Efficient Multiple Description Coding Using Sparse Linear Regression

SURF Final Report

By:	Mentor:	Co-mentor:
Oliver Åstrand	Michelle Effros	Parham Noorzad
oast@kth.se	effros@caltech.edu	parham@caltech.edu

November 6, 2017

1 Abstract

With standard network communication protocols, if a packet is lost in transmission it needs to be resent, and the data can only be reconstructed once every packet is received. This becomes tedious when a large portion of the packets are lost due to disturbances. A possible solution to this problem is Multiple Description Coding (MDC), where each of the packets ("descriptions") on their own allow the receiver to partially reconstruct the transmitted data. In this paper, we suggest the use of Sparse Linear Regression Codes (SPARC) as a way of creating an MDC. The SPARC naturally splits each codeword into multiple sections, and the encoder can then minimize the expected distortion for scenarios when different subsets of these sections are received. Since it is unknown to the encoder how many descriptions will be received, our code minimizes expected distortion with respect to the packet loss distribution. Further research is needed to establish its theoretical relationship with, and possible convergence to, the rate-distortion function. Our simulations suggest that the proposed coding algorithm achieves promising performance both when a large and a small portion of the descriptions are received.

2 Introduction

This paper deals with a new approach for Multiple Description Coding (MDC). It is a solution to the problem of delays due to packet loss, which is discussed below. With standard network protocols data that is to be sent over a network is split into smaller parts, called packets. Each packet is then sent across the network individually. This introduces the problem of packet loss, where one or more packets is lost during transmission. The receiver then has to tell the sender to resend those specific packages. Until every packet is received, the data can

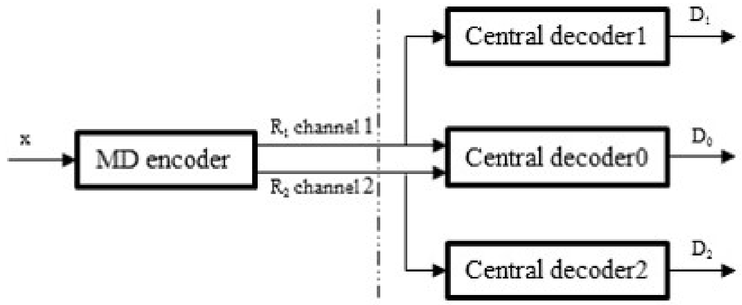

Figure 1: Schematic figure of a MDC with two descriptions. There are three non trivial scenarios, either description one, two or both are available to the decoder. These are then used to reconstruct the encoded data with different distortions D_1, D_2 and D_0 respectively.

not be reconstructed. This works very well if we want to make sure that exactly what is sent is what is received and we have few packet losses. But if the network has a lot of disturbances, and a significant portion of the packets are lost in each transmission, then this becomes tedious and we have to wait for very long to reconstruct any data.

The fundamental idea of Multiple Description Coding is allowing partial reconstruction of the data [1]. That is, if only a subset of the packets, also called descriptions in this context, are received, then the receiver can still reconstruct the data, with some extra distortion. The problem of MDC then becomes how to construct the codes such that each code on its own contains enough information about the original data in order to reconstruct it as well as possible, and so that the descriptions together allow good reconstruction without too much redundant information. There exist theoretical codes that do this today, however they are rarely used in practice. This is because in order to get good performance, they have very high computational complexity.

In this paper we use the idea of Sparse Linear Regression (SPARC) for Multiple Description Coding. A SPARC encodes the information of the data into a vector $\hat{\beta}$ which is sparse. This process is described in detail in Section 3. Because of the way $\hat{\beta}$ is defined it can naturally be split into sections, which can be seen as descriptions on their own. Exactly how this is done is elaborated on in Section 4. What is new about this code is that it has a polynomial computational complexity.

Our results show that SPARC indeed can be used for MDC and they achieve distortion that is near the optimal rate-distortion curve for a single Gaussian source when split into two descriptions. When the code puts equal probability on any number of descriptions being received, it still performs close to optimal regardless of the number of packets received. Details of the performance are given in Section 5.

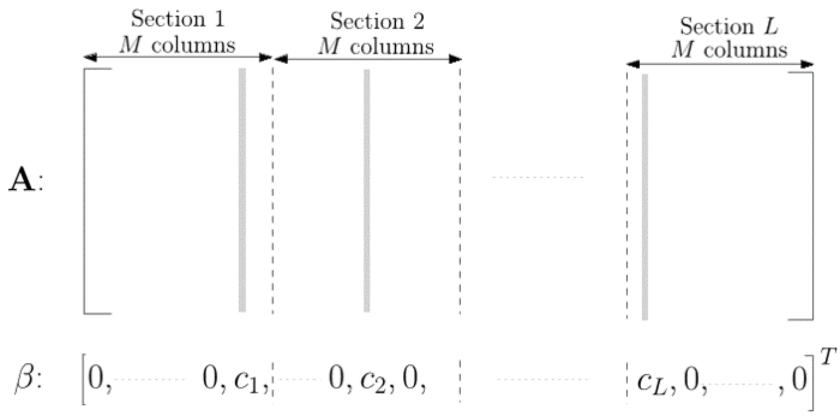

Figure 2: Design of the matrix A and vector β. Each entry in A is i.i.d. generated from a Gaussian. The non zero values in β are all equal to c ($c_1 = c_2 = ... = c_L = c$).

3 Sparse Linear Regression Coding

A SPARC for a single i.i.d. Gaussian source is defined by its $N \times ML$ design matrix, A, which has i.i.d. $\mathcal{N}(0,1)$ elements. Here n is the encoding block length and M and L are constants that partition A into L sections of M columns each, see Figure 2. A codeword can be expressed as $A\beta$, where β is a column vector of length ML, with exactly one nonzero element in each of the L sub-vectors of length M. Thus β has one non zero entry in $1 \leq m \leq M$, one in $M+1 \leq m \leq 2M$, and so forth. All non zero values of β take the same value, here denoted by c, which is defined by

$$c = \sqrt{\frac{1 - e^{-2R}}{L}} \tag{1}$$

where R is the target rate of the code. Figure 2 describes the design of A and a vector β.

The format of β means that every codeword is defined by the L elements that are non-zero. There are M choices for each of the L non-zero entries, which gives a total of M^L possible codewords. If these are to be encoded with a rate R, we get the relationship:

$$M^L = e^{nR} \tag{2}$$

For a block of i.i.d. measurements X from the source, the encoder attempts to find $\hat{\beta}$, the β with lowest expected distortion by

$$\hat{\beta} = \text{argmin}_{\beta \in \mathcal{B}} ||X - A\beta|| \tag{3}$$

where $\mathcal{B}$ is the set of possible β:s. If the distortion is still larger than the target distortion D, then the encoder declares an error. To construct a reproduction $\hat{X}$ of X, the decoder multiplies the vector $\hat{\beta}$ received from the encoder by A:

$$\hat{X} = A\hat{\beta}. \tag{4}$$

The authors of [2] showed that this coding algorithm achieves the rate distortion function when the block length n goes to infinity. In [3] the authors demonstrated a SPARC-algorithm that achieves the rate distortion function with the polynomial computational complexity of $\mathcal{O}\left(\left(\frac{n}{\log(n)}\right)^2\right)$.

4 Designing an MDC Using SPARC

In order to use Sparse Linear Regression Coding for Multiple Description Coding (MD-SPARC) with k descriptions, we define k subsections with $a = L/k$ sections each. We treat each of these subsections of $\hat{\beta}$ as a description. In a regular SPARC, we find $\hat{\beta}$ according to (3). This is equivalent to minimizing the distortion when all subsections of beta (descriptions) are received. When coding an MDC, we do not only care about this case, but also the case when only a subset of the descriptions is received. The distortion when a certain subset is used for reconstruction is:

$$D_i = ||X - A_i\beta_i||^2 \tag{5}$$

where $\hat{\beta}_i$ ($i \in \mathcal{P}(k)$, the set of possible subsets of k-descriptions) describes the subsections of $\hat{\beta}$ that are received, and A_i is the matrix of corresponding sections of A. Instead of minimizing (3) we now want to minimize the weighted sum of achieved distortions under different packet loss scenarios. Let the weight when subset i is received be λ_i. These weights should be set equal to the probability of receiving that subset of descriptions. We now have the encoding function

$$\hat{\beta} = \text{argmin}_{\beta \in \mathcal{B}} \sum_{i \in \mathcal{P}(k)} \lambda_i ||X - A_i\beta_i||^2. \tag{6}$$

To exhaustively search the space $\mathcal{B}$ of possible β:s has complexity M^L, which according to (2) is exponential in n. We therefore have to find another algorithm for finding β. In [3], the authors use a greedy algorithm, which has polynomial time complexity for the same problem. They also show that this algorithm approaches the rate distortion curve as the block length goes to infinity. With slight modification we therefore define the algorithm by first initializing β as the solution to the problem:

$$\beta_0 = \text{argmin}_{\ \beta \in R^{ML}} ||\beta||^2 + \alpha||A\beta - X||^2 \tag{7}$$

$$\beta_0 = \left(\alpha I + A^T A\right)^{-1} A^T X. \tag{8}$$

Here α is a Lagrange multiplier determining the weight between a low norm and proximity to being an exact solution. Then all elements but the largest in each section are set to zero, and the largest is set to c according to (1).

Data: Vector to encode X, design matrix A, number of sections L, priorities of distortion λ, target rate R, Lagrangian parameter α.
Result: Encoded vector $\hat{\beta}$
$\beta = \left(\alpha I + A^T A\right)^{-1} A^T X$;
for *each of L sections in β* **do**
    set all but largest entry to zero;
    set non-zero value to c from (1)
end
while *β hasn't converged* **do**
    for *each of L sections in β* **do**
        for *each of M entries in section* **do**
            set all elements in section but one to zero;
            compute distortion from (9);
        end
        keep choice of non-zero element with lowest distortion;
    end
end

Algorithm 1: Encoding of MD-SPARC

5 Simulation results

The algorithm was implemented using Python and the NumPy package for matrix handling. For our experiments, n-dimensional data was generated from a $\mathcal{N}(0,1)$ source, encoded and decoded using the algorithm, and then the distortion was calculated. This was repeated many times to get an average performance. When the number of descriptions is fixed to two, there are three non trivial scenarios, as illustrated in Figure 1. The rate was set so that receiving two descriptions corresponds to rate 1. If only description one or two is used for reconstruction, then the rate achieved is 0.5. Setting the block length to $n = 10$, the regular SPARC algorithm was run with exhaustive search over all possible β:s. These distortions are therefore optimal for any SPARC method with the block length $n = 10$. They are shown as black dots in Figure 3 and 4. The greedy approach from Algorithm 1 was then used to simulate the performance of MD-SPARC. The different values of λ correspond to prioritizing different numbers of descriptions used for reconstruction. The encoder tries to find:

$$\hat{\beta} = \text{argmin}_{\beta \in \mathcal{B}} \, \lambda \left(D_1 + D_2\right)/2 + (1-\lambda) D_0. \tag{9}$$

Here $\lambda = 0$ corresponds to only minimizing the distortion when both descriptions are used for reconstruction, $\lambda = 1$ minimizes the distortion when only one description is used, and $\lambda = 0.5$ places some weight on both scenarios. The results of these simulations are shown in Figure 3. As expected, the algorithm performs best for one description when $\lambda = 0$ and best for two when $\lambda = 1$. When $\lambda = 0.5$, the algorithm performs somewhere in between for both scenarios. Since the greedy algorithm has a lower time complexity than the optimal SPARC using

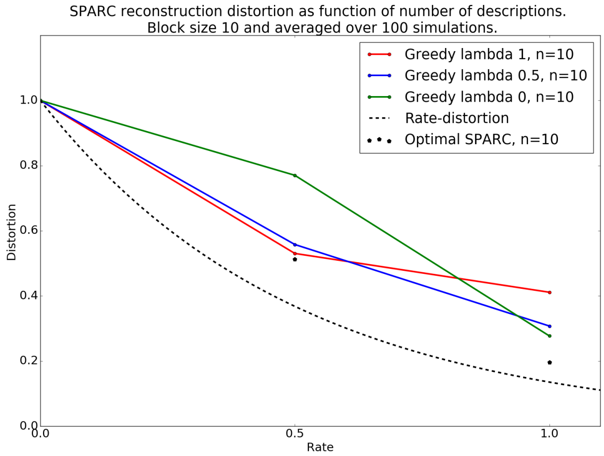

Figure 3: Distortion of reconstructed data encoded using MD-SPARC. Data was encoded using a block length of $n = 10$. The black dots in correspond to the optimal SPARC distortion with a block length of $n = 10$. The value of lambda determines if the algorithm was optimized for using one description ($\lambda = 0$) or two descriptions ($\lambda = 1$) for reconstruction. The choice of $\lambda = 0.5$ places some weight on both scenarios.

exhaustive search. The block length could be extended to $n = 50$. The results of these simulations are shown in Figure 4. Due to this increase in block length the algorithm is able to achieve comparable distortion to the optimal SPARC.

The algorithm can also be used to encode images. The pixels were split into 8×8 blocks which were encoded separately. First the Python package scikit-image loaded the pixel intensities as a scalar from 0 to 1, these were then normalized by subtracting the mean and dividing by the standard deviation. Then they were encoded by the MD-SPARC with four descriptions. At the receiving end, the intensities were decoded and the normalizing process was reversed. Any values that ended up outside the allowed interval of $[0, 1]$ were assigned to the closest endpoint of the interval. An example of resulting reconstructions is depicted in Figure 5. The quality of the reconstruction clearly increases with the number of descriptions.

6 Discussion and future work

The encoding algorithm seems to be approaching the rate distortion function as the block length n goes to infinity. One question for future work would involve theoretical evaluation to see if it approaches the optimal rate distortion bound. The greedy algorithm in this paper uses a slightly different method than the one from [3]. It is possible that there is some better choice of the non-zero coefficients such that the distortion approaches the optimal bound.

We also see that when $\lambda = 0.5$ the distortion is close to that of the algorithm with optimal choice of lambda for each of the scenarios. This is a good attribute

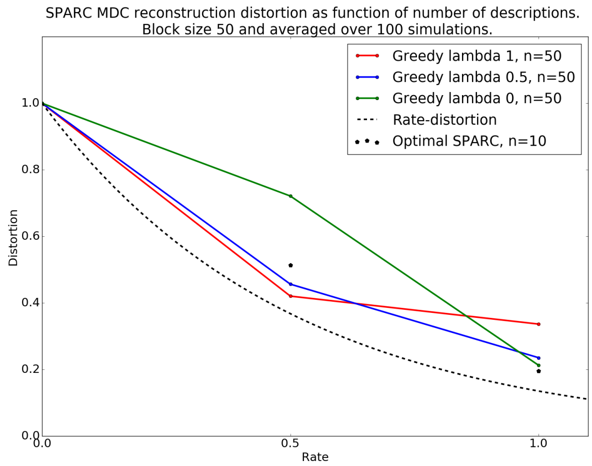

Figure 4: Same graph as in Figure 3, but the MD-SPARC data was encoded using a block length of $n = 50$.

Figure 5: Reconstruction of an image encoded using the MD-SPARC. The far left is the original image. Then from left to right the reconstruction is dependent on one, two, three and four descriptions respectively.

of the algorithm, which seems to persist when the block length is increased.

Using Sparse Linear Regression is clearly a viable method for Multiple Description Coding. It has polynomial time complexity and simulations still suggest that the distortion approaches the rate distortion curve as the encoder block length tends to infinity. However this needs to be evaluated theoretically.

References

[1] Goyal V.K. *Multiple description coding: compression meets the network*, IEEE Signal Processing Magazine, Volume 18, 2001.

[2] Venkataramanan R. & Tatikonda S. *The Rate-Distortion Function and Excess-Distortion Exponent of Sparse Regression Codes With Optimal Encoding*, IEEE Transactions on Information Theory, Volume 63, 2017.

[3] Venkataramanan R., Sarkar T. & Tatikonda S. *Lossy Compression via Sparse Linear Regression: Computationally Efficient Encoding and Decoding*, IEEE Transactions on Information Theory, Volume 60, 2014.