Lossy Compression via Sparse Linear Regression: Computationally Efficient Encoding and Decoding

Ramji Venkataramanan
Dept. of Engineering, Univ. of Cambridge
ramji.v@eng.cam.ac.uk

Tuhin Sarkar
Dept. of EE, IIT Bombay
tuhin91@gmail.com

Sekhar Tatikonda
Dept. of EE, Yale University
sekhar.tatikonda@yale.edu

***Abstract*—We propose computationally efficient encoders and decoders for lossy compression using a Sparse Regression Code. Codewords are structured linear combinations of columns of a design matrix. The proposed encoding algorithm sequentially chooses columns of the design matrix to successively approximate the source sequence. It is shown to achieve the optimal distortion-rate function for i.i.d Gaussian sources with squared-error distortion. For a given rate, the parameters of the design matrix can be varied to trade off distortion performance with encoding complexity. An example of such a trade-off is: computational resource (space or time) per source sample of $O((n/\log n)^2)$ and probability of excess distortion decaying exponentially in $n/\log n$, where n is the block length. The Sparse Regression Code is robust in the following sense: for any ergodic source, the proposed encoder achieves the optimal distortion-rate function of an i.i.d Gaussian source with the same variance. Simulations show that the encoder has very good empirical performance, especially at low and moderate rates.**

I. Introduction

Developing efficient codes for lossy compression at rates approaching the Shannon rate-distortion limit has long been an important goal of information theory. Efficiency is measured in terms of the storage complexity of the codebook as well the computational complexity of encoding and decoding. The Shannon-style i.i.d random codebook achieves the optimal distortion-rate trade-off but its storage and computational complexities grow exponentially with the block length. In this paper, we study a class of codes called Sparse Superposition or Sparse Regression Codes (SPARCs) for lossy compression with a squared-error distortion criterion. We present computationally efficient encoding and decoding algorithms that attain the optimal rate-distortion function for i.i.d Gaussian sources.

Sparse Regression codes were recently introduced by Barron and Joseph for communication over the AWGN channel and shown to approach the Shannon capacity with feasible decoding [1], [2], [3]. The codebook construction is based the statistical framework of high-dimensional linear regression. The codewords are sparse linear combinations of columns of an $n \times N$ design matrix or 'dictionary', where n is the block-length and N is a low-order polynomial in n. This structure enables the design of computationally efficient compression encoders based on sparse approximation ideas (e.g., [4], [5]). We propose one such encoder and analyze it performance.

SPARCs for lossy compression were first considered in [6] where some preliminary results were presented. The rate-distortion and error exponent performance of these codes under minimum-distance (optimal) encoding was characterized in [7]. The main contributions of this paper are the following.

- We propose a computationally efficient encoding algorithm for SPARCs which achieves the optimal distortion-rate function for i.i.d Gaussian sources with growing block length n. The algorithm is based on successive approximation of the source sequence by columns of the design matrix. The parameters of the design matrix can be chosen to trade off performance with complexity. For example, one choice of parameters discussed in Section IV yields a $n \times O(n^2)$ design matrix, per-sample encoding complexity proportional to $(\frac{n}{\log n})^2$, and probability of excess distortion decaying exponentially in $\frac{n}{\log n}$. To the best of our knowledge, this is the fastest known rate of decay among lossy compression codes with feasible encoding and decoding.
- With the proposed encoder, SPARCs share the following robustness property of random i.i.d Gaussian codebooks [8], [9]: for a given rate R, any ergodic source with variance σ^2 can be compressed with distortion close to the i.i.d Gaussian distortion-rate function $\sigma^2 e^{-2R}$.

We briefly review related work in developing computationally efficient codes for lossy compression. It was shown in [10] that the optimal rate-distortion function of memoryless sources can be approached by concatenating optimal codes over sub-blocks of length much smaller than the overall block length. Nearest neighbor encoding is used over each of these sub-blocks, which is feasible due to their short length. For this scheme, it is not known how rapidly the probability of excess distortion decays to zero with the overall block length. For sources with finite alphabet, various coding techniques have been proposed recently to approach the rate-distortion bound with computationally feasible encoding and decoding, e.g. [11], [12], [13]. The rates of decay of the probability of excess distortion for these schemes vary, but in general they are slower than exponential in the block length.

The survey paper by Gray and Neuhoff [14] contains an extensive discussion of various compression techniques and their performance versus complexity trade-offs. These include scalar quantization with entropy coding, tree-structured vector quantization, multi-stage vector quantization, and trellis-coded quantization. Though these techniques have good empirical performance, they have not been proven to attain the op-

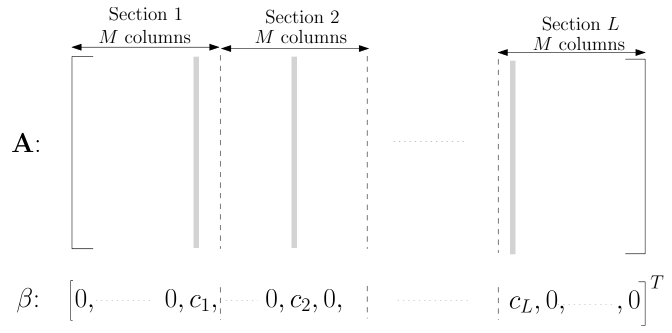

Fig. 1. $\mathbf{A}$ is an $n \times ML$ matrix and β is a $ML \times 1$ vector. The positions of the non-zeros in β correspond to the gray columns of $\mathbf{A}$ which combine to form the codeword $\mathbf{A}\beta$.

timal rate-distortion trade-off with computationally feasible encoders and decoders.

Notation: Upper-case letters are used to denote random variables, lower-case for their realizations, and bold-face letters for random vectors and matrices. All vectors have length n. The source sequence is denoted by $\mathbf{S} \triangleq (S_1, \ldots, S_n)$, and the reconstruction sequence by $\hat{\mathbf{S}} \triangleq (\hat{S}_1, \ldots, \hat{S}_n)$. $\|\mathbf{X}\|$ is the ℓ_2-norm of vector $\mathbf{X}$, and $|\mathbf{X}| = \|\mathbf{X}\|/\sqrt{n}$ is the normalized version. $\mathcal{N}(\mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2. $\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the inner product $\sum_i a_i b_i$. All logarithms are with base e.

II. The Sparse Regression Codebook

A sparse regression code (SPARC) is defined in terms of a design matrix $\mathbf{A}$ of dimension $n \times ML$ whose entries are i.i.d. $\mathcal{N}(0, 1)$. Here n is the block length and M and L are integers whose values will be specified shortly in terms of n and the rate R. As shown in Figure 1, one can think of the matrix $\mathbf{A}$ as composed of L sections with M columns each. Each codeword is a linear combination of L columns, with one column from each section. Formally, a codeword can be expressed as $\mathbf{A}\beta$, where β is a $ML \times 1$ vector $(\beta_1, \ldots, \beta_{ML})$ with the following property: there is exactly one non-zero β_i for $i \in \{1, \ldots, M\}$, one non-zero β_i for $i \in \{M+1, \ldots, 2M\}$, and so forth. The non-zero value of β in section i is set to c_i where the value of c_i will be specified in the next section. Denote the set of all β's that satisfy this property by $\mathcal{B}_{M,L}$.

Since there are M columns in each of the L sections, the total number of codewords is M^L. To obtain a compression rate of R nats/sample, we need

$$M^L = e^{nR} \quad \text{or} \quad L \log M = nR \tag{1}$$

Encoder: This is defined by a mapping $g : \mathbb{R}^n \to \mathcal{B}_{M,L}$. Given the source sequence $\mathbf{S}$ and target distortion D, the encoder attempts to find a $\hat{\beta} \in \mathcal{B}_{M,L}$ such that $\|\mathbf{S} - \mathbf{A}\hat{\beta}\|^2 \leq D$. If such a codeword is not found, an error is declared.

Decoder: This is a mapping $h : \mathcal{B}_{M,L} \to \mathbb{R}^n$. On receiving $\hat{\beta} \in \mathcal{B}_{M,L}$ from the encoder, the decoder produces reconstruction $h(\hat{\beta}) = \mathbf{A}\hat{\beta}$.

Storage Complexity: The storage complexity of the dictionary is proportional to nML. There are several choices for the pair (M, L) which satisfy (1). For example, $L = 1$ and $M = e^{nR}$ recovers the Shannon-style random codebook in which the number of columns in $\mathbf{A}$ is e^{nR}, i.e., the storage complexity is exponential in n.

For our constructions, we choose M to be a low-order polynomial in n. Then L is $\Theta(n/\log n)$, and the number of columns ML in the dictionary is a low-order polynomial in n. This reduction in storage complexity can be harnessed to develop computationally efficient encoders for the SPARC. The results in Section IV show that this choice of (M, L) offers a good trade-off between complexity and error performance.

III. Computationally Efficient Encoder

The source sequence $\mathbf{S}$ is generated by an ergodic source with mean 0 and variance σ^2.

The SPARC is defined by the $n \times ML$ design matrix $\mathbf{A}$. The jth column of $\mathbf{A}$ is denoted $\mathbf{A}_j$, $1 \leq j \leq ML$. The non-zero value of β in section i is chosen to be

$$c_i = \sqrt{\frac{2R\sigma^2}{L}\left(1 - \frac{2R}{L}\right)^{i-1}}, \quad i = 1, \ldots, L. \tag{2}$$

Given source sequence $\mathbf{S}$, the encoder determines $\hat{\beta} \in \mathcal{B}_{M,L}$ according to the following algorithm.

- *Step* 0: Set $\mathbf{R}_0 = \mathbf{S}$.
- *Step* i, $i = 1, \ldots, L$: Pick

  $$m_i = \underset{j:\ (i-1)M+1 \leq\ j\ \leq iM}{\text{argmax}} \left\langle \mathbf{A}_j,\ \frac{\mathbf{R}_{i-1}}{\|\mathbf{R}_{i-1}\|} \right\rangle. \tag{3}$$

  Set

  $$\mathbf{R}_i = \mathbf{R}_{i-1} - c_i \mathbf{A}_{m_i}, \tag{4}$$

  where c_i is given by (2).
- *Step* $L+1$: The codeword $\hat{\beta}$ has non-zero values in positions m_i, $1 \leq i \leq L$. The value of the non-zero in section i given by c_i.

In summary, the algorithm sequentially chooses the m_i's, section by section, to minimize a 'residue' in each step.

A. Computational Complexity

Each of the L stages of the encoding algorithm involves computing M inner products and finding the maximum among them. Therefore the number of operations per source sample is proportional to ML. If we choose $M = L^b$ for some $b > 0$, (1) implies $L = \Theta(n/\log n)$, and the number of operations per source sample is of the order $(n/\log n)^{b+1}$. We note that due to the sequential nature of the algorithm, only one section of the design matrix needs to be kept in memory at each step. When we have several source sequences to be encoded in succession, the encoder can have a pipelined architecture which requires computational space (memory) of the order nLM and has constant computation time per source symbol.

The code structure automatically yields low decoding complexity. The encoder can represent the chosen β with L binary sequences of $\log_2 M$ bits each. The ith binary sequence indicates the position of the non-zero element in section i. Hence the decoder complexity corresponding to locating the L

non-zero elements using the received bits is $L\log_2 M$, which is $O(1)$ per source sample. Reconstructing the codeword then requires L additions per source sample.

IV. Main Result

Theorem 1. *Consider a length n source sequence $\mathbf{S}$ generated by an ergodic source having mean 0 and variance σ^2. Let $\delta_0, \delta_1, \delta_2$ be any positive constants such that*

$$\Delta \triangleq \delta_0 + 5R(\delta_1 + \delta_2) < 0.5. \tag{5}$$

Let $\mathbf{A}$ be an $n \times ML$ design matrix with i.i.d $\mathcal{N}(0,1)$ entries and M, L satisfying (1)*. On the SPARC defined by $\mathbf{A}$, the proposed encoding algorithm produces a codeword $\mathbf{A}\hat{\beta}$ that satisfies the following for sufficiently large M, L.*

$$P\left(\, |\mathbf{S} - \mathbf{A}\hat{\beta}|^2 \; > \; \sigma^2 e^{-2R}(1 + e^R \Delta)^2 \right) < p_0 + p_1 + p_2 \tag{6}$$

where

$$p_0 = P\left(\left| \frac{|\mathbf{S}|}{\sigma} - 1 \right| > \delta_0 \right), \quad p_1 = 2ML \exp\left(-n\delta_1^2/8\right), \quad p_2 = \left(\frac{M^{2\delta_2}}{8 \log M} \right)^{-L}. \tag{7}$$

Proof. A sketch of the proof is given in Section V. The full version can be found in [15].

Corollary 1. *If the source sequence $\mathbf{S}$ generated according to an i.i.d $\mathcal{N}(0,\sigma^2)$ distribution,*

$$p_0 < 2\exp(-3n\delta_0^2/4),$$

and the SPARC with the proposed encoder attains the optimal distortion-rate function $\sigma^2 e^{-2R}$, with probability of excess distortion decaying exponentially in L.

Remarks:

1) The probability measure in (6) is over the space of source sequences and design matrices.
2) Ergodicity of the source is only needed to ensure that $p_0 \to 0$ as $n \to \infty$.
3) For an i.i.d $\mathcal{N}(0,\sigma^2)$ source, Corollary 1 says that with the choice $M = L^b (b > 0)$ we can achieve a distortion within any constant gap of the optimal distortion-rate function $\sigma^2 e^{-2R}$ with the probability of excess distortion falling exponentially in $L = \Theta(n/\log n)$.
4) For a given rate R, Theorem 1 guarantees that the proposed encoder achieves a squared-error distortion close to the Gaussian $D^*(R)$ for all ergodic sources with variance σ^2. Lapidoth [8] also shows that for any ergodic source of a given variance, one cannot attain a squared-error distortion smaller than this using an i.i.d Gaussian codebook with minimum-distance encoding.

Gap from $D^*(R)$: To achieve distortions close to the Gaussian $D^*(R)$ with high probability, we need p_0, p_1, p_2 to all go to 0. In particular, for $p_2 \to 0$ with growing L, from (7) we require that $M^{2\delta_2} > 8 \log M$. Or,

$$\delta_2 > \frac{\log\log M}{2 \log M} + \frac{\log 8}{2 \log M}. \tag{8}$$

To approach $D^*(R)$, note that we need n, L, M to all go to ∞ while satisfying (1): n, L for the probability of error in (7) to be small, and M in order to allow δ_2 to be small according to (8). When n, L, M are sufficiently large, (8) dictates how small Δ can be: the distortion is approximately $\frac{\log\log M}{\log M}$ higher than the optimal value $D^*(R) = \sigma^2 e^{-2R}$.

Performance versus Complexity Trade-off: Recall that the encoding complexity is $O(ML)$ operations per source sample. The performance of the encoder improves as M, L increase – both in terms of the gap from the optimal distortion (8) and the probability of error (7).

- Choosing $M = L^b$ for $b > 0$ yields $L \sim n/\log n$ and the resulting encoding complexity is $\Theta\left((n/\log n)^{b+1}\right)$; the gap from $D^*(R)$ governed by (8) is approximately $\frac{\log\log n}{b \log n}$.
- At the other extreme, the Shannon codebook has $L = 1, M = e^{nR}$. Here the SPARC consists of only one section, and the proposed algorithm essentially performs minimum-distance encoding. The encoding complexity is $O(e^{nR})$ (exponential). From (8), δ_2 is approximately $\frac{\log n}{n}$. The gap Δ from $D^*(R)$ is now dominated by δ_0 and δ_1 whose typical values for the i.i.d Gaussian case are $\Theta(1/\sqrt{n})$ (from (7) and Corollary 1).

Successive Refinement Interpretation: The proposed encoder may be interpreted in terms of successive refinement [16]. We can think of each section of the design matrix $\mathbf{A}$ as a codebook of rate R/L. For step i, $i = 1, \ldots, L$, the residue $\mathbf{R}_{i-1}$ acts as the 'source' sequence, and the algorithm attempts to find the column *within* Section i that minimizes the distortion. The distortion after step i is the variance of the new residue $\mathbf{R}_i$. The minimum mean-squared distortion with a Gaussian codebook [8] at rate R/L is

$$D_i^* = |R_{i-1}|^2 \exp(-2R/L) \approx |R_{i-1}|^2 \, (1 - 2R/L) \tag{9}$$

for $R/L \ll 1$. The typical value of the distortion in Section i is close to D_i^* since the algorithm is equivalent to maximum-likelihood encoding within each section (see (10) in Section V). Since the rate R/L is infinitesimal, the deviations from D_i^* in each section can be quite large. However, since the number of sections L is very large, the final distortion $|\mathbf{R}_L^2|$ is close to the typical value $\sigma^2 e^{-2R}$ with excess distortion probability that falls *exponentially* in L. We emphasize that the successive refinement interpretation is only true for the proposed encoder, and is not an inherent feature of the sparse regression codebook.

Figure 2 shows the performance of the proposed encoder on a unit variance i.i.d Gaussian source. The dimension of $\mathbf{A}$ is $n \times ML$ with $M = L^b$. The curves show the average distortion obtained at various rates for $b = 2$ and $b = 3$. The value of L was increased with rate in order to keep the total computational

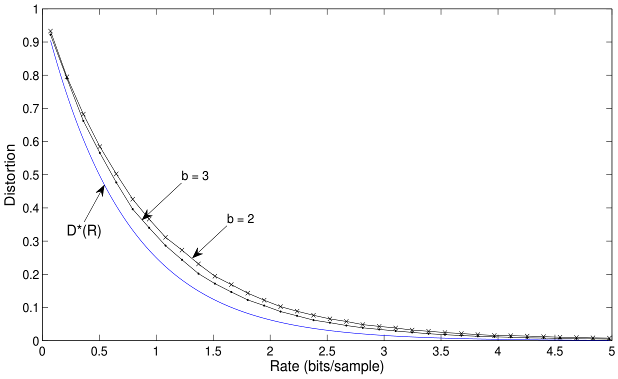

Fig. 2. Average distortion of the proposed encoder for i.i.d $\mathcal{N}(0,1)$ source at various rates. With $M = L^b$, distortion-rate curves are shown for $b = 2$ and $b = 3$ along with $D^*(R) = e^{-2R}$.

complexity ($\propto nL^{b+1}$) similar across different rates. (Recall that block length n is determined by (1).) The reduction in distortion obtained by increasing b from 2 to 3 comes at the expense of an increase in computational complexity by a factor of L. Simulations were also performed for a unit variance Laplacian source. The resulting distortion-rate curve was virtually identical to Figure 2 which is consistent with Theorem 1.

V. Proof of Theorem 1

We first present a non-rigorous analysis of the proposed encoding algorithm based on the following observations.

1) $|\mathbf{A}_j|^2$ is approximately equal to 1 when n is large, for $1 \leq j \leq ML$. This is because $|\mathbf{A}_j|^2$ is the normalized sum of squares of n i.i.d $\mathcal{N}(0,1)$ random variables.
2) Similarly, $|\mathbf{S}|^2$ is approximately equal to σ^2 for large n.
3) If $X_1, X_2 \ldots, X_M$ are i.i.d $\mathcal{N}(0,1)$ random variables, then $\max\{X_1, \ldots, X_M\}$ is approximately equal to $\sqrt{2\log M}$ for large M [17].

Step i, $i = 1, \ldots, L$: We show that if $|\mathbf{R}_{i-1}|^2 \approx \sigma^2\left(1 - \frac{2R}{L}\right)^{i-1}$, then

$$|\mathbf{R}_i|^2 \approx \sigma^2\left(1 - 2R/L\right)^i. \tag{10}$$

(10) is true for $i = 0$ (the second observation above).

For each $j \in \{(i-1)M+1, \ldots, iM\}$, the statistic

$$T_j^{(i)} \triangleq \langle \mathbf{A}_j,\ \mathbf{R}_{i-1}/\|\mathbf{R}_{i-1}\|\rangle \tag{11}$$

is a $\mathcal{N}(0,1)$ random variable. This is because it is the projection of i.i.d $\mathcal{N}(0,1)$ random vector $\mathbf{A}_j$ in the direction of $\mathbf{R}_{i-1}$ and $\mathbf{R}_{i-1}$ is *independent* of $\mathbf{A}_j$. This independence holds because $\mathbf{R}_{i-1}$ is a function of the source sequence $\mathbf{S}$ and the columns $\{\mathbf{A}_{j'}\}$ $1 \leq j' \leq (i-1)M$, which are all independent of $\mathbf{A}_j$ for $(i-1)M+1 \leq j \leq iM$. Further, the $T_j^{(i)}$'s are mutually independent for $(i-1)M+1 \leq j \leq iM$. This can be seen by conditioning on the realization of $\mathbf{R}_{i-1}/\|\mathbf{R}_{i-1}\|$.

We therefore have

$$\max_{(i-1)M+1 \leq j \leq iM} T_j^{(i)} = \left\langle \mathbf{A}_{m_i},\ \frac{\mathbf{R}_{i-1}}{\|\mathbf{R}_{i-1}\|}\right\rangle \approx \sqrt{2\log M}. \tag{12}$$

From (4), we have

$$\begin{aligned}|\mathbf{R}_i|^2 &= |\mathbf{R}_{i-1}|^2 + c_i^2|\mathbf{A}_{m_i}|^2 - \frac{2c_i\|\mathbf{R}_{i-1}\|}{n}\left\langle \mathbf{A}_{m_1}, \frac{\mathbf{R}_{i-1}}{\|\mathbf{R}_{i-1}\|}\right\rangle \\ &\overset{(a)}{\approx} \sigma^2\left(1 - \frac{2R}{L}\right)^{i-1} + c_i^2 - \frac{2c_i\sigma\sqrt{\left(1-\frac{2R}{L}\right)^{i-1}}}{\sqrt{n}}\sqrt{2\log M} \\ &\overset{(b)}{=} \sigma^2\left(1 - \frac{2R}{L}\right)^{i}.\end{aligned} \tag{13}$$

(a) follows from (12) and the induction hypothesis. (b) is obtained by substituting for c_i from (2) and for n from (1). Therefore, the final residue after Step L is

$$|\mathbf{R}_L|^2 = |\mathbf{S} - \mathbf{A}\hat{\beta}|^2 \approx \sigma^2\left(1 - \frac{2R}{L}\right)^L \leq \sigma^2 e^{-2R} \tag{14}$$

where we have used $(1+x) \leq e^x$ for $x \in \mathbb{R}$.

Sketch of Formal Proof:

The essence of the proof is in analyzing the deviation from the typical values of the residual distortion at each step of the algorithm. These deviations arise from atypicality concerning the source, the design matrix and the maximum computed in each step. We introduce some notation to capture the deviations. The norm of the residue at stage i is expressed as

$$|\mathbf{R}_i|^2 = \sigma^2\left(1 - \frac{2R}{L}\right)^i (1+\Delta_i)^2, \quad i = 0, \ldots, L. \tag{15}$$

$\Delta_i \in [-1, \infty)$ measures the deviation of the residual distortion $|\mathbf{R}_i|^2$ from its typical value given in (13).

The norm of $\mathbf{A}_{m_i}$, the column of $\mathbf{A}$ chosen in step i, is written as

$$|\mathbf{A}_{m_i}|^2 = 1 + \gamma_i, \quad i = 1, \ldots, L. \tag{16}$$

We express the maximum of the statistic $T_j^{(i)}$ in Step i as

$$\max_{(i-1)M+1 \leq j \leq iM} T_j^{(i)} = \left\langle \mathbf{A}_{m_i}, \frac{\mathbf{R}_{i-1}}{\|\mathbf{R}_{i-1}\|}\right\rangle = \sqrt{2\log M}(1+\epsilon_i) \tag{17}$$

ϵ_i measures the deviation of the maximum computed in step i from $\sqrt{2\log M}$. Armed with this notation, we have from (4)

$$\begin{aligned}|\mathbf{R}_i|^2 &= |\mathbf{R}_{i-1}|^2 + c_1^2|\mathbf{A}_{m_1}|^2 - \frac{2c_1\|\mathbf{R}_{i-1}\|}{n}\left\langle \mathbf{A}_{m_1}, \frac{\mathbf{R}_{i-1}}{\|\mathbf{R}_{i-1}\|}\right\rangle \\ &= \sigma^2(1 - 2R/L)\Big[(1+\Delta_{i-1})^2 \\ &\qquad + \frac{2R/L}{1-2R/L}(\Delta_{i-1}^2 + \gamma_i - 2\epsilon_i(1+\Delta_{i-1}))\Big].\end{aligned} \tag{18}$$

From (18) and (15), we obtain

$$(1+\Delta_i)^2 = (1+\Delta_{i-1})^2 + \frac{2R/L}{1-2R/L}(\Delta_{i-1}^2 + \gamma_i - 2\epsilon_i(1+\Delta_{i-1})) \tag{19}$$

for $i = 1, \ldots, L$. The goal is to bound the final distortion

$$|\mathbf{R}_L|^2 = \sigma^2\left(1 - \frac{2R}{L}\right)^L (1+\Delta_L)^2. \tag{20}$$

We find an upper bound for $(1+\Delta_L)^2$ that holds under an event whose probability is close to 1. Accordingly, define $\mathcal{A}$ as the event where all of the following hold:

$$|\Delta_0| < \delta_0, \quad \sum_{i=1}^{L} \frac{|\gamma_i|}{L} < \delta_1, \quad \sum_{i=1}^{L} \frac{|\epsilon_i|}{L} < \delta_2.$$

for $\delta_0, \delta_1, \delta_2$ as specified in the statement of Theorem 1. We upper bound the probability of the event $\mathcal{A}^c$ using the following lemmas (proofs in [15]).

Lemma 1. $P\left(\frac{1}{L}\sum_{i=1}^{L}|\gamma_i| > \delta\right) < 2ML\exp\left(-n\delta^2/8\right)$ *for* $\delta \in (0,1]$.

Lemma 2. *For* $\delta > 0$, $P\left(\frac{1}{L}\sum_{i=1}^{L}|\epsilon_i| > \delta\right) < \left(\frac{M^{2\delta}}{\kappa \log M}\right)^{-L}$.

Using these lemmas, we have $P(\mathcal{A}^c) < p_0 + p_1 + p_2$, where p_0, p_1, p_2 are given by (7). The remainder of the proof consists of obtaining a bound for $(1+\Delta_L)^2$ under the condition that $\mathcal{A}$ holds. This is done via the following lemma, proved in [15].

Lemma 3. *When* $\mathcal{A}$ *is true and* L *is sufficiently large,*

$$|\Delta_i| \leq |\Delta_0| w^i + \frac{4R/L}{1-2R/L} \sum_{j=1}^{i} w^{i-j}(|\gamma_j| + |\epsilon_j|), \quad 1 \leq i \leq L \tag{21}$$

where $w = \left(1 + \frac{R/L}{1-2R/L}\right)$.

Lemma 3 implies that when $\mathcal{A}$ holds and L is sufficiently large,

$$\begin{aligned} |\Delta_L| &\leq w^L \left[|\Delta_0| + \frac{4R}{(1-2R/L)w} \left(\sum_{j=1}^{L} \frac{|\gamma_j|}{L} + \sum_{j=1}^{L} \frac{|\epsilon_j|}{L} \right) \right] \\ &\overset{(a)}{\leq} w^L \left[\delta_0 + \frac{4R}{(1-R/L)}(\delta_1 + \delta_2) \right] \\ &\overset{(b)}{\leq} \exp\left(\frac{R}{1-2R/L} \right) \left[\delta_0 + \frac{4R}{(1-R/L)}(\delta_1 + \delta_2) \right] \\ &\leq e^R \left(\delta_0 + 5R(\delta_1 + \delta_2) \right) = e^R \Delta \end{aligned} \tag{22}$$

where Δ is defined in the statement of the theorem. (a) is true because $\mathcal{A}$ holds and (b) is obtained using $1 + x \leq e^x$ with $x = \frac{R/L}{1-2R/L}$. The distortion can then be bounded as

$$|R_L|^2 = \sigma^2 e^{-2R}(1+\Delta_L)^2 \leq \sigma^2 e^{-2R}(1 + e^R\Delta)^2. \tag{23}$$

VI. Conclusion

We showed that Sparse Regression codes achieve the i.i.d Gaussian distortion-rate function with a successive-approximation encoder. In terms of block length n, the encoding complexity is a low-order polynomial in n and the probability of excess distortion decays exponentially in $n/\log n$. The gap from the distortion-rate function $D^*(R)$ is $O(\log\log M/\log M)$, as given in (8). An important direction for future work is designing feasible encoders for SPARCs with faster convergence to $D^*(R)$ as design matrix dimension or block length increases. The results of [18], [19] show that the optimal gap from $D^*(R)$ (among all codes) is $\Theta(1/\sqrt{n})$. The fact that SPARCs achieve the optimal error-exponent with minimum-distance encoding [20] suggests that it is possible to design encoders with faster convergence to $D^*(R)$ at the expense of slightly higher computational complexity.

The results of this paper together with those in [2], [3] show that SPARCs with computationally efficient encoding and decoding achieve rates close to the Shannon-theoretic limits for both lossy compression and communication. Further, [21] demonstrates how source and channel coding SPARCs can be nested to effect binning and superposition, which are key ingredients of multi-terminal source and channel coding schemes. Sparse regression codes therefore offer a promising framework to develop fast, rate-optimal codes for a variety of models in network information theory.

References

[1] A. Barron and A. Joseph, "Least squares superposition codes of moderate dictionary size are reliable at rates up to capacity," *IEEE Trans. on Inf. Theory*, vol. 58, pp. 2541–2557, Feb 2012.

[2] A. Barron and A. Joseph, "Toward fast reliable communication at rates near capacity with Gaussian noise," in *Proc. 2010 IEEE ISIT*.

[3] A. Joseph and A. Barron, "Fast sparse superposition codes have exponentially small error probability for $R < \mathcal{C}$," *Submitted to IEEE Trans. Inf. Theory*, 2012. http://arxiv.org/abs/1207.2406.

[4] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, pp. 3397 –3415, Dec. 1993.

[5] A. R. Barron, A. Cohen, W. Dahmen, and R. A. DeVore, "Approximation and learning by greedy algorithms," *Annals of Statistics*, vol. 36, pp. 64–94, 2008.

[6] I. Kontoyiannis, K. Rad, and S. Gitzenis, "Sparse superposition codes for Gaussian vector quantization," in *2010 IEEE Inf. Theory Workshop*, p. 1, Jan. 2010.

[7] R. Venkataramanan, A. Joseph, and S. Tatikonda, "Gaussian rate-distortion via sparse linear regression over compact dictionaries," in *Proc. 2012 IEEE ISIT*. http://arxiv.org/abs/1202.0840.

[8] A. Lapidoth, "On the role of mismatch in rate distortion theory," *IEEE Trans. Inf. Theory*, vol. 43, pp. 38 –47, Jan 1997.

[9] D. Sakrison, "The rate of a class of random processes," *IEEE Trans. Inf. Theory*, vol. 16, pp. 10 – 16, Jan 1970.

[10] A. Gupta, S. Verdú, and T. Weissman, "Rate-distortion in near-linear time," in *Proc. 2008 IEEE ISIT*, pp. 847 –851.

[11] I. Kontoyiannis and C. Gioran, "Efficient random codebooks and databases for lossy compression in near-linear time," in *IEEE Inf. Theory Workshop on Networking and Inf. Theory*, pp. 236 –240, June 2009.

[12] M. Wainwright, E. Maneva, and E. Martinian, "Lossy source compression using low-density generator matrix codes: Analysis and algorithms," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1351 –1368, 2010.

[13] S. Korada and R. Urbanke, "Polar codes are optimal for lossy source coding," *IEEE Trans. Inf. Theory*, vol. 56, pp. 1751 –1768, April 2010.

[14] R. Gray and D. Neuhoff, "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, pp. 2325 –2383, Oct 1998.

[15] R. Venkataramanan, T. Sarkar, and S. Tatikonda, "Lossy compression via sparse linear regression: Computationally efficient encoders and decoders," http://arxiv.org/abs/1212.1707.

[16] W. Equitz and T. Cover, "Successive refinement of information," *IEEE Trans. Inf. Theory*, vol. 37, pp. 269 –275, Mar 1991.

[17] H. David and H. Nagaraja, *Order Statistics*. John Wiley & Sons, 2003.

[18] A. Ingber and Y. Kochman, "The dispersion of lossy source coding," in *Data Compression Conference (DCC)*, pp. 53 –62, March 2011.

[19] V. Kostina and S. Verdú, "Fixed-length lossy compression in the finite blocklength regime," *IEEE Trans. on Inf. Theory*, vol. 58, no. 6, pp. 3309–3338, 2012.

[20] R. Venkataramanan, A. Joseph, and S. Tatikonda, "Lossy compression via sparse linear regression: Performance under minimum-distance encoding," 2012. http://arxiv.org/abs/1202.0840.

[21] R. Venkataramanan and S. Tatikonda, "Sparse regression codes for multi-terminal source and channel coding," in *50th Allerton Conf. on Commun., Control, and Computing*, 2012.