $o(\log n)$. For a given block length n, in order to get a good performance, it is crucial to choose k appropriately. Note that the order k determines the order of the count matrix $\mathbf{m}$ which is used to measure the *complexity* of the quantized sequence. Choosing k to be too big or too small compared to the length of our sequence are both problematic. If k is too small, then the count matrix $\mathbf{m}$ will not capture all useful structures existing in the sequence. These structures potentially help the universal lossy coder to describe the sequence with fewer number of bits. On the other hand, if k is too large compared to the block length n, then $H_k(y^n)$ gives an unreliable underestimate of the complexity of the sequence. In that case, while $H_k(\hat{x}^n)$ might be small suggesting that $\hat{x}^n$ can be described efficiently, an actual lossy coder might need a much higher rate to describe it. Hence, we need to make a balanced choice between the two extremes.

In order to demonstrate this interaction between k and n, Fig. 10 shows the average performance of Alg. 1 when applied to a BSMS with transition probability $p=0.25$. Here the block length is $n=2\times10^4$, and each point corresponds to the average performance over 50 simulations. The performances are shown for $k=6$, $k=7$ and $k=8$. At each simulation, a sequence of length n is generated, and is coded by Alg. 1 using the three different values of k. For each value of k, and each simulated sequence, the algorithm starts from $\alpha=5$ and step by step decreases α to $\alpha=3$, while using the quantized sequence of the previous step as an initial point, except for $\alpha=5$. At $\alpha=5$, the algorithm is initialized at x^n. The cooling schedule is fixed to $\beta_t=(1/\gamma)^{\lceil t/n\rceil}$ with $\gamma=0.8$.

Fig. 10(a) shows the final cost in terms of $H_k(\hat{x}^n)+\alpha d_n(x^n,\hat{x}^n)$, for different values of k, and Fig. 10(b) shows the real cost which is measured as $L_k(\hat{x}^n)+\alpha d_n(x^n,\hat{x}^n)$. It can be observed that while increasing the context length k almost always reduces the minimized cost $H_k(\hat{x}^n)+\alpha d_n(x^n,\hat{x}^n)$, it uniformly increases the real cost in all cases examined. This confirms that, for a fixed block length, increasing the context length does not necessarily help the performance. In other words, while using

larger values of k might result in the reduction of the achieved conditional empirical entropy, but, unless it is done carefully, it can in fact deteriorate the overall performance.

2) Cooling schedule $\{\beta_t\}$: In all of our simulations the cooling schedule follows the generic form of $\beta_t = \beta_0(1/\gamma)^{\lceil t/n \rceil}$, for some $\gamma < 1$, but usually > 0.7. This is a common schedule used in simulated annealing literature. By this scheme, the running time is divided into intervals of length n, and the temperature remains constant during each interval, and decreases by a factor γ in the next interval. Hence larger values of γ correspond to slower cooling procedures. The specific values of γ and β_0 can be chosen based on the signal to be coded.

3) Number of iterations r: Although we have not yet derived a convergence rate for Alg. 1, from our simulations results, we suspect that for large classes of natural signals, $r = mn$ iterations, where $m = O(\log n)$, is enough for deriving a reasonable approximation of the solution to the exhaustive search algorithm. However, we do not expect a similar result to hold for all signals, and there might exist sources for which the convergence rate of simulated annealing is too slow.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, two new implementable block and SB lossy compression algorithms are proposed. The algorithms are based on simulated annealing and Gibbs sampling, and are shown to be capable of getting arbitrarily closely to the rate-distortion curve of any stationary ergodic source. For coding a source sequence x^n, the block coding algorithm starts from some initial reconstruction block, and updates one of its coordinates at each iteration. The algorithm can be viewed as a process of systematically introducing 'noise' into the original source block, but in a biased direction that results in a decrease of its description complexity.

In practice, the proposed algorithms, in their present form, are only applicable to the cases where the size of the reconstruction alphabet, $|\hat{\mathcal{X}}|$, is small. The reason is twofold: first, for larger alphabet sizes the contexts will be too sparse to make the empirical entropy a useful measure of compressibility, even for small values of k. Second, the size of the count matrix $\mathbf{m}$ grows exponentially with $|\hat{\mathcal{X}}|$ which makes storing it for large values of $|\hat{\mathcal{X}}|$ impractical. Despite these facts, there are practical applications where this constraint is satisfied. An example is lossy compression of binary images, like the one presented in Section VI. Another application for lossy compression of binary data is shown in [42] where one needs to compress a stream of 0 and 1 bits with some distortion. Moreover, surprisingly, in lossy compression of continuous sources, as shown in [43], in many cases the optimal reconstruction alphabet is discrete and has small cardinality. In those cases, the approach proposed here can be applied and in fact has

already been explored and shown to be effective in [44].

The convergence rate of the new algorithms and the effect of different parameters on it is a topic for further study. As an example, one might wonder how the convergence rate of the algorithm is affected by choosing an initial point other than the source output block itself. Although our theoretical results on universal asymptotic optimality remain intact for any initial starting point, in practice the choice of the starting point might significantly impact the number of iterations required.

Finally, note that in the non-universal setup, where the optimal achievable rate-distortion tradeoff is known in advance, this extra information can be used as a stopping criterion for the algorithm. For example, we can set it to stop after reaching optimum performance to within some fixed distance.

APPENDIX A: PROOF OF THEOREM 1

From part (1) of Theorem 5 in [29],

$$\liminf_{n\to\infty}\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n)+\alpha d(X^n,\hat{X}^n)\right]\geq\min_{D\geq 0}\left[R(D,\mathbf{X})+\alpha D\right], \tag{A-1}$$

almost surely. (A-1) states that the probability that a sequence of codes asymptotically outperforms the fundamental rate-distortion limit is zero.

In order to establish the upper bound, we split the cost function into two terms as

$$\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n)+\alpha d(X^n,\hat{X}^n)\right]=\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n)-H_{k_n}(\hat{X}^n)\right]+\left[H_{k_n}(\hat{X}^n)+\alpha d(X^n,\hat{X}^n)\right]. \tag{A-2}$$

From [35], for $k_n=o(\log n)$ and any given $\epsilon>0$, there exists $N_\epsilon\in\mathbb{N}$ such that for *any* individual infinite-length sequence $\hat{\mathbf{x}}=(\hat{x}_1,\hat{x}_2,\ldots)$ and any $n\geq N_\epsilon$,

$$\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{x}^n)-H_{k_n}(\hat{x}^n)\right]\leq\epsilon. \tag{A-3}$$

Hence, the first term on the right hand side of (A-2) can be made arbitrary small.

Now consider an arbitrary point $(R(D,\mathbf{X}),D)$ on the rate-distortion curve corresponding to source $\mathbf{X}$. For any $\delta>0$, there exists a process $\tilde{\mathbf{X}}$ such that $(\mathbf{X},\tilde{\mathbf{X}})$ are jointly stationary ergodic such that [7]

1) $\bar{H}(\tilde{\mathbf{X}})\leq R(D,\mathbf{X})$,
2) $\mathrm{E}[d(X_0,\tilde{X}_0)]\leq D+\delta$.

On the other hand, since for each source block X^n, the reconstruction block $\hat{X}^n$ minimizes $H_k(\hat{x}^n)+\alpha d(X^n,\hat{x}^n)$, it follows that

$$H_{k_n}(\hat{X}^n)+\alpha d(X^n,\hat{X}^n)\leq H_{k_n}(\tilde{X}^n)+\alpha d(X^n,\tilde{X}^n). \tag{A-4}$$

For a fixed k, from the definition of the k^{th} order entropy, we have

$$H_k(\tilde{X}^n) = \sum_{u^k \in \hat{\mathcal{X}}^k} \|\mathbf{m}_{\cdot,u^k}(\tilde{X}^n)\|_1 \mathcal{H}\left(\mathbf{m}_{\cdot,u^k}(\tilde{X}^n)\right). \tag{A-5}$$

But, since $\tilde{\mathbf{X}}$ is a stationary ergodic process,

$$\begin{aligned}\mathbf{m}_{u_{k+1},u^k}(\tilde{X}^n) &= \frac{1}{n-k}\sum_{i=k+1}^{n} \mathbb{1}_{\tilde{X}_{i-k}^{i}=u^{k+1}} \\ &\stackrel{n\to\infty}{\longrightarrow} \mathrm{P}\left(\tilde{X}_{-k}^{0} = u^{k+1}\right),\end{aligned} \tag{A-6}$$

almost surely. Therefore, combining (A-5) and (A-6), as n goes to infinity, $H_k(\tilde{X}^n)$ converges to $H(\tilde{X}_0|\tilde{X}_{-k}^{-1})$ with probability one. From the monotonicity of $H_k(\hat{x}^n)$ in k, (A-4), and the convergence we just established, it follows that, for any $\hat{x}^n$ and any k,

$$H_{k_n}(\hat{X}^n) + \alpha d(X^n, \hat{X}^n) \le H(\tilde{X}_0|\tilde{X}_{-k}^{-1}) + \epsilon + \alpha d(X^n, \tilde{X}^n), \quad \text{eventually a.s.} \tag{A-7}$$

On the other hand, since $\mathbf{X}$ and $\tilde{\mathbf{X}}$ are jointly stationary and ergodic,

$$d(\tilde{X}^n, X^n) = \frac{1}{n}\sum_{i=1}^{n} d(X_i, \tilde{X}_i) \stackrel{n\to\infty}{\longrightarrow} \mathrm{E}[d(\tilde{X}_0, X_0)] \le D + \delta, \quad \text{a.s.} \tag{A-8}$$

Combining (A-2), (A-3), (A-7), and (A-8) yields

$$\limsup_{n\to\infty}\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n) + \alpha d(X^n, \hat{X}^n)\right] \le H(\tilde{X}_0|\tilde{X}_{-k}^{-1}) + 2\epsilon + \alpha(D+\delta) \ \text{ a.s.} \tag{A-9}$$

The arbitrariness of k, ϵ and δ, and the fact that $\bar{H}(\tilde{\mathbf{X}}) = \lim_{k\to\infty} H(\tilde{X}_{k+1}|\tilde{X}^k) \le R(D, \mathbf{X})$ implies

$$\limsup_{n\to\infty}\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n) + \alpha d(X^n, \hat{X}^n)\right] \le R(D, \mathbf{X}) + \alpha D, \ \text{ a.s.} \tag{A-10}$$

Since the point $(R(D, \mathbf{X}), D)$ was chosen arbitrarily, it follows that

$$\limsup_{n\to\infty}\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n) + \alpha d(X^n, \hat{X}^n)\right] \le \min_{D\ge 0}[R(D, \mathbf{X}) + \alpha D]. \tag{A-11}$$

Finally, combining (A-1), and (A-11) we get the desired result:

$$\lim_{n\to\infty}\left[\frac{1}{n}\ell_{\mathsf{LZ}}(\hat{X}^n) + \alpha d(X^n, \hat{X}^n)\right] = \min_{D\ge 0}[R(D, \mathbf{X}) + \alpha D]. \tag{A-12}$$

REFERENCES

[1] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27:379–423 and 623–656, 1948.

[2] R.G. Gallager. *Information Theory and Reliable Communication*. NY: John Wiley, 1968.

[3] T. Berger. *Rate-distortion theory: A mathematical basis for data compression*. NJ: Prentice-Hall, 1971.

[4] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *Information Theory, IEEE Transactions on*, 24(5):530–536, Sep. 1978.

[5] I. H. Witten, R. M. Neal, , and J. G. Cleary. Arithmetic coding for data compression. *Commun. Assoc. Comp. Mach.*, 30(6):520–540, 1987.

[6] S. Jalali and T. Weissman. New bounds on the rate-distortion function of a binary Markov source. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 571–575, June 2007.

[7] R. Gray, D. Neuhoff, and J. Omura. Process definitions of distortion-rate functions and source coding theorems. *Information Theory, IEEE Transactions on*, 21(5):524–532, Sep. 1975.

[8] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[9] V. Cerny. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, Jan. 1985.

[10] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 6:721–741, Nov. 1984.

[11] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86(11):2210–2239, Nov 1998.

[12] J. Vaisey and A. Gersho. Simulated annealing and codebook design. In *Proceedings ICASSP'88*, pages 1176–1179, Apr. 1988.

[13] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *Communications, IEEE Transactions on*, 28(1):84–95, Jan. 1980.

[14] D. J. Sakrison. The rate of a class of random processes. *Information Theory, IEEE Transactions on*, 16:10–16, Jan. 1970.

[15] J. Ziv. Coding of sources with unknown statisticspart ii: Distortion relative to a fidelity criterion. *Information Theory, IEEE Transactions on*, 18:389–394, May 1972.

[16] D. L. Neuhoff, R. M. Gray, and L.D. Davisson. Fixed rate universal block source coding with a fidelity criterion. *Information Theory, IEEE Transactions on*, 21:511–523, May 1972.

[17] D. L. Neuhoff and P. L. Shields. Fixed-rate universal codes for Markov sources. *Information Theory, IEEE Transactions on*, 24:360–367, May 1978.

[18] J. Ziv. Distortion-rate theory for individual sequences. *Information Theory, IEEE Transactions on*, 24:137–143, Jan. 1980.

[19] R. Garcia-Munoz and D. L. Neuhoff. Strong universal source coding subject to a rate-distortion constraint. *Information Theory, IEEE Transactions on*, 28:285–295, Mar. 1982.

[20] K. Cheung and V. K. Wei. A locally adaptive source coding scheme. *Proc. Bilkent Conf on New Trends in Communication, Control, and Signal Processing*, pages 1473–1482, 1990.

[21] J.L. Bentley, D.D. Sleator, R.E. Tarjan, and V.K. Wei. A locally adaptive data compression algorithm. *Communications of the ACM*, 29(4):320–330, Apr. 1986.

[22] H. Morita and K. Kobayashi. An extension of LZW coding algorithm to source coding subject to a fidelity criterion. In *Proc. 4th Joint Swedish-Soviet Int. Workshop on Information Theory*, pages 105–109, Gotland, Sweden, 1989.

[23] Y. Steinberg and M. Gutman. An algorithm for source coding subject to a fidelity criterion based on string matching. *Information Theory, IEEE Transactions on*, 39:877–886, Mar. 1993.

[24] E.H. Yang and J.C. Kieffer. On the performance of data compression algorithms based upon string matching. *Information Theory, IEEE Transactions on*, 44(1):47–65, Jan. 1998.

[25] T. Luczak and T. Szpankowski. A suboptimal lossy data compression based on approximate pattern matching. *Information Theory, IEEE Transactions on*, 43:1439–1451, Sep. 1997.

[26] R. Zamir and K. Rose. Natural type selection in adaptive lossy compression. *Information Theory, IEEE Transactions on*, 47(1):99–111, Jan. 2001.

[27] I. Kontoyiannis. An implementable lossy version of the Lempel-Ziv algorithm-part i: optimality for memoryless sources. *Information Theory, IEEE Transactions on*, 45(7):2293–2305, Nov. 1999.

[28] Zhen Zhang and V.K. Wei. An on-line universal lossy data compression algorithm via continuous codebook refinement. i. basic results. *Information Theory, IEEE Transactions on*, 42(3):803–821, May 1996.

[29] E.H. Yang, Z. Zhang, and T. Berger. Fixed-slope universal lossy data compression. *Information Theory, IEEE Transactions on*, 43(5):1465–1476, Sep. 1997.

[30] M.J. Wainwright and E. Maneva. Lossy source encoding via message-passing and decimation over generalized codewords of ldgm codes. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 1493–1497, Sep. 2005.

[31] A. Gupta and S. Verdú. Nonlinear sparse-graph codes for lossy compression of discrete non-redundant sources. In *Information Theory Workshop*, pages 541–546, Sep. 2007.

[32] J. Rissanen and I. Tabus. Rate-distortion without random codebooks. In *Workshop on Information Theory and Applications (ITA)*, Sep. 2006.

[33] A. Gupta, S. Verdú, and T. Weissman. Linear-time near-optimal lossy compression. In *Proc. IEEE Int. Symp. Inform. Theory*, 2008.

[34] Z. Sun, M. Shao, J. Chen, K. Wong, and X. Wu. Achieving the rate-distortion bound with low-density generator matrix codes. *Communications, IEEE Transactions on*, 58(6):1643 –1653, June 2010.

[35] E. Plotnik, M.J. Weinberger, and J. Ziv. Upper bounds on the probability of sequences emitted by finite-state sources and on the redundancy of the Lempel-Ziv algorithm. *Information Theory, IEEE Transactions on*, 38(1):66–72, Jan. 1992.

[36] P. Bremaud. *Markov chains, Gibbs fields, Monte Carlo simulation, and queues*. Springer, New York, 1991.

[37] R. M. Gray, D. L. Neuhoff, and D. S. Ornstein. Nonblock source coding with a fidelity criterion. *The Annals of Probability*, 3(3):478–491, June 1975.

[38] K. Marton. On the rate distortion function of stationary sources. *Probl. Contr. Inform. Theory*, 4:289–297, 1975.

[39] R. M. Gray. Block, sliding-block, and trellis codes. In *Janos Bolyai Colloquiem on Info. Theory*, Keszthely, Hungary, Aug. 1975.

[40] S. Mallat. *A Wavelet tour of signal processing*. Academic Press, Boston, 1997.

[41] R. Gray. Rate distortion functions for finite-state finite-alphabet Markov sources. *Information Theory, IEEE Transactions on*, 17(2):127–134, Mar. 1971.

[42] G. Motta, E. Ordentlich, and M.J. Weinberger. Defect list compression. In *Proc. IEEE Int. Symp. Inform. Theory*, pages 1000–1004, July 2008.

[43] K. Rose. A mapping approach to rate-distortion computation and analysis. *Information Theory, IEEE Transactions on*, 40(6):1939–1952, Nov. 1994.

[44] D. Baron and T. Weissman. An MCMC approach to lossy compression of continuous sources. In *Proc. Data Compression Conference*, pages 40–48, Mar. 2010.

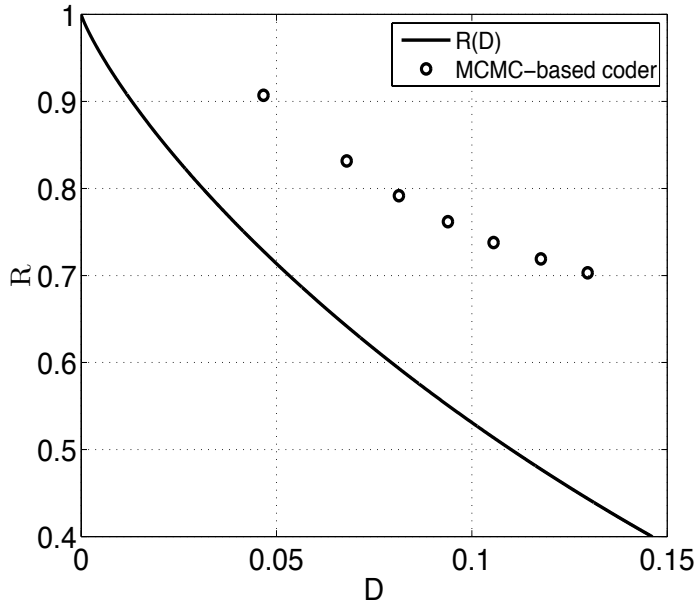


Fig. 1. Comparing the performance of Alg. 1 with the optimal rate-distortion tradeoff for an i.i.d. $\mathrm{Bern}(p)$ source. ($p = 0.5$, $n = 15 \times 10^3$, $k = 8$, $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, $\gamma = 0.9$, $r = 10n$ and $\alpha = 4 : -0.5 : 1$).

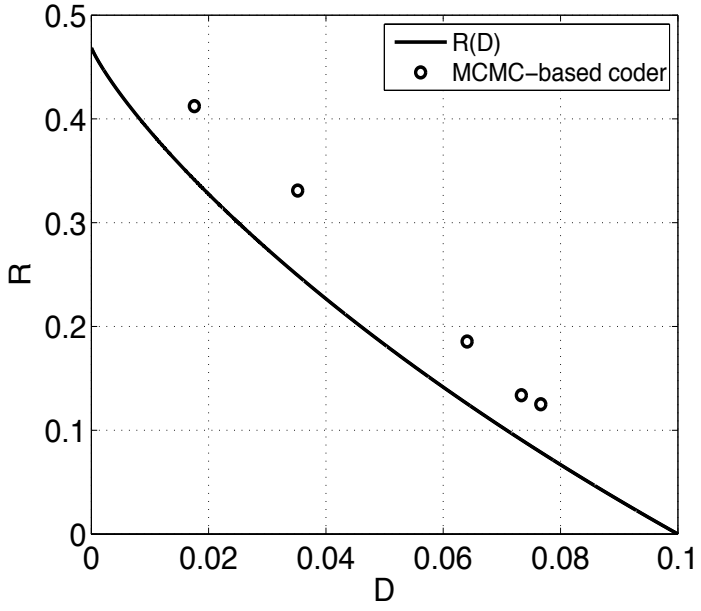


Fig. 2. Comparing the performance of Alg. 1 with the optimal rate-distortion tradeoff for an i.i.d. $\mathrm{Bern}(p)$ source. ($p = 0.1$, $n = 5 \times 10^4$, $k = 10$, $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, $\gamma = 0.9$, $r = 10n$ and $\alpha = (5, 4.5, 4, 3.5, 2)$).

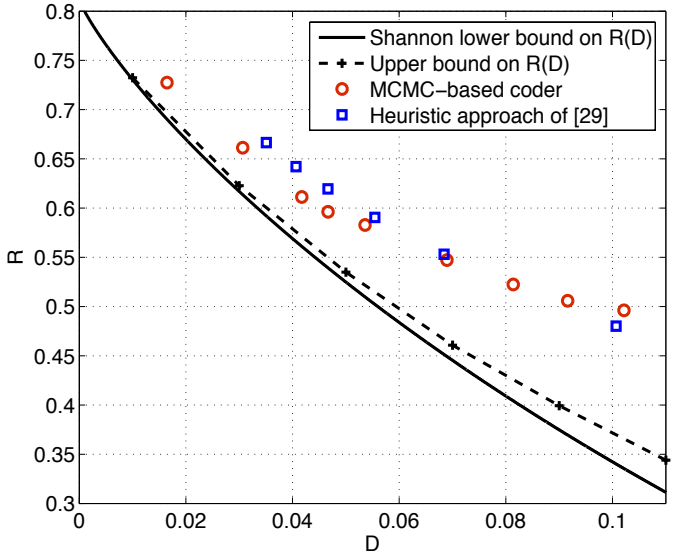


Fig. 3. Comparing the rate-distortion performance of Alg. 1 with the heuristic approach proposed in [29], when both algorithms are applied to a BSMS(p) ($p = 0.25$, $n = 2 \times 10^4$, $k = 8$, $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, $\gamma = 0.8$, $r = 10n$ and $\alpha = 5 : -0.5 : 1$)

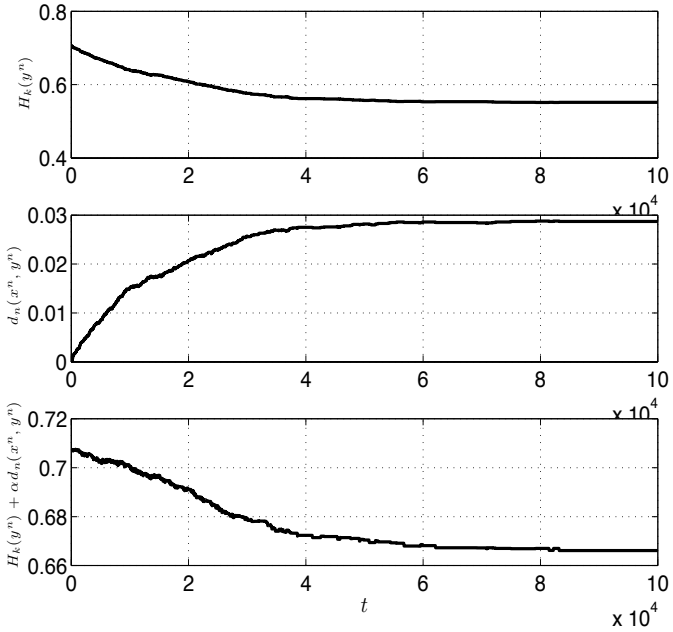


Fig. 4. Sample paths demonstrating evolutions of the empirical conditional entropy, average distortion, and energy function when Alg. 1 is applied to the output of a BSMS source, with $p = 0.2$ ($n = 10^4$, $k = 7$, $\alpha = 4$, $r = 10n$, and $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, where $\gamma = 0.8$).

Fig. 5. Original image with empirical conditional entropy of 0.1025

(a) Reconstruction image with empirical conditional entropy of 0.0600 and average distortion of 0.0337 per pixel ($\alpha = 0.1$).

(b) Reconstruction image with empirical conditional entropy of 0.0824 and average distortion of 0.0034 per pixel ($\alpha = 3.3$).

Fig. 6. Applying Alg. 1 to a 2-D binary image ($\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, where $\gamma = 0.99$, $r = 50N$)

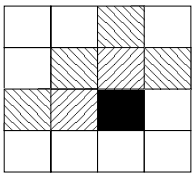


Fig. 7. The 6^{th} order context used in coding of 2-D images

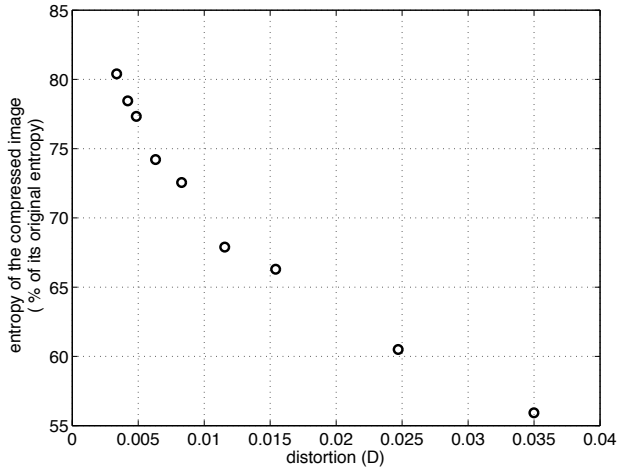


Fig. 8. Size of the compressed image in terms of the entropy of the original image (in percentage) versus distortion ($\alpha = 0.1 : 0.4 : 3.3$, $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, where $\gamma = 0.99$, $r = 50N$).

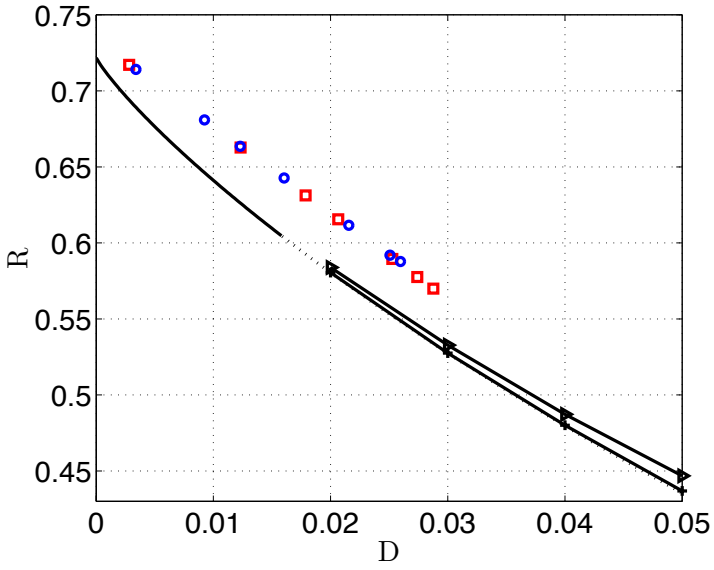
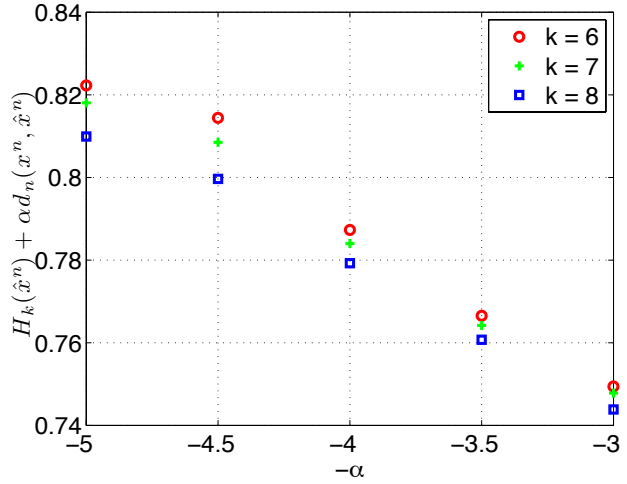
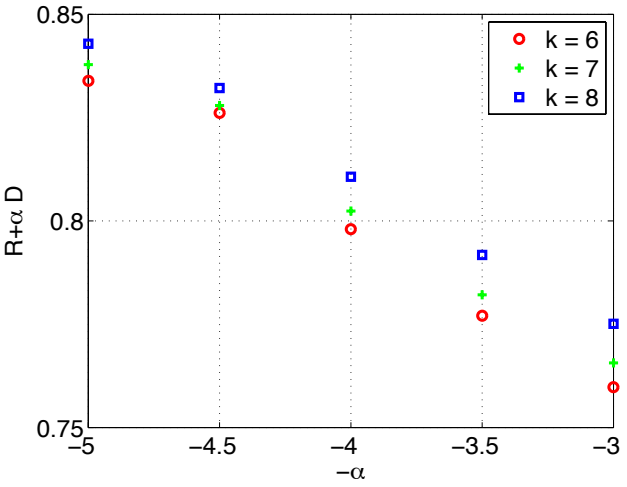


Fig. 9. Comparing the algorithm rate-distortion performance with the Shannon lower bound for a BSMS(p). Red squares and blue circles correspond to $k_f = 4$ ($K_f = 2^9$) and $k_f = 5$ ($K_f = 2^{11}$), respectively. ($p = 0.2$, $n = 5 \times 10^4$, $k = 8$, and $\beta_t = K_f \alpha \log(t+1)$.)



(a)



(b)

Fig. 10. Effect of parameter k on the performance of Alg. 1 while coding a BSMS(p). ($p = 0.25$, $\beta_t = (1/\gamma)^{\lceil t/n \rceil}$, where $\gamma = 0.8$, $n = 2 \times 10^4$ and $r = 10n$).