Probabilistic Compressed Sensing

BY ROBERT CRANDALL

Program in Applied Mathematics

Abstract

We investigate applications of Minimum Mean Squared Error estimation to compressed sensing and sparse approximation. We will focus on a recent paper by Elad & Yavneh in the denoising literature, and discuss applications of this work in the compressed sensing framework. We propose an novel method that is significantly faster than the one proposed in the paper, but exhibits comparable performance. We discuss directions for future research.

1 Introduction: Compressed Sensing

1.1 Classical Sampling Theory

In classical sampling theory, signals are classified in terms of their frequency content. A signal whose Fourier transform is compactly supported is said to be band-limited. The well known Nyquist-Shannon sampling theorem states that any bandlimited signal can be recovered exactly from samples taken at the Nyquist rate $2B$, where B (the bandwidth) is the largest frequency in the signal. A reconstruction is obtained through the Whittaker-Shannon interpolation formula:

$$x(t) = \sum_{n=-\infty}^{\infty} x(n\,T)\,\mathrm{sinc}\left(\frac{t-n\,T}{T}\right)$$

That is, the original signal is recovered by convolving the sampled version with a sinc function. If sampling is performed at below the Nyquist rate, information is lost to aliasing and a perfect reconstruction cannot be guaranteed. Most modern signal processing relies on this assumption, but much recent attention has been given to other signal models that potentially allow for much lower sampling rates.

1.2 A New Direction: Sparse Approximations

An arbitrary signal must be fully sampled in order to assure perfect reconstruction. By restricting the class of signals we can reduce the number of necessary samples, as is clear from the Nyquist-Shannon theorem; in this case a priori knowledge of a signal's frequency content allows us to reconstruct it from a smaller number of samples. What if we impose different restrictions on our signals of interest? Much attention has been given recently to the class of *sparse* signals; a vector $x \in \mathbb{R}^n$ is said to be sparse if it has few nonzero entries:

$$\|x\|_0 = \#\{i : x_i \neq 0\} \ll n.$$

$\|\cdot\|_0$ denotes the "zero norm" (a notational convenience, since this is not a norm at all, but a limit of the pseudo-norms $\|\cdot\|_p$ as $p \to 0$) which counts the number of nonzero entries in a signal.

Sparse reconstruction deals fundamentally with the underdetermined linear system

$$y = \Phi x$$

where $\Phi \in \mathbb{R}^{m \times n}$ and $\|x\|_0$ is small. Assume for simplicity that Φ is full-rank; then we know that this system has a solution. However, it has infinitely many solutions; the nullspace of Φ is nontrivial, so for a given solution v satisfying $y = \Phi\, v$, any vector in $v + \mathrm{null}(\Phi)$ is also a solution. We must now decide how to select a "good" solution from this affine space.

1.2.1 Regularization Methods

A standard way to restrict the solution set of such an underdetermined linear system is to perform a *regularization* of the problem [6]. We introduce some cost function $J(x)$ that evaluates the quality of a potential solution; reconstruction is then performed by solving the optimization problem

$$\min_x J(x), \;\; s.t. \;\; y = \Phi\, x.$$

For example, the least-squares problem seeks to minimize the standard Euclidean norm of the solution: $J(x) = \|x\|^2$. In this case the solution is easily found using the pseudoinverse:

$$x_{l.s.} = \Phi^*(\Phi^*\Phi)^{-1}y = \Phi^+ y.$$

Since we have assumed Φ is full rank, $\Phi^*\Phi$ is indeed invertible and we obtain a unique solution. More generally, ANY convex cost function J will lead to a unique solution. Unfortunately, when the true signal x is sparse the least squares reconstruction is often very poor. In Figure 1 a simple image with an approximately sparse wavelet transform is measured using Fourier measurements with 50% undersampling, then reconstructed using least-squares (b) and a compressed sensing method (c). The compressed sensing reconstruction method used is Iterative Hard Thresholding, to be discussed later; it takes advantage of the inherent sparsity of the image to produce a much better reconstruction from the undersampled data.

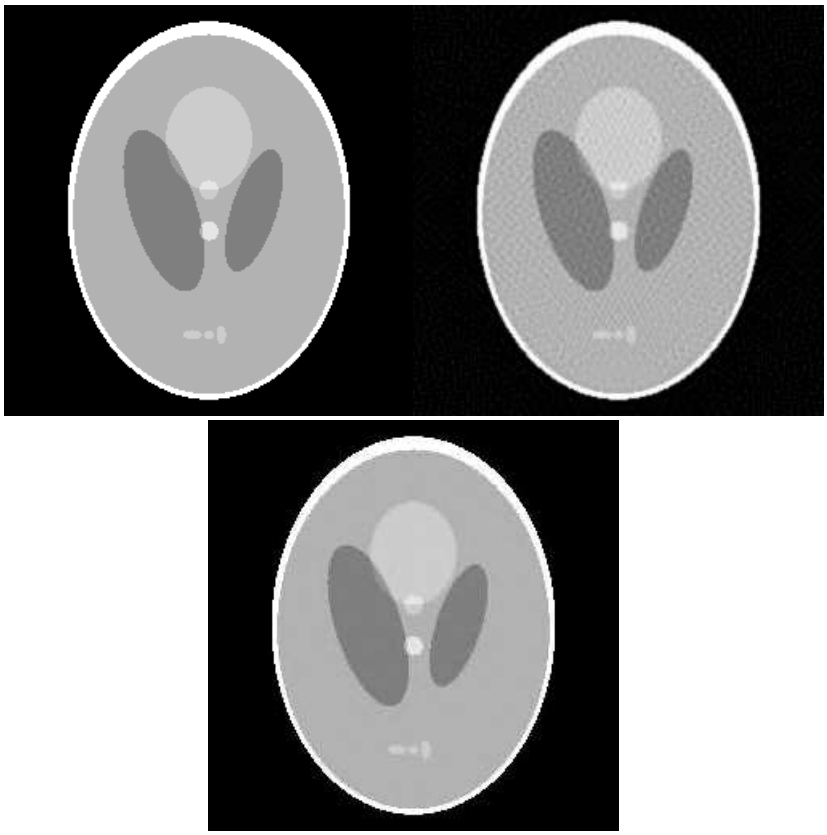

Figure 1. (a) original image, (b) image reconstruction from least squares, and (c) image reconstruction using CS

Since an l^2 cost function performs poorly, we turn next to the l^1 norm, $J(x) = \|x\|_1$. This turns the optimization into a linear programming task, for which there is an abundance of solution methods. Furthermore, l^1 minimization is known to promote sparsity; a standard result in linear programming is that there is always an l^1-minimizing solution with at most m nonzero coefficients. To further promote sparsity, we can examine the l^p pseudonorms with $p < 1$: $J(x) = \|x\|_p^p = \sum |x_i|^p$. A lower value for p will lead to increased sparsity. The cost function in this case is no longer convex (as it is for $p \geqslant 1$), so finding a solution becomes more difficult.

Figure 2. l^p balls for $p = \frac{1}{2}, 1, 2$. Smaller values for p lead to increased sparsity

Finally, the "ideal" cost function to use for sparse reconstruction is $J(x) = \|x\|_0$; we seek the sparsest solution that is consistent with the measurement y. Solving this problem directly is intractible for most applications, since it requires a combinatorial search of all possible sparse solutions. For instance, even for a very small problem with $\Phi \in \mathbb{R}^{128\times 256}$ with an assumed sparsity level $s = 10$, we must search all $\left(\begin{smallmatrix} 256 \\ 10 \end{smallmatrix}\right) \approx 2.8 \times 10^{17}$ possible supports. The development of practical algorithms for seeking sparse solution will be the focus of the latter part of this paper.

1.2.2 The Noisy Case

We note here that in practice we will almost never work with an exactly linear system $y = A\,x$. First, real world signals are rarely truly sparse, but rather compressible (they are well approximated in l^2 by a sparse signal). Second, real measurements always contain noise and errors, so we are in fact working with the affine system

$$y = A\,x + v.$$

Thus, in applications we don't look for a reconstruction that is exactly consistent with the measurement y, since this would be hopeless; instead we relax the optimization problem to allow for error. For example we might seek

$$\min_x J(x) \;\; s.t. \|y - A\,x\|_2^2 \leqslant \epsilon,$$

where ϵ is an error threshold selected based on SNR assumptions for a particular problem.

1.2.3 Compressed Sensing

A particular type of sparse reconstruction problem that has gained great popularity in the last decade is compressed sensing. The theory of compressed sensing, formally introduced by two seminal papers by Candes, Romberg and Tao [1] and Donoho [2], lays down a new sampling framework for sparse or compressible signals, allowing for sampling rates that are potentially much lower than those required by Nyquist-Shannon. It focuses on (1) the design of measurement systems that take advantage of signal sparsity to reduce sampling rates and (2) the development of algorithms for reconstruction of signals from these undersampled measurements.

Fortunately, the assumption of sparsity is a good one in many applications. For example, most natural images are well approximated by sparse signals, which is why JPEG and other lossy image compression algorithms are so widely used. The basic idea behind JPEG compression is to decompose an image using a sparsifying linear transform, and encode only the significant (large) coefficients: for example an image f is often nearly sparse under a discrete cosine transform A, so we might take $y = A\,f$ and set all of the small coefficients of y to zero, obtaining a compressed version of the image $\hat{y}$. We can then recover the image by taking $\hat{f} = A^{-1}\hat{y}$.

Experience has shown that most images are compressed very well by JPEG; we can throw out nearly all of the coefficients in a typical image and still obtain a reconstruction that is recognizable to the human eye. It is not uncommon for a megapixel image to be compressed to around 100 kilobytes with minimal noticeable loss in quality. This means that a typical camera is acquiring significantly more information than is necessary; all this "extra" information is simply discarded in the compression process. Is there a way to avoid obtaining all this extra information in the first place? This is the heart of compressed sensing: using a priori knowledge of signal sparsity, we can design a measurement framework that acquires a signal directly in a compressed form, which can then reconstructed in post-processing.

1.3 The Compressed Sensing Framework

We are interested in signals that are sparse or compressible in a certain basis Ψ; that is, signals of interest satisfy $x = \Psi\alpha$ (sparse) or $x = \Psi\alpha + \epsilon$ (compressible) where α is a sparse representation of x. In the example of JPEG images above, Ψ is a discrete cosine transform. The choice of basis will depend on the application and the class of signals we wish to measure.

The measurement process used in compressed sensing is also described by a linear transform. Given a signal $x \in \mathbb{R}^n$ and a "measurement matrix" $\Phi \in \mathbb{R}^{m\times n}$, we obtain a measurement by correlating the signal with vectors in the measurement basis; in matrix form,

$$y = \Phi x + v = \Phi\Psi\alpha + v$$

where v is a measurement noise or error term. In compressed sensing $m < n$, so we are *undersampling* the signal; m/n provides a measure of the compression ratio. To reconstruct x we must then solve an underdetermined linear (or affine if $v \neq 0$) system. The assumption of signal sparsity will be used to seek a "good" solution, as discussed above for the general sparse linear reconstruction problem. For a carefully chosen sparsity basis Ψ and measurement matrix Φ, it is possible to have $m \ll n$.

To illustrate this we give a quick example. First, a definition:

Definition 1. *Spark of a Matrix*
The spark *of a matrix A,* $\text{spark}(A)$*, is the smallest number of columns in A that can be combined to form a linearly dependent set.*

Suppose $\|x\|_0 \leqslant s$, and let $A \in \mathbb{R}^{m\times n}$ be a matrix with $\text{spark}(A) > 2s$; then in particular any $2\,s$ columns of A are linearly independent. It is easily seen that there is only one s-sparse solution to the linear system $y = A\,x$. To see this, let x_1, x_2 be two s-sparse vectors satisfying $y = A\,x_1 = A\,x_2$. Then $A\,(x_1 - x_2) = 0$. Clearly $x_1 - x_2$ has no more than $2\,s$ nonzero entries. Since the columns of A corresponding to the support of $x_1 - x_2$ are linearly independent by assumption, $A(x_1 - x_2) = 0 \Longrightarrow x_1 = x_2$.

Example 2. Suppose Φ is the standard 10×10 discrete Fourier matrix. Let $\Phi_S \in \mathbb{C}^{4\times 10}$ be formed by choosing any 4 columns of Φ. Any 4 columns of Φ_S are linearly independent; it follows that any 2-sparse vector $x \in \mathbb{C}^{10}$ can be reconstructed exactly from the four Fourier measurements $y = \Phi_S\,x$.

It is clear from this example why the term *compressed sensing* is used; we have shown that, assuming we have hardware that will take Fourier measurements, we can acquire a signal *directly* in a compressed form; we only need to make 4 measurements, and the extra 6 entries never need to be computed or stored! Of course, we have only shown that the reconstruction is possible in the sense that there is a unique solution for this toy example. We have not given a method for acquiring such undersampled measurements, or for reconstructing the original signal. These two problems, measurement system design and signal reconstruction, form the core of compressed sensing research. We give a brief description of the measurement problem below, and then turn our attention to the reconstruction problem which is the focus of this paper.

1.3.1 Measurement System Design

The measurements used for compressed sensing must be carefully designed to exploit signal sparsity. There are two important aspects in the design of a measurement system; first, we need to find a good sparsifying basis that will allow us to work with the desired class of signals (that is, we must find a basis Ψ under which the signals of interest are sparse or compressible). Second, we must design a measurement basis that is compatible with the sparsifying basis and, for any real application, practical to implement in hardware. To define what we mean by "compatible" measurement and sparsity bases, we introduce the concept of coherence.

Definition 3. *Coherence*

Let $\Phi, \Psi \in \mathbb{R}^{n\times n}$ be orthormal matrices (this is not necessary but simplifies the definition). Then the coherence *$\mu(\Phi, \Psi)$ between these matrices is defined as*

$$\mu(\Phi, \Psi) = \sqrt{n}\max_{i,j} |\langle \phi_i, \psi_j \rangle|.$$

The coherence falls in the range $\mu \in [1, \sqrt{n}]$.

This is a measurement of the maximum correlation between vectors in each basis. If μ is small we say the bases are incoherent.

In order for compressed sensing to be practical, the measurement and sparsity bases must be largely incoherent. In essence this means that each measurement provides a sufficient mixture of elements in the signal's sparse representation so that we do not need to take all n measurements to reconstruct the signal. Consider first the case where Φ and Ψ are both the identity. This is a worst-case scenario with maximal coherence. Each measurement gives us information about only one coefficient of the sparse signal; clearly in this case we will need to take all n measurements to guarantee perfect reconstruction and compressed sensing cannot be applied. On the other extreme take the case where Ψ is the identity but Φ is a discrete Fourier matrix. This is the best case; the coherence is 1 so the bases are maximally incoherent (this is a well known property of the time and frequency domains; the definition of incoherence extends this idea to other basis pairs). Each Fourier measurement provides a mixture of information from each sparse coefficient, and we can reconstruct signals from fewer than the full n measurements (see the example above; a 2-sparse signal in $\mathbb{R}^{10}$ can be reconstructed from only four frequency-domain measurements). In many applications we cannot obtain perfect incoherence, often because the measurement basis is fixed; for example, in MRI measurements are obtained in the Fourier domain by design. Various wavelets are often used as sparsity bases for this application; these are only moderately incoherent with the Fourier basis.

1.3.2 Signal Reconstruction

The second crucial step in compressed sensing, and the focus of this paper, is signal reconstruction. Supposing that we have an adquate measurement system, we need a way to recover the signal in question from the (possibly noisy) measurement

$$y = \Phi x + v.$$

For simplicity here we have combined the sparsity matrix and measurement matrix into a single matrix Φ, and x is the sparse representation of the signal.

As discussed above, we will perform some regularization of this system; we select a regularizing function $J(x)$ that promotes sparsity, and give some constraint on the maximum allowable reconstruction error $\|y - \Phi x\|$. There are several ways to set up the problem; for example, we can enforce sparsity strictly and minimize error,

$$\min_x \|y - \Phi x\|_2 \;\; s.t. \; \|x\|_0 \leqslant s$$

we can constrain error and seek to minimize the regularizing function,

$$\min_x \; J(x) \, s.t. \, \|y - \Phi x\|_2^2 \leqslant \epsilon$$

or we can seek to minimize some weighted combination of the two terms:

$$\min_x \left[J(x) + \lambda \|y - \Phi x\|_2^2 \right]$$

Different choices will lead to different algorithms; we present a few examples in the next section.

1.4 Algorithms for Signal Reconstruction

The first method used for reconstruction in the compressed sensing literature is l^1 minimization (e.g. [1] deals with l^1 reconstruction of sparse signals from frequency measurements). That is, we take $J(x) = \|x\|_1$ and solve the problem using linear programming techniques. However, these techniques are often prohibitively slow for large problems. We will focus on two classes of algorithms: greedy pursuit algorithms, and iterative methods.

1.4.1 Orthogonal Matching Pursuit

Orthogonal matching pursuit (or OMP) is a basis-pursuit algorithm for greedily solving the compressed sensing optimization problem. Starting from the initial zero solution, we increase the sparsity by one at each step by incrementally adding columns of Φ to the support set until a stopping criterion is reached. The coefficient added at step k is determined by computing the correlations $\langle \phi_i,\ r^{k-1} \rangle$ between the columns of Φ and the residual $r^{k-1} = y - \Phi\alpha^{k-1}$; the column with maximum correlation is added to the support, since this is the one that best reduces the residual error

$$y - \Phi\,\hat{\alpha}^k.$$

After updating the support to $S^k = S^{k-1} \cup \phi_i$ the current solution $\hat{\alpha}^k$ is easily computed using least squares (restricted to the given support). The stopping criterion will be that either a maximum allowed sparsity is reached, or the residual error $\|y - \Phi\hat{\alpha}^k\|$ falls below some threshold ϵ chosen based on the assumed SNR.

1.4.2 Shrinkage/ Thresholding Algorithms

Greedy pursuit algorithms, like linear programming methods, can be slow for large problems. A faster class of methods known collectively as iterative shrinkage algorithms and iterative thresholding algorithms offer a fast method for obtaining sparse approximations. For an overview of such methods, see for example [5] and [7].

Each of these methods relies on a similar iteration. Given an approximation $\hat{\alpha}^k$ at step k, we perform a back-projection of the residual $y - \Phi\hat{\alpha}^k$ and add it to the current approximation:

$$\hat{\alpha}^k + \Phi^*(y - \Phi\hat{\alpha}^k).$$

We will refer to this step as a *Landweber iteration* [8]. Depending on the specific algorithm, we then apply a shrinkage or thresholding operator to this Landweber iterate.

We will focus on a variant which seeks to solve

$$\min_{\alpha} \|y - \Phi\alpha\|_2^2 \ s.t. \|\alpha\|_0 \leqslant s;$$

that is, we strictly enforce sparsity below a given level s. Using an optimization transfer technique (see [8]), we derive the Iterative Hard Thresholding (IHT) algorithm:

$$\hat{\alpha}^{k+1} = H_s(\hat{\alpha}^k + \Phi^*(y - \Phi\hat{\alpha}^k)).$$

Here H_s is the *hard thresholding* operator that sets all but the largest s elements of its argument to zero, thus enforcing sparsity at each step. The effectiveness of this algorithm for compressed sensing problems is demonstrated in [9]. A modification of this algorithm will serve as the focus of the remainder of this paper.

2 Probabilistic Algorithms

So far we have outlined the compressed sensing framework and given two simple algorithms for reconstruction, namely OMP and IHT. These are deterministic algorithms in that they always produce the same output given a fixed input. These algorithms rely only on the assumption that the signal in question is sparse, as well as some assumptions on the measurement and sparsity bases to ensure convergence and uniqueness of solutions. If we introduce further information about the signal space, we should hopefully be able to create more accurate algorithms. Specifically, if we have a reasonable probabilistic model describing the signals of interest, we can exploit some Bayesian estimation techniques to improve on algorithms like OMP and IHT.

2.1 Estimation Theory: Oracle, MAP and MMSE

Two well known tools in estimation theory are maximum a-posteriori probability (MAP) estimation and minimum mean-squared error (MMSE) estimation. In the context of sparse reconstruction, a MAP estimator seeks to find the most likely sparse approximation α given the measurement y. That is,

$$\hat{\alpha}^{\text{MAP}} = \max_{\alpha} p(\alpha|y).$$

This conditional probability is a function of the assumed signal distribution $p(\alpha)$ and the noise distribution $p(v)$, where $y = \Phi\alpha + v$. The MMSE estimator, on the other hand, minimizes the expected mean-squared error

$$E(\|\hat{\alpha} - \alpha\|^2),$$

and is just the expected value of α conditioned on the given measurement:

$$\hat{\alpha}^{\text{MMSE}} = E(\alpha|y).$$

Clearly, each of these estimators is highly dependent on the given priors $p(\alpha)$ and $p(v)$.

We can also consider the "oracle" estimator, which is an idealized estimator in which we know a priori the location of the nonzero coefficients of α; it is given by $E(\alpha|S_{\text{true}}, y)$ where S_{true} denotes the support set of α.

It is illustrative to derive the form of these estimators for the compressed sensing problem, which is done in the next section.

2.2 The MMSE Estimator

We now derive the MMSE estimator for α which is given in [theorem below]. This derivation follows [4] closely, except here we are minimizing the error on the sparse representation α rather than the measurement domain signal $\Phi\alpha$.

NOTE: I think the following can and should be made more concise

Theorem 4. *The MMSE estimator for α is given by*

$$\boxed{\hat{\alpha}^{\text{MMSE}} = \frac{1}{\sum_{S\in\Omega} p(y|S)\,p(S)} \sum_{S\in\Omega} \left[\int_{\mathbb{R}^n} \alpha \, \frac{p(y|\alpha)\,p(\alpha|S)}{p(y|S)} \, d\,\alpha \right] p(y|S)\,p(S)}$$

Lemma 5. *We will use this basic probability result several times:*

$$\int p(x|y,z)p(y|z)\,d\,y = \int \frac{p(x,y,z)}{p(y,z)} \frac{p(y,z)}{p(z)}\,d\,y = \frac{p(x,z)}{p(z)} = p(x|z)$$

From the lemma we have

$$p(\alpha|y) = \sum_{S\in\Omega} p(\alpha|S,y)\,P(S|y).$$

Using this we rewrite the conditional MSE as

$$\begin{aligned}\mathrm{MSE}_y &= \int_{\mathbb{R}^n} \|\alpha - \hat{\alpha}\|^2 \left(\sum_{S\in\Omega} p(\alpha|S,y)\, p(S|y) \right) d\,\alpha \\ &= \sum_{S\in\Omega} \left[\int_{\mathbb{R}^n} \|\alpha - \hat{\alpha}\|^2 p(\alpha|S,y)\, d\,\alpha \right] P(S|y) \\ &= \sum_{S\in\Omega} \mathrm{MSE}_{S,y} P(S|y)\end{aligned}$$

where $\mathrm{MSE}_{S,y}$ denotes the MSE conditioned over both the measurement y and a particular support S. This conditional error can be expanded as

$$\begin{aligned}\mathrm{MSE}_{S,y} &= E(\|\hat{\alpha} - \alpha\|^2|S,y) \\ &= E(\|\hat{\alpha}\|^2 - 2\langle\alpha,\hat{\alpha}\rangle + \|\alpha\|^2|S,y\big) \\ &= \|\hat{\alpha}\|^2 - 2\langle\hat{\alpha}, E(\alpha|S,y)\rangle + E(\|\alpha\|^2|S,y\big).\end{aligned}$$

Expanding the last term in the sum gives

$$\begin{aligned}E(\|\alpha\|^2|S,y) &= E(\|E(\alpha|S,y) + \alpha - E(\alpha|S,y)\|^2|S,y) \\ &= \|E(\alpha|S,y)\|^2 + E(\|\alpha - E(\alpha|S,y)\|^2|S,y) \\ &= \|E(\alpha|S,y)\|^2 + V_{S,y}(\alpha),\end{aligned}$$

where $V_{S,y}$ denotes the conditional variance of α. Thus, we have

$$\begin{aligned}\mathrm{MSE}_{S,y}(\alpha) &= \|\hat{\alpha}\|^2 - 2\langle\hat{\alpha}, E(\alpha|S,y)\rangle + \|E(\alpha|S,y)\|^2 + V_{S,y}(\alpha) \\ &= \|\hat{\alpha} - E(\alpha|S,y)\|^2 + V_{S,y}(\alpha).\end{aligned}$$

To compute MSE_y we now need to compute $P(S|y)$.

The distribution of y conditioned on a given sparse support S is

$$p(y|S) = \int_{\mathbb{R}^n} p(y|\alpha,S)\, p(\alpha|S)\, d\,\alpha = \int_{\mathbb{R}^n} p(y|\alpha)\, p(\alpha|S)\, d\,\alpha,$$

where $p(y|\alpha,S) = p(y|\alpha)$ since S is determined once α is known. The measurement matrix Φ is assumed to be known, so $p(y|\alpha)$ is determined entirely by the noise distribution; in particular

$$p(y|\alpha) = p_v(y - \Phi\alpha).$$

Using Bayes' formula, we rewrite $P(S|y)$ as

$$\begin{aligned}P(S|y) &= \frac{p(y|S)P(S)}{p(y)} \\ &= \frac{1}{p(y)} \int_{\mathbb{R}^n} p(y|\alpha)p(\alpha|S)\, P(S)\, d\,\alpha.\end{aligned}$$

Rather than compute $p(y)$ directly we simply use the normalization requirement

$$\sum_{S\in\Omega} P(S|y) = 1$$

to find

$$p(y) = \sum_{S\in\Omega} p(y|S)\, P(S).$$

Thus,

$$P(S|y) = \frac{p(y|S)P(S)}{\sum_{S'\in\Omega} p(y|S')P(S')},$$

and the conditional MSE is given by

$$\begin{aligned}\mathrm{MSE}_y &= \sum_{S\in\Omega} \mathrm{MSE}_{S,y}\, P(S|y) \\ &= \sum_{S\in\Omega} \|\hat{\alpha} - E(\alpha|S,y)\|^2 P(S|y) + \sum_{S\in\Omega} V_{S,y}(\alpha) P(S|y) \\ &= E(\|\hat{\alpha} - M_{S,y}(\alpha)\|^2|y) + E(V_{S,y}(\alpha)|y).\end{aligned}$$

The second term is independent of the estimator and depends only on the noise variance. The MMSE, then, is found by minimizing the first term:

$$\begin{aligned}\hat{\alpha}^{\mathrm{MMSE}} &= \arg\min_{\hat{\alpha}} E(\|\hat{\alpha} - M_{S,y}(\alpha)\|^2|y) \\ &= \boxed{\sum E(\alpha|S,y)P(S|y)} \\ &= \frac{1}{\sum_{S\in\Omega} p(y|S)\,P(S)} \left\{ \sum_{S\in\Omega} \left[\int_{\mathbb{R}^n} \alpha \frac{p(y|\alpha)p(\alpha|S)}{p(y|S)}\, d\,\alpha \right] p(y|S)P(S) \right\}.\end{aligned}$$

Note that the term in square brackets $\int \alpha \frac{p(y|\alpha)p(\alpha|S)}{p(y|S)}\, d\,\alpha$ is exactly the oracle estimate $E(\alpha|S, y)$ obtained if we assume S is the correct support. The MMSE can be viewed as a weighted sum of the oracle estimators on each possible support, with weight $P(S|y)$; that is, each potential solution is weighted by its probability of correctly explaining the observed measurement.

2.3 The MAP Estimator

The MAP estimator is simply the oracle estimate on the most likely support, or

$$\hat{\alpha}^{\mathrm{MAP}} = E(\alpha|S_{\mathrm{MAP}}, y)$$

where

$$S_{\mathrm{MAP}} = \arg\max_{S} P(S|y).$$

Like the MMSE, the MAP estimate requires a computation of $P(S|y)$ over every possible support; however, $E(\alpha|S, y)$ need only be computed for S_{MAP}.

NOTE: Elad's book seems to indicate that the MAP obtained by maximizing $P(S|y)$ is not necessarily equal to that obtained by maximizing $p(\alpha|y)$... *"An alternative way to define the MAP estimation is by maximizing the posterior $P(S|y)$ ... In the case considered here, this approach leads to a very similar solution... In more complex scenarios, though, this approach may lead to an entirely different and more stable result"* pg 211. Look into this

2.4 A Particular Case: Gaussian Priors

2.4.1 Signal Modeling

2.4.2 MMSE Derivation

For illustrative purposes let us compute the MMSE with the assumption of Gaussian priors. We will assume that the true signal has a given known sparsity k, and that any k-sparse support is equally likely; that is,

$$P(S) = \begin{cases} \dfrac{1}{|\Omega_k|} \text{ if } S \in \Omega_k \\ 0 \text{ otherwise} \end{cases}.$$

The noise term v is drawn from an i.i.d. Gaussian with mean zero and variance σ_v^2:

$$p_v(v) = \frac{1}{(2\pi\sigma_v^2)^{m/2}} \exp\Big(-\frac{\|v\|^2}{2\sigma_v^2} \Big).$$

If α is known, the measurement y depends only on v, so we have

$$p(y|\alpha) = p_v(v) = p_v(y - \Phi\,\alpha) = \frac{1}{(2\pi\sigma_v^2)^{m/2}} \exp\Big(-\frac{\|y - \Phi\alpha\|^2}{2\sigma_v^2} \Big).$$

Once a support S has been chosen according to $P(S)$, the nonzero entries of α are chosen from an i.i.d. Gaussian with mean zero and variance σ_α^2:

$$p(\alpha|S) = \frac{1}{(2\pi\sigma_\alpha^2)^{k/2}} \exp\Big(-\frac{\|\alpha\|^2}{2\sigma_\alpha^2} \Big)$$

For simplicity suppose that α is sparse in the standard basis, or $\Psi = I$. Equivalently, we can let Φ represent the combined sparsity-measurement operator. Then we have

$$y = \Phi\,\alpha + v.$$

We will first need to compute $P(S|y)$. From Bayes rule, this is

$$P(S|y) = \frac{p(y|S)\,P(S)}{p(y)}.$$

$P(S)$ is given above. We will not need to compute $p(y)$ directly, as it can easily be determined using the normalization requirement $\sum P(S|y) = 1$. That leaves $p(y|S)$, which is as follows (the domain of integration S is the k-dimensional subspace of $\mathbb{R}^n$ corresponding to the given support S)

$$p(y|S) = \int_S p(y|a)p(\alpha|S)\,d\,\alpha$$

$$= \frac{1}{(2\pi\sigma_v^2)^{m/2}\cdot(2\pi\sigma_\alpha)^{k/2}} \int_S \exp\Big(-\frac{\|y - \Phi\alpha\|^2}{2\sigma_v^2} \Big)\exp\Big(-\frac{\|\alpha\|^2}{2\sigma_\alpha^2} \Big)\,d\,\alpha$$

$$= \frac{\exp\Big(-\frac{\|y\|^2}{2\sigma^2} \Big)\exp\Big(\frac{\langle z_S, (\sigma_\alpha^2\Phi_S^*\Phi_S + \sigma_v^2 I) z_S\rangle}{2\sigma_v^2\sigma_\alpha^2} \Big)}{(2\pi\sigma_v^2)^{m/2}\cdot(\sigma_\alpha^2)^{k/2}} \sqrt{\det\Big(\Big[\frac{\sigma_\alpha^2\Phi_S^*\Phi_S + \sigma_v^2 I}{\sigma_v^2\sigma_\alpha^2} \Big]^{-1} \Big)}$$

where

$$z_S = E(\alpha|S, y) = \sigma_\alpha^2(\sigma_\alpha^2\Phi_S^*\Phi_S + \sigma_v^2 I)^{-1}\Phi_S^* y_S.$$

Note that the operator $\sigma_\alpha^2(\sigma_\alpha^2\Phi_S^*\Phi_S + \sigma_v^2 I)^{-1}\Phi_S^*$ is essentially a modified pseudoinverse that accounts for the probability priors. This gives

$$P(S|y) \propto \sqrt{\det\Big(\Big[\frac{\sigma_\alpha^2\Phi_S^*\Phi_S + \sigma_v^2 I}{\sigma_v^2\sigma_\alpha^2} \Big]^{-1} \Big)} \cdot \exp\Big(\frac{\big\langle z_S, \big(\sigma_\alpha^2\Phi_S^*\Phi_S + \sigma_v^2 I\big) z_S \big\rangle}{2\sigma_v^2\sigma_\alpha^2} \Big),$$

The constant of proportionality is determined by a normalization requirement; it is simply

$$\left(\sum_{S\in\Omega_k} \sqrt{\det\Big(\Big[\frac{\sigma_\alpha^2\Phi_S^*\Phi_S + \sigma_v^2 I}{\sigma_v^2\sigma_\alpha^2} \Big]^{-1} \Big)} \cdot \exp\Big(\frac{\big\langle z_S, \big(\sigma_\alpha^2\Phi_S^*\Phi_S + \sigma_v^2 I\big) z_S \big\rangle}{2\sigma_v^2\sigma_\alpha^2} \Big) \right)^{-1}.$$

The final expression for the optimal mean-squared-error estimator is

$$\hat{\alpha}^{\mathrm{MMSE}} = \sum_{S\in\Omega_k} P(S|y)\cdot z_S.$$

Thus, the MMSE solution is a weighted sum of solutions over every possible support as claimed.

Note that we have still not obtained a practical algorithm, since this expression requires computing z_S for each of the $\binom{n}{k}$ supports in Ω_k which will quickly become intractable even at moderate problem sizes. To generate a practical algorithm, we will attempt to sample from the set $\{z_S : S \in \Omega_k\}$, and hope that most of the energy of the MMSE estimate is concentrated in a small number of the most likely supports.

2.5 RandOMP: An Approximate MMSE Estimator

In [4] Elad and Yavneh introduce a novel method for sampling from the set of possible supports S in such a way that a useable approximation of the MMSE estimator is obtained. Start by considering the 1-sparse case ($s = 1$) with Gaussian priors. This reduces the number of possible supports to n; we simply sweep over each column of Φ, and the expression for $\hat{\alpha}^{\text{MMSE}}$ is greatly simplified. First, $\Phi_S = \phi_i$ where ϕ_i is a column of Φ and $S = \{i\}$; the oracle estimate on S is

$$E(\alpha|S, y) := z = \sigma_\alpha^2(\sigma_\alpha^2 \Phi_S^* \Phi_S + \sigma_v^2 I)^{-1} \Phi_S^* y_S$$

$$= \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_v^2}\, \phi_i^* y.$$

The conditional probability of a given support is

$$P(S|y) \propto \exp\left(\frac{1}{2\sigma_v^2} \cdot \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_v^2} \cdot (\phi_i^* y)^2 \right).$$

Recall now the OMP algorithm introduced above. At each step of this algorithm, we choose the next atom based on which of the remaining columns ϕ_i is most correlated with the residual r^k; that is, we maximize $|\phi_i^* r^k|$. RandOMP instead selects the next atom randomly, with probability

$$P_i \propto \exp\left(\frac{1}{2\sigma_v^2} \cdot \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_v^2} \cdot \left(\phi_i^* r^k \right)^2 \right).$$

We then run OMP several times, obtaining distinct solutions thanks to the probabilistic support selection. At the end we average the obtained solutions together to obtain the RandOMP result. For the 1-sparse case, this gives exactly the weight term in the MMSE sum, so as the number of representations averaged goes to inifinity, RandOMP approaches the MMSE estimate. It turns out that this method continues to work well even as s increases, even though we are no longer precisely approaching MMSE.

The simplifying assumption $s = 1$ that allows us to treat each column ϕ_i individually (rather than working with all of the subdictionaries Φ_S) is the key step here in creating a workable algorithm, and it is not precisely clear why this assumption appears to work so well in practice.

Figure 3. Reconstruction error vs.

Figure 4. Reconstruction error vs. SNR for OMP and RandOMP

2.6 A Novel Algorithm: Randomized IHT

We now present a novel algorithm that applies the same idea used in RandOMP to IHT. In order to sample from the set of possible supports in this case, we will replace the hard thresholding operator H_s with a randomized version $H_{s,P}$. P is a discrete probability distribution on $\{1, \ldots, n\}$ that gives the probability that a given coefficient will appear in the support of $H_{s,P}(x)$. To be precise, the computation of $H_{s,P}(x)$ proceeds as follows:

Algorithm

Computing $H_{s,P}(x)$

We start with an empty support set $S = \{\}$.

1. Select an element $i \in \{1, \ldots, n\} \backslash S$ with probability P_i
2. Increment the support $S \to S \cup \{i\}$
3. Renormalize the probabilities for the set $\{1, \ldots, n\} \backslash S$ of unused elements
4. Repeat until $|S| = s$
5. Set $\left(H_{s,P}(x)\right)_i = \begin{cases} x_i & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$

For the case of Gaussian priors, we would set $P_i \propto \exp\left(\frac{1}{2\sigma_v^2} \cdot \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_v^2} \cdot \left[\alpha_i^k\right]^2\right)$ [NOTE: needs a little more clarification here on why we plug in α_i]. We then generate multiple solutions using IHT with this randomized thresholding operator, and average these solutions together, hopefully approaching the MMSE estimate. To summarize, the randomized IHT algorithm works as follows:

1. Generate N solution candidates using the randomized iterative method
$$\hat{\alpha}_i^{k+1} = H_{s,P}(\hat{\alpha}_i^k + \Phi^*(y - \Phi\hat{\alpha}_i^k))$$
2. Combine the N solutions $\hat{\alpha}_i$ by averaging to form the final solution:
$$\hat{\alpha} = \frac{1}{N}\sum_{i=1}^{N} \hat{\alpha}_i.$$

It is not immediately obvious whether this algorithm will even converge, since we are potentially changing the support at each step of the iteration (contrast with RandOMP, where elements in the support are fixed once chosen). However, empirical results are promising; this algorithm gives similar performance improvements over IHT as we saw for OMP vs RandOMP.

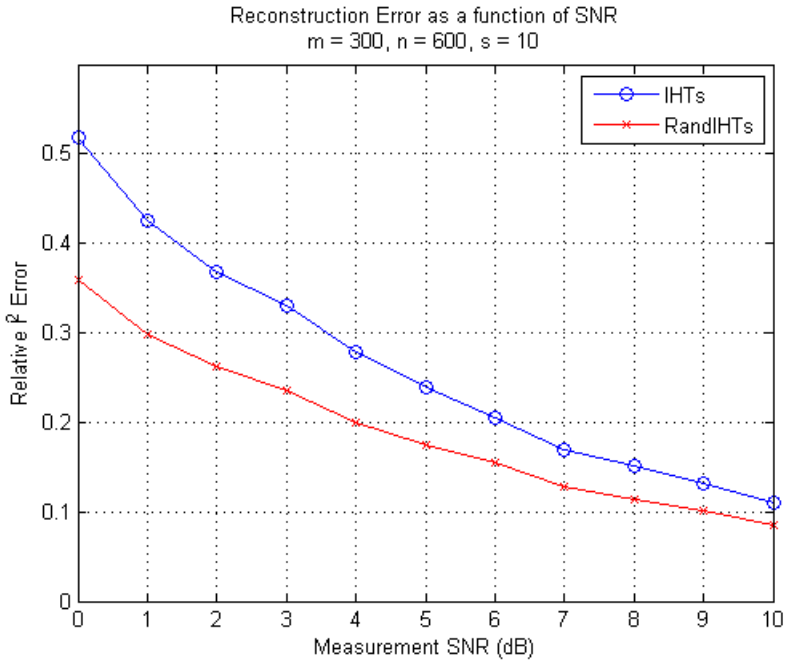

Figure 5. Reconstruction error vs. SNR for IHT and randomized version

Despite the random support selection at each step, the algorithm appears to converge approximately to a steady solution. The solution should oscillate less at higher SNR; note that the scaling term $\frac{1}{2\sigma_v^2} \cdot \frac{\sigma_\alpha^2}{\sigma_\alpha^2 + \sigma_v^2}$ used in the calculation of P grows rapidly with SNR, so that large coefficients will almost certainly be selected by the thresholding step when there is little noise.

Figure 6. Reconstruction error vs. number of iterations for randomized IHT

2.6.1 A Simplification

What if, instead of randomizing the thresholding operator at each step, we randomized only the last thresholding operation? The result obtained by the Landweber iteration at the final step

$$\hat{\alpha}^k + \Phi^*(y - \Phi\hat{\alpha}^k)$$

can be thresholded multiple times using $H_{s,P}$ to generate distinct sparse representations. Do these mimic the oracle estimates over various supports that are used in the MMSE sum? Some numerical simulation suggests that this is indeed the case. Modifying only the final step leads to a result that is still better than the IHT solution, and only slightly worse than if we had randomized each step:

Figure 7. Reconstruction error vs. SNR for IHT with the thresholding operator randomized at each step, and with the thresholding operator randomized only for the last step

In other words, we can run standard IHT to convergence, apply one additional Landweber iteration, then perform a random thresholding $H_{s,P}$ multiple times to obtain a sequence of distinct sparse representations. These can then be averaged to give an MMSE-type estimate. A very interesting observation here is that we need not have started with the IHT solution at all; it seems that a sparse representation obtained from an arbitrary algorithm could be improved upon by performing this Landweber iteration + thresholding step! This is a very attractive possibility since it requires minimal computation; the

Figure 8. Improvement on solutions from various algorithms obtained by this method

3 Conclusions, Future Work

In future work we will further examine the Landweber iteration + random thresholding algorithm introduced in this paper. We will search for potential results on necessary or sufficient conditions on the initial solution. One very interesting topic is quantifying the expected improvement obtained by replacing MAP with MMSE in terms of the measurement/sparsity dictionary and known priors. Is there a way to automatically detect whether an MMSE-type solution is worthwile? Under what conditions will a Landweber iteration + thresholing produce a solution that is worse than the starting solution?

...

4 Bibliography

CHECK FORMATTING

1. E. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. (IEEE Trans. on Information Theory, 52(2) pp. 489 - 509, February 2006)
2. D. Donoho, Compressed sensing. (IEEE Trans. on Information Theory, 52(4), pp. 1289 - 1306, April 2006)

3. B. K. Natarajan, "Sparse approximate solutions to linear systems," SIAM J. Comput., vol. 24, pp. 227-234, 1995.

4. M. Elad and I. Yavneh, "A plurality of sparse representations is better than the sparsest one alone," IEEE Trans. Inf. Theory, vol. 55, no. 10, pp. 4701-4714, Oct. 2009.

5. T. Blumensath and M. Davies, Iterative thresholding for sparse approximations, J. Fourier Anal. Appl., Volume 14, Numbers 5-6, 629-654, 2008.

6. M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing

7. M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky, A wide-angle view at iterated shrinkage algorithms, in Proceedings of the SPIE (Wavelet XII), San Diego, CA, 2007.

8. L. Landweber, "An iterative formula for fredholm integrals of the first kind," American Journal of Mathematics, vol. 73, pp. 615-624, Jul 1951.

9. T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal., 27(3):265-274, 2009.