Source Coding with Fixed Lag Side Information

Emin Martinian and Gregory W. Wornell
Massachusetts Institute of Technology
Cambridge, MA 02139
{emin,gww}@allegro.mit.edu

Abstract

We consider source coding with fixed lag side information at the decoder. We focus on the special case of perfect side information with unit lag corresponding to source coding with feedforward (the dual of channel coding with feedback) introduced by Pradhan [8]. We use this duality to develop a linear complexity algorithm which achieves the rate-distortion bound for any memoryless finite alphabet source and distortion measure.

1 Introduction

There is a growing consensus that understanding complex, distributed systems requires a combination of ideas from communication and control [1]. Adding communication constraints to traditional control problems or adding real-time constraints to communication problems has recently yielded interesting results [2–6]. We consider a related aspect of this interaction by exploring the possible advantages that the feedback/feedforward in control scenarios can provide in compression. Specifically, we explore a variant of the Wyner-Ziv problem [7] where causal side information about the source is available with a fixed lag to the decoder and explore how such side information may be used.

For example, consider a remote sensor that sends its observations to a controller as illustrated in Fig. 1. The sensor may be a satellite or aircraft reporting the upcoming temperature, wind speed, or other weather data to a vehicle. The sensor observations must be encoded via lossy compression to conserve power or bandwidth. In contrast to the standard lossy compression scenario, however, the controller directly observes the original, uncompressed data after some delay. The goal of the sensor observations is to provide the controller with information about upcoming events *before* they occur. Thus

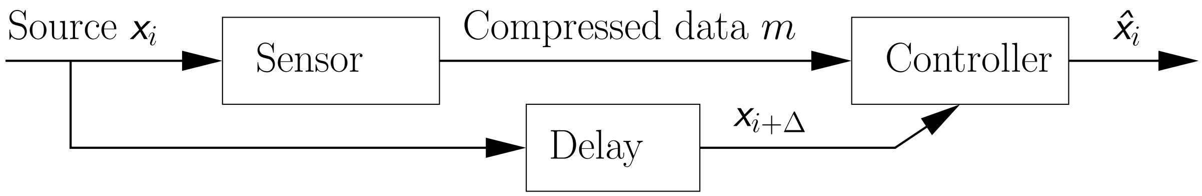

Figure 1: A sensor compresses and sends the source sequence x_1, x_2, ..., to a controller which reconstructs the quantized sequence $\hat{x}_1$, $\hat{x}_2$, ..., in order to take some control action. After a delay or lag of Δ, the controller observes the original, uncompressed data directly.

at first it might not seem that observing the true, uncompressed data *after* they occur would be useful. Our main goal is to try to understand how these delayed observations of the source data (which we call side information) can be used. Our main result is that such information can be quite valuable.

The following toy problem helps illustrate some relevant issues. Imagine that Alice plays a game where she will be asked 10 Yes/No questions. Of these questions, 5 have major prizes while the others have minor prizes. After answering each question, she is told the correct answer as well as what the prize for that question is and receives the prize if she is correct. Bob knows all the questions and the corresponding prizes beforehand and wishes to help Alice by preparing a "cheat-sheet" for her. But Bob only has room to record 5 answers. Is there a cheat-sheet encoding strategy that guarantees that Alice will always correctly answer the questions with the 5 best prizes? No such strategy exists using a classical compression scheme. Instead, as illustrated in Section 3, the optimal strategy requires an encoding which uses the fact that Alice gains information about the prize *after* answering.

In the rest of the paper, we study the fixed lag side information problem. Since solving the general problem seems difficult, we begin by focusing on perfect side information with a unit lag. This special case is the feedforward source coding problem introduced by Pradhan [8] and is dual (in the sense of [9, 10]) to channel coding with feedback. By using the feedforward side information, it is possible to construct low complexity source coding systems which can achieve the rate-distortion bound. Specifically, [8] describes how to adapt the Kailath-Schalkwijk scheme for the Gaussian channel with feedback [11] to the Gaussian source squared distortion scenario with feedforward side information. In this paper, we consider finite alphabet sources with arbitrary memoryless distributions and arbitrary distortion measures. Since Ooi and Wornell's channel coding with feedback scheme [12] achieves the best error exponent with minimum complexity, we investigate the source coding dual. Specifically, we show that the source coding dual of the Ooi–Wornell scheme achieves the rate-distortion bound with linear complexity.

We begin by describing the problem in Section 2. Next we present a simple example of how feedforward side information can be useful in the binary erasure quantization problem in Section 3. In Section 4, we consider the more complicated example of quantizing a binary source with respect to Hamming distortion. We present our source coding algorithm for general sources in Section 5 and show that it achieves the rate-distortion function with low complexity. We close with some concluding remarks in Section 6.

2 Problem Description

Random variables are denoted using the sans serif font (*e.g.*, x) with deterministic values using serif fonts (*e.g.*, x). We represent the ith element of a sequence as x_i and denote a subsequence including elements from i to j as x_i^j.

We consider (memoryless) source coding with fixed lag side information and represent an instance of the problem with the tuple $(\mathcal{X}, \mathcal{W}, p_{\mathsf{x},\mathsf{w}}, d(\cdot,\cdot), \Delta)$ where $\mathcal{X}$ and $\mathcal{W}$ represent the source and side information alphabets, $p_{\mathsf{x},\mathsf{w}}(x,w)$ represents the source and side information joint distribution, $d(\cdot,\cdot)$ represents the distortion measure, and Δ represents the delay or lag. Specifically, the source and side information each consist of a sequence of n random variables x_1^n and w_1^n taking values in $\mathcal{X}$ and $\mathcal{W}$ generated according to the distribution $p_{\mathsf{x}_1^n,\mathsf{w}_1^n}(x_1^n) = \prod_{i=1}^n p_{\mathsf{x},\mathsf{w}}(x_i, w_i)$.

A rate R encoder, $f(\cdot)$, maps x_1^n to a bit sequence represented as an integer $m \in$

$\{1, 2, \ldots, 2^{nR}\}$. The corresponding decoder $f^{-1}(\cdot)$ works as follows. At time i, the decoder takes as input m as well as the side information samples, $w_1^{i-\Delta}$, and produces the ith reconstruction $\hat{x}_i$. A distortion of $d(x_i, \hat{x}_i)$ is then incurred for the ith sample where $d(\cdot, \cdot)$ is a mapping from $\mathcal{X} \times \mathcal{X}$ to the interval $[0, d_{\max}]$.

The basic problem can be specialized to the original (non-causal) Wyner-Ziv problem [7], by allowing a negative delay $\Delta = -\infty$. Similarly, setting $\Delta = 0$ yields a causal version of the Wyner-Ziv problem. Finally, letting the side information be exactly the same as the source with a positive delay yields the feedforward source coding problem studied in [8]. For all these cases, the goal is to understand the fundamental rate-distortion-complexity performance. To show that the benefits of fixed lag side information are worth investigating, we focus on the feedforward case where $w = x$ with unit delay $\Delta = 1$ throughout the rest of this paper.

For memoryless sources, the information feedforward rate-distortion function, $R_{\mathrm{f}}^{(I)}(D)$, is defined to be the same as Shannon's classical rate-distortion function:

$$R_{\mathrm{f}}^{(I)}(D) = \inf_{p_{\hat{x}|x}: E[d(x,\hat{x})] \leq D} I(\hat{x}; x). \tag{1}$$

The operational feedforward rate-distortion function, $R_{\mathrm{f}}(D)$, is the minimum rate required such that there exists a sequence of encoders and decoders with average distortion, $\frac{1}{n}\sum_{i=1}^{n} d(x_i, \hat{x}_i)$, asymptotically approaching D. As observed in [8] and shown in the appendix, the information and operational feedforward rate-distortion functions are the same. Thus feedforward does not reduce the rate required, but as we argue in the rest of this paper, it allows us to approach the rate-distortion function with low complexity.

3 Example: Binary Source & Erasure Distortion

The simplest example in channel coding with feedback is the erasure channel and in this case the algorithm in Fig. 2 achieves capacity. At time 1 the encoder puts message bit m_1 into the channel. If it is received correctly, the transmitter then transmits m_2, otherwise m_1 is repeated until it is successfully received. The same process is used for m_2, m_3, etc. For example, to send the message 0101 though a channel where samples 2, 3, 6, and 7 are erased, the transmitter would send 01110111 and the receiver would see $0 * *10 * *1$. In general, if there are n message bits, m_1, m_2, ..., m_n, and e erasures, then exactly $n + e$ channel uses are required. This yields a transmission rate of $n/(n + e)$ which is exactly the channel capacity.

The dual to the binary erasure channel (BEC) is the binary erasure quantization problem (BEQ). In the BEQ, each source sample can be either 0, 1, or * where * represents "don't care". The distortion measure is such that 0 and 1 cannot be changed but * can be quantized to either 0 or 1 with no distortion. The BEQ models the game introduced in the introduction.[1] To develop a source coding with feedforward algorithm for the BEQ, we can dualize the channel coding with feedback algorithm for the BEC as illustrated in Fig. 3.

Assume the source is $x_1^8 = 0 * *10 * *1$. The encoder compresses this to $m = 0101$ by ignoring all the * symbols and sends this to the receiver. At time 1, the receiver chooses $\hat{x}_1$

[1] Yes/No answers for questions with major prizes map to 1/0 values for the source while questions with minor prizes map to a value of * for the source. The distortion measure represents the restriction that questions with major prizes must be answered correctly while the answers for the other questions are irrelevant.

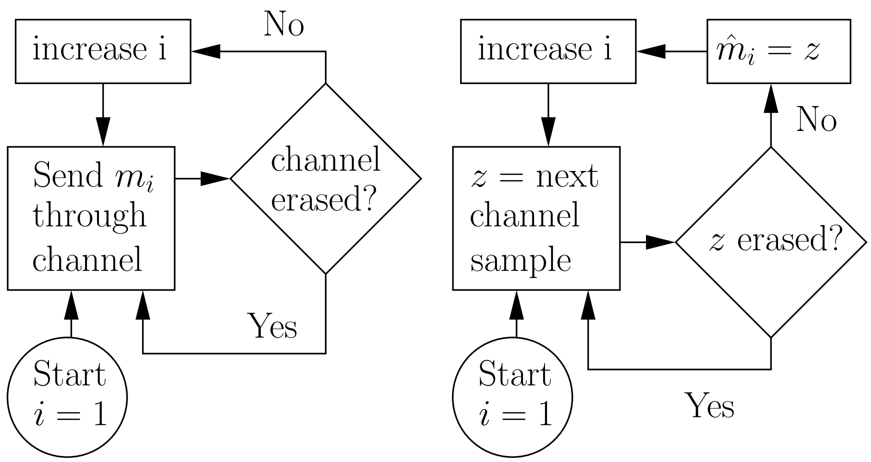

Figure 2: Encoder (left) for transmitting a message $m = m_1, m_2, \ldots$ across an erasure channel with feedback and decoder (right) for producing an estimate of the transmitted message $\hat{m}$.

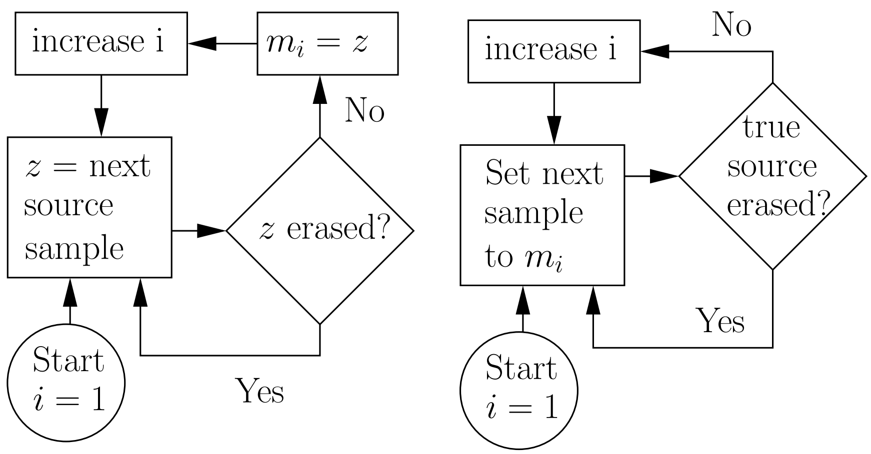

Figure 3: Encoder (left) and decoder (right) for the binary erasure quantization problem which is dual to the binary erasure channel.

to be the first bit in the encoding (i.e., $\hat{x}_1 = m_1 = 0$). From the feed-forward the receiver realizes this is correct after it makes its choice. Next, the receiver chooses $\hat{x}_2$ to be the next bit received (i.e., $\hat{x}_2 = m_2 = 1$). After choosing this reconstruction, the receiver is told that in fact $x_2 = *$ so even though $x_2 \neq \hat{x}_2$, no distortion is incurred. At this point, the receiver realizes that m_2 must have been intended to describe something after x_2. So at time 3, the receiver chooses $\hat{x}_3 = m_2 = 1$. Once again the receiver learns that this is incorrect since $x_3 = *$, but again no penalty is incurred. Again, the receiver decides that m_2 must have been intended to describe something else so it chooses $\hat{x}_4 = m_2 = 1$ at time 4. This turns out to be correct and so at time 5 the receiver chooses $\hat{x}_5 = m_3$, etc.

In the encoder/decoder described above, the encoder sends the non-erased bits of x_1^n and the decoder tries to match up the compressed data to the source. This system yields distortion 0 provided that at least $n - e$ bits are sent where e denotes the number of * symbols in the source vector. It is straightforward to show that no encoder/decoder can do better for any value of n or e. A system not taking advantage of the feedforward could asymptotically achieve the same performance but it would require more complexity and more redundancy. Thus just as in the erasure channel with feedback, we see that for the erasure source, feedforward allows us to achieve the minimum possible redundancy with minimum complexity.

4 Example: Binary Source & Hamming Distortion

The example in Section 3 illustrates that feedforward can be useful in source coding by using some special properties of the BEQ problem. Next, we consider a somewhat more complicated example to illustrate that a key idea in developing *lossy* compression algorithms for source coding with feedforward is the use of classical *lossless* compression algorithms. Specifically, we consider a binary source which is equally likely to be either zero or one, and we consider the Hamming distortion measure $d(x, \hat{x}) = |x - \hat{x}|$. As is well known, the rate-distortion function for this case is $R(D) = 1 - H_b(D)$ where $H_b(\cdot)$ is the binary entropy function. In the following we outline a scheme which achieves a distortion of $D_0 = 0.11$ and rate $R(D_0) = 1 - H_b(0.11) \approx 1/2$.

Let $\mathcal{C}(\cdot)$ and $\mathcal{C}^{-1}(\cdot)$ be a lossless compression and decompression algorithm for a Bernoulli(D_0) source. Specifically, $\mathcal{C}(\cdot)$ takes as input t bits with a fraction D_0 ones and maps them into $H_b(D_0)t \approx t/2$ uniformly distributed bits while $\mathcal{C}^{-1}(\cdot)$ maps t' approximately uniformly distributed bits into $t'/H_b(D_0) \approx 2t'$ bits with a fraction D_0 ones. To simplify the exposition, we assume that for $t' \geq \mathtt{M}$, these approximations are exact. A more careful treatment appears in Section 5.

The feedforward lossy compression system encoder takes a sequence of source samples, x_1^n, where $n = \mathtt{M}(2^K - 1)$ and encodes by producing the following codewords:

$$b_1 \triangleq x_{n-\mathtt{M}+1}^{n} \tag{2}$$

$$b_2 \triangleq x_{n-3\mathtt{M}+1}^{n-\mathtt{M}} \oplus \mathcal{C}^{-1}(b_1) = x_{n-3\mathtt{M}+1}^{n-\mathtt{M}} \oplus \mathcal{C}^{-1}\left(x_{n-\mathtt{M}+1}^{n}\right) \tag{3}$$

$$b_3 \triangleq x_{n-7\mathtt{M}+1}^{n-3\mathtt{M}} \oplus \mathcal{C}^{-1}(b_2) = x_{n-7\mathtt{M}+1}^{n-3\mathtt{M}} \oplus \mathcal{C}^{-1}\left(x_{n-3\mathtt{M}+1}^{n-\mathtt{M}} \oplus \mathcal{C}^{-1}(b_1)\right) \tag{4}$$

$$= x_{n-7\mathtt{M}+1}^{n-3\mathtt{M}} \oplus \mathcal{C}^{-1}\left(x_{n-3\mathtt{M}+1}^{n-\mathtt{M}} \oplus \mathcal{C}^{-1}\left(x_{n-\mathtt{M}+1}^{n}\right)\right) \tag{5}$$

$$\vdots \qquad\qquad \vdots \tag{6}$$

$$b_K \triangleq x_1^{\mathtt{M}2^{K-1}} \oplus \mathcal{C}^{-1}(b_{K-1}) = x_1^{\mathtt{M}2^{K-1}} \oplus \mathcal{C}^{-1}\left(x_{2^{K-1}\cdot\mathtt{M}+1}^{3\mathtt{M}2^{K-1}} \oplus \mathcal{C}^{-1}(b_{K-2})\right) \tag{7}$$

$$= x_1^{\mathtt{M}2^{K-1}} \oplus \mathcal{C}^{-1}\left(x_{2^{K-1}\cdot\mathtt{M}+1}^{3\mathtt{M}2^{K-1}} \oplus \mathcal{C}^{-1}\left(x_{3\mathtt{M}2^{K-1}+1}^{7\mathtt{M}2^{K-2}} \oplus \ldots \oplus \mathcal{C}^{-1}\left(x_{n-\mathtt{M}+1}^{n}\right)\right)\right) \tag{8}$$

according to the general rule

$$b_i \triangleq x_{n-(2^i-1)\mathtt{M}+1}^{n-(2^{i-1}-1)\mathtt{M}} \oplus \mathcal{C}^{-1}(b_{i-1}). \tag{9}$$

The output of the encoder is the $\mathtt{M} \cdot 2^{K-1}$ bit sequence b_K for the last block.

As we see from (7), b_K is a description of the first block of source samples corrupted by the addition of $\mathcal{C}^{-1}(b_{K-1})$. The decoder reconstructs the first $\mathtt{M} \cdot 2^{K-1}$ source samples via

$$\hat{x}_1^{\mathtt{M}2^{K-1}} \triangleq b_K = x_1^{\mathtt{M}2^{K-1}} \oplus \mathcal{C}^{-1}(b_{K-1}). \tag{10}$$

The distortion for this block is approximately D_0 since, by assumption, the decompresser $\mathcal{C}^{-1}(\cdot)$ maps its input to a sequence with a fraction D_0 ones. The error between the reconstruction and the true source obtained via feedforward is a description of future source samples shaped by the function $\mathcal{C}(\cdot)$. Thus, to reconstruct the next block, the decoder uses the feedforward, $x_1^{\mathtt{M}2^{K-1}}$, to produce

$$\hat{x}_{\mathtt{M}2^{K-1}+1}^{3\mathtt{M}2^{K-2}} \triangleq \mathcal{C}\left(x_1^{\mathtt{M}2^{K-1}} \oplus b_K\right) = b_{K-1} = x_{\mathtt{M}2^{K-1}+1}^{3\mathtt{M}2^{K-2}} \oplus \mathcal{C}^{-1}(b_{K-2}). \tag{11}$$

Once again the distortion is approximately D_0 since the decompresser maps its input to a sequence with D_0 ones.

The decoder proceeds in this manner and obtains a distortion of approximately D_0 for each block except the last which yields no distortion. The average distortion is therefore roughly D_0. Since $\mathtt{M}2^{K-1}$ bits are required to describe b_K in encoding the $\mathtt{M}(2^K-1)$ source samples, the average bit rate is $2^{K-1}/(2^K-1) \approx 1/2$. Thus by taking advantage of the source feedforward, we can obtain a point on the rate distortion curve simply by using a low complexity lossless compression algorithm.

5 Finite Alphabet Sources & Arbitrary Distortion

In this section, we generalize the construction in Section 4 to arbitrary rates, source distributions and distortion measures. We require two components: a lossless compression/decompression algorithm and a shaping algorithm. Using these subsystems, we describe our feedforward source coding algorithm and present an analysis of its rate and distortion.

5.1 Feedforward Source Coding Subsystems

Our lossless compression and shaping algorithms must be efficient in some sense for the overall feedforward source coding algorithm to approach the rate-distortion function. Instead of delving into the details of how to build efficient compression and shaping algorithms, we define admissible systems to illustrate the required properties. We then describe how efficient subsystems can be combined.

5.1.1 Lossless Compression Subsystem

We define a (δ, ϵ, m) admissible lossless compression system as follows. On input of m samples from which are δ-strongly typical[2] according to the distribution $p_{\hat{\mathsf{x}}}$, the compressor, denoted $\mathcal{C}_{\hat{\mathsf{x}}}(\cdot)$, returns $m \cdot H(\hat{\mathsf{x}}) + \epsilon$ bits. If the input is not δ-strongly typical, the output is undefined. The corresponding decompresser, $\mathcal{C}_{\hat{\mathsf{x}}}^{-1}(\cdot)$ takes the resulting bits and reproduces the original input.

5.1.2 Shaping Subsystem

We define a (δ, ϵ, m) admissible shaping system as follows. On input of a sequence of m bits, and a semi-infinite sequence of samples, x_1^∞ which is δ-strongly typical according to the distribution p_{x}, the shaper $\mathcal{S}_{\hat{\mathsf{x}}|\mathsf{x}}(\cdot)$ returns a sequence of $m' = m \cdot [H(\hat{\mathsf{x}})/H(\hat{\mathsf{x}}|\mathsf{x})] + \epsilon$ samples, $\hat{\mathsf{x}}_1^{m'}$, such that $(\mathsf{x}_1^{m'}, \hat{\mathsf{x}}_1^{m'})$ is δ-strongly typical according to the distribution $p_{\hat{\mathsf{x}},\mathsf{x}}$. If the input is not δ-strongly typical, the output is undefined. The corresponding deshaper takes the pair of sequences $\hat{\mathsf{x}}_1^{m'}$ and $\mathsf{x}_1^{m'}$ as input and returns the original sequence of m bits.

The compression and shaping systems described previously are fixed-to-variable and variable-to-fixed systems respectively. Hence, for notational convenience we define the corresponding length functions $\mathcal{L}(\mathcal{C}_{\hat{\mathsf{x}}}(\cdot))$ and $\mathcal{L}\left(\mathcal{S}_{\hat{\mathsf{x}}|\mathsf{x}}(\cdot)\right)$ as returning the length of their respective arguments.

[2]A sequence is δ-strongly typical if the empirical fraction of occurrences of each possible outcome differs by at most δ from the expected fraction of outcomes and no probability zero outcomes occur.

5.1.3 Efficient Shaping and Compression Systems

We call a lossless compression system or a shaping system efficient if both δ and ϵ can be made arbitrarily small for m large enough. Efficient lossless compression systems can be implemented in a variety of ways. For example, arithmetic coding is one well-known approach. Perhaps less well-known is that shaping systems can also be implemented via arithmetic coding [12]. Specifically, by using the decompresser for an arithmetic code as a shaper, we can map a sequence of bits into a sequence with an arbitrary distribution. The compressor for the arithmetic code takes the resulting sequence and returns the original bit sequence.

5.2 Feedforward Encoder and Decoder

Since the encoder for our feedforward lossy compression system is based on a variable-to-fixed shaper and a fixed-to-variable compressor, it is a variable-to-variable system. In practice, one could use buffering, padding, or other techniques to account for this when encoding a fixed length source or when required to produce a fixed length encoding. We do not address this issue further here. Instead, we assume that there is a nominal source block size parameter, $\mathtt{N}$, and buffering, padding, look-ahead, *etc.* is used to ensure that the system encodes $\mathtt{N}$ source samples (or possibly slightly more or less). Also, we assume that there is a minimum block size parameter, $\mathtt{M}$, which may be chosen to achieve an efficient shaping or lossless compression subsystem.

Once $\mathtt{N}$ and $\mathtt{M}$ are fixed, the feedforward encoder takes as input a stream of inputs x_1^∞ and encodes it as described in Table 1. The feedforward decoder takes as input the resulting bit string, b, and decodes it as described in Table 2. Section 4 describes an example of the encoding and decoding algorithm with a shaper (denoted $\mathcal{C}^{-1}(\cdot)$) mapping uniform bits to Bernoulli(0.11) bits. This example does not require a compressor because the $p_{\hat{\mathsf{x}}}$ distribution is incompressible.

Table 1: The Feedforward Encoder.

Initialize $T = 1$, $L = \mathtt{M}$, and reverse the input so that in the following $\mathsf{x}_1^n = \mathsf{x}_n^1$.
Take the block of source samples x_T^{T+L} and generate a "noisy version" $\hat{\mathsf{x}}_T^{T+L}$ (*e.g.*, by generating each $\hat{\mathsf{x}}_i$ from the corresponding x_i according to $p_{\hat{\mathsf{x}}|\mathsf{x}}$).
while $L + T < \mathtt{N}$ **do**
    Compress $\hat{\mathsf{x}}_T^{T+L}$ to obtain the bit sequence $b = \mathcal{C}_{\hat{\mathsf{x}}}\left(\hat{\mathsf{x}}_T^{T+L}\right)$.
    $T \leftarrow T + L + 1$
    $L \leftarrow \mathcal{L}\left(\mathcal{S}_{\hat{\mathsf{x}}|\mathsf{x}}\left(b, \mathsf{x}_T^\infty\right)\right)$.
    $\hat{\mathsf{x}}_T^{T+L} \leftarrow \mathcal{S}_{\hat{\mathsf{x}}|\mathsf{x}}\left(b, \mathsf{x}_T^\infty\right)$
end while
return $\mathcal{C}_{\hat{\mathsf{x}}}\left(\hat{\mathsf{x}}_T^{T+L}\right)$

5.3 Rate-Distortion Analysis

Theorem 1. *By using efficient lossless compression and shaping subsystems, the distortion in encoding an i.i.d. sequence generated according to p_{x} can be made to approach $E[d(\mathsf{x}, \hat{\mathsf{x}})]$ as closely as desired.*

Proof. First we note that by assumption, we can choose $\mathtt{M}$ large enough so that the probability of the source sequence being non-typical can be made negligible. For a typical

Table 2: The Feedforward Decoder.

Initialize T to the length of the sequence encoded in b.
while $T > 1$ **do**
    $L \leftarrow \mathcal{L}\left(\mathcal{C}_{\hat{x}}^{-1}(b)\right)$
    $T \leftarrow T - L + 1$
    $\hat{x}_T^{T+L} \leftarrow \mathcal{C}_{\hat{x}}^{-1}(b)$
    Get x_T^{T+L} via the feedforward information
    $b \leftarrow \mathcal{S}_{\hat{x}|x}^{-1}\left(\hat{x}_T^{T+L}, x_T^{T+L}\right)$
end while
return the reversed version of $\hat{x}_1^n$

source sequence, we can focus on how the encoder maps x_i to $\hat{x}_i$ since the decoder simply maps a bit sequence to the $\hat{x}_i$ sequence chosen by the encoder. The encoder maps blocks of source samples, x_T^{T+L}, to blocks of quantized samples, $\hat{x}_T^{T+L}$, by using an admissible shaping algorithm. As described in Section 5.1.2, the shaper produces a δ-strongly typical sequence. Thus the total expected distortion is at most

$$E[d(x, \hat{x})] + d_{\max} \cdot \delta + d_{\max} \cdot \Pr[x_1^n \text{ not typical}] \tag{12}$$

where the first two terms are the distortion for a typical sequence produced by the shaper and the last term is the contribution from a non-typical source sequence. □

Theorem 2. *By using efficient lossless compression and shaping subsystems, the rate in encoding an i.i.d. sequence generated according to p_x can be made to approach $I(x; \hat{x})$ as closely as desired.*

Proof. Imagine that the parameter $\mathtt{N}$ is chosen so that K passes of the loop in the encoding algorithm are executed. Also, let L_j denote the value of L in line 3 of the encoder in the jth pass. We know $L_1 = \mathtt{M}$ by construction. By definition of an admissible shaping system in Section 5.1.2 and line 6 of the encoder we have that $L_{j+1} \geq L_j \cdot [H(\hat{x})/H(\hat{x}|x)]$. Using this relation and assuming that each block of length L_j is typical, we can compute the total number of samples encoded via

$$n = \sum_{j=1}^{K} L_j \geq \sum_{j=0}^{K-1} \mathtt{M} \cdot \left[\frac{H(\hat{x})}{H(\hat{x}|x)}\right]^j = \mathtt{M} \cdot \frac{[H(\hat{x})/H(\hat{x}|x)]^K - 1}{H(\hat{x})/H(\hat{x}|x) - 1}. \tag{13}$$

The bit rate required to encode these samples is

$$R = L_K \cdot H(\hat{x}) + \epsilon \leq \mathtt{M} \cdot H(\hat{x}) \cdot [H(\hat{x})/H(\hat{x}|x)]^{K-1} + \epsilon \cdot K \cdot [H(\hat{x})/H(\hat{x}|x)]^K. \tag{14}$$

This follows by the assumption that the admissible lossless compression system in Section 5.1.1 requires $m \cdot H(\hat{x}) + \epsilon$ bits to encode a block of m typical samples.

Therefore the number of bits per sample when the source blocks are typical is obtained by dividing (14) by (13) to obtain

$$R/n \leq \left\{ \mathtt{M} \cdot H(\hat{x}) \cdot \left[\frac{H(\hat{x})}{H(\hat{x}|x)}\right]^{K-1} + \epsilon \cdot K \cdot \left[\frac{H(\hat{x})}{H(\hat{x}|x)}\right]^{K} \right\} \Big/ \left\{ \mathtt{M} \cdot \frac{[H(\hat{x})/H(\hat{x}|x)]^K - 1}{H(\hat{x})/H(\hat{x}|x) - 1} \right\} \tag{15}$$

$$= \left\{ H(\hat{x}) \left[1 - \frac{H(\hat{x}|x)}{H(\hat{x})}\right] + \frac{\epsilon K}{\mathtt{M}} \left[\frac{H(\hat{x})}{H(\hat{x}|x)} - 1\right] \right\} \Big/ \left\{ 1 - \left[\frac{H(\hat{x})}{H(\hat{x}|x)}\right]^{-K} \right\} \tag{16}$$

$$= I(\hat{x}; x) \cdot \left\{ 1 + \frac{\epsilon K}{\mathtt{M} H(\hat{x}|x)} \right\} \Big/ \left\{ 1 - \left[\frac{H(\hat{x})}{H(\hat{x}|x)}\right]^{-K} \right\}. \tag{17}$$

An extra term must also be added to account for the possibility that the source is atypical. By assumption we can choose $\mathtt{N}$ so that K is large enough to make the second term in braces negligible, and then we can choose $\mathtt{M}$ so that the probability of any source block being typical is negligible. Also, by making $\mathtt{M}$ large enough we can make the first term in curly braces negligible. Thus the bit rate can be made as close to $I(\hat{x}; x)$ as desired. □

Combining the previous theorems indicates that we can approach the feedforward rate-distortion function with only the complexity required for lossless compression and shaping systems.

Corollary 1. *When linear complexity admissible lossless compression and shaping systems are used, the resulting feedforward rate-distortion function can be approached arbitrarily closely with linear complexity.*

In particular, we can use the lossless compression and shaping systems described in [12] which are based on arithmetic coding and the dual of arithmetic coding respectively.

6 Concluding Remarks

In this paper we describe a lossy compression algorithm to encode a finite-alphabet source in the presence of feedforward information. In particular, we show that although memoryless feedforward does not change the rate-distortion function, it allows us to construct a low complexity lossy compression system which approaches the rate-distortion function. In practice, the particular scheme described here may require modifications and other methods of using feedforward information or similar knowledge may be more appropriate. Our main goal therefore is not necessarily to advocate a particular scheme but to show that when compression, observation, and control interact, additional resources such as feedforward may provide advantages not available in the classic compression framework. One interesting possibility for future work includes studying the general problem in Section 2 when the fixed lag side information, w, is not exactly the same as the source. Similarly, investigating the effects of memory in the source and different values for the delay, Δ, would also be valuable.

A Information/Operational R(D) Equivalence

Proposition 1. *The information/operational feedforward rate-distortion functions are equal.*

Proof. Since the decoder must deterministically produce $\hat{x}_i$ from x_1^{i-1} and the nR bits produced by the encoder we have

$$nR \geq \sum_{i=1}^{n} H(\hat{x}_i|x_1^{i-1}) \geq \sum_{i=1}^{n} \left[H(\hat{x}_i|x_1^{i-1}) - H(\hat{x}_i|x_1^{i})\right] = \sum_{i=1}^{n} \left[H(x_i|x_1^{i-1}) - H(x_i|x_1^{i-1}, \hat{x}_i)\right]$$

$$\stackrel{(a)}{=} \sum_{i=1}^{n} \left[H(x_i) - H(x_i|x_1^{i-1}, \hat{x}_i)\right] \stackrel{(b)}{\geq} \sum_{i=1}^{n} \left[H(x_i) - H(x_i|\hat{x}_i)\right]$$

where (a) follows since the source is memoryless and (b) follows since conditioning reduces entropy. From this point standard convexity arguments establish that (1) is a lower bound to the average rate. □

References

[1] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, "Future directions in control in an information-rich world," *IEEE Control Systems Magazine*, vol. 23, pp. 20–33, April 2003.

[2] V. S. Borkar, S. K. Mitter, and S. Tatikonda, "Optimal sequential vector quantization of Markov sources," *SIAM J. Control Optim.*, vol. 40, no. 1, pp. 135–148, 2001.

[3] N. Elia and S. K. Mitter, "Stabilization of linear systems with limited information," *IEEE Transactions on Automatic Control*, vol. 46, pp. 1384–1400, September 2001.

[4] A. Sahai, *Any-time Information Theory.* PhD thesis, Massachusetts Institute of Technology, September 2000.

[5] S. Tatikonda, *Control Under Communication Constraints.* PhD thesis, Massachusetts Institute of Technology, September 2000.

[6] W. S. Wong and R. W. Brockett, "Systems with finite communication bandwidth constraints. I. state estimation problems," *IEEE Transactions on Automatic Control*, vol. 42, pp. 1294–1299, September 1997.

[7] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 1–10, January 1976.

[8] S. S. Pradhan, "Source coding with feedforward: Gaussian sources," in *Proc. International Symposium on Information Theory*, (Chicago, IL), 2004.

[9] S. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Transactions on Information Theory*, vol. 49, pp. 1181–1203, May 2003.

[10] R. J. Barron, B. Chen, and G. W. Wornell, "The duality between information embedding and source coding with side information and some applications," *IEEE Transactions on Information Theory*, vol. 49, pp. 1159–1180, May 2003.

[11] J. P. M. Schalkwijk, "A coding scheme for additive noise channels with feedback–II: Band-limited signals," *IEEE Transactions on Information Theory*, vol. 12, pp. 183–189, April 1966.

[12] J. M. Ooi and G. W. Wornell, "Fast iterative coding techniques for feedback channels," *IEEE Transactions on Information Theory*, vol. 44, pp. 2960–2976, November 1998.