OPTIMIZATION TECHNIQUES FOR SOLVING BASIS PURSUIT PROBLEMS

By

Kristen Michelle Cheman

A Thesis Submitted to the Graduate

Faculty of North Carolina State University

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF APPLIED MATHEMATICS

Approved:

Dr. Kartik Sivaramakrishnan, Thesis Adviser

Dr. Ralph Smith, Member

Dr. Ernie Stitzinger, Member

North Carolina State University
Raleigh, North Carolina

March 29, 2006
(For Graduation May 2006)

OPTIMIZATION TECHNIQUES FOR SOLVING BASIS PURSUIT PROBLEMS

By

Kristen Michelle Cheman

An Abstract of a Thesis Submitted to the Graduate

Faculty of North Carolina State University

in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF APPLIED MATHEMATICS

The original of the complete thesis is on file
in the Department of Mathematics

Examining Committee:

Dr. Kartik Sivaramakrishnan, Thesis Adviser

Dr. Ralph Smith, Member

Dr. Ernie Stitzinger, Member

North Carolina State University
Raleigh, North Carolina

March 29, 2006
(For Graduation May 2006)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I would like to express my gratitude to my project advisor, Dr. Kartik Sivaramakrishnan, whose expertise and understanding added considerably to my graduate experience. I appreciate his vast knowledge and skill in the areas of mathematics and optimization, and his patience in helping me during the completion of my project.

I would like to thank the other members of my committee, Dr. Ralph Smith and Dr. Ernie Stitzinger, for their words of wisdom and professional guidance.

I would like to thank my family for the love and support they provided during graduate school, and always. I must also acknowledge Lisa Bieryla and Amy Oliver, without whom I could not have finished this thesis, much less my Master's coursework.

I would like to express my appreciation to Denise Seabrooks and Brenda Currin for all their work behind the scenes.

Finally, I must recognize that my graduate experience and this research would not have been possible without the financial assistance of the Mathematics Department at North Carolina State University in the form of a generous Teaching Assistantship.

ABSTRACT

Signal processing is important for data reduction and signal denoising. Given a signal $b \in \mathbb{R}^m$, we seek to represent b in fewer than m atoms from an overcomplete dictionary A. This paper will analyze the Basis Pursuit (BP) and Method of Frames (MOF) methods for achieving signal representations. A noisy version of the problem will be presented, where we aim to retrieve a signal b from a signal b', which contains unwanted data.

CHAPTER 1

1.1 Introduction

Signal processing allows us to analyze and manipulate analog and digital signals, including audio, video, image, sonar, radar, and a number of other signals. This versatility is more and more important to us as our society aims to do things faster, cheaper, and in a more efficient manner. Consider the following application of signal processing:

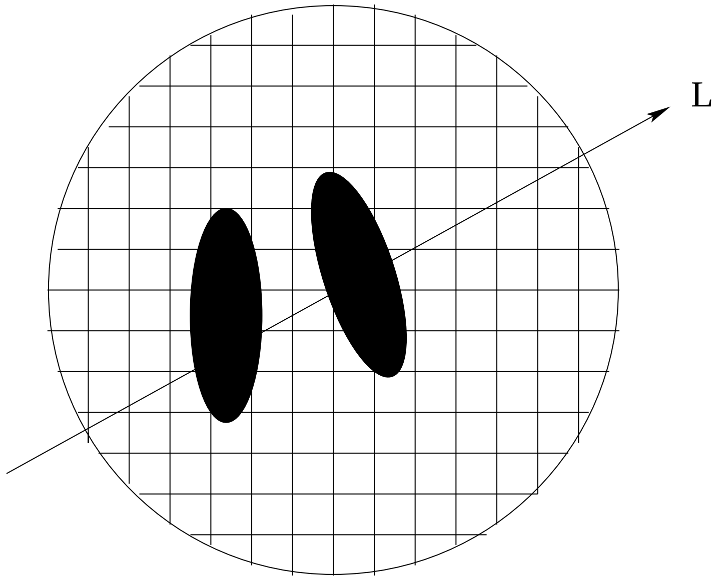
Today's medical technicians use *computerized axial tomography (CAT scan)* to make diagnoses regarding the presence of tumors and other cell abnormalities. A beam is sent at different angles through a suspicious part of the body in order to produce a three-dimensional picture of the internal organ. Mathematically, we can describe the detection process as follows. Let $S \subset \mathbb{R}^3$ be a particular part of the body. Assume that S is made up of a collection of smaller cells S_i, $i = 1, \ldots, n$. Let x_i be the density within cell S_i and assume that the density is constant throughout the entire cell. We want to determine the values x_i.

A beam L is sent through the set S and intersects several of the cells along the way as illustrated in Figure 1.1. Let $I(L)$ be the set of cells that are intersected by L. For all $i \in I(L)$, we know the length $a_i(L)$ of the path of the beam within that cell. Note that the cells absorb some of the energy of the beam. Since energy absorbed by a cell is proportional to its density and the length of the path traversed by the beam in the cell, the amount of energy absorbed in this case is $\sum_{i \in I(L)} a_i(L)x_i$. In addition, given the beam's initial energy and measurements of its energy at the exit point, we can estimate the energy absorbed $b(L)$. We have the relationship $\sum_{i \in I(L)} a_i(L)x_i = b(L)$. We repeat the experiment with m beams L_j, $j = 1, \ldots, m$ to get:

$$\sum_{i \in I(L_j)} a_i(L)x_i = b(L_j), \; j = 1, \ldots, m. \tag{1.1}$$

Since x_i represents the density of cell material, we have $x_i \geq 0$, $i = 1, \ldots, n$. Because

Figure 1.1: A model of the CAT Scan procedure.



of the cost of x-rays and their danger to a patient, the number of beams m is not usually equal to the number of cells n. Normally, the system has too few equations (i.e. $m < n$) to have a unique solution x. However, it is also possible that the system has too many or inconsistent equations. It is important to note that the model is only an approximation and it is unlikely to find a solution that perfectly states what is happening in the body.

There are several approximations in (1.1); for instance, we generally do not know the right hand side $b(L_j)$, $j = 1, \ldots, m$. We will find instead an $x \in \mathbb{R}^n$ that minimizes the residual:

$$f(x) = \sum_{j=1}^{m} \left(b(L_j) - \sum_{i \in I(L_j)} a_i(L_j) x_i \right)^2 .$$

Thus, our problem becomes:

$$\text{Minimize } \sum_{j=1}^{m} \left(b(L_j) - \sum_{i \in I(L_j)} a_i(L_j) x_i \right)^2 \text{ subject to } x \geq 0.$$

To simplify the analysis, one also attempts to find a solution x that is as sparse as possible. This can be achieved by minimizing $\sum_{j=1}^{n} x_j$ since $x_j \geq 0$. Our problem becomes:

$$\text{Minimize } \sum_{j=1}^{m} \left(b(L_j) - \sum_{i \in I(L_j)} a_i(L_j) x_i \right)^2 + \gamma \sum_{j=1}^{n} x_j$$
$$\text{subject to } x \geq 0,$$

where $\gamma > 0$ is a parameter. Note that we have a multiobjective optimization problem where we must minimize two terms, which are both quadratic. The scalar γ is chosen to achieve a suitable tradeoff between these objectives. This gives rise to a *quadratic programming problem*. (See Ruszczynski [5].)

The tumor detection problem that we have described involves several areas of signal processing. Minimizing the sum of the components of x corresponds to "data reduction" and minimizing the residual corresponds to "signal denoising." We will develop techniques to tackle both of these problems in this thesis.

Data reduction and signal denoising problems both come with several requirements. First, we have to know a little bit about the signal that we are dealing with. Second, both problems require time and memory storage, which can be expensive. Our goal is to achieve a balance between cost and benefit.

Signal processing methods have become more effective and more efficient over the years. In [2], Donoho compares traditional Fourier transformation methods for representing a signal in terms of sinusoids with different frequencies and amplitudes to more recent methods. While the traditional methods of dealing with signal processing problems are useful in analyzing new methods, the alternatives provide faster, more effective results. (See [2].)

In this thesis, we will compare several optimization methods that achieve data and noise reduction. Our emphasis is on data reduction and we will discuss it first in the thesis.

1.2 Data Reduction

When performing data reduction, we hope to express a signal b using less information. So, if our signal has length m, we'd like to compress the amount of information necessary to reproduce the signal to be something much less than m. To do this, we start with a dictionary A that is a collection of n atoms, each of length m. The atoms contained in the dictionary are pieces of information about signals similar to the one we would like to represent. Most dictionaries, and all the dictionaries that we will consider, are *overcomplete*, i.e. they are made up of more than m atoms. Overcompleteness results from redundancies that occur when several dictionaries are combined. Our job is to express the signal b with as few atoms as possible from the dictionary A [2].

1.3 Goals

Since our dictionary A is overcomplete and contains redundancies, we are able to represent b as a combination of atoms from A in an infinite number of ways. Since this is the case, we must establish a criteria for finding a *best* representation. We have the following objectives in mind:

1. We want to produce the *sparsest* representation of the original signal that is possible, i.e. one that uses the fewest number of atoms from the dictionary.

2. We want to solve the problem in as little time as possible.

Clearly, each of our goals can be accomplished to different degrees. However, we would like to achieve each of them simultaneously without having to sacrifice too much of one to get another [2].

1.4 Previous work on data reduction

Signal representation and its applications have been studied for nearly a century. In Tropp [6], more recent work from the past decade is presented, where sparse signal representations have been used for the compression and analysis of audio, image, and video data. In [2], several different methods for achieving sparse signal representations are considered, which have been developed in the past few years as alternatives to traditional techniques. In particular, Donoho discusses the Matching Pursuit (MP) method, which is a stepwise approach to finding a sparse solution to a system of equations, as well as the Best Orthogonal Basis (BOB) method, which tailors the problem solving technique to the particular dictionary that is involved. He compares these methods to Basis Pursuit (BP) and Method of Frames (MOF), which find a sparse solution using the 1-norm and 2-norm, respectively. A comparative study of each of these techniques appear in [2]. We will compare BP and MOF in detail in Chapter 2.

1.5 Organization of Thesis

In this paper, we will compare BP and MOF techniques by analyzing an artificial signal from Donoho's Atomizer toolbox for MATLAB. We will create our own code for performing these methods, as well as use Donoho's code from Atomizer for this purpose.

The second chapter of our thesis is organized as follows. Section 1 describes a mathematical model for our problem. Section 2 discusses the details of BP and MOF techniques for sparse representation of signals. Section 3 considers noisy problems where we combine our techniques in Section 2 with signal denoising. Section 4 presents the results of our computational experiments with the various techniques. We conclude with our findings in Section 5, and also discuss areas of future study.

CHAPTER 2

2.1 Introduction to the problem

Given a signal b of length m, we can think of b as a vector in $\mathbb{R}^m$. Suppose we have a dictionary of n atoms, each of length m. We can collect the atoms as columns of an m-by-n matrix A. Assume that A is overcomplete with $m << n$. Our aim is to produce a representation of the signal by expressing it as a linear combination of atoms (i.e. columns) from A. In other words, we would like to solve the system $Ax = b$. Without loss of generality, we assume that $rank(A) = m$. (Note that if A doesn't have full row rank, (a) we can use QR factorization and throw away redundant equations to find a system that does have full row rank or (b) the system $Ax = b$ is inconsistent and has no solution.) Thus, the system $Ax = b$ is undetermined and has infinitely many solutions. (See Meyer [4].)

We are able to retrieve a best solution from among the infinite number of solutions to the system by minimizing support$(x) = \{i|x_i \neq 0\} < m$. This ensures that the amount of information necessary to represent b is less than the amount of information provided by the signal itself. Our problem becomes:

$$\text{Minimize support}(x) \text{ subject to } Ax = b.$$

Minimizing support(x) is the same as minimizing the 0-norm of x. So we have

$$\text{Minimize } ||x||_0 \text{ subject to } Ax = b. \tag{2.1}$$

Unfortunately, the 0-norm is a nonconvex function making (2.1) difficult to solve.

2.2 Methods of decomposition

Next, we will discuss two different approaches to solving (2.1). Let us consider instead the 1-norm and 2-norm, which are both more accessible than the 0-norm. First, let us look at the 1-norm.

2.2.1 Basis Pursuit

The first method that we will look at is Basis Pursuit (BP). BP finds the best representation of a signal by minimizing the 1-norm of the components of x, i.e. the coefficients in the representation. Ideally, we would like the components of x to be zero or as close to zero as possible.

We would like to solve:

$$\text{Minimize } ||x||_1 \text{ subject to } Ax = b. \tag{2.2}$$

In finding x, since the nonzero coefficients correspond to columns of the dictionary, we can use the indices of the nonzero components of x to identify the columns of A that are necessary to reproduce the signal. This collection is a basis for the representation. Using the 1-norm allows us to assign a cost to each atom that we use in our representation. For example, we won't charge the norm when it gives a zero coefficient, but we will charge it proportionally for small and large coefficients.

Because we have an additional condition to solving the system of equations, we can rewrite our problem as a linear programming problem (LP) of the form:

$$\text{Minimize } c^T x \text{ subject to } Ax = b, x \geq 0$$

where $c^T x$ is the objective function, $Ax = b$ is a collection of equality constraints, and $x \geq 0$ is a set of bounds. In our case, we will drop the condition that $x \geq 0$ and assign the selection of a sparse solution x to be our objective function.

Starting with the problem (2.2), note that $||x||_1 = |x_1| + \cdots + |x_n|$. We can rewrite the problem as:

$$\text{Minimize } |x_1| + \cdots + |x_n| \text{ subject to } Ax = b. \tag{2.3}$$

This is not an LP as yet, since the objective function is not linear. However, we can transfer the nonlinearities to the set of constraints by adding the new variables

$t_1, \ldots, t_n$. This gives:

$$\begin{aligned} \text{Minimize } & t_1 + t_2 + \cdots + t_n \\ \text{subject to } & |x_1| \leq t_1 \\ & |x_2| \leq t_2 \\ & \vdots \\ & |x_n| \leq t_n \\ & Ax = b. \end{aligned}$$

By observing that $|x_i| \leq t_i$ if and only if $-t_i \leq x_i \leq t_i$, we can transform our problem into a linear programming problem by adding the following n inequalities:

$$\begin{aligned} \text{Minimize } & t_1 + t_2 + \cdots + t_n \\ \text{subject to } & x_1 \leq t_1 \\ & x_1 \geq -t_1 \\ & x_2 \leq t_2 \\ & x_2 \geq -t_2 \\ & \vdots \\ & x_n \leq t_n \\ & x_n \geq -t_n \\ & Ax = b. \end{aligned}$$

Note that $x_i \leq t_i$ implies that $Ix \leq It$ and so $Ix - It \leq 0$. Likewise, $x_i \geq t_i$ implies $Ix + It \geq 0$. Thus, we have:

$$\begin{aligned} &\text{Minimize } e^T t \\ \text{subject to } &Ix - It \leq 0 \\ &Ix + It \geq 0 \\ &Ax = b. \end{aligned}$$

where $e = [1, 1, \ldots, 1]^T$. Finally, note that our objective function and constraints are now linear. Thus, we have rewritten our problem as an LP.

We are able to reduce the size of our problem by examining the *dual problem*. From duality theory, starting with a linear program in standard form, i.e.

$$\text{Minimize } c^T x \text{ subject to } Ax = b,\ x \geq 0,$$

we can rewrite the problem as the following dual linear program:

$$\text{Maximize } b^T y \text{ subject to } A^T y + z = c,$$

which is equivalent. (For more information on duality theory see [1].) Using this equivalence, we can rewrite our problem in terms of the dual variables y, v, and w, which correspond to the constraints from the primal problem with no restrictions on x or t. Thus we have:

$$\text{Maximize} \begin{bmatrix} b^T & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ v \\ w \end{bmatrix}$$

$$\text{subject to} \begin{bmatrix} A^T & I & I \\ 0^T & -I & I \end{bmatrix} \begin{bmatrix} y \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ e \end{bmatrix}$$

$$y \text{ unrestricted}$$

$$v \leq 0$$

$$w \geq 0.$$

which can also be written as:

$$\begin{aligned} &\text{Maximize } b^T y \\ &\text{subject to } A^T y + Iv + Iw = 0 \\ &\qquad - Iv + Iw = e \\ &\qquad v \leq 0 \\ &\qquad w \geq 0. \end{aligned}$$

Note that $v \leq 0$ implies $-v \geq 0$, so we can replace v with $-v$ to get:

$$\begin{aligned} &\text{Maximize } b^T y \\ &\text{subject to } A^T y - v + w = 0 \\ &\qquad v + w = e \\ &\qquad v \geq 0 \\ &\qquad w \geq 0 \end{aligned}$$

Then, $v + w = e$ implies $w = e - v$. Thus, we can substitute for w as follows:

$$\begin{aligned} &\text{Maximize } b^T y \\ &\text{subject to } A^T y - v + (e - v) = 0 \\ &\qquad\qquad v \geq 0 \\ &\qquad\qquad e - v \geq 0. \end{aligned}$$

Finally, we have the equivalent dual problem:

$$\text{Maximize } b^T y \text{ subject to } A^T y - 2v = -e, 0 \leq v \leq e. \tag{2.4}$$

Now we have changed our original problem, which had more constraints than variables (since $m < n$), into a problem with more variables than constraints. We have reduced the size of our problem and can use Matlab's LINPROG command to solve it. Our code for minimizing the 1-norm by solving the dual problem can be found in Appendix A.

Because (2.4) is equivalent to an LP, we can use any LP algorithm to solve it. In this paper, we will use the Interior Point Method (IPM). IMP starts with feasible iterates x^0 from the original problem and (y^0, s^0) in the dual. Beginning with (x^0, y^0, s^0), it proceeds to optimize the LP by generating iterates within the feasible region of solutions until optimality is reached. The interior iterates lie in close proximity to the "central path," which is a pathway to the optimal set of the linear program. (See Wright [8].)

The main idea of each iteration is to solve a system of linear equations of the form $(AD^2A^T)u$ = right hand side, where D^2 is a diagonal matrix whose entries are $(\frac{x_i}{s_i})$, $i = 1, 2, \ldots, n$. The system can be solved using iterative methods such as the conjugate gradient method (see [4]) without having to form AD^2A^T. (This is done in Donoho's code.) To do this, the only item we need is a blackbox that, given AD^2A^T and $u \in \mathbb{R}^m$, computes $(AD^2A^T)u$. This simplicity is one advantage of IPM over the simplex method. IPM also allows one to solve an LP to any degree of precision $\epsilon > 0$ in a polynomial number of operations. This allows us to calculate the optimal support by solving the LPs in an inexpensive manner, which is another advantage over the simplex method [8].

2.2.2 Method of Frames

Next, we will compare BP to MOF. MOF takes a similar approach to MOF, but instead minimizes the 2-norm of the coefficients. In other words, it solves:

$$\text{Minimize } ||x||_2 \text{ subject to } Ax = b. \tag{2.5}$$

By replacing the 1-norm with the 2-norm, we change a linear programming problem to a quadratic programming problem with linear constraints. In this case, we can find x by solving the constrained least squares problem: $x = A^T(AA^T)^{-1}b$. Our code for finding a sparse solution x by minimizing the 2-norm can be found in Appendix B.

2.3 Signal Denoising

Since we aren't always given a clear signal to work with, it would be nice to have some procedure for dealing with noisy data. Suppose we are given a signal $b \in \mathbb{R}^m$, which is corrupted by additional noise $z \in \mathbb{R}^m$. This leaves us with a noisy signal $b' = b + z$. We would like to use the data we are given in b' to produce an estimate $y = Ax$ of the original signal b. Note that y is a linear combination of basis elements from an overcomplete m-by-n dictionary A.

In the noisy version of our problem, we have two main goals:

1. We want our estimate y to be close to b' in a least squares sense, which will ensure that y is close to b.

2. We want y to be sparse, so it should involve as few atoms (columns) from A as possible. (As before, this implies that we should minimize $||x||_1$.)

Mathematically speaking, our goals are to minimize both $||x||_1$ and $||Ax - b||_2$. We can transform this multiobjective optimization problem into a regularized problem by solving:

$$\text{Minimize } \gamma||x||_1 + ||Ax - b||_2, \tag{2.6}$$

where $\gamma > 0$ is a parameter that we can vary over the interval $(0, \infty)$ in order to trade off between the two objectives.

Fix $\gamma > 0$. We can rewrite (2.6) as:

$$\text{Minimize } \gamma||x||_1 + \frac{1}{2}{||Ax - b||_2}^2$$

or

$$\text{Minimize } \gamma||x||_1 + \frac{1}{2}p^T p \text{ subject to } Ax - Ip = b,$$

where I is the m-by-m identity matrix and $p = Ax - b$.

Note that $||x||_1 = \sum_{i=1}^{n} |x_i|$. Let $x_i = u_i - v_i$ for $u_i, v_i \geq 0$, $i = 1, \ldots, n$. Then $|x_i| = u_i + v_i$ and $||x||_1 = \sum_{i=1}^{n} |x_i| = \sum_{i=1}^{n}(u_i + v_i)$. Substituting these values for x, we have:

$$\begin{aligned} \text{Minimize } & \sum_{i=1}^{n}(u_i + v_i) + \frac{1}{2}p^T p \\ \text{subject to } & A(u - v) - Ip = b \\ & u \geq 0 \\ & v \geq 0, \end{aligned}$$

where $u, v \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$.

This gives:

$$\begin{aligned}
&\text{Minimize } e^T u + e^T v + \frac{1}{2} p^T p \\
&\text{subject to } \begin{bmatrix} A & -A & -I \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = b \\
&u \geq 0 \\
&v \geq 0,
\end{aligned}$$

i.e.

$$\begin{aligned}
&\text{Minimize } \frac{1}{2} \begin{bmatrix} p & u & v \end{bmatrix}^T \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix} + \begin{bmatrix} 0 & e^T & e^T \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix} \\
&\text{subject to } \begin{bmatrix} -I & A & -A \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix} = b \\
&u \geq 0 \\
&v \geq 0.
\end{aligned}$$

The noisy problem for MOF is similar and involves solving:

$$\text{Minimize } \gamma^2 ||x||_2 + \frac{1}{2} ||Ax - b||_2{}^2.$$

In either case, when solving for x, we can produce our representation $y = Ax$. Our code for the signal denoising problem can be found in Appendix C.

2.4 Computational results

We will analyze the data reduction and signal denoising problems using BP and MOF. To do this, we will use the WaveLab and Atomizer software for MATLAB from [2]. WaveLab is a library of MATLAB routines for wavelet, wavelet packet, and cosine packet analysis. Atomizer contains a collection of dictionaries and artificial signals. It borrows routines from WaveLab and includes codes for several different methods for finding signal representations in overcomplete dictionaries. These programs are available for download at http://www-stat.stanford.edu/∼atomizer/ and http://www-stat.stanford.edu/∼wavelab/.

2.4.1 Analysis of data reduction

First, we will look at the data reduction problem. Recall that we would like to solve the system $Ax = b$, for a particular dictionary A and signal b, by producing a sparse solution x. We can use either BP and MOF, which utilize the 1-norm and 2-norm, respectively, to achieve this goal.

We will look at the artificial signal TwinSine1 from the Atomizer toolbox. TwinSine1 is the sum of two cosines with frequencies that are close to one another. In particular, TwinSine1 is the function:

$$b = \sqrt{\frac{2}{m}} \left[\cos\left(\frac{(\frac{m}{2} - 2) - 1}{4} \cdot \pi \cdot t \right) + \cos\left(\frac{(\frac{m}{2} + 2) - 1}{4} \cdot \pi \cdot t \right) \right], \tag{2.7}$$

where t varies between $\frac{1}{m}$ to 1 (in steps of $\frac{1}{m}$). We choose $m = 256$. A plot of TwinSine1 can be seen in Figure 2.1.

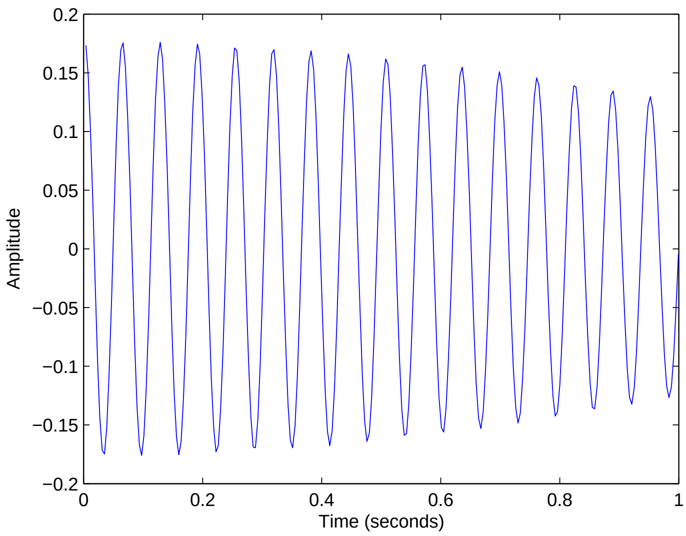
We will decompose the signal in terms of atoms from a fourfold overcomplete discrete cosine transform (DCT) dictionary. DCT is an example of a frequency dictionary. It converts data into sets of frequencies and compresses the data by deleting the frequencies that are less meaningful [3]. An overcomplete dictionary is obtained by sampling the frequencies more finely [2]. In this case, we will work with a dictionary that is four times finer than the original dictionary. The dictionary elements are:

$$\sqrt{\frac{2}{m}} \cdot \cos\left(2 \cdot \pi \cdot \frac{k}{m * l} \right), \; k = 0, 1, \ldots, \left(\frac{m * l}{2} + 1 \right)$$

(the odd columns in A) and

$$\sqrt{\frac{2}{m}} \cdot \sin\left(2 \cdot \pi \cdot \frac{k}{m * l} \right), \; k = 1, \ldots, \left(\frac{m * l}{2} \right)$$

Figure 2.1: The graph of TwinSine1.



(the even columns in A), where $l = 4$ is the fineness. Since $m = 256$ and $l = 4$, the matrix A has $n = m \cdot l = 256 \cdot 4 = 1024$ columns.

Using the BP and MOF routines from WaveLab, we are able to produce representations of TwinSine1. Our code for executing BP and MOF on the TwinSine1 signal can be found in Appendix C. Recall that for BP we solve:

$$\text{Minimize } ||x||_1 \text{ subject to } Ax = b.$$

By default, BP solves for x with one digit of accuracy. We found x in this way, as well as three and six digits of accuracy, in order to ensure a more sparse result. In the MOF problem, we swap the 1-norm with the 2-norm and solve:

$$\text{Minimize } ||x||_2 \text{ subject to } Ax = b.$$

Figure 2.2 shows (a) the graph of TwinSine1, (b) the coefficient vector x found using BP with six digits of accuracy, and (c) the coefficient vector x using MOF. Table 2.4.1 shows the amount of time that it took each routine to run, as well as the support data for both BP and MOF for the TwinSine1 signal using a fourfold overcomplete DCT dictionary. The support was calculated by counting the number

Figure 2.2: Comparison of BP and MOF representations for TwinSine1.

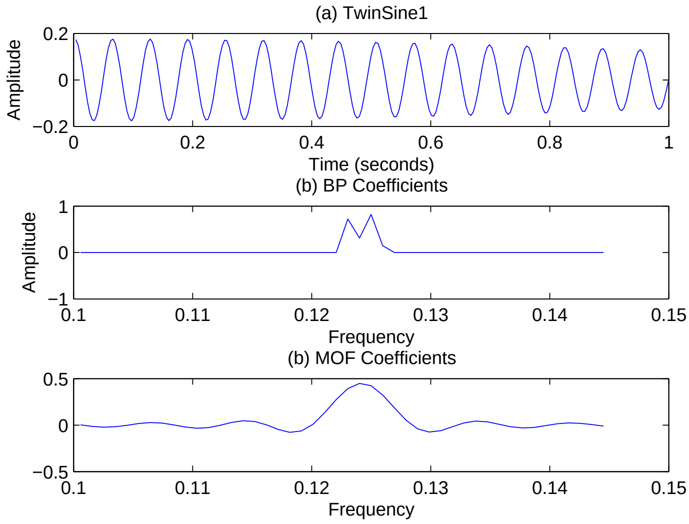


Table 2.1: Time and support data for TwinSine1.

Method	Time (seconds)	Support
BP (1 digit)	1.272	18
BP (3 digit)	5.408	11
BP (6 digit)	9.113	2
MOF	.110	881

of components $x_i,\ i = 1, \ldots, n$ such that $|x_i| > 1e - 4$. (See [2].)

Since we know ahead of time what b = TwinSine1 and our dictionary A look like, we know that only two atoms from A are necessary to represent the signal. From Table 2.4.1, note that MOF produces a speedy representation of b, but that the coefficient vector has a much larger support than the coefficient vector for BP. This implies that the representation found by MOF is not optimal. BP produces a sparser result than MOF. The sparsity of the coefficient vector that is produced by BP increases as we increase the accuracy of the algorithm. Note that when the accuracy of BP is increased to 6 decimal places, it correctly finds the two discrete cosine dictionary elements of frequencies $\frac{((\frac{m}{2}-1)-2)}{4} \cdot \pi$ and $\frac{((\frac{m}{2}+2)-1)}{4} \cdot \pi$ in A, which correspond to the optimal support of size two.

2.4.2 Analysis of signal denoising

In the noisy version of the problem, we have a signal $b = \text{TwinSine2}$ to which we add noise using the NoiseMaker function from Atomizer. The noisy TwinSine2 function is:

$$b' = \sqrt{\frac{2}{m}}\left[\cos\left(\frac{(\frac{m}{2}-3)-1}{4}\cdot\pi\cdot t\right)+\cos\left(\frac{(\frac{m}{2}-1)-1}{4}\cdot\pi\cdot t\right)\right], \tag{2.8}$$

where t varies between $\frac{1}{m}$ to 1 (in steps of $\frac{1}{m}$). We choose $m = 128$. A plot of TwinSine2 and noisy TwinSine2 are shown in Figure 2.3 (a) and (b), respectively. We assume that we know b', not b, which is the signal that we would like to represent. We will still work with the same fourfold overcomplete DCT dictionary A.

We would like to use data from b' to produce an estimate $y = Ax$ of the original signal b. We will use BP and MOF to do this. We alter the BP technique from before to solve instead:

$$\text{Minimize } \frac{1}{2}(||b' - Ax||_2)^2 + \gamma(||x||_1).$$

We will perform BP to 1, 3, and 6 digits of accuracy. We also change MOF to solve:

$$\text{Minimize } \frac{1}{2}(||b' - Ax||_2)^2 + \gamma^2(||x||_2)^2.$$

Our code for signal denoising using BP can be found in Appendix D.

Recall that signal denoising involves a multiobjective optimization problem. First, we will discuss our pursuit of a sparse representation. This is controlled by the term $||x||_1$ in BP and the term $(||x||_2)^2$ in MOF. Figure 2.3 (c) and (d) shows the graphs of the coefficient vectors retrieved by BP and MOF, respectively. Clearly, the representation obtained by MOF is a poor one, since the amplitude of the corresponding coefficient vector is nearly zero. On the other hand, BP appears to retrieve exactly the two atoms from A that make up b.

Our next goal was to achieve an approximation $y = Ax$. To do this, we minimize $(||b' - Ax||_2)^2$. A comparison of y_{BP} (the approximation obtained by BP) and y_{MOF} (the approximation obtained by MOF) is shown in Figure 2.4. It is

Figure 2.3: Denoising noisy TwinSine2 with a fourfold overcomplete discrete cosine dictionary

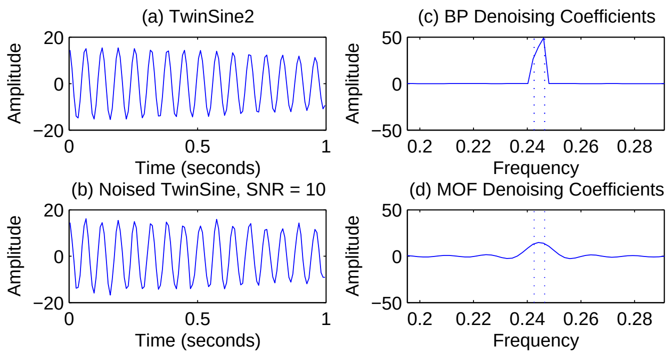
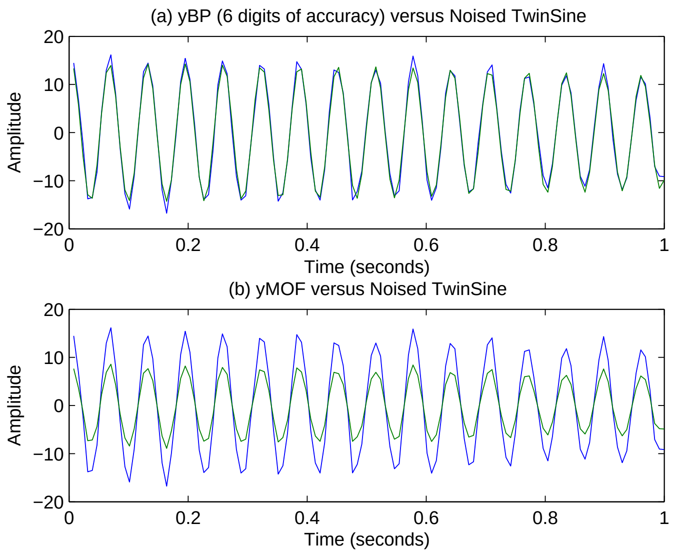


Figure 2.4: A comparison of $y = Ax$ and b'.



obvious that BP produces the nearest approximation to the original signal.

Our results are supported by the data in Table 2.4.2. Note that while MOF is a faster technique, BP obtains a sparser result for all three variations of accuracy. BP with 1 digit of accuracy achieves the smallest residual $||b' - y||_2$, which indicates

Table 2.2: Time, support, and residual data for Noisy TwinSine2.

Method	Time (seconds)	Support	$\|\|b' - y\|\|_2$
BP (1 digit)	.33048	486	11.412
BP (3 digit)	.44063	5	12.127
BP (6 digit)	.50072	2	12.126
MOF	.030043	512	52.804

that it produces the nearest approximation of b'. Thus, it produces the nearest approximation of b. However, the support of the corresponding coefficient vector makes the representation a useless one. BP with 6 digits of accuracy produces a closer approximation and smaller support than BP with 3 digits of accuracy; however, if time is a factor, BP with 3 digits yields a close enough result in less time.

2.5 Conclusion

We have seen that, when there is a sparse solution to a system $Ax = b$, BP will usually find it. As we increase the accuracy of BP, we increase the sparsity of the resulting coefficient vector x. For data reduction, we can achieve a best approximation by choosing BP over MOF. For signal denoising, we must make a compromise between sparsity and the error in making the approximation. BP achieves both of these goals simultaneously.

In the future, one might look at problems larger than those explored in this thesis. Furthermore, one might use BP to look at multi-dimensional signals in addition to the one-dimensional signals seen here. Higher dimensions lead to larger optimization problems, which might require refinements to BP techniques or the creation of new techniques altogether. Finally, because of the connection between BP and LP, one might consider developing new LP techniques as a topic of future study.

LITERATURE CITED

[1] V. Chvátal, *Linear Programming*, W.H. Freeman, New York, 1983.

[2] S. Chen, D. Donoho, and M. Saunders, *Atomic Decomposition by Basis Pursuit*, SIAM Review, 43 (2001), pp. 129-159.

[3] *Discrete Cosine Transform*, http://www.answers.com/topic/discrete-cosine-transform.

[4] C. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.

[5] A. Ruszczynski, *Nonlinear Optimization*, Princeton University Press, 2006.

[6] J. Tropp, *Topics in Sparse Approximation*, Ph.D. Thesis, Department of Computational and Applied Mathematics, University of Texas, Austin, TX, August 2004; see also http://www-personal.umich.edu/∼jtropp/.

[7] *Noise Reduction*, http://en.wikipedia.org/.

[8] S. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.

[9] Y. Ye, *An extension of Karmarkar's algorithm and the trust region method for convex quadratic programming*, Progress in Mathematical Programming (N. Megiddo ed.), Springer-Verlag, New York 1989.

[10] Y. Ye, *On affine-scaling algorithm for nonconvex quadratic programming*, Mathematical Programming, 1992.

APPENDIX A
MATLAB code for minimizing the 1-norm of x

Below is the code for solving

$$\text{Minimize } ||x||_1 \text{ subject to } Ax = b \tag{A.1}$$

by using the dual problem:

$$\begin{aligned} &\text{Maximize } b^T y \\ &\text{subject to } A^T y + 2v = e \\ &\qquad 0 \le v \le e, \end{aligned}$$

where $e = [1, 1, \ldots, 1]^T$. Given a dictionary A and signal b, we can find a sparse solution x as follows:

```
% x :- desired sparse solution
% support :- support set for vector x
% lsupport :- cardinality of the support set

[m,n] = size(A);
Amatrix = [A' 2*eye(n)];
bvector = ones(n,1);
cvector = [-b; zeros(n,1)];
lb = [-inf*ones(m,1); zeros(n,1)];
ub = [inf*ones(m,1); ones(n,1)];

% track the time required to solve the dual problem
profile on
[primal,obj,exitflag,output,dual] = linprog(cvector,[],[],Amatrix,bvector,lb,ub);
profile viewer
% x is the dual variable corresponding to the equality constraints in dual problem
```

```
x = dual.eqlin;

% Let us compute the support set of x
support = find(abs(x)/max([1 normest(A) norm(b)]) > 1e-8);
% Compute the cardinality of the support set
lsupport = length(support);
```

APPENDIX B
MATLAB code for minimizing the 2-norm of x

Below is the code norm2.m, which was used to solve the problem

$$\min ||x||_2 \text{ subject to } Ax = b \tag{B.1}$$

by solving the least squares problem $A^T Ax = A^T b$ given a dictionary A and signal b.

```
% x :- desired sparse solution
% support2 :- support set for vector x
% lsupport2 :- cardinality of the support set
%
% Min ||x||_2
% st Ax = b

profile on
y = (A*A')\b; x2 = A'*y;
profile viewer

support2 = find(abs(x2)/max([1 normest(A) norm(b)]) > 1e-8);
lsupport2 = length(support2);
```

APPENDIX C
MATLAB codes for solving data reduction problems using MOF and BP with different tolerances for accuracy

C.1 Data reduction with BP (1 digit of accuracy)

% BASIS PURSUIT (BP):

% Min $||x||_1$ subject to $Ax = b$

% via a Primal-Dual Logarithmic Barrier Interior Point method

% Use the following parameters based on the dictionary that you want to use.

m = 256; % problem size

par1 = 4; % overcompleteness

par2 = 0;

par3 = 0;

% Build the matrix basis A for the dictionary

A = MakeDict(m,'DCT',par1,par2,par3);

% Generate the input signal b

b1 = InputSignal('TwinSine1', m);

% Use Donoho's BP algorithm to find the solution x using the default 1 digit of accuracy

profile on

x1 = BP_Interior(b1,'DCT',par1,par2,par3); % change the dictionary

profile viewer

support1 = find(abs(x1) > 1e-4);

lsupport1 = size(support1)

C.2 Data reduction with BP (3 digits of accuracy)

% BASIS PURSUIT (BP):

% Min $||x||_1$ subject to $Ax = b$

% via a Primal-Dual Logarithmic Barrier Interior Point method

% Use the following parameters based on the dictionary that you want to use.

m = 256; % problem size

par1 = 4; % overcompleteness

par2 = 0;

par3 = 0;

% Build the matrix basis A for the dictionary

A = MakeDict(m,'DCT',par1,par2,par3);

% Generate the input signal b

b1 = InputSignal('TwinSine1', m);

% Increase the level of accuracy

FeaTol = 1e-3;

PDGapTol = 1e-3;

CGAccuracy = 1e-3;

% Use Donoho's BP algorithm to find the solution x

% with 3 digits of accuracy for improved resolution.

profile on

x1 = BP_Interior(b1,'DCT',par1,par2,par3,FeaTol,PDGapTol,CGAccuracy); % change the dictionary

profile viewer

support1 = find(abs(x1) > 1e-4);

lsupport1 = size(support1)

C.3 Data reduction with BP (6 digits of accuracy)

% BASIS PURSUIT (BP):

% Min $||x||_1$ subject to $Ax = b$

% via a Primal-Dual Logarithmic Barrier Interior Point method

% Use the following parameters based on the dictionary that you want to use.

m = 256; % problem size

par1 = 4; % overcompleteness

par2 = 0;

par3 = 0;

% Build the matrix basis A for the dictionary

A = MakeDict(m,'DCT',par1,par2,par3);

% Generate the input signal b

b1 = InputSignal('TwinSine1', m);

% Increase the level of accuracy

FeaTol = 1e-6;

PDGapTol = 1e-6;

CGAccuracy = 1e-3;

% Use Donoho's BP algorithm to find the solution x

% with 6 digits of accuracy for improved resolution.

profile on

x1 = BP_Interior(b1,'DCT',par1,par2,par3,FeaTol,PDGapTol,CGAccuracy); % change the dictionary

profile viewer

support1 = find(abs(x1) > 1e-4);

lsupport1 = size(support1)

C.4 Data reduction with MOF

```
% METHOD OF FRAMES using a CG solver:
% Min ||x||_2 subject to Ax = b

% Use the following parameters based on the dictionary that you want to use.

m = 256; % problem size
par1 = 4; % overcompleteness
par2 = 0;
par3 = 0;

% Build the matrix basis A for the dictionary
A = MakeDict(m,'DCT',par1,par2,par3);

% Generate the input signal b
b1 = InputSignal('TwinSine1', m);

% Use Donoho's MOF algorithm to find the solution x
% accuracy of the solver is 1e-5

profile on
x1 = MOF(b1,'DCT',par1,par2,par3); % change the dictionary
profile viewer

support1 = find(abs(x1) > 1e-4);
lsupport1 = size(support1)
```

APPENDIX D
MATLAB codes for solving signal denoising problems using BP

% Solves $\gamma||x||_1 + \frac{1}{2}(||Ax - b||_2)^2$
% where $\gamma = \sigma\sqrt{2\log(p)}$
% Basis pursuit with denoising

[m,n] = size(A);
% γ is chosen as in Donoho's paper
gamma = sqrt(2*log(n));

% Our QP is
% Min $\gamma \sum_{i=1}^{n}(u_i + v_i) + \frac{1}{2}p^T p$
% subject to $A(u - v) - Ip = b, u \geq 0 and v \geq 0$
% We split $p = (p_1 - p_2)$ in the actual formulation

H = sparse(2*m+2*n,2*m+2*n);
H(1:2*m,1:2*m) = [speye(m) -speye(m); -speye(m) speye(m)];
f = [zeros(2*m,1); gamma*ones(2*n,1)];

Aeq = [-speye(m) speye(m) A -A];
beq = b;

% We will use Ye's spsolqp routine to solve the QP
% Desired accuracy is 1e-2

```
    sol = spsolqp(H,Aeq,beq,f,1e-2);
% x is (u-v)
x = sol(2*m+1:2*m+n) - sol(2*m+n+1:2*m+2*n);
```

% This code requires the routine *spsolqp* and subroutines *spphase1* and *spphase2* from Ye [9] and [10].