Empirical Rate-Distortion Study of Compressive Sensing-based Joint Source-Channel Coding

by

Muriel Lantosoa Rambeloarison

S.B., E.E.C.S., M.I.T., 2011

Submitted to the Department of Electrical Engineering
and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

Author .....
Department of Electrical Engineering
and Computer Science
May 25, 2012

Certified by .....
Prof. Muriel Médard
Professor of EECS
Thesis Supervisor

Accepted by .....
Prof. Dennis M. Freeman
Chairman, Masters of Engineering Thesis Committee

Empirical Rate-Distortion Study of Compressive Sensing-based Joint Source-Channel Coding

by

Muriel Lantosoa Rambeloarison

Submitted to the Department of Electrical Engineering
and Computer Science
on May 25, 2012, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this thesis, we present an empirical rate-distortion study of a communication scheme that uses compressive sensing (CS) as joint source-channel coding. We investigate the rate-distortion behavior of both point-to-point and distributed cases.

First, we propose an efficient algorithm to find the ℓ_1-norm regularization parameter that is required by the Least Absolute Shrinkage and Selection Operator (LASSO) which we use as a CS decoder. We then show that, for a point-to-point channel, the rate-distortion follows two distinct regimes: the first one corresponds to an almost constant distortion, and the second one to a rapid distortion degradation, as a function of rate. This constant distortion increases with both increasing channel noise level and sparsity level, but at a different gradient depending on the distortion measure. In the distributed case, we investigate the rate-distortion behavior when sources have temporal and spatial dependencies. We show that, taking advantage of both spatial and temporal correlations over merely considering the temporal correlation between the signals allows us to achieve an average of a factor of approximately 2.5 times improvement in the rate-distortion behavior of the joint source-channel coding scheme.

Thesis Supervisor: Prof. Muriel Médard
Title: Professor of EECS

Acknowledgments

I would like to thank Professor Muriel Médard for taking me in the Network Coding and Reliable Communications (RCNC) group, first as a Senior to conduct my Undergraduate Advanced Project, and ultimately as a Master's student. I have learned so much from my interactions with her, and I am beyond grateful for the guidance and support she's offered me all through the past year and a half.

I would also like to thank Professor Dennis M. Freeman, my academic advisor, whose help and patience has made my Course VI MIT journey the best I could have hoped it to be.

A very special thanks goes out to Soheil Feizi and Georgios Angelopoulos, who have been my direct contacts for research questions throughout my time with the RCNC group. Thank you both for all your help and mentoring.

To everyone else in the RCNC group and especially my (real and honorary) officemates: Weifei Zheng, Tong Wang, Jason Cloud, Flávio du Pin Calmon and Amy Zhang. You made my time with the group worth it. Thank you for all the jokes, interesting discussions, and the many (free) food trips. Keep being fantastic.

My deepest gratitude go towards my family away from home: Maryanne Sharp, Benjamin Sharp, Mary Sharp, Miangaly Randriamihaja, Harivony Rakotoarivelo, Emma Diouf, Marie-José Montpetit, and the rest of the UWC crew. I would not have made it without you.

Last but definitely not the least, I would like to thank my family: my daddy Roger (Monsieur Papa), my mummy Eliane (Moumoune), and my sisters Domoina (Doudou Michel) and Josée (ma petite Jojo), who have always been there for me and have encouraged and motivated me all along my journey so far. Merci pour tout. Je suis très honorée de vous dédier ce modeste travail.

Ho an'ny tanindrazako sy ny fianankaviako.

Contents

List of Figures

Chapter 1

Introduction

Sampling has traditionally been dictated by Shannon's theorem, which states that a bandwitdh-limited signal can be perfectly reconstructed without any aliasing if it is sampled at the Nyquist rate, that is at twice the highest frequency present in the signal. Compressive sensing (CS) – also referred to as compressed sensing, compressive sampling, or sparse sampling – is a novel technique which allows to reconstruct signals using much fewer measurements than such traditional sampling methods by taking advantage of the sparsity of the signals to be compressed. As explained in [1], a discrete-time signal, which we consider in this thesis, is said to be sparse when its number of degrees of freedom is significantly smaller compared to its finite length. As such, a signal can be sparse by itself, or it can be made sparse when expressed in a proper basis.

As mentioned in [1], applications for CS cover data acquisition, data compression, but also channel coding. It is this last application, which is extended to source and channel coding, that we explore in the scheme we investigate.

1.1 Previous Work and Overall Goal

Previous work related to the rate-distortion analysis of CS have been focused on its performance related to image compressing [2] and quantized CS measurements [3]. References [4], [5] and [6] derive bounds for the rate-distortion, while [7] presents a

rate-distortion analysis by representing the compressive sensing problem using a set of differential equations derived from a bipartite graph.

The purpose of the work conducted for this thesis is to analyze empirically the rate-distortion behavior of the coding scheme presented in [8]. This scheme, which is explained in more details in Section 1.3, proposes to use CS for both source and channel coding in wireless network with AWGN channels. In this Master of Engineering thesis, we first present an algorithm to find the ℓ_1-norm regularization parameter central to the Least Absolute Shrinkage and Selection Operator (LASSO) we choose as CS decoder. We then consider a point-to-point channel and a distributed setting to illustrate how the rate-distortion varies as a function of channel noise level and sparsity level of the original signals we investigate.

1.2 Rate-Distortion Theory

As explained in [9], rate-distortion theory studies how much compression can be achieved using lossy compression methods. To explain it in the light of its use in this thesis, we consider the simplified communication scheme illustrated in Figure 1-1.

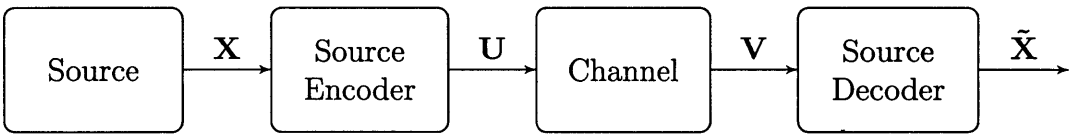


Figure 1-1: Simplied communication scheme

A system designer's goal is to encode the source while minimizing the channel capacity required for transmission and tolerating some average distortion between the source output signal $\mathbf{X}$ and the decoder output $\tilde{\mathbf{X}}$. What rate-distortion aims to study is thus this fundamental trade-off between channel capacity and distortion.

For our rate-distortion study of CS, we consider two distortion measures: the mean-squared error (MSE) and a scaled version of the percent root-mean-square difference (PRD) often used to quantify errors in biomedical signals [10], which is

defined as follows:

$$PRD = \frac{\sqrt{\sum_{n=1}^{N} |\mathbf{X} - \tilde{\mathbf{X}}|^2}}{\sqrt{\sum_{n=1}^{N} |\mathbf{X}|^2}} \tag{1.1}$$

where $\mathbf{X}$ is the original signal of length N and $\tilde{\mathbf{X}}$ its reconstruction.

1.3 Joint Source-Channel Coding based on Compressive Sensing

The main idea behind the work proposed in [8] is to take advantage of the inherent redundancy of sources in a communication scheme in order to achieve a joint source-channel-network coding scheme which is both power- and rate-efficient. While this original scheme includes linear network coding, the scope of this thesis only includes the joint source-channel coding.

In the framework presented in [8], we consider N sources that exhibit both temporal and spatial redundancies, which means that each signal from each source is sparse (temporal) and that the difference between signals from different sources is also sparse (spatial). A schematic view of the proposed scheme is illustrated in Figure 1-2.

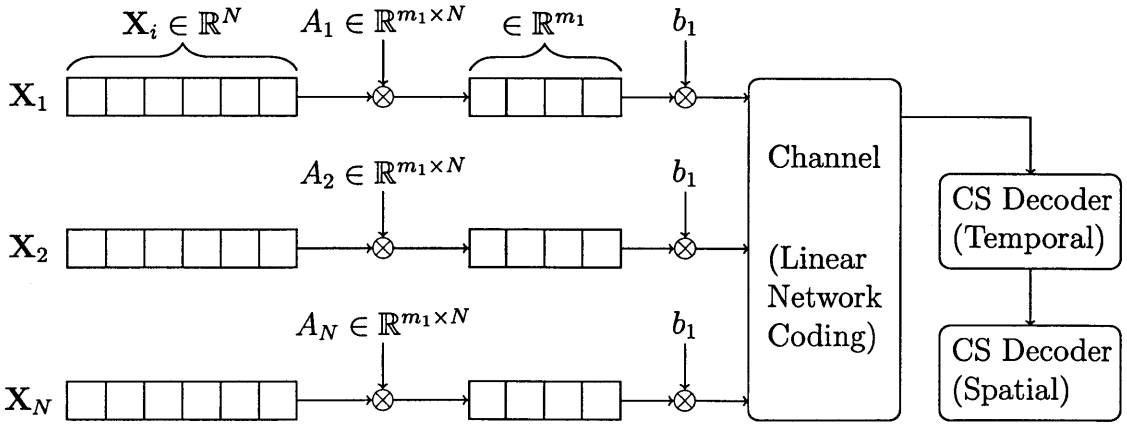


Figure 1-2: Schematic view of joint source-channel-network coding: in the original scheme, the "Channel" block is replaced by a linear network coding block.

The joint channel-source coding is then performed in four steps, as following:

1. *Temporal pre-coding:* each source s_i projects its N-dimensional signal $\mathbf{X}_i$ to a lower dimensional space by multiplying it by the random projection matrix A_i, where A_i obeys the restricted eigenvalue condition [11].

2. *Spacial pre-coding:* each source s_i transmits with probability b_i at each time. Together with Step 1, this step represents the CS encoding.

3. *Transmission through channel:* The CS-encoded signals are transmitted through a noisy or noiseless channel. In the original scheme, the CS-encoded signals are then sent through a network, where the nodes in the network perform linear network coding over the real field.

4. *Decoding:* Each receiver r_i uses CS decoders (in this thesis, we consider a LASSO decoder) to reconstruct the source signals.

1.4 Thesis Outline

In Chapter 2, we review prior results in compressive sensing, and introduce the notation and parameters for our simulations. We also propose the cross-validation with modified bisection algorithm to find the ℓ_1-norm regularization parameter required for the CS reconstruction in our rate-distortion simulations. Rate-distortion results for the joint source-channel coding scheme are presented in Chapter 3, for both the point-to-point and the distributed settings. Finally, we conclude this thesis in Chapter 4, where we also propose leads for future work.

Chapter 2

Background and Simulations Setup

The field of compressive sensing (CS) grew out of looking for sparse solutions for undetermined systems of linear equations. Indeed, CS exploits the notion of recovering sparse solutions from a number of measurements much lower than that required if using traditional sampling methods.

As explained in [12], the algorithms used to solve CS problems have been drawn from a variety of fields such geophysics, statistics, and theoretical computer science, amongst others. Reconstructing a sparse signal through CS can, for example, be done using greedy (e.g.: Orthogonal Matching Pursuit [13]) and combinatorial algorithms. Another type of reconstruction algorithm, which is the one we focus on in this thesis, is based on ℓ_1-norm minimization, which can be expressed as convex optimizations, for which there exist reliable solvers. An example of such ℓ_1-norm minimization technique for CS is the Least Absolute Shrinkage and Selection Operator (LASSO), which we use as our CS decoder.

2.1 Compressive sensing

As explained earlier, CS relies on the sparse nature of signals. By definition, a signal $\mathbf{X} \in \mathbb{R}^N$ is said to be k-sparse vector when k of its coefficients are non-zero, that is $||\mathbf{X}||_{\ell_0} = k$, where $||\cdot||_{\ell_p}$ represents the ℓ_p-norm of a given vector. Some signals are naturally sparse, whereas others might need to be expressed in a particular basis to

become sparse. For example, in the domain of image compression, transforms such as the wavelet transform [14] or the discrete cosine transform [15] are often used to interpret images as sparse signals.

In the CS framework, we want to acquire m linear measurements from the sparse signal $\mathbf{X}$. In order to do so, we defined a CS measurement matrix $\Phi \in \mathbb{R}^{m \times N}$. As such, this matrix Φ, which we interchangeably refer to as a sensing or a CS measurement matrix, represents a *dimensionality reduction* [12], which maps $\mathbb{R}^N$ to $\mathbb{R}^m$, where $m \ll N$. Let $\mathbf{Y_n} = \Phi\mathbf{X}$ be the noiseless observation vector, such that $\mathbf{Y_n} \in \mathbb{R}^m$. The signal $\mathbf{X}$ can be recovered by using m measurements if Φ obeys the *Restricted Eigenvalue* (RE) *Condition* [8], or more restrictively the *Restricted Isometry Property* [1, 12], which are both reproduced below.

Definition 1 (Restricted Eigenvalue Condition) *Let $\Phi \in \mathbb{R}^{m \times N}$ be a measurement matrix and $\mathcal{S}$ be the support of a sparse vector $\mathbf{X}$. Let $\mathcal{C}(\mathcal{S}, \alpha)$ be defined as the subset*

$$\mathcal{C}(\mathcal{S}, \alpha) = \{\mathbf{X} \in \mathbb{R}^N : ||\mathbf{X}_{\mathcal{S}^c}||_{\ell_1} \leq \alpha ||\mathbf{X}_{\mathcal{S}}||_{\ell_1}\}$$

where $\mathcal{S}^c$ is the complement set of $\mathcal{S}$. Φ satisfies the restricted eigenvalue (RE) condition over $\mathcal{S}$ with parameter (γ, α) if

$$\frac{\frac{1}{m}||\Phi\mathbf{X}||_{\ell_2}^2}{||\mathbf{X}||_{\ell_2}^2} \geq \gamma > 0 \tag{2.1}$$

for all $\mathbf{X} \in \mathcal{C}(\mathcal{S}, \alpha) - \{0\}$.

Definition 2 (Restricted Isometry Property) *Let $\Sigma_k = \{\mathbf{X} : ||\mathbf{X}||_{\ell_0} \leq k\}$ be the set of all k-sparse signal. A matrix Φ satisfies the restricted isometry property (RIP) of order k if there exists a $\delta_k \in (0, 1)$, such that*

$$(1 - \delta_k)||\mathbf{X}||_{\ell_2}^2 \leq ||\Phi\mathbf{X}||_{\ell_2}^2 \leq (1 + \delta_k)||\mathbf{X}||_{\ell_2}^2 \tag{2.2}$$

holds for all $\mathbf{X} \in \Sigma_k$. In other words, an equivalent definition of the RIP is that all subsets of k columns of Φ are nearly orthogonal.

As explained in [1], the RIP is important in the field of CS because it guarantees a robustness to noise. Also, having the CS measurement matrix Φ obey the RIP – and thus the RE condition – is a sufficient condition for reconstruction algorithms to reconstruct successfully a sparse signal from noisy measurements [12].

Some examples of sensing matrices $\Phi \in \mathbb{R}^{m \times N}$ that satisfy the RIP are derived through its connection with the Johnson-Lindenstrauss lemma [16, 17]. Amongst others, such matrices include the random matrices whose entries ϕ_{ij} are independent realization of Gaussian random variables:

$$\phi_{ij} \sim \mathcal{N}\left(0, \frac{1}{m}\right), \tag{2.3}$$

matrices whose entries are independent realizations of Bernoulli random variables (also known as Rademacher matrices):

$$\phi_{ij} := \frac{1}{\sqrt{m}} \begin{cases} -1 & \text{with probability 0.5,} \\ +1 & \text{with probability 0.5,} \end{cases} \tag{2.4}$$

as well as related distributions, such as

$$\phi_{ij} := \frac{1}{\sqrt{m}} \begin{cases} -\sqrt{3} & \text{with probability 1/6,} \\ 0 & \text{with probability 1/3,} \\ +\sqrt{3} & \text{with probability 1/6.} \end{cases} \tag{2.5}$$

Given the RIP and the RE condition, it is thus possible to also consider noisy measurements and expect to still obtain a reconstruction of the sparse signal $\mathbf{X}$. In this case, the measurement vector $\mathbf{Y}$ is expressed as

$$\mathbf{Y} = \mathbf{Y_n} + \mathbf{Z} = \Phi\mathbf{X} + \mathbf{Z} \tag{2.6}$$

where $\mathbf{Z}$ is a noise vector characterized by $||\mathbf{Z}||_{\ell_2} \leq \xi$. We assume that z_i is distributed

as a zero-mean random Gaussian variable with variance σ^2.

The CS reconstruction, or decoding, consists of finding a solution $\tilde{\mathbf{X}}$ which solves the following linear program

$$\tilde{\mathbf{X}} = \arg\min_{\mathbf{X}} ||\mathbf{X}||_{\ell_1} \text{ subject to } ||\mathbf{Y} - \Phi\mathbf{X}||_{\ell_2} \leq \xi \tag{2.7}$$

It was shown in [18] that the CS reconstruction in (2.7) can be formulated as a Least Absolute Shrinkage and Selection Operator (LASSO) problem, which is expressed as

$$\tilde{\mathbf{X}} = \arg\min_{\mathbf{X}} \frac{1}{2m}||\mathbf{Y} - \Phi\mathbf{X}||_{\ell_2}^2 + \lambda||\mathbf{X}||_{\ell_1} \tag{2.8}$$

where $\lambda \geq 0$ is the ℓ_1-norm regularization parameter. By definition, given a vector $\mathbf{X}$ and a solution $\tilde{\mathbf{X}}$, the LASSO problem involves a ℓ_1-penalization estimation, which shrinks the estimates of the coefficients of $\tilde{\mathbf{X}}$ towards zero relative to their maximum likelihood estimates [18]. Equation (2.8) thus outputs a solution $\tilde{\mathbf{X}}$ that is desired to have a number of non-zero coefficients close to k, while maintaining a high-fidelity reconstruction of the original signal. Thus, as λ increases, so does the number of coefficients forced to zero.

In the next section, we propose an algorithm to choose λ, based on work presented in [19] and [20].

2.2 Cross-validation with modified bisection

[19] and [20] have demonstrated the effective use of the cross-validation method in choosing the ℓ_1-norm regularization parameter λ . As explained in [21], cross-validation is a statistical technique which allows to choose a model which best fits a set of data. It operates by dividing the available data into a training set to learn the model and a testing set to validate it. The goal is then to select the model that best fits both the training and testing sets.

In our case, we use a modified version of this algorithm to choose the value the value of λ which minimizes the energy of the relative error between some original signal

and its reconstruction. As such, the $m \times N$ sensing matrix Φ in (2.8) is separated into a training and a cross-validation matrix, as shown in (2.9),

$$\Phi \rightarrow \begin{bmatrix} \Phi_{tr} \\ \Phi_{cv} \end{bmatrix} \tag{2.9}$$

where Φ_{tr} and Φ_{cv} are, respectively, $m_{tr} \times N$ and $m_{cv} \times N$ matrices, and $m_{tr} + m_{cv} = m$. In order for the cross-validation algorithm to work properly, Φ_{tr} and Φ_{cv} must be adequately normalized and have the same distribution as Φ. There is a strong correlation and a trade-off between the performance of the algorithm and a higher number of cross-validation measurements chosen [20]: indeed, the larger m_{cv} is, the higher the robustness to additive measurement or channel noise. However, as the number of training measurements m_{tr} decreases, so does the reconstruction performance of the algorithm. Hence it is necessary to find a reasonable consensus between m_{cv} and m_{tr}.

2.2.1 Details of the algorithm

For the coding scheme we are considering, we fix the number of cross-validation measurements at 10% of the total number of measurements, so $m_{cv} = \lfloor 0.1\, m \rceil$, which provide a reasonable trade-off between complexity and performance of the algorithm [19]. $\lfloor \cdot \rceil$ denotes the closest integer operator.

Algorithm 1 summarizes the cross-validation technique used to find the best value of λ for the rate-distortion simulations.

We are given an original signal $\mathbf{X}$, a cross-validation matrix Φ_{cv}, and a training matrix Φ_{tr}. The cross-validation algorithm is initiated by first generating the cross-validation and the training measurement vectors $\mathbf{Y}_{cv}$ and $\mathbf{Y}_{tr}$, which are corrupted by the zero-mean Gaussian noise vector, which represents the channel noise (Lines 1 and 2 of Algorithm 1). It is important to note that the noise vectors $\mathbf{Z}_{cv}$ and $\mathbf{Z}_{tr}$ have the same variance per symbol.

On Line 3, we choose the highest value of λ that we are willing to consider for

Algorithm 1 Cross-validation with modified bisection method

1: $\mathbf{Y}_{cv} = \Phi_{cv}\mathbf{X} + \mathbf{Z}_{cv}$
2: $\mathbf{Y}_{tr} = \Phi_{tr}\mathbf{X} + \mathbf{Z}_{tr}$
3: $\lambda = \lambda_{\text{init}}$
4: Let ϵ be an empty vector with coefficients ϵ_i
5: **while** i $\leq$ MaxIterations **do**
6: Solve $\tilde{\mathbf{X}}_{tr}^{[\lambda]} = \arg\min_{\mathbf{X}} \frac{1}{2m}||\mathbf{Y}_{tr} - \Phi_{tr}\mathbf{X}||_{\ell_2}^2 + \lambda||\mathbf{X}||_{\ell_1}$
7: $\epsilon_i \leftarrow ||\mathbf{Y}_{cv} - \Phi_{cv}\tilde{\mathbf{X}}_{tr}^{[\lambda]}||_{\ell_2}$
8: $\lambda \leftarrow \lambda/1.5$
9: **end while**
10: $\lambda^\star = \arg\min_{\lambda} \epsilon = \arg\min_{\lambda} ||\mathbf{Y}_{cv} - \Phi_{cv}\tilde{\mathbf{X}}_{tr}^{[\lambda]}||_{\ell_2}$

our algorithm. It is selected such that we know it leads to an all-zero reconstructed signal $\tilde{\mathbf{X}}_{tr}^{[\lambda]} = \mathbf{0}$. The vector ϵ defined on Line 4 is used to store the cross-validation error values.

The following section of the cross-validation algorithm is then reiterated for a chosen number of times: our results show that 12 repetitions allow us to obtain a satisfying value for $\lambda^\star$. Line 6 depicts how the cross-validation algorithm is initiated by first obtaining an estimation $\tilde{\mathbf{X}}_{tr}^{[\lambda]}$ of the reconstructed signal using Φ_{tr} as a measurement matrix for the value of λ that is currently tested. The cross-validation error $||\mathbf{Y}_{cv} - \Phi_{cv}\tilde{\mathbf{X}}_{tr}^{[\lambda]}||_{\ell_2}$ is then computed (Line 7) and stored in the vector ϵ. The next value for λ to be investigated is obtained by dividing the current value by 1.5 (Line 8). The optimal value $\lambda^\star$ is then the one that minimizes the cross-validation error (Line 10).

Figure 2-1 shows the typical behavior of the cross-validation error $||\mathbf{Y}_{cv} - \Phi_{cv}\tilde{\mathbf{X}}_{tr}^{[\lambda]}||_{\ell_2}$ as a function of the tested values of λ for a given channel noise level.[1]

As seen, there is a sweet spot (at around 1.5^{-7}) at which the cross-validation error reaches a minimum before slightly increasing again: this minimum value corresponds to the value of $\lambda^\star$ we would choose for this particular setting.

In the field of CS, cross-validation is mainly used with homotopy continuation algorithms such as LARS [22], which iterate over an equally-spaced range of decreasing

[1]Our definition of channel noise measurement is provided in the introduction to Chapter 3.

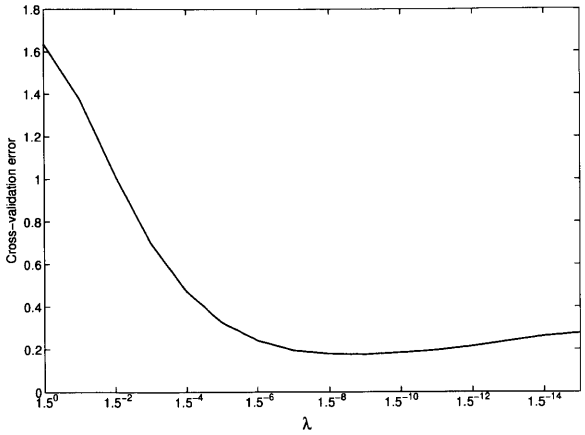


Figure 2-1: Typical behavior of the cross-validation error as a function of λ: this plot was generated with a sparsity ratio of $k/N = 0.025$ for a signal of length $N = 1024$, 5% channel noise, and 512 CS measurements

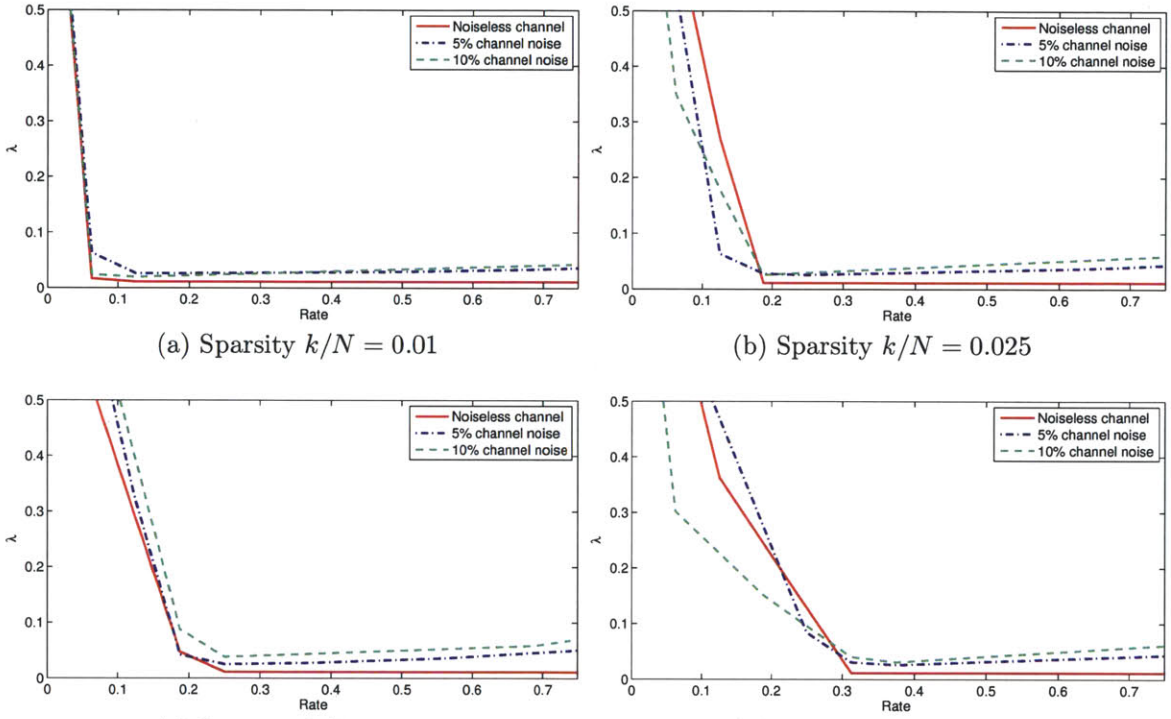
values for λ. While this iterative process allows for better accuracy for smaller range steps, it comes at the cost of a latency. Indeed, the higher the number of values of λ tested, the longer it takes the first algorithm to output $\lambda^\star$, owing to the time-consuming decoding (Line 6). In our scheme, we circumvent this latency issue by considering a decreasing geometrical sequence of values of λ, which still guarantees that we find a solution for $\lambda^\star$ of the same order as the one predicted by an homotopy continuation algorithm, but in a fraction of the time. We are able to obtain a solution after a maximum of 15 iterations of Lines 6 to 8, by using a method comparable to the bisection method [23] to obtain the values of λ to be tested. However, in order to improve the accuracy, we choose a common ratio of 1.5^{-1} instead of 2^{-1}. By abuse of notation, we refer to this technique as a *"cross-validation with modified bisection method."*

2.2.2 Performance as a function of noise and sparsity levels

When investigating the performance of the cross-validation with modified bisection algorithm, it is interesting to focus on the values that $\lambda^\star$ corresponding to higher values of rate. Indeed, for such rates, where the number of measurements is adequate to obtain an acceptation reconstruction, the value of $\lambda^\star$ tends to converge towards a

constant value. The low rate regions can thus be regarded as transient regions where the reconstruction error is relatively high and fluctuating, resulting in higher than expected values $\lambda^\star$, which could also be error-prone in the reconstruction.

Figure 2-2 illustrates how $\lambda^\star$ evolves as a function of noise at different sparsity levels. We can see that the almost constant value towards which $\lambda^\star$ converges to at



(a) Sparsity $k/N = 0.01$

(b) Sparsity $k/N = 0.025$

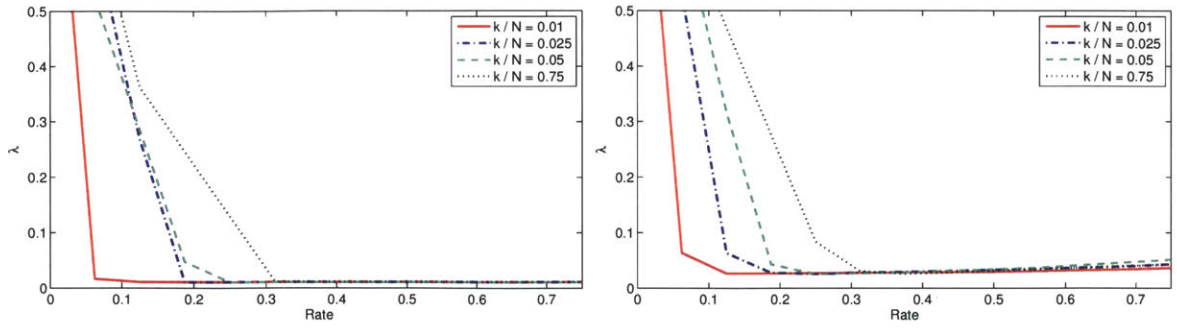
(c) Sparsity $k/N = 0.05$

(d) Sparsity $k/N = 0.075$

Figure 2-2: Behavior of $\lambda^\star$ as a function of noise

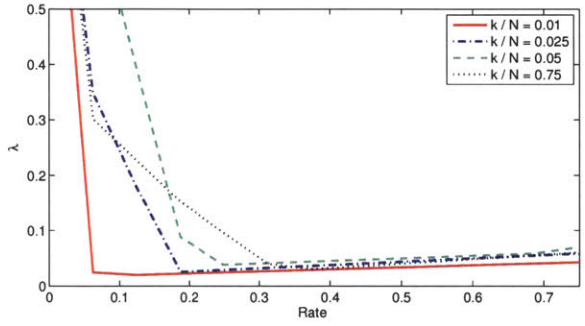
higher rates increases with increasing channel noise. It is also interesting to note that the higher the sparsity ratio, the higher the rate at which this constant value of $\lambda^\star$ is achieved is.

Observing the behavior of $\lambda^\star$ as a function of sparsity, as depicted on Figure 2-3, shows that the sparsity level also affects the rate at which the value of $\lambda^\star$ settles: the higher the sparsity level, the higher this rate value is. Additionally, we notice that, at a given noise level, the value towards which $\lambda^\star$ converges is approximately the same regardless of the sparsity ratio.



(a) Noiseless channel

(b) 5% noise channel



(c) 10% noise channel

Figure 2-3: Behavior of $\lambda^\star$ as a function of sparsity

2.3 Signal model and measurement matrix

In this section and the subsequent one, we present the details of our simulations setup. For each setting we investigate, we consider a k-sparse signal $\mathbf{X}$ of length $N = 1024$, and define its sparsity ratio as $k/N = \alpha$. The signal $\mathbf{X}$ is formed of spikes of magnitudes ± 1 and ± 0.5, where each magnitude has a probability of $\alpha/4$. Figure 2-4 shows an example of an original signal generated for our simulations as well as the corresponding reconstructed signal. Some of the noise in the reconstructed signal is due to the noise that is applied to the channel.

2.4 Simulation software

The simulations were implemented in MATLAB using `cvx` [24], a modeling system for convex optimization which uses disciplined convex programming to solve the LASSO optimization in (2.8) [25]. The code snippet in Figure 2-5 shows how the LASSO

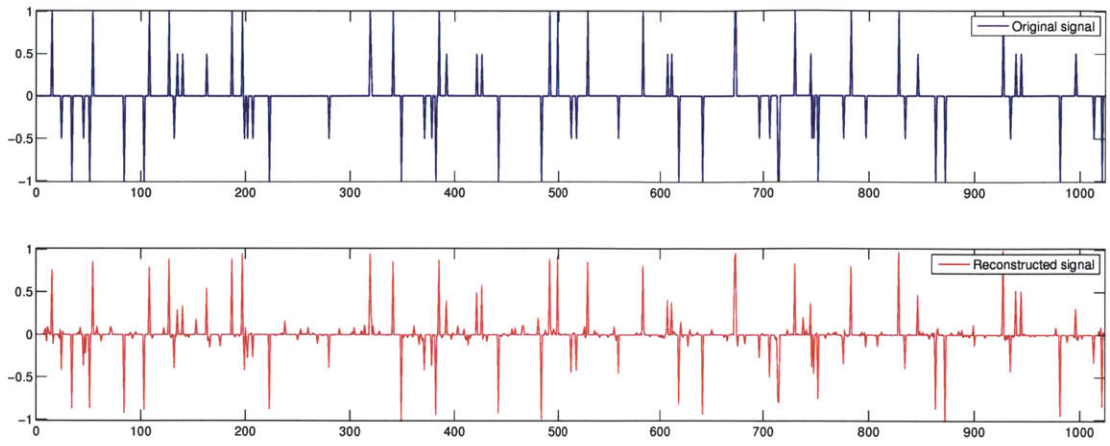


Figure 2-4: Example of original and corresponding reconstructed signals

```
% A - measurement matrix
% X - original signal or length N
% Y - measurement vector
% λ - l1 parameter

cvx_begin;
    variable x(N); % Declare vector variable
    minimize(1/2 * pow_pos(norm(A*X - Y), 2) +  λ * norm(X, 1));
cvx_end;
```

Figure 2-5: Typical `cvx` run

reconstruction is described using `cvx`:

When `cvx` is given a specification such as the one in Figure 2-5, the program uses the following four steps to output a solution to the sparse problem and thus output a CS reconstructed signal:

- The convexity of the problem is first verified.
- An interior-point method (IPM) problem compatible with the original ℓ_1 minimization problem is then generated. Reference [25] shows that under certain conditions, among which is the fact that the problem we wish to solve is convex, it can be transformed into an IPM form, which is easily solvable.
- The transformed problem is solved using the solvers `SDPT3` or `SeDuMi` citecvx.

- The solution outputted by the IPM-problem is then transformed back to the original problem to obtain the desired solution. [24]

It is important to note that, in the code snippet presented above, the factor multiplying the ℓ_2 norm of the CS reconstruction error is set at $\frac{1}{m}$, instead of $\frac{1}{2m}$ as in (2.8). This substitution allows us to obtain values to obtain values of $\lambda^\star$ that are not of the order of 10^{-10} or less from the cross-validation with bisection method algorithm.

Chapter 3

Results and Analysis

In this chapter, we evaluate the performance of the compressive sensing-based joint source-channel coding scheme for both a point-to-point channel, and a distributed case. As such, we investigate both noiseless and noisy channels.

When we consider a noisy channel, we often characterize the standard deviation of channel noise by a given percentage of the power the transmitted compressive-sensing encoding signal. As such, given a CS measurement matrix $\Phi \in \mathbb{R}^{m \times N}$ and an original signal $\mathbf{X} \in \mathbb{R}^N$, we obtain a noiseless measurement vector $\mathbf{Y_n} = \Phi\mathbf{X}$, whose power is calculated as

$$P_{Y_n} = \frac{||\mathbf{Y_n}||_{\ell_2}}{m} \tag{3.1}$$

where $\mathbf{Y_n} \in \mathbb{R}^m$.

For a given channel noise percentage of α, the standard deviation of the random Gaussian channel noise $\mathbf{Z}$ defined in (2.6) is thus defined as

$$\sigma_{\mathbf{Z}} = 0.01 \times \alpha \times P_{Y_n} \tag{3.2}$$

We shall often refer to this definition in the presentation of our rate-distortion results.

3.1 Joint CS-based source-channel coding for a point-to-point channel

In this section, we evaluate the performance of a joint source-channel coding scheme using compressive sensing (CS) proposed in [8] and whose schematic is shown in Figure 3-1.

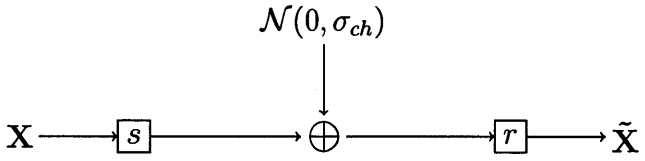


Figure 3-1: Point-to-point channel setting

The signal and measurement models are defined in Section 2.3. The sensing-communication scheme is performed in the following steps:

Step 1 (Encoding) The CS encoding is done by taking m measurements of the signal $\mathbf{X}$ of length $N = 1024$ using a measurement matrix $\Phi \in \mathbb{R}^{m \times N}$ distributed as in (2.4) to obtain a measurement vector $\mathbf{Y_n} = \Phi\mathbf{X}$.

Step 2 (Transmission through channel) The measurement vector $\mathbf{Y}$ is transmitted through a noiseless or noisy channel. The signal reaching the receiver is $\mathbf{Y} = \mathbf{Y_n} + \mathbf{W} = \Phi\mathbf{X} + \mathbf{Z}$, where $\mathbf{Z} \in \mathbb{R}^m$ is additive zero-mean random Gaussian noise.

Step 3 (Decoding) At the receiver, the LASSO decoder outputs an reconstructed signal $\tilde{\mathbf{X}}$ of $\mathbf{X}$ by solving the following optimization

$$\tilde{\mathbf{X}} = \arg\min_{\mathbf{X}} \frac{1}{2m}||\mathbf{Y} - \Phi\mathbf{X}||_{\ell_2}^2 + \lambda^\star||\mathbf{X}||_{\ell_1} \tag{3.3}$$

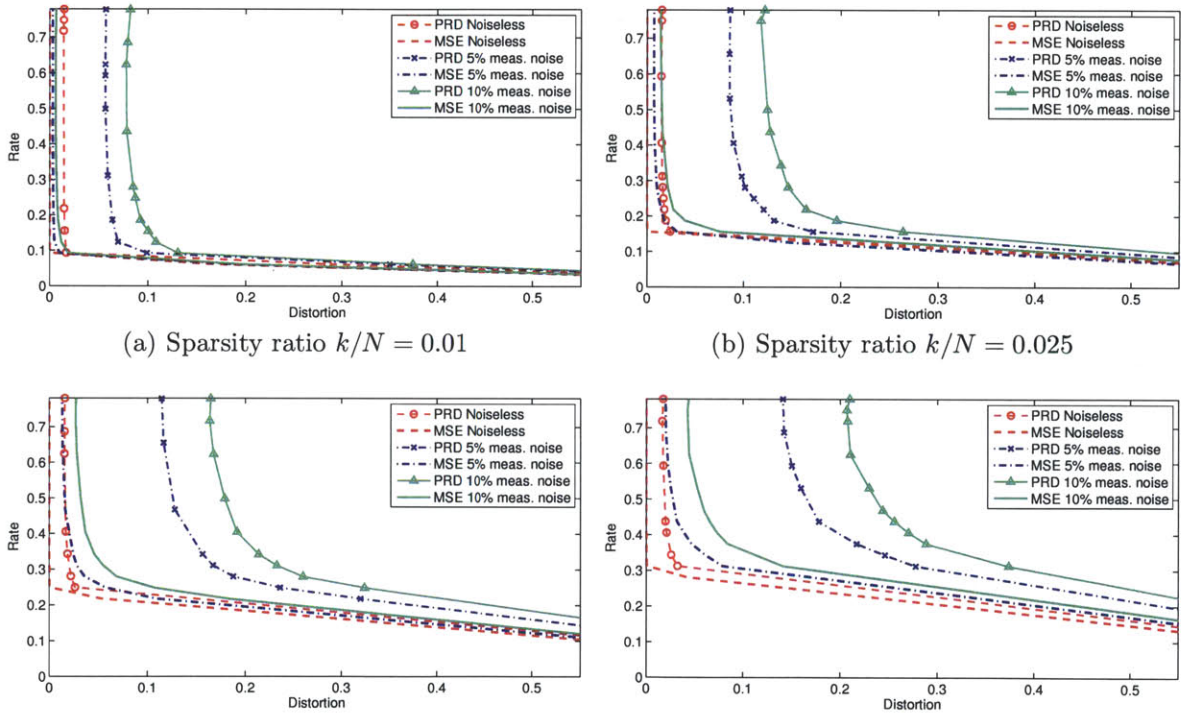
where we use Algorithm 1 to find $\lambda^\star$.

The rate is calculated as m/N and we compare how both the channel noise level and the sparsity of the original signal affect the rate-distortion behavior of the scheme,

using PRD and MSE as distortion measures. In these simulations, each point has been achieved by averaging the distortion values obtained by running each setting (channel noise, m, and sparsity ratio) 15 times. We investigate three different noise levels, which are respectively noiseless, 5% and 10% channel noise, as well as four sparsity ratios $k/N = [0.01, 0.025, 0.05, 0.075]$.

3.1.1 Rate distortion as a function of noise level

Figure 3-2 how the rate-distortion behaves for different noise levels we consider. On each subfigure, we compare both distortion metrics.



(a) Sparsity ratio $k/N = 0.01$

(b) Sparsity ratio $k/N = 0.025$

(c) Sparsity ratio $k/N = 0.05$

(d) Sparsity ratio $k/N = 0.075$

Figure 3-2: Rate-Distortion for a point-to-point channel for different sparsity levels

As seen in Figure 3-2, we can distinguish two regimes in the rate-distortion curves: the first one corresponds to an almost constant distortion $D^\star$ after the number of measurements exceeds some critical value $m^\star$. As expected, both $m^\star$ and $D^\star$ increase slightly with increasing channel noise. However, we observe that this increase is much more important when PRD is used a distortion measure.

The second observed regime demonstrates a rapid degradation of the distortion, as the number of measurements is insufficient to properly reconstruct the original signal. This rapid degradation corresponds to the settings of the simulations where the number of measurements is inferior to $m^\star$. As expected, this second regime coincides with the rate regions where the values of $\lambda^\star$ are high and fluctuating.

3.1.2 Rate distortion as a function of sparsity level

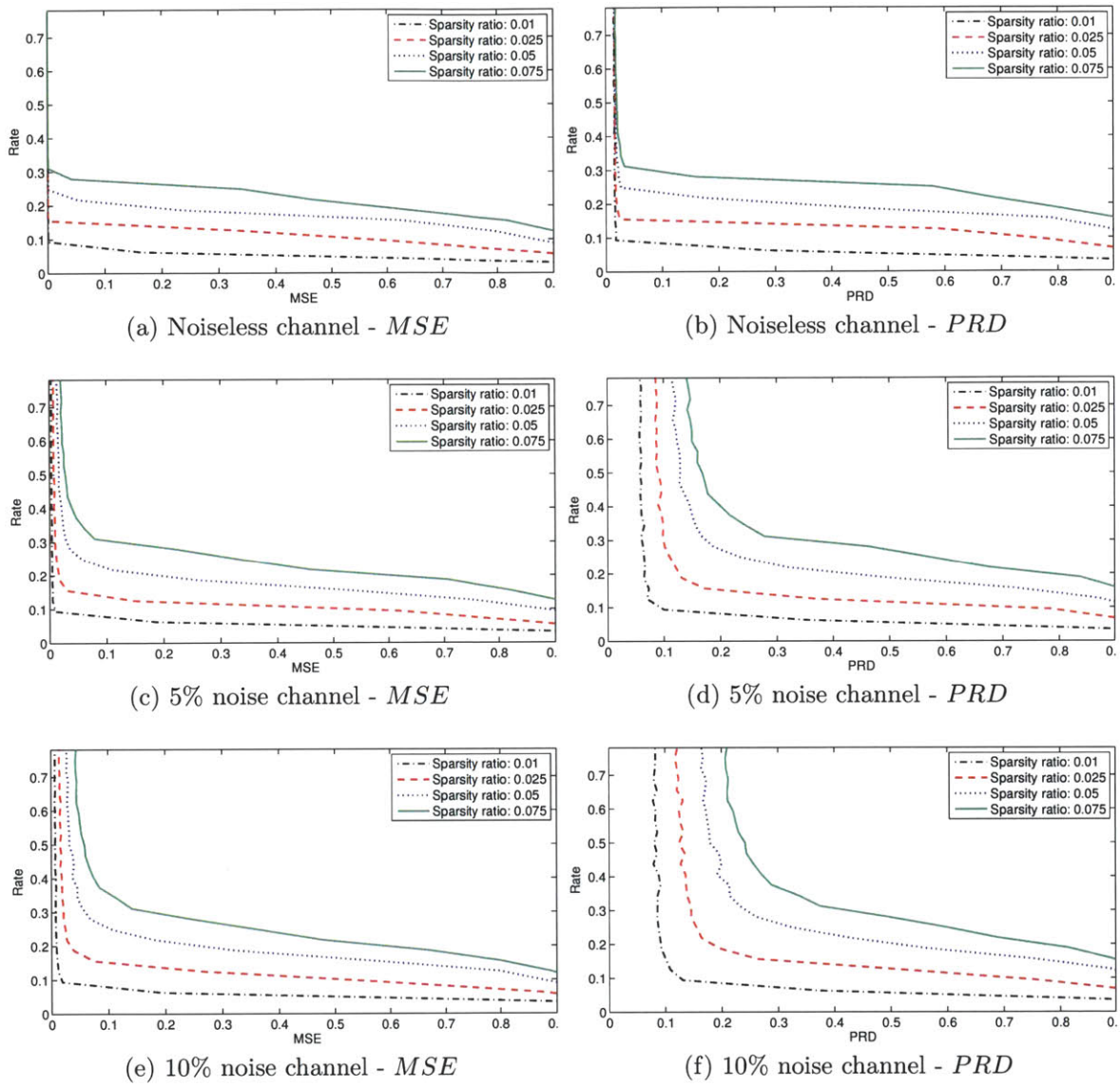
It is also interesting to observe how the scheme operates at a fixed channel noise level and with a varying sparsity level. The comparison of the rate-distortion behaviors with this setting is illustrated in Figure 3-5 for the two distortion metrics we consider.

For a given noise level, we observe an upper-right shift of the curves for increasing sparsity ratio. In particular, we can see that the value of $m^\star$ increases almost linearly with the sparsity ratio. We also notice that the value of $m^\star$ increases much sharply when MSE is used as a distortion measure. As before, we can observe that the changes in rate-distortion curves are much distinguishable when the distortion measure is PRD.

3.2 Joint CS-based source-channel coding for a distributed case

In this section, we evaluate the performance of the compressive sensing-based joint source-channel coding scheme for a distributed setting. We consider the single-hop network depicted in Figure 3-4 with two sources s_1 and s_2, whose samples exhibit both spatial and temporal redundancies [8]. As previously defined, the temporal redundancy refers to the fact that each signal is sparse; the spatial redundancy refers to the fact that the difference between the two signals at the two sources is sparse.

In our simulations, $\mathbf{X}_1$ is k_1-sparse and $\mathbf{X}_2 = \mathbf{X}_1 + \mathbf{E}$, where $\mathbf{E}$ is a k_2-sparse error signal; we assume that $k_1 \gg k_2$. The goal is to reconstruct both $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$ at the receiver r. We present two ways of performing these reconstructions, and in both



(a) Noiseless channel - MSE

(b) Noiseless channel - PRD

(c) 5% noise channel - MSE

(d) 5% noise channel - PRD

(e) 10% noise channel - MSE

(f) 10% noise channel - PRD

Figure 3-3: Rate-Distortion for a point-to-point channel as a function of sparsity level for different channel noise levels

cases, the total rate and the total distortion were respectively calculated as following

$$R_{total} = \frac{m_1 + m_2}{N} \tag{3.4}$$

$$D_{total} = D_1 + D_2 \tag{3.5}$$

where m_i is the number of compressive sensing measurements taken at source s_i and D_i is the distortion measured between the original signal $\mathbf{X}_i$ and its reconstruction

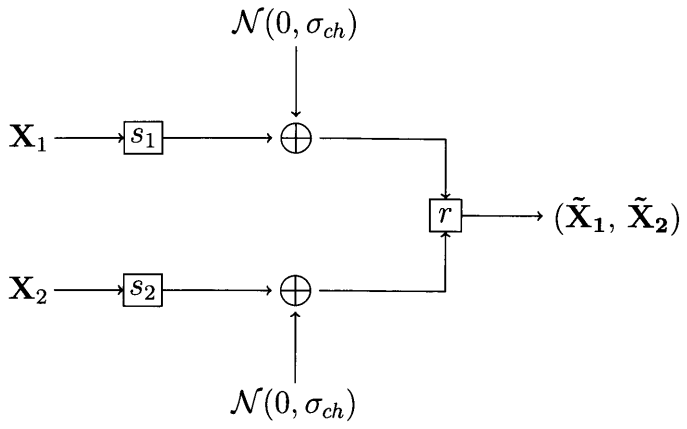


Figure 3-4: Single-hop network for distributed cases

$\tilde{\mathbf{X}}_i$. For both of the reconstruction cases we analyze, we present the results of the simulations for noiseless channels as well as channels with 5%, and 10% noise.

3.2.1 Case 1: Only temporal dependency is considered

In this case, we treat s_1 and s_2 as if there were two independent sources: $\mathbf{X}_1$ and $\mathbf{X}_2$ are compressed and decompressed independently. Algorithm 2 summarizes how this process is done.

Algorithm 2 Distributed case 1: only temporal decency is considered

1: $\mathbf{Y}_1 = \Phi_1 \mathbf{X}_1 + \mathbf{Z}_1$
2: $\mathbf{Y}_2 = \Phi_2 \mathbf{X}_2 + \mathbf{Z}_2$
3: Decompress $\mathbf{Y}_1$ to obtain $\tilde{\mathbf{X}}_1$ by solving
$\tilde{\mathbf{X}}_1 = \arg\min_{\mathbf{X}_1} \frac{1}{2m} ||\mathbf{Y}_1 - \Phi_1 \mathbf{X}_1||_{\ell_2}^2 + \lambda^\star ||\mathbf{X}_1||_{\ell_1}$
4: Decompress $\mathbf{Y}_2$ to obtain $\tilde{\mathbf{X}}_2$ by solving
$\tilde{\mathbf{X}}_2 = \arg\min_{\mathbf{X}_2} \frac{1}{2m} ||\mathbf{Y}_2 - \Phi_2 \mathbf{X}_2||_{\ell_2}^2 + \lambda^\star ||\mathbf{X}_2||_{\ell_1}$

The signals that r receives from s_1 and s_2 are respectively shown in Lines 1 and 2 of Algorithm 2. The vector $\mathbf{Z}_i$ represents an additive zero-mean Gaussian noise associated with the channel from s_i to r, and $\Phi_i \in \mathbb{R}^{m_i \times N}$ are random sensing matrices similar to (2.4) with which the signals $\mathbf{X}_1$ and $\mathbf{X}_2$ are encoded.

Lines 3 and 4 of the algorithm correspond to the CS LASSO decoding performed at r to obtain the reconstructions $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_2$ of the original signals $\mathbf{X}_1$ and $\mathbf{X}_2$.

3.2.2 Case 2: Both spatial and temporal dependencies are considered

In this case, we also take advantage of the spatial correlation in addition to the temporal correlation between $\mathbf{X}_1$ and $\mathbf{X}_2$, as shown in Algorithm 3.

The reconstructions are run by first obtaining an estimate $\tilde{\mathbf{X}}_1$ for $\mathbf{X}_1$ using m_1 measurements. We then take m_2 measurements of $\mathbf{X}_2$, where $m_2 \ll m_1$, which allows us to reconstruct the error signal, using $\tilde{\mathbf{X}}_1$. Given how $\mathbf{X}_2$ was generated, we thus generate $\tilde{\mathbf{X}}_2$ by adding the reconstructions for $\mathbf{X}_1$ and $\mathbf{E}$, as summarized in Algorithm 3.

Algorithm 3 Distributed case 2: both spatial and temporal dependencies are considered

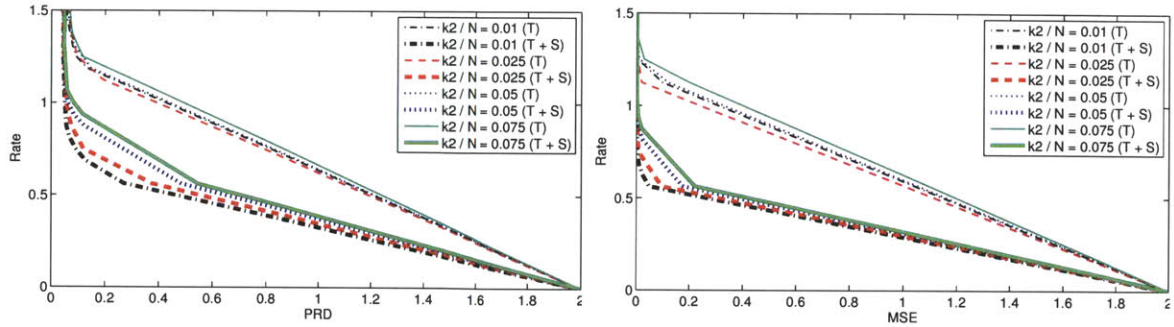
1: $\mathbf{Y}_1 = \Phi_1\mathbf{X}_1 + \mathbf{Z}_1$
2: Decompress $\mathbf{Y}_1$ to obtain $\tilde{\mathbf{X}}_1$ by solving
$\tilde{\mathbf{X}}_1 = \arg\min_{\mathbf{X}_1} \frac{1}{2m}||\mathbf{Y}_1 - \Phi_1\mathbf{X}_1||_{\ell_2}^2 + \lambda^\star||\mathbf{X}_1||_{\ell_1}$
3: $\mathbf{Y}_2 = \Phi_2\mathbf{X}_2 + \mathbf{Z}_2$
4: $\mathbf{Y}_2 = \Phi_2(\mathbf{X}_1 + \mathbf{E}) + \mathbf{Z}_2$, and we already have an estimate for $\mathbf{X}_1$
5: Let $\mathbf{Y}_\mathbf{E} = \mathbf{Y}_2 - \Phi_1\tilde{\mathbf{X}}_1$
6: Thus $\mathbf{Y}_\mathbf{E} = \Phi_2\mathbf{E} + \mathbf{Z}_\mathbf{E}$
7: Decompress $\mathbf{Y}_\mathbf{E}$ to obtain $\tilde{\mathbf{E}}$ by solving

$$\tilde{\mathbf{E}} = \arg\min_{\mathbf{E}} \frac{1}{2m}||\mathbf{Y}_E - \Phi_1\tilde{\mathbf{X}}_1||_{\ell_2}^2 + \lambda^\star||\mathbf{X}_1||_{\ell_1}$$

8: Hence $\tilde{\mathbf{X}}_2 = \tilde{\mathbf{X}}_1 + \tilde{\mathbf{E}}$

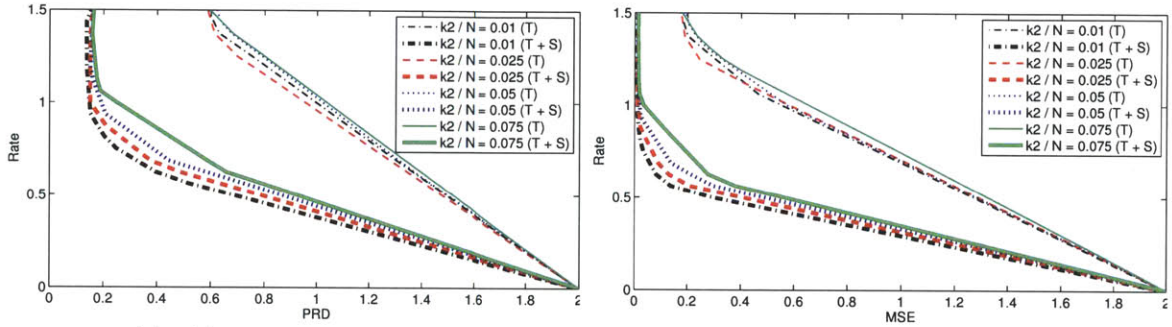
Lines 1 and 3 of Algorithm 3 corresponds to the signal received at r from source s_1 and s_2 respectively, where, as before, $\Phi_i \in \mathbb{R}^{m_i \times N}$ is generated using (2.4) and $\mathbf{Z}_i$ is a random Gaussian noise vector corresponding to the noisy channel between s_i and r. We set $m_1 \gg m_2$. The receiver then uses the LASSO decoder to obtain $\tilde{\mathbf{X}}_1$ (Line 2). Given the spatial dependency between $\mathbf{X}_1$ and $\mathbf{X}_2$, Lines 3 and 4 are equivalent for $\mathbf{Y}_2$. The noisy measurement vector $\mathbf{Y}_\mathbf{E}$ for the error $\mathbf{E}$ can thus be defined (Line 5), and decoded to obtain an estimate for the error $\mathbf{E}$ (Line 7). Line 8 shows how $\tilde{\mathbf{X}}_2$ is computed as the sum $\tilde{\mathbf{X}}_1 + \tilde{\mathbf{E}}$.

The compared performance of the two algorithms for the distributed case are shown on Figure 3-5 for a noiseless, 5%, and 10% channel noise settings.



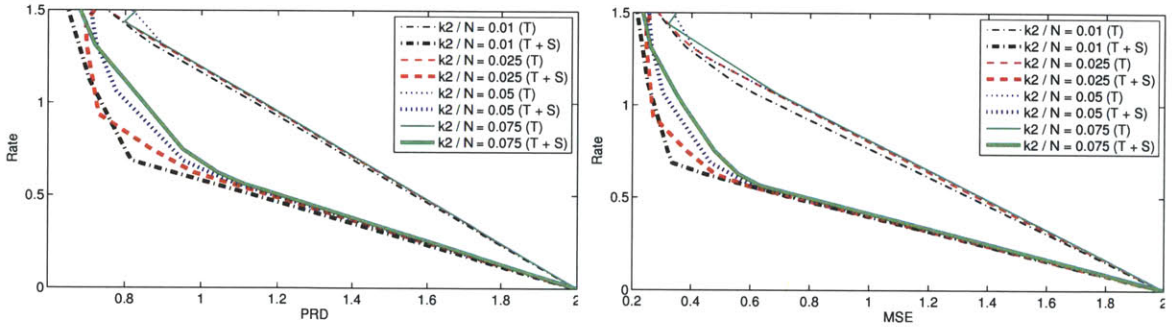
(a) Noiseless channel - PRD

(b) Noiseless channel - MSE



(c) 5% channel noise - PRD

(d) 5% channel noise - MSE



(e) 10% channel noise - PRD

(f) 10% channel noise - MSE

Figure 3-5: Distributed Case: (T) is temporal correlation only case; (T+S) is temporal and spatial correlation case

We observe that, for the noiseless channel, at a rate of 0.5, we obtain on average a factor of 2.5 times improvement when using Algorithm 3 over Algorithm 2 with PRD as a distortion measure. When using MSE, an average improvement of almost 3 times is obtained for the same setting.

When the channel is noisy, the similar average improvements at a rate of 0.5 are respectively factor of 2 times and 2.5 times for PRD and MSE. These results prove that taking advantage of the spatial and temporal correlations between the two

signals allows to achieve a much improved rate-distortion behavior.

Chapter 4

Conclusions and Future Work

In this thesis, we empirically evaluated the rate-distortion behavior of a joint source-channel coding scheme, based on compressive sensing for both a point-to-point channel and a distributed setting.

We first proposed an efficient algorithm to choose the ℓ_1-norm regularization parameter λ from the Least Absolute Shrinkage and Selection Operator, which we used as a compressive sensing decoder. This algorithm, which combines cross-validation and modified bisection, offers a reasonable trade-off between accuracy and computation time.

Using the values of λ obtained with this algorithm, we characterized the rate-distortion behavior of the joint source-channel scheme in a point-to-point channel using two distortion measures and showed that there exists an optimal sampling rate above which the distortion remains relatively constant, and below which it degrades sharply.

We then studied a single-hop network with two spatially and temporally correlated sparse sources and a receiver which uses compressive sensing decoders to reconstruct the source signals. We observed the effect of these signal correlations on the rate-distortion behavior of the scheme and showed that taking both spatial and temporal correlation in consideration allows us to achieve on average a factor of 2.5 times improvement in rate-distortion compared to only taking temporal correlation.

Future work The next immediate step following this research would be to include the random linear network coding part of the scheme to obtain a full rate-distortion analysis of the original scheme of [8]. Still in terms of rate-distortion study, another lead for future work could be the theoretical verification of the linear relation between sparsity and rate for CS observed in Figures 3-2 and 3-3.

It would also be interesting to test the joint source-channel-network coding scheme, as well as the cross-validation method on more natural signals, such as ECG signals and other medical signals in order to not only see how both performs, but also to compare the associated rate-distortion behavior to our empirical study.

Finally, we focused on using the LASSO as a CS decoder. However, it would be interesting to investigate other reconstruction algorithms among the greedy and combinatorial ones which are used for CS reconstruction. The performance of LASSO as opposed to these alternative techniques could then be evaluated.

Appendix A

Selected Matlab code

A.1 Cross-validation with modified bisection method

```
1  function [λ_star, results] = crossValidation(m, noise_percentage, ...
       sparsity_ratio)
2  % Returns the value of λ that is output by the cross validation
3  % combined with bisection when looking for values of λ.
4
5  % Default values
6  N = 1024;
7  maxIterations = 16;
8  λ_initial = 1;
9
10 % Create A: total measurement matrix (Rademacher matrix)
11 A = sign(rand([m,N]) - 0.5);
12 ind = find(A == 0);
13 A(ind) = ones(size(ind));
14 A = (1/sqrt(m)) * A;
15
16 % Generate and normalize cross-validation matrices
17 cv = round(0.1 * m);
18 tr = m - cv;
19
20 training = A(1 : tr, :);
21 testing = A(tr + 1:  end, :);
22
23 training = training ./ norm(training(:, 1), 2);
24 testing = testing * sqrt(cv / tr) / norm(testing(:, 1), 2);
25
26 % Generate sparse signal
27 xs = zeros(N, 1);
28 k = round(sparsity_ratio * N);
29
30 for i = 1 : k,
```

```
31      position = randi(N);
32      sig = 2 * round(rand()) - 1;
33      magnitude = randi(2)/2;
34      xs(position) = sig * magnitude;
35  end
36
37  % Calculate noise variance
38  transmissionPower = sum(sum(xs.^2)) / N;
39  noiseVar = noise_percentage * transmissionPower;
40
41  % Generate training and cross-validation data
42  y_tr = training * xs + randn(tr, 1) * sqrt(noiseVar);
43  noiseVarPerMeas = noiseVar / length(y_tr);
44  y_cv = testing * xs + randn(cv, 1) * sqrt(cv * noiseVarPerMeas);
45
46  % Algorithm
47  λ = λ_initial;
48  results = zeros(1, maxIterations);
49  λs = zeros(1, maxIterations);
50
51  for j = 1 : maxIterations,
52      fprintf('.');
53      λs(j) = λ;
54
55      cvx_clear;
56      cvx_begin 'quiet';
57          variable x(N);
58          minimize(1/2 * pow_pos(norm(training*x - y_tr), 2) +  ...
59              λ * norm(x, 1));
60      cvx_end;
61
62      results(j) = norm(y_cv - testing*x, 2);
63      λ = λ / 1.5;
64  end
65
66  [¬, i] = min(results);
67  λ_star = λs(i);
68
69  return
```

A.2 Single CS run for point-to-point channel

```
1  function [results, x_o, x_r] = singleRun(m, k, λ, ...
       measNoisePercentage, N, signalModel, signalNoise, measModel)
2  % Returns the PRD and MSE between the original and the reconstructed
3  % signals - for a point-to-point channel.
4  % N      - size of signal
5  % m      - number of measurements
```

```
% k      - signal sparsity

% Generate simulations data
[A, b, x_o] = generateData(m, N, k, measModel, signalModel, ...
    signalNoise, measNoisePercentage);

% CVX - LASSO
cvx_begin 'quiet';
    variable x_r(N);
    minimize(1/2 * pow_pos(norm(A*x_r - b), 2) +  λ * norm(x_r, 1));
    cvx_end;

% Computing results:
results = zeros(4, 1);

% * PRD calculations
prd = norm(x_r - x_o, 2) / norm(x_o, 2);
prd_avg = prd / N;

% * MSE calculation
energy = sum(sum((x_r - x_o).^2)) / sum(sum((x_o).^2));


results(1) = prd;
results(2) = prd_avg;
results(3) = energy;
results(4) = λ;

end
```

A.3 Single CS run for distributed setting

```
function [results, X1, X1_r, X2, X2_r, E] = ...
    singleDistributedRun(N, m1, m2, k1, k2, distcase, λ1, λ2, ...
    measNoisePercentage, signalModel, signalNoise, measModel)
% Returns the PRD and MSE between the original and the reconstructed
% signals - for a point?to?point channel.
% N - size of signal
% m - number of measurements
% k1 - sparsity of signal from source s_1
% k2 - sparsity of error signal
% λ1 - optimal λ for X_1
% λ2 - optimal λ for X_2, which is either calculated for a
% sparsity level of k1 + k2 (distributed case 1) or k2 ...
    (distributed case 2).
% distcase - if 1: only temporal correlation is considered; if 2: ...
    both temporal and spatial correlations are considered.

```

```
% Generate simulations data: first generate compressed X1 (signal ...
    from source 1)
k1N = round(k1 * N);
[A1, b1, X1] = generateData(m1, N, k1N, measModel, signalModel, ...
    signalNoise, measNoisePercentage);

% Generate the error signal and X2 (signal from source 2)
E = zeros(N, 1);
k2N = round(k2 * N);

for i = 1 : k2N,
    % Pick random position & sign
    % Magnitude = 0.5 wp 0.5; 1 with 0.5
    position = randi(N);
    sig = 2 * round(rand()) - 1;
    magnitude = randi(2)/2;
    E(position) = sig * magnitude;
end

X2 = X1;
X2 = X2 + E;

% Generate measurement matrix and noisy signal for X2
A2 = sign(rand([m2, N]) - 0.5);
ind = find(A2 == 0);
A2(ind) = ones(size(ind));
A2 = (1/sqrt(m2))  * A2;

b2 = A2 * X2;
transPower = sum(sum(X2.^2)) / N;
measNoise2 = sqrt(measNoisePercentage * transPower);
b2 = b2 + randn(m2, 1) * measNoise2;

% For either distributed case, we always want to decompress X1
cvx_begin 'quiet';
    variable X1_r(N);
    minimize(1/2 * pow_pos(norm(A1*X1_r - b1), 2) +  λ1 * ...
        norm(X1_r, 1));
cvx_end;

switch distcase
    case 1, % Only temporal correlation

        % Decompress X2
        cvx_clear
        cvx_begin 'quiet';
            variable X2_r(N);
            minimize(1/2 * pow_pos(norm(A2*X2_r - b2), 2) +  λ2 * ...
                norm(X2_r, 1));
        cvx_end;

    case 2, % Both temporal and spatial correlation are considered

        % Decompress error to obtain X2_r
```

```
63          c = b2 - A2 * X1_r;
64          cvx_clear
65          cvx_begin 'quiet';
66              variable E_r(N);
67              minimize(1/2 * pow_pos(norm(A2*E_r - c), 2) +  λ2 * ...
                    norm(E_r, 1));
68              cvx_end;
69
70          X2_r = X1_r + E_r;
71
72  end
73
74  % Computing results:
75  results = zeros(5, 1);
76
77  % PRD calculations
78  prd1 = norm(X1_r - X1, 2) / norm(X1, 2);
79  prd2 = norm(X2_r - X2, 2) / norm(X2, 2);
80
81  prd = prd1 + prd2;
82  prd_avg = prd / N;
83
84  % Average energy
85  energy1 = sum(sum((X1_r - X1).^2)) / sum(sum((X1).^2));
86  energy2 = sum(sum((X2_r - X2).^2)) / sum(sum((X2).^2));
87
88  energy = energy1 + energy2;
89
90  results(1) = prd;
91  results(2) = prd_avg;
92  results(3) = energy;
93  results(4) = λ1;
94  results(5) = λ2;
95  results(6) = (m1 + m2) / N; % Rate
96
97  end
```

Bibliography

[1] E. J. Candès and M. B. Wakin, "An Introduction to Compressive Sampling," *IEEE Signal Processing Magazine*, pp. 21–30, March 2008.

[2] A. Schulz, L. Velho, and E. da Silva, "On the Empirical Rate-Distortion Performance of Compressive Sensing," in *2009 16th IEEE International Conference on Image Processing (ICIP 2009)*, November 2009, pp. 3049–3052.

[3] W. Dai, H. V. Pham, and O. Milenkovic, "Distortion-Rate Functions for Quantized Compressive Sensing," in *2009 IEEE Information Theory Workshop on Networking and Information Theory (ITW 2009)*, June 2009, pp. 171–175.

[4] B. Mulgrew and M. Davies, "Approximate Lower Bounds for Rate-Distortion in Compressive Sensing Systems," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011)*, no. 3849-3852, April 2008.

[5] J. Chen and Q. Liang, "Rate Distortion Performance Analysis of Compressive Sensing," in *2011 IEEE Global Telecommunications Conference (GLOBECOM 2011)*, 2011, pp. 1–5.

[6] A. K. Fletcher, S. Rangan, and V. K. Goyal, "On the Rate-Distortion Performance of Compressive Sensing," in *2007 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, vol. 3, April 2007, pp. 885–888.

[7] F. Wu, J. Fu, Z. Lin, and B. Zeng, "Analysis on Rate-Distortion Performance of Compressive Sensing for Binary Sparse Source," in *Data Compression Conference*, March 2009, pp. 113–122.

[8] S. Feizi and M. Médard, "A Power Efficient Sensing/Communication Scheme: Joint Source-Channel-Network Coding by Using Compressive Sensing." Annual Allerton Conference on Communication, Control, and Computing, 2011.

[9] L. D. Davisson, "Rate-Distortion Theory and its Applications," *Proceedings of the IEEE*, vol. 60, no. 7, pp. 800–808, July 1972.

[10] F. Chen, F. Lim, O. Abari, A. Chandrakasan, and V. Stojanović, "Energy-Aware Design for Compressed Sensing Systems for Wireless Sensors under Performance and Reliability Constraints," *to be published*, 2011.

[11] G. Raskutti, M. Wainwright, and B. Yu, "Restricted Eigenvalue Properties for Correlated Gaussian designs," *The Journal of Machine Learning Research*, vol. 11, pp. 2241–2259, 2010.

[12] M. Davenport, M. Duarte, Y. Eldar, and G. Kutyniok, *Compressed Sensing: Theory and Applications.* Cambridge University Press, 2012.

[13] S. Mallat and Z. Zhang, "Matching Pursuit with Time-Frequency Dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[14] S. Mallat, "A Theory for Multiresolution Signal Decomposition: the Wavelet Representation," *IEEE Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[15] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Transactions on Computers*, vol. 23, no. 1, pp. 90–93, January 1974.

[16] D. Achlioptas, "Database-friendly random projections: Johnson-Lindenstrauss with binary coins," *Journal of Computer and System Sciences*, vol. 66, no. 4, pp. 671–687, 2003.

[17] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A Simple Proof of the Restricted Isometry Property for Random Matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, 2008.

[18] R. Tibshirani, "Regression Shrinkage and Selection via the LASSO," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[19] R. Ward, "Compressed Sensing with Cross Validation," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 5773–5782, December 2009.

[20] P. Boufounos, M. F. Duarte, and R. G. Baraniuk, "Sparse Signal Reconstruction from Noisy Compressive Measurements using Cross Validation," *IEEE/SP 14th Workshop on Statistical Signal Processing*, pp. 299–303, 2007.

[21] P. Refailzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia of Database Systems*, pp. 532–538, 2009.

[22] B. Efron, J. Johnstone, I. Hastie, and R. Tibshirani, "Least Angle Regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[23] R. L. Burden and J. D. Faires, *Numerical Analysis.* PWS Publishers, 1985.

[24] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 1.21," http://cvxr.com/cvx, April 2011.

[25] ——, "Graph Implementations for Nonsmooth Convex Programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes on Control and Information Sciences, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.