Sparse Solution of Underdetermined Linear Equations by Stagewise Orthogonal Matching Pursuit

David L. Donoho [1], Yaakov Tsaig [2], Iddo Drori [1], Jean-Luc Starck [3]

March 2006

Abstract

Finding the sparsest solution to underdetermined systems of linear equations $y = \Phi x$ is NP-hard in general. We show here that for systems with 'typical'/'random' Φ, a good approximation to the sparsest solution is obtained by applying a fixed number of standard operations from linear algebra.

Our proposal, Stagewise Orthogonal Matching Pursuit (StOMP), successively transforms the signal into a negligible residual. Starting with initial residual $r_0 = y$, at the s-th stage it forms the 'matched filter' $\Phi^T r_{s-1}$, identifies all coordinates with amplitudes exceeding a specially-chosen threshold, solves a least-squares problem using the selected coordinates, and subtracts the least-squares fit, producing a new residual. After a fixed number of stages (e.g. 10), it stops. In contrast to Orthogonal Matching Pursuit (OMP), many coefficients can enter the model at each stage in StOMP while only one enters per stage in OMP; and StOMP takes a fixed number of stages (e.g. 10), while OMP can take many (e.g. n). StOMP runs much faster than competing proposals for sparse solutions, such as ℓ_1 minimization and OMP, and so is attractive for solving large-scale problems.

We use phase diagrams to compare algorithm performance. The problem of recovering a k-sparse vector x_0 from (y, Φ) where Φ is random $n \times N$ and $y = \Phi x_0$ is represented by a point $(n/N, k/n)$ in this diagram; here the interesting range is $k < n < N$. For n large, StOMP correctly recovers (an approximation to) the sparsest solution of $y = \Phi x$ over a region of the sparsity/indeterminacy plane comparable to the region where ℓ_1 minimization is successful. In fact, StOMP outperforms both ℓ_1 minimization and OMP for extremely underdetermined problems.

We rigorously derive a conditioned Gaussian distribution for the matched filtering coefficients at each stage of the procedure and rigorously establish a large-system limit for the performance variables of StOMP . We precisely calculate large-sample phase transitions; these provide asymptotically precise limits on the number of samples needed for approximate recovery of a sparse vector by StOMP .

We give numerical examples showing that StOMP rapidly and reliably finds sparse solutions in compressed sensing, decoding of error-correcting codes, and overcomplete representation.

Keywords: compressed sensing, decoding error-correcting codes, sparse overcomplete representation. phase transition, large-system limit. random matrix theory. Gaussian approximation. $\ell_1$minimization. stepwise regression. thresholding, false discovery rate, false alarm rate. MIMO channel, mutual access interference, successive interference cancellation. iterative decoding.

Acknowledgements This work was supported by grants from NIH, ONR-MURI, a DARPA BAA, and NSF DMS 00-77261, DMS 01-40698 (FRG) and DMS 05-05303.

[1]: Department of Statistics, Stanford University, Stanford CA, 94305
[2]: Institute for Computational Mathematics in Engineering, Stanford University, Stanford CA, 94305
[3]: DAPNIA/SEDI-SAP, Service d'Astrophysique, Centre Europeen d'Astronomie/Saclay, F-91191Gif-sur-Yvette Cedex France.

1 Introduction

The possibility of *exploiting sparsity* in signal processing is attracting growing attention. Over the years, several applications have been found where signals of interest have sparse representations and exploiting this sparsity offers striking benefits; see for example [11, 28, 26, 25, 7]. At the ICASSP 2005 conference a special session addressed the theme of exploiting sparsity, and a recent international workshop, SPARS05, was largely devoted to this topic.

Very recently, considerable attention has focused on the following *Sparse Solutions Problem* (SSP). We are given an $n \times N$ matrix Φ which is in some sense 'random', for example a matrix with iid Gaussian entries. We are also given an n-vector y and we know that $y = \Phi x_0$ where x_0 is an unknown sparse vector. We wish to recover x_0; however, crucially, $n < N$, the system of equations is underdetermined and so of course, this is not a properly-stated problem in linear algebra. Nevertheless, sparsity of x_0 is a powerful property that sometimes allows unique solutions. Applications areas for which this model is relevant include:

App1: *Compressed Sensing.* x_0 represents the coefficients of a signal or image in a known basis which happens to sparsely represent that signal or image. Φ encodes a measurement operator, i.e. an operator yielding linear combinations of the underlying object. Here $n < N$ means that we collect fewer data than unknowns. Despite the indeterminacy, sparsity of x_0 allows for accurate reconstruction of the object from what would naively seem to be 'too few samples' [17, 8, 48].

App2: *Error Correction.* Information is transmitted in a coded block in which a small fraction of the entries may be corrupted. From the received data, one constructs a system $y = \Phi x_0$; here x_0 represents the values of errors which must be identifed/corrected, y represents (generalized) check sums, and Φ represents a generalized checksum operator. It is assumed that the number of errors is smaller than a threshold, and so x_0 is sparse. This sparsity allows to perfectly correct the gross errors [9, 48, 28].

App3: *Sparse Overcomplete Representation.* x_0 represents the synthesis coefficients of a signal y, which is assumed to be sparsely represented from terms in an overcomplete expansion; those terms are the columns of Φ. The sparsity allows to recover a unique representation using only a few terms, despite the fact that representation is in general nonunique [43, 11, 21, 20, 50, 51].

In these applications, several algorithms are available to pursue sparse solutions; in some cases attractive theoretical results are known, guaranteeing that the solutions found are the sparsest possible solutions. For example, consider the algorithm of ℓ_1 minimization, which finds the solution to $y = \Phi x$ having minimal ℓ_1 norm. Also called Basis Pursuit (BP) [11], this method enjoys some particularly striking theoretical properties, such as rigorous proofs of exact reconstruction under seemingly quite general circumstances [21, 35, 32, 7, 16, 8, 17, 18]

Unfortunately, some of the most powerful theoretical results are associated with fairly heavy computationally burdens. The research reported here began when, in applying the theory of compressed sensing to NMR spectroscopy, we found that a straightforward application of the ℓ_1 minimization ideas in [17, 58] required several days solution time per (multidimensional) spectrum. This seemed prohibitive computational expense to us, even though computer time is relatively less precious than spectrometer time.

In fact, numerous researchers have claimed that general-purpose ℓ_1 minimization is much too slow for large-scale applications. Some have advocated a heuristic approach, Orthogonal Matching Pursuit (OMP), (also called greedy approximation and stepwise regression in other fields) [43, 52, 53, 55, 54], which though often effective in empirical work, does not offer the strong theoretical guarantees that attach to ℓ_1 minimization. (For other heuristic approaches, see [50, 51, 29].)

In this paper we describe Stagewise Orthogonal Matching Pursuit (StOMP), a method for approximate sparse solution of underdetermined systems with the property either that Φ is 'random' or that the nonzeros in x_0 are randomly located, or both. StOMP is significantly faster than the earlier methods BP and OMP on large-scale problems with sparse solutions. Moreover, StOMP permits a theoretical analysis showing that StOMP is similarly succcessful to BP at finding sparse solutions.

Our analysis uses the notion of *comparison of phase transitions* as a performance metric. We consider the *phase diagram*, a 2D graphic with coordinates measuring the relative sparsity of x_0 (number

of nonzeros in x_0/number of rows in Φ), as well as the indeterminacy of the system $y = \Phi x$ (number of rows in Φ/number of columns in Φ). StOMP 's large-n performance exhibits two phases (success/failure) in this diagram, as does the performance of BP. The "success phase" (the region in the phase diagram where StOMP successfully finds sparse solutions) is large and comparable in size to the success phase for ℓ_1 minimization. In a sense StOMP is more effective at finding sparse solutions to large extremely underdetermined problems than either ℓ_1 minimization or OMP; its phase transition boundary is even higher at extreme sparsity than the other methods. Moreover, StOMP takes a few seconds to solve problems that may require days for ℓ_1 solution. As a result StOMP is well suited to large-scale applications. Indeed we give several explicitly worked-out examples of realistic size illustrating the performance benefits of this approach.

Our analysis suggests the slogan

noiseless underdetermined problems behave like noisy well-determined problems,

i.e. coping with incompleteness of the measurement data is (for 'random Φ') similar to coping with Gaussian noise in complete measurements. At each StOMP stage, the usual set of matched filter coefficients is a mixture of 'noise' caused by cross-talk (non-orthogonality) and true signal; setting an appropriate threshold, we can subtract identified signal, causing a reduction in cross-talk at the next iteration. This is more than a slogan; we develop a theoretical framework for rigorous asymptotic analysis. Theorems 1-3 below allow us to track explicitly the successful capture of signal, and the reduction in cross-talk, stage by stage, rigorously establishing (asymptotic) success below phase transition, together with the failure that occurs above phase transition. The theory agrees with empirical finite-n results.

Our paper is organized as follows. Section 2 presents background on the sparse solutions problem; Section 3 introduces the StOMP algorithm and documents its favorable performance; Section 4 develops a performance measurement approach based on the phase diagram and phase transition. Section 5 analyzes the computational complexity of the algorithm. Section 6 develops an analytic large-system-limit for predicting phase transitions which agree with empirical performance characteristics of StOMP . Section 7 develops the Gaussian noise viewpoint which justifies our thresholding rules. Section 8 describes the performance of StOMP under variations [adding noise, of distribution of nonzero coefficients, of matrix ensemble] and documents the good performance of StOMP under all these variations.

Section 9 presents computational examples in applications App1-App3 that document the success of the method in simulated model problems. Section 10 describes the available software package which reproduces the results in this paper and Section 11 discusses the relationship of our results to related ideas in multiuser detection theory and to previous work in the sparse solutions problem.

2 Sparse Solution Preliminaries

Recall the Sparse Solutions Problem (SSP) mentioned in the introduction. In the SSP, an unknown vector $x_0 \in R^N$ is of interest; it is assumed sparse, which is to say that the number k of nonzeros is much smaller than N. We have the linear measurements $y = \Phi x_0$ where Φ is a known n by N matrix, and wish to recover x_0.

Of course, if Φ were a nonsingular square matrix, with $n = N$, we could easily recover x from y; but we are interested in the case where $n < N$. Elementary linear algebra tells us that x_0 is then not uniquely recoverable from y by linear algebraic means, as the equation $y = \Phi x$ may have many solutions. However, we are seeking a sparse solution, and for certain matrices Φ, sparsity will prove a powerful constraint. Some of the rapidly accumulating literature documenting this phenomenon includes [21, 20, 32, 55, 56, 50, 51, 8, 18, 16, 57, 58, 48].

For now, we consider a specific collection of matrices where sparsity proves valuable. Until we say otherwise, let Φ be a random matrix taken from the *Uniform Spherical ensemble* (USE); the columns of Φ are iid points on the unit sphere $\mathbf{S}^{n-1}$ [16, 17]. Later, several other ensembles will be introduced.

3 Stagewise Orthogonal Matching Pursuit

StOMP aims to achieve an approximate solution to $y = \Phi x_0$ where Φ comes from the USE and x_0 is sparse. In this section, we describe its basic ingredients. In later sections we document and analyse its

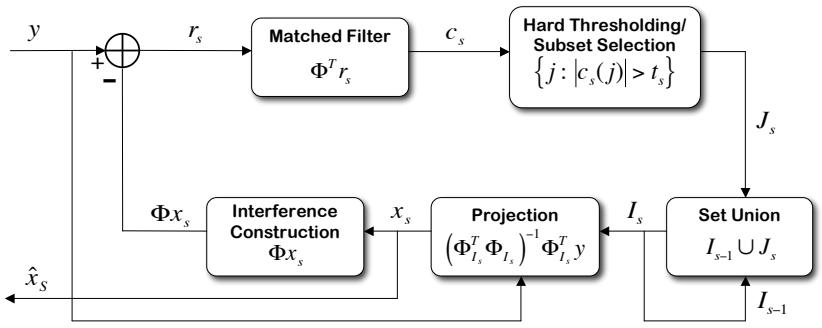

Figure 1: Schematic Representation of the StOMP algorithm.

performance.

3.1 The Procedure

StOMP operates in S stages, building up a sequence of approximations $x_0, x_1, \ldots$ by removing detected structure from a sequence of residual vectors $r_1, r_2, \ldots$. Figure 1 gives a diagrammatic representation.

StOMP starts with initial 'solution' $x_0 = 0$ and initial residual $r_0 = y$. The stage counter s starts at $s = 1$. The algorithm also maintains a sequence of estimates $I_1, \ldots, I_s$ of the locations of the nonzeros in x_0.

The s-th stage applies matched filtering to the current residual, getting a vector of residual *correlations*

$$c_s = \Phi^T r_{s-1},$$

which we think of as containing a small number of significant nonzeros in a vector disturbed by Gaussian noise in each entry. The procedure next performs hard thresholding to find the significant nonzeros; the thresholds, are specially chosen based on the assumption of Gaussianity [see below]. Thresholding yields a small set J_s of "large" coordinates:

$$J_s = \{j : |c_s(j)| > t_s \sigma_s\};$$

here σ_s is a formal noise level and t_s is a threshold parameter. We merge the subset of newly selected coordinates with the previous support estimate, thereby updating the estimate:

$$I_s = I_{s-1} \cup J_s.$$

We then project the vector y on the columns of Φ belonging to the enlarged support. Letting Φ_I denote the $n \times |I|$ matrix with columns chosen using index set I, we have the new approximation x_s supported in I_s with coefficients given by

$$(x_s)_{I_s} = (\Phi_{I_s}^T \Phi_{I_s})^{-1} \Phi_{I_s}^T y.$$

The updated residual is

$$r_s = y - \Phi x_s.$$

We check a stopping condition and, if it is not yet time to stop, we set $s := s + 1$ and go to the next stage of the procedure. If it is time to stop, we set $\hat{x}_S = x_s$ as the final output of the procedure.

Remarks:

1. The procedure resembles Orthogonal Matching Pursuit (known to statisticians as Forward Stepwise Regression). In fact the two would give identical results if S were equal to n and if, by coincidence, the threshold t_s were set in such a way that a single new term were obtained in J_s at each stage.

2. The thresholding strategy used in StOMP (to be described below) aims to have numerous terms enter at each stage, and aims to have a fixed number of stages. Hence the results will be different from OMP.

3. The formal noise level $\sigma_s = \|r_s\|_2/\sqrt{n}$, and typically the threshold parameter takes values in the range $2 \le t_s \le 3$.

4. There are strong connections to: *stagewise/stepwise regression* in statistical model building; *successive interference cancellation* multiuser detectors in digital communications and *iterative decoders* in error-control coding. See the discussion in Section 11.

Our recommended choice of S (10) and our recommended threshold-setting procedures (see Section 3.4 below) have been designed so that when x_0 is sufficiently sparse, the following two conditions are likely to hold upon termination:

- All nonzeros in x_0 are selected in I_S.
- I_S has no more than n entries.

The next lemma motivates this design criterion. Recall that Φ is sampled from the USE and so columns of Φ are in general position. The following is proved in Appendix A.

Lemma 3.1 *Let the columns of Φ be in general position. Let I_S denote the support of $\hat{x}_S$. Suppose that the support I_0 of x_0 is a subset of I_S. Suppose in addition that $\#I_S \le n$. Then we have perfect recovery:*

$$\hat{x}_S = x_0. \tag{3.1}$$

3.2 An Example

We give a simple example showing that the procedure works in a special case.

We generated a coefficient vector x_0 with $k = 32$ nonzeros, having amplitudes uniformly distributed on $[0, 1]$. We sampled a matrix Φ at random from the USE with $n = 256$, $N = 1024$, and computed a linear measurement vector $y = \Phi x_0$. Thus the problem of recovering x_0 given y is $1 : 4$ underdetermined (i.e. $\delta = n/N = .25$), with underlying sparsity measure $\rho = k/n = .125$. To this SSP, we applied StOMP coupled with the CFAR threshold selection rule to be discussed below. The results are illustrated in Figure 2.

Panels (a)-(i) depict each matched filtering output, its hard thresholding and the evolving approximation. As can be seen, after 3 stages a result is obtained which is quite sparse and, as the final panel shows, matches well the object x_0 which truly generated the data. In fact, the error at the end of the third stage measures $\|\hat{x}_3 - x_0\|_2/\|x_0\|_2 = 0.022$, i.e. a mere 3 stages were required to achieve an accuracy of 2 decimal digits.

3.3 Approximate Gaussianity of Residual MAI

At the heart of our procedure are two thresholding schemes often used in Gaussian noise removal. (N.B. at this point we assume there is no noise in y!) To explain the relevance of Gaussian 'noise' concepts, note that at stage 1, the algorithm is computing

$$\tilde{x} = \Phi^T y;$$

this is essentially the usual matched filter estimate of x_0. If $y = \Phi x_0$ and x_0 vanishes except in one coordinate, the matched filter output $\tilde{x}$ equals x_0 perfectly. Hence $z = \tilde{x} - x_0$ is a measure of the disturbance to exact reconstruction caused by multiple nonzeros in x_0. The same notion arises in digital communications where it is called *Multiple-Access Interference* (MAI) [60]. Perhaps surprisingly - because there is no noise in the problem - the MAI in our setting typically has a Gaussian behavior. More

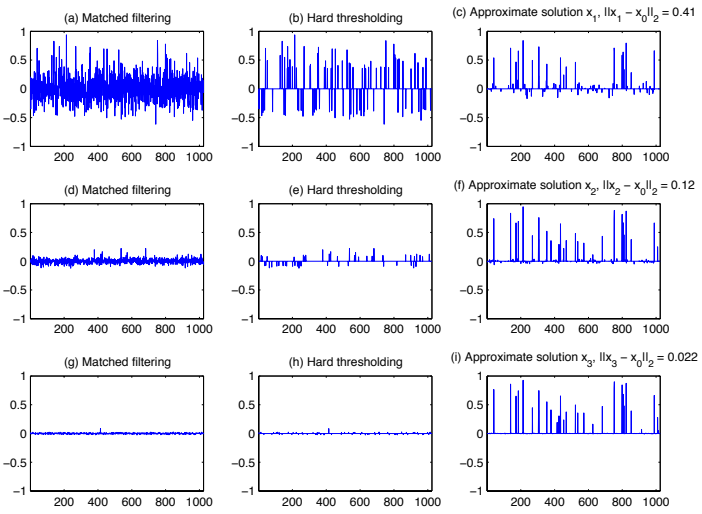

Figure 2: Progression of the StOMP algorithm. Panels (a),(d),(g): successive matched filtering outputs c_1,c_2, c_3; Panels (b),(e),(h): successive thresholding results; Panels (c),(f),(i): successive partial solutions. In this example, $k = 32$, $n = 256$, $N = 1024$.

specifically, if Φ is a matrix from the USE and if n and N are both large, then the entries in the MAI vector z have a histogram which is nearly Gaussian with standard deviation

$$\sigma \approx \|x_0\|_2/\sqrt{n}. \tag{3.2}$$

The heuristic justification is as follows. The MAI has the form

$$z(j) = \tilde{x}(j) - x_0(j) = \sum_{j \neq \ell} \langle \phi_j, \phi_\ell \rangle x_0(\ell).$$

The thing we regard as 'random' in this expression is the matrix Φ. The term $\xi_k^j \equiv \langle \phi_j, \phi_k \rangle$ measures the projection of a random point on the sphere $\mathbf{S}^{n-1}$ onto another random point. This random variable has approximately a Gaussian distribution $N(0, \frac{1}{n})$. For Φ from the USE, for a given fixed ϕ_j, the different random variables $(\xi_k^j : k \neq j)$ are independently distributed. Hence the quantity $z(j)$ is an iid sum of approximately normal r.v.'s, and so, by standard arguments, should be approximately normal with mean 0 and variance

$$\sigma_j^2 = Var[\sum_{j \neq \ell} \xi_\ell^j x_0(\ell)] = (\sum_{j \neq \ell} x_0(\ell)^2) \cdot Var(\xi_1^j) \approx n^{-1} \|x_0\|_2^2$$

Setting $\sigma^2 = \|x_0\|^2/n$, this justifies (3.2).

Computational experiments validate Gaussian approximation for the MAI. In Figure 3, Panels (a),(d),(g) display Gaussian QQ-plots of the MAI in the sparse case with $k/n = .125$, $.1875$ and $.25$, in the Uniform Spherical Ensemble with $n = 256$ and $N = 1024$. In each case, the QQ-plot appears straight, as the Gaussian model would demand.

Through the rest of this paper, the phrase *Gaussian approximation* means that the MAI has an approximately Gaussian marginal distribution. (The reader interested in formal proofs of Gaussian approximation can consult the literature of multiuser detection e.g. [46, 61, 12]; such a proof is implicit in the proofs of Theorems 1 and 2 below. The connection between our work and MUD theory will be amplified in Section 11 below).

Properly speaking, the term 'MAI' applies only at stage 1 of StOMP . At later stages there is *residual MAI*, i.e. MAI which has not yet been cancelled. This can be defined as

$$z_s(j) = x_0(j) - \phi_j^T r_s / \|P_{I_{s-1}} \phi_j\|_2^2, \quad j \notin I_{s-1};$$

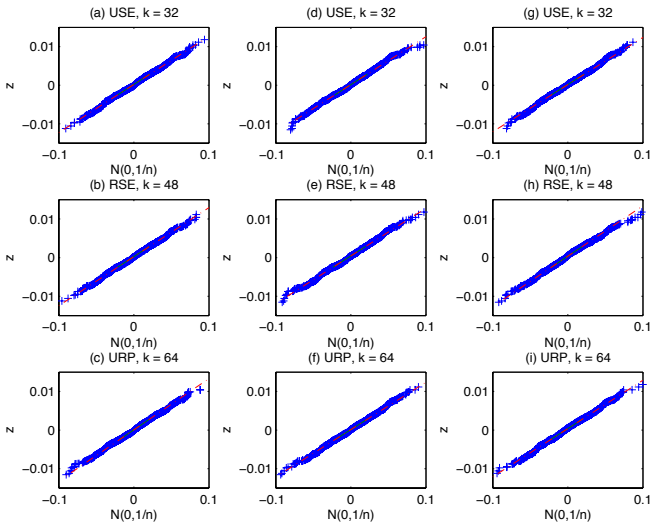

Figure 3: QQ plots comparing MAI with Gaussian distribution. Left column: $k/n = .125$, middle column: $k/n = .1875$, right column: $k/n = .25$. Top row: USE, middle row: RSE, bottom row: URP. The RSE and URP ensembles are discussed in Section 8. The plots all appear nearly linear, indicating that the MAI has a nearly Gaussian distribution.

the coordinates $j \in I_{s-1}$ are ignored at stage s - the residual in those coordinates is deterministically 0.

Empirically, residual MAI has also a Gaussian behavior. Figure 4 shows quantile-quantile plots for the first few stages of the CFAR variant, comparing the residual MAI with a standard normal distribution. The plots are effectively straight lines, illustrating the Gaussian approximation. Later, we provide theoretical support for a perturbed Gaussian approximation to residual MAI.

3.4 Threshold Selection

Our threshold selection proposal is inspired by the Gaussian behavior of residual MAI. We view the vector of correlations c_s at stage s as consisting of a small number of 'truly nonzero' entries, combined with a large number of 'Gaussian noise' entries. The problem of separating 'signal' from 'noise' in such problems has generated a large literature including the papers [24, 27, 26, 1, 23, 37], which influenced our way of thinking.

We adopt language from statistical decision theory [39] and the field of multiple comparisons [38]. Recall that the support I_0 of x_0 is being (crudely) estimated in the StOMP algorithm. If a coordinate belonging to I_0 does not appear in I_S, we call this a *missed detection*. If a coordinate not in I_0 does appear in I_S we call this a *false alarm*. The coordinates in I_S we call *discoveries*, and the coordinates in $I_S \backslash I_0$ we call *false discoveries*. (Note: false alarms are also false discoveries. The terminological distinction is relevant when we normalize to form a rate; thus the false alarm rate is the number of false alarms divided by the number of coordinates not in I_0; the false discovery rate is the fraction of false discoveries within I_S.)

We propose two strategies for setting the threshold. Ultimately, each strategy should land us in a position to apply Lemma 3.1: i.e. to arrive at a state where $\#I_S \leq n$ and there are no missed detections. Then, Lemma 3.1 assures us, we perfectly recover: $\hat{x}_S = x$. The two strategies are:

- *False Alarm Control.* We attempt to guarantee that the number of total false alarms, across all stages, does not exceed the natural codimension of the problem, defined as $n - k$. Subject to this, we attempt to make the maximal number of discoveries possible. To do so, we choose a threshold so the False Alarm rate at each stage does not exceed a per-stage budget.

- *False Discovery Control.* We attempt to arrange that the number of False Discoveries cannot exceed

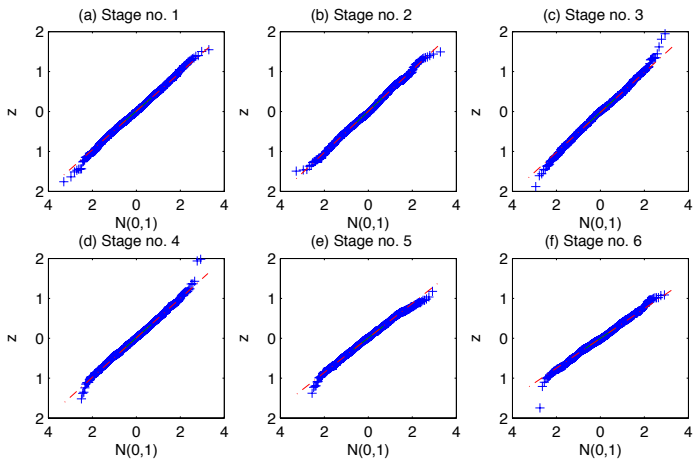

Figure 4: QQ plots comparing residual MAI with Gaussian distribution. Quantiles of residual MAI at different stages of StOMP are plotted against Gaussian quantiles. Near-linearity indicates approximate Gaussianity.

a fixed fraction q of all discoveries, and to make the maximum number of discoveries possible subject to that constraint. This leads us to consider Simes' rule [2, 1].

The False Alarm Control strategy requires knowledge of the number of nonzeros k or some upper bound. False Discovery Control does not require such knowledge, which makes it more convenient for applications, if slightly more complex to implement and substantially more complex to analyse [1]. The choice of strategy matters; the basic StOMP algorithm behaves differently depending on the threshold strategy, as we will see below.

Implementation details are available by downloading the software we have used to generate the results in this paper; see Section 10 below.

4 Performance Analysis by Phase Transition

When does StOMP work? To discuss this, we use the notions of phase diagram and phase transition.

4.1 Problem Suites, Performance Measures

By *problem suite* $\mathcal{S}(k, n, N)$ we mean a collection of Sparse Solution Problems defined by two ingredients: (a) an ensemble of random matrices Φ of size n by N; (b) an ensemble of k-sparse vectors x_0. By *standard* problem suite $\mathcal{S}_{st}(k, n, N)$ we mean the suite with Φ sampled from the uniform spherical ensemble, with x_0 a random variable having k nonzeros sampled iid from a standard $N(0, 1)$ distribution.

For a given problem suite, a specific algorithm can be run numerous times on instances sampled from the problem suite. Its performance on each realization can then be measured according to some numerical or qualitative criterion. If we are really ambitious, and insist on perfect recovery, we use the performance measure $1_{\{\hat{x}_S \neq x_0\}}$. More quantitative is the ℓ_0-norm, $\|\hat{x}_S - x_0\|_0$, the number of sites at which the two vectors disagree. Both these measures are inappropriate for use with floating point arithmetic, which does not produce exact agreement. We prefer to use instead $\ell_{0,\epsilon}$, the number of sites at which the reconstruction and the target disagree by more than $\epsilon = 10^{-4}$. We can also use the quantitative measure $relerr_2 = \|\hat{x}_S - x_0\|_2 / \|x_0\|_2$, declaring success when the measure is smaller than a fixed threshold (say ϵ).

For a qualitative performance indicator we simply report the fraction of realizations where the qualitative condition was true; for a quantitative performance measure, we present the mean value across instances at a given k, n, N.

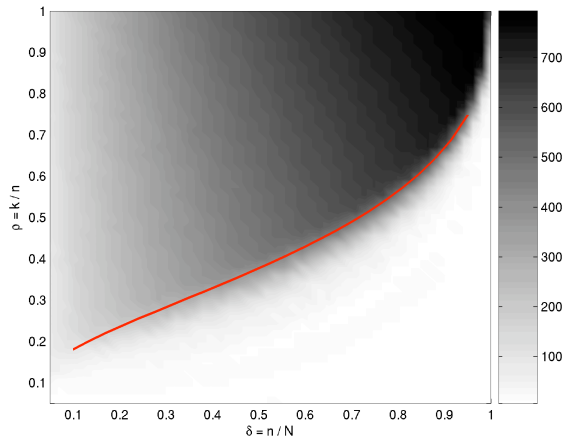

Figure 5: Phase Diagram for ℓ_1 minimization. Shaded attribute is the number of coordinates of reconstruction which differ from optimally sparse solution by more than 10^{-4}. The diagram displays a rapid transition from perfect reconstruction to perfect disagreement. Overlaid red curve is theoretical curve ρ_{ℓ_1}.

4.2 Phase Diagram

A *phase diagram* depicts performance of an algorithm at a sequence of problem suites $\mathcal{S}(k, n, N)$. The average value of some performance measure as displayed as a function of $\rho = k/n$ and $\delta = n/N$. Both of these variables $\rho, \delta \in [0, 1]$, so the diagram occupies the unit square.

To illustrate such a phase diagram, consider a well-studied case where something interesting happens. Let x_1 solve the optimization problem:

$$(P_1) \quad \min \|x\|_1 \text{ subject to } y = \Phi x.$$

As mentioned earlier, if $y = \Phi x_0$ where x_0 has k nonzeros, we may find that $x_1 = x_0$ exactly when k is small enough. Figure 5 displays a grid of $\delta - \rho$ values, with δ ranging through 50 equispaced points in the interval $[.05, .95]$ and ρ ranging through 50 equispaced points in $[.05, .95]$; here $N = 800$. Each point on the grid shows the mean number of coordinates at which original and reconstruction differ by more than 10^{-4}, averaged over 100 independent realizations of the standard problem suite $\mathcal{S}_{st}(k, n, N)$. The experimental setting just described, i.e. the $\delta - \rho$ grid setup, the values of N, and the number of realizations, is used to generate phase diagrams later in this paper, although the problem suite being used may change.

This diagram displays a *phase transition*. For small ρ, it seems that high-accuracy reconstruction is obtained, while for large ρ reconstruction fails. The transition from success to failure occurs at different ρ for different values of δ.

This empirical observation is explained by a theory that accurately predicts the location of the observed phase transition and shows that, asymptotically for large n, this transition is perfectly sharp. Suppose that problem (y, Φ) is drawn at random from the standard problem suite, and consider the event $E_{k,n,N}$ that $x_0 = x_1$ i.e. that ℓ_1 minimization exactly recovers x_0. The paper [19] defines a function $\rho_{\ell_1}(\delta)$ (called there ρ_W) with the following property. Consider sequences of (k_n), (N_n) obeying $k_n/n \to \rho$ and $n/N_n \to \delta$. Suppose that $\rho < \rho_{\ell_1}(\delta)$. Then as $n \to \infty$

$$Prob(E_{k_n,n,N_n}) \to 1.$$

On the other hand, suppose that $\rho > \rho_{\ell_1}(\delta)$. Then as $n \to \infty$

$$Prob(E_{k_n,n,N_n}) \to 0.$$

The theoretical curve $(\delta, \rho_{\ell_1}(\delta))$ described there is overlaid on Figure 5, showing good agreement between asymptotic theory and experimental results.

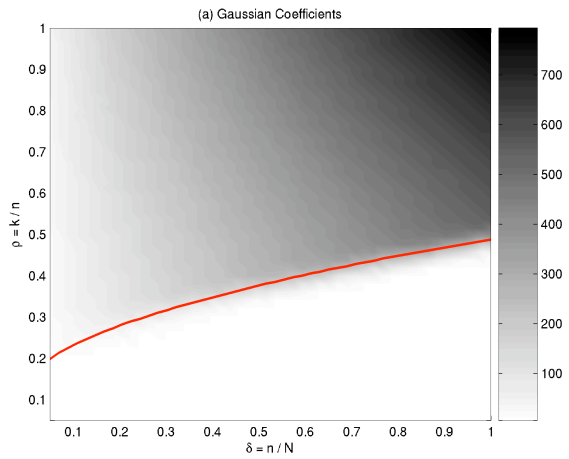

Figure 6: Phase diagram for CFAR thresholding. Overlaid red curve is heuristically-derived analytical curve ρ_{FAR} (see Appendix B). Shaded attribute: number of coordinates wrong by more than 10^{-4} relative error.

4.3 Phase Diagrams for StOMP

We now use phase diagrams to study the behavior of StOMP . Figure 6 displays performance of StOMP with CFAR thresholding with per-iteration false alarm rate $(n-k)/(S(N-k))$. The problem suite and underlying problem size, $N = 800$, are the same as in Figure 5. The shaded attribute again portrays the number of entries where the reconstruction misses by more than 10^{-4}. Once again, for very sparse problems (ρ small), the algorithm is successful at recovering (a good approximation to) x_0, while for less sparse problems (ρ large), the algorithm fails. Superposed on this display is the graph of a heuristically-derived function ρ_{FAR}, which we call the *Predicted Phase transition for CFAR thresholding*. Again the agreement between the simulation results and the predicted transition is reasonably good. Appendix B explains the calculation of this predicted transition, although it is best read only after first reading Section 6.

Figure 7 shows the number of mismatches for the StOMP algorithm based on CFDR thresholding with False Discovery Rate $q = 1/2$. Here $N = 800$ and the display shows that, again, for very sparse problems (ρ small), the algorithm is successful at recovering (a good approximation to) x_0, while for less sparse problems ρ large, the algorithm fails. Superposed on this display is the graph of a heuristically-derived function ρ_{FDR}, which we call the *Predicted Phase transition for CFDR thresholding*. Again the agreement between the simulation results and the predicted transition is reasonably good, though visibly not quite as good as in the CFAR case.

5 Computation

Since StOMP seems to work reasonably well, it makes sense to study how rapidly it runs.

5.1 Empirical Results

Table 1 shows the running times for StOMP equipped with CFAR and CFDR thresholding, solving an instance of the problem suite $\mathcal{S}_{st}(k, n, N)$. We compare these figures with the time needed to solve the same problem instance via ℓ_1 minimization and OMP. Here ℓ_1 minimization is implemented using Michael Saunders' PDCO solver [49]. The simulations used to generate the figures in the table were all executed on a 3GHz Xeon workstation, comparable with current desktop CPUs.

Table 1 suggests that a tremendous saving in computation time is achieved when using the StOMP scheme over traditional ℓ_1 minimization. The conclusion is that CFAR- and CFDR- based methods have a large

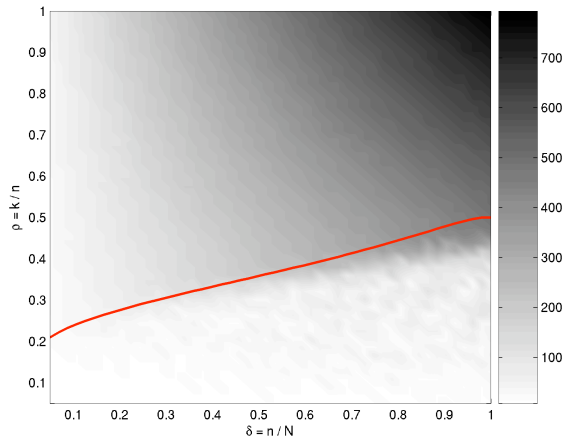

Figure 7: Phase diagram for CFDR thresholding. Overlaid red curve is heuristically-derived curve ρ_{FDR} (see Appendix B). Shaded attribute: number of coordinates wrong by more than 10^{-4} relative error.

Problem Suite (k,n,N)	ℓ_1	OMP	CFAR	CFDR
(10,100,1000)	0.97	0.37	0.02	0.03
(100,500,1000)	22.79	0.79	0.42	0.32
(100,1000,10000)	482.22	7.98	5.24	3.28
(500,2500,10000)	7767.16	151.81	126.36	88.48

Table 1: Comparison of execution times (in seconds) for instances of the random problem suite $\mathcal{S}_{st}(k, n, N)$.

domain of applicability for sparsely solving random underdetermined systems, while running much faster than other methods in problem sizes of current interest.

5.2 Complexity Analysis

The timing studies are supported by a formal analysis of the asymptotic complexity. In this analysis, we consider two scenarios.

- **Dense Matrices.** In this scenario, the matrix Φ defining an underdetermined linear system is an explicit $n \times N$ dense matrix stored in memory. Thus, applying Φ to an N-vector x involves nN flops.
- **Fast Operators.** Here, the linear operator Φ is not explicitly represented in matrix form. Rather, it is implemented as an operator taking a vector x, and returning Φx. Classical examples of this type include the Fourier transform, Hadamard transform, and Wavelet transform, just to name a few; all of these operators are usually implemented without matrix multiplication. Such fast operators are of key importance in large-scale applications. As a concrete example, consider an imaging scenario where the data is a $d \times d$ array, and Φ is an n by d^2 partial Fourier operator, with $n = \mu d^2$ proportional to d^2. Direct application of Φ would involve $nd^2 = O(d^4)$ operations, whereas applying a 2-D FFT followed by random sampling would require merely $O(d^2 \log(d))$ flops; the computational gain is evident. In our analysis below, we let V denote the cost of one application of a linear operator or its adjoint (corresponding to one matrix-vector multiplication).

In fact, as we will now see, the structure of the StOMP algorithm makes it a prime choice when fast operators are available, as nearly all its computational effort is invested in solving partial least-squares systems involving Φ and Φ^T. In detail, assume we are at the s-th stage of execution. StOMP starts by applying matched filtering to the current residual, which amount to one application of Φ^T, at a cost

of nN flops. Next, it applies hard-thresholding to the residual correlations and updates the active set accordingly, using at most $2N$ additional flops. The core of the computation lies in calculating the projection of y onto the subset of columns Φ_{I_s}, to get a new approximation x_s. This is implemented via a Conjugate Gradient (CG) solver [34]. Each CG iteration involves application of Φ_{I_s} and $\Phi_{I_s}^T$, costing at most $2nN + O(N)$ flops. The number of CG iterations used is a small constant, independent of n and N, which we denote ν. In our implementation we use $\nu = 10$. Finally, we compute the new residual by applying Φ to the new approximation, requiring an additional nN flops. Summarizing, the total operation count per StOMP stage amounts to $(\nu + 2)nN + O(N)$. The total number of StOMP stages, S, is a prescribed constant, independent of the data; in the simulations in this paper we set $S = 10$.

Readers familiar with OMP have by now doubtless recognized the evident parallelism in the algorithmic structure of StOMP and OMP. Indeed, much like StOMP , at each stage OMP computes residual correlations and solves a least-squares problem for the new solution estimate. Yet, unlike StOMP , OMP builds up the active set one element at a time. Hence, an efficient implementation would necessarily maintain a Cholesky factorization of the active set matrix and update it at each stage, thereby reducing the cost of solving the least-squares system. In total, k steps of OMP would take at most $4k^3/3+knN+O(N)$ flops. Without any sparsity assumptions on the data, OMP takes at most n steps, thus, its worst-case performance is bounded by $4n^3/3+n^2N+O(N)$ operations. A key difference between StOMP and OMP is that the latter needs to store the Cholesky factor of the active set matrix in its explicit form, taking up to $n^2/2$ memory elements. When n is large, as is often the case in 2- and 3-D image-reconstruction scenarios, this greatly hinders the applicability of OMP. In contrast, StOMP has very modest storage requirements. At any given point of the algorithm execution, one needs only store the current estimate x_s, the current residual vector r_s, and the current active set I_s. This makes StOMP very attractive for use in large-scale applications.

Table 2 summarizes our discussion so far, offering a comparison of the computational complexity of StOMP , OMP and ℓ_1 minimization via linear programming (LP). For the LP solver, we use a primal-dual barrier method for convex optimization (PDCO) developed by Michael Saunders [49]. The estimates listed in the table all assume worst-case behavior. Examining the bounds in the dense matrix case closely, we notice that StOMP is the only algorithm of the three admitting quadratic order complexity estimates. In contrast, OMP and PDCO require cubic order estimates for their worst-case performance bound. Therefore, for large scale problems StOMP can dominate due to its simple structure and efficiency. In the case where fast operators are applicable, StOMP yet again prevails; it is the only algorithm of the three requiring a constant number $(S \cdot (\nu + 2))$ of matrix-vector multiplications to reach a solution.

Algorithm	Dense Matrices	Fast Operators
StOMP	$S(\nu + 2)nN + O(N)$	$S(\nu + 2) \cdot V + O(N)$
OMP	$4n^3/3 + n^2N + O(N)$	$4n^3/3 + 2n \cdot V + O(N)$
ℓ_1 min. with PDCO	$S(2N)^3/3 + O(nN)$	$2N \cdot V + O(nN)$

Table 2: Worst-Case Complexity Bounds for StOMP , OMP and PDCO. S denotes the maximum number of stages, ν denotes the maximum number of CG iterations employed per stage of StOMP, and V stands for the cost of one matrix-vector product (implemented as a fast operator).

To convey the scale of computational benefits in large problems, we conduct a simple experiment in a setting where Φ can be implemented as a fast operator. We consider a system $y = \Phi x$ where Φ is made from only $n = \delta N$ rows of the Fourier matrix. Φ can be implemented by application of a Fast Fourier Transform followed by a coordinate selection. Table 3 gives the results. Clearly the advantage of StOMP is even more convincing.

Problem Suite	(k,n,N)	ℓ_1	OMP	CFAR	CFDR
$\mathcal{S}_{PFE}$	(500,10000,20000)	237.69	53.64	2.07	3.16
$\mathcal{S}_{PFE}$	(1000,20000,50000)	810.39	299.54	5.63	9.47

Table 3: Comparison of execution times (in seconds) in the random partial Fourier suite $\mathcal{S}_{PFE}(k, n, N)$. Because of the fast operator, StOMP outperforms OMP.

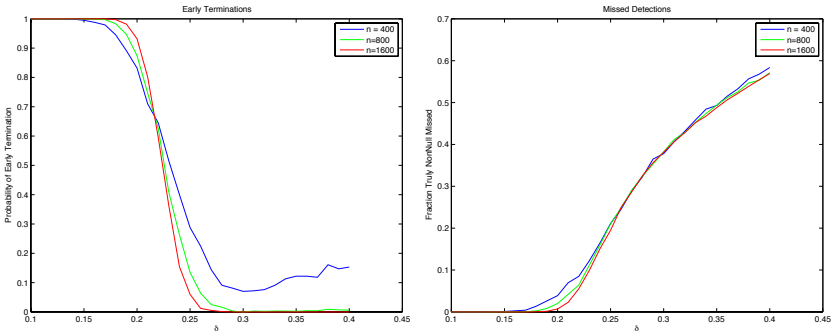

Figure 8: Empirical Transition Behaviors, varying n. (a) Fraction of cases with termination before stage S. (b) Fraction of missed detections. Averages of 1000 trials with $n = 400, 800, 1600$ and $k = \lfloor \rho n \rfloor$, $N = \lfloor n/\delta \rfloor$, $\delta = 1/4$ and ρ varying. Sharpness of each transition seems to be increasing with n.

To make the comparison still more vivid, we point ahead to an imaging example from Section 9.1 below. There an image of dimensions $d \times d$ is viewed as a vector x of length $N = d^2$. Again the system $y = \Phi x$ where Φ is made from only $n = \delta N$ rows of the Fourier matrix. One matrix-vector product costs $V = 4N \log N = 8d^2 \log d$.

How do the three algorithms compare in this setting? Plugging-in $S = 10$, $\nu = 10$, and V as above, we see that the leading term in the complexity bound for StOMP is $960 \cdot d^2 \log d$. In contrast, for OMP the leading term in the worst-case bound becomes $\frac{4\delta^3}{3} d^6 + 16\delta d^4 \log d$, and for ℓ_1 minimization the leading term is $16 d^4 \log d$. The computational gains from StOMP are indeed substantial. Moreover, to run OMP in this setting, we may need up to $\frac{\delta^2}{2} d^4$ memory elements to store the Cholesky factorization, which renders it unusable for anything but the smallest d. In Section 9.1, we present actual running times of the different algorithms.

6 The Large-System Limit

Figures 6 and 7 suggest phase transitions in the behavior of StOMP , which would imply a certain well-defined asymptotic 'system capacity' below which StOMP successfully finds a sparse solution, and above which it fails. In this section, we review the empirical evidence for a phase transition in the large-system limit and develop theory that rigorously establishes it. We consider the problem suite $\mathcal{S}(k, n, N; USE, \pm 1)$ defined by random Φ sampled from the USE, and with y generated as $y = \Phi x_0$, where x_0 has k nonzero coefficients in random positions having entries ± 1. This ensemble generates a slightly 'lower' transition than the ensemble used for Figures 6 and 7 where the nonzeros in x_0 had iid Gaussian $N(0, 1)$ entries.

6.1 Evidence for Phase Transition

Figure 8 presents results of simulations at fixed ratios $\delta = n/N$ but increasing n. Three different quantities are considered: in panel (a), the probability of *early termination*, i.e. termination before stage $S = 10$ because the residual has been driven nearly to zero; in panel (b) the missed detection rate, i.e. the fraction of nonzeros in x_0 that are not supported in the reconstruction $\hat{x}_S$. Both quantities undergo transitions in behavior near $\rho = .2$.

Significantly, the transitions become more sharply defined with increasing n. As n increases, the early termination probability behaves increasingly like a raw discontinuity $1_{\{k/n \geq \rho_{FAR}(n/N)\}}$ as $n \to \infty$, while the fraction of missed detections properties behave increasingly like a discontinuity in derivative $(k/n - \rho_{FAR}(n/N))_+$. In statistical physics such limiting behaviors are called first-order and second-order phase transitions, respectively.

(k,n,N)	ρ_1	ρ_2	ρ_3	ρ_4	d_1	d_2	d_3	d_4	ν_1	ν_2	ν_3	ν_4
(70,280,800)	0.250	0.130	0.074	0.041	1.000	0.773	0.630	0.521	2.857	2.630	2.487	2.377
(105,420,1200)	0.250	0.130	0.075	0.043	1.000	0.775	0.633	0.524	2.857	2.632	2.490	2.381
(140,560,1600)	0.250	0.131	0.075	0.043	1.000	0.776	0.632	0.522	2.857	2.633	2.489	2.379
(210,840,2400)	0.250	0.130	0.075	0.043	1.000	0.776	0.630	0.522	2.856	2.632	2.486	2.378
(78,280,800)	0.279	0.158	0.106	0.078	1.000	0.778	0.643	0.546	2.857	2.635	2.499	2.403
(117,420,1200)	0.279	0.157	0.104	0.073	1.000	0.781	0.646	0.546	2.857	2.638	2.502	2.403
(156,560,1600)	0.279	0.159	0.105	0.076	1.000	0.782	0.645	0.545	2.857	2.639	2.502	2.401
(235,840,2400)	0.280	0.159	0.106	0.076	1.000	0.778	0.644	0.542	2.856	2.635	2.501	2.399

Table 4: Dimension-normalized ratios ρ_s, d_s, ν_s. Problems of different sizes (k,n,N) with identical ratios of $\rho = k/n$ and $\delta = n/N$. Note similarity of entries in adjacent rows within each half of the table. Top half of table: just below phase transition; bottom half: just above1 phase transition.

(k,n,N)	α_1	α_2	α_3	α_4	β_1	β_2	β_3	β_4
(70,280,800)	0.041	0.035	0.032	0.031	0.481	0.429	0.444	0.421
(105,420,1200)	0.040	0.035	0.032	0.033	0.479	0.423	0.424	0.480
(140,560,1600)	0.040	0.035	0.032	0.031	0.478	0.428	0.427	0.481
(210,840,2400)	0.040	0.036	0.031	0.031	0.478	0.423	0.429	0.485
(78,280,800)	0.039	0.034	0.029	0.028	0.434	0.331	0.261	0.221
(117,420,1200)	0.038	0.033	0.029	0.030	0.436	0.341	0.295	0.248
(156,560,1600)	0.038	0.033	0.030	0.028	0.429	0.341	0.277	0.225
(235,840,2400)	0.039	0.033	0.030	0.028	0.433	0.332	0.278	0.220

Table 5: Dimension-normalized detector operating characteristics α_s, β_s. Problems of different sizes (k,n,N) with identical ratios of $\rho = k/n$ and $\delta = n/N$. Note similarity of entries in adjacent rows within each half of the table. Top half of table: just below phase transition; bottom half: just above phase transition.

6.2 Evidence for Intensivity

In statistical physics, a system property is called *intensive* when it tends to a stable limit as the system size increases. Many properties of StOMP , when expressed as ratios to the total system size, are intensive. Such properties include: the number of detections at each stage, the number of true detections, the number of false alarms, and the squared norm of the residual r_s. When sampling from the standard problem suite, all these properties - after normalization by the problem size n - behave as intensive quantities.

Table 4 illustrates this by considering 6 different combinations of k, n, N, all six at the same value of determinacy $\delta = n/N$, in two groups of three, each group at one common value of ρ. Within each group with common values of $\delta = n/N$ and $\rho = k/n$, we considered three different problem sizes n.

Stage s of StOMP considers an n_s-dimensional subspace, using k_s nonzeros out of N_s possible terms, where

$$\begin{aligned} k_s &= k - \#\text{true discoveries prior to stage } s, \\ n_s &= n - \#\text{discoveries prior to stage } s, \\ N_s &= N - \#\text{discoveries prior to stage } s. \end{aligned}$$

The table presents dimension-normalized ratios $\rho_s = k_s/n$, $d_s = n_s/n$, $\nu_s = N_s/n$. If these quantities are intensive, they will behave similarly at the same stage even at different problem sizes. The evidence of the table suggests that they are indeed intensive.

Also important in what follows are two threshold *detector operating characteristics*: the stage-specific false-alarm rate

$$\alpha_s = Prob\{|\langle \phi_j, r_s \rangle| > t_s \sigma_s | j \in I_0^c \cap I_{s-1}^c\}$$

and the stage-specific correct detection rate

$$\beta_s = Prob\{|\langle \phi_j, r_s \rangle| > t_s \sigma_s | j \in I_0 \cap I_{s-1}^c\}.$$

There is also evidence of intensivity for these quantities; see Table 5.

6.3 Limit Quantities

We have seen that the dimension-normalized quantities $\rho_s = k_s/n_s$ and $d_s = n_s/n$ are empirically nearly constant for large n. We now present a theoretical result to explain this.

For our result, we fix $S > 0$ and consider the CFAR algorithm designed for that specified S. We also fix $\rho, \delta \in (0,1)$. Let $k = k_n = \lfloor \rho n \rfloor$, $N_n = \lfloor n/\delta \rfloor$. Run StOMP on an instance of the problem suite $\mathcal{S}(k, n, N; USE, \pm 1)$. Let $\|r_s\|_2$ denote the norm of the residual at stage s.

Recall the notation p.lim for *limit in probability*; a sequence of random variables $(v_n : n \geq 1)$ has the nonstochastic limit $\bar{v}$ in probability, written $\bar{v} = \text{p.lim}_{n\to\infty} v_n$, if, for each $\epsilon > 0$, $Prob\{|v_n - \bar{v}| > \epsilon\} \to 0$ as $n \to \infty$. In the result below, let $k_{s,n}$ denote the random quantity k_s on a problem from the standard suite at size n. Similarly for $r_{s,n}$,$d_{s,n}$, $n_{s,n}$, $\alpha_{s,n}$, $\beta_{s,n}$. Also, if $n_{s,n} = 0$, the iteration stops immediately, and the monitoring variables at that and all later stages up to stage S are assigned values in the obvious way: $r_{s,n} = 0$, $\beta_{s,n} = 0$, $\alpha_{s,n} = 0$, etc.

Theorem 1 Large-System Limit. *There are constants $\bar{\sigma}_s$, $\bar{\mu}_s$ depending on $s = 1, \ldots, S$, on δ and on ρ, so that*

$$\bar{\sigma}_s^2 = \operatorname*{p.lim}_{n\to\infty} \|r_{s,n}\|_2^2/n, \qquad \bar{\mu}_s = \operatorname*{p.lim}_{n\to\infty} \|r_{s,n}\|_2^2/k_{s,n}, \quad s = 1, \ldots, S.$$

We also have large-system limits in probability for the detector operating characteristics

$$\bar{\alpha}_s = \operatorname*{p.lim}_{n\to\infty} \alpha_{s,n}, \qquad \bar{\beta}_s = \operatorname*{p.lim}_{n\to\infty} \beta_{s,n}, \quad s = 1, \ldots, S,$$

where the limits depend on s, δ and ρ. Finally, the normalized dimensions also have large-system limits:

$$\bar{\rho}_s = \operatorname*{p.lim}_{n\to\infty} k_{s,n}/n_{s,n}, \qquad \bar{d}_s = \operatorname*{p.lim}_{n\to\infty} n_{s,n}/n, \quad s = 1, \ldots, S,$$

with limits depending on δ and on ρ.

See Appendix C for the proof. It is best studied after first becoming familiar with Section 7.

6.4 The Predicted Phase Transition

Fix a small $\eta > 0$; we say that StOMP is *successful*, if at termination of the S-stage StOMP algorithm,

- the active set I_S contains all but a fraction η of the elements of I_0:

$$|I_0 \backslash I_S| \leq \eta |I_0|; \qquad \text{and}$$

- the active set I_S contains no more than n elements:

$$|I_S| \leq (1 - \eta) n.$$

Lemma 3.1 motivates this definition (in the case $\eta = 0$). When this property holds, it is typically the case that $\hat{x}_S \approx x_0$, as experiments have shown.

The existence of large-system limits allows us to derive phase transitions in the 'Success' property; the corresponding curves ρ_{FAR} and ρ_{FDR} decorate Figures 6 and 7. Empirically, these transitions happen at about the same place as apparent transitions for other candidate definitions of 'Success', such as exact equality $\hat{x}_S = x_0$. The key point is that the transitions in this property can be calculated analytically, and are rigorously in force at large n, whereas empirical phase transitions are simply *interpretations*.

This analytic calculation works by tracking the large-system limit variables $(\bar{\rho}_s, \bar{d}_s, \bar{\nu}_s, \bar{\sigma}_s)$ as a function of s; thus we use dimension-normalized units, $1 \leftrightarrow n$, $\rho \leftrightarrow k$, $1/\delta \leftrightarrow N$, and this state vector is initialized to $(\rho, 1, 1/\delta, \rho)$.

The heart of the calculation is an iteration over $s = 1, \ldots, S$. At stage s, we first calculate the model false alarm rate and the model true detect rate:

$$\bar{\alpha}_s = \operatorname*{p.lim}_{n\to\infty} Prob\{|\langle \phi_j, r_s \rangle| > t_s \sigma_{s,n} | j \in I_0^c \cap I_{s-1}^c\}, \tag{6.1}$$

Method	$\delta = .05$	$\delta = .10$	.15	.20	.25	.35	.50	.65	.80
Empirical	0.1250	0.1562	0.1792	0.2000	0.2225	0.2607	0.3212	0.3663	0.3852
Theoretical	0.1247	0.1498	0.1703	0.1869	0.2076	0.2518	0.2913	0.3567	0.4008

Table 6: Empirical and Theoretical Phase Transitions. Comparisons at several values of indeterminacy δ. Top half of table: empirical calculation ($N = 1600$); bottom half: theoretical calculation.

$$\bar{\beta}_s = \operatorname*{p.lim}_{n\to\infty} Prob\{|\langle \phi_j, r_s\rangle| > t_s \sigma_{s,n} | j \in I_0 \cap I_{s-1}^c\}. \tag{6.2}$$

This part of the calculation requires theoretical developments from the next section; specifically Corollaries 7.1,7.2. We then update the limit quantities in the obvious way:

$$\bar{d}_s = \bar{d}_{s-1} - \bar{\beta}_s \bar{\rho}_s - \bar{\alpha}_s(\bar{\nu}_s - \bar{\rho}_s),$$

$$\bar{\rho}_{s+1} = \bar{\rho}_s(1 - \bar{\beta}_s), \qquad \bar{\nu}_{s+1} = \bar{\nu}_s - \bar{\alpha}_s(\bar{\nu}_s - \bar{\rho}_s) + (\bar{\rho}_{s+1} - \bar{\rho}_s).$$

The calculation announces *success* if, at or before stage S,

$$\bar{d}_s \geq \eta, \qquad \bar{\rho}_s \leq \eta\rho.$$

Otherwise, it announces *failure*.

This calculation evaluates a specific parameter combination (δ, ρ) for success or failure. We are really interested in the boundary $\rho_{FAR,S}(\delta)$ which separates the 'success' region from the 'failure' region. By binary search, we obtain a numerical approximation to this boundary.

In this calculation, there is no notion of problem size n; in principle the calculation is applicable to all large problem sizes. The assumption being made is that certain variables (such as the empirical false alarm rate) are intensive, and, though random, can be approximated by a limit quantity. This has been established for the relevant variables by Theorem 1.

Table 6 compares the calculations made by this approach with the results of a StOMP simulation. The degree of match is apparent. The difference between the empirical transition and the theoretical prediction is smaller than the width of the transition; compare Figure 8. Since the empirical transition point is not a sharply defined quantity, the degree of match seems quite acceptable.

7 The Conditioned Gaussian Limit

Underlying Theorem 1 and the subsequent phase-transition calculations is a particular model for the statistical behavior of coefficients $\langle \phi_j, r_s\rangle$. We now introduce and derive that model.

7.1 The Conditioned Gaussian Model

Our model considers the quantities $\langle \phi_j, r_s\rangle$ driving the StOMP algorithm. There are two kind of behaviors: one for $j \notin I_0$ – the *null case* – and one for $j \in I_0$ – the *non-null case*.

7.1.1 Null Case

Define jointly Gaussian random variables $Z_0, Z_1, \ldots, Z_S$, with means zero and variances $\bar{\sigma}_s^2$ defined by Theorem 1. The variances are decreasing: $\bar{\sigma}_s^2 > \bar{\sigma}_{s+1}^2$. The random variables have the covariance structure

$$Cov(Z_u, Z_s) = \bar{\sigma}^2_{\max(u,s)}.$$

That is to say, the process $(Z_s : s = 1, \ldots, S)$ behaves as a *time-reversed martingale*.

Consider the coefficient $\langle \phi_j, r_s\rangle$ obtained by matched filtering of the s-th residual, and suppose that j is a truly null coordinate, i.e. j is not in the support of x_0. For a random variable X let $\mathcal{L}(X)$ denote the probability law of X. Consider the (USE,$\pm$) problem suite with given values of $\delta = n/N$ and $\rho = k/n$, and n large. Our *conditioned Gaussian* model says that, in the CFAR case

$$\mathcal{L}(\langle \phi_j, r_s\rangle | j \notin I_0 \cup I_{s-1}) \approx \mathcal{L}(Z_s | |Z_i| < t\bar{\sigma}_i, i = 1, ..., s-1).$$

In words, we model each null coefficient as a certain Gaussian random variable *conditioned on* certain non-exceedance events involving other, correlated random variables. In particular, we do not model it simply as a Gaussian random variable (except if $s = 1$). To enforce this distinction, we let $\tilde{Z}_s$ denote the random variable Z_s conditioned on $\{|Z_i| < t\bar{\sigma}_i, i = 1, ..., s-1\}$.

7.1.2 Non-Null Case

Define jointly Gaussian random variables X_0,X_1,...,X_S, with means $\bar{\mu}_s$ and variances $\bar{\sigma}_s^2$ again deriving from Theorem 1. There is again the covariance appropriate to a time-reversed martingale:

$$Cov(X_u, X_s) = \bar{\sigma}^2_{\max(u,s)}.$$

Consider now the coefficient $\langle \phi_j, r_s \rangle$ obtained by matched filtering of the s-th residual, where j is a truly *non-null* coordinate, i.e. j is not in the support of x_0. Consider again the standard problem suite with given values of δ and ρ and n large. The conditioned Gaussian model says that

$$\mathcal{L}(\langle \phi_j, r_s \rangle | j \in I_0 \cap I^c_{s-1}) \approx \mathcal{L}(X_s || X_i| < t\bar{\sigma}_i, i = 1, ..., s-1).$$

In words, we model each non-null coefficient at stage s as a certain nonzero-mean Gaussian random variable conditioned on a certain sequence of non-exceedances at earlier stages in the sequence. In this case, the conditioning event looks the same as in the non-null case; however the random variables X_i do not have mean zero. We let $\tilde{X}_s$ denote the random variable X_s conditioned on $\{|X_i| < t\bar{\sigma}_i, i = 1, ..., s-1\}$.

7.1.3 The Gaussian Approximation

The CG model, which will later be seen to be highly accurate, explains why the Gaussian approximation sometimes works. The model has the following consequence. Let $p_{\tilde{Z}_s}(z)$ denote the marginal probability density of the CG variable $\tilde{Z}_s$ and let $g_{\bar{\sigma}_s}(z)$ denote the probability density of a Gaussian $N(0, \sigma_s^2)$. Under a simple normal approxmation, we would have $p_{\tilde{Z}_s}(z) \approx g_{\bar{\sigma}_s}(z)$. Under our model,

$$p_s(z) = \frac{Prob\{|Z_1| < t\bar{\sigma}_1, \ldots, |Z_{s-1}| < t\bar{\sigma}_{s-1} | Z_s = z\} g_{\bar{\sigma}_s}(z)}{Prob\{|Z_1| < t\bar{\sigma}_1, \ldots, |Z_{s-1}| < t\bar{\sigma}_{s-1}\}}.$$

We have the identity $p_{\tilde{Z}_s}(z) = \lambda_s(z) g_{\bar{\sigma}_s}(z)$, where

$$\lambda_s(z) = \frac{Prob\{|Z_1| < t\bar{\sigma}_1, \ldots, |Z_{s-1}| < t\bar{\sigma}_{s-1} | Z_s = z\}}{Prob\{|Z_1| < t\bar{\sigma}_1, \ldots, |Z_{s-1}| < t\bar{\sigma}_{s-1}\}}.$$

A parallel definition for the random variables $\tilde{X}_s$ sets

$$\xi_s(x) = p_{\tilde{X}_s}(x) / p_{X_s}(x).$$

In Figure 9 Panel (a) we display exact results for λ_s under our model, with a choice of $\bar{\sigma}$ obtained from analyzing the case $\delta = .2$, $\rho = .2$. As one can see, each λ_s is effectively 1 near zero, and drops to zero in the tails. In this case, each underlying $\bar{\sigma}_s$ is small and each $g_{\bar{\sigma}_s}$ is effectively concentrated over the region where λ_s is nearly 1. Hence the Gaussian approximation to the conditioned Gaussian model is not bad, for the parameters ρ and δ underlying this situation. Panel (b) depicts ξ_s with a choice of $\bar{\mu}$, $\bar{\tau}$ obtained from analyzing the case $\delta = .2$, $\rho = .2$. Now we have only a vaguely Gaussian shape.

7.2 Derivation of the Model

The first part of this section will prove:

Theorem 2 *Let $\tilde{Z}_s$ be as defined in Section 7.1.1. Then, for $w \in \mathbf{R}$,*

$$P\{\langle \phi_j, r_s \rangle \le w |\ j \notin I_0\ \&\ j \notin I_{s-1}\} \to P\{\tilde{Z}_s \le w\}, \quad n \to \infty, \quad s = 1, \ldots, S.$$

We immediately gain a formula for computing the limiting threshold false-alarm probability:

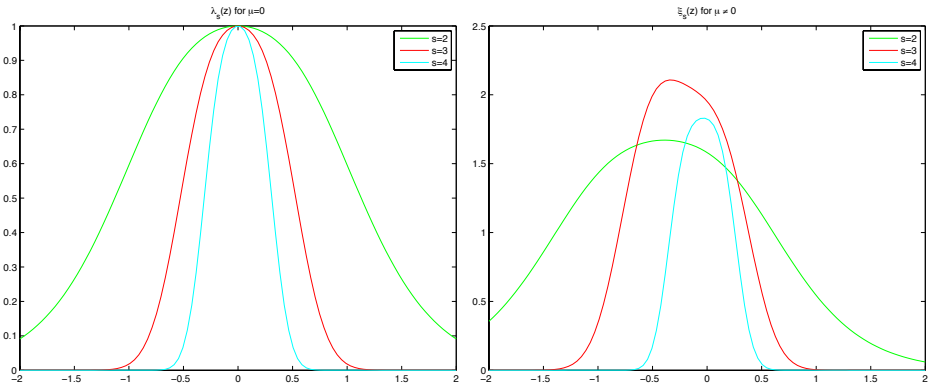

Figure 9: Density Correction Factors. (a) $\lambda_s(z)$, the factor multiplying the standard normal density to get the conditioned Gaussian density, null case. (b) $\xi_s(z)$, the factor multiplying the (non-standard) normal density to get the conditioned Gaussian density, nonnull case. Here $s = 2, 3, 4$ $\delta = 1/4$ and $\rho = .2$.

Corollary 7.1

$$\bar{\alpha}_s = P\{|\tilde{Z}_s| \geq t_s \bar{\sigma}_s\}. \tag{7.3}$$

The comparable result in the Non-null case is:

Theorem 3 *Let $\tilde{X}_s$ be as defined in Section 7.1.1. Then*

$$P\{\langle \phi_j, r_s \rangle \leq w |\ j \in I_0 \ \& \ j \notin I_{s-1}\} \to P\{\tilde{X}_s \leq w\}, \quad n \to \infty.$$

We obtain a formula for computing the limiting threshold correct detection rate:

Corollary 7.2

$$\bar{\beta}_s = P\{|\tilde{X}_s| \geq t_s \bar{\sigma}_s\} \tag{7.4}$$

The formulas (7.3) and (7.4) explain how to perform in principle the calculations (6.1)-(6.2) needed for calculating phase transitions in Section 6.4. For complete documentation of the calculation procedure, see Section 10.

7.2.1 Null Case

Suppose we are given a *deterministic* vector $v_0 \in \mathbf{R}^n$ and a sequence of deterministic orthoprojectors $Q_1, Q_2, Q_3, \ldots$, where $Q_1 = Id$. We are given a *random* vector ψ_0 Gaussian distributed with mean zero and diagonal covariance matrix $\frac{1}{n} I_n$. Define $v_i = Q_i v_0$.

Lemma 7.1 *Define random variables $Z_s = \langle \psi_0, v_s \rangle$, $s = 1, \ldots, S$. Then $(Z_s, s = 1, \ldots, S)$ is a jointly Gaussian sequence, with mean zero, variances $\sigma_s^2 = \|v_s\|_2^2/n$ and covariances $Cov(Z_s, Z_u) = \sigma^2_{\max(s,u)}$.*

This is self-evident, and we omit the proof.

Now introduce the random variable

$$\phi_0 = \psi_0 / \|\psi_0\|_2. \tag{7.5}$$

Of course ϕ_0 is a random point on $\mathbf{S}^{n-1}$, in fact uniformly distributed.

Lemma 7.2 *Define random variables $W_s = \langle \phi_0, v_s \rangle$, $s = 1, \ldots, S$. For fixed S, the sequence $(W_s, s = 1, \ldots, S)$ is asymptotically well-approximated by a joint Gaussian sequence (Z_s), with variances $\sigma_s^2 = \|v_s\|_2^2/n$ and covariances $Cov(Z_s, Z_u) = \sigma^2_{\max(s,u)}$. More precisely, for a sequence ϵ_n depending on n only, $E(W_s - Z_s)^2/\sigma_s^2 \leq \epsilon_n \to 0$, $n \to \infty$.*

Proof. Of course, the Gaussian sequence we have in mind is just (Z_s) introduced above, linked to W_s by (7.5). Then $W_s - Z_s = Z_s(\|\psi_0\|_2^{-1} - 1)$. Now $\sqrt{n} \cdot \|\psi_0\|_2$ has a χ_n distribution, so that $\|\psi_0\|_1$ converges in probability to 1 as $n \to \infty$. In fact, using the analytic expression for the χ_n^2 density in terms of beta functions, one can show that $E(\|\psi_0\|_2 - 1)^4 \to 0$. Moreover $EZ_s^4 = 3\bar{\sigma}_s^4$. Hence

$$E(W_s - Z_s)^2 = EZ_s^2(\|\psi_0\|_2^{-1} - 1)^2 \le (EZ_s^4)^{1/2} \cdot (E(\|\psi_0\|_2^{-1} - 1)^4)^{1/2} \to 0, \quad n \to \infty.$$

Putting $\epsilon_n = (3E(\|\psi_0\|_2^{-1} - 1)^4)^{1/2}$ gives the claimed result. □

It is useful to restate the last lemma. Let $H_{1,\ldots,s;n}(w_1, \ldots, w_s; \sigma)$ denote the joint distribution function of $W_1,\ldots,W_s$, conditional on $\sigma_1 = \|v_1\|_2/\sqrt{n},\ldots,\ \sigma_s = \|v_s\|_2/\sqrt{n}$. Let $G_{1,\ldots,s;n}(w_1, \ldots, w_s; \sigma)$ denote the joint distribution function of $Z_1,\ldots,Z_s$, again conditional on $\sigma_1 = \|v_1\|_2/\sqrt{n},\ldots,\ \sigma_s = \|v_s\|_2/\sqrt{n}$. Then the last lemma implies

$$H_{1,\ldots,s;n}(w_1, \ldots, w_s; \sigma) \to G_{1,\ldots,s;n}(w_1, \ldots, w_s; \sigma), \qquad n \to \infty.$$

However, a certain degree of uniformity is present, which will be important for us. In the following result, $\sigma = (\sigma_1, \ldots, \sigma_s)$, and $\sigma > 0$ means $\sigma_1 > 0$, $\sigma_2 > 0$, $\ldots$, $\sigma_s > 0$.

Lemma 7.3 *The family of functions $H_{1,\ldots,s;n}(w_1, \ldots, w_s; \sigma)$ is uniformly continuous in w, uniformly in $n > n_0$, locally uniformly at each $\sigma > 0$. The family $G_{1,\ldots,s}(w_1, \ldots, w_s; \sigma)$ is uniformly continuous in w and locally uniformly continuous in σ at each $\sigma > 0$.*

The result is a tedious exercise using standard ideas and we omit the proof.

Suppose that we have a sample (y, Φ) from the standard problem suite and that the random variable ϕ_0 introduced above is stochastically independent of (y, Φ). Suppose that StOMP has been run through $s - 1$ stages. Recall the values y, r_1, r_2 etc. produced by the StOMP algorithm, and condition on those results, defining $v_0 = y$, $v_1 = r_1$, etc. As ϕ_0 is a random point on $\mathbf{S}^{n-1}$ but not a column in the matrix Φ, the probability distribution of $\langle \phi_0, r_s \rangle$, *conditional on y, r_1, ...* is exactly that of W_s above, with parameters $\sigma_1 = \|y\|_1/\sqrt{n}$, $\sigma_2 = \|r_2\|_2/\sqrt{n}$, etc. Now let $p_{s,n}(\sigma)$ denote the probability density of $\sigma = (\sigma_1, \ldots, \sigma_s)$ induced by StOMP , and let $P_{s,n}$ denote the corresponding probability measure. Let $F_{1,\ldots,s;n}$ denote the joint cumulative distribution function of the random variables $\langle \phi_0, y \rangle$, $\langle \phi_0, r_2 \rangle$, $\ldots$, $\langle \phi_0, r_s \rangle$. Then we have the exact formula

$$F_{1,\ldots,s;n}(w_1, \ldots, w_s) = \int H_{1,\ldots,s;n}(w_1, \ldots, w_s; \sigma) p_{s,n}(\sigma) d\sigma. \tag{7.6}$$

Now by Theorem 1 there exist constants $\bar{\sigma}_s$ so that, for $\epsilon > 0$,

$$P\{|\bar{\sigma}_1 - \|y\|_2/\sqrt{n}| < \epsilon, |\bar{\sigma}_s - \|r_s\|_2/\sqrt{n}| < \epsilon, s = 2, \ldots, S\} \to 1. \tag{7.7}$$

Combining this with the uniform equicontinuity of Lemma 7.3 and the scale mixture identity (7.6), we conclude that

$$F_{1,\ldots,s;n}(w_1, \ldots, w_s) \to G_{1,\ldots,s;n}(w_1, \ldots, w_s; \bar{\sigma}), \qquad n \to \infty. \tag{7.8}$$

Of course ϕ_0 is of no interest to us per se. Consider now a column ϕ_j from Φ, and suppose that j is both a null coordinate – $j \notin I_0$ – and a surviving coordinate at stage s – $j \notin I_{s-1}$. It might seem that $\langle \phi_j, r_s \rangle$ would have the same distribution as $\langle \phi_0, r_s \rangle$ but this is only true for $s = 1$. At later stages $s > 1$ of the algorithm, $\langle \phi_j, r_s \rangle$ behaves as W_s subjected to conditioning:

$$\mathcal{L}(\langle \phi_j, r_s \rangle | j \notin I_0 \cup I_{s-1}) = \mathcal{L}(\langle \phi_0, r_s \rangle | |\langle \phi_0, r_i \rangle| < t\sigma_i, i = 1, \ldots, s-1) \tag{7.9}$$

We now make the observation that probabilities of hyper-rectangles can be computed simply from the joint cumulative distribution function. We state without proof an elementary fact:

Lemma 7.4 *Let $U_1,\ldots,U_s$ denote random variables with joint cumulative distribution function $H_{1,\ldots,s}(u_1, \ldots, u_s)$. The rectangle probability $R_{1,\ldots,s}(u_1, \ldots, u_s; H_{1,\ldots,s}) \equiv Prob\{|U_i| < u_i, i = 1, \ldots, s\}$ can be expressed as a linear combination*

$$R_{1,\ldots,s}(u_1, \ldots, u_s; H_{1,\ldots,s}) = \sum_{\pm_i} c_{1,\ldots,s}(\pm_1, \ldots, \pm_s) H_{1,\ldots,s}(\pm_1 u_1, \ldots, \pm_s u_s),$$

with coefficients $c_{1,\ldots,s}(\pm_1,\ldots,\pm_s)$. *The rectangle probability* $Q^s_{1,\ldots,s-1}(u_1,\ldots,u_s) \equiv Prob\{U_s \le u_s, |U_i| < u_i, i = 1,\ldots,s-1\}$ *similarly has a representation*

$$Q^s_{1,\ldots,s-1}(u_1,\ldots,u_s; H_{1,\ldots,s}) = \sum_{\pm_i} c^s_{1,\ldots,s-1}(\pm_1,\ldots,\pm_s) H_{1,\ldots,s}(\pm_1 u_1,\ldots,\pm_{s-1}u_{s-1}, u_s).$$

It follows that, if we have a sequence $H_{1,\ldots,s;n}$ of such CDF's converging uniformly to a limit CDF $H_{1,\ldots,n}$, then we also have convergence of the corresponding rectangle probabilities just mentioned.

A conditional probability is a ratio of two such terms:

$$Prob\{Z_s \le w | |Z_i| < w_i, i = 1,\ldots,s\} = \frac{Q^s_{1,\ldots,s-1}(w_1,\ldots,w_{s-1},w; G_{1,\ldots,s})}{R_{1,\ldots,s-1}(w_1,\ldots,w_{s-1}; G_{1,\ldots,s-1})}$$

The probability law given on the right-hand side of (7.9) has cumulative distribution function

$$\tilde{F}_{1,\ldots,s;n}(w) = Prob\{\langle \phi_0, r_s\rangle \le w | |\langle \phi_0, r_i\rangle| < t\sigma_i, i = 1,\ldots,s-1\}$$

Invoking Lemmas 7.4 and 7.3, as well as (7.7), we get

$$\begin{aligned}
\tilde{F}_{1,\ldots,s;n}(w) &= \int \frac{Q^s_{1,\ldots,s-1}(t\sigma_1,\ldots,t\sigma_{s-1},w; G_{1,\ldots,s;n}(\cdot;\sigma))}{R^s_{1,\ldots,s-1}(t\sigma_1,\ldots,t\sigma_{s-1}; G_{1,\ldots,s-1;n}(\cdot;\sigma))} p_{s,n}(\sigma)d\sigma \\
&\to \frac{Q^s_{1,\ldots,s-1}(t\bar{\sigma}_1,\ldots,t\bar{\sigma}_{s-1},w; G_{1,\ldots,s}(\cdot;\bar{\sigma}))}{R^s_{1,\ldots,s-1}(t\bar{\sigma}_1,\ldots,t\bar{\sigma}_{s-1}; G_{1,\ldots,s-1}(\cdot;\bar{\sigma}))}, \qquad n \to \infty, \\
&= Prob\{Z_s \le w | |Z_i| < t\bar{\sigma}_i, i = 1,\ldots,s-1\}.
\end{aligned}$$

The middle step invoked the fact that, in the sense of convergence in probability, $G_{1,\ldots,s;n}(\cdot;\sigma) \to_P G_{1,\ldots,s}(\cdot;\bar{\sigma})$ in uniform norm, locally uniformly in $\sigma > 0$.

7.2.2 Non-null Case

The technical side of the argument parallels the null case, and we will not repeat it. The only point we clarify here is the calculation of the means μ_s and standard deviations τ_s.

For this calculation, we propose to model y as a $\sum \pm_i \psi_i$, where the $\pm_i$ are arbitrary signs, and ψ_i are Gaussian random vectors. This model corresponds to 'Gaussianizing' the SSP instance (y, Φ)

A vector v uniformly distributed on the unit sphere in $\mathbf{R}^n$ is Gaussianized by multiplying it by an independent scalar random variable $n^{-1/2}\chi_n$ where χ_n is Chi-distributed on n degrees of freedom. The resulting vector $n^{-1/2}\chi_n \cdot v$ is distributed $N(0, I_n)$.

Now apply such Gaussianization independently to each of the columns of Φ, producing the columns of a matrix Ψ, the vector $y = \Psi x_0$ has indeed the distribution of $\sum \pm_i \psi_i$. We will make some computations using this Gaussianization; the result, exactly true in the Gaussian case, is asymptotically correct for the original pre-Gaussianization model. The same approach was used, less explicitly, in the last subsection. Gaussianization has also been heavily used in the Banach space literature; see also [16, 17] for examples in the present spirit.

We start with a typical Bayesian calculation.

Lemma 7.5 *Suppose that* $\psi_1,\ldots,\psi_k$ *are Gaussian vectors in* $\mathbf{R}^n$ *distributed* $N(0, \frac{1}{n}I_n)$. *Suppose that* $y = \sum_1^k \psi_i$. *Given* y, ψ_1 *has a Gaussian conditional distribution:*

$$\mathcal{L}(\psi_1|y) = N(y/k, \frac{k-1}{k}\frac{1}{n}I_n).$$

We omit the proof of this well-known fact. Consider now a *deterministic* vector v_0 and *deterministic* orthoprojectors Q_1, Q_2, ..., Q_S, yielding vectors $v_i = Q_i v_0 \in \mathbf{R}^n$. Because projections of Gaussians are Gaussian and linear combinations of Gaussians are Gaussian, we immediately obtain:

Lemma 7.6 *Let* ψ_0 *be a random vector in* $\mathbf{R}^n$ *with Gaussian distribution* $N(v_0/k, \frac{k-1}{k}\frac{1}{n}I_n)$. *Define random variables* $X_s = \langle \psi_0, v_s\rangle$. *Their marginal distribution is precisely*

$$\mathcal{L}(X_s) = N(\|v_s\|_2^2/k, \frac{k-1}{k}\frac{1}{n}\|v_s\|_2^2).$$

We again omit the elementary proof. Consider now $\phi_0 = \psi_0/\|\psi_0\|_2$. In parallel with Lemma 7.2 we have:

Lemma 7.7 *Define the family of random variables $V_s = \langle \phi_0, v_s \rangle$, $s = 1, \ldots, S$. This family is well approximated by the Gaussian random variables X_s defined above. In fact, for a sequence ϵ_n depending only on n, $E(X_s - V_s)^2/Var(V_s) \le \epsilon_n \to 0$.*

Clearly, the above elements can be combined to give our result, in much the same fashion that used in the null case can be carried out in the present case. Let

$$\tilde{H}_{1,\ldots,s;n}(w) = Prob\{\langle \phi_0, r_s \rangle \le w | |\langle \phi_0, r_i \rangle| < t\sigma_i, i = 1, \ldots, s-1\}.$$

Define Gaussian random variables $\bar{X}_s$ with mean $\bar{\mu}_s$ and variance $\bar{\sigma}_s^2$. Let $\tilde{X}_s$ denote the random variable $\bar{X}_s$ conditional on the event $\{|\bar{X}_1| \le t\bar{\sigma}_1, \ldots, |\bar{X}_{s-1}| \le t\bar{\sigma}_{s-1}\}$. By the same approach as in the null case we obtain:

$$\tilde{H}_{1,\ldots,s;n}(w) \to P\{\tilde{X}_s \le w\}, \quad n \to \infty.$$

$$\bar{\mu}_s = \operatorname*{p.lim}_{n\to\infty} \|r_{s,n}\|_2^2/k_{s,n}, \quad \bar{\sigma}_s^2 = \operatorname*{p.lim}_{n\to\infty} \frac{k_{s,n}-1}{k_{s,n}} \cdot \|r_{s,n}\|_2^2/n.$$

Here of course the presence of the factor $\frac{k_{s,n}-1}{k_{s,n}}$ does not affect the limit, as $k_{s,n}$ will eventually be large with overwhelming probability. This completes our proof of Theorem 3. □

8 Variations

8.1 How Many Stages?

In the experiments reported here, we chose $S = 10$ stages. Our main consideration in choosing the number of stages is the speed of the resulting algorithm. Obviously, choosing S smaller or larger would modify the speed and modify the phase transition diagram, and so give us a larger or smaller range of effectiveness. Because we make available the code that performed our experiments (see Section 10), it is straightforward to study variations on the procedure described here.

8.2 Varying Coefficient distributions

The phase transitions displayed in Section 4 were computed assuming the nonzero coefficients have a Gaussian distribution. The phase transitions in Section 6 assumed the nonzero coefficients in x_0 have a symmetric distribution on $\{\pm 1\}$. There are small differences, with the Gaussian coefficients leading to transitions at higher values of ρ. We have of course tried other distributions as well. Experiments in Figure 10, Panel (a) show the case of a uniform distribution on the coefficients, while Figure 10, Panel (b) illustrates the power law case. We conjecture that, among coefficient distributions, the worst phase transition is approximately given by the sign case, where we have worked to give a rigorous theory.

8.3 Noisy Data

The methods discussed above extend quite naturally to the case of data contaminated with white Gaussian noise. Indeed, suppose that our observations y obey

$$y = \Phi x + \xi$$

where ξ denotes an iid $N(0, \eta^2)$ noise. The matched filter will obey the same conditioned normal approximation, with different variances. Hence, to the extent that our approach was applicable before, it remains applicable. We remark, however, that CFDR seems most appropriate in the noisy case.

Figure 11 displays the performance of CFDR thresholding in the noisy case. The transition behavior is less clear-cut than in the noiseless case. It seems to indicate graceful smoothing out of the sharp transition seen in the noiseless case.

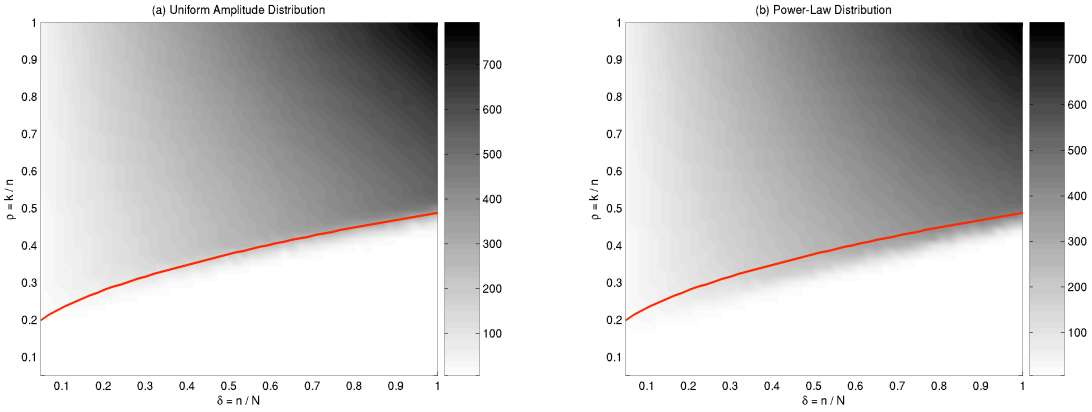

Figure 10: Phase Diagrams for CFAR thresholding when the nonzeros have nonGaussian distributions. Panel (a) uniform amplitude distribution; Panel (b) Power law distribution $x_0(j) = c/j$, $j = 1, \ldots, k$. Compare to Figure 6. Shaded attribute: number of entries where reconstruction misses by 10^{-4}.

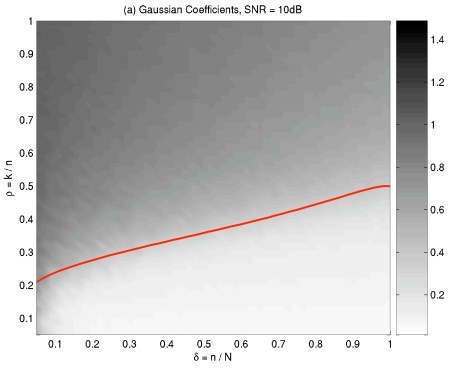

Figure 11: Performance of CFDR thresholding, noisy case. Relative ℓ_2 error as a function of indeterminacy and sparsity. Performance at signal-to-noise ratio 10. Shaded attribute gives relative ℓ_2 error of reconstruction. Signal-to-Noise ratio is $\|x_0\|_2/\|\Phi^T\xi\|_2$.

8.4 Other Matrix Ensembles

Our attention has focused on the case where Φ is a random matrix, generated from the uniform spherical ensemble. Similar results follow immediately for two closely related ensembles:

URPE *Uniform Random Projection* ensemble. Φ contains the first n rows of an N by N random orthogonal matrix [8]; and

GE *Gaussian* ensemble. The entries of Φ are iid $N(0, 1/n)$.

In fact we have already used (more than once) the fact that GE and USE are intimately related. Matrices in the two ensembles differ only by the normalization of the columns – a member of URP can be obtained by sampling from the Gaussian ensemble and normalizing the columns to unit length. Also, the close relationship of URPE and GE is quite evident by viewing one as produced from the other by a Gram-Schmidt process on the rows. Figure 3 Panels(c),(f), and (i) show that for the URPE, the MAI for matched filtering obeys the Gaussian approximation. Extensive experiments have shown that StOMP has the same behavior at the URPE, GE, and USE, but we omit details for reasons of space. Scripts generating such experiments are included in the software publication; see Section 10.

More interestingly, we considered other random ensembles, the most well-known ones being

- *Random Signs ensemble.* The entries of the matrix are $\pm 1/\sqrt{n}$, the signs chosen randomly.
- *Partial Fourier ensemble.* n rows are chosen at random from an N by N Fourier matrix.
- *Partial Hadamard ensemble.* n rows are chosen at random from an N by N Hadamard matrix. (Possible only for certain N).

These are important for various applications of compressed sensing. For each ensemble, we found that the Gaussian approximation applies. Figure 3 Panels(b),(e), and (h) illustrate the MAI for matched filtering at the RSE. Thus, we propose StOMP for such ensembles as well.

9 Stylized Applications

We now illustrate the performance of StOMP and the thresholding strategies.

9.1 Compressed Sensing

Recently, there has been considerable interest both from theorists [33, 7, 17, 8] and from practitioners [42, 47, 31, 40, 41] in the possibility of dramatically reducing the 'number of samples' that 'have to be measured' in various remote sensing problems. In effect, one views the problem as one of reconstructing a high-dimensional vector $x_0 \in \mathbf{R}^N$ from a low-dimensional data sample $y \in \mathbf{R}^n$, which is obtained from x_0 by linear sampling. Here although N samples 'seem to be needed' according to standard linear algebra, everything we have shown in this paper (as well as the cited prior work) shows that $n < N$ samples can suffice to get either an exact or approximate reconstruction of x_0.

We now study the performance of StOMP and the thresholding strategies in concrete instances, inspired by applications in spectroscopy and imaging.

9.1.1 Bumps

Our first example uses the object *Bumps* from the Wavelab package [5], rendered with $N = 4096$ samples. This object, shown in panel (a) of Figure 12, is a caricature of signals arising in NMR spectroscopy, characterized by a few localized peaks. Such signals are known to have wavelet expansions with relatively few significant coefficients. We considered a Compressed Sensing (CS) scenario where $n_{CS} = 640$ sensed samples are taken, reflecting random linear combinations of the wavelet coefficients of *Bumps*. The details are the same as for hybrid CS in [58]. In our simulations, we compared the performance of StOMP equipped with CFDR and CFAR thresholding to that of Basis Pursuit (i.e. ℓ_1 minimization) and Matching Pursuit (i.e. OMP). The results are summarized in Figure 12. Evidently, the accuracy of reconstruction is comparable for all the algorithms used. However, the speed of the two StOMP implementations is unrivaled

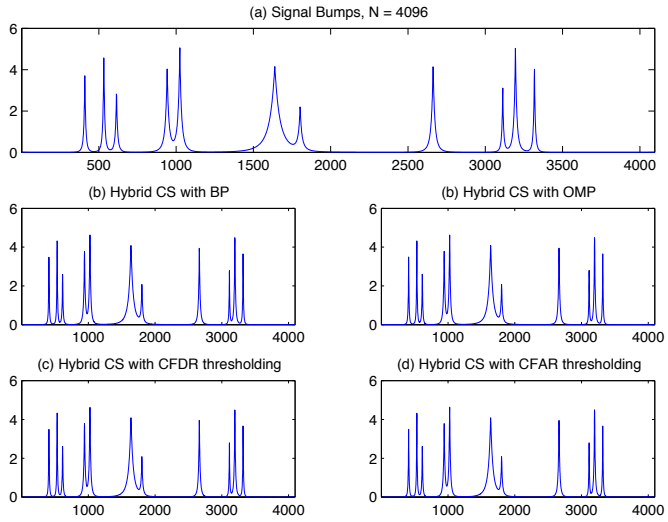

Figure 12: Reconstruction of *Bumps* with hybrid CS. Panel (a): Signal *Bumps*, with 4096 samples; Panel (b): Reconstruction with BP (ℓ_1), $\|x_{BP} - x_0\|_2/\|x_0\|_2 = 0.037$, $t_{BP} = 1062$ sec.; Panel (c): Reconstruction with OMP, $\|x_{OMP} - x\|_2/\|x_0\|_2 = 0.037$, $t_{OMP} = 37$ sec.; Panel (d): Reconstruction with CFDR, $\|x_{CFDR} - x\|_2/\|x_0\|_2 = 0.037$, $t_{CFDR} = 2.8$ sec.; Panel (e): Reconstruction with CFAR, $\|x_{CFAR} - x\|_2/\|x_0\|_2 = 0.037$, $t_{CFAR} = 2.6$ sec.

by BP or OMP; compare the 2.6 seconds required by StOMP with CFAR to generate a solution, with the 37 seconds needed by OMP, or nearly 18 minutes of computation time entailed by ℓ_1 minimization. As for the results appearing in Table 1, all the simulations described in this section were obtained on a 3GHz Xeon workstation.

We now consider a larger-scale example in 2 dimensions. Figure 13(a) shows a synthesized image of 2-D Gaussian spectra, created in the following manner. 40 Gaussians were generated with standard deviations randomly varying, amplitudes drawn from an exponential distribution, and positions i.i.d uniform. The image is discretized on a grid of 256×256 pixels. We applied multiscale CS as in [58]. We used a wavelet basis and formed a matrix Φ which gathered random linear combinations of the coefficients in the 3 finest wavelet scales (see [58] for details). The total number of sensed samples was $n_{CS} = 13004$ (here the number of pixels $N = 256^2 = 65536$). Figure 13, panels (b)-(d), present reconstruction results for BP, CFDR and CFAR, respectively. (We did not consider OMP in our simulations; it seems impractical to apply in such large-scale applications, due to memory constraints.) All 3 algorithms produced faithful reconstructions. However, careful investigation of the error and timing measurements reveals that CFDR and CFAR outperformed BP in terms of both speed and accuracy.

9.1.2 Mondrian

We now consider a 'geometric' example. Panel (a) of Figure 14 displays a photograph of a painting by Piet Mondrian, the Dutch neo-plasticist. This image has a relatively sparse expansion in a tensor wavelet basis, and therefore is suitable for CS. (This test image, while of a relatively simple geometric nature, still presents a challenging trial, as its wavelet expansion is not *very* sparse. In fact, out of 262144 wavelet coefficients, there are only 798 coefficients with magnitude smaller than 10^{-2}.) More 'naturalistic' images would be equally fitting candidates, provided they admit sparse representations in an appropriate basis/frame (such as the Curvelets frame, for instance).

Much as with the 2-D *Bumps* image, we used the Mondrian image in a Multiscale CS setting, applied to the 4 finest scales of the wavelet expansion. A total of $n_{CS} = 69834$ sensed samples were used overall. Since $N = 512^2 = 262144$, this stands for roughly one quarter of the original dimension of the data. Figure 14, panels (b)-(d) have reconstruction results. Indeed, all three algorithms performed well in terms

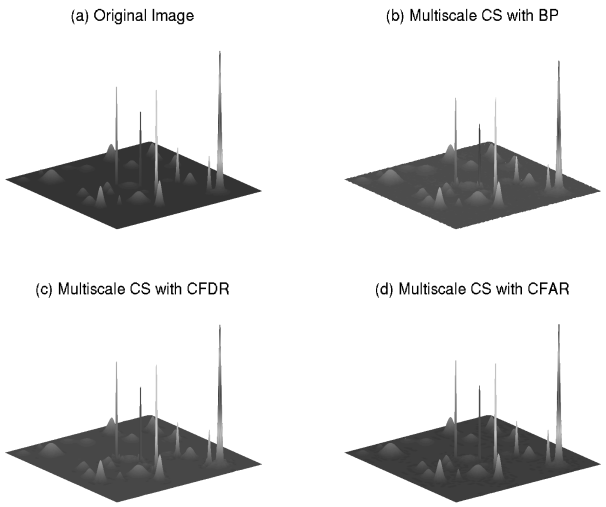

Figure 13: Reconstruction of 2-D *Bumps* with Multiscale CS. Panel (a): 2-D Gaussian spectra, on a 256×256 grid; Panel (b): Reconstruction with BP (ℓ_1), $\|x_{BP} - x_0\|_2/\|x_0\|_2 = 0.13$, $t_{BP} = 40$ sec.; Panel (c): Reconstruction with CFDR, $\|x_{CFDR} - x_0\|_2/\|x_0\|_2 = 0.12$, $t_{CFDR} = 16$ sec.; Panel (d): Reconstruction with CFAR, $\|x_{CFAR} - x_0\|_2/\|x_0\|_2 = 0.05$, $t_{CFAR} = 32$ sec.

of reconstruction accuracy. Of the three, ℓ_1 minimization produced the most accurate reconstruction, as measured by ℓ_2 distance to the original. It did so, however, at an outstanding cost; over 30 hours of computation were required by BP (with the PDCO solver [49]) to reach a solution. In contrast, StOMP with CFAR produced a result of comparable accuracy in just over a minute. (Observant readers may notice that here data size is comparable to the 2-D spectra example, but the computation time required by direct ℓ_1 minimization is significantly larger. The cause is our specific implementation of BP. The primal-dual barrier method favors solution vectors which contain many exact zeros.)

We find it instructive to take a closer look at the reconstruction results in the wavelet domain. To that end, we zoomed in on a horizontal slice of 100 wavelet coefficients at one scale below the finest, as displayed in panel (a) of 15. Comparing the reconstruction results of the iterative thresholding algorithms with the original slice of coefficients reveals a great deal about their performance when the underlying signal is not sufficiently sparse. Both CFDR and CFAR successfully recovered the significant coefficients, while keeping the rest of the coefficients at zero. In fact, it makes sense to view the small coefficients as the result of digitization noise, in which case the thresholding algorithms are actually removing noise, while remaining faithful to the original signal. In contrast, ℓ_1 minimization tends to exacerbate the noise under insufficient sparsity conditions, as was discussed in detail in [58]. In short, StOMP is a dependable choice even beyond its region of success.

9.2 Error Correction

Virtually every digital communication system employs error-control coding as an integral part of its operation. There is elegant coding theory showing showing how to encode n items in a block of N transmitted numbers with the ability to correct up to k arbitrary errors; unfortunately for general linear coding schemes the task of identifying the k most likely sites for the errors is known to be NP-hard [3]. Lately, there has been much interest in developing good fast decoding schemes. The literature in IEEE Transactions on Information Theory on message passing decoding and turbo decoding is literally too volumnious to charcterize. Recently, Candès and Tao pointed out that ℓ_1-minimization/sparsity ideas have a role to play in decoding linear error-correcting codes over $\mathbf{Z}$ [8, 48]. Naturally, it follows that StOMP also has an opportunity to contribute.

Here is the experimental *mise en place*. Assume θ is a digital signal of length n, with entries ± 1,

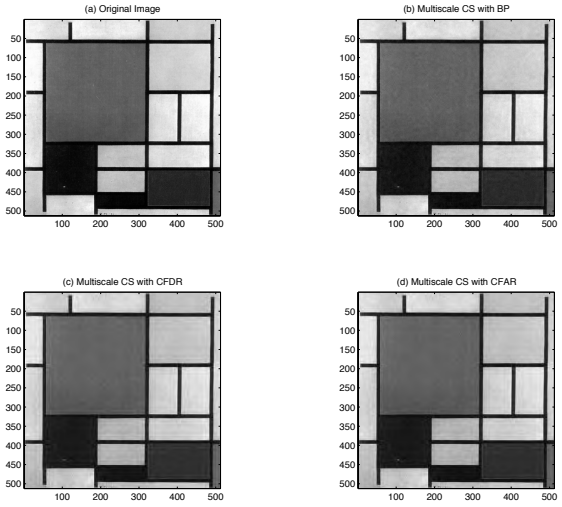

Figure 14: Reconstruction of 2-D imagery with Multiscale CS. Panel (a): *Mondrian* painting, 512×512 pixels; Panel (b): Reconstruction with BP (ℓ_1), $\|x_{BP} - x\|_2/\|x\|_2 = 0.32$, t_{BP} 30 hours.; Panel (c): Reconstruction with CFDR, $\|x_{CFDR} - x\|_2/\|x\|_2 = 0.42$, $t_{CFDR} = 16$ sec.; Panel (d): Reconstruction with CFAR, $\|x_{CFAR} - x\|_2/\|x\|_2 = 0.36$, $t_{CFAR} = 64$ sec.

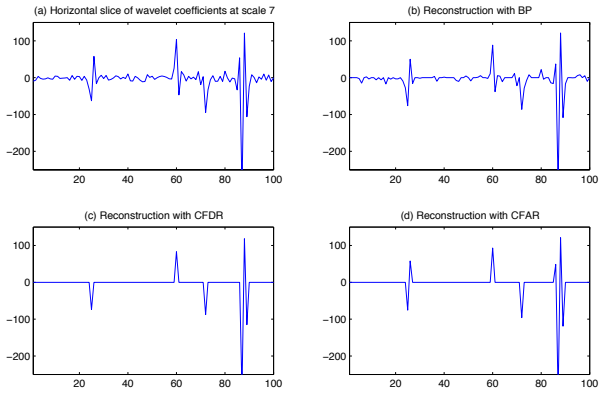

Figure 15: Zoom in on a horizontal slice of wavelet coefficients. Panel (a): Original horizontal slice of coefficients, at scale 7; Panel (b): Reconstruction with BP; Panel (c): Reconstruction with CFDR; Panel (d): Reconstruction with CFAR.

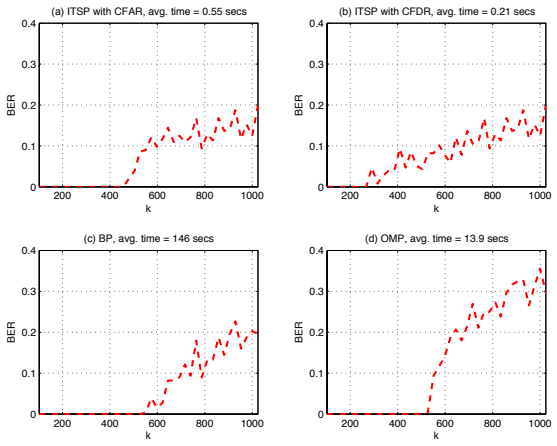

Figure 16: Performance of decoders with varying noise sparsity. (a) CFAR thresholding; (b) CFDR thresholding; (c) Basis Pursuit; (d) OMP. Vertical axes: Bitwise Error Rate. Horizontal axes: Number of Errors. Coding Redundancy: 3.

representing bits to be transmitted over a digital communication channel. Prior to transmission, we encode θ with an ECC constructed in the following manner. Let Q be a 'random' orthogonal matrix of size $R \cdot n \times R \cdot n$, where R is the redundancy factor of the code. Partition $Q = \left[E\ D^T\right]$, where E is the $Rn \times n$ 'encoding matrix' and D is the $(R-1)n \times Rn$ 'decoding matrix'. Then, the encoding stage amounts to computing $x = E\theta$, with x the encoded signal, of length Rn. At the decoder, we receive $r = x + z$, where z has nonzeros in k random positions, and the nonzero amplitudes are Gaussian iid $N(0, \sigma_n)$. In words, the signal is sent over a sparse noisy channel. There are no assumed bounds on noise power. The minimum ℓ_1-norm decoder solves

$$(EC_1) \qquad \min \|\alpha\|_1 \text{ subject to } D\alpha = Dr;$$

Call the solution $\hat{\alpha}$. At the output of the decoder, we compute

$$\hat{\theta} = \text{sgn}(E^T(r - \hat{\alpha})).$$

The key property being exploited is the mutual orthogonality of D and E. Specifically, note that $Dr = D(E\theta + z) = Dz$. Hence, (EC_1) is essentially solving for the sparse error patten.

Figure 16 presents results of tests at redundancy $R = 4$ and decoded data length $n = 256$. We performed each experiment multiple times while varying the noise sparsity k. At each instance, we recorded the Bitwise error rate (BER) and the decoding time. For comparative purposes, we repeated the experiment using Basis Pursuit and OMP. The results are summarized in plots showing BER as a function of noise sparsity; see panels (a)-(d) of Figure 16. In terms of BER, BP prevailed, decoding successfully even when gross errors contaminated more than half the received signal values. Yet, StOMP with CFAR thresholding came remarkably close to the performance of true ℓ_1 minimization. And it did so in a fraction of the time needed by BP; compare the average decoding time of 0.55 seconds required by StOMP to 146 seconds needed by BP.

Actually StOMP can outperform ℓ_1 decoding in terms of error-resistance at very small δ. Consider Figure 17. It presents the results of a similar experiment, in a slightly different setting. Here we set $n = 4096$ and $R = 17/16$, i.e. we choose a long blocklength code with very little redundancy. The phase diagram in Figure 6 shows that at $\delta = 1/16$, $\rho_{FAR}(\delta) > \rho_{\ell_1}(\delta)$, and so StOMP decoding is predicted to outperform ℓ_1 decoding at such low redundancy. In our experiment, both CFAR and CFDR thresholding performed better than ℓ_1 minimization and OMP in terms of BER. Again, comparing timing measures, we see that StOMP runs at about 1/100th the time needed by BP. To summarize, StOMP provides a rapid, yet dependable, alternative to costly ℓ_1 minimization.

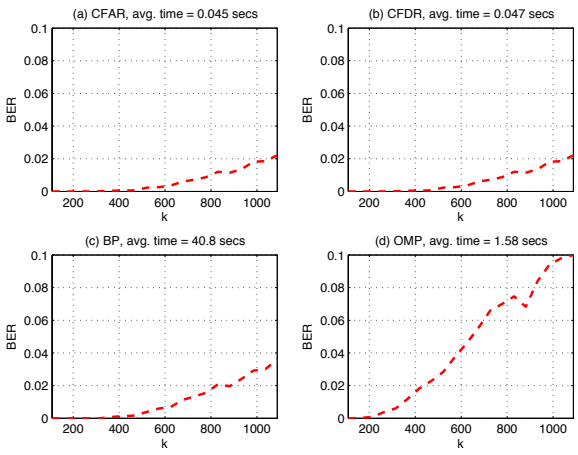

Figure 17: Performance of Decoders in a setting with small coding redundancy 1/16. (a) CFAR thresholding; (b) CFDR thresholding; (c) Basis Pursuit; (d) OMP.

9.3 Component Separation in Overcomplete Systems

We consider now the problem of separating a signal into its harmonic and impulsive components; see also [11, 21]. In detail, assume we have a signal y of length n, known to admit sparse synthesis by a few selected sinusoids and a few spikes, with a total of k such components. Formally, we have $y = [I\ F]x$, where I is an $n \times n$ identity matrix, F is an $n \times n$ Fourier matrix, and x is a $2n$ coefficient vector. [21] established bounds on the sparsity k, under which ℓ_1 minimization successfully recovers the coefficient vector x in this underdetermined setting. Here we investigate the performance of StOMP as an alternative to direct ℓ_1 minimization.

Figure 18(a) shows a signal y of length $n = 512$, consisting of 2 harmonic terms perturbed by 32 spikes, with amplitudes drawn at random from a normal distribution $N(0, 1/2)$, for a total of $k = 34$ nonzero synthesis coefficients in the time-frequency dictionary. The spike and sinusoid components of y are plotted individually in panels (b) and (c). We solved this underdetermined system using StOMP with CFAR and CFDR thresholding. Results are portrayed in the second and third rows of Figure 18. Both thresholding strategies perfectly recovered the synthesis coefficients in 4 stages, validating our claim that StOMP is a fast alternative to ℓ_1 minimization.

10 Reproducing our Results

The phrase *reproducible research* [5, 22] describes a discipline for publication of computational research together with a complete software environment reproducing that research. In that spirit, all the figures appearing in this paper can be reproduced using the *SparseLab* library for Matlab. SparseLab is a collection of Matlab functions and scripts, developed at Stanford University, available freely on the internet at *http://www-stat.stanford.edu/~sparselab*. It includes an array of tools to solve sparse approximation problems, supplemented with detailed examples and demos. SparseLab has been used by the authors to create all the figures and tables used in this article, and the toolbox contains scripts which will reproduce all the calculations of this paper.

11 Related Work

We briefly discuss several significant precursors to this work.

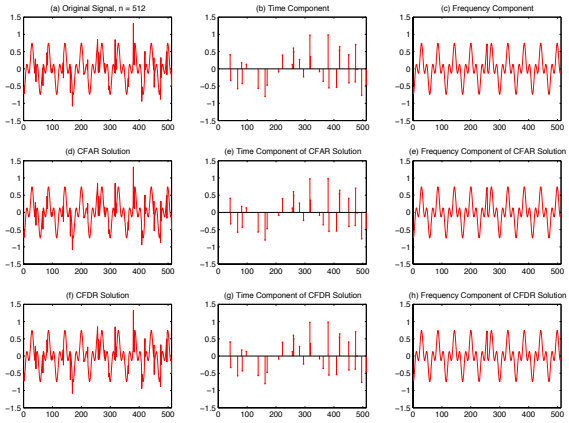

Figure 18: Time-Frequency Separation with StOMP . Top row: Original signal, with its time and frequency components; Middle row: Corresponding output of StOMP with CFAR thresholding; Middle row: Corresponding output of StOMP with CFDR thresholding.

11.1 Statistical Modelling

Statisticians have, since the advent of automatic computing, developed a very large of model building strategies. These include *forward stepwise regression* and *screening regressions*, both closely related to the present work.

In our notation, the statistical modelling problem goes as follows. One is given data $y = \Phi x + \xi$ where ξ is Gaussian noise, y is the response, the columns of Φ are predictors and x is a vector of coefficients. It is believed that most potential predictors are irrelevant, and that only a few predictors should be used, but it is not known which ones these are likely to be. Equivalently, most of the coefficients in x are zero, but the positions of the nonzeros are unknown.

Forward stepwise regression simply selects the predictor having best correlation with the current residual at each stage and adds it to the current model, which it then fits by least-squares. This procedure has been used extensively by persons of our acquaintance for more than 50 years; it is the same thing we have called OMP in this paper (a term that arose in signal processing about 15 years ago). The method of screening regressions selects all predictors having a significant correlation with the original signal; this is the same as stage one of StOMP . No doubt over the last 5 decades many people have tried iterative screening regressions at least on an informal basis.

What is different in our proposal? Here, our data may be noiseless; the underlying 'matrix of predictors' (here Φ) must however be random, e.g. generated with random independent columns. This randomness of Φ alone is somehow responsible for the phenomena described here.

Our work shows that stagewise model building – seemingly *ad hoc* and hard to analyze – can, if the underlying matrix of predictors is random, have impressive theoretical properties. It suggests that stagewise model building in Gaussian linear modeling and perhaps elsewhere may have provably good properties.

11.2 Digital Communications

Also mentioned earlier is the connection between our calculations and several key notions in multi-user detection theory. Randomly-spread CDMA systems can be thought of as posing the problem of solving $y = \Phi x_0$ where x_0 is a transmitted binary vector, y is the received vector and Φ is a rectangular array with random entries. This is like our problem, except (a) x_0 need not be sparse; and (b) Φ need not have more columns than rows.

In the MUD literature, the idea that $z = x_0 - \Phi^T y$ looks like Gaussian noise is clearly established in the work of Poor, Verdú, and others [46, 61, 12, 36]. Also, the idea that sophisticated multistage

algorithms can be applied to successively reduce MAI – e.g. onion-peeling schemes [10] or iterative decoders [4, 6, 13] - is completely consistent with our approach in this paper: stagewise least-squares projection when the nonzeros in x_0 have a power-law distribution is something like onion-peeling; ℓ_1 minimization is something like a Bayesian maximum a posteriori iterative decoder. Finally the specific analysis technique we have developed – the conditioned Gaussian limit – bears some resemblance to density evolution schemes used in the MUD/CDMA literature, eg [4, 6, 13, 30].

However, there are important differences in the problem, the primary one being that in the binary CDMA case the vector x_0 is not sparse and takes known values ± 1, which cause important differences in results. Also (perhaps) the study of very large N and n is less of interest in CDMA at the moment.

11.3 Component Separation

The immediate inspiration for this paper was extensive work by Jean-Luc Starck and Michael Elad [50, 51] who attacked very large-scale problems in image processing and component separation. They found that by stagewise application of hard thresholding, residualization, and matched filtering, they could often obtain results comparable to ℓ_1 optimization but much more rapidly. Our paper arose from Starck's insistence that such an approach was essential to attacking very large scale problems with N in the millions, and demanded theoretical attention. Focusing on the special case of 'random' Φ, we found his insistence to be prophetic.

11.4 Mathematical Analysis

The mathematical developments in this paper, and the form of the StOMP algorithm itself, arise from a lemma in the paper [16, Section 5] by one of the authors. That lemma concerned behavior of random linear programs, and was originally developed in order to show that ℓ_1-minimzation can find sparse solutions when Φ is drawn from USE; the authors noticed that it implicitly introduces the conditioned Gaussian model developed here.

11.5 Fine Points

Two small but essential points:

- The *notion of phase transition* considered here is weaker than notions often mentioned in connection with the study of ℓ_1 optimization. As the papers [18, 19] may help clarify, much of the literature discusses the notion of *strong equivalence* of ℓ_1 and ℓ_0; in this notion that for a given Φ, every sparse x_0 generates a problem (y, Φ) for which the ℓ_1 solution is the unique sparsest solution [21, 20, 16, 32, 35, 48, 55, 8, 9]. In contrast, in discussing ℓ_1 minimization in Section 4.2 above, we used the notion of *weak equivalence*, which says that equivalence holds for the typical sparse x_0 (rather than for every sparse x_0) .

  In the setting of random matrices Φ sampled from the uniform spherical ensemble, independently of x_0, strong equivalence is not the relevant notion, even though it has attracted the most theoretical attention. If Φ is random and x_0 is fixed in advance, then x_0 is overwhelmingly likely to be typical for the realized Φ. The empirically observable phase transition is thus the weak transition, whose theoretical large-system transition point for ℓ_1 minimization was derived in [19] and is given by the curve in Figure 5. For parallel discussion see [54].

  The notion discussed here for StOMP is still weaker, because our probabilistic theory only studies *approximate* reconstruction of *typical* x_0. Hence it might seem that the phase transition for StOMP mapped out here is less useful in practice than the weak equivalence transition for ℓ_1 minimization. Despite this, empirically we have found that for $N = 1600$, below its phase transition curve StOMP yields reconstructions which are numerically just as accurate as ℓ_1 minimization yields below its phase transition curve.

- *Iterative thresholding* is an algorithm which, like StOMP , successively strips structure out of a residual vector; like StOMP , it computes the vector of current correlations $c_s = \Phi^T r_s$ and applies a threshold; however, instead of orthogonal projection to remove detected structure, it uses a

relaxation strategy $r_{s+1} = r_s - \lambda \cdot A \cdot \text{Threshold}(c_s)$, subtracting structure associated to columns surviving thresholding.

The use of alternating thresholding to obtain sparse solution has been extensively deployed by Starck and co-workers [50, 51] and studied by Daubechies and co-workers [15]. An early reference applying a kind of alternating thresholding to seek sparse solutions to underdetermined systems was Coifman and Wickerhauser (1993) [14]. To our knowledge, the alternating thresholding approach has yielded today's most successful applications of large-scale sparse solutions [40, 41, 50, 51].

StOMP is subtly, but crucially different; our inspiration is the idea that for Gaussian random matrices Φ (and their close relatives), selection followed by projection affords certain definite probabilistic structure, which we have exposed and analyzed carefully here, in Theorems 1,2, and 3. The initially similar-sounding proposals for alternating soft and hard thresholding do not seem to offer any comparably strong analytic framework.

12 Summary

We described a simple algorithm for obtaining sparse approximate solutions of certain underdetermined systems of linear equations. The StOMP algorithm iteratively thresholds, selects and projects. It selects nonzeros by thresholding the output of the matched filter applied to the current residual signal, where the threshold is set above the mutual interference level. It projects the residual on a lower-dimensional subspace complementary to the span of the selected nonzeros.

StOMP is effective when the matrix Φ and/or the object x_0 render the multiple access interference approximately Gaussian. Relying on such Gaussianity, we proposed two natural strategies for threshold selection, one based on false alarm rate control and one based on false discovery rate control.

We rated the success of this approach using the phase diagram and showed that, in the standard problem suite, for either thresholding rule the phase diagram has two phases – success and failure – with a narrow transition zone between them. The transition zone shrinks in size as the problem size grows. For each method, the "success phase" of StOMP is comparable in size to the "success phase" of ℓ_1 minimization. Computational experiments showed that within the intersection of the two success phases, StOMP can deliver results comparable to l_1 minimization with dramatically less computation. Also, in numerous examples StOMP runs much faster than standard greedy approximation (matching pursuit).

Supporting the practical advantages of StOMP is a theoretical framework that accurately derives the boundary of the success phase. This boundary can be regarded as a well-defined asymptotic "sampling theory" for StOMP ; to reconstruct a sparse vector by StOMP , an asymptotically precise number of samples will be required.

StOMP extends naturally to noisy data, and also extends to underdetermined systems outside the 'random' ones where the method was derived, important examples being the partial Fourier ensemble and pairs of incoherent orthogonal bases such as (Dirac,Fourier). Stylized applications in compressed sensing, suppression of impulsive noise, and time-frequency separation are given, and software is available.

Appendices

A: Proof of Lemma 1

Consider the restriction $\tilde{x}_S = x_S|_{I_S}$, i.e. 'throw away' the entries in x_S lying outside I_S. By hypothesis all the entries that are thrown out are zero, and the residual from StOMP is zero, so $y = \Phi_{I_S}\tilde{x}_S$.

Similarly, consider the restriction $\tilde{x}_0 = x_0|_{I_S}$, i.e. form the vector x_0 with entries outside of I_S thrown away. Since by hypothesis, x_0 is supported inside I_S, we also have $y = \Phi_{I_S}\tilde{x}_0$.

Hence $0 = \Phi_{I_S}(\tilde{x}_S - \tilde{x}_0)$. By hypothesis the columns of Φ are in general position, so there is no nontrivial relation $0 = \Phi_{I_S}\tilde{v}$ where $\#I_S < n$ and $\tilde{v}$ is a column vector compatible with Φ_{I_S}. Hence $\tilde{x}_S = \tilde{x}_0$ and also $x_S = x_0$. □

B: Heuristic Phase Transition Calculation for the Standard Ensemble

We have found that the following heuristic model can be used to derive a predicted phase transition curve that qualitatively matches the behavior of the empirical phase transition in the standard problem suite. We use notation paralleling the notation of Section 6.

Under this heuristic, the normalized coefficients $\langle \phi_i, r_s \rangle / \|Q_s \phi_i\|_2$ for $i \in I_{s-1}^c$ act as random samples from a Gaussian mixture distribution:

$$(1 - \tilde{\epsilon}_s) N(0, \tilde{\sigma}_s^2) + \tilde{\epsilon}_s N(0,1).$$

where $\tilde{\epsilon}_s = k_s/N_s$ and $\tilde{\sigma}_s = \sqrt{k_s/n_s}$.

We therefore define

$$\tilde{\alpha}_s = P\{|N(0, \tilde{\sigma}_s^2)| > t_s \cdot \tilde{\sigma}_s\},$$

and

$$\tilde{\beta}_s = P\{|N(0,1)| > t_s \cdot \tilde{\sigma}_s\}.$$

We start with $(\tilde{\rho}_1, \tilde{d}_1, \tilde{\nu}_1, \tilde{\sigma}_1) = (\rho, 1, 1/\delta, \rho)$ and apply the obvious updating formulas

$$\begin{aligned}
\tilde{d}_{s+1} &= \tilde{d}_s - \tilde{\alpha}_s(\tilde{\nu}_s - \tilde{\rho}_s) - \tilde{\beta}_s \tilde{\rho}_s \\
\tilde{\rho}_{s+1} &= \tilde{\rho}_s - (1 - \tilde{\beta}_s)\tilde{\rho}_s \\
\tilde{\nu}_{s+1} &= \tilde{\nu}_s - (1 - \tilde{\alpha}_s)(\tilde{\nu}_s - \tilde{\rho}_s) + (\tilde{\rho}_{s+1} - \tilde{\rho}_s) \\
\tilde{\sigma}_{s+1} &= \sqrt{\frac{\tilde{\rho}_{s+1}}{\tilde{d}_{s+1}}}.
\end{aligned}$$

If, at or before stage S, we achieve $\tilde{d}_s > \eta$ and $\tilde{\rho}_s < \eta \cdot \rho$, we declare StOMP to be successful at that given (δ, ρ).

For CFAR thresholding, t_s is constant, set to the $1 - \alpha_S/2$ quantile of the standard normal distribution, where $\alpha_S = 1 - (\frac{1-\delta}{1-\rho\delta})^{1/S}$. For CFDR thresholding, t_s is variable, set to the solution of

$$\frac{P\{|N(0, \tilde{\sigma}_s^2)| > t\tilde{\sigma}_s\}}{P\{|N(0,1)| > t\tilde{\sigma}_s\}} = \frac{\tilde{\epsilon}_s}{1 - \tilde{\epsilon}_s} \cdot \frac{q}{1-q}.$$

Here q is the assumed false discovery rate; we used $1/2$ in all our work.

These heuristic formulas cannot be precisely correct, for two reasons: (i) they omit the effect of conditioning caused by earlier stages, as discussed in Section 7 – the underlying coefficient distribution is a mixture, but not of normal distributions; and (ii) the pseudo-standard deviation is merely a crude upper bound; the true value reflects considerations from random matrix theory.

However, if the variance $\tilde{\sigma}_s$ drops very rapidly with increasing s, often this heuristic is reasonably accurate.

C: Proof of Theorem 1

By *state vector* $h_{s,n}$ we mean a four-tuple

$$h_{s,n} = (\frac{k_{s,n}}{n}, \frac{n_{s,n}}{n}, \frac{N_{s,n}}{n}, \frac{\|r_s\|_2^2}{n}), \qquad 1 \le s \le S+1;$$

note that this is a random variable. For $s = 1$ we have

$$h_{1,n} = (\frac{k_n}{n}, 1, \frac{N_n}{n}, \frac{\|y\|_2^2}{n});$$

while later steps depend on the specific realization (y, Φ) and the progress of the algorithm. Note that if at step $\ell < S$, $n_{\ell,n} = 0$, so the algorithm must stop prior to planned termination, we put, by definition

$$h_{s,n} = h_{\ell,n}, \quad \ell \le s \le S+1.$$

The distance between two state vectors $d(h, h')$ is just the Euclidean distance between the 4-tuples.

By *history* $H_{s,n}$ we mean the sequence of state vectors up to stage s,

$$H_{s,n} = (h_{1,n}, \ldots, h_{s,n}).$$

This is again a random variable. By *distance between two histories* we mean the maximal distance between corresponding states of the history:

$$d(H_s, H'_s) = \max_{1 \le \ell \le s} d(h_\ell, h'_\ell).$$

We now observe that all the quantities $\alpha_{s,n}$, $\ldots$, $\nu_{s,n}$ defined in the statement of Theorem 1 are uniformly continuous functions of the history $H_{s+1,n}$. Therefore, it is sufficient to prove that there exists a large-system limit for the history $H_{S+1,n}$, i.e. that, given (δ, ρ), there is $H_{S+1}(\delta, \rho)$ so that for each $\epsilon > 0$,

$$P_n\{d(H_{S+1,n}, \bar{H}_{S+1}) > \epsilon\} \to 0, \quad n \to \infty. \tag{12.1}$$

This can be done inductively, as follows. First, for $s = 1$, recall that we assume $y = \sum_{\ell=1}^{k} \pm_\ell \phi_{i_\ell}$. We have, by elementary calculations,

$$\operatorname*{p.lim}_{n\to\infty} \|y\|_2^2/n = \lim_{n\to\infty} k_n/n = \rho.$$

This proves:

Lemma 12.1 (Large-system limit, $s = 1$) *Let $n/N_n \to \delta$ and $k_n/n \to \rho$ as $n \to \infty$. Then*

$$\operatorname*{p.lim}_{n\to\infty} h_{1,n} = \bar{h}_1(\delta, \rho) \equiv (\rho, 1, \delta^{-1}, \rho). \tag{12.2}$$

Hence, the inductive process is begun. Below we will show that there is a deterministic sequence $(\bar{h}_s : s = 2, \ldots, S+1)$ obeying

$$\bar{h}_{s+1} = \bar{h}_s - \Delta_s(\bar{H}_s), \quad s = 1, \ldots, S, \tag{12.3}$$

for certain 4-tuple-valued functions Δ_s of s-step histories; for each $\epsilon > 0$, this sequence obeys

$$P_n\{d(h_{s+1,n}, \bar{h}_{s+1}) > 2\epsilon | d(h_{s,n}, \bar{h}_s) < \epsilon\} \to 0, \quad n \to \infty, \quad s = 1, \ldots, S, \tag{12.4}$$

then (12.1) follows.

To establish (12.4) we consider in turn each individual component $\bar{h}_s(\ell)$, $\ell = 1, \ldots 4$. The first three components share much in common, owing to the similar relations

$$k_{s+1} = k_s - |J_s \cap I_0|, \tag{12.5}$$

$$n_{s+1} = n_s - |J_s|,$$

and

$$N_{s+1} = N_s - |J_s|.$$

Consider the first component $h_{s,n}(1) = k_{s,n}/n$. In the next subsection we prove the following:

Lemma 12.2 *Suppose either that $s = 1$ or that $s > 1$ and (12.4) has been proven for $1 \le \ell \le s-1$. Let $\tilde{X}_s$ denote the conditioned Gaussian random variable defined as in Section 7 by the sequences $(\bar{\mu}_\ell : 1 \le \ell \le s)$ and $(\bar{\sigma}_\ell : 1 \le \ell \le s)$, where*

$$\bar{\mu}_s = \bar{h}_s(4)/\bar{h}_s(1), \qquad \bar{\sigma}_s = \sqrt{\bar{h}_s(4)}. \tag{12.6}$$

Then

$$\frac{k_{s+1,n}}{n} = \frac{k_{s,n}}{n}(1 - P\{|\tilde{X}_s| > t\bar{\sigma}_s\}) + o_p(1), \quad n \to \infty. \tag{12.7}$$

Now of course $P\{|\tilde{X}_s| > t\bar{\sigma}_s\}$ depends on the limit history $\bar{H}_s$; so it makes sense to write $\Delta_{s,1}(\bar{H}_s) \equiv \bar{h}_s(1) \cdot P\{|\tilde{X}_s| > t\bar{\sigma}_s\}$. Exploiting the result $k_{s,n}/n \to_P \bar{h}_s(1)$ which we regard as proved at a previous stage of the argument, we rewrite (12.7) as

$$\bar{h}_{s+1}(1) = \bar{h}_s(1) - \Delta_{s,1}(\bar{H}_s), \qquad s = 1, \ldots, S.$$

We can argue very similarly for the second and third components of the state vector, n_s/n and N_s/n. Put $\bar{\epsilon}_s = \bar{h}_s(1)/\bar{h}_s(3)$, $\bar{\nu}_s = \bar{h}_s(3)$, and define

$$\Delta_{s,2}(\bar{H}_s) = \bar{\nu}_s \cdot \left[(1 - \bar{\epsilon}_s) \cdot P\{|\tilde{Z}_s| > t\bar{\sigma}_s\} + \bar{\epsilon}_s \cdot P\{|\tilde{X}_s| > t\bar{\sigma}_s\} \right],$$

where $\tilde{X}_s$ is a conditioned Gaussian random variable defined by $(\bar{\mu}_s)$ and $(\bar{\sigma}_s)$ as in (12.6), and where $\tilde{Z}_s$ is a conditioned Gaussian random variable with the same $\bar{\sigma}$-sequence but with means $\mu = 0$. This defines a continuous function of s-step histories, and putting $\Delta_{s,2} = \Delta_{s,3}$ one can show

$$\bar{h}_{s+1}(\ell) = \bar{h}_s(\ell) - \Delta_{s,\ell}(\bar{H}_s), \qquad s = 1, \ldots, S, \quad \ell = 2, 3.$$

However, the arguments being so similar to those behind Lemma 12.2, we omit them.

This leaves only the need to argue for the fourth and final component of the state vector. Now the analog of (12.5) is

$$\|r_{s+1}\|_2^2 = \|r_s\|_2^2 - c_s^T (\Phi_{J_s}^T \Phi_{J_s})^{-1} c_s,$$

where

$$c_s = \Phi_{J_s}^T r_s,$$

and Φ_{J_s} is the submatrix of Φ with columns from $J_s = \{j : |c_s| > t\sigma_s\}$. The subsection after next proves

Lemma 12.3 *Suppose either that $s = 1$ or that $s > 1$ and that (12.4) has been proven for $1 \le \ell \le s-1$. Let $\tilde{X}_s$ denote the conditioned Gaussian random variable defined as in Section 7 by the sequences $(\bar{\mu}_\ell : 1 \le \ell \le s)$ and $(\bar{\sigma}_\ell : 1 \le \ell \le s)$, obeying (12.6). Similarly, let $\tilde{Z}_s$ be a conditioned Gaussian random variable defined as in Section 7 by $\bar{\mu}_\ell = 0$ and by $(\bar{\sigma}_\ell : 1 \le \ell \le s)$ again as in (12.6). Put $\bar{\epsilon}_s \equiv \bar{h}_s(1)/\bar{h}_s(3)$, and define the mixture density*

$$p_s(x) = (1 - \bar{\epsilon}_s) \cdot p_{\tilde{Z}_s}(x) + \bar{\epsilon}_s \cdot p_{\tilde{X}_s}(x).$$

Put $\bar{\nu}_s \equiv h_s(3)$ and

$$\Gamma_s(\bar{H}_s) = \bar{\nu}_s \cdot \int x^2 1_{\{|x| > t\bar{\sigma}_s\}} p_s(x) dx;$$

and, with $\bar{d}_s \equiv \bar{h}_s(2)$, set

$$\Delta_{s,4}(\bar{H}_s) \equiv \frac{\Gamma_s(\bar{H}_s)}{\bar{d}_{s+1} + \Gamma_s(\bar{H}_s)/\bar{\sigma}_s^2}.$$

Then

$$n^{-1}\|r_{s+1}\|_2^2 = n^{-1}\|r_s\|_2^2 - \Delta_{s,4}(\bar{H}_s) + o_p(1), \qquad n \to \infty.$$

It follows that

$$\bar{h}_{s+1}(4) = \bar{h}_s(4) - \Delta_{s,4}(\bar{H}_s), \qquad s = 1, \ldots, S.$$

This completes the proof of (12.3)-(12.4). □

The Role of Interleaving in Theorems 1,2,3.

We have stated Theorems 1, 2, and 3 in the body of our paper as if they are independent propositions, proved separately. Actually we only present arguments to prove them in an *interleaved, sequential* fashion.

We think of the 3 theorems as $3S + 1$ 'little' theorems, call them little Theorem 1_ℓ, $\ell = 1, \ldots, S+1$, and little Theorems 2_ℓ, 3_ℓ, $\ell = 1, \ldots, S$. Thus little Theorem 1_1 yields the special case $s = 1$ of 'big' Theorem 1. Little Theorem 1_2 yields the special case $s = 2$, etc. This decomposition of the original 'big' theorems into component theorems involves slightly different hypotheses than the original ones; in fact

our proof of little Theorem 1_s depends on exploiting the conclusions previously established in proving little Theorems 1_{s-1}, 2_{s-1}, and 3_{s-1}.

Thus, our proofs really show that, once little Theorem 1_1 is proved, little Theorems 2_1 and 3_1 can be proved.

Then, little Theorem 1_2 can be proved, followed by little Theorems 2_2 and 3_2.

In general, for $s > 1$, our proof of little Theorem 1_s requires that all the little Theorems 1_ℓ, for $1 \le \ell < s$ be proved as well as the little Theorems 2_ℓ and 3_ℓ, for $1 \le \ell < s$. We felt it best from an expository viewpoint to hide this interleaving until now. The interleaving has finally become explicit in the statement of Lemmas 12.2-12.3 above.

Analysis of k_s/n

$$\begin{aligned} |J_s \cap I_0| &= \sum_{i \in I_{s-1}^c \cap I_0} 1_{\{|\langle \phi_i, r_s \rangle| > t\sigma_s\}} \\ &= \sum_{\ell=1}^{k_s} V_{\ell,s}, \end{aligned}$$

say.

Now the $(\phi_j : j \in I_0)$ are independent identically distributed random vectors. Membership in I_{s-1}^c is a condition imposed symmetrically on all of the random vectors in $(\phi_j : j \in I_{s-1}^c \cap I_0)$. It follows that, conditional on I_{s-1}, these are conditionally exchangeable random variables, which proves:

Lemma 12.4 *Conditional on $H_{s,n}$, the $(V_{\ell,s} : \ell = 1, \ldots, k_s)$ are exchangeable random variables.*

Exchangeability allows easy characterisation of the mean and variance of $n^{-1}\sum_\ell V_\ell$. Now

$$E(V_{\ell,s}|H_s) = P_n\{|\langle \phi_i, r_s \rangle| > t\sigma_s | i \in I_0 \cap I_{s-1}^c, I_0, I_{s-1}\}.$$

so

$$E(n^{-1}\sum_\ell V_{\ell,s}|H_s) = \frac{k_s}{n} \cdot P_n\{|\langle \phi_i, r_s \rangle| > t\sigma_s | i \in I_0 \cap I_{s-1}^c, I_0, I_{s-1}\ \};$$

meanwhile,

$$Var(n^{-1}\sum_{\ell=1}^{k_{s,n}} V_{\ell,s}|H_s) = k_s/n^2 \cdot Var(V_{1,s}|H_s) + (k_s^2 - k_s)/n^2 Cov(V_{1,s}, V_{2,s}|H_s).$$

We now use interleaving. We assume that little Theorems 1_ℓ,2_ℓ, 3_ℓ have all been proved, for $1 \le \ell \le s$ and we are trying to prove little Theorem 1_{s+1}. Combining Theorems 1_s and 3_s we obtain

Lemma 12.5 *With $\tilde{X}_s$ as in the statement of Lemma 12.2,*

$$\lim_{n\to\infty} P_n\{|\langle \phi_i, r_s \rangle| > t_s\sigma_s | i \in I_0 \cap I_{s-1}^c, I_0, I_{s-1}\} = P\{|\tilde{X}_s| > t\bar{\sigma}_s\}.$$

We omit the proof of the following technical lemma.

Lemma 12.6

$$0 = \operatorname*{p.lim}_{n\to\infty} Cov(V_{1,s}, V_{2,s}|H_{s,n}).$$

Combining these lemmas proves Lemma 12.2 □

Analysis of $\|r_s\|_2^2/n$

It is enough to show that for $\bar{d}_s \equiv \bar{h}_s(2)$,

$$\operatorname*{p.lim}_{n\to\infty} n^{-1} c_s^T (\Phi_{J_s}^T \Phi_{J_s})^{-1} c_s = \frac{\Gamma_s(\bar{H}_s)}{\bar{d}_{s+1} + \Gamma_s(\bar{H}_s)/\bar{\sigma}_s^2}. \tag{12.8}$$

We may choose coordinates in R^{n_s} so that r_s is aligned with e_1, the standard unit vector. Then in fact $\tilde{c}_s^T \equiv c_s^T/\|r_s\|_2$ is the first row of Φ_{J_s} and so, letting B_{J_s} denote the $n_s - 1 \times |J_s|$ matrix consisting of all but the first row of Φ_{J_s},

$$\|r_{s+1}\|_2^2 - \|r_s\|_2^2 = c_s^T (\tilde{c}_s \tilde{c}_s^T + B_{J_s}^T B_{J_s})^{-1} c_s.$$

The matrix Φ almost has Gaussian iid entries in the following sense. Let D be an n_s-by-n_s diagonal matrix with diagonal entries independently and identically distributed as $n_s^{-1/2} \times \chi_{n_s}$, where χ_d denotes the usual χ-distribution on d degrees of freedom; in addition let D be stochastically independent of Φ. Then ΦD is a random matrix with Gaussian iid $N(0, 1/n)$ entries.

Hence, conditionally on I_{s-1}, the entries in the matrix $\Psi = B_{J_s} D_{J_s}$ are iid Gaussian random variables. Although we omit details, clearly our problem approximately reduces to studying

$$c_s^T (\tilde{c}_s \tilde{c}_s^T + \Psi^T \Psi)^{-1} c_s.$$

Invoke now the Sherman-Morrison formula: for a column vector w and a compatible nonsingular matrix A,

$$w^T (w w^T + A)^{-1} w = \frac{w^T A^{-1} w}{1 + w^T A^{-1} w}.$$

Note now that Ψ is stochastically independent of c_s. Pick an orthogonal matrix $V \in SO(k_s)$ whose first column is proportional to c_s and which is randomly sampled from the canonical uniform measure on $SO(k_s)$ subject to this constraint. Also let V be stochastically independent of Ψ. Define $\tilde{\Psi} = \Psi V$. Now $e_1 \propto V c_s$ and by a simple computation,

$$c_s^T (\Psi^T \Psi)^{-1} c_s = \|c_s\|^2 (\tilde{\Psi}^T \tilde{\Psi})_{11}^{-1}.$$

At the same time, $\tilde{\Psi}$ has iid Gaussian $N(0, 1/n)$ entries.

Some well-known facts about random Gaussian matrices allow us to study the random diagonal entry $(\tilde{\Psi}^T \tilde{\Psi})_{1,1}^{-1}$; compare the following restatement of results of Bai and Yin quoted in [60]. We omit the proof.

Lemma 12.7 *Let W_n be a $p_n \times q_n$ matrix with entries i.i.d. $N(0, 1/n)$. Let $(p_n - q_n)/n \to \gamma > 0$ as $n \to \infty$. Then*

$$\operatorname*{p.lim}_{n\to\infty} (W_n^T W_n)_{1,1}^{-1} = \frac{1}{\gamma}.$$

In the case of interest to us, $p_n = |I_{s-1}^c|$ while $q_n = |J_s|$. Hence, $p_n - q_n = |I_s^c|$ and $\gamma = \bar{d}_{s+1}$.

We now again use interleaving. We assume that little Theorems 1_ℓ,2_ℓ, 3_ℓ have all been proved, for $1 \le \ell \le s$ and we are trying to prove the part of little Theorem 1_{s+1} referring to the fourth component of $\bar{h}_{s+1}$. Restating little Theorems 2_s and 3_s we obtain that the typical component of c_s at indices $i \in I_0 \cap I_{s-1}^c$ has approximately the probability distribution of $\tilde{X}_s$, while the typical component of c_s at indices $i \in I_0^c \cap I_{s-1}^c$ has approximately the probability distribution of $\tilde{Z}_s$. Hence

$$\begin{aligned}
\|c_{s,n}\|_2^2 &= \sum_i \langle \phi_i, r_s \rangle^2 1_{\{|\langle \phi_i, r_s \rangle| > t\sigma_s\}} \\
&= \sum_{i \in I_0^c \cap I_{s-1}^c} + \sum_{i \in I_0 \cap I_{s-1}^c} \\
&\sim (N_s - k_s) \int z^2 1_{\{|z| > t\bar{\sigma}_s\}} p_{\tilde{Z}_s}(z) dz + k_s \int x^2 1_{\{|x| > t\bar{\sigma}_s\}} p_{\tilde{X}_s}(x) dx \\
&\sim n \cdot \bar{\nu}_s \cdot \int x^2 1_{\{|x| > t\bar{\sigma}_s\}} p_s(x) dx \sim n \cdot \Gamma_s(\bar{H}_s).
\end{aligned}$$

Lemma 12.8 *With $\tilde{X}_s$ as in the statement of Lemma 12.2, We have*

$$\operatorname*{p.lim}_{n\to\infty} \|c_{s,n}\|_2^2/n = \Gamma_s(\bar{H}_s).$$

Combining the above yields (12.8). □

References

[1] F. Abramovich, Y. Benjamini, D.L. Donoho and I.M. Johnstone (2006) Adapting to unknown sparsity by controlling the False Discovery Rate. in press, *Annals of Statitsics*.

[2] Y. Benjamini and Y. Hochberg (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B.* **57**, pp. 289–300.

[3] E.R.Berlekamp, R.J.McEliece, and H.C.A. van Tilborg. (1978) On the inherent intractability of certain coding problems. *IEEE Trans. Info. Thry.*, 24:384–386, 1978.

[4] J. Boutros and G. Caire (2002) Iterative multiuser joint decoding: unified framework and asymptotic analysis. *IEEE Trans. Info. Thry.* **48** 1772-

[5] J. Buckheit and D. L. Donoho (1995) WaveLab and reproducible research, in A. Antoniadis, Editor, *Wavelets and Statistics*, Springer.

[6] G. Caire, R. Müller and T. Tanaka (2004) Iterative multiuser joint decoding: optimal power allocation and low-complexity implementation. *IEEE Trans. Info. Thry.* **50**(9), pp. 1950–1973.

[7] E.J. Candès, J. Romberg and T. Tao. (2004) Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. To Appear, *IEEE Trans. Info. Thry.*

[8] E.J. Candès and T. Tao. (2004) Near Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? To Appear, *IEEE Trans. Info. Thry.*

[9] E.J. Candès and T. Tao. (2005) Decoding via Linear Programming. *IEEE Trans. Info. Thry.*

[10] Naftali Chayat and Shlomo Shamai (1999) Convergence properties of iterative soft onion peeling. in *Proc. 1999 IEEE Info. Thry. Workshop*, Kruger National Park, South Africa.

[11] S. Chen, D. Donoho, and M.A. Saunders (1999) Atomic Decomposition by Basis Pursuit. *SIAM J. Sci Comp.*, **20**(1), pp. 33–61.

[12] E. Chong, J. Zhang, and D. Tse (2001) Output MAI distribution of linear MMSE multiuser receivers in DS-CDMA systems. *IEEE Trans. Info. Thry.* **47**(3), pp. 1128–1144.

[13] S.Y. Chung, T.J. Richardson and R.L. Urbanke (2001) Analysis of Sum-Product decoding of low-density parity-check codes using a Gaussian approximation. *IEEE Trans. Info Thry.* **47**(2), pp. 657–670.

[14] Ronald R. Coifman and Mladen Victor Wickerhauser (1993) Wavelets and adapted waveform analysis: A toolkit for signal processing and numerical analysis. In *Different Perspectives on Wavelets*, Ingrid Daubechies, editor. number 47 in Proceedings of Symposia in Applied Mathematics, American Mathematical Society. Providence, Rhode Island.

[15] I. Daubechies, M. Defrise and C. De Mol (2004) An iterative thresholding algorithm for linear inverse problems with a sparsity constraint *Comm. Pure Applied Mathematics* **57**, 1413 - 1457.

[16] D.L. Donoho (2004) For most large underdetermined systems of linear equations, the minimal ℓ_1-norm solution is also the sparsest solution. Submitted. *Com.. Pure Appl. Math.*, to appear 2006.

[17] D.L. Donoho (2004) Compressed sensing. *IEEE Trans. Info. Thry.*, to appear 2006.

[18] D.L. Donoho (2005) Neighborly Polytopes and the Sparse Solution of Underdetermined Systems of Linear Equations. To appear, *IEEE Trans. Info. Thry.*.

[19] D.L. Donoho (2005) High-dimensional centrosymmetric polytopes with neighborliness proportional to dimension. to appear, *Discrete and Computational Geometry*, 2006.

[20] D.L. Donoho and M. Elad (2003) Optimally Sparse Representation from Overcomplete Dictionaries via ℓ_1 norm minimization. *Proc. Natl. Acad. Sci. USA*, **100**(5), pp. 2197–2002.

[21] D.L. Donoho and X. Huo (2001) Uncertainty Principles and Ideal Atomic Decomposition. *IEEE Trans. Info. Thry.* **47**(7), pp. 2845-62.

[22] D.L. Donoho and X. Huo (2004) BeamLab and Reproducible Research. *International Journal of Wavelets, Multiresolution and Information Processing*, **2**(4), pp. 391–414.

[23] D.L. Donoho and J. Jin (2006) Asymptotic Minimaxity of False Discovery Rate Thresholding for Exponential Data. *Ann. Statist.*, to appear.

[24] D.L. Donoho and I.M. Johnstone (1994) Minimax risk over ℓ_p-balls for ℓ_q-error. *Probability Theory and Related Fields* **99**, pp. 277–303.

[25] D.L. Donoho and I.M. Johnstone (1994) Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, **81**, pp. 425–455.

[26] D.L. Donoho, I.M. Johnstone, A.S. Stern and J.C. Hoch (1992) Maximum Entropy and the Nearly Black Object (with discussion). *Journal of the Royal Statistical Society, Series B*, **54**(1), pp. 41–81.

[27] D.L. Donoho and I.M. Johnstone (1995) Adapting to unknown smoothness via wavelet shrinkage. *J. Amer. Statist. Assoc.*, **90**, pp. 1200–1224.

[28] D.L. Donoho and B.F. Logan (1992) Signal Recovery and the Large Sieve. *SIAM Journal of Applied Math*, **52**, pp. 577–591.

[29] M. F. Duarte, M.B. Wakin and R.G. Baraniuk (2005) Fast Reconstruction of Piecewise Smooth Signals from Random Projections (Online Proc. SPARS Workshop, Nov. 2005)

[30] H. El Gamal and A.R. Hammons, Jr. (2001) Analyzing the Turbo decoder using the Gaussian Approximation. *IEEE Trans. Info. Thry.*, **47**(2), pp. 671–686.

[31] Ray Freeman and Eriks Kupce (2003) New Methods for Fast Multidimensional NMR. *Journal of Biomolecular NMR*, **27**, pp. 101–113.

[32] Jean-Jacques Fuchs (2002) On Sparse Representations in Arbitrary Redundant Bases. *IEEE Trans. Info. Thry*, **50**(6), pp. 1341–44.

[33] A.C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan and M. Strauss (2002) Near-optimal sparse Fourier representations via sampling. *Proc 34th ACM symposium on Theory of Computing*, pp. 152–161, ACM Press.

[34] G. Golub and C. van Loan (1996) *Matrix computations*, third edition, The Johns Hopkins University Press, London.

[35] R. Gribonval and M. Nielsen (2003) Sparse Representations in Unions of Bases. *IEEE Trans. Info. Thry* **49**(12), pp. 1320–25.

[36] D. Guo, S. Verdú and L. Rasmussen (2002) Asymptotic normality of linear multiuser receiver outputs. *IEEE Trans. Info. Thry.* **48**(12), 3080-3095.

[37] I.M. Johnstone and B.W. Silverman (2004) Needles and straw in haystacks: empirical Bayes estimates of possibly sparse sequences. *Annals of Statistics*, **32**, pp. 1594–1649.

[38] Y. Hochberg and A. Tamhane (1987) *Multiple Comparison Procedures*, John Wiley & Sons, New York.

[39] E. Lehmann and J. Romano (2003) *Testing Statistical Hypotheses* (3rd edition), Springer, New York.

[40] M. Lustig, J.H. Lee, D.L. Donoho and J.M. Pauly (2004) Faster Imaging with Randomly Perturbed Spirals and L_1 Reconstruction. *Proc. of the ISMRM 13th annual meeting.*

[41] M. Lustig, J.M Santos, D.L. Donoho and J.M. Pauly (2006) Rapid MR Angiography with Randomly Under-Sampled 3DFT Trajectories and Non-Linear Reconstruction. *Proc. of the SCMR 9th annual Scientific meeting*, to appear.

[42] M.W. Maciejewski, A.S. Stern, G.F. King and J.C. Hoch (2006) Nonuniform Sampling in Biomolecular NMR. *Handbook of Modern Magnetic Resonance*, Graham A. Webb, Springer.

[43] S. Mallat and Z. Zhang (1993) Matching Pursuit with time-frequency dictionaries, *IEEE Transactions on Signal Processing*, **41**(12), pp. 3397–3415.

[44] P. McCullagh (1987) *Tensor Methods in Statistics*, Chapman and Hall.

[45] A. Paulraj, R. Nabar and D. Gore (2003) *Introduction to Space-Time Wireless Communications*, Cambridge University Press.

[46] H.V. Poor and S. Verdú (1997) Probability of error in MMSE multiuser detection. *IEEE Trans. Info. Thry.* **43**(3), pp. 858–871.

[47] D. Rovnyak, D.P. Frueh, M. Sastry, Z.J. Sun, A.S. Stern, J.C. Hoch and G. Wagner (2004) Accelerated acquisition of high resolution triple-resonance spectra using non-uniform sampling and maximum entropy reconstruction. *Journal of Magnetic Resonance*, **170**(1), pp. 15–21.

[48] M. Rudelson and R. Vershynin (2005) Geometric approach to error-correcting codes and reconstruction of signals. Technical report, Dept. of Mathematics, Univ. of California at Davis.

[49] M. A. Saunders (2002) PDCO: Primal-Dual interior method for Convex Objectives. Available at: http://www.stanford.edu/group/SOL/software/pdco.html

[50] J.-L. Starck, M. Elad and D.L. Donoho (2004) Redundant Multiscale Transforms and their Application for Morphological Component Analysis. *Advances in Imaging and Electron Physics*, **132**.

[51] J.-L. Starck, M. Elad and D.L. Donoho (2005) Image Decomposition Via the Combination of Sparse Representation and a Variational Approach. *IEEE Trans. Image Proc.*, **14**(10), pp. 1570–1582.

[52] V.N. Temlyakov (1999) Greedy algorithms and m-term approximation, *J. Approx. Theory*, **98**, pp. 117–145.

[53] V.N. Temlyakov (2003) Nonlinear methods of approximation. *Found. Comput. Math.*, **3**, pp. 33–107.

[54] J.A. Tropp and A. Gilbert (2005) Signal Recovery from Partial Information by Orthogonal Matching Pursuit. Manuscript.

[55] J.A. Tropp (2003) Greed is Good: Algorithmic Results for Sparse Approximation. *IEEE Trans Info. Thry.* **50**(11), pp. 2231–42.

[56] J.A. Tropp (2004) Just Relax: Convex programming methods for Subset Selection and Sparse Approximation. Manuscript.

[57] Y. Tsaig and D.L. Donoho (2005) Breakdown of equivalence between the minimal ℓ_1-norm solution and the sparsest solution. *EURASIP Journal of Applied Signal Processing*, in press.

[58] Y. Tsaig and D.L. Donoho (2005) Extensions of compressed sensing. *EURASIP Journal of Applied Signal Processing*, in press.

[59] D.N.C. Tse and S. Verdú (2000) Optimum asymptotic multiuser efficiency of randomly spread CDMA, *IEEE Trans Info. Thry.*, **46**(7), pp. 2718–2722.

[60] S. Verdú. (1998) *Multiuser Detection*. Cambridge University Press.

[61] S. Verdú and S. Shamai (1999) Spectral efficiency of CDMA with random spreading. *IEEE Trans. Info. Thry.* **45**(2), pp. 622–640.