ATOMIC DECOMPOSITION BY BASIS PURSUIT

BY

SHAOBING CHEN and DAVID L. DONOHO

TECHNICAL REPORT NO. 479

MAY 1995

PREPARED UNDER THE AUSPICES

OF

NATIONAL SCIENCE FOUNDATION GRANT DMS92–04864

DEPARTMENT OF STATISTICS

STANFORD UNIVERSITY

STANFORD, CALIFORNIA
Atomic Decomposition by Basis Pursuit

By

Shaobing Chen and David L. Donoho

Technical Report No. 479

May 1995

Prepared Under the Auspices

Of

National Science Foundation Grant DMS92-04864

Department of Statistics
Stanford University
Stanford, California
Atomic Decomposition by Basis Pursuit

Shaobing Chen & David L. Donoho
Statistics Dept., Stanford University

May, 1995

Abstract

The Time-Frequency and Time-Scale communities have recently developed a large number of overcomplete waveform dictionaries – stationary wavelets, wavelet packets, cosine packets, chirplets, and warplets, to name a few. Decomposition into overcomplete systems is not unique, and several methods for decomposition have been proposed – including the Method of Frames, Matching Pursuit, and, for special dictionaries, the Best Orthogonal Basis.

Basis Pursuit is a principle for decomposing a signal into an “optimal” superposition of dictionary elements – where optimal means having the smallest $l^1$ norm of coefficients among all such decompositions. We give examples exhibiting several advantages over the Method of Frames, Matching Pursuit and Best Ortho Basis, including better sparsity, and super-resolution. BP has interesting relations to ideas in areas as diverse as ill-posed problems, in abstract harmonic analysis, total variation de-noising, and multi-scale edge de-noising.

Basis Pursuit in highly overcomplete dictionaries leads to large-scale optimization problems. With signals of length 8192 and the wavelet packet dictionary, one gets an equivalent linear program of size 8192 by 212,992. Such problems can be attacked successfully only because of recent advances in linear programming by interior point methods. We obtain reasonable success with a primal-dual logarithmic barrier method and conjugate gradient solver.

Key Words and Phrases. Overcomplete signal representation, De-Noising, Time-Frequency Analysis, Time-Scale Analysis, $l^1$ norm optimization, Matching Pursuit, Wavelets, Wavelet Packets, Cosine Packets, Interior point methods for linear programming, Total Variation De-Noising, Multi-Scale Edges.

Acknowledgements. Michael Saunders (Department of Operations Research, Stanford University) provided advice, software, and wisdom about large-scale optimization problems.

This research was partially supported by NSF DMS-92-09130, by the NASA Astrophysical Data Program, and by other sponsors.

EDICS. 2.3.2, 2.3.3, 2.2.2, 2.2.6, 2.4.4, 2.5.1, 2.5.2, 2.5.5.

Contact Information. Email address: donoho@playfair.stanford.edu. Mail address: Department of Statistics, Stanford University, Stanford, CA 94305. Telephone: (415) 723-3350.
1 Introduction

Over the last five years or so, there has been an explosion of interest in alternatives to traditional signal representations. Instead of just representing signals as superpositions of sinusoids (the traditional Fourier representation) we now have available alternate dictionaries — collections of parameterized waveforms — of which the Wavelets dictionary is only the most well-known. Wavelets, Steerable Wavelets, Segmented Wavelets, Gabor dictionaries, Multi-scale Gabor Dictionaries, Wavelet Packets, Cosine Packets, Chirplets, Warplets, and a wide range of other dictionaries are now available. Each such dictionary $\mathcal{D}$ is a collection of waveforms $(\phi_{\gamma})_{\gamma \in \Gamma}$, with $\gamma$ a parameter, and we envision a decomposition of a signal $s$ as

$$s = \sum_{\gamma \in \Gamma} \alpha_{\gamma} \phi_{\gamma}, \quad (1.1)$$

or an approximate decomposition

$$s = \sum_{i=1}^{m} \alpha_{\gamma_i} \phi_{\gamma_i} + R^{(m)}, \quad (1.2)$$

where $R^{(m)}$ is a residual. Depending on the dictionary, such a decomposition is a decomposition into pure tones (Fourier dictionary), bumps (wavelet dictionary), chirps (chirplet dictionary), etc.

Most of the new dictionaries are overcomplete, either because they start out that way, or because we merge complete dictionaries, obtaining a new mega-dictionary consisting of several types of waveforms (e.g. Fourier & Wavelets dictionaries). The decomposition (1.1) is then nonunique, because some elements in the dictionary have representations in terms of other elements.

1.1 Goals of Adaptive Representation

Nonuniqueness gives us the possibility of adaptation, i.e. of choosing among many representations one which is most suited to our purposes. We are motivated by the aim of achieving simultaneously the following goals.

- **Sparsity.** We should obtain the sparsest possible representation of the object — i.e. the one with the fewest significant coefficients.

- **Perfect separation.** When the signal is made up of a superposition of a few very disparate phenomena (e.g. impulses and sinusoids), those should be clearly separated and marked.

- **Superresolution.** We should obtain a resolution of sparse objects that is much higher-resolution than that possible with traditional non-adaptive approaches.

- **Stability.** Small perturbations of $s$ should not seriously degrade the results.

An important constraint, which is perhaps in conflict with all the goals:

- **Speed.** It should be possible to obtain a representation in order $O(n)$ or $O(n \log(n))$ time.
1.2 Finding a Representation

Several methods have been proposed for obtaining signal representations in overcomplete dictionaries. These range from general approaches, like the Method of Frames [15], and the method of Matching Pursuit [47], to clever schemes derived for specialized dictionaries, like the method of Best Orthogonal Basis [12]. These methods will be described in more detail below.

In our view, these methods have advantages and shortcomings. The principal emphasis of the proposers of these methods is in achieving sufficient computational speed. While the resulting methods are practical to apply to real data, we show below by computational examples, that the methods, either quite generally or in important special cases, lack qualities of sparsity-preservation and of stable super-resolution.

1.3 Basis Pursuit

Basis Pursuit (BP) finds signal representations in overcomplete dictionaries by convex optimization: it obtains the decomposition which minimizes the $\ell^1$ norm of the coefficients occurring in the representation. Because of the non-differentiability of the $\ell^1$ norm, this optimization principle leads to decompositions which can have very different properties from the Method of Frames – in particular they can be much sparser. Because it is based on global optimization, it can stably super-resolve in ways that Matching Pursuit can not.

BP can be used with noisy data by solving an optimization problem trading off a quadratic misfit measure with an $\ell^1$ norm of coefficients. Examples show that it can stably suppress noise while preserving structure which is well-expressed in the dictionary under consideration.

BP has connections with a number of ideas occurring in the fields of inverse problems and harmonic analysis. Such connections help us understand why BP might be a good idea. BP is also closely connected with linear programming. Recent advances in large scale linear programming – associated with the interior point method – can be applied to BP, and make it possible, with certain dictionaries, to nearly-solve the BP optimization problem in nearly-linear time.

We have implemented a primal-dual log barrier interior point method as part of a computing environment – Atomizer – which accepts any of a wide range of dictionaries. Instructions for Internet access of Atomizer are given below. Experiments with standard time-frequency dictionaries indicate some of the potential benefits of BP. Experiments of BP with some nonstandard dictionaries – like the stationary wavelet dictionary and the Heaviside dictionary – indicate important connections of BP with methods like Mallat and Hwang’s Multi-Scale Edge Representation and Osher, Rudin and Fatemi’s Total Variation-based De-Noising methods.

1.4 Contents

The contents of our paper go as follows: in Section 2 we establish vocabulary and notation for the rest of the article, describing a number of dictionaries, and existing methods for overcomplete representation. In Section 3, we discuss the principle of Basis Pursuit, its relations to existing methods and to ideas in other fields. In section 4, we discuss methodological issues associated with BP – in particular some of the interesting nonstandard ways it can be deployed. In Section 5 we describe Basis Pursuit De-Noising, a method for dealing
with problem (1.2). In Section 6 we discuss recent advances in large scale linear programming, and resulting algorithms for BP. In section 7 we discuss a number of connections with other work.

2 Overcomplete Representations

Let \( s = (s_t : 0 \leq t < n) \) be a discrete-time signal of length \( n \); this may also be viewed as a vector in \( \mathbb{R}^n \). We are interested in the reconstruction of this signal using superpositions of elementary waveforms. Traditional methods of analysis and reconstruction involve the use of orthogonal bases, such as the Fourier Basis, various Discrete Cosine transform bases, and orthogonal wavelet bases. Such situations can be viewed as follows: one has a list of \( n \) waveforms, and one complete to represent \( s \) as a linear combination of these waveforms. The waveforms in the list, viewed as vectors in \( \mathbb{R}^n \), are linearly independent, and so the representation is unique.

2.1 Dictionaries and Atoms

A considerable focus of activity in the recent signal processing literature has been the development of signal representations outside the basis setting. We use terminology introduced by Mallat and Zhang [47]. A **dictionary** is a collection of parameterized waveforms \( D = \{ \phi_\gamma : \gamma \in \Gamma \} \). The waveforms \( \phi_\gamma \) are discrete-time signals of length \( n \) called **atoms**. Depending on the dictionary, the parameter \( \gamma \) can have the interpretation of indexing frequency, in which case the dictionary is a frequency or Fourier dictionary, of indexing time/scale jointly, in which case the dictionary is a time-scale dictionary, or of indexing time/frequency jointly, in which case the dictionary is a time-frequency dictionary. Usually dictionaries are complete or overcomplete, in which case they contain exactly \( n \) atoms, or more than \( n \) atoms, but one could also have continuum dictionaries containing an infinity of atoms, and undercomplete dictionaries for special purposes, containing fewer than \( n \) atoms. Dozens of interesting dictionaries have been proposed over the last few years; we focus in this paper on a half dozen or so; most of what we do applies in other cases as well.

2.1.1 Trivial Dictionaries

We begin with some overly simple examples. The **Dirac** dictionary is simply the collection of waveforms which are zero except in one point: \( \gamma \in \{0, 1, \ldots, n-1\} \) and \( \phi_\gamma(t) = 1_{\{t=\gamma\}} \). This is of course also an orthogonal basis of \( \mathbb{R}^n \) – the standard basis. The **Heaviside** dictionary is the collection of waveforms which jump at one particular point: \( \gamma \in \{0, 1, \ldots, n-1\}; \phi_\gamma(t) = 1_{\{t \geq \gamma\}} \). Atoms in this dictionary are not orthogonal, but every signal has a representation

\[
s = s_0 \phi_0 + \sum_{\gamma=1}^{n-1} (s_\gamma - s_{\gamma-1}) \phi_\gamma.
\]  

(2.1)

2.1.2 Frequency Dictionaries

A Fourier dictionary is a collection of sinusoidal waveforms \( \phi_\gamma \) indexed by \( \gamma = (\omega, \nu) \), where \( \omega \in [0, 2\pi) \) is an angular frequency variable and \( \nu \in \{0, 1\} \) indicates phase type – sine or cosine. In detail,

\[
\phi(\omega, 0) = \cos(\omega t), \quad \phi(\omega, 1) = \sin(\omega t).
\]
For the standard Fourier dictionary, we let $\gamma$ run through the set of all cosines with Fourier frequencies $\omega_k = 2\pi k/n$, $k = 0, \ldots, n/2$, and all sines with Fourier frequencies $\omega_k$, $k = 1, \ldots, n/2 - 1$. This dictionary consists of $n$ waveforms; it is in fact a basis, and a very simple one – the atoms are all mutually orthogonal. An overcomplete Fourier dictionary is obtained by sampling the frequencies more finely. Let $\ell$ be a whole number $> 1$ and let $\Gamma_\ell$ be the collection of all cosines with $\omega_k = 2\pi k/(\ell \cdot n)$, $k = 0, \ldots, \ell n/2$, and all sines with frequencies $\omega_k$, $k = 1, \ldots, \ell n/2 - 1$. This is an $\ell$-fold overcomplete system. We also use below complete and overcomplete dictionaries based on cosine transforms and sine transforms.

### 2.1.3 Time-Scale Dictionary

There are several types of Wavelet dictionary; to fix ideas, we consider the Haar dictionary, with "Father Wavelet" $\varphi = 1_{[0,1]}$ and "Mother Wavelet" $\psi = 1_{[1/2,1]} - 1_{[0,1/2]}$. The dictionary is a collection of translations and dilations of the basic mother wavelet, together with translations of a father wavelet. It is indexed by $\gamma = (a, b, \nu)$, where $a \in (0, \infty)$ is a scale variable, $b \in [0, n]$ indicates location, and $\nu \in \{0, 1\}$ indicates gender. In detail,

$$\phi_{(a,b,1)} = \psi(a(t - b)) \cdot \sqrt{a}, \quad \phi_{(a,b,0)} = \varphi(a(t - b)) \cdot \sqrt{a}.$$

For the standard Haar dictionary, we let $\gamma$ run through the discrete collection of mother wavelets with dyadic scales $a_j = 2^j/n$, $j = j_0, \ldots, \log_2(n) - 1$, and locations which are integer multiples of the scale $b_{j,k} = k \cdot a_j$, $k = 0, \ldots, 2^j - 1$, and the collection of father wavelets at the coarse scale $j_0$. Hence

$$\Gamma_{\text{Haar}} = \{(a_j, b_{j,k}, 1), j = j_0, \ldots, \log_2(n) - 1, k = 0, \ldots, 2^j - 1\}, \quad \{(a_{j_0}, b_{j_0,k}, 0), k = 0, \ldots, 2^{j_0} - 1\}$$

This dictionary consists of $n$ waveforms; it is an orthonormal basis. An overcomplete wavelet dictionary is obtained by sampling the locations more finely – one location per sample point. This gives the so-called Stationary Haar dictionary:

$$\Gamma_{\text{StatHaar}} = \{(a_j, t, 1), j = j_0, \ldots, \log_2(n) - 1, t = 0, \ldots, n - 1\}, \quad \{(a_{j_0}, t, 0), t = 0, \ldots, n - 1\}$$

This dictionary consists of $O(n \log_2(n))$ waveforms. It is called stationary since the whole dictionary is invariant under circulant shift. The standard wavelet dictionary is a subdictionary of $n$ particular waveforms.

A variety of other wavelet bases are possible. The most important variations are smooth wavelet bases, using splines or using wavelets defined recursively from two-scale filtering relations. Although the rules of construction are more complicated (boundary conditions [52], orthogonality versus bi-orthogonality [16], etc.), these have the same indexing structure as the standard Haar dictionary. In this paper, we use Symmlet-8 smooth wavelets – i.e. Daubechies Nearly Symmetric wavelets with eight vanishing moments; compare e.g [16] for examples.

Further variations on 1-d wavelets include Segmented Wavelet Bases [1, 17, 22] and multi-wavelets [38, 61]. In two dimensions, an even wider range of 2-d wavelet dictionaries is possible, because of the possibility of directional preference; examples include hyperbolic cross dictionaries and steerable wavelets [2, 60]. The ideas of this paper ought to be interesting in connection with such variations, but we do not explore them here.
2.1.4 Time-Frequency Dictionaries

Much recent activity in the wavelet communities has focused on the study of time-frequency phenomena. The standard example the Gabor dictionaries, is due to Gabor (1946); in our notation, we take \( \gamma = (\omega, \tau, \theta, \delta t) \) where \( \omega \in [0, \pi) \) is a frequency, \( \tau \) is a location, \( \theta \) is a phase, and \( \delta t \) is the duration, and consider atoms

\[
\phi_\gamma(t) = \exp\left\{ -(t - \tau)^2 / (\delta t)^2 \right\} \cdot \cos(\omega(t - \tau) + \theta).
\]

Such atoms indeed consist of frequencies near \( \omega \) and essentially vanish far away from \( \tau \). For fixed \( \delta t \), discrete dictionaries can be built from time-frequency lattices, \( \omega_k = k\Delta\omega \) and \( \tau_\ell = \ell\Delta\tau \), and \( \theta \in \{0, \pi/2\} \); with \( \Delta\tau \) and \( \Delta\omega \) chosen sufficiently fine these are complete. For further discussions see e.g. [15].

Recently, Coifman and Meyer [10] developed the wavelet packet and cosine packet dictionaries especially to meet the computational demands in discrete-time signal processing. For 1-d discrete time signals of length \( n \), these dictionaries each contain about \( n \log_2(n) \) waveforms. The wavelet packet dictionary includes, as special cases, the standard orthogonal wavelets dictionary, the Dirac dictionary, and a collection of oscillating waveforms spanning a range of frequencies and durations. The cosine packet dictionary contains, as special cases, the standard orthogonal Fourier dictionary, and a variety of Gabor-like elements — sinusoids of various frequencies weighted by windows of various widths and locations.

In this paper, we will often use the wavelet packet and cosine packet dictionaries as examples of overcomplete systems, and we will give a number of examples decomposing signals into these time-frequency dictionaries. A simple block-diagram will help us visualize the atoms appearing in the decomposition. This diagram, adapted from Coifman and Wickerhauser [12], associates to each cosine packet or wavelet packet a rectangle in the time-frequency phase plane. The association is illustrated in Figure 2.1 for a certain wavelet packet. When a signal is a superposition of several such waveforms, we indicate which waveforms appear in the superposition by shading the corresponding rectangles in the time-frequency plane.

2.1.5 Further Dictionaries

New dictionaries are appearing on an almost daily basis; we cannot discuss all that we would like. Interesting dictionaries include Chirplets [48, 54], Warplets [3], and Joint-Time-Frequency Segmented Wavelet Packets [40]. Much of what we do can be worked out for other cases.

A final remark is that we can always merge dictionaries to create mega-dictionaries; examples we will use below include mergers of Dirac and overcomplete Fourier, and mergers of Wavelets with Heavisides.

2.2 Linear Algebra

Suppose now that we have a discrete dictionary of \( p \) waveforms and collect all these waveforms as columns of an \( n \) by \( p \) matrix \( \Phi \), say. The decomposition problem (1.1) can be written

\[
\Phi \alpha = s
\]  

(2.2)

where \( \alpha = (\alpha_\gamma) \) is the vector of coefficients occurring in the representation (1.1). When the dictionary furnishes a basis, then \( \Phi \) is an \( n \) by \( n \) invertible matrix and we have the unique
representation

\[ \alpha = \Phi^{-1}s. \]

When the atoms are, in addition, mutually orthonormal, then \( \Phi^{-1} = \Phi^T \), and so the decomposition formula is very simple.

An important (but trivial) comment. Given a dictionary of waveforms, one can distinguish analysis from synthesis. Synthesis is the operation of building up a signal by superposing atoms: it involves a matrix which is \( n \) by \( p \). Analysis involves the operation of associating, to each signal, a vector of coefficients, attached to atoms; it involves a matrix which is \( p \) by \( n \). When \( p \neq n \), synthesis and analysis are very different linear operations, and we must take care to distinguish them.

The problem we are dealing with in (2.2) is the problem of synthesis – how to generate a given signal from a given dictionary. The \( n \) by \( p \) matrix \( \Phi \) is the synthesis operator. As its transpose \( \Phi^T \) has the same shape as an analysis operator (it is \( p \) by \( n \)), one might expect that the analysis \( \hat{\alpha} = \Phi^Ts \) gives us coefficients which can then be used to reconstruct \( s \). In general, this only happens when \( p = n \), and when the atoms are orthonormal; then \( \Phi \) is an orthogonal matrix and \( \Phi^{-1} = \Phi^T \).

In the overcomplete case we are interested in, \( p \gg n \), and \( \Phi \) is not an invertible matrix. There are then many solutions to (2.2), and a given approach selects a particular solution. One does not uniquely and automatically solve the synthesis problem by applying a simple, linear, analysis operator.

Nevertheless, both \( \Phi \) and \( \Phi^T \) will play key roles in our approach.
2.3 Visualization of $\Phi$ and $\Phi^T$

We now illustrate the difference between synthesis ($s = \Phi\alpha$) and analysis ($\Phi^Ts$).

*Stationary Haar Dictionary.* Panel 2.2a shows the signal Blocks. Panel 2.2b shows the time-scale structure of a synthesis of Blocks – i.e. a vector $\alpha$ yielding $s = \Phi\alpha$, using the Stationary Haar dictionary mentioned in Section 2.1. To visualize the decomposition, we present a multi-resolution display, where we plot, at level $j$, all the coefficients ($\alpha_j$) of scale $a_j = 2^j/n$, arranged according to spatial position. Panel 2.2c gives an analysis of Blocks, the coefficients $a = \Phi^Ts$, again arranged in multi-resolution fashion. Evidently, there is a large difference in sparsity between analysis and synthesis. In Panel 2.2d we compare the sorted coefficients of analysis and synthesis.

*Wavelet Packet Dictionary.* Panel 2.3a shows the signal Carbon. Panel 2.3b shows the time-frequency structure of a sparse synthesis of Carbon, a vector $\alpha$ yielding $s = \Phi\alpha$, using the Wavelet Packet dictionary. To visualize the decomposition, we present a phase-plane display with shaded rectangles, as described above. Panel 2.3c gives an analysis of Carbon, the coefficients $a = \Phi^Ts$, again displayed in a phase-plane. Once again, between analysis and synthesis there is a large difference in sparsity. In Panel 2.3d we compare the sorted coefficients of the overcomplete representation (synthesis) with the analysis coefficients.

2.4 Computational Complexity of $\Phi$ and $\Phi^T$

Different dictionaries can impose drastically different computational burdens. In this paper we report computational experiments on a variety of signals and dictionaries. We study primarily 1-d signals of length $n$ several thousand. (Signals of this length occur naturally
in study of short segments of speech (a quarter second to a half a second), and the output of various scientific instruments (e.g. FT-NMR spectrometers). In our experiments, we study dictionaries overcomplete by substantial factors, say 10. Hence the typical matrix $\Phi$ we are interested in is of size "Thousands" by "Tens-of-Thousands".

The nominal cost of storing and applying an arbitrary $n$ by $p$ matrix to a $p$-vector is a constant times $np$. Hence with an arbitrary dictionary of the sizes we are interested in, simply to verify whether (1.1) holds for given vectors $\alpha$ and $s$ would require tens of millions of multiplications and tens of millions of words of memory. In contrast, most signal processing algorithms for signals of length 1000 require only thousands of memory words and a few thousand multiplications.

Fortunately, certain dictionaries have fast implicit algorithms. By this we mean that $\Phi\alpha$ and $\Phi^T s$ can be computed, for arbitrary vectors $\alpha$ and $s$, (a) without ever storing the matrices $\Phi$ and $\Phi^T$, and (b) using special properties of the matrices to accelerate computations.

The most well-known example is the standard Fourier dictionary, for which we have the fast Fourier transform algorithm. A typical implementation requires $2 \cdot n$ storage locations and $4 \cdot n \cdot J$ multiplications, if $n$ is dyadic: $n = 2^J$. Hence for very long signals we can apply $\Phi$ and $\Phi^T$ with much less storage and time than the matrices would nominally require. Simple adaptation of this idea leads to an algorithm for overcomplete Fourier dictionaries.

Wavelets give a more recent example of a dictionary with a fast implicit algorithm; if the Haar or S8-Symmlet is used, both $\Phi$ and $\Phi^T$ may be applied in $O(n)$ time. For the stationary wavelet dictionary, $O(n \log_2(n))$ time is required.

Cosine Packets and Wavelet Packets also have fast implicit algorithms. Here both $\Phi$ and $\Phi^T$ can be applied in order $O(n \log_2(n))$ time and order $(n \log_2(n))$ space – much better
than the nominal $np = n^2 \log^2(n)$ one would expect from naive use of the matrix definition. In fact, Coifman and Meyer aimed exactly for this: to develop a multiscale dictionary with fast algorithms for applying $\Phi$ and $\Phi^T$.

For the viewpoint of this paper, it makes sense only to consider dictionaries with fast implicit algorithms. Among dictionaries we have not discussed such algorithms may or may not exist. For doubly-segmented wavelet packets and for certain types of steerable wavelets, such algorithms exist. For others, say Chirplets, fast implicit algorithms do not seem to exist.

2.5 Existing Decomposition Methods

There are several currently popular approaches to obtaining solutions to (2.2).

2.5.1 Frames

The Method of Frames (MOF) [15] picks out, among all solutions of (2.2), one whose coefficients have minimum $l^2$ norm.

$$\min \|\alpha\|_2 \text{ subject to } \Phi \alpha = s. \quad (2.3)$$

The solution of this problem is unique; label it $\alpha^\dagger$. Geometrically, the collection of all solutions to (2.2) is an affine subspace in $\mathbb{R}^n$; MOF selects the element of this subspace closest to the origin. It is sometimes called a minimum-length solution.

The optimization problem (2.3) is equivalent to a standard chapter in linear algebra; there is a matrix $\Phi^\dagger$, the generalized inverse of $\Phi$, which calculates the minimum length solution to a system of linear equations:

$$\alpha^\dagger = \Phi^\dagger s = \Phi^T (\Phi\Phi^T)^{-1} s$$

In general numerical calculation of the generalized inverse matrix $\Phi^\dagger$ involves tasks like singular value decomposition and requires order $\min(n,p)^3$ time. However, there exist iterative methods for computation of $\alpha^\dagger$ which require only the repeated application of $\Phi$ and $\Phi^T$ to various strategically chosen vectors (e.g. relaxation and conjugate gradients [15, 35]) and so the calculation and storage of $\Phi^\dagger$ can be avoided.

For certain overcomplete dictionaries, MOF is available in closed form. A nice example: the standard wavelet packet dictionary. One can compute that for all vectors $v$,

$$\|\Phi^T v\|^2 = L_n \cdot \|v\|^2; \quad L_n = \log_2(n).$$

(This is an example of a so-called "Tight Frame"). In short

$$\Phi^\dagger = L_n^{-1} \Phi^T.$$ 

Now for the Wavelet Packet dictionary, $\Phi^T$ is simply the analysis operator, and can be calculated in $n \log_2(n)$ operations.

There are two key problems with the Method of Frames. First, MOF is not sparsity-preserving. If the underlying object has a very sparse representation in terms of the dictionary, then the coefficients found by MOF are likely to be very much less sparse. Each atom in the dictionary that has nonzero inner product with the signal is, at least potentially, and also usually, a member of the solution.
As an example, we consider Figure 2.4a, which shows the signal Hydrogen, made of a single atom in a wavelet packet dictionary. The result of a frame decomposition in that dictionary is depicted in a phase-plane portrait, figure 2.4c. While the underlying signal can be synthesized from a single atom, the frame decomposition involves many atoms, and the phase-plane portrait exaggerates greatly the intrinsic complexity of the object.

Second, MOF is intrinsically resolution-limited. No object can be reconstructed with features sharper than those allowed by the underlying operator $\Phi^\dagger\Phi$. Suppose the underlying object is sharply localized: $\alpha = 1_{\{x=x_o\}}$. The reconstruction will not be $\alpha$, but instead $\Phi^\dagger\Phi\alpha$ which, in the overcomplete case, will be spatially spread out. Figure 2.5 presents a signal, TwinSine-1, which consists of the superposition of two sinusoids taken from an overcomplete dictionary of cosines with overcompleteness factor $\ell = 4$. The two frequencies in TwinSine-1 are separated by less than the so-called Rayleigh Distance $2\pi/n$. In this case, reconstruction by MOF, figure 2.5b, is simply convolution with the Dirichlet kernel. The result is the synthesis from coefficients with a broad oscillatory appearance, consisting not of two but of many frequencies, and giving no visual clue that the object may be synthesized from two frequencies alone.

### 2.5.2 Matching Pursuit

Mallat and Zhang [47] have discussed a general method for approximate decomposition (1.2) which addresses the sparsity issue directly. Starting from an initial approximation $s^{(0)} = 0$ and residual $R^{(0)} = s$, it builds up a sequence of sparse approximations stepwise. At stage $k$, it identifies that dictionary atom which best correlates with the residual and then adds to the current approximation a scalar multiple of that atom, so that $s^{(k)} = s^{(k-1)} + \alpha_k \phi_{\gamma_k}$.
where $\alpha_k = \langle R^{(k-1)}, \varphi_{nk} \rangle$, and $R^{(k)} = s - s^{(k)}$. After $m$ steps, one has a representation of the form (1.2), with residual $R = R^{(m)}$. A similar algorithm was proposed, for Gabor dictionaries, by S. Qian and D. Chen [58].

An intrinsic feature of the algorithm is that when stopped after a few steps, it yields an approximation using only a few atoms. When the dictionary is orthogonal, the method works perfectly. If the object is made up of only $m \ll n$ atoms and the algorithm is run for $m$ steps, it recovers the underlying sparse structure exactly.

When the dictionary is not orthogonal, the situation is less clear. Because the algorithm is myopic, one expects that, in certain cases, it might choose wrongly in the first few iterations and, in such cases, end up spending most of its time correcting for any mistakes made in the first few terms. In fact this does seem to happen.

To see this, we consider an attempt at super-resolution. Figure 2.5a portrays again the signal TwinSine-1 consisting of sinusoids at two closely spaced frequencies. When MP is applied in this case (Figure 2.5c), using an overcomplete dictionary, the initial frequency selected is in between the two frequencies making up the signal. Because of this mistake, MP is forced to make a series of alternating corrections which suggest a highly complex and organized structure. MP misses entirely the doublet structure. One can certainly say in this case that MP has failed to super-resolve.

Second, one can give examples of dictionaries and signals where MP is arbitrarily sub-optimal in terms of sparsity. While these are somewhat artificial, they have a character not so different from the super-resolution example.

DeVore and Temlyakov’s Example. Vladimir Temlyakov, in a talk at the IEEE Conference on Information Theory and Statistics, October 1994, described an example in which the straightforward greedy algorithm is not sparsity-preserving. In our adaptation of this
example, based on Temlyakov's joint work with R.A. DeVore [19], one constructs a dictionary having \( n + 1 \) atoms. The first \( n \) are the Dirac basis; the final atom involves a linear combination of the first \( n \) with decaying weights. The signal \( s \) has an exact decomposition in terms of \( A \) atoms; but the greedy algorithm goes on forever, with an error of size \( O(1/\sqrt{m}) \) after \( m \) steps.

We illustrate this decay in Figure 2.6a. For this example we set \( A = 10 \) and choose the signal \( s_t = 10^{-1/2} \cdot 1_{\{1 \leq t \leq 10\}} \). The dictionary consists of Dirac elements \( \phi_n = \delta_n \) for \( 1 \leq n \leq n \), and

\[
\phi_{n+1}(t) = \begin{cases} 
  c & 1 \leq t \leq 10 \\
  c/(t - 10) & 10 < t \leq n 
\end{cases}
\]

with \( c \) chosen to normalize \( \phi_{n+1} \) to unit norm.

**Shaobing Chen's Example.** The DeVore-Temlyakov example applies to the original MP algorithm as announced by Mallat and Zhang in 1992. A later refinement (see also Pati [57]) involves an extra step of orthogonalization. One takes all \( m \) terms which have entered at stage \( m \) and solves the least squares problem

\[
||s - \sum_{i=1}^{m} a_i \phi_{n_i}||_2 = \min!
\]

for coefficients \( (a_i^{(m)}) \). Then one forms the residual \( \tilde{R}^{(m)} = s - \sum_{i=1}^{m} a_i^{(m)} \phi_{n_i} \), which will be orthogonal to all terms currently in the model. This method is called Orthogonal Matching Pursuit (OMP) by Pati [57]. The DeVore-Temlyakov example does not apply to OMP, but Shaobing Chen found in Summer 1993 an example of similar flavor which does. In this example, a special signal and dictionary are constructed, with the following flavor. The dictionary is composed of atoms \( \phi_\gamma \) with \( \gamma \in \{1, \ldots, n\} \). The first \( A \) atoms come from the Dirac dictionary; with \( \gamma \in \{1, \ldots, A\} \), \( \phi_\gamma = \delta_\gamma \). The signal is a simple equiweighted linear combination of the first \( A \) atoms: \( s = A^{-1} \sum_{i=1}^{A} \phi_i \). Dictionary atoms with \( \gamma > A \) are a linear combination of the corresponding Dirac \( \delta_\gamma \) and \( s \). OMP chooses all atoms except the first \( A \) before ever choosing one of the first \( A \). As a result, instead of the ideal behavior one might hope for, terminating after just \( A \) steps, one gets \( n \) steps before convergence, and the rate is relatively slow.

We illustrate the behavior of the error in Figure 2.6b. We chose \( A = 10 \), and \( n = 1024 \). The dictionary \( \phi_i = \delta_i \) for \( 1 \leq i \leq 10 \); and \( \phi_i = \sqrt{a} \cdot s + \sqrt{1 - a} \cdot e_i \) for \( 11 \leq i \leq n \), where \( a = 2/10 \). With these parameters, \( ||\tilde{R}^{(m)}||_2 = (1 - a)/\sqrt{1 + (m - 1) \cdot a} \), whereas one might have hoped for the ideal behavior \( \tilde{R}^{(m)} = 0, m \geq 11 \).

### 2.5.3 Best Orthogonal Basis

For certain dictionaries, it is possible to develop specific decomposition schemes custom-tailored to the dictionary.

The wavelet packets and cosine packets dictionaries are examples; in addition to fast implicit algorithms, they have other very special properties. Certain special subcollections of the elements in these dictionaries amount to orthogonal bases; one gets in this way a wide range of orthonormal bases (in fact \( \geq 2^n \) such orthogonal bases for signals of length \( n \)). Although it mixes metaphors rather badly, we will sometimes say that the wavelet packets and cosine packets dictionaries furnish libraries of orthogonal bases.

Coifman and Wickerhauser [12] have proposed a method of adaptively picking from among these many bases a single orthogonal basis which is the "best basis". If \( (s[B]_f) \)
denotes the vector of coefficients of \( s \) in orthogonal basis \( B \), and if we define the "entropy"
\[
\mathcal{E}(s[B]) = \sum_f e(s[B]_f),
\]
where \( e(s) \) is a scalar function of a scalar argument, they give a fast algorithm for solving
\[
\min\{\mathcal{E}(s[B]) : \text{B ortho basis } \subset \mathcal{D}\}.
\]

The algorithm is fast – it delivers a “best basis” in order \( n \log(n) \) time – and in some cases delivers near-optimal sparsity representations. In particular, when the object in question has a sparse representation in an orthogonal basis taken from the library, one expects that BOB will work well.

Unfortunately, when the signal is composed of a moderate number of highly non-orthogonal components, the method may not deliver sparse representations – the demand that BOB find an orthogonal basis prevents it from finding a highly sparse representation. An example comes from the signal \texttt{WernerSorrows}, which is a superposition of several chirps, sinusoids and Diracs, yielding a highly complex phase plane; Figure 2.7a. When analyzed with Cosine Packet Dictionary, and original Coifman-Wickerhauser entropy, BOB finds nothing: it chooses a global sinusoid basis as best; the lack of time-varying structure in that basis means that all chirp and transient structure in the signal is missed entirely – Figure 2.7b.

2.5.4 An "ideal" method: Massive Optimization

To avoid the limitations of Matching Pursuit, which employs greedy search, and BOB, which employs search only among bases which are orthogonal, one might consider a true global optimization proposal based on exhaustive enumeration of all bases which can be
formed from the dictionary. The idea would be to cycle through all linearly independent subsets of size \( n \) from the dictionary and decompose the signal in each such basis. A numerical measure of simplicity would be applied to the coefficients of each decomposition to rank the different decompositions and to select a “best” decomposition.

Unfortunately, for most measures of simplicity, the amount of work required to do this sort of optimization is comparable to the number of different bases which can be formed, namely \( \binom{p}{n} \). This amount of work is already out of the question even for moderately overcomplete dictionaries and moderately long signals. For example, with \( p \geq 2n \), and \( n \) in the thousands, it is astronomically large.

3 Basis Pursuit

We now discuss our approach to the problem of overcomplete representations. We assume that we are given a dictionary \( \{\phi_\gamma\} \) of vectors in \( \mathbb{R}^n \) and that the cardinality of the dictionary exceeds \( n \). We assume that the dictionary is overcomplete, so that there are in general many representations \( s = \sum_\gamma \alpha_\gamma \phi_\gamma \).

The principle of basis pursuit is to find a representation of the signal whose coefficients have minimal \( \ell^1 \) norm. Formally, one solves the problem

\[
\min ||a||_1 \text{ subject to } \Phi a = s. \tag{3.1}
\]

From one point of view, (3.1) is very similar to the method of frames (2.3); we are simply replacing the \( \ell^2 \) norm in the definition of the frames method with the \( \ell^1 \) norm in basis pursuit. However, this apparently slight change has major consequences. The method of
frames requires the solution of a quadratic optimization problem, and so involves essentially just linear algebra. In contrast, Basis Pursuit requires the solution of a convex, nonquadratic optimization problem, which requires considerably more effort and sophistication.

3.1 Linear Programming

To explain the last comment – and also to explain the name Basis Pursuit – we develop a connection with linear programming.

The linear program in so-called standard form [14, 34] is a constrained optimization problem defined in terms of a variable \( x \in \mathbb{R}^m \) by

\[
\min c^T x \text{ subject to } Ax = b, \text{ and to } x \geq 0
\]

(3.2)

where \( c^T x \) is the objective function, \( Ax = b \) is the collection of equality constraints, and \( x \geq 0 \) is the non-negativity constraint.

Although the Basis Pursuit problem (3.1) involves nonlinear optimization, it can be equivalently reformulated, via the “method of slack variables”, as a linear program in the standard form (3.2). Let \( u \) and \( v \) be \( p \)-vectors. Consider the constrained optimization problem defined in terms of \( u \) and \( v \) by

\[
\min 1^T u + 1^T v \text{ subject to } \Phi(u - v) = s, u, v \geq 0.
\]

(3.3)

This is an optimization problem on a subset of \( \mathbb{R}^{2p} \); if we define \( a \equiv u - v \), any solution of this problem is a solution of (3.1). Indeed \( \Phi a = \Phi(u - v) \), while \( 1^T u + 1^T v = \sum_i |\alpha_i| \).

Now (3.3) can be written as a linear program in standard form by making the following translations:

\[
\begin{align*}
m &\leftrightarrow 2p \\
x &\leftrightarrow (u, v) \\
c &\leftrightarrow (1, 1) \\
A &\leftrightarrow (\Phi, -\Phi) \\
b &\leftrightarrow s
\end{align*}
\]

Hence, the solution of (3.1) can be obtained by solving an equivalent linear program. (The equivalence of minimum \( \ell^1 \) optimizations with linear programming has been known since the 1950's; see [5]). The connection of Basis Pursuit with linear programming is useful in several ways.

3.1.1 Solutions as Bases

It is well known that in solving a linear programming problem with \( A \) an \( n \) by \( m \) matrix and \( m > n \), the LP Solution will always have at most \( n \) nonzero coefficients at the solution – i.e. at most \( n \) of the entries in the optimal \( x \) will be nonzero. Moreover, in the generic case, the solution is so-called nondegenerate, and there are exactly \( n \) nonzeros. In this case, the nonzero coefficients are associated with \( n \) columns of \( A \); and these columns make up a basis of \( \mathbb{R}^n \). The identity of the columns in this optimal basis is not, in general, known in advance, but instead depend on the problem data (in this case \( b \)). Moreover, there is a sense in which finding a solution to the LP is identical with finding this basis, since once
the basis is identified, the solution is uniquely dictated by the basis. (Indeed, if the \( n \) by \( n \) matrix \( B \) contains as columns the \( n \) optimizing columns of \( A \), then the constraint equation \( Ax = b \) forces the nonzero coefficients of the solution to be obtained from \( B^{-1}b \).) In this sense, linear programming is truly a process of Basis Pursuit.

Translating the LP results into BP terminology, we have the decomposition

\[
    s = \sum_{i=1}^{n} \alpha_i \phi_{i,n}.
\]

The waveforms \( (\phi_{i,n}) \) are linearly independent but not necessarily orthogonal. The collection \( \gamma_i \) is not, in general, known in advance, but instead depends on the problem data (in this case \( s \)). So the selected waveforms are signal-adaptive.

### 3.1.2 Algorithms

BP is an optimization principle, not an algorithm. Over the last forty years, a tremendous amount of work has been done on the solution of linear programs. Until the 1980's, most work focused on variants of Dantzig's Simplex algorithm, which many readers have no doubt studied. In the last ten years, some spectacular breakthroughs have been made by the use of so-called "interior point methods" which use an entirely different principle.

From our point of view, we are free to consider any algorithm from the LP literature as a candidate for computing the solutions to the BP optimization problem; both the simplex and interior point algorithms offer interesting insights into BP. When it is useful to consider BP in the context of a particular algorithm, we will indicate this by label: either BP-Simplex or BP-Interior.

**BP-Simplex.** In the standard implementation of the simplex method for LP, one first finds an initial basis \( B \) consisting of \( n \) linearly independent columns of \( A \) for which the corresponding solution \( B^{-1}b \) is feasible – non-negative. Then one iteratively improves the current basis by, at each step, swapping one term in the basis for one term not in the basis, using that swap which best improves the objective functional. The fundamental theorem of LP tells us there will always exist a swap which improves or maintains the objective value, except at the optimal solution. Moreover, LP researchers have shown how one can select terms to swap in such a way as to always guarantee convergence to an optimal solution (anticycling rules) [34]. Hence the simplex algorithm is explicitly a process of "Basis Pursuit" – iterative improvement of a basis until no improvement is possible, at which point the solution is achieved.

Translating this LP algorithm into BP terminology, one starts from any linearly independent collection of \( n \) atoms from the dictionary. One calls this the current decomposition. Then one iteratively improves the current decomposition by swapping atoms in the current decomposition for new atoms, with the goal of improving the objective functional. By application of anti-cycling rules, there is a way to select swaps that guarantees convergence to an optimal solution (assuming exact floating point arithmetic).

**BP-Interior.** The collection of feasible points \( \{x : Ax = b, x \geq 0\} \) is a convex polyhedron in \( R^n \), or "simplex". The simplex method, viewed geometrically, works by walking around the boundary of this simplex, jumping from one vertex of the polyhedron to an adjacent vertex at which the objective is better. The interior point method instead starts from a point well in the interior of the simplex and goes "through the middle" of the simplex. The solution of an LP is always at an extreme point of the simplex, so as the interior
point method approaches convergence, the current iterate approaches the boundary. At some point, it "becomes clear" which extreme point of the simplex attains the optimum, and one abandons the basic interior point iteration and "jumps" to the optimizing extreme point.

Translating this LP algorithm into BP terminology, one starts from a solution to the overcomplete representation problem \( \Phi a(0) = s \) containing many more than \( n \) nonzero elements. One iteratively modifies the coefficients, maintaining feasibility \( \Phi a(k) = s \), but applying a transformation which has the effect of successively sparsifying the vector \( a \). At some iteration, the vector has \( \leq n \) significantly nonzero entries, and it "becomes clear" that those correspond to the atoms appearing in the final solution. One forces all the other coefficients to zero and "jumps" to the decomposition in terms of the \( \leq n \) selected atoms.

### 3.2 Examples

We now give computational examples of BP in action. All figures presented in this article are reproducible using Atomizer, the software package we have developed as part of this project; it is available over Internet.

#### 3.2.1 Carbon

The synthetic signal Carbon is a composite of 6 atoms: a Dirac, a sinusoid, and 4 mutually orthogonal wavelet packet atoms, adjacent in the time-frequency plane. The wavelet packet dictionary of depth \( D = \log_2(n) \) is employed, based on filters for Symmlets with 8 vanishing moments.

Figure 3.1 displays the results in phase-plane form; for comparison, we include the phase planes obtained using MOF, MP, and BOB. First, note that MOF uses all basis functions which are not orthogonal to the 6 atoms, i.e. all the atoms at those times and frequencies which overlap with some atom appearing in the signal. The corresponding phase plane is very diffuse or smeared out. Second, MP is able to do a relatively good job on the sinusoid and the Dirac, but makes mistakes in handling the 4 close atoms. Third, BOB cannot handle the nonorthogonality between the dirac and the cosine; it gives a distortion – a coarsening – of the underlying phase plane picture. Finally, BP finds the "exact" decomposition in the sense that the four atoms in the quad are correctly identified, the Dirac is correctly identified, and the sinusoid is analysed as accurately as the wavelet packet dictionary will allow.

#### 3.2.2 TwinSine

The signal TwinSine-1 consists of 2 cosines with frequencies closer together than the Rayleigh distance.

In Figure 2.5d, we analyze these in a factor 4- overcomplete Fourier dictionary (built from the DCT) containing the two cosines in questions as elements of the dictionary. Recall that in this example, MP failed, in the sense that it began by choosing at the first step a frequency in between the two ideal ones and then never corrected the error. In contrast, BP resolves the two frequencies correctly.

Next we consider TwinSine-2, again consisting of 2 cosines with frequencies closer together than the Rayleigh distance. Unlike with TwinSine-1, we shift the frequencies slightly so that they do not correspond to frequencies in the dictionary. In Figure 3.2, we
analyze these in a factor 4 overcomplete Fourier dictionary (built from DCT) containing the two cosines in questions as elements of the dictionary. This time MP fails yet again. As the figure shows, BP again resolves the two frequencies correctly.

### 3.2.3 FM Signal

Figure 3.3a displays the artificial signal **FM-Cosine** consisting of a frequency-modulated sinusoid superposed with a pure sinusoid:

\[
s = \cos(\xi_0 t) + \cos((\xi_0 t + \alpha \cos(\xi_1 t))t)
\]

In Figure 3.3b-d we analyze it using the Cosine Packet dictionary, based on a primitive bell of width 16 samples. It is evident that BOB cannot resolve the nonorthogonality between the sinusoid and the FM signal. Neither can MP. However, BP yields a clean representation of the two structures

### 3.2.4 Gong

Figure 3.4a displays the Gong signal, which is a decaying sinusoid that vanishes up until time \( t_0 \), and then follows an exponentially decaying sinusoid for \( t > t_0 \).

In Figures 3.4b-3.4d, we analyze it with the Cosine Packet dictionary, using a primitive bell of width 16 samples. BP gives the finest representation of the decay structure; visually somewhat better than the BOB and MP results.
3.3 Comparisons

We briefly compare BP with the three main methods introduced in section 2.5.

3.3.1 Matching Pursuit

At first glance MP and BP seem quite different. MP is an iterative algorithm, which does not explicitly seek any overall goal, but merely applies a simple rule repeatedly. In contrast, BP is a principle of global optimization without any specified algorithm. The contrast of MP with a specific algorithm, BP-Simplex, may be instructive. Matching Pursuit starts from an “empty model” and builds up a signal model an atom at a time, at each step adding to the model only the most important new atom among all those not so far in the model. In contrast, BP-Simplex, starts from a “full” model (i.e. representation of the object in a basis) and then iteratively improves the “full” model, by taking relatively useless terms out of the model, swapping them for useful new ones. Hence, MP is a sort of build-up approach, while BP-Simplex is a sort of swap-down approach.

3.3.2 Best Orthogonal Basis

To make BP and BOB most comparable, suppose that they are both working with the Cosine Packets dictionary, and note that the $l^1$-norm of coefficients is what Coifman and Wickerhauser [12] call an “additive measure of information”. So suppose we apply the Coifman-Wickerhauser Best Basis algorithm with entropy $E = l^1$. Then the two methods compare as follows: in BOB, we are optimizing $E$ only over orthogonal bases taken from the
dictionary, while in BP we are optimizing $E$ over all bases formed from the dictionary.

This last remark suggests that it might be interesting to apply the BOB procedure with the $\ell^1$ norm as entropy in place of the standard Coifman-Wickerhauser entropy. In Figure 2.7c we try this, on the WernerSorrows example of section 2.5.3. The signal is analysed in a Cosine Packets dictionary, with primitive bell width 16. The $\ell^1$ entropy results in a time-varying basis which reveals clearly some of the underlying signal structure. The $\ell^1$ entropy all by itself improves the performance of BOB; but BP does better still (Figure 2.7d).

This connection between BP and BOB suggests an interesting algorithmic idea. In the standard implementation of the simplex method for LP, one starts from an initial basis and then iteratively improves the basis by swapping one term in the basis for one term not in the basis, using the swap which best improves the objective functional. Which initial basis? It seems natural in BP-Simplex to use the Coifman-Wickerhauser algorithm and employ as a start the best orthogonal basis.

With this choice of starting basis, BP can be seen as a method of refining BOB by swapping in non-orthogonal atoms in place of orthogonal ones whenever this will improve the objective.

3.3.3 Method of Frames

As already discussed, MOF and BP differ in the replacement of an $l^2$ objective functional by an $l^1$ objective. BP-Interior has an interesting relation to the method of frames. BP-Interior uses a special "equi-weighted" starting point and iteratively solves a sequence of weighted least-squares problems, adjusting the weights at each stage based on the current decomposition. When used with the wavelet packets or cosine packets dictionaries, the initial starting point turns out to be exactly the same as the method of frames solution. Hence one can say that BP sequentially "improves" on the method of frames, by adaptively reweighting an underlying weighted least-squares problem.

Figure 3.5 shows a "movie" of BP-Interior in action on the FM-Cosine example, using a cosine packets dictionary. Six stages in the evolution of the phase plane are shown, and one can see how the phase plane improves in clarity, step-by-step.

3.4 Connections

We briefly list some connections of BP with important topics outside of signal processing.

3.4.1 Atomic Decomposition

The terminology "atomic decomposition" arose in the mathematical discipline of harmonic analysis [9]. Decompositions based on optimizations like BP have been used extensively in that field [29, 30, 36]; we give two examples.

Yves Meyer has described in his book on wavelets [51] the so-called Bump Algebra $B$, which is defined by a BP-type optimization. Begin with a dictionary $D_{Bump}$ of Gaussian "Bumps": $D = (\phi_\gamma : \gamma = (t, s))$, where $\phi_\gamma$ is the normalized Gaussian $\exp\{-(x-t)^2/s^2\}$. With $f(x)$ a function on the real line $R$, define the norm

$$\|f\|_B = \inf\{|\alpha|_1 : f = \sum \alpha_\gamma \phi_\gamma\}$$

22
The Bump algebra is the collection of all objects of finite $B$-norm

$$B = \{ f : \| f \|_B < \infty \}.$$  

Evidently, the objects in $B$ are defined by the same type of optimization as in BP. The two approaches differ in the specification of dictionary and in the choice of domain (discrete-time-signal versus function of a real variable).

Hans Feichtinger [28] has discussed the Segal Algebra, defined as follows. Start with the Gabor Dictionary $D_{Gabor} = (\phi_\gamma)$

$$\phi_\gamma = \cos(\omega t + \theta) \exp(-(x - t)^2/\sigma^2)$$

where $\gamma = (t, \sigma, \omega, \theta)$, and define the norm via

$$\| f \|_S = \inf \{ |\alpha|_1 : f = \sum \alpha_\gamma \phi_\gamma \}$$

The Segal algebra is the collection of all objects of finite $S$-norm

$$S = \{ f : \| f \|_S < \infty \}.$$  

Evidently, the objects in $S$ are defined by the same type of optimization as in BP. The two approaches differ in the specification of dictionary and in the choice of domain (discrete-time-signal versus function of a real variable).

This connection can help us to understand that there is a big difference between exact and approximate solution of the optimization problem defining BP. In the Bump Algebra case, Meyer shows that an orthogonal wavelet transform, with nice wavelets, gives a
fixed, nonadaptive, linear transform furnishing a near-solution to the nonlinear optimization problem defining \( B \). In the Segal Algebra case, Feichtinger shows that in an overcomplete Gabor-like dictionary, MOF gives a fixed, nonadaptive linear reconstruction formula which nearly solves the optimization problem defining \( S \). In short, linear methods give approximate solutions; exact solutions are, on the other hand, highly nonlinear.

### 3.4.2 Ill-Posed Problems

While we have not emphasized the point so far, it is inevitable that we should point out the unpleasant fact that there is a connection between *Overcomplete Representation* and ill-posed linear inverse problem.

As we have seen, nonuniqueness of overcomplete representations is equivalent to saying that the problem requires the solution of a linear system \( \Phi \alpha = s \) of less than full rank. There is a fully developed theory of linear inverse problems in which what we are calling the Method of Frames goes under the name “Generalized Inverse”, and which states that, in general, one cannot hope to do better than MOF. Such theory would seem to say that, in general, alternatives to MOF are based on wishful thinking.

However, it is known that for recovering *sparse* objects, one can do substantially better than MOF, and that one successful approach to doing so is based on minimization of \( \ell^1 \)-penalties [27]. An important phenomenon that occurs in the study of the \( \ell^1 \) method is *Super-Resolution*. The so-called Rayleigh limit is precisely the criterion for resolvability of two nearby spikes by the MOF; it is a limit which all linear methods obey. Super-resolution refers to the fact that when the true object to be recovered is highly sparse, nonlinear methods can resolve to finer than the Rayleigh limit [23], as the TwinSine examples showed.

This connection suggests that in the non-sparse case, there is little benefit from alternatives to MOF; but that in the sparse case, BP ought to be useful for high-resolution decomposition in overcomplete dictionaries.

### 4 Variations

The recent development of time-frequency dictionaries motivates most of what we have done so far. However, the methods we have developed are general and can be applied to other dictionaries – with interesting results.

#### 4.1 Stationary Smooth Wavelets

The usual (orthonormal) dictionaries of (periodized) smooth wavelets consist of wavelets at scales indexed by \( j = j_0, \ldots, \log_2(n) - 1 \); at the \( j \)-th scale, there are \( 2^j \) wavelets of width \( n/2^j \). The wavelets at this scale are all circulant shifts of each other, the shift being \( n/2^j \) samples. Some authors [60] have suggested that this scheme can be less than satisfactory, essentially because the shift between adjacent wavelets is too large. They would say that if the important “features” of the signal are (fortuitously) “aligned with” the wavelets in the dictionary, then the dictionary will provide a sparse representation of the signal; however, because there are so few wavelets at level \( j \), then most likely, the wavelets in the dictionary are not “precisely aligned” with features of interest, and the dictionary may therefore provide a very diffuse representation.

The stationary dictionary has, at the \( j \)-th level, \( n \) wavelets, (not \( 2^j \)); these are all the circulant shifts of the basic wavelet of width \( \approx n/2^j \). Since this dictionary always contains
wavelets "aligned with" any given feature, the hope is that such a dictionary provides a superior representation.

For the stationary dictionary, both $\Phi$ and $\Phi^T$ have fast implicit algorithms, which operate in $O(n \log_2(n))$ multiplications and $O(n \log_2(n))$ storage; hence the dictionary is suitable for our computational implementation.

Panel 4.1a shows the signal HeaviSine and 4.1b shows the result of BP with the Stationary Symmlet-8 dictionary mentioned in section 2.1.

There is a surprisingly close agreement of the BP representation in a stationary wavelet dictionary with ideas about signal representation associated with the "Multi-Scale Edges" ideas of Mallat and Hwang [46]. The Multi-Scale Edge method analyzes the continuous wavelet transform (CWT) at scale $2^{-j}$ and identifies the maxima of this transform. Then it selects those maxima which are "important" by thresholding based on amplitude. These "important" maxima identify important features of the signal. An iterative method for reconstruction is proposed by Mallat and Hwang which reconstructs an object having the same values of the CWT at "maxima". This is almost (but not quite) the same thing, as saying that one is identifying "important" wavelets located at the corresponding maxima, and reconstructing the object using just those maxima.

Panel 4.1c shows a CWT of HeaviSine based on the same Symmlet-8 wavelet; Panel 4.1d shows the maxima of the CWT. At fine scales, there is virtually a 1-1 relationship between the maxima of the transform and the wavelets selected by BP; compare panel 4.1b. So in a stationary wavelet dictionary, the global optimization principle BP yields results which are close to certain heuristic methods.

An important contrast: Meyer has a counterexample to multi-scale edge approaches, which shows that the Mallat-Hwang approach may fail in certain cases [53]; but there can be no such counterexamples to BP.

4.2 Decaying Exponentials

In several scientific fields – Spectroscopy, Free Oscillations of the Earth, Helioseismology included – one obtains data which, according to the physics of the situation, amount to superpositions of decaying sinusoids, decaying exponentially at various rates. It is interesting to consider applying BP to the analysis of such signals, and hence looking at a dictionary indexed by $\gamma = (\omega, \delta t, \nu)$, where $\omega$ is an oscillation frequency, $\delta t$ is a relaxation time and $\nu$ is a phase indicator, with

$$\phi_{(\omega, \delta t, 0)} = \exp(-t/\delta t) \cdot \cos(\omega t), \quad \phi_{(\omega, \delta t, 1)} = \exp(-t/\delta t) \cdot \sin(\omega t).$$

In forming dictionaries with this indexing, many choices are possible. If one considers a discrete dictionary with a single fixed duration $\delta t$, one is modelling the situation as mono-decay-time situation (all components decay at the same rate). If one allows a list of decays, one has a multiple-decay time situation. If one allows just sinusoids at the Fourier frequencies of the underlying grid, resolution is only possible up to the Rayleigh distance; if one uses sinusoids at a finer grid, resolution is being attempted at sub-Rayleigh criteria.

Because of the similarity to undamped sinusoids, it is not hard to develop fast implicit algorithms for $\Phi$ and $\Phi^T$ for a mono-decay-time dictionary, or for any dictionary with a small number of decay-times. We often use a geometric progression of decay times $2^\nu$, so that there are only logarithmically many decay times to be considered; the resulting algorithms require $O(n \log_2(n)^2)$ multiplications and $O(n \log_2(n))$ storage.
Panel 4.2a shows the signal MultiGong, which is a pure superposition of three decaying sinusoids. Panel 4.2b displays the decomposition in a multi-duration dictionary of decaying sinusoids, using a display showing frequency crossed with duration. A sparse decomposition has been obtained, despite the fact that none of the decay times of the underlying signal belong to the geometric progression of decay times in the dictionary.

4.3 Dictionary Mergers

An important methodological tool is the ability to combine dictionaries to make bigger, more expressive dictionaries. We mention here two possibilities.

Jump+Sine. Merge the Heaviside dictionary with a Fourier dictionary. Either dictionary can efficiently represent objects that the other cannot; for example, Heavisides have difficulty representing sinusoids, while sinusoids have difficulty representing jumps. Their combination might therefore be able to offer the advantages of both. Figure 4.3 shows this to be so; panels (b) and (c) show the coefficients of representations in terms of “pure heaviside” or “pure sinusoid” dictionaries; panels (c) and (d) show the considerably sparser coefficients of a mixed representation in terms of both heaviside and sinusoid terms. Compare also Figure 5.4b.

Jump+Wavelet. For similar reasons, one might want to merge Heavisides with Wavelets. In fact, we have found it sometimes preferable instead to merge “tapered heavisides” with wavelets; these are step discontinuities which start at 0, jump at time $t_0$ to a level one unit higher, and later decay to the original 0 level. Figure 5.5b shows that the merged dictionary can give a more sparse decomposition of object HeaviSine than either of the individual dictionaries alone.
Figure 4.5: BP Example

Signal Length: 256

(a) Signal: Heavisine

(b) Pure Heaviside Representation

(c) Pure Sinusoid Representation

(d) Jump Coef by BP using Jump+Sine

(e) Sine Coef by BP using Jump+Sine

Signal Length: 64

Heavisine

HS-DST with DST oversampledness = 2
Many other possibilities might be considered: mergers of Dirac and exponentially decaying sinusoids are especially useful in NMR signal processing, for example.

4.4 Dictionary Alteration

There is an interesting “stepwise” method of treating problems involving a merger of several dictionaries \( D_k, 1 \leq k \leq m \). Let \( D_0 \) denote an initial orthogonal dictionary, such as the standard basis.

For \( 1 \leq k \leq m \), execute the following loop.

1. Define the current dictionary \( D = D_0 \cup D_k \).

2. Solve BP in the current dictionary.

3. The result of step 2 will utilize at most \( n \) nonzero coefficients. Redefine \( D_0 \) to consist of exactly those \( n \) corresponding atoms.

This approach is most interesting in connection with BP-Simplex. In usual BP-Simplex, one is iteratively improving a decomposition by making swaps of atoms in the current decomposition for atoms not in the current decomposition. In the variant above, one is restricting the swaps to come from certain subdictionaries, i.e. at one stage one only considers improvements using swaps for Heavisides, at the next stage only swaps for Sines, etc.

As an example, Figure 4.4a-b shows the HeaviSine signal decomposed in the initial dictionary – sines only; Figure 4.4c-d shows the decomposition after Heavisides are swapped in. Figure 4.4e-f shows the decomposition after Sines are swapped in. And so on. The iterative approach quickly converges in this case.

An attractive feature of this approach is the way that it nicely modularizes the job of dealing with several dictionaries.

4.5 Scaling of the Dictionary

An important issue we have not so far discussed is the issue of how dictionary elements should be normalized. We have assumed so far that the dictionary is normalized in \( \ell^2 \), but this is not obligatory. By changing the normalization conventions, we can change the nature of the decomposition drastically, because we essentially change the nature of the objective function. In general, suppose we have two dictionaries, \( D_0 = \{ \phi_\gamma \} \) and \( D_1 = \{ \psi_\gamma \} \) which differ in that \( \phi_\gamma = \sigma_\gamma \psi_\gamma \), for scalars \( \sigma_\gamma \). Then BP with dictionary \( D_1 \) is the same as solving a weighted \( \ell^1 \) optimization problem in dictionary \( D_0 \):

\[
\min \sum_\gamma |a_\gamma \sigma_\gamma| \quad \text{Subject to} \quad \sum_\gamma a_\gamma \phi_\gamma = s. \quad (4.1)
\]

When dealing with time-scale or time-frequencies, it makes sense to consider the \( \ell^\infty \) normalization of atoms as used in the definition of the Bump Algebra and Segal Algebra, in which case \( \sigma_\gamma = \|\phi_\gamma\|_2/\|\phi_\gamma\|_\infty \) which means that \( \sigma_\gamma \) is proportional to the reciprocal of the square root of duration of the waveform. The weighted optimization problem (4.1) then penalizes the inclusion of very short-duration atoms in the decomposition.

One could profitably consider normalizations which penalize scale even more severely; these would correspond to weighted optimizations measuring essentially the norms of certain smoothness spaces. In that case, one is seeking the “smoothest” representation of an object.
The $\ell_\infty$ normalization will be employed below in decompositions using wavelets and tapered Heavisides.

5 De-Noising

We now adapt BP to the case of noisy data. We assume data of the form

$$y = s + \sigma z$$

where $(z_i)$ is a standard white Gaussian noise, $\sigma > 0$ is a noise level, and $s$ is the clean signal. In this setting, $s$ is unknown, while $y$ is known. We don't want to get an exact decomposition of $y$, so we don't apply BP directly, instead decompositions like (1.2) becomes relevant.

5.1 Proposal

Basis Pursuit De-Noising (BPDN) refers to solution of

$$\min_a \frac{1}{2} \|y - \Phi a \|^2_2 + \lambda \cdot \|a\|_1$$

The solution $a^{(\lambda)}$ is a function of the parameter $\lambda$. It yields a decomposition into signal-plus-residual

$$y = s^{(\lambda)} + r^{(\lambda)}$$

where $s^{(\lambda)} = \Phi a^{(\lambda)}$. The size of the residual is controlled by $\lambda$. As $\lambda \to 0$, the residual goes to zero and the solution behaves exactly like BP applied to $y$. As $\lambda \to \infty$, the residual gets large; we have $r^{(\lambda)} \to y$ and $s^{(\lambda)} \to 0$. 

29
The optimization problem can be posed as an instance of quadratic programming with inequality constraints. Using the same notation as in section 3.1,

$$\min \| y - \Phi(u - v) \|^2 + \lambda \cdot 1^T u + \lambda \cdot 1^T v \text{ subject to } u, v \geq 0. \quad (5.2)$$

Setting $x = (u, v)$, (5.2) can be reexpressed as

$$\min \| A^T x \|^2 + b^T x + c \text{ subject to } x \geq 0, \quad (5.3)$$

for suitable $A$, $b$ and $c$; this is minimization of an inhomogeneous quadratic form in $x$, subject to positivity constraints.

Quadratic programming is almost as venerable a subject as linear programming, and there is a considerable amount of literature on the subject. Key fact: quadratic programming retains much of the structure of linear programming, and most of the linear programming methodology can be adapted to this problem. Hence we can speak of specific types of algorithms: BPDN-Simplex and BPDN-Interior.

5.2 Choice of $\lambda$

For definiteness, we generally tune $\lambda$ as follows. Assuming the dictionary is normalized so that $\| \phi_\gamma \|_2 = 1$ for all $\gamma$, we set $\lambda$ to the value

$$\lambda_p = \sqrt{2 \log(p)} \cdot \sigma$$

where $p$ is the cardinality of the dictionary.

This can be motivated as follows. In the case of a dictionary which is an orthonormal basis, a number of papers [21, 26] have carefully studied an approach to de-noising by so-called “soft-thresholding in an orthonormal basis”. In detail, suppose that $\Phi$ is an orthogonal matrix, and define empirical $\phi$-coefficients by

$$\hat{y} = \Phi^T y.$$  

Define the soft threshold nonlinearity $\eta_\lambda(y) = \text{sgn}(y) \cdot (|y| - \lambda)_+$ and define the thresholded empirical coefficients by

$$\hat{\alpha}_\gamma = \eta_\lambda(\bar{y}_\gamma), \quad \gamma \in \Gamma.$$  

This is soft thresholding of empirical orthogonal coefficients. The papers just cited show that thresholding at $\lambda_n$ has a number of optimal and near-optimal properties as regards mean-squared error.

We claim that (again in the case of an ortho-basis) the thresholding estimate $\hat{\alpha}$ is also the solution of (5.1). Observe that the soft thresholding nonlinearity solves the scalar minimum problem:

$$\eta_\lambda(y) = \frac{1}{2} \arg \min_\xi (y - \xi)^2 + \lambda \cdot |\xi|; \quad (5.4)$$

Note that, because of the orthogonality of $\Phi$,

$$\| y - \Phi a \|^2 = \| \bar{y} - a \|^2$$

and so we can rewrite (5.1) in this case as

$$\min_a \frac{1}{2} \sum_\gamma (\bar{y}_\gamma - a_\gamma)^2 + \lambda \cdot \sum_\gamma |a_\gamma|.$$
Now applying (5.4) coordinatewise establishes the claim.

The scheme we have suggested here – to be applied in overcomplete as well as orthogonal settings – therefore includes soft-thresholding in ortho-bases as a special case. Formal arguments similar to those in [25] can be used to give a proof that mean-squared error properties of the resulting procedure are near-optimal under certain conditions.

An important remark: the choice of \( \lambda \) given here is based on the assumption that the atoms are \( \ell^2 \)-normalized: \( \| \phi_i \|_2 = 1 \). The investigation of how to choose \( \lambda \) for other normalizations is underway.

5.3 Examples

We present three examples of BPDN in action with time-frequency dictionaries.

5.3.1 Carbon

Figure 5.1 displays de-noising results on the signal Carbon, at signal to noise ratio 1. Panel a) displays the noiseless signal; panel b) displays a noisy version. Panels c)-f) display de-noising results for MOF, BOB, MP, and BP, respectively. It is visually evident that BP obtains a pleasing, artifact-free reconstruction, although the amplitude is somewhat attenuated.

5.3.2 Gong

The signal Carbon is unusual, as it consists of an exact superposition of 6 atoms in the dictionary being used also for analysis. In contrast, naturally acquired data would almost
never consist of a precise superposition of atoms in the dictionary being used. The signal Gong being a decaying sinusoid, is not an exact superposition of a few atoms in either the wavelet packet or cosine packet dictionaries. Figure 5.2 displays results at signal to noise ratio 1. Panel a) displays the noiseless signal and panel b) displays a noisy version. Panels c)-f) display de-noising results for MOF, BOB, MP, and BP, respectively. It is visually evident that BP outperforms the other methods.

5.3.3 TwinSine

Figure 5.3 employs the signal TwinSine-1, described earlier, to investigate super-resolution in the noisy case. Panels a) and b) give the noiseless and noisy TwinSine-1, respectively. Using a 4-fold overcomplete dictionary of cosines, reconstructions by the MOF, MP, and by BPDN are given. MOF gives a reconstruction which is inherently resolution-limited and oscillatory. As in the noiseless case, MP gives a reconstruction which goes wrong at step 1 – it selects the average of the two frequencies in the TwinSine-1 signal. BP correctly resolves the non-negative doublet structure.

5.4 Total Variation De-Noising

Recently, Rudin, Osher and Fatemi [56] have called attention to the possibility of de-noising images using total-variation penalized least-squares. More specifically, they propose the optimization problem

$$\min_{g} \frac{1}{2}||y - g||_2^2 + \lambda \cdot TV(g)$$  (5.5)
where $TV(g)$ is a discrete measure of the total variation of $g$. A solution of this problem is the de-noised object. Li and Santosa [44] have developed an alternative algorithm for this problem based on interior point methods for convex optimization.

For the 1-dimensional case (signals rather than images) it is possible to implement what amounts to total variation de-noising by applying BPDN with a Heaviside dictionary. Indeed, if $s$ is an arbitrary object, it has a unique decomposition in Heavisides (recall (2.1)). Suppose that the object is 0 at $t = 0$ and $t = n - 1$, and that the decomposition is $s = \sum_i a_i H_{ti}$; then the total variation is given by

$$TV(s) = \sum_{i \neq 0} |a_i|.$$ 

Moreover to get approximate equality even for objects not obeying zero-boundary conditions, one has only to normalize $\phi_0$ appropriately. Consequently, total variation denoising is essentially a special instance of proposal (5.1).

We have studied BPDN in the Heaviside dictionary, thereby obtaining essentially a series of tests of TV De-Noising. For comparison, we considered also soft thresholding in orthogonal wavelet dictionaries based on the S8-Symmlet smooth wavelet. We also constructed a new dictionary, based on the $Jump+Wave$ merger of S8-Symmlet wavelets with “Smoothly Tapered Heavisides”, which is to say, atoms $\phi_\gamma$, which jump at a given point $\gamma$ and then decay smoothly away from the discontinuity. For comparability with the Heaviside dictionary, we normalized the $Jump+Wave$ dictionary so that every $\|\phi_\gamma\|_{TV} \approx 1$.

A typical result, for the object $\text{Blocky}$, is presented in Figure 5.4. From the point of view of visual appearance, total variation reconstruction (panel d) far outperforms the other methods.
Of course, the object Blocky has a very sparse representation in terms of Heavisides. When we consider an object like \texttt{HeaviSine}, which is piecewise smooth rather than piecewise constant, the object will no longer have a sparse representation. On the other hand, using the \texttt{Jump+Wave} dictionary based on a merger of wavelets with tapered Heavisides will lead to a sparse representation – Figure 5.5c. One can predict that a Heaviside dictionary will perform less well than this merged dictionary.

This completely obvious comment, translated into a statement about total variation denoising, becomes a surprising prediction. One expects that the lack of sparse representation of smooth objects in the Heaviside dictionary will translate into worse performance of TV denoising than of BPDN in the merged \texttt{Jump+Wave} dictionary.

To test this, we conducted experiments. Figure 5.5 compares TV de-noising, wavelet de-noising, and BPDN in the merged \texttt{Jump+Wave} dictionary. TV De-Noising now exhibits visually distracting stairstep artifacts; the dictionary \texttt{Jump+Wave} seems to us to behave much better. Figure 5.6 gives similar pictures for the smooth, spatially variable object Cusp.

6 Solutions of Large-Scale Linear Programs

As indicated in section 3.1, the optimization problem (3.1) is equivalent to a linear program (3.2). Also, as in section 5.1, the optimization problem (5.1) is equivalent to a quadratic program (5.3). The problems in question are large-scale; we have conducted decompositions of signals of length \( n = 8192 \) in the wavelet packet dictionary, leading to a linear program of size 8192 by 212,992.
Over the last ten years there has been a rapid expansion in the size of linear programs which have been successfully solved using digital computers. Much of the practical work has been done by researchers in the Operations Research community, studying large-scale optimization problems arising, for example, in airline scheduling. Such problems tend to have matrices \( A \) which are sparse but which are otherwise without special algebraic structure. In 1985 it would have been considered ambitious to solve 1000 by 1000 linear programs of the kind arising in various operations research problems; such problems would have been considered the domain of supercomputers and large mainframe computers. By 1994, programs of this size were considered accessible by laptop computers, and supercomputers had been used to solve problems such as model Delta 2 in [45], where \( n \) was 45, 116 and \( m \) was 101, 790. A good overview of the recent rapid progress in this field and the current state of the art is afforded by the article of Lustig, Marsten and Shanno [45] and the accompanying discussion by Bixby [4], Saunders [59], Todd [63], and Vanderbei [64].

Much of the rapid expansion in the size of linear programs which have been solved is due to the "interior point revolution" initiated by Karmarkar's proof that a pseudopolynomial time algorithm could be based on an interior-point method [41]. Since then a very wide array of interior point algorithms have been proposed and considerable practical [45] and theoretical [55] understanding is now available. (At the same time, the existence of practical alternatives to the time-honored simplex method has led to major improvements in simplex implementation, so that even the simplex method has improved radically in the size of problems treated in the last ten years [4].)

6.1 Log-Barrier Method for LP

Mathematical work on interior point methods over the last ten years has led to a large variety of approaches, with names like "projective scaling", "(primal/dual) affine scaling", "(primal/dual) logarithmic barrier", "predictor-corrector". We cannot summarize all these ideas here; many of them are mentioned in [45] and others are covered in the references of that article.

Our approach relies on the Primal-Dual Logarithmic Barrier Method from the mathematical programming literature. This is described in [45], which we follow here, who credit [49, 43] for the method. In outlining this algorithm, one needs to introduce not only the standard LP as discussed above, but also the associated dual LP, which takes the form

\[
\max b^T y \text{ subject to } A^T y + z = c, \quad z \geq 0;
\]

here \( A, b, \) and \( c \) are the same as in the original, "primal" LP, the state variables \( y \) are called dual to \( x \), and \( z \) are new slack variables. One associates the dual with a logarithmic barrier problem

\[
\max b^T y + \mu \sum_{i=1}^{m} \ln z_i \text{ subject to } A^T y + z = c
\]

where the constraint \( z_i > 0 \forall i \) is implicit in the use of the log term in the objective. As \( \mu \to 0 \), the solution of the logarithmic barrier problem tends to the solution of the dual LP.

Define the Lagrangian

\[
L_\mu(x, y, z) = b^T y + \mu \sum_{i=1}^{m} \ln z_i - x^T (A^T y + z - c),
\]

36
and define the primal-dual barrier problem as the solution of the saddlepoint problem

\[(BP(\mu)) \quad \min_{x \geq 0} \max_{z \geq 0, y} L_\mu(x, y, z).\]

Solutions \((x^{[\mu]}, y^{[\mu]}, z^{[\mu]})\) of the saddlepoint problem are related to solutions of the LP: as \(\mu \to 0\), the solutions \(x^{[\mu]}, y^{[\mu]}, z^{[\mu]}\) of the Lagrangian tend to \(x^*, y^*, z^*\) which furnish the solutions of the LP and its dual.

A saddlepoint of the Lagrangian must satisfy the first order conditions

\[
\begin{align*}
XZe &= \mu e \\
Ax &= b \\
A^T y + z &= c
\end{align*}
\]

where the matrix \(X\) is a diagonal matrix with components of \(x\) on the diagonal, \(Z\) is also a diagonal matrix with components of \(z\) on the diagonal, and \(e\) is a vector of all ones.

To find a solution to this (nonlinear) system of equations, one might consider applying Newton’s method, getting an adjustment to the current guess \((x^{(k)}, y^{(k)}, z^{(k)})\) of the form

\[
x^{(k+1)} = x^{(k)} + \Delta x, \quad y^{(k+1)} = y^{(k)} + \Delta y, \quad z^{(k+1)} = z^{(k)} + \Delta z,
\]

where the increments satisfy

\[
\begin{align*}
Z^{(k)} \Delta x + X^{(k)} \Delta z &= \mu e - X^{(k)} Z^{(k)} e \\
A \Delta x &= b - Ax^{(k)} \\
A^T \Delta y + \Delta z &= c - A^T y^{(k)} - z^{(k)} \tag{6.6}
\end{align*}
\]

The system (6.6) is the basis for an approach where one applies this Newton step with \(\mu\) decreasing from iteration to iteration. This can be motivated by the idea that solving a sequence of barrier problems \((BP(\mu_k))\) with \(\mu_k\) tending to zero will, in the limit, produce a solution to the corresponding LP.

In detail, we utilize the following schema

1. Initialize \(x^{(0)}, y^{(0)}, z^{(0)}, \mu^{(0)}\). Set \(k = 0\). Specify \(\delta > 0\).

2. while \((x^{(k)})^T z^{(k)} > \delta\),
   
   (a) Solve for \(\Delta x^{(k)}, \Delta y^{(k)}, \Delta z^{(k)}\)
   
   (b) Calculate the primal and dual step sizes \(\rho_p, \rho_d\):

   \[
   \begin{align*}
   \rho_p &= \min_{\Delta x_i < 0} (\max(\Delta x_i) \cdot .9995) \\
   \rho_d &= \min_{\Delta z_i < 0} (\max(\Delta z_i) \cdot .9995)
   \end{align*}
   \]

   (c) Update the variables:

   \[
   \begin{align*}
   x^{(k+1)} &= x^{(k)} + .9995 \rho_p \Delta x \\
y^{(k+1)} &= y^{(k)} + .9995 \rho_d \Delta y, \\
z^{(k+1)} &= z^{(k)} + .9995 \rho_d \Delta z \\
\mu^{(k+1)} &= (1 - \min(\rho_p, \rho_d, .9995)) \mu^{(k)}
   \end{align*}
   \]

37
(d) Increase \(k\) by 1.

For fuller discussions of this and related algorithms, again see [45], or references there. While in principle we could have based our approach on other interior point schemes, the primal-dual approach naturally incorporates several features we found useful:

- Feasible initial values \(x^{(0)}\) lead to iterates \(x^{(k)}\) which remain feasible throughout the sequence of iterations.

- It is easy to check for closeness to the solution value; at the limiting solution \(c^T x^* = b^T y^*\), and the duality gap \(c^T x^{(k)} - b^T y^{(k)}\) quantifies the distance from this ideal.

Modifications of basic Newton steps, like predictor-corrector schemes and power series schemes, have been found useful in the optimization literature [45, 42]; these remain interesting ideas for future investigation.

### 6.2 Implementation Heuristics: Operations Research

A mathematical idea, like the primal-dual log barrier idea just described, is rather far removed from a practical implementation in working software. Such implementations are based on a variety of ideas and heuristics which make up an extensive body of knowledge forming the current state of the art in numerical implementation of interior-point methods, as catalogued, say in [45] and the ensuing discussion.

At the moment the successful numerical implementations we are aware of have settled on a collection of properties of the underlying mathematical ideas which many major implementations seek to exploit:

- The algorithms are iterative, starting from an initial feasible solution located at or near the "center" of the feasible region, and iteratively improving the current feasible solution. These are motivated by mathematical results showing that the infinite iteration limit is a solution of the linear program.

- Numerical implementations stop after a relatively small number of iterations: for example, on moderate sized problems a few dozen iterations would be common.

- Each iteration requires the solution of a system of equations involving \(A, A^T\), and other problem data like \(x^{(k)}\). In the primal-dual log barrier method, the system is

\[
ADA^T \triangle y^{(k)} = Au
\]

where \(D = X^{-1}Z\) and \(u = Z^{-1}(XZe - \mu e)\).

Thus the numerical solution to a linear program by good interior point methods amounts to a sequence of several dozen solutions of special systems of linear equations. This leads to a slogan: if those systems can be solved rapidly, then it is possible to rapidly solve the LP.

Of course, in general solving systems of equations is not rapid: a general \(n \times n\) system \(Bx = c\) takes order \(O(n^2)\) time to solve by standard elimination methods or by modern stable factorization schemes [37, 34]. So in order for practical algorithms to be based on the interior point heuristic, it is necessary to be able to solve the systems of equations which arise much more rapidly than one could solve general systems. In the current state of the
art [59], one attempts to do this by exploiting sparsity of the underlying matrix $A$. In the operations research setting, the matrix $A$ underlying the linear program is often sparse, containing far fewer than $n \times m$ nonzero elements in the $n$ by $m$ array $A$. For example in the airline scheduling problem Delta Model 2 of size 45,116 by 101,790 studied in [45], the matrix $A$ has only a 342,180 nonzero elements, not the several billion that it might conceivably have had. One also acts on faith that the sparsity of $A$ will carry over to Cholesky factorization of $ADA^T$; often this is the case, but there are examples where the Cholesky factorization is effectively dense (one speaks of catastrophic fill-in in such cases).

In short, the computational state of the art is based on three heuristics: (i) that one need solve only a limited number of systems of linear equations; (ii) that the individual systems are sparse; and (iii) that they can be rapidly solved by sparse Cholesky factorization techniques.

6.3 Implementation Heuristics: Basis Pursuit

The optimization problems we are interested in have a key difference from the successful large-scale applications outlined in [45]. The matrix $A$ we deal with is not at all sparse; it is generally completely dense. For example, if $A$ is generated from a Fourier dictionary, most of the elements of $A$ will be of the same order of magnitude. Because of this density, it is unlikely that existing large scale interior point computer codes could be easily applied to the problems described in this paper.

In our application we have a substitute for sparsity, which goes back to our comments in section 2.2. In this work we consider only dictionaries which have fast implicit algorithms for $\Phi a$ and $\Phi^T a$, and which therefore lead to linear programs where the $A$ matrix admits of fast implicit algorithms for both $Au$ and $A^Tv$. Now whenever one has fast implicit algorithms, it is natural to think of solving equations by conjugate gradient methods; such methods allow one to solve equations $Bw = b$ using only applications $Bw^{(k)}$ to various strategically chosen vectors $w^{(k)}$. Adapting such ideas, one develops fast implicit algorithms for $ADA^T w^{(k)}$ and attempts to solve the central equations (6.7) avoiding the costly step of explicitly forming or operating with the matrices $ADA^T$ or $A$.

In our application, we also have help from the fact that we do not really need an exact solution of the optimization problem. In our application, we have a natural initial solution – from the Method of Frames – which would be viewed by some researchers as already an acceptable method of atomic decomposition. By starting from this decomposition and applying a strategy based on a limited number of iterations of our algorithm, we get what we view as an iterative improvement on MOF. In our opinion, even when this results in stopping too soon, getting an answer far from the ideal answer, we are going some of the way towards an ideal answer. This ideal answer is perhaps desirable to obtain, but it is also potentially very costly to obtain. Our point of view is better understood by examining Figure 3.5, which shows that even intermediate steps of the BP-interior algorithm, long before convergence, correspond to useful improvements over the Method of Frames decomposition.

Codifying this approach, our strategy for routine numerical implementation of BP is as follows:

- We employ the framework set by the "primal-dual logarithmic barrier method" from the mathematical programming literature.

- We suppose fast implicit algorithms for $ADA^T \Delta y^{(k)}$ and $Au$. 

39
• We take 10 to 30 Newton steps.

• Each Newton step involves *approximate* solution of the central equations (6.7) using conjugate gradient ideas.

• The conjugate gradient solver is run a certain number of iterations, that number following a schedule based on the Newton step number. For example at early steps, we might go 5 iterations and at late steps several dozen.

This strategy for routine numerical processing has the following features

• It does not guarantee a solution at termination.

• It does guarantee reasonable solution time and linear storage requirements.

These features are suited for settings where routine processing is to be performed and it seems more important to have moderate processing time than ultimate resolution and sparsity.

In certain cases, we would instead apply a more literal translation of the log-barrier method into software, using formal convergence criteria. In such cases we would be willing to allow the computational effort to vary according to problem difficulty.

It is important to acknowledge freely that the key heuristics we are employing in the above strategy have not been the subject of rigorous analysis and verification.

First, the heuristic that "a few dozen Newton steps are enough" is based on empirical results with so-called "long-steps" algorithms. Most theoretical work is about "short-steps" algorithms, which take steps much, much shorter than the step lengths we have proposed above, and which seem to require a number of steps which grows with the problem dimension like $\sqrt{n}$. Algorithms requiring so many steps seem to be impractical for our purposes, because each step is necessarily expensive. Little theory is available for long-steps algorithms, but Todd [63] has constructed special linear programs in which the algorithm requires at least $n^{1/3}$ steps (of any length) to make a significant progress towards a solution. The empirical work seems to show that in practice, on usual linear programs, the number of required steps of usual "long-steps" algorithms grows more like $\log(n)$. Theoretical support for this empirical experience is so far lacking.

Second, the heuristic that "the underlying linear systems can be solved approximately using conjugate gradients" is again based on empirical results. Previous work at ATT Bell Labs [8] and by Mehrotra [50] has discussed the possibility of using preconditioned conjugate methods. Lustig, Marsten, and Shanno are openly critical of what they consider over-optimistic claims for the CG approach [45, Rejoinder]. There is not a great deal of convergence theory for interior-point methods based on approximately-solved systems; perhaps the chapter in the recent book of [55] will call attention to the need for more theoretical understanding of this issue.

We stress that our strategy is to "pursue an optimal basis"; while we would like to reach the optimal basis, we make no specific claims that we can always reach it in reasonable time; perhaps the "pursuit" language will help remind one of this fact. We do believe that the pursuit process, carried out for whatever length of time we are willing to invest in it, makes a useful improvement over the Method of Frames.
6.4 Log Barrier Algorithm for BPDN

Much of the spirit of what has been so far said applies to the quadratic programming setting. We skip the preliminaries and jump to a description of the main iteration, which is based again on logarithmic barrier methods. Form the Lagrangian

\[ L(x, \mu) = c^T x + \frac{1}{2} \| b - Ax \|_2^2 - \mu \sum_{i=1}^m \ln x_i; \]

our goal is to minimize this as a function of \( x \). The barrier constraint forces consideration only of \( x_i > 0 \).

The Newton step from the necessary conditions for this problem may be derived as follows. Let \( r = b - Ax \) and note that

\[ \nabla_x L = c - ATr - \mu X^{-1}e \]
\[ \nabla^2_x L = ATA + \mu X^{-2} \]

here the matrix \( X \) is again diagonal with the entries of \( x \) along the diagonal.

From these expressions we get that the Newton step \( \Delta x \) solves

\[ (ATA + \mu X^{-2}) \Delta x = -c + ATr + \mu X^{-1}e \]

Let \( \Delta x = Xz \); the Newton step can be rewritten as

\[ (XATX + \mu I)z = \mu e + XATr - Xc \]

which is the same as solving the least-squares problem

\[ \min_z \| \begin{pmatrix} r \\ s \end{pmatrix} - \begin{pmatrix} AX \\ \sqrt{\mu}I \end{pmatrix} z \|_2 \quad (6.8) \]

where \( s = \sqrt{\mu e - Xc}/\sqrt{\mu} \).

We can derive an approach to solving the QP by considering a sequence of Newton steps with \( \mu \) varying from iteration to iteration.

1. Initialize \( x^{(0)}, \mu^{(0)} \), set \( k = 0 \), and specify \( \delta > 0 \).

2. While \( \mu^{(k)} > \delta \)
   - Solve for \( \Delta x^{(k)} \) using (6.8).
   - Calculate the step size \( \rho \)
     \[ \rho = \min_{\Delta x_i < 0} (x_i / |\Delta x_i|) \cdot 0.9995 \]
   - Update the variables:
     \[ x^{(k+1)} = x^{(k)} + 0.9995 \rho \Delta x^{(k)} \]
     \[ \mu^{(k+1)} = (1 - \min(\rho, 0.9995)) \mu^{(k)} \]
   - Increment \( k \) by 1.
6.5 Reproducible Research

The foregoing discussion does not do justice to the details involved in the full-scale implementation of Atomizer. As a complement to this article, we are releasing the underlying software environment by placing it on internet for access either by anonymous FTP or World-Wide-Web browsers.

Web Browser: http://playfair.stanford.edu/~wavelab
FTP Client: playfair.stanford.edu directory: pub/chen_s

This paper has been written following the discipline of Reproducible Research described in [6].

- All the code underlying all the figures in this article is made available,
- Together with the underlying software environment necessary to execute that code,
- Together with documentation of both the tools and environment,
- Using standard internet methods (FTP, WWW) for anonymous, 24-hour-a-day access.

As a result any reader can try out our methods to reproduce the figures, can try them out with different choices of parameter settings, and can try them out with different datasets. This should make it much easier to understand and criticize our work than would be possible with a verbal description of the underlying ideas, which, no matter how thorough, would necessarily lack certain implementation details.

7 Discussion

7.1 Why so Many Figures?

As a result of adhering to our discipline of Reproducible Research, we distribute by Internet code to reproduce the figures of this article. This gives a special rationale for including figures in this article. Our publication of a figure is not merely the advertising of a specific computational result; it is the publication of an algorithm, of datasets, and of a series of scripts which invoke the algorithm and datasets. The figures in this article are only the visible result of this publication process. The present article serves not only to announce the possibility and benefits of Basis Pursuit, but also to make available, worldwide and electronically, the specific computational tools which are required to apply it to a variety of problems.

7.2 What's in a Name?

The field of overcomplete representations in signal analysis is full of ideas which appeared first elsewhere, under other names.

- The Method of Frames is “the same as” the method of generalized inverses; various algorithms for obtaining the MOF solution are equivalent to well-known methods in linear algebra, such as relaxation and conjugate gradients, etc.

- Orthogonal Matching Pursuit is “the same as” what statisticians call “stepwise regression” (though MP in its original form would better be called “stagewise regression”).
• The MP algorithm, in a "point-spread" dictionary, is essentially what Astronomers in the 1970's call christened algorithm Clean.

• And so on.

In the same vein, Basis Pursuit is "just" a variant of minimum $\ell^1$ estimation.

Observations like this show that it is of course unnecessary to introduce new names like "Basis Pursuit". On the other hand, names serve as organizing tools which help to arrange and demarcate lines of research. Introducing the name "Basis Pursuit" helps to make clear the domain of application in many of the same problems as Matching Pursuit, and our emphasis on the operating characteristics of the method in a certain class of problems. For example, it is not quite right to say that our software implementation finds the minimum $\ell^1$ solution in general; it may be impractical in some problems to take enough Newton steps to obtain the full $\ell^1$ solution; but all the evidence points to the fact that going in the direction of an $\ell^1$ solution - pursuit - might still be useful. Also, hopefully, the name Basis Pursuit will conjure up a specific application of $\ell^1$ methods - and a range of vocabulary (dictionaries, sparsity, superresolution) and methodology (dictionary scaling, merged dictionaries) that goes with that application.

7.3 Direct Antecedents

While above we have drawn loose connections with literature in related fields, we would also like to acknowledge that there is a considerable body of work directly related to the present project.

Interior Point Methods for Minimum $\ell^1$ Problems. We have learned from Fadil Santosa that Coleman and Li [13] have developed interior point methods for $\ell^1$ norm minimization problems. Their work differs in that it is considered to apply to minimizing the $\ell^1$ norm $|y - \Phi\alpha|_1$ of residuals from decompositions in an undercomplete dictionary, with no constraint on $\alpha$, while we consider instead minimizing the $\ell^1$ norm of coefficients $||\alpha||_1$ in an overcomplete dictionary with the constraint $y = \Phi\alpha$. Their work does not specifically discuss large-scale problems or the CG-type iterations we use, and the details of their algorithm differ, but their work could be adapted to solve problems of our type. Their algorithmic approach differs from ours in that they apply an affine-scaling approach, while we apply a primal-dual logarithmic barrier method.

Interior Point Methods in Large-Scale Image Processing. In fact, Li and Santosa [44] have adapted the Coleman-Li algorithm to implement Total-Variation De-Noising of 2-d images - in an alternative to the algorithm of Osher-Rudin. As we have pointed out above, in dimension 1 at least, TVDN is a special case of BPDN, obtained by using the Heaviside dictionary. In dimension > 1, the connection of the two ideas is not so close; and so it is unclear that BPDN in a two-d case would connect directly with the Li-Santosa work on TVDN. At the algorithmic level it is particularly helpful in Li and Santosa's work to have a single specific type of matrix $A$ to deal with in their linear program, which affords a great deal of potential for special accelerations.

$\ell^1$ Methods in Signal Processing. There is a long tradition of applying minimum $\ell^1$ methods in signal processing, for reasons of achieving sparsity and super-resolution (as a pointer to this literature, we suggest the many references in [23, 24, 27]). In some sense, the examples we have given in this paper simply make explicit ideas implicit in much work in the applied optics, MR spectroscopy, and seismology literature. Nevertheless, we believe
that framing these ideas as we do renders a service to the overcomplete representation community that may help focus their attention and interests. Much of the old knowledge about minimum $l^1$ methods was achieved before interior point methods came on the scene, and so such methods may have seemed relatively impractical at the time.

$L^1$ Methods in Statistics. We are happy to point to the work of our colleagues in Statistics who have done a great deal of important work on the use of penalized $l^1$ methods. A methodological link is to Robert Tibshirani's work [62] with the Lasso – which is precisely the same thing as BPDN when scaling is handled in the right way. Tibshirani envisioned applications more in line with statistical model fitting (undercomplete dictionaries, smaller-scale problems); in particular the viewpoint is somewhat different from that dictated by large scale applications. A theoretical link is to the work of Sara van de Geer and Enno Mammen, who have studied the accuracy of Total-Variation De-Noising; [31]. It is also worth pointing to work by Donoho and Johnstone on $l^1$ methods in NMR signal processing, such as [27].

We hope that interested readers will be prompted by this article to consult some of the earlier related work, and that they will perhaps call our attention to other related work which we do not reference here.

References


