Fast Solution of ℓ_1-norm Minimization Problems When the Solution May be Sparse

David L. Donoho* and Yaakov Tsaig†

October 2006

Abstract

The minimum ℓ_1-norm solution to an underdetermined system of linear equations $y = Ax$, is often, remarkably, also the sparsest solution to that system. This sparsity-seeking property is of interest in signal processing and information transmission. However, general-purpose optimizers are much too slow for ℓ_1 minimization in many large-scale applications.

The Homotopy method was originally proposed by Osborne *et al.* for solving noisy overdetermined ℓ_1-penalized least squares problems. We here apply it to solve the noiseless underdetermined ℓ_1-minimization problem min $\|x\|_1$ subject to $y = Ax$. We show that Homotopy runs much more rapidly than general-purpose LP solvers when sufficient sparsity is present. Indeed, the method often has the following *k-step solution property*: if the underlying solution has only k nonzeros, the Homotopy method reaches that solution in only k iterative steps. When this property holds and k is small compared to the problem size, this means that ℓ_1 minimization problems with k-sparse solutions can be solved in a fraction of the cost of solving one full-sized linear system.

We demonstrate this k-step solution property for two kinds of problem suites. First, incoherent matrices A, where off-diagonal entries of the Gram matrix $A^T A$ are all smaller than M. If y is a linear combination of at most $k \leq (M^{-1} + 1)/2$ columns of A, we show that Homotopy has the k-step solution property. Second, ensembles of $d \times n$ random matrices A. If A has iid Gaussian entries, then, when y is a linear combination of at most $k < d/(2\log(n)) \cdot (1 - \epsilon_n)$ columns, with $\epsilon_n > 0$ small, Homotopy again exhibits the k-step solution property with high probability. Further, we give evidence showing that for ensembles of $d \times n$ partial orthogonal matrices, including partial Fourier matrices, and partial Hadamard matrices, with high probability, the k-step solution property holds up to a dramatically higher threshold k, satisfying $k/d < \hat{\rho}(d/n)$, for a certain empirically-determined function $\hat{\rho}(\delta)$.

Our results imply that Homotopy can efficiently solve some very ambitious large-scale problems arising in stylized applications of error-correcting codes, magnetic resonance imaging, and NMR spectroscopy. Our approach also sheds light on the evident parallelism in results on ℓ_1 minimization and Orthogonal Matching Pursuit (OMP), and aids in explaining the inherent relations between Homotopy, LARS, OMP, and Polytope Faces Pursuit.

Key Words and Phrases: LASSO. LARS. Homotopy Methods. Basis Pursuit. ℓ_1 minimization. Orthogonal Matching Pursuit. Polytope Faces Pursuit. Sparse Representations. Underdetermined Systems of Linear Equations.

Acknowledgements. This work was supported in part by grants from NIH, ONR-MURI, and NSF DMS 00-77261, DMS 01-40698 (FRG) and DMS 05-05303. We thank Michael Lustig and Neal Bangerter of the Magnetic Resonance Systems Research Lab (MRSRL) at Stanford University for supplying us with 3-D MRI data for use in our examples. YT would like to thank Michael Saunders for helpful discussions and references.

*Department of Statistics, Stanford University, Stanford CA, 94305

†Institute for Computational and Mathematical Engineering, Stanford University, Stanford CA, 94035

1 Introduction

Recently, ℓ_1-norm minimization has attracted attention in the signal processing community [9, 26, 20, 7, 5, 6, 8, 27, 31, 32, 44], as an effective technique for solving underdetermined systems of linear equations, of the following form. A measurement vector $y \in \mathbf{R}^d$ is generated from an unknown signal of interest $x_0 \in \mathbf{R}^n$ by a linear transformation $y = Ax_0$, with a known matrix A of size $d \times n$. An important factor in these applications is that $d < n$, so the system of equations $y = Ax$ is underdetermined. We solve the convex optimization problem

$$(P_1) \qquad \min_x \|x\|_1 \;\text{ subject to }\; y = Ax,$$

obtaining a vector x_1 which we consider an approximation to x_0.

Applications of (P_1) have been proposed in the context of time-frequency representation [9], overcomplete signal representation [20, 26, 27, 31, 32], texture/geometry separation [48, 49], compressed sensing (CS) [14, 7, 56], rapid MR Imaging [37, 38], removal of impulsive noise [21], and decoding of error-correcting codes (ECC) [5, 44]. In such applications, the underlying problem is to obtain a solution to $y = Ax$ which is as sparse as possible, in the sense of having few nonzero entries. The above-cited literature shows that often, when the desired solution x_0 is sufficiently sparse, (P_1) delivers either x_0 or a reasonable approximation.

Traditionally, the problem of finding sparse solutions has been catalogued as belonging to a class of combinatorial optimization problems, whereas (P_1) can be solved by linear programming, and is thus dramatically more tractable in general. Nevertheless, general-purpose LP solvers ultimately involve solution of 'full' $n \times n$ linear systems, and require many applications of such solvers, each application costing order $O(n^3)$ flops.

Many of the interesting problems where (P_1) has been contemplated are very large in scale. For example, already 10 years ago, Basis Pursuit [9] tackled problems with $d = 8,192$ and $n = 262,144$, and problem sizes approached currently [55, 7, 56, 38] are even larger. Such large-scale problems are simply too large for general-purpose strategies to be used in routine processing.

1.1 Greedy Approaches

Many of the applications of (P_1) can instead be attacked heuristically by fitting sparse models, using greedy stepwise least squares. This approach is often called Matching Pursuit or Orthogonal Matching Pursuit (OMP) [12] in the signal processing literature. Rather than minimizing an objective function, OMP constructs a sparse solution to a given problem by iteratively building up an approximation; the vector y is approximated as a linear combination of a few columns of A, where the *active set* of columns to be used is built column by column, in a greedy fashion. At each iteration a new column is added to the active set – the column that best correlates with the current residual.

Although OMP is a heuristic method, in some cases it works marvelously. In particular, there are examples where the data y admit sparse synthesis using only k columns of A, and greedy selection finds exactly those columns in just k steps. Perhaps surprisingly, optimists can cite theoretical work supporting this notion; work by Tropp *et al.* [52, 54] and Donoho *et al.* [19] has shown that OMP can, in certain circumstances, succeed at finding the sparsest solution. Yet, theory provides comfort for pessimists too; OMP fails to find the sparsest solution in certain scenarios where ℓ_1 minimization succeeds [9, 15, 52].

Optimists can also rely on empirical evidence to support their hopeful attitude. By now several research groups have conducted studies of OMP on problems where A is random with

iid Gaussian elements and x_0 is sparse but random (we are aware of experiments by Tropp, by Petukhov, as well as ourselves) all with the conclusion that OMP – despite its greedy and heuristic nature – performs roughly as well as (P_1) in certain problem settings. Later in this paper we document this in detail. However, to our knowledge, there is no theoretical basis supporting these empirical observations.

This has the air of mystery – why do (P_1) and OMP behave so similarly, when one has a rigorous foundation and the other one apparently does not? Is there some deeper connection or similarity between the two ideas? If there is a similarity, why is OMP so fast while ℓ_1 minimization is so slow?

In this paper we aim to shed light on all these questions.

1.2 A Continuation Algorithm To Solve (P_1)

In parallel with developments in the signal processing literature, there has also been interest in the statistical community in fitting regression models while imposing ℓ_1-norm constraints on the regression coefficients. Tibshirani [51] proposed the so-called LASSO problem, which we state using our notation as follows:

$$(L_q) \qquad \min_x \|y - Ax\|_2^2 \ \text{ subject to } \ \|x\|_1 \le q;$$

in words: a least-squares fit subject to an ℓ_1-norm constraint on the coefficients. In Tibshirani's original proposal, $A_{d\times n}$ was assumed to have $d > n$, i.e. representing an overdetermined linear system.

It is convenient to consider instead the unconstrained optimization problem

$$(D_\lambda) \qquad \min_x \|y - Ax\|_2^2/2 + \lambda\|x\|_1,$$

i.e. a form of ℓ_1-penalized least-squares. Indeed, problems (L_q) and (D_λ) are equivalent under an appropriate correspondence of parameters. To see that, associate with each problem (D_λ) : $\lambda \in [0,\infty)$ a solution $\tilde{x}_\lambda$ (for simplicity assumed unique). The set $\{\tilde{x}_\lambda : \lambda \in [0,\infty)\}$ identifies a *solution path*, with $\tilde{x}_\lambda = 0$ for λ large and, as $\lambda \to 0$, $\tilde{x}_\lambda$ converging to the solution of (P_1). Similarly, $\{\tilde{x}_q : q \in [0,\infty)\}$ traces out a solution path for problem (L_q), with $\tilde{x}_q = 0$ for $q = 0$ and, as q increases, $\tilde{x}_q$ converging to the solution of (P_1). Thus, there is a reparametrization $q(\lambda)$ defined by $q(\lambda) = \|\tilde{x}_\lambda\|_1$ so that the solution paths of (D_λ) and $(L_{q(\lambda)})$ coincide.

In the classic overdetermined setting, $d > n$, Osborne, Presnell and Turlach [40, 41] made the useful observation that the solution path of (D_λ), $\lambda \ge 0$ (or (L_q), $q \ge 0$) is polygonal. Further, they characterized the changes in the solution $\tilde{x}_\lambda$ at vertices of the polygonal path. Vertices on this solution path correspond to solution subset models, i.e. vectors having nonzero elements only on a subset of the potential candidate entries. As we move along the solution path, the subset is piecewise constant as a function of λ, changing only at critical values of λ, corresponding to the vertices on the polygonal path. This evolving subset is called the *active set*. Based on these observations, they presented the HOMOTOPY algorithm, which follows the solution path by jumping from vertex to vertex of this polygonal path. It starts at $\tilde{x}_\lambda = 0$ for λ large, with an empty active set. At each vertex, the active set is updated through the addition and removal of variables. Thus, in a sequence of steps, it successively obtains the solutions $\tilde{x}_{\lambda_\ell}$ at a special problem-dependent sequence λ_ℓ associated to vertices of the polygonal path. The name HOMOTOPY refers to the fact that the objective function for (D_λ) is undergoing a homotopy from the ℓ_2 constraint to the ℓ_1 objective as λ decreases.

Malioutov *et al.* [39] were the first to apply the HOMOTOPY method to the formulation (D_λ) in the *underdetermined* setting, when the data are noisy. In this paper, we also suggest using the HOMOTOPY method for solving (P_1) in the underdetermined setting.

Homotopy Algorithm for Solving (P_1): *Apply the* HOMOTOPY *method: follow the solution path from $x_{\lambda_0} = 0$ to $\tilde{x}_0$. Upon reaching the $\lambda = 0$ limit, (P_1) is solved.*

Traditionally, to solve (P_1), one would apply the simplex algorithm or an interior-point method, which, in general, starts out with a dense solution and converges to the solution of (P_1) through a sequence of iterations, each requiring the solution of a full linear system. In contrast, the HOMOTOPY method starts out at $x_{\lambda_0} = 0$, and successively builds a sparse solution by adding or removing elements from its active set. Clearly, in a sparse setting, this latter approach is much more favorable, since, as long as the solution has few nonzeros, HOMOTOPY will reach the solution in a few steps.

Numerically, each step of the algorithm involves the rank-one update of a linear system, and so if the whole procedure stops in k steps, yielding a solution with k nonzeros, its overall complexity is bounded by $k^3 + kdn$ flops. For $k \ll d$ and $d \propto n$, this is far better than the $d^3/3$ flops it would take to solve just one $d \times d$ linear system. Moreover, to solve (D_λ) for all $\lambda \geq 0$ by a traditional approach, one would need to repeatedly solve a quadratic program for every λ value of interest. For any problem size beyond the very smallest, that would be prohibitively time consuming. In contrast, HOMOTOPY delivers all the solutions $\tilde{x}_\lambda$ to (D_λ), $\lambda \geq 0$.

1.3 LARS

Efron, Hastie, Johnstone, and Tibshirani [24] developed an approximation to the HOMOTOPY algorithm which is quite instructive. The HOMOTOPY algorithm maintains an active set of nonzero variables composing the current solution. When moving to a new vertex of the solution polygon, the algorithm may either add new elements to or remove existing elements from the active set. The LARS procedure is obtained by following the same sequence of steps, only omitting the step that considers removal of variables from the active set, thus constraining its behavior to adding new elements to the current approximation. In other words, once activated, a variable is never removed.

In modifying the stepwise rule, of course, one implicitly obtains a new polygonal path, the LARS path, which in general may be different from the LASSO path. Yet, Efron et al. observed that in practice, the LARS path is often identical to the LASSO path. This equality is very interesting in the present context, because LARS is so similar to OMP. Both algorithms build up a model a step at a time, adding a new variable to the active set at each step, and ensuring that the new variable is in some sense the most important among the potential candidate variables. The details of determining importance differ but in both cases involve the inner product between the candidate new variables and the current residual.

In short, a stepwise algorithm with a greedy flavor can sometimes produce the same result as full-blown ℓ_1 minimization. This suggests a possibility which can be stated in two different ways:

- ... that an algorithm for quasi ℓ_1 minimization runs just as rapidly as OMP.
- ... that an algorithm visibly very similar to OMP can be just as effective as ℓ_1 minimization.

1.4 Properties of Homotopy

Another possibility comes to mind. The HOMOTOPY algorithm is, algorithmically speaking, a variant of LARS, differing only in a few lines of code. And yet this small variation makes HOMOTOPY rigorously able to solve a global optimization problem.

More explicitly, the difference between HOMOTOPY and LARS lies in the provision by HOMOTOPY for terms to leave as well as enter the active set. This means that the number of steps required by HOMOTOPY can, in principle, be significantly greater than the number of steps required by LARS, as terms enter and leave the active set numerous times. If so, we observe "model churning" which causes HOMOTOPY to run slowly. We present evidence that under favorable conditions, such churning does not occur. In such cases, the HOMOTOPY algorithm is roughly as fast as both LARS and OMP.

In this paper, we consider two settings for performance measurement: deterministic incoherent matrices and random matrices. We demonstrate that, when a k-sparse representation exists, $k \ll d$, the HOMOTOPY algorithm finds it in k steps. Results in each setting parallel existing results about OMP in that setting. Moreover, the discussion above indicates that each step of HOMOTOPY is identical to a step of LARS and therefore very similar to a corresponding step of OMP. We interpret this parallelism as follows.

> *The* HOMOTOPY *algorithm, in addition to solving the ℓ_1 minimization problem, runs just as rapidly as the heuristic algorithms* OMP/ LARS*, in those problem suites where those algorithms have been shown to correctly solve the sparse representation problem.*

Since there exists a wide range of cases where ℓ_1-minimization is known to find the sparsest solution while OMP is known to fail [9, 15, 52], HOMOTOPY gives in some sense the best of both worlds by a single algorithm: speed where possible, sparsity where possible.

In addition, our viewpoint sheds new perspective on previously-observed similarities between results about ℓ_1 minimization and about OMP. Previous theoretical work [19, 52, 54] found that these two seemingly disparate methods had comparable theoretical results for finding sparse solutions. Our work exhibits inherent similarities between the two methods which motivate these parallel findings.

Figure 1 summarizes the relation between ℓ_1 minimization and OMP, by highlighting the intimate bonds between HOMOTOPY, LARS, and OMP. We elaborate on these inherent connections in a later section.

1.5 Main Results

Our results about the stopping behavior of HOMOTOPY require some terminology.

Definition 1 *A problem suite $\mathcal{S}(\mathrm{E},\mathrm{V};d,n,k)$ is a collection of problems defined by three components: (a) an ensemble E of matrices A of size $d \times n$; (b) an ensemble V of n-vectors α_0 with at most k nonzeros; (c) an induced ensemble of left-hand sides $y = A\alpha_0$.*

Where the ensemble V is omitted from the definition of a problem suite, we interpret that as saying that the relevant statements hold for any nonzero distribution. Otherwise, V will take one of three values: (UNIFORM), when the nonzeros are iid uniform on $[0, 1]$; (GAUSS), when the nonzeros are iid standard normal; and (BERNOULLI), when the nonzeros are iid equi-probable Bernoulli distributed with values ± 1.

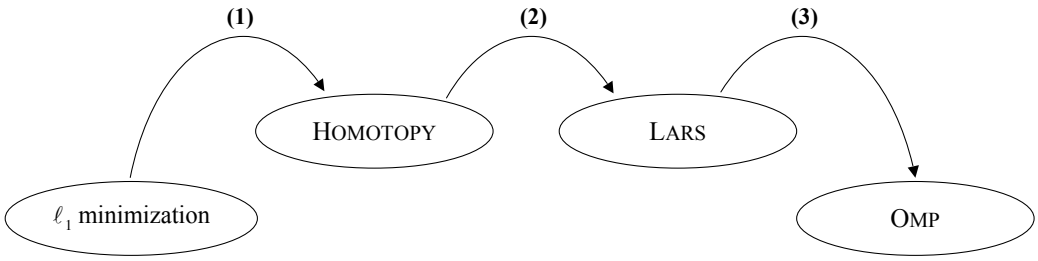

Figure 1: Bridging ℓ_1 minimization and OMP. **(1)** HOMOTOPY provably solves ℓ_1 minimization problems [24]. **(2)** LARS is obtained from HOMOTOPY by removing the sign constraint check. **(3)** OMP and LARS are similar in structure, the only difference being that OMP solves a least-squares problem at each iteration, whereas LARS solves a linearly-penalized least-squares problem. There are initially surprising similarities of results concerning the ability of OMP and ℓ_1 minimization to find sparse solutions. We view those results as statements about the k-step solution property. We present evidence showing that for sufficiently small k, steps **(2)** and **(3)** do not affect the threshold for the k-sparse solution property. Since both algorithms have the k-step solution property under the given conditions, they both have the sparse solution property under those conditions.

Definition 2 *An algorithm $\mathcal{A}$ has the* k-step solution property *at a given problem instance (A, y) drawn from a problem suite $\mathcal{S}(\mathrm{E},\mathrm{V}; d, n, k)$ if, when applied to the data (A, y), the algorithm terminates after at most k steps with the correct solution.*

Our main results establish the k-step solution property in specific problem suites.

1.5.1 Incoherent Systems

The *mutual coherence* $M(A)$ of a matrix A whose columns are normalized to unit length is the maximal off-diagonal entry of the Gram matrix $A^T A$. We call the collection of matrices A with $M(A) \le \mu$ the *incoherent ensemble* with coherence bound μ (denoted INC_μ). Let $\mathcal{S}(\mathrm{INC}_\mu; d, n, k)$ be the suite of problems with $d \times n$ matrices drawn from the incoherent ensemble INC_μ, with vectors α_0 having $\|\alpha_0\|_0 \le k$. For the incoherent problem suite, we have the following result:

Theorem 1 *Let (A, y) be a problem instance drawn from $\mathcal{S}(\mathrm{INC}_\mu; d, n, k)$. Suppose that*

$$k \le (\mu^{-1} + 1)/2. \tag{1.1}$$

Then, the HOMOTOPY *algorithm runs k steps and stops, delivering the solution α_0.*

The condition (1.1) has appeared elsewhere; work by Donoho *et al.* [20, 18], Gribonval and Nielsen [31], and Tropp [52] established that when (1.1) holds, the sparsest solution is unique and equal to α_0, and both ℓ^1 minimization and OMP recover α_0. In particular, OMP takes at most k steps to reach the solution.

In short we show here that for the general class of problems $\mathcal{S}(\mathrm{INC}_\mu; d, n, k)$, where a unique sparsest solution is known to exist, and where OMP finds it in k steps, HOMOTOPY finds that

same solution in the same number of steps. Note that HOMOTOPY always solves the ℓ_1 minimization problem; the result shows that it operates particularly rapidly over the sparse incoherent suite.

1.5.2 Random Matrices

We first consider $d \times n$ random matrices A from the *Uniform Spherical Ensemble* (USE). Such matrices have columns independently and uniformly distributed on the sphere S^{d-1}. Consider the suite $\mathcal{S}(\text{USE}; d, n, k)$ of random problems where A is drawn from the USE and where α_0 has at most k nonzeros at k randomly-chosen sites, with the sites chosen independently of A.

Empirical Finding 1 *Fix $\delta \in (0,1)$. Let $d = d_n = \lfloor \delta n \rfloor$. Let (A, y) be a problem instance drawn from $\mathcal{S}(\text{USE}; d, n, k)$. There is $\epsilon_n > 0$ small, so that, with high probability, for*

$$k \leq \frac{d_n}{2\log(n)}(1 - \epsilon_n), \tag{1.2}$$

the HOMOTOPY *algorithm runs k steps and stops, delivering the solution α_0.*

Remarks:

- We conjecture that corresponding to this empirical finding is a theorem, in which the conclusion is that as $n \to \infty$, $\epsilon_n \to 0$.
- The empirical result is in fact stronger. We demonstrate that for problem instances with k emphatically not satisfying (1.2), but rather

  $$k \geq \frac{d_n}{2\log(n)}(1 + \epsilon_n),$$

  then, with high probability for large n, the HOMOTOPY algorithm fails to terminate in k steps. Thus, (1.2) delineates a boundary between two regions in (k, d, n) space; one where the k-step property holds with high probability, and another where the chance of k-step termination is very low.
- This empirical finding also holds for matrices drawn from the Random Signs Ensemble (RSE). Matrices in this ensemble are constrained to have their entries ± 1 (suitably normalized to obtain unit-norm columns), with signs drawn independently and with equal probability.

We next consider ensembles made of partial orthogonal transformations. In particular, we consider the following matrix ensembles:

- *Partial Fourier Ensemble (PFE).* Matrices in this ensemble are obtained by sampling at random d rows of an n by n Fourier matrix.
- *Partial Hadamard Ensemble (PHE).* Matrices in this ensemble are obtained by sampling at random d rows of an n by n Hadamard matrix.
- *Uniform Random Projection Ensemble (URPE).* Matrices in this ensemble are obtained by generating an n by n random orthogonal matrix and sampling d of its rows at random.

As it turns out, when applied to problems drawn from these ensembles, the HOMOTOPY algorithm maintains the k-step property at a substantially greater range of signal sparsities. In detail, we give evidence supporting the following conclusion.

Empirical Finding 2 *Fix $\delta \in (0,1)$. Let $d = d_n = \lfloor \delta n \rfloor$. Let (A, y) be a problem instance drawn from $\mathcal{S}(\mathrm{E},\mathrm{V}; d, n, k)$, with $E \in \{$ PFE,PHE,URPE $\}$, and $V \in \{$ UNIFORM,GAUSS,BERNOULLI $\}$. The empirical function $\hat{\rho}(\delta)$ depicted in Figure 8, marks a transition such that, for $k/d_n \leq \hat{\rho}(\delta)$, the* HOMOTOPY *algorithm runs k steps and stops, delivering the solution α_0.*

A third, and equally important, finding we present involves the behavior of the HOMOTOPY algorithm when the k-step property does not hold. We provide evidence to the following.

Empirical Finding 3 *Let (A, y) be a problem instance drawn from $\mathcal{S}(\mathrm{E},\mathrm{V}; d, n, k)$, with $E \in \{$ PFE,PHE,URPE $\}$, $V \in \{$ UNIFORM,GAUSS,BERNOULLI $\}$, and (d, n, k) chosen so that the k-step property does not hold. With high probability,* HOMOTOPY*, when applied to (A, y), operates in either of two modes:*

1. *It recovers the sparse solution α_0 in $c_s \cdot d$ iterations, with c_s a constant depending on E, V, empirically found to be less than* 1.6*.*
2. *It does not recover α_0, returning a solution to (P_1) in $c_f \cdot d$ iterations, with c_f a constant depending on E, V, empirically found to be less than* 4.85*.*

We interpret this finding as saying that we may safely apply HOMOTOPY to problems whose solutions are not known *a priori* to be sparse, and still obtain a solution rapidly. In addition, if the underlying solution to the problem is indeed sparse, HOMOTOPY is particularly efficient in finding it. This finding is of great usefulness in practical applications of HOMOTOPY.

1.6 An Illustration: Error Correction in a Sparse Noisy Channel

The results presented above have far-reaching implications in practical scenarios. We demonstrate these with a stylized problem inspired by digital communication systems, of correcting gross errors in a digital transmission. Recently, researchers pointed out that ℓ_1-minimization/sparsity ideas have a role to play in decoding linear error-correcting codes over $\mathbf{Z}$ [5, 44], a problem known, in general, to be NP-hard. Applying the HOMOTOPY algorithm in this scenario, we now demonstrate the following remarkable property, an implication of Empirical Finding 2.

Corollary 1 *Consider a communications channel corrupting a fraction $\epsilon < 0.24$ of every n transmitted integers, the corrupted sites chosen by a Bernoulli(ϵ) iid selection. For large n, there exists a rate-*1/5 *code, that, with high probability, can be decoded, without error, in at most* $.24n$ HOMOTOPY *steps.*

In other words, even when as many as a quarter of the transmitted entries are corrupted, HOMOTOPY operates at peak performance, recovering the encoded bits in a fixed number of steps, known in advance. This feature enables the design of real-time decoders obeying a hard computational budget constraint, with a fixed number of flops per decoding effort.

Let us describe the coding strategy in more detail. Assume θ is a digital signal of length p, with entries ± 1, representing bits to be transmitted over a digital communication channel. Prior to transmission, we encode θ with a rate 1/5 code constructed in the following manner. Let H be a $n \times n$ Hadamard matrix, $n \approx 5p$. Construct the 'encoding matrix' E by selecting

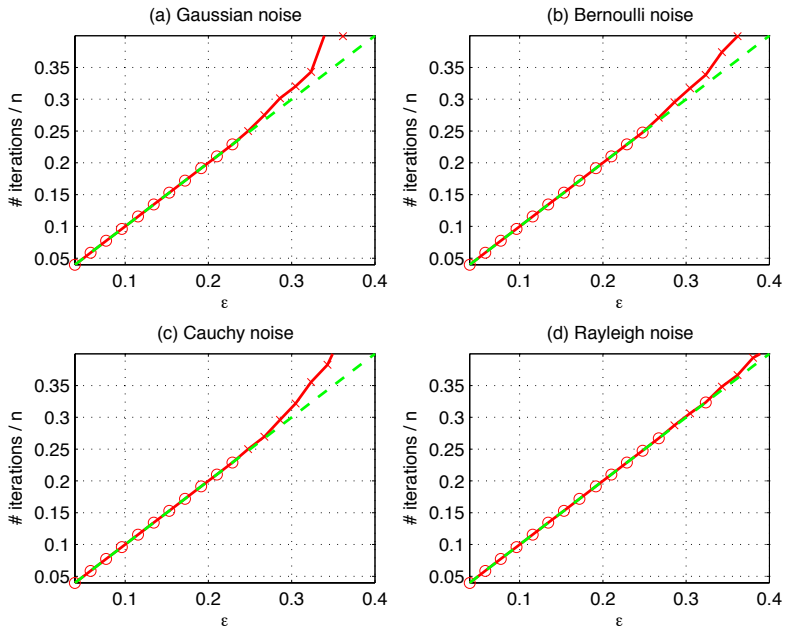

Figure 2: Performance of HOMOTOPY in decoding a rate-1/5 partial Hadamard code. Each panel shows the number of iterations, divided by n, that HOMOTOPY takes in order to recover the coded signal, versus ϵ, the fraction of corrupt entries in the transmitted signal, with the noise distributed (a) N(0,1); (b) ± 1 with equal probabilities; (c) Cauchy; (d) Rayleigh. In each plot 'o' indicates k-step recovery, and 'x' implies recovery in more than k steps.

p rows of H at random. The 'decoding matrix' D is then composed of the other $4p$ rows of H. The encoding stage amounts to computing $x = E^T\theta$, with x the encoded signal, of length n. At the decoder, we receive $r = x + z$, where z has nonzeros in k random positions, and the nonzero follow a specified distribution. At the decoder, we apply HOMOTOPY to solve

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

Call the solution $\hat{\alpha}$. The decoder output is then

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

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

To demonstrate our claim, we set $p = 256$, and considered error rates $k = \epsilon n$, with ϵ varying in $[0.04, 0.4]$. In each case we generated a problem instance as described above, and measured the number of steps HOMOTOPY takes to reach the correct solution. Results are plotted in Figure 2, showing four different noise distributions. Inspecting the results, we make two important observations. First, as claimed, for the noise distributions considered, as long as the fraction of corrupt entries is less that $0.24n$, HOMOTOPY recovers the correct solution in k steps, as implied by Empirical Finding 2. Second, as ϵ increases, the k-step property may begin to fail, but HOMOTOPY still does not take many more than k steps, in accord with Empirical Finding 3.

1.7 Relation to Earlier Work

Our results seem interesting in light of previous work by Osborne et al. [40] and Efron et al. [24]. In particular, our proposal for fast solution of ℓ_1 minimization amounts simply to applying to (P_1) previously known algorithms (HOMOTOPY a.k.a LARS/LASSO) designed for fitting equations to noisy data. We make the following contributions.

- *Few Steps.* Efron et al. considered the overdetermined noisy setting, and remarked in passing that while, in principle HOMOTOPY could take many steps, they had encountered examples where it took a direct path to the solution, in which a term, once entered into the active set, stayed in the active set [24]. Our work in the noiseless underdetermined setting formally identifies a precise phenomenon, namely, the k-step solution property, and delineates a range of problem suites where this phenomenon occurs.

- *Similarity of* HOMOTOPY *and* LARS. Efron et al., in the overdetermined noisy setting, commented that, while in principle the solution paths of HOMOTOPY and LARS could be different, in practice they were came across examples where HOMOTOPY and LARS yielded very similar results [24]. Our work in the noiseless case formally defines a property which implies that HOMOTOPY and LARS have the same solution path, and delineates a range of problem suites where this property holds. In addition, we present simulation studies showing that, over a region in parameter space, where ℓ^1 minimization recovers the sparsest solution, LARS also recovers the sparsest solution.

- *Similarity of* HOMOTOPY *and* OMP. In the random setting, a result of Tropp and Gilbert [54] as saying that OMP recovers the sparsest solution under a condition similar to (1.2), although with a somewhat worse constant term; their result (and proof) can be interpreted in the light of our work as saying that OMP actually has the k-step solution property under their condition.

  In fact it has the k-step solution property under the condition (1.2). Below, we elaborate on the connections between HOMOTOPY and OMP leading to these similar results.

- *Similarity of* HOMOTOPY *and Polytope Faces Pursuit.* Recently, Plumbley introduced a greedy algorithm to solve (P_1) in the dual space, which he named *Polytope Faces Pursuit* (PFP) [43, 42]. The algorithm bears strong resemblance to HOMOTOPY and OMP. Below, we elaborate on Plumbley's work, to show that the affinities are not coincidental, and in fact, under certain conditions, PFP is equivalent to HOMOTOPY. We then conclude that PFP maintains the k-step solution property under the same conditions required for HOMOTOPY.

In short, we provide theoretical underpinnings, formal structure, and empirical findings. We also believe our viewpoint clarifies the connections between HOMOTOPY, LARS, and PFP.

1.8 Contents

The paper is organized as follows. Section 2 reviews the HOMOTOPY algorithm in detail. Section 3 presents running-time simulations alongside a formal complexity analysis of the algorithm, and discusses evidence leading to Empirical Finding 3. In Section 4 we prove Theorem 1, the k-step solution property for the sparse incoherent problem suite. In Section 5 we demonstrate Empirical Finding 1, the k-step solution property for the sparse USE problem suite. Section 6 follows, with evidence to support Empirical Finding 2. In Section 7 we elaborate on the relation between ℓ_1

minimization and OMP hinted at in Figure 1. This provides a natural segue to Section 8, which discusses the connections between HOMOTOPY and PFP. Section 9 then offers a comparison of the performance of HOMOTOPY, LARS, OMP and PFP in recovering sparse solutions. In Section 10, we demonstrate the applicability of HOMOTOPY and LARS with examples inspired by NMR spectroscopy and MR imaging. Section 11 discusses the software accompanying this paper. Section 12 briefly discusses approximate methods for recovery of sparse solutions, and a final section has concluding remarks.

2 The Homotopy Algorithm

We shall start our exposition with a description of the HOMOTOPY algorithm. Recall that the general principle undergirding homotopy methods is to trace a solution path, parametrized by one or more variables, while evolving the parameter vector from an initial value, for which the corresponding solution is known, to the desired value. For the LASSO problem (D_λ), this implies following the solution x_λ, starting at λ large and $x_\lambda = 0$, and terminating when $\lambda \to 0$ and x_λ converging to the solution of (P_1). The solution path is followed by maintaining the optimality conditions of (D_λ) at each point along the path.

Specifically, let $f_\lambda(x)$ denote the objective function of (D_λ). By classical ideas in convex analysis, a necessary condition for x_λ to be a minimizer of $f_\lambda(x)$ is that $0 \in \partial_x f_\lambda(x_\lambda)$, i.e. the zero vector is an element of the subdifferential of f_λ at x_λ. We calculate

$$\partial_x f_\lambda(x_\lambda) = -A^T(y - Ax_\lambda) + \lambda \partial \|x_\lambda\|_1, \tag{2.3}$$

where $\partial\|x_\lambda\|_1$ is the subgradient

$$\partial\|x_\lambda\|_1 = \left\{ u \in R^n \;\middle|\; \begin{array}{ll} u_i = \mathrm{sgn}(x_{\lambda,i}), & x_{\lambda,i} \neq 0 \\ u_i \in [-1,1], & x_{\lambda,i} = 0 \end{array} \right\}.$$

Let $I = \{i : x_\lambda(i) \neq 0\}$ denote the support of x_λ, and call $c = A^T(y - Ax_\lambda)$ the vector of *residual correlations*. Then, equation (2.3) can be written equivalently as the two conditions

$$c(I) = \lambda \cdot \mathrm{sgn}(x_\lambda(I)), \tag{2.4}$$

and

$$|c(I^c)| \leq \lambda, \tag{2.5}$$

In words, residual correlations on the support I must all have magnitude equal to λ, and signs that match the corresponding elements of x_λ, whereas residual correlations off the support must have magnitude less than or equal to λ. The HOMOTOPY algorithm now follows from these two conditions, by tracing out the optimal path x_λ that maintains (2.4) and (2.5) for all $\lambda \geq 0$. The key to its successful operation is that the path x_λ is a piecewise linear path, with a discrete number of vertices [24, 40].

The algorithm starts with an initial solution $x_0 = 0$, and operates in an iterative fashion, computing solution estimates x_ℓ, $\ell = 1, 2, \ldots$. Throughout its operation, it maintains the *active set* I, which satisfies

$$I = \{j : |c_\ell(j)| = \|c_\ell\|_\infty = \lambda\}, \tag{2.6}$$

as implied by conditions (2.4) and (2.5). At the ℓ-th stage, HOMOTOPY first computes an update direction d_ℓ, by solving

$$A_I^T A_I d_\ell(I) = \mathrm{sgn}(c_\ell(I)), \tag{2.7}$$

with d_ℓ set to zero in coordinates not in I. This update direction ensures that the magnitudes of residual correlations on the active set all decline equally. The algorithm then computes the step size to the next breakpoint along the homotopy path. Two scenarios may lead to a breakpoint. First, that a non-active element of c_ℓ would increase in magnitude beyond λ, violating (2.5). This first occurs when

$$\gamma_\ell^+ = \min_{i \in I^c} \left\{ \frac{\lambda - c_\ell(i)}{1 - a_i^T v_\ell}, \frac{\lambda + c_\ell(i)}{1 + a_i^T v_\ell} \right\}, \tag{2.8}$$

where $v_\ell = A_I d_\ell(I)$, and the minimum is taken only over positive arguments. Call the minimizing index i^+. The second scenario leading to a breakpoint in the path occurs when an active coordinate crosses zero, violating the sign agreement in (2.4). This first occurs when

$$\gamma_\ell^- = \min_{i \in I} \{-x_\ell(i)/d_\ell(i)\}, \tag{2.9}$$

where again the minimum is taken only over positive arguments. Call the minimizing index i^-. HOMOTOPY then marches to the next breakpoint, determined by

$$\gamma_\ell = \min\{\gamma_\ell^+, \gamma_\ell^-\}, \tag{2.10}$$

updates the active set, by either appending I with i^+, or removing i^-, and computes a solution estimate

$$x_\ell = x_{\ell-1} + \gamma_\ell d_\ell.$$

The algorithm terminates when $\|c_\ell\|_\infty = 0$, and the solution of (P_1) has been reached.

Remarks:

1. In the algorithm description above, we implicitly assume that at each breakpoint on the homotopy path, at most one new coordinate is considered as candidate to enter the active set (Efron *et al.* called this the "one at a time" condition [24]). If two or more vectors are candidates, more care must be taken in order to choose the correct subset of coordinates to enter the active set; see [24] for a discussion.

2. In the introduction, we noted that the LARS scheme closely mimics HOMOTOPY, the main difference being that LARS does not allow removal of elements from the active set. Indeed, to obtain the LARS procedure, one simply follows the sequence of steps described above, omitting the computation of i^- and γ_ℓ^-, and replacing (2.10) with

   $$\gamma_\ell = \gamma_\ell^+.$$

   The resulting scheme adds a single element to the active set at each iteration, never removing active elements from the set.

3. The HOMOTOPY algorithm may be easily adapted to deal with noisy data. Assume that rather than observing $y_0 = A\alpha_0$, we observe a noisy version $y = A\alpha_0 + z$, with $\|z\|_2 \leq \epsilon_n$. Donoho *et al.* [19, 16] have shown that, for certain matrix ensembles, the solution x_q of (L_q) with $q = \epsilon_n$ has an error which is at worst proportional to the noise level. To solve for x_{ϵ_n}, we simply apply the HOMOTOPY algorithm as described above, terminating as soon as the residual satisfies $\|r_\ell\| \leq \epsilon_n$. Since in most practical scenarios it is not sensible to assume that the measurements are perfectly noiseless, this attribute of HOMOTOPY makes it an apt choice for use in practice.

3 Computational Cost

In earlier sections, we claimed that HOMOTOPY can solve the problem (P_1) much faster than general-purpose LP solvers, which are traditionally used to solve it. In particular, when the k-step solution property holds, HOMOTOPY runs in a fraction of the time it takes to solve one full linear system, making it as efficient as fast stepwise algorithms such as OMP. To support these assertions, we now present the results of running-time simulations, complemented by a formal analysis of the asymptotic complexity of HOMOTOPY.

3.1 Running Times

To evaluate the performance of the HOMOTOPY algorithm applied to (P_1), we compared its running times on instances of the problem suite $\mathcal{S}(\text{USE},\text{GAUSS}; d, n, k)$ with two state-of-the-art algorithms for solving Linear Programs. The first, LP_Solve, is a Mixed Integer Linear Programming solver, implementing a variant of the Simplex algorithm [2]. The other, PDCO, a Primal-Dual Convex Optimization solver, is a log-barrier interior point method, written by Michael Saunders of the Stanford Optimization Laboratory [45]. Table 1 shows the running times for HOMOTOPY, LP_Solve and PDCO, for various problem dimensions. The figures appearing in the table were measured on a 3GHz Xeon workstation.

(d,n,k)	HOMOTOPY	LP_Solve	PDCO
(200,500,20)	0.03	2.04	0.90
(250,500,100)	0.94	9.27	1.47
(500,1000,100)	1.28	45.18	4.62
(750,1000,150)	2.15	85.78	7.68
(500,2000,50)	0.34	59.52	6.86
(1000,2000,200)	7.32	407.99	22.90
(1600,4000,320)	31.70	2661.42	122.47

Table 1: Comparison of execution times (in seconds) of HOMOTOPY, LP_Solve and PDCO, applied to instances of the random problem suite $\mathcal{S}(\text{USE},\text{GAUSS}; d, n, k)$.

Examination of the running times in the table suggests two important observations. First, when the underlying solution to the problem (P_1) is sparse, a tremendous saving in computation time is achieved using HOMOTOPY, compared to traditional LP solvers. For instance, when A is 500×2000, and y admits sparse synthesis with $k = 50$ nonzeros, HOMOTOPY terminates in about 0.34 seconds, 20 times faster than PDCO, and over 150 times faster than LP_Solve. Second, when the linear system is highly underdetermined (i.e. d/n is small), even when the solution is not sparse, HOMOTOPY is more efficient than either LP_Solve or PDCO. This latter observation is of particular importance for applications, as it implies that HOMOTOPY may be 'safely' used to solve ℓ_1 minimization problems even when the underlying solution is not known to be sparse.

3.2 Complexity Analysis

The timing studies above are complemented by a detailed analysis of the complexity of the HOMOTOPY algorithm. We begin by noting that the bulk of computation is invested in the solution of the linear system (2.7) at each iteration. Thus, the key to an efficient implementation is maintaining a Cholesky factorization of the gram minor $A_I^T A_I$, updating it with the

addition/removal of elements to/from the active set. This allows for fast solution for the update direction; $O(d^2)$ operations are needed to solve (2.7), rather than the $O(d^3)$ flops it would ordinarily take. In detail, let $\ell = |I|$, where I denotes the current active set. The dominant calculations per iteration are the solution of (2.7) at a cost of $2\ell^2$ flops, and computation of the step to the next vertex on the solution path, using $nd + 6(n - \ell)$ flops. In addition, the Cholesky factor of $A_I^T A_I$ is either updated by appending a column to A_I at a cost of $\ell^2 + \ell d + 2(\ell + d)$ flops, or 'downdated' by removing a column of A_I using at most $3\ell^2$ flops.

To conclude, without any sparsity constraints on the data, k HOMOTOPY steps would take at most $4kd^2/3 + kdn + O(kn)$ flops. However, if the conditions for the k-step solution property are satisfied, a more favorable estimate holds.

Lemma 1 *Let (A, y) be a problem instance drawn from a suite $\mathcal{S}$(E,V;d, n, k). Suppose the k-step solution property holds, and the* HOMOTOPY *algorithm performs k steps, each time adding a single element into the active set. The algorithm terminates in $k^3 + kdn + 1/2k^2(d-1) + n(8k + d + 1) + 1/2k(5d - 3)$ flops.*

Notice that, for $k \ll d$, the dominant term in the expression for the operation count is kdn, the number of flops needed to carry out k matrix-vector multiplications. Thus, we may interpret Lemma 1 as saying that, under such favorable conditions, the computational workload of HOMOTOPY is roughly proportional to k applications of a $d \times n$ matrix. For comparison, applying least-squares to solve the underdetermined linear system $Ax = y$ would require $2d^2n - 2d^3/3$ operations [30]. Thus, for $k \ll d$, HOMOTOPY runs in a fraction of the time it takes to solve one least-squares system.

To visualize this statement, panel (a) of Figure 3 displays the operation count of HOMOTOPY on a grid with varying sparsity and indeterminacy factors. In this simulation, we measured the total operation count of HOMOTOPY as a fraction of the solution of one $d \times n$ least-squares system, for random instances of the problem suite $\mathcal{S}$(USE,UNIFORM;d, n, k). We fixed $n = 1000$, varied the indeterminacy of the system, indexed by $\delta = d/n$, in the range $[0.1, 1]$, and the sparsity of the solution, indexed by $\rho = k/d$, in the range $[0.05, 1]$. For reference, we superposed the theoretical bound ρ_W below which the solution of (P_1) is, with high probability, the sparsest solution (see section 9 for more details). Close inspection of this plot reveals that, below this curve, HOMOTOPY delivers the solution to (P_1) rapidly, much faster than it takes to solve one $d \times n$ least-squares problem.

3.3 Number of Iterations

Considering the analysis just presented, it is clear that the computational efficiency of HOMOTOPY greatly depends on the number of vertices on the polygonal HOMOTOPY path. Indeed, since it allows removal of elements from the active set, HOMOTOPY may, conceivably, require an arbitrarily large number of iterations to reach a solution. We note that this property is not shared by LARS or OMP; owing to the fact that these algorithms never remove elements from the active set, after d iterations they terminate with zero residual.

In [24], Efron *et al.* briefly noted that they had observed examples where HOMOTOPY does not 'drop' elements from the active set very frequently, and so, overall, idoesn't require many more iterations than LARS to reach a solution. We explore this initial observation further, and present evidence leading to Empirical Finding 3. Specifically, consider the problem suite $\mathcal{S}$(E,V;d, n, k), with E $\in$ {USE,PFE,PHE,URPE }, and V $\in$ { UNIFORM,GAUSS,BERNOULLI }. The space (d, n, k) of dimensions of underdetermined problems may be divided into three regions:

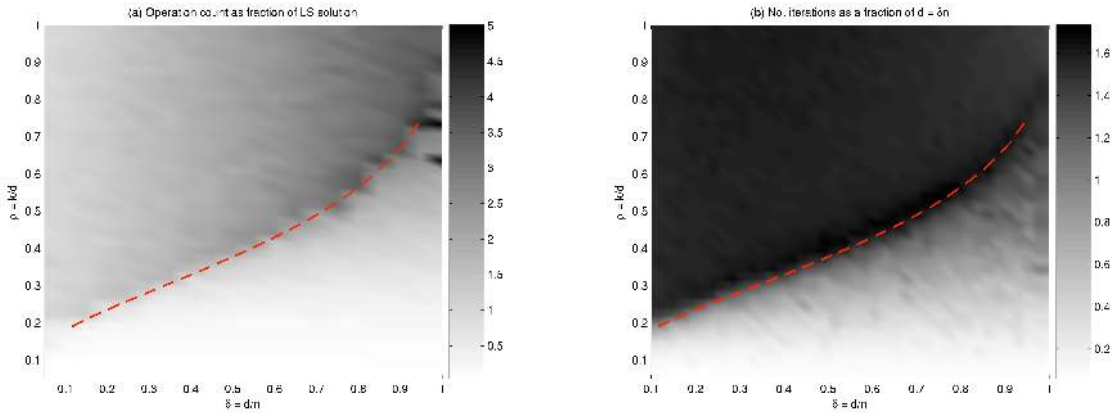

Figure 3: Computational Cost of HOMOTOPY. Panel (a) shows the operation count as a fraction of one least-squares solution on a ρ-δ grid, with $n = 1000$. Panel (b) shows the number of iterations as a fraction of $d = \delta \cdot n$. The superimposed dashed curve depicts the curve ρ_W, below which HOMOTOPY recovers the sparsest solution with high probability.

1. *k-step region.* When (d, n, k) are such that the k-step property holds, then HOMOTOPY successfully recovers the sparse solution in k steps, as suggested by Empirical Findings 1 and 2. Otherwise;

2. *k-sparse region.* When (d, n, k) are such that ℓ_1 minimization correctly recovers k-sparse solutions but HOMOTOPY takes more than k steps, then, with high probability, it takes no more than $c_s \cdot d$ steps, with c_s a constant depending on E, V, empirically found to be ~ 1.6. Otherwise;

3. *Remainder.* With high probability, HOMOTOPY does not recover the sparsest solution, returning a solution to (P_1) in $c_f \cdot d$ steps, with c_f a constant depending on E, V, empirically found to be ~ 4.85.

We note that the so-called "constants" c_s, c_f depend weakly on δ and n.

A graphical depiction of this division of the space of admissible (d, n, k) is given in panel (b) of Figure 3. It shows the number of iterations HOMOTOPY performed for various (d, n, k) configurations, as a shaded attribute on a grid indexed by δ and ρ. Inspection of this plot reveals that HOMOTOPY performs at most $\sim 1.6 \cdot d$ iterations, regardless of the underlying solution sparsity. In particular, in the region below the curve ρ_W, where, with high probablity, HOMOTOPY recovers the sparsest solution, it does so in less than d steps.

More extensive evidence is given in Table 2, summarizing the results of a comprehensive study. We considered four matrix ensembles E, each coupled with three nonzero ensembles V. For a problem instance drawn from a $\mathcal{S}$(E,V; d, n, k), we recorded the number of iterations required to reach a solution. We repeated this at many different (d, n, k) configurations, generating 100 independent realizations for each (d, n, k), and computing the average number of iterations observed at each instance. Table 2 displays the estimated constants c_s, c_f for different combinations of matrix ensemble and coefficient ensemble. Thus, the results in Table 2 read, e.g., 'Applied to a problem instance drawn from $\mathcal{S}$(USE,UNIFORM; d, n, k), HOMOTOPY takes, with high probability, no more than $1.69 \cdot d$ iterations to obtain the minimum ℓ_1 solution'.

Coeff. Ensemble / Matrix Ensemble	Uniform		Gauss		Bernoulli	
	c_s	c_f	c_s	c_f	c_s	c_f
USE	1.65	1.69	1.6	1.7	1.23	1.68
PFE	0.99	3.44	0.99	1.42	0.86	1.49
PHE	0.99	4.85	0.92	1.44	0.87	1.46
URPE	1.05	4.4	1	1.46	1	1.44

Table 2: Maximum number of Homotopy iterations, as a fraction of d, for various matrix / coefficient ensembles.

4 Fast Solution with the Incoherent Ensemble

Let the $d \times n$ matrix A have unit-length columns a_j, $\|a_j\|_2 = 1$. The mutual coherence

$$M(A) = \max_{i \neq j} |\langle a_i, a_j \rangle|$$

measures the smallest angle between any pair of columns. As $n > d$, this angle must be greater than zero: the columns cannot be mutually orthogonal; in fact, there is an established lower bound on the coherence of A:

$$M(A) \geq \sqrt{\frac{n-d}{d(n-1)}},$$

Matrices which satisfy this lower bound with equality are known as optimal Grassmannian frames [50]. There is by now much work establishing relations between the sparsity of a coefficient vector α_0 and the coherence of a matrix A needed for successful recovery of α_0 via ℓ_1 minimization [20, 18, 53, 31] or Omp [19, 52]. In detail, for a general matrix A with coherence $M(A)$, both ℓ_1 minimization and OMP recover the solution α_0 from data $y = A\alpha_0$ whenever the number of nonzeros in α_0 satisfies

$$\|\alpha_0\|_0 < (M(A)^{-1} + 1)/2. \tag{4.1}$$

see, for example, Theorem 7 of [18] or Corollary 3.6 of [52]. Comparing (4.1) with (1.1), we see that Theorem 1 essentially states that a parallel result holds for the Homotopy algorithm.

Before proceeding to the proof of the Theorem, we make a few introductory assumptions. As in the above, we assume that the Homotopy algorithm operates with a problem instance (A, y) as input, with $y = A\alpha_0$ and $\|\alpha_0\|_0 = k$. To simplify notation, we assume, without loss of generality, that α_0 has its nonzeros in the first k positions. Further, we operate under the convention that at each step of the algorithm, only one vector is introduced into the active set. If two or more vectors are candidates to enter the active set, assume the algorithm inserts them one at a time, on separate stages. Finally, to fix notation, let x_ℓ denote the Homotopy solution at the ℓ-th step, $r_\ell = y - Ax_\ell$ denote the residual at that step, and $c_\ell = A^T r_\ell$ be the corresponding residual correlation vector. To prove Theorem 1, we introduce two useful notions.

Definition 3 *We say that the* Homotopy *algorithm has the* Correct Term Selection Property *at a given problem instance (A, y), with $y = A\alpha_0$, if at each iteration, the algorithm selects a new term to enter the active set from the support set of α_0.*

Homotopy has the correct term selection property if, throughout its operation, it builds the solution using only correct terms. Thus, at termination, the support set of the solution is guaranteed to be a subset of the support set of α_0.

Definition 4 *We say that the* HOMOTOPY *algorithm has the* Sign Agreement Property *at a given problem instance* (A, y)*, with* $y = A\alpha_0$*, if at every step* ℓ*, for all* $j \in I$*,*

$$sgn(x_\ell(j)) = sgn(c_\ell(j)).$$

In words, HOMOTOPY has the sign agreement property if, at every step of the algorithm, the residual correlations in the active set agree in sign with the corresponding solution coefficients. This ensures that the algorithm never removes elements from the active set.

These two properties are the fundamental building blocks needed to ensure that the k-step solution property holds. In particular, these two properties are necessary and sufficient conditions for successful k-step termination, as the following lemma shows.

Lemma 2 *Let* (A, y) *be a problem instance drawn from a suite* $\mathcal{S}(\mathrm{E},\mathrm{V}; d, n, k)$*. The* HOMOTOPY *algorithm, when applied to* (A, y)*, has the* k*-step solution property if and only if it has the correct term selection property and the sign agreement property.*

Proof. To argue in the forward direction, we note that, after k steps, correct term selection implies that the active set is a subset of the support of α_0, i.e. $I \subseteq \{1, ..., k\}$. In addition, the sign agreement property ensures that no variables leave the active set. Therefore, after k steps, $I = \{1, ..., k\}$ and the HOMOTOPY algorithm recovers the correct sparsity pattern. To show that after k steps, the algorithm terminates, we note that, at the k-th step, the step-size γ_k is chosen so that, for some $j \in I^c$,

$$|c_k(j) - \gamma_k a_j^T A_I d_k(I)| = \lambda_k - \gamma_k, \tag{4.2}$$

with $\lambda_k = \|c_k(I)\|_\infty$. In addition, for the k-th update, we have

$$A_I d_k(I) = A_I (A_I^T A_I)^{-1} A_I^T r_k = r_k,$$

since r_k is contained in the column space of A_I. Hence

$$c_k(I^c) = A_{I^c}^T A_I d_k(I),$$

and $\gamma_k = \lambda_k$ is chosen to satisfy (4.2). Therefore, the solution at step k has $Ax_k = y$. Since y has a unique representation in terms of the columns of A_I, we may conclude that $x_k = \alpha_0$.

The converse is straightforward. In order to terminate with the solution α_0 after k steps, the HOMOTOPY algorithm must select one term out of $\{1, ..., k\}$ at each step, never removing any elements. Thus, violation of either the correct term selection property or the sign agreement property would result in a number of steps greater than k or an incorrect solution. □

Below, we will show that, when the solution α_0 is sufficiently sparse, both these properties hold as the algorithm traces the solution path. Theorem 1 then follows naturally.

4.1 Correct Term Selection

We now show that, when sufficient sparsity is present, the HOMOTOPY solution path maintains the correct term selection property.

Lemma 3 *Suppose that* $y = A\alpha_0$ *where* α_0 *has only* k *nonzeros, with* k *satisfying*

$$k \leq (\mu^{-1} + 1)/2. \tag{4.3}$$

Assume that the residual at the ℓ-th step can be written as a linear combination of the first k columns in A, i.e.

$$r_\ell = \sum_{j=1}^{k} \beta_\ell(j) a_j.$$

Then the next step of the HOMOTOPY *algorithm selects an index from among the first k.*

Proof. We will show that at the ℓ-th step,

$$\max_{1 \le i \le k} |\langle r_\ell, a_i \rangle| > \max_{i > k} |\langle r_\ell, a_i \rangle|, \tag{4.4}$$

and so at the end of the ℓ-th iteration, the active set is a subset of $\{1, ..., k\}$.

Let $G = A^T A$ denote the gram matrix of A. Let $\hat{i} = \arg\max_{1 \le i \le k} |\beta_i|$. The left hand side of (4.4) is bounded below by

$$\begin{aligned}
\max_{1 \le i \le k} |\langle r_\ell, a_i \rangle| &\ge |\langle r_\ell, a_{\hat{i}} \rangle| \\
&\ge |\sum_{j=1}^{k} \beta_\ell(j) G_{\hat{i}j}| \\
&\ge |\beta_\ell(\hat{i})| - \sum_{j \ne \hat{i}} |G_{\hat{i}j}||\beta_\ell(j)| \\
&\ge |\beta_\ell(\hat{i})| - M(A) \sum_{j \ne \hat{i}} |\beta_\ell(j)| \\
&\ge |\beta_\ell(\hat{i})| - M(A)(k-1)|\beta_\ell(\hat{i})|, \qquad (4.5)
\end{aligned}$$

Here we used: $\|a_j\|_2^2 = 1$ for all j and $G_{\hat{i}j} \le M(A)$ for $j \ne \hat{i}$. As for the right hand side of (4.4), we note that for $i > k$ we have

$$\begin{aligned}
|\langle r_\ell, a_i \rangle| &\le \sum_{j=1}^{k} |\beta_\ell(j)||G_{ij}| \\
&\le M(A) \sum_{j=1}^{k} |\beta_\ell(j)| \\
&\le kM(A)|\beta_\ell(\hat{i})|. \qquad (4.6)
\end{aligned}$$

Combining (4.5) and (4.6), we get that for (4.4) to hold, we need

$$1 - (k-1)M(A) > kM(A). \tag{4.7}$$

Since k was selected to exactly satisfy this bound, relation (4.4) follows. □

Thus, when k satisfies (4.3), the HOMOTOPY algorithm only ever considers indices among the first k as candidates to enter the active set.

4.2 Sign Agreement

Recall that an index is removed from the active set at step ℓ only if condition (2.4) is violated, i.e. if the signs of the residual correlations $c_\ell(i), i \in I$ do not match the signs of the corresponding elements of the solution x_ℓ. The following lemma shows that, when α_0 is sufficiently sparse, an

even stronger result holds: at each stage of the algorithm, the residual correlations in the active set agree in sign with the direction of change of the corresponding terms in the solution. In other words, the solution moves in the right direction at each step. In particular, it implies that throughout the HOMOTOPY solution path, the sign agreement property is maintained.

Lemma 4 *Suppose that $y = A\alpha_0$, where α_0 has only k nonzeros, with k satisfying*

$$k \leq (\mu^{-1} + 1)/2.$$

For $\ell \in \{1, ..., k\}$, let $c_\ell = A^T r_\ell$, and let the active set I be defined as in (2.6). Then the update direction d_ℓ defined by (2.7) satisfies,

$$sgn(d_\ell(I)) = sgn(c_\ell(I)). \tag{4.8}$$

Proof. Let $\lambda_\ell = \|c_\ell\|_\infty$. We will show that for $i \in I$, $|d_\ell(i) - \mathrm{sgn}(c_\ell(i))| < 1$, implying that (4.8) holds. To do so, we note that (2.7) can be rewritten as

$$\lambda_\ell(A_I^T A_I - I_d)d_\ell(I) = -\lambda_\ell d_\ell(I) + c_\ell(I),$$

where I_d is the identity matrix of appropriate dimension. This yields

$$\begin{aligned}
\|\lambda_\ell d_\ell(I) - c_\ell(I)\|_\infty &\leq \|A_I^T A_I - I_d\|_{(\infty,\infty)} \cdot \|\lambda_\ell d_\ell(I)\|_\infty \\
&\leq \frac{1 - M(A)}{2} \|\lambda_\ell d_\ell(I)\|_\infty \\
&\leq \frac{1 - M(A)}{2} (\|c_\ell(I)\|_\infty + \|\lambda_\ell d_\ell(I) - c_\ell(I)\|_\infty) \\
&\leq \frac{1 - M(A)}{2} (\lambda_\ell + \|\lambda_\ell d_\ell(I) - c_\ell(I)\|_\infty),
\end{aligned}$$

where $\|\cdot\|_{(\infty,\infty)}$ denotes the induced ℓ^∞ operator norm. Rearranging terms, we get

$$\|\lambda_\ell d_\ell(I) - c_\ell(I)\|_\infty \leq \lambda_\ell \cdot \frac{1 - M}{1 + M} < \lambda_\ell,$$

thus,

$$\|d_\ell(I) - \mathrm{sgn}(c_\ell(I))\|_\infty < 1.$$

Relation (4.8) follows. □

4.3 Proof of Theorem 1

Lemmas 3 and 4 establish the correct term selection property and the sign agreement property at a single iteration of the algorithm. We will now give an inductive argument showing that the two properties hold at every step $\ell \in \{1, ..., k\}$.

In detail, the algorithm starts with $x_0 = 0$; by the sparsity assumption on $y = A\alpha_0$, we may apply Lemma 3 with $r_0 = y$, to get that at the first step, a column among the first k is selected. Moreover, at the end of the first step we have $x_1 = \gamma_1 d_1$, and so, by Lemma 4,

$$\mathrm{sgn}(x_1(I)) = \sigma_1(I).$$

By induction, assume that at step ℓ, only indices among the first k are in the active set, i.e.

$$r_\ell = \sum_{j=1}^{k} \beta_\ell(j) a_j,$$

and the sign condition (2.4) holds, i.e.

$$\mathrm{sgn}(x_\ell(I)) = \sigma_\ell(I).$$

Applying Lemma 3 to r_ℓ, we get that at the $(l+1)$-th step, the term to enter the active set will be selected from among the first k indices. Moreover, for the updated solution we have

$$\mathrm{sgn}(x_{\ell+1}(I)) = \mathrm{sgn}(x_\ell(I) + \gamma_{\ell+1} d_{\ell+1}(I)).$$

By Lemma 4 we have

$$\mathrm{sgn}(d_{\ell+1}(I)) = \sigma_{\ell+1}(I).$$

We observe that

$$c_{\ell+1} = c_\ell - \gamma_{\ell+1} A^T A d_{\ell+1},$$

whence, on the active set,

$$|c_{\ell+1}(I)| = |c_\ell(I)|(1 - \gamma_{\ell+1}),$$

and so

$$\sigma_{\ell+1}(I) = \sigma_\ell(I).$$

In words, once a vector enters the active set, its residual correlation maintains the same sign. We conclude that

$$\mathrm{sgn}(x_{\ell+1}(I)) = \sigma_{\ell+1}(I). \tag{4.9}$$

Hence no variables leave the active set. To summarize, both the correct term selection property and the sign agreement property are maintained throughout the execution of the HOMOTOPY algorithm.

We may now invoke Lemma 2 to conclude that the k-step solution property holds. This completes the proof of Theorem 1. □

5 Fast Solution with the Uniform Spherical Ensemble

Suppose now that A is a random matrix drawn from the Uniform Spherical Ensemble. It is not hard to show that such matrices A are naturally incoherent; in fact, for $\epsilon > 0$, with high probability for large n

$$M(A) \leq \sqrt{\frac{4\log(n)}{d}} \cdot (1+\epsilon).$$

Thus, applying Theorem 1, we get as an immediate corollary that if k satisfies

$$k \leq \sqrt{d/\log(n)} \cdot (1/4 - \epsilon'), \tag{5.10}$$

then HOMOTOPY is highly likely to recover any sparse vector α_0 with at most k nonzeros. However, as it turns out, HOMOTOPY operates much more efficiently when applied to instances from the random suite $\mathcal{S}(\mathrm{USE}; d, n, k)$, than what is implied by incoherence. We now discuss evidence leading to Empirical Finding 1. We demonstrate, through a comprehensive suite of simulations, that the formula $d/(2\log(n))$ accurately describes the breakdown of the k-step solution property. Before doing so, we introduce two important tools that will be used repeatedly in the empirical analysis below.

5.1 Phase Diagrams and Phase Transitions

In statistical physics, a *phase transition* is the transformation of a thermodynamic system from one phase to another. The distinguishing characteristic of a phase transition is an abrupt change in one or more of its physical properties, as underlying parameters cross a region of parameter space. Borrowing terminology, we say that property $\mathcal{P}$ exhibits a phase transition at a sequence of problem suites $\mathcal{S}(\mathrm{E},\mathrm{V};d,n,k)$, if there is a threshold function $\mathrm{Thresh}(d,n)$ on the parameter space (k,d,n), such that problem instances with $k \leq \mathrm{Thresh}(d,n)$ have property $\mathcal{P}$ with high probability, and problem instances above this threshold do not have property $\mathcal{P}$ with high probability. As we show below, the the k-step solution property of HOMOTOPY, applied to problem instances drawn from $\mathcal{S}(\mathrm{USE};d,n,k)$, exhibits a phase transition at around $d/(2\log(n))$. Thus, there is a sequence (ϵ_n) with each term small and positive, so that for $k < (1-\epsilon_n)\cdot d/(2\log(n))$, HOMOTOPY delivers the sparsest solution in k steps with high probability; on the other hand, if $k > (1+\epsilon_n)\cdot d/(2\log(n))$, then with high probablity, HOMOTOPY, fails to terminate in k steps.

To visualize phase transitions, we utilize *phase diagrams*. In statistical physics, a phase diagram is a graphical depiction of a parameter space decorated by boundaries of regions where certain properties hold. Here we use this term in an analogous fashion, to mean a graphical depiction of a subset of the parameter space (d,n,k), illustrating the region where an algorithm has a given property $\mathcal{P}$, when applied to a problem suite $\mathcal{S}(\mathrm{E},\mathrm{V};d,n,k)$. It will typically take the form of a 2-D grid, with the threshold function associated with the corresponding phase transition dividing the grid into two regions: A region where property $\mathcal{P}$ occurs with high probablity, and a region where $\mathcal{P}$ occurs with low probablity.

While phase transitions are sometimes discovered by formal mathematical analysis, more commonly, the existence of phase transitions is unearthed by computer simulations. We now describe an objective framework for estimating phase transitions from simulation data. For each problem instance (A,y) drawn from a suite $\mathcal{S}(\mathrm{E},\mathrm{V};d,n,k)$, we associate a binary outcome variable $Y_k^{d,n}$, with $Y_k^{d,n} = 1$ when property $\mathcal{P}$ is satisfied on that realization, and $Y_k^{d,n} = 0$ otherwise. Within our framework, $Y_k^{d,n}$ is modeled as a Bernoulli random variable with probability $\pi_k^{d,n}$. Thus, $P(Y_k^{d,n}=1) = \pi_k^{d,n}$, and $P(Y_k^{d,n}=0) = 1-\pi_k^{d,n}$. Our goal is to estimate a value $\tilde{k}^{d,n}$ where the transition between these two states occurs. To do so, we employ a logistic regression model on the mean response $E\{Y_k^{d,n}\}$ with predictor variable k,

$$E\{Y_k^{d,n}\} = \frac{\exp(\beta_0+\beta_1 k)}{1+\exp(\beta_0+\beta_1 k)}. \tag{5.11}$$

Logistic response functions are often used to model threshold phenomena in statistical data analysis [34]. Let $\hat{\pi}_k^{d,n}$ denote the value of the fitted response function. We may then compute a value $\hat{k}_\eta^{d,n}$ indicating that, with probability exceeding $1-\eta$, for a problem instance drawn from $\mathcal{S}(\mathrm{E},\mathrm{V};d,n,k)$ with $k < \hat{k}_\eta^{d,n}$, property $\mathcal{P}$ holds. $\hat{k}$ is given by

$$\hat{\pi}_{\hat{k}} = 1-\eta.$$

Thus, computing $\hat{k}_\eta^{d,n}$ for a range of (d,n) values essentially maps out an *empirical phase transition* of property $\mathcal{P}$. The value η fixes a location on the transition band below which we define the region of success. In our work, we set $\eta = 0.25$.

5.2 The k-Step Solution Property

We now apply the method described in the previous section to estimate an empirical phase transition for the k-step solution property of the HOMOTOPY algorithm. Before doing so, we

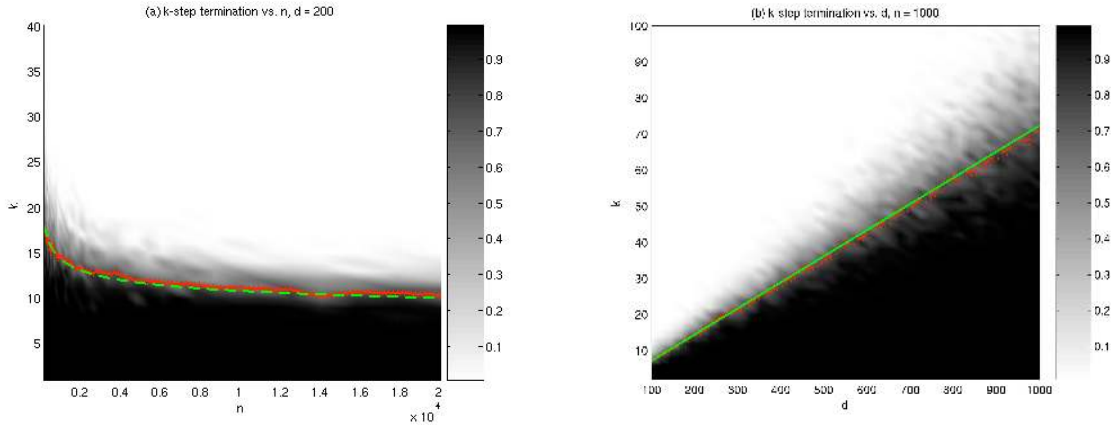

Figure 4: k-Step Solution Property of HOMOTOPY for the Problem Suite $\mathcal{S}(\text{USE}, \text{UNIFORM}; d, n, k)$. Shaded attribute is the proportion of successful terminations, out of 100 trials, in the case: (a) $k = 1 \dots 40$, $d = 200$, and $n = 200 \dots 20000$; (b) $k = 2 \dots 100$, $d = 100 \dots 1000$, and $n = 1000$. Overlaid are curves for $d/(2\log(n))$ (solid), $\hat{k}^{d,n}_{0.25}$ (dashed), and the confidence bound for $\hat{k}^{d,n}_{0.25}$ with 95% confidence level (dotted).

briefly describe the experimental setup. For (d, n, k) given, we generated a problem instance (A, y) drawn from the problem suite $\mathcal{S}(\text{USE}, \text{UNIFORM}; d, n, k)$. To this instance we applied the HOMOTOPY algorithm, and recorded the outcome of the response variable $Y^{d,n}_k$. We did so in two cases; first, fixing $d = 200$, varying n through 40 log-equispaced points in $[200, 20000]$ and k through 40 equispaced points in $[1, 40]$; second, keeping n fixed at 1000, varying d through 40 equispaced points in $[100, 1000]$ and k through 40 equispaced points in $[2, 100]$. For each (d, n, k) instance we ran 100 independent trials. To estimate the regression coefficients β_0, β_1 in (5.11), we used maximum likelihood estimation [34].

Figure 4 displays two phase diagrams, giving a succinct depiction of the simulation results. Panel (a) displays a grid of (k, n) values, $k \in [1, 40], n \in [200, 20000]$. Each point on this grid takes a value between 0 and 1, representing the ratio of successful outcomes to overall trials; these are precisely the observed $E\{Y^{d,n}_k\}$, on a grid of (k, n) values, with d fixed. Overlaid are curves for the estimated $\hat{k}^{d,n}_\eta$ for each n with $\eta = 0.25$, and the theoretical bound $d/(2\log(n))$. Panel (b) has a similar configuration on a (k, d) grid, with n fixed at 1000. Careful inspection of these phase diagrams reveals that the estimated $\hat{k}^{d,n}_\eta$ closely follows the curve $d/(2\log(n))$; Panel (a) shows the logarithmic behavior of the phase transition as n varies, and panel (b) shows its linear behavior as d varies.

Table 3 offers a summary of the results of this experiment, for the three nonzero ensembles used in our simulations: UNIFORM, GAUSS, and BERNOULLI. In each case, two measures for statistical goodness-of-fit are reported: the mean-squared error (MSE) and the R^2 statistic. The MSE measures the squared deviation of the estimated k values from the proposed formula $d/(2\log(n))$. The R^2 statistic measures how well the observed phase transition fits the theoretical model; a value close to 1 indicates a good fit. These statistical measures give another indication that the empirical behavior is in line with theoretical predictions. Moreover, they support the theoretical assertion that the k-step solution property of the HOMOTOPY algorithm is independent of the coefficient distribution.

The results in Figure 4 and Table 3 all demonstrate consistency between the empirical phase

Coefficient Ensemble	MSE	R^2
UNIFORM	0.09	0.98
GAUSS	0.095	0.98
BERNOULLI	0.11	0.98

Table 3: Deviation of estimated $\hat{k}_\eta^{d,n}$ from $d/(2\log(n))$ for $\mathcal{S}(\text{USE},\text{V};d,n,k)$, for different coefficient ensembles V.

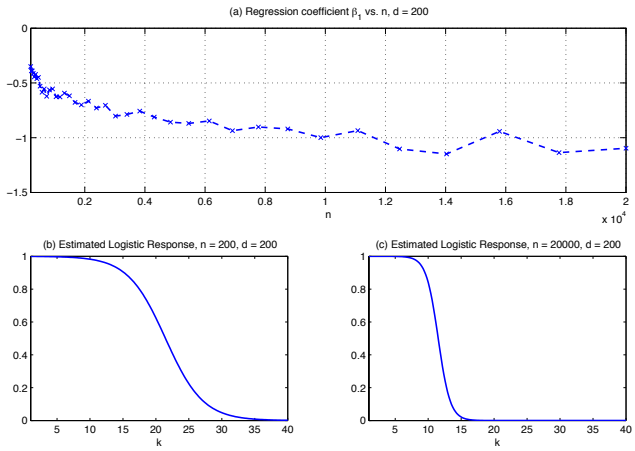

Figure 5: Sharpness of Phase Transition. The top panel shows the behavior of the regression coefficient β_1 of (5.11) for $d = 200$ and $n = 200 \ldots 20000$. The bottom panels show two instances of the fitted transition model, for (b) $d = 200, n = 200$ and (c) $d = 200, n = 20000$.

transition and the bound $d/(2\log(n))$, even for very modest problem sizes. We now turn to examine the sharpness of the phase transition as the problem dimensions increase. In other words, we ask whether for large d, n, the width of the transition band becomes narrower, i.e. ϵ_n in (1.2) becomes smaller as n increases. We note that the regression coefficient β_1 in (5.11) associated with the predictor variable k dictates how sharp the transition from $\pi_k^{d,n} = 1$ to $\pi_k^{d,n} = 0$ is; a large negative value for β_1 implies a 'step function' behavior. Hence, we expect β_1 to grow in magnitude as the problem size increases. This is indeed verified in panel (a) of Figure 5, which displays the behavior of β_1 for increasing n, and d fixed at 200. For illustration, panels (b) and (c) show the observed mean response $Y_k^{d,n}$ vs. k for $n = 200$, with $\beta_1 \approx -0.35$, and $n = 20000$, with $\beta_1 \approx -1.1$, respectively. Clearly, the phase transition in panel (c) is much sharper.

5.3 Correct Term Selection and Sign Agreement

In Section 4, we have identified two fundamental properties of the HOMOTOPY solution path, namely, correct term selection and sign agreement, that constitute necessary and sufficient conditions for the k-step solution property to hold. We now present the results of a simulation study identifying phase transitions in these cases.

In detail, we repeated the experiment described in the previous section, generating problem instances from the suite $\mathcal{S}(\text{USE},\text{UNIFORM};d,n,k)$ and applying the HOMOTOPY algorithm. As the outcome of the response variable $Y_k^{d,n}$, we recorded first the property { The HOMOTOPY

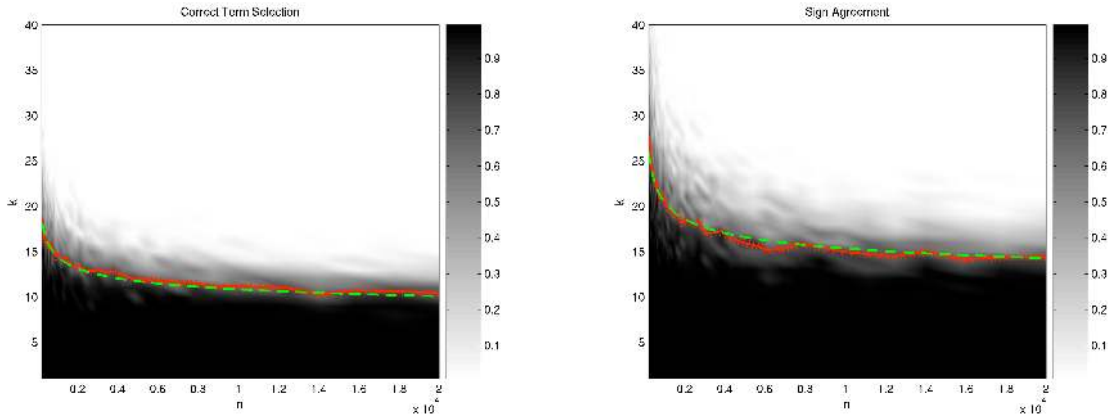

Figure 6: Phase diagrams for (a) Correct term selection property; (b) Sign agreement property, on a k-n grid, with $k = 1 \dots 40$, $d = 200$, and $n = 200 \dots 20000$. Overlaid are curves for $\hat{k}^{d,n}_{0.25}$ (dashed), and the confidence bound for $\hat{k}^{d,n}_{0.25}$ with 95% confidence level (dotted). Panel (a) also displays the curve $d/(\sqrt{2} \cdot \log(n))$ (solid), and panel (b) displays the curve $d/(2\log(n))$ (solid). The transition on the right-hand display occurs significantly above the transition in the left-hand display. Hence the correct term selection property is the critical one for overall success.

algorithm selected only terms from among $\{1, ..., k\}$ as entries to the active set }, and next, the property { At each iteration of the HOMOTOPY algorithm, the signs of the residual correlations on the active set match the signs of the corresponding solution terms }. Finally, we estimated phase transitions for each of these properties. Results are depicted in panels (a) and (b) of Figure 6. Panel (a) displays a phase diagram for the correct term selection property, and Panel (b) has a similar phase diagram for the sign agreement property.

The results are quite instructive. The phase transition for the correct term selection property agrees well with the formula $d/(2\log(n))$. Rather surprisingly, the phase transition for the sign agreement property seems to be at a significantly higher threshold, $d/(\sqrt{2}\cdot\log(n))$. Consequently, the 'weak link' for the HOMOTOPY k-step solution property, i.e. the attribute that dictates the sparsity bound for the k-step solution property to hold, is the addition of incorrect terms into the active set. We interpret this finding as saying that the HOMOTOPY algorithm would typically err first 'on the safe side', adding terms off the support of α_0 into the active set (*a.k.a* false discoveries), before removing correct terms from the active set (*a.k.a* missed detections).

5.4 Random Signs Ensemble

Matrices in the Random Signs Ensemble (RSE), are constrained to have elements of constant amplitude, with signs independently drawn from an equi-probable Bernoulli distribution. This ensemble is known to be effective in various geometric problems associated with underdetemined systems, through work of Kashin [36], followed by Garnaev and Gluskin [28]. Previous work [55, 56, 54] gave theoretical and empirical evidence to the fact that many results developed in the USE case hold when the matrices considered are drawn from the RSE. Indeed, as we now demonstrate, the RSE shares the k-step solution property with USE.

To show this, we conducted a series of simulations, paralleling the study described in Section 5.2. In our study, we replaced the problem suite $\mathcal{S}(\text{USE}; d, n, k)$ by $\mathcal{S}(\text{RSE}; d, n, k)$. Just as above, we estimated $\hat{k}^{d,n}$ in two cases: First, fixing $d = 200$, varying n through 40 log-equispaced

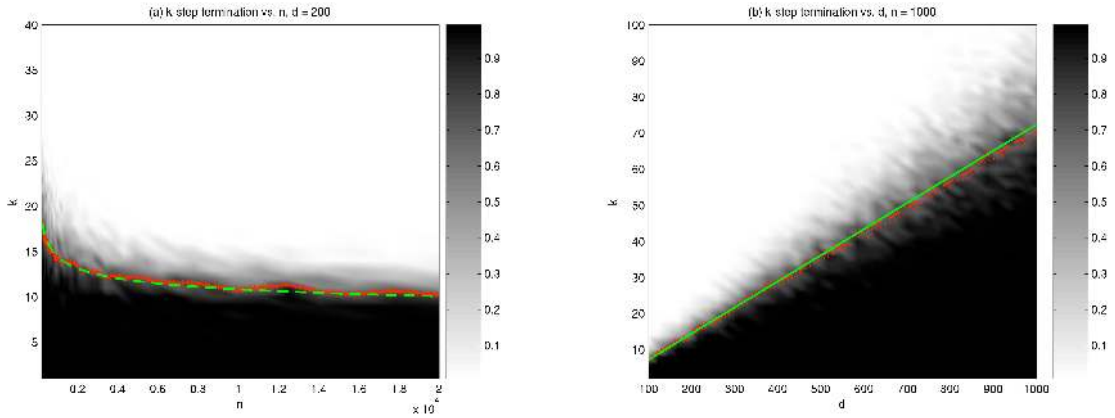

Figure 7: k-Step Termination of HOMOTOPY for the problem suite $\mathcal{S}(\text{RSE},\text{UNIFORM}; d, n, k)$. Shaded attribute is the proportion of successful terminations, out of 100 trials, in the case: (a) $k = 1 \ldots 40$, $d = 200$, and $n = 200 \ldots 20000$; (b) $k = 2 \ldots 100$, $d = 100 \ldots 1000$, and $n = 1000$. Overlaid are curves for $d/(2\log(n))$ (solid), $\hat{k}^{d,n}_{0.25}$ (dashed), and the confidence bound for $\hat{k}^{d,n}_{0.25}$ with 95% confidence level (dotted).

points in $[200, 20000]$ and k through 40 equispaced points in $[1, 40]$; second, keeping n fixed at 1000, varying d through 40 equispaced points in $[100, 1000]$ and k through 40 equispaced points in $[2, 100]$. The resulting phase diagrams appear in panels (a),(b) of Figure 7. Careful inspection of the results indicates that the empirical phase transition for RSE agrees well with the formula $d/(2\log(n))$ that describes the behavior in the USE case. This is further verified by examining Table 4, which has MSE and R^2 measures for the deviation of the empirical phase transition from the threshold $d/(2\log(n))$, for different coefficient ensembles. In summary, these empirical results give strong evidence to support the conclusion that the HOMOTOPY algorithm has the k-step solution property when applied to the the problem suite $\mathcal{S}(\text{RSE}; d, n, k)$ as well. An important implication of this conclusion is that in practice, we may use matrices drawn from the RSE to replace matrices from the USE without loss of performance. This may result in reduced complexity and increased efficiency, as matrices composed of random signs can be generated and applied much more rapidly than general matrices composed of real numbers.

Coefficient Ensemble	MSE	R^2
UNIFORM	0.12	0.98
GAUSS	0.11	0.98
BERNOULLI	0.08	0.99

Table 4: Deviation of estimated $\hat{k}^{d,n}_{\eta}$ from $d/(2\log(n))$ for $\mathcal{S}(\text{RSE},\text{V}; d, n, k)$, for different coefficient ensembles V.

6 Fast Solution with Partial Orthogonal Ensembles

We now turn our attention to a class of matrices composed of partial orthogonal matrices, i.e. matrices constructed by sampling a subset of the rows of an orthogonal matrix. In particular, we will be interested in the following three ensembles:

- *Partial Fourier Ensemble (PFE).* Matrices in this ensemble are obtained by taking an n by n Fourier matrix and sampling d rows at random. The PFE is of utmost importance in medical imaging and spectroscopy, as it can be used to model reconstruction of image data from partial information in the frequency domain [7, 6, 37, 38].

- *Partial Hadamard Ensemble (PHE).* Matrices in this ensemble are obtained by taking an n by n Hadamard matrix and sampling d of its rows at random. This ensemble is known to be important for various extremal problems in experimental design [33].

- *Uniform Random Projections (URPE).* Matrices in this ensemble are obtained by taking an n by n random orthogonal matrix and sampling d of its rows at random. It serves as a general model for the PFE and PHE.

As we now demonstrate, the k-step solution property of HOMOTOPY holds at a significantly greater range of signal sparsities than observed in Section5.

We first consider the PFE and PHE. Partial Fourier matrices and partial Hadamard matrices have drawn considerable attention in the sparse approximation community [7, 6, 56, 55, 29]. We mention two incentives:

- **Modeling of Physical Phenomena.** Fourier operators play a key role in numerous engineering applications modeling physical phenomena. For instance, acquisition of 2- and 3-D Magnetic Resonance images is modeled as applying a Fourier operator to spatial data. Thus, application of a partial Fourier operator would correspond to acquisition of partial frequency data. Likewise, the Hadamard operator, albeit not as common, is often used in Hadamard Transform Spectroscopy.

- **Fast Implementations.** Both the Fourier transform and the Hadamard transform have special structure allowing for rapid computation. They can be computed by algorithms which run much faster than the naïve implementation from definition. In detail, the Fast Fourier Transform (FFT) can be applied to an n-vector using $O(n\log(n))$ operations, rather than the n^2 operations needed by direct computation [3]. Similarly, a Fast Hadamard Transform (FHT) exists, which applies a Hadamard operator using $O(n\log(n))$ operations. With these fast algorithms, rapid computation of partial Fourier or partial Hadamard products is straightforward. Specifically, note that we may write a d by n matrix A from the PFE as $A = R \cdot F$, with F an n by n Fourier matrix, and R a d by n random sampling matrix. Translating to linear operations, application of A amounts to applying F, using the FFT, followed by sampling d entries of the resulting vector. Similarly for partial Hadamard matrices.

The experimental setup leading to Empirical Finding 2 utilizes a more extensive phase diagram. Expressly, we fix n and generate a grid of points indexed by variables $\delta = d/n$ and $\rho = k/d$, with δ in $(0, 1]$, ρ in $(0, 1]$. Thus, at a fixed δ, $d = \lfloor \delta n \rfloor$, and $k = \lfloor \rho d \rfloor$ ranges between 1 and d. In a sense, this ρ-δ grid gives a complete picture of all sparse underdetermined settings of interest. We considered two discretizations. First, we fixed $n = 1024$, varied δ in the range $[0.1, 1]$, and ρ in $[0.05, 1]$. Second, we fixed $\delta = 0.5$, varied n in $[256, 1024]$, and ρ in $[0.05, 1]$. Data collection and estimation of an empirical phase transition were done in the same manner described in Section 5.2. Figures 8 and 9 summarize the simulation results, each depicting three phase diagrams, for three different nonzero distributions.

The results are indeed surprising; for problems from the suite $\mathcal{S}(\text{PFE}; d, n, k)$, HOMOTOPY maintains the k-step solution property at a much broader range of sparsities k than observed

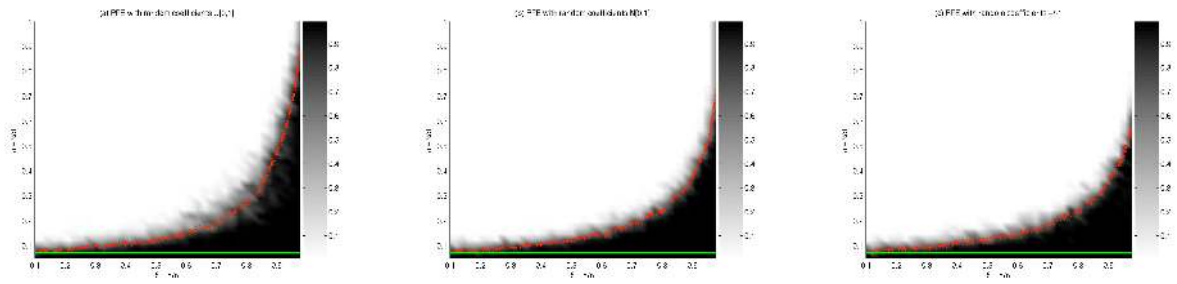

Figure 8: k-Step Solution Property of HOMOTOPY for the Suite $\mathcal{S}(\text{PFE},\text{V};d,n,k)$. Each phase diagram presents success rates on a ρ-δ grid, with $\rho = k/d$ in $[0.05, 1]$, $\delta = d/n$ in $[0.1, 1]$, and $n = 1024$, for: (a) V = UNIFORM; (b) V = GAUSS; (c) V = BERNOULLI. Overlaid are curves for $\hat{\rho}_{0.25}(\delta)$ (dash-dotted), with the confidence bounds for 95% confidence level (dotted), and $\rho = 1/(2\log(n))$ (solid). The range of sparsities in which the k-step solution property holds is much more extensive than implied by the bound $k < d/(2\log(n))$.

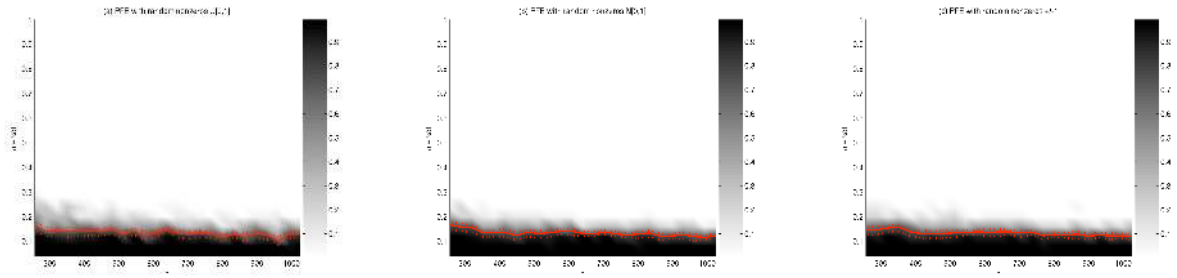

Figure 9: k-Step Solution Property of HOMOTOPY for the Suite $\mathcal{S}(\text{PFE},\text{V};d,n,k)$. Each phase diagram presents success rates on a ρ-n grid, with $\rho = k/d$ in $[0.05, 1]$, n in $[128, 8192]$, and $d = 125$, for: (a) V = UNIFORM; (b) V = GAUSS; (c) V = BERNOULLI. Overlaid are curves for $\hat{\rho}_{0.25}(\delta)$ (solid), with the confidence bounds for 95% confidence level (dotted).

for problems from $\mathcal{S}(\text{USE};d,n,k)$, particularly at high values of the indeterminacy ratio δ. For instance, at $\delta = 3/4$, the empirical results indicate that when the nonzeros are uniformly distributed, the empirical phase transition is at $\hat{k} = \hat{\rho}d \approx 174$, whereas the formula $d/(2\log(n)) \approx 55$, implying that the range of sparsities for which HOMOTOPY terminates in k steps has grown threefold.

We conducted parallel simulation studies for the PHE and the URPE, and we summarize the empirical findings in Figure 10. It displays the empirical phase transitions $\hat{\rho}(\delta)$ for the three partial orthogonal ensembles with different nonzero distributions. Careful inspection of the curves reveals that the k-step property of HOMOTOPY holds at roughly the same region of (δ, ρ) space for all combinations of matrix ensembles and coefficient distributions considered, with mild variations in the location and shape of the phase transition for different nonzero distributions. In particular, the data indicate that problems with underlying nonzero distribution BERNOULLI exhibit the lowest k-step breakdown. Thus, in some sense, the 'hardest' problems are those with uniform-amplitude nonzeros in the solution.

The empirical phase transition $\hat{\rho}(\delta)$ does not yield itself to a simple representation in functional form. It does, however, have several important characteristics, which we now highlight.

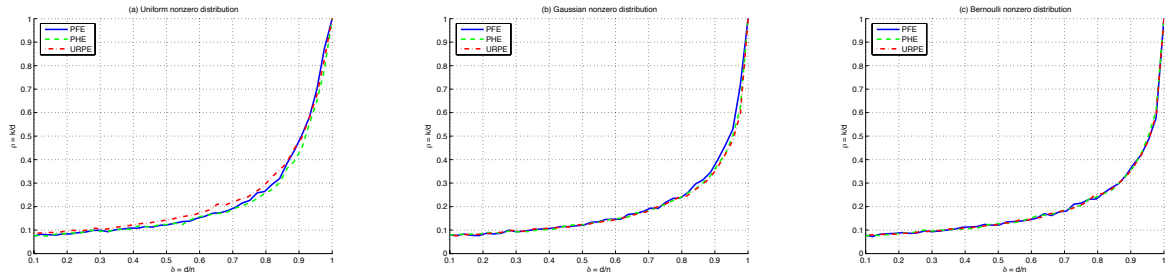

Figure 10: Estimated Phase Transition Curves. Each panel displays empirical phase transition curves for the PFE (solid), PHE (dashed), URPE (dash-dotted), with (a) Uniform nonzero distribution; (b) Gaussian nonzero distribution; (c) Bernoulli nonzero distribution.

- First, notice that at the right end of the phase diagram, we have $\hat{\rho}(\delta = 1) = 1$. In other words, when the matrix A is square, HOMOTOPY always recovers the sparsest solution in k steps. This is hardly surprising, since, at $\delta = 1$, the matrix A is orthogonal, and the data y is orthogonal to columns of A off the support of α_0, ensuring correct term selection. In addition, the update step $d_\ell(I)$ in (2.7) is equal to $\mathrm{sgn}(c_\ell(I))$, implying that the sign agreement property holds. Thus, k-step termination is guaranteed.

- Second, at the left end of the phase diagram, as $\delta \to 0$, we may rely on recent theoretical developments to gauge the behavior of the phase transition curve. In their work on randomly projected polytopes, Donoho and Tanner showed that an upper bound for the value $\rho(\delta)$ below which ℓ_1 minimization recovers the sparsest solution with high probability, asymptotically behaves like $1/(2\log(1/\delta))$ as $\delta \to 0$ [22]. In particular, this same upper bound holds in our case, as failure to recover the sparsest solution necessarily implies failure of the k-step property.

- Lastly, and most importantly, we note that the k-step phase transition for the partial orthogonal ensembles is stable with increasing n, in the following sense. For $\delta \in (0, 1]$ fixed, the range of ρ in which the k-step property holds does not change with n. Evidence to the stability of the phase transition is given in Figure 9, which demonstrates that for δ fixed at 0.5, the phase transition occurs at an approximately constant value of ρ. The implication of this property is that the empirical phase transitions computed here may be used to estimate the k-step behavior of HOMOTOPY for problem instances of any size.

7 Bridging ℓ_1 Minimization and OMP

In the introduction, we alluded to the fact that there is undeniable parallelism in results about the ability of ℓ_1 minimization and OMP to recover sparse solutions to systems of underdetermined linear equations. Indeed, earlier reports [9, 19, 15, 52, 53, 54] documented instances where, under similar conditions, both ℓ_1 minimization and OMP recover the sparsest solution. Yet, these works did not offer any insights as to why the two seemingly disparate techniques should offer similar performance in such cases. This section develops a linkage between ℓ_1 minimization and OMP.

As a first step, we clear up some confusion in terminology. Previous work studying the performance of OMP set conditions on the sparsity k under which the algorithm 'correctly recovers the sparsest solution' [19, 52, 54]. Yet, in effect, what was proven is that OMP 'selects

a correct term at each iteration, and terminates with the correct solution after k iterations'; we now recognize this as the k-step solution property. We note that the k-step solution property is not, in general, a necessary condition for OMP to correctly recover the sparsest solution; it is clearly sufficient, however. Its usefulness lies in the simplicity of k-step solutions; when the k-step property holds, the solution path is the simplest possible, consisting of k correct steps. Thus, it yields itself naturally to focused analysis.

We now argue that, through a series of simplifications, we can move from ℓ_1 minimization to OMP, at each step maintaining the k-step solution property. This was already alluded to in Figure 1, which shows the sequence of steps leading from ℓ_1 minimization to OMP; the two links in the sequence are HOMOTOPY and LARS. In the following subsections, we elaborate on each of the bridging elements.

7.1 ℓ_1 Minimization $\longrightarrow$ Homotopy

For the phrase 'algorithm $\mathcal{A}$ has the k-step solution property' to make sense, the algorithm must have a stepwise structure, building a solution approximation term by term. Thus, we cannot, in general, speak of ℓ_1 minimization as having the k-step solution property, as there are many algorithms for such minimization, including those with no meaningful notion of stepwise approximate solution construction. This is where the HOMOTOPY algorithm fits in, bridging the high-level notion of ℓ_1 minimization with the lower-level stepwise structure of OMP. Earlier work [24, 40] has shown that HOMOTOPY is a correct algorithm for solving the ℓ_1 minimization problem (P_1). In addition it builds an approximate solution in a stepwise fashion, thus making the k-step solution property applicable.

7.2 Homotopy $\longrightarrow$ LARS

As noted earlier, the LARS procedure is a simplification of the HOMOTOPY algorithm, achieved by removing the condition for sign agreement of the current solution and the residual correlations. This brings us one step closer to OMP; while HOMOTOPY allows for removal of terms from the active set, both LARS and OMP insert terms, one by one, into the active set, never removing any active elements.

Interestingly, when the k-step property holds, this simplification changes nothing in the resulting solution. To see that, recall that HOMOTOPY has the k-step solution property if and only if it has the correct term selection property and the sign agreement property. Correct term selection is all that is needed by LARS to reach the solution in k steps; after k correct terms are inserted into the active set, the algorithm would terminate with zero residual. The sign agreement property implies that HOMOTOPY would successfully terminate in k steps as well. In summary, when the k-step solution property holds, the solution paths of HOMOTOPY and LARS coincide. In particular, the assertion of Theorem 1 holds for LARS as well.

Corollary 2 *Let (A, y) be a problem instance drawn from $\mathcal{S}(\text{INC}_\mu; d, n, k)$, with $k \leq (\mu^{-1}+1)/2$. Then,* LARS *runs k steps and stops, delivering the solution α_0.*

Similar conclusions can be made about the assertions of Empirical Findings 1 and 2.

7.3 LARS $\longrightarrow$ OMP

We already hinted at the fact that HOMOTOPY and OMP share an inherent affinity. We now demonstrate that the two methods are based on the same fundamental procedure; solving a

sequence of least-squares problems on an increasingly large subspace, defined by the active columns of A.

More precisely, assume we are at the ℓ-th iteration, with a solution estimate $x_{\ell-1}$ and an active set I consisting of $\ell - 1$ terms. Recall the procedure underlying OMP: It appends a new term to the active set by selecting the vector a_j that maximizes the absolute residual correlation; $j = \arg\max_{j \notin I} |a_j^T(y - Ax_{\ell-1})|$, and $I := I \bigcup \{j\}$. It then projects y onto $\mathcal{R}(A_I)$ by solving

$$A_I^T A_I x_\ell(I) = A_I^T y, \tag{7.12}$$

thus guaranteeing that the residual is orthogonal to the subspace $\mathcal{R}(A_I)$. Once y is spanned by the active columns, the algorithm terminates with zero residual.

While LARS was derived from a seemingly disparate perspective, it in fact follows the same basic procedure, replacing (7.12) with a penalized least-squares problem. Recalling the discussion in Section 2, we note that LARS also selects a new term according to the maximum correlation criterion, and then solves

$$A_I^T A_I x_\ell(I) = A_I^T y - \lambda_\ell \cdot \mathbf{s}, \tag{7.13}$$

where $\mathbf{s}$ is a vector of length ℓ, recording the signs of residual correlations of each term at the point it entered the active set, and λ_ℓ is the correlation magnitude at the breakpoint on the LARS path. Indeed, (7.12) and (7.13) are identical if not for the penalization term on the right hand side of (7.13).

As it turns out, this modification does not have an effect on the k-step solution property of the procedure. We offer two examples. First, for the incoherent problem suite $\mathcal{S}(\text{INC}_\mu; d, n, k)$; the statement of theorem 1 hold for OMP as well (see, e.g., Corollary 3.6 in [52]). Second, for the random problem suite $\mathcal{S}(\text{USE}; d, n, k)$; Tropp *et al.* proved results which can now be reinterpreted as saying that, when $k \leq d/(c \log(n))$, then with high probability for large n, OMP has the k-step solution property, paralleling the statement of Empirical Finding 1. We believe, in fact, that the bound $k \leq d/(2 \log(n))$ holds for OMP as well.

To summarize, we exhibited a series of transformations, starting with ℓ_1 minimization, and ending with greedy pursuit. Each transformation is characterized clearly, and maintains the k-step property of the solution path. We believe this sequence clarifies initially surprising similarities between results about ℓ_1 minimization and OMP.

8 Homotopy and PFP

Polytope Faces Pursuit was introduced as a greedy algorithm to solve (P_1) in the dual space. As pointed out in [43, 42], although the algorithm is derived from a seemingly different viewpoint, it exhibits interesting similarities to HOMOTOPY. To study these in more detail, we first briefly describe the algorithm.

Define the standard-form linear program equivalent to (P_1)

$$(\bar{P}_1) \qquad \min \mathbf{1}^T \bar{x} \;\text{ subject to }\; y = \bar{A}\bar{x},\; \bar{x} \geq 0,$$

where $\bar{A} = [A \;\; -A]$. The equivalence of (P_1) and $(\bar{P}_1)$ is well established; the solutions $\bar{x}$ of $(\bar{P}_1)$ and x of (P_1) have the correspondence $\bar{x}(j) = \max\{x(j), 0\}$ for $1 \leq j \leq n$, and $\bar{x}(j) = \max\{-x(j-n), 0\}$ for $n+1 \leq j \leq 2n$. The Lagrangian dual of the linear program $(\bar{P}_1)$ is given by

$$(\bar{D}_1) \qquad \max y^T v \;\text{ subject to }\; \bar{A}^T v \leq \mathbf{1}.$$

By the strong duality theorem, an optimal solution $\bar{x}$ of the primal problem $(\bar{P}_1)$ is associated with an optimal solution v to the dual problem $(\bar{D}_1)$, such that $\mathbf{1}^T\bar{x} = y^T v$. PFP operates in the dual space, tracing a solution to the dual problem $(\bar{D}_1)$, which then yields a solution to the primal problem $(\bar{P}_1)$ (or, equivalently, (P_1)). To do so, the algorithm maintains an active set, I, of nonzero coordinates, and carries out a familiar procedure at each iteration: Selection of a new term to enter the active set, removal of violating terms from the active set (if any), and computation of a new solution estimate. Specifically, At the ℓ-th stage, PFP first selects a new term to enter the active set, via the criterion

$$i_\ell = \arg\max_{i \notin I} \left\{ \frac{\bar{a}_i^T r_\ell}{1 - \bar{a}_i^T v_\ell} \;\middle|\; \bar{a}_i^T r_\ell > 0 \right\}, \tag{8.14}$$

with $r_\ell = y - \bar{A}\bar{x}_\ell$ the current residual. It then updates the active set, $I \leftarrow I \cup \{\bar{a}_i\}$, and computes a primal solution estimate, the projection of y onto the subspace spanned by the active vectors,

$$\bar{x}_{\ell+1} = (\bar{A}_I^T \bar{A}_I)^{-1} \bar{A}_I^T y, \tag{8.15}$$

and a dual solution estimate

$$v_{\ell+1} = \bar{A}_I (\bar{A}_I^T \bar{A}_I)^{-1} \mathbf{1}. \tag{8.16}$$

Finally, the algorithm verifies that all the active coefficients $\bar{x}_{\ell+1}(I)$ are non-negative; coordinates with negative coefficients are removed from the active set.

The affinities between HOMOTOPY and PFP are not merely cosmetic. In fact, the two algorithms, derived from seemingly distinct viewpoints, carry out nearly identical procedures. To highlight these similarities, assume that both algorithms are applied to the standard form linear program $(\bar{P}_1)$. We begin by examining the criteria for selection of a new term to enter the active set. In particular, assume we are at the ℓ-th stage of the HOMOTOPY algorithm. Recalling the discussion in Section 2 above, the HOMOTOPY term selection criterion (when applied to $(\bar{P}_1)$) is

$$i^+ = \arg\min_{i \in I^c}{}^+ \left\{ \frac{\lambda - c_\ell(i)}{1 - \bar{a}_i^T v_\ell} \right\}, \tag{8.17}$$

where $v_\ell = \bar{A}_I d_\ell(I) = \bar{A}_I (\bar{A}_I^T \bar{A}_I)^{-1} \mathbf{1}$ as in (8.16) above, and $\min^+$ indicates that the minimum is taken over positive arguments. Comparing (8.17) with (8.14), we note that the main difference lies in the residual; since PFP projects the data y onto the space spanned by the active vectors, denoted $\mathcal{R}_I$, its residual is necessarily orthogonal to $\mathcal{R}_I$. HOMOTOPY, in contrast, does not enforce orthogonality, and its residual can be expressed as the sum of a component in $\mathcal{R}_I$ and a component in the orthogonal complement $\mathcal{R}_I^\perp$. Formally,

$$r_\ell = P_I r_\ell + r_\ell^\perp,$$

where $P_I = \bar{A}_I (\bar{A}_I^T \bar{A}_I)^{-1} \bar{A}_I^T$ is the orthogonal projector onto $\mathcal{R}_I$, and $r_\ell^\perp = (I_d - P_I) r_\ell$ is the component of r_ℓ in $\mathcal{R}_I^\perp$. The component of r_ℓ in $\mathcal{R}_I$ obeys

$$P_I r_\ell = \bar{A}_I (\bar{A}_I^T \bar{A}_I)^{-1} c_\ell = \lambda v_\ell. \tag{8.18}$$

Plugging this into (8.17), we get

$$\begin{aligned} i^+ &= \arg\min_{i \in I^c}{}^+ \left\{ \frac{\lambda - \bar{a}_i^T (r_\ell^\perp + \lambda v_\ell)}{1 - \bar{a}_i^T v_\ell} \right\} \\ &= \arg\min_{i \in I^c}{}^+ \left\{ \lambda - \frac{\bar{a}_i^T r_\ell^\perp}{1 - \bar{a}_i^T v_\ell} \right\} \\ &= \arg\max_{i \in I^c} \left\{ \frac{\bar{a}_i^T r_\ell^\perp}{1 - \bar{a}_i^T v_\ell} \;\middle|\; \bar{a}_i^T r_\ell^\perp > 0 \right\}, \end{aligned}$$

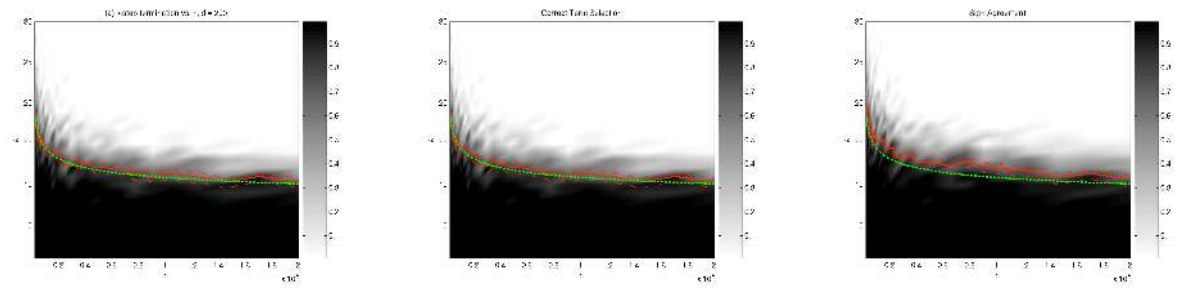

Figure 11: Phase Diagrams of PFP for the Problem Suite $\mathcal{S}(\text{USE},\text{UNIFORM};d,n,k)$. Shaded attribute is the proportion of successful executions among total trials, for: (a) k-step solution property; (b) correct term selection property; (c) sign agreement property. Overlaid on each plot are curves for $d/(2\log(n))$ (solid), $\hat{k}^{d,n}_{0.25}$ (dashed), and the confidence bound for $\hat{k}^{d,n}_{0.25}$ with 95% confidence level (dotted).

where the last equality holds because $\bar{A}^T v_\ell < 1$ throughout the HOMOTOPY solution path. Thus, conditional on $r^{PFP}_\ell = r^\perp_\ell$, the HOMOTOPY selection criterion is equivalent to PFP's selection criterion. Now, since $r^\perp_\ell = y - P_I y$, as long as no elements are removed from the active set, $r^{PFP}_\ell = r^\perp_\ell$ by definition. In summary, the discussion above establishes the following.

Corollary 3 *As long as no elements are removed from the active set, the solution path of* PFP *is identical to the* HOMOTOPY *solution path. In particular, with the sign condition removed,* PFP *is equivalent to* LARS.

This finding is interesting in light of the earlier discussion; it implies that when the k-step solution property holds for HOMOTOPY, it also holds for PFP. In particular, the statement of Theorem 1 holds for PFP as well. Formally,

Corollary 4 *Let (A, y) be a problem instance drawn from $\mathcal{S}(\text{INC}_\mu; d, n, k)$, with $k \leq (\mu^{-1}+1)/2$. Then,* PFP *runs k steps and stops, delivering the solution α_0.*

Similarly, Empirical Findings 1 and 2 hold for PFP as well. Figure 11 displays phase diagrams for the k-step solution property (Panel (a)), correct term selection property (Panel (b)), and sign agreement property (Panel (c)), when PFP is applied to instances of the problem suite $\mathcal{S}(\text{USE},\text{UNIFORM};d,n,k)$. The three properties exhibit clear phase transition at around $d/(2\log(n))$. Analogously, the statements in Empirical Findings 2 can be verified to hold for PFP as well; we do not bring such evidence here for brevity of space.

9 Recovery of Sparse Solutions

Previous sections examined various properties of HOMOTOPY, all centered around the k-step solution property. The motivation is clear; The HOMOTOPY algorithm is at peak efficiency when it satisfies the k-step solution property. Yet, we noted that failure to terminate in k steps need not imply failure to recover the sparsest solution. We now turn to study this latter property in more detail.

Definition 5 *An algorithm $\mathcal{A}$ has the* k-sparse Solution Property *at a given problem instance (A, y) drawn from $\mathcal{S}(\text{E},\text{V};d,n,k)$ if, when applied to the data (A, y), the algorithm terminates with the solution α_0.*

Earlier papers used a different term for the same concept [18, 15, 55]; they would use the term ℓ_0 *equivalence* to indicate recovery of the sparsest solution, i.e. equivalence to the solution of the combinatorial optimization problem

$$(P_0) \qquad \min_x \|x\|_0 \text{ subject to } y = Ax.$$

For the HOMOTOPY algorithm, the k-sparse solution property has been well-studied. Indeed, since the HOMOTOPY algorithm is guaranteed to solve (P_1), it will have the k-sparse solution property whenever the solutions to (P_0) and (P_1) coincide. In a series of papers [15, 13, 17], Donoho has shown that for problem instances from the suite $\mathcal{S}(\text{USE}; d, n, k)$, the equivalence between solutions of (P_0) and (P_1) exhibits a clear phase transition, denoted ρ_W, that can be computed numerically. This phase transition provides a theoretical prediction for the conditions under which HOMOTOPY satisfies the k-sparse solution property. Yet, at present, no such predictions exist for LARS, OMP, or PFP.

We now present the results of an extensive simulation study, examining the validity of the k-sparse solution property for HOMOTOPY, LARS, OMP, and PFP, comparing the performance of these four closely related algorithms. The study we conduct closely parallels the experiments described above. In particular, we measure the k-sparse solution property for the three algorithms, applied to problems instances from the suites $\mathcal{S}(\text{USE},\text{UNIFORM}; d, n, k)$, $\mathcal{S}(\text{USE},\text{GAUSS}; d, n, k)$ and $\mathcal{S}(\text{USE},\text{BERNOULLI}; d, n, k)$. We generated 100 independent trials at each point on the parameter grid, indexed by $\delta = d/n$ and $\rho = k/d$, with δ ranging through 40 equispaced points in $[0.1, 1]$, ρ ranging through 40 equispaced points in $[0.05, 1]$, and n fixed at 1000. The resulting phase diagrams appear in Figure 12. Parallel results summarizing a similar experiment done for the URPE are presented in Figure 13. Preliminary inspection of these phase diagrams reveals that the suggestive similarities between the three algorithms indeed translate into comparable performance in recovering the sparsest solution. More careful examination commands several important observations.

- **Similarity of LARS and HOMOTOPY.** Throughout all the displayed phase diagrams, the empirical phase transition of LARS closely follows the transition of HOMOTOPY. In other words, almost anytime ℓ_1 minimization successfully recovers the sparsest solution, LARS succeeds in finding it as well. This is a rather surprising find, as, at least in the USE case, for $k > d/(2\log(n))$, LARS is bound to err and add elements not in the support of α_0 into its active set. The evidence we present here suggests that inside the ℓ_1-ℓ_0 equivalence phase, LARS still correctly recovers the support of α_0 and obtains the sparsest solution.

- **Universality of LARS and HOMOTOPY.** Examining Figure 12, we note that for the suite $\mathcal{S}(\text{USE}; d, n, k)$, the performance of LARS and HOMOTOPY is independent of the underlying distribution of the nonzero coefficients. Universality does not hold for OMP, whose performance is affected by the distribution of the nonzeros. In particular, OMP's worst performance is recorded when the underlying nonzeros have constant amplitude and a random sign pattern; this has been noted previously in the analysis of OMP [15, 52].

- **Increased Performance with $\mathcal{S}(\text{URPE}; d, n, k)$.** Comparison of Figure 13 with Figure 12 reveals that, in general, all four algorithms perform better when applied to problem instances from the suite $\mathcal{S}(\text{URPE}; d, n, k)$. Universality, however, no longer holds for this problem suite. In particular, HOMOTOPY and LARS show a marked increase in performance when applied to the suite $\mathcal{S}(\text{URPE},\text{UNIFORM}; d, n, k)$. In our experiments, we observed similar phenomena when applying the three algorithms to problems drawn from the suites

$\mathcal{S}(\mathrm{PFE}; d, n, k)$ and $\mathcal{S}(\mathrm{PHE}; d, n, k)$ as well. This is indeed an important finding, as these suites are used extensively in applications.

10 Stylized Applications

Earlier, we gave theoretical and empirical evidence showing that in certain problem scenarios, when sufficient sparsity is present, the HOMOTOPY path and the LARS path agree. We now present examples inspired by concrete applications in NMR spectroscopy and MR imaging, that demonstrate the performance of the two algorithms in solving sparse approximation problems. In accord with earlier sections, we shall compare and contrast the results with those obtained by OMP.

10.1 Spectroscopy

Nuclear Magnetic Resonance (NMR) Spectroscopy is a widely used method of structure determination in modern chemistry; see [35]. In its essence, NMR spectroscopy is based on the behavior of atomic nuclei in a magnetic field. Thus, an NMR spectrometer records the resonance frequencies of nuclei when exposed to a strong magnetic field and irradiated with an RF pulse. The resulting signal is a superposition of all excited frequencies. For an caricature of 1-D NMR spectra, consider the signal *Bumps*, depicted in Figure 14(a). It is a synthetic signal, made up of a few peaks at varying amplitudes. Figure 14(b) shows its wavelet expansion on a scale-by-scale basis. Clearly, *Bumps* has relatively few significant wavelet coefficients, and it admits a sparse representation in a wavelet basis.

As a first example, we consider reconstruction of NMR spectra from partial frequency information [46, 47]. In detail, let x be an NMR signal of length n, which admits sparse synthesis in a wavelet basis, i.e. $\alpha = Wx$ is sparse, with W a wavelet analysis operator. Let Φ be a $d \times n$ matrix drawn from the partial Fourier ensemble, i.e. constructed by selecting d rows of an $n \times n$ Fourier matrix at random. At the sampling stage, we compute $y = \Phi x$; In words, y is computed by sampling d frequencies of x at random. At the reconstruction stage, we solve:

$$(CS_1) \qquad \min \|\alpha\|_1 \text{ subject to } y = \Phi W^T \alpha,$$

where we used the fact that W is an orthogonal operator, whence $x = W^T\alpha$. Readers familiar with the notion of Compressed Sensing (CS) [14, 56] will recognize (CS_1) as a particular case of a Compressed Sensing problem. Indeed, work by Donoho [14] has shown that when the signal x is compressible (i.e. has a sparse representation in a certain basis), the solution of (CS_1) is faithful to the original signal x.

For the purposes of this simulation, the signal *Bumps* was digitized to $n = 1024$ samples. We measured $d = 512$ random frequency measurements, and applied the HOMOTOPY algorithm to solve (CS_1), as well as LARS, OMP, and PFP. Figure 15 summarizes the results at several stages of execution, and Table 5 has corresponding error and timing measures. The results indicate that all four algorithms faithfully recovered the signal, at roughly the same error. Of the four algorithms, OMP was fastest, though only marginally, compared to HOMOTOPY and LARS. We also note that, owing to the rapid coefficient decay of the wavelet expansion of *Bumps*, already after 200 iterations, we obtain accurate reconstructions, at a fraction of the time it takes the execution to complete. This suggests that, in practice, if we have knowledge of the sparsity structure of the solution, we may enforce early termination without compromising the quality of reconstruction. As the example in the next section illustrates, the savings are particularly significant in large-scale applications.

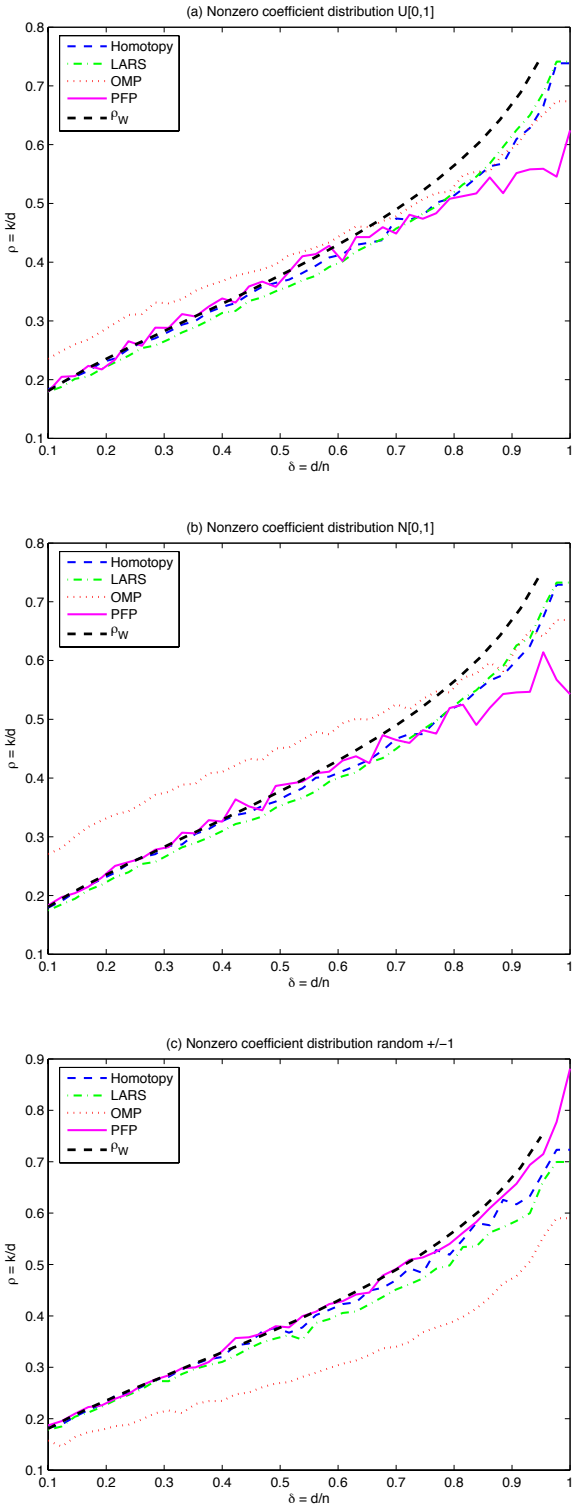

Figure 12: The ℓ_0 Equivalence Property for the Suite $\mathcal{S}(\text{USE},\text{V}; d, n, k)$. Each panel has plots of the empirical phase transition curves of Homotopy (dashed), Lars (dash-dotted), Omp (dotted) and Pfp (solid) alongside the theoretical curve ρ_W, on a ρ-δ grid, with $n = 1000$, for: (a) V = Uniform; (b) V = Gauss; (c) V = Bernoulli.

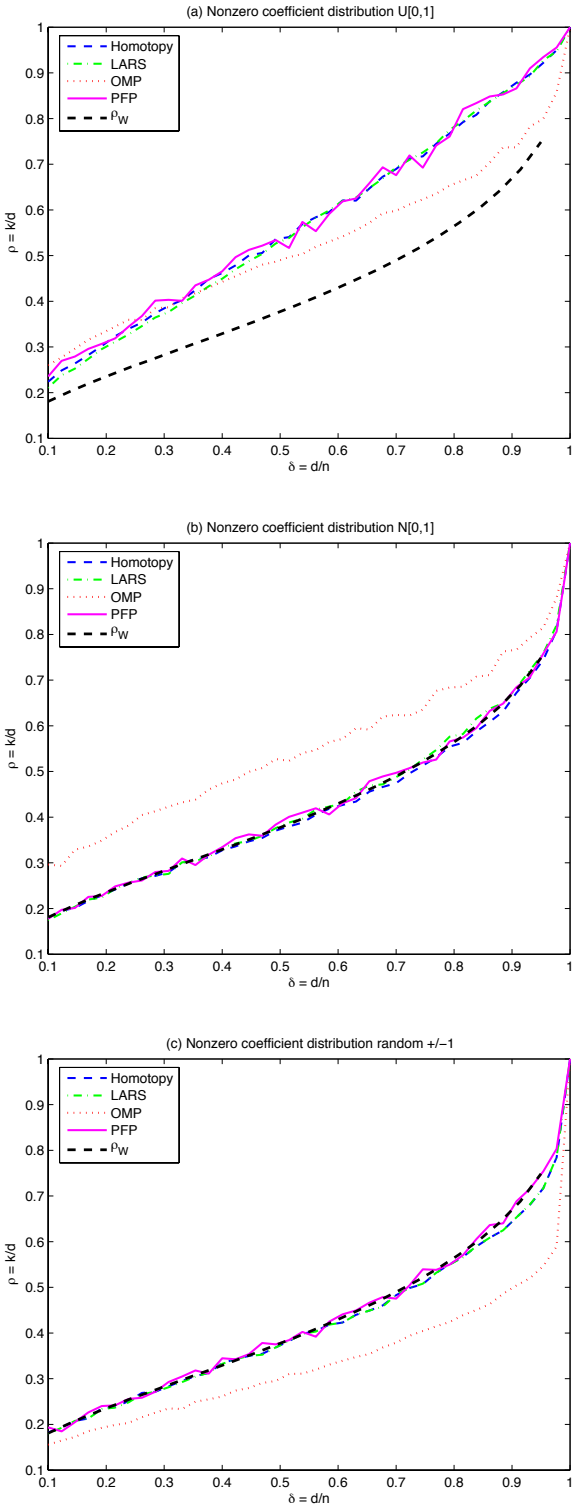

Figure 13: The ℓ_0 Equivalence Property for the Suite $\mathcal{S}(\text{URPE}, \text{V}; d, n, k)$. Each panel has plots of the empirical phase transition curves of Homotopy (dashed), Lars (dash-dotted), Omp (dotted) and Pfp (solid) alongside the theoretical curve ρ_W, on a ρ-δ grid, with $n = 1000$, for: (a) V = Uniform; (b) V = Gauss; (c) V = Bernoulli.

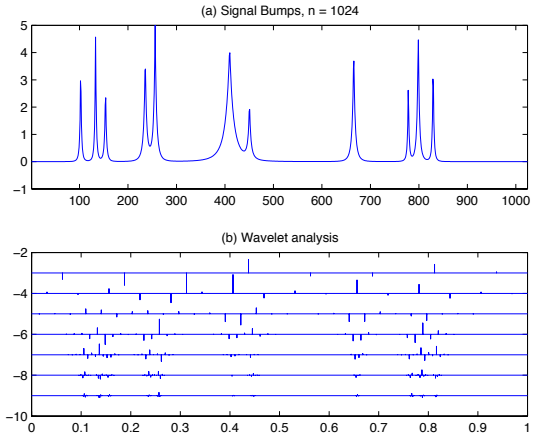

Figure 14: (a) Signal *Bumps*, with $n = 1024$ samples; (b) Its wavelet expansion.

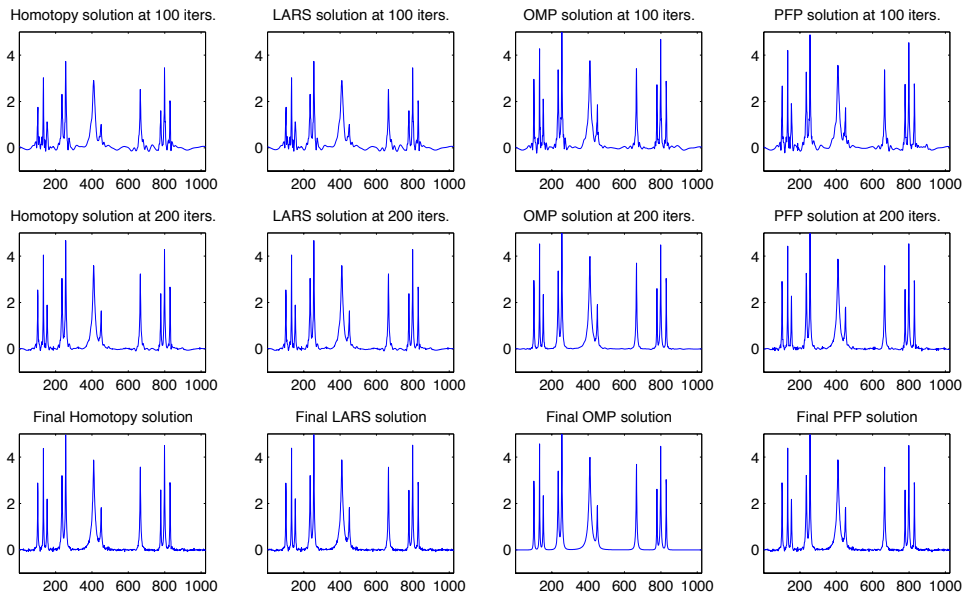

Figure 15: CS reconstruction of *Bumps*: Left to right: Solutions of Homotopy, Lars, Omp, and Pfp; Top to bottom: Solutions after 100 iterations, 200 iterations, and at termination.

10.2 MR Imaging

Magnetic Resonance Imaging (MRI) is a technique based on the same principles as NMR spectroscopy. It is used primarily in medical settings to produce high quality images of internal organs in the human body. Unlike NMR, here we are interested in the spatial location of protons exposed to a magnetic field. Acquisition of an MR image involves adapting the gradients of the external magnetic field to get a spatial frequency map of the exposed organ. As MRI

Algorithm	After 100 iters.		After 200 iters.		Final result	
	$relerr_2$	Time	$relerr_2$	Time	$relerr_2$	Time
HOMOTOPY	0.280	0.6 secs	0.105	1.5 secs	0.052	12.4 secs
LARS	0.280	0.5 secs	0.105	1.5 secs	0.049	10.7 secs
OMP	0.120	0.4 secs	0.021	1.3 secs	0.017	7.0 secs
PFP	0.154	1.5 secs	0.053	3.7 secs	0.050	19.5 secs

Table 5: Error measures and execution times for CS reconstruction of *Bumps* with HOMOTOPY, LARS and OMP.

often deals with enormous volumes of data, acquisition time is a key factor in the success of this technique. Thus, the ability to reconstruct a faithful image from partial frequency information is of great benefit.

The scenario we consider in this section parallels the spectroscopy example, with some notable differences. Most importantly, rather than acquiring 1-D spectra, here we need to map a 3-D volume corresponding to a certain part of the human body. This implies that a huge mass of data needs to be sampled and subsequently reconstructed. Thus, speed becomes a crucial factor; computationally-intensive methods which are acceptable choices for 1-D reconstruction are rendered useless in the 3-D case. In an attempt to reduce acquisition time and speed up computations, traditional methods employ frequency sampling on coarser grids, or only at low frequencies; the results are often blurred or aliased. Here we shall follow the Compressed Sensing paradigm and sample Fourier space at random, with subsequent reconstruction via ℓ_1 minimization.

In detail, we treat the 3-D volume as an array of 2-D slices. Let x_j denote one such slice, column-stacked into a vector of length n. Just as we did in the 1-D case, at the sampling stage, we collect data $y_j = \Phi x_j$, where Φ is a $d \times n$ matrix drawn from the partial Fourier ensemble. In words, we sample d frequencies of x_j at random. Note that in practice, this frequency sampling is done physically by the MRI scanner; this operation is simulated here with a multiplication by Φ. At the reconstruction stage, we solve (CS_1), with the wavelet operator W replaced by its 2-D counterpart. We do so for each slice x_j, until the entire 3-D volume is reconstructed.

In our simulations, we used real data provided to us courtesy of Neal Bangerter of the MRSRL Lab at Stanford University, see [1] for more details. It is a 3-D volume of a part of a human leg, containing 230 slices of dimensions 128×128. Figure 16, panels (a),(d) and (g), show X- and Y-axis projections of the data along with one vertical slice. Clearly, computing the full HOMOTOPY solution would be prohibitively time-consuming, considering the data size. Hence, to approximately solve (CS_1) at the reconstruction stage, we performed only 800 iterations of LARS. For comparative purposes, we also performed 800 iterations of OMP. The results are depicted in the two rightmost columns of Figure 16. Indeed, we see that 800 iterations were sufficient to obtain a visually faithful result. In particular, all the major arteries appearing in the original data are also present in the reconstructed data. Comparison of the reconstructions obtained by LARS and OMP reveals that the reconstruction result of OMP tends to suffer from streaking artifacts, hindering the readability of the final result. Thus, qualitatively, LARS produced a better reconstruction.

10.3 Component Separation in a Noisy Environment

In Section 2, we commented that HOMOTOPY may be easily adapted to treat noisy data, by terminating at a prescribed point on the HOMOTOPY path. We now illustrate the use of this

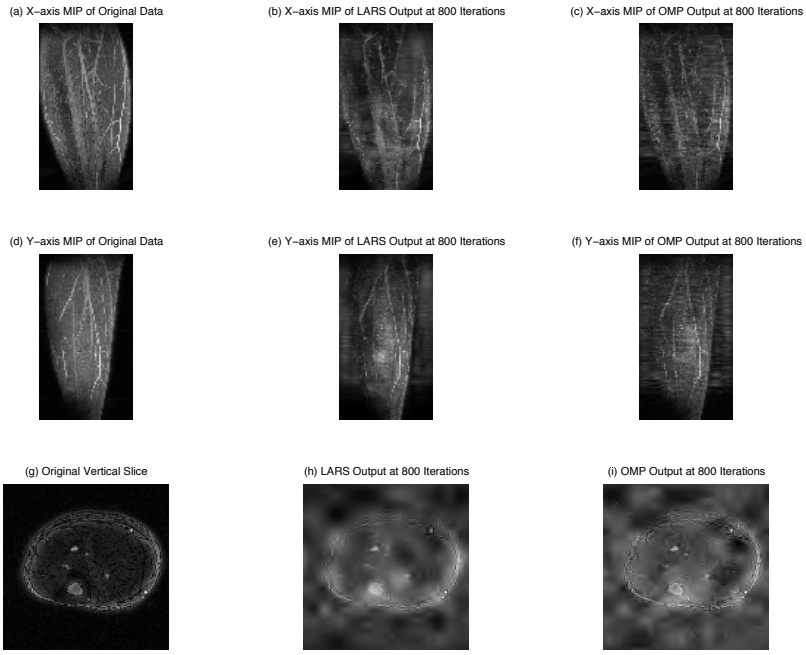

Figure 16: CS reconstruction of 3-D MRI data: Left column: X- and Y-axis projections of the original data, and one vertical slice; Middle column: corresponding reconstructions with LARS; Right column: corresponding reconstructions with OMP.

strategy, in the context of component separation. In the particular example we present here, we consider the problem of separating a signal contaminated with white noise into its harmonic and impulsive components. This relatively simple problem, explored in [9, 20], inspired more ambitious applications such as texture-geometry separation [48] and astronomical image representation [49].

In detail, let y_0 be a signal of length d, composed of a few sinusoids and a few spikes, with a total of k such components, $k \ll d$. Note that y may be written as $y = A\alpha_0$, with $A = [I\ F]$ a $d \times 2d$ matrix representing an overcomplete dictionary composed of the $d \times d$ identity matrix I, and the $d \times d$ Fourier matrix F. α_0 is then a $2d$ coefficient vector, with $\|\alpha_0\|_0 = k$. In our noisy setting, y_0 is not directly observable for measurement. Instead, we measure data $y = y_0 + z$, with the noise power $\|z\|_2 \leq \epsilon_n$. To solve this underdetermined system for the time and frequency components, we apply the HOMOTOPY algorithm, stopping when the residual gets below the noise level ϵ_n.

Figure 17 presents the results of this simulation. Panel (a) displays a signal y_0 of length $d = 512$, composed of 2 harmonic terms superposed with 40 spikes, with amplitudes distributed normally. These are plotted in panels (b) and (c). Panel (d) shows the observed data, buried in white gaussian noise, with a signal to noise ratio of 3. Thus, the noise power is one third the signal power. Panels (e)-(h) on the bottom row of Figure 17 show the corresponding output of the HOMOTOPY algorithm, applied to the data y. As evident from the figure, HOMOTOPY manages to recover most of the components of α_0, leaving mostly noise in the residual. The

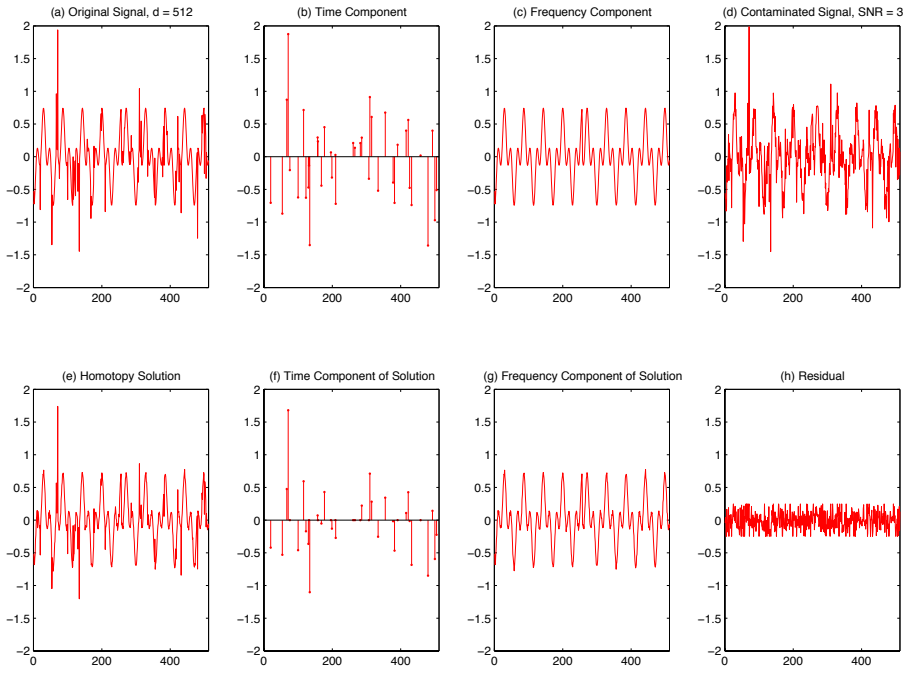

Figure 17: Time-Frequency Separation in the presence of noise: Top row: The original signal, its time and frequency components, and the noisy signal with SNR $= 3$. Bottom row: Corresponding output of HOMOTOPY.

relative reconstruction error is $\|\hat{\alpha} - \alpha_0\|_2 / \|\alpha_0\|_2 = 0.19$. In summary, as this example illustrates, HOMOTOPY may be straightforwardly adapted to deal with noisy data, to successfully recover sparse solutions of underdetermined systems.

11 Accompanying Software

`SparseLab` is a library of Matlab functions we make available on the web at *http://sparselab.stanford.edu*. It has been developed in the spirit of *reproducible research* [4, 10], namely the idea that publication of computational research demands also the publication of the complete software ennvironment needed for reproducing that research. SparseLab has been used by the authors to create the figures and tables used in this article, and the toolbox contains scripts which will reproduce all the calculations of this paper. The current release includes about 400 Matlab files, datasets, and demonstration scripts.

12 On Approximate ℓ_1 Minimization Algorithms

In the belief that true ℓ_1 minimization is far too slow for large-scale applications, numerous authors have explored approximate algorithms based on iterative thresholding or similar ideas; examples include [23, 11, 48, 49, 25]; related ideas include projection onto convex sets [6]. The reader should keep in mind that there are applications where such methods indeed operate faster

than the methods discussed here, while still producing a faithful approximate solution; see, for example, Section 9 of [23]. On the other hand, the present methods for ℓ_1 minimization are much faster than general purpose optimizers, so much of the apparent advantage claimed for approximate ℓ_1 schemes has been neutralized by the algorithms studied here. Furthermore, approximate methods inevitably trade off aptitude for speed; in general, the range of sparsities in which such methods reliably recover the sparsest solution is considerably limited, compared to ℓ_1 minimization. A further advantage of the HOMOTOPY viewpoint is the stimulus it provides for future research: the empirical phenomena identified here may lead to deep research in polytope theory, perhaps comparable to the kinds of results in [22].

13 Conclusions

In this paper, we study the HOMOTOPY algorithm, introduced by Osborne *et al.* [40] and Efron *et al.* [24]. It was originally proposed in the context of overdetermined noisy systems. We propose here to use HOMOTOPY to obtain sparse solutions of underdetermined systems of linear equations. HOMOTOPY follows a piecewise linear solution path, starting from the zero solution, and terminating at the solution of (P_1). It maintains an active set of column vectors of the matrix A, adding or removing a vector at each step, and updating the solution accordingly. It is a particularly effective algorithm for solving ℓ_1 minimization problems when the underlying solution is sparse, and the underlying system is incoherent or random. In such scenarios, it significantly outperforms state-of-the-art general LP solvers. Moreover, its computational complexity is then comparable to greedy stepwise algorithms such as OMP. In fact, the similarity between the HOMOTOPY, LARS, and OMP algorithms illuminates the parallelism evident in results on the performance of ℓ_1 minimization and OMP in recovering sparse solutions to underdetermined linear systems.

Such claims we made concrete by formally showing that for two matrix classes, deterministic incoherent matrices and random uniform spherical matrices, HOMOTOPY has the following *k-step solution* property: If the data admit sparse representation using k columns of A, HOMOTOPY finds it in exactly k steps. We proved that for problems drawn from the suite $\mathcal{S}(\text{INC}_\mu; d, n, k)$, HOMOTOPY has the k-step solution property whenever $k < (\mu^{-1} + 1)/2$. We also observed empirically that for instances of the problem suite $\mathcal{S}(\text{USE}; d, n, k)$, HOMOTOPY has the k-step solution property with high probability when k is less than a threshold, empirically $k < d/(2\log(n)) \cdot (1 - \epsilon_n)$.

Using the notions of phase diagrams and phase transitions, we presented empirical evidence supporting these assertions. We showed that for problem instances drawn from $\mathcal{S}(\text{USE}; d, n, k)$, the k-step solution property of HOMOTOPY exhibits an empirical phase transition at $d/(2\log(n))$, regardless of the distribution of the nonzeros in the solution. Simulations with the suite $\mathcal{S}(\text{RSE}; d, n, k)$ resulted in similar findings. For the PFE and PHE, we noted that HOMOTOPY has the k-step solution property at a much wider range of sparsities, compared with the USE. Thus, in many applications involving PFE and PHE matrices, HOMOTOPY would be particularly efficient in finding the sparsest solution. Similar conclusions were made for the problem suite $\mathcal{S}(\text{URPE}; d, n, k)$.

Comparison of the performance of HOMOTOPY, LARS, OMP, and PFP in recovering sparse solutions led to several interesting findings. First, we observed that LARS, an approximation to HOMOTOPY that only adds terms to its active set, performed remarkably similarly to HOMOTOPY. Second, both algorithms exhibited an increase in performance when applied to problem instances from the suites $\mathcal{S}(\text{URPE}; d, n, k)$, $\mathcal{S}(\text{PFE}; d, n, k)$ and $\mathcal{S}(\text{PHE}; d, n, k)$. The performance of OMP was comparable in some cases, yet it was highly affected by the underlying

distribution of the nonzero coefficients. In particular, when the coefficients follow a random sign pattern with constant amplitude, OMP's region of successful recovery was significantly smaller than that of HOMOTOPY or LARS.

HOMOTOPY extends naturally to deal with noisy data, and produces satisfactory results even when it is stopped before completing the solution path. Examples inspired by applications in NMR spectroscopy and MR imaging were given to demonstrate its usefulness in practical scenarios. These examples and all the simulations in this paper are reproducible with the accompanying software.

Note

At one point, we made an effort to include Iddo Drori in this project. Drori has since presented simulation results related to some of our work in Section 5 above in a talk at the 2006 ICASSP meeting.

References

[1] N.K. Bangerter, B.A. Hargreaves, J.H. Brittain, B. Hu, S.S. Vasanawala, and D.G. Nishimura. 3d fluid-suppressed t2-prep flow-independent angiography using balanced ssfp. In *Proc. of the 12th ISMRM*, page 11, 2004.

[2] Michel Berkelaar, Kjell Eikland, and Peter Notebaert. Lp_solve: (mixed-integer) linear programming system. http://lpsolve.sourceforge.net.

[3] William L. Briggs and Van Emden Henson. *The DFT: An Owner's Manual for the Discrete Fourier Transform*. SIAM, 1995.

[4] Jon Buckheit and David L. Donoho. Wavelab and reproducible research. *Wavelets and Statistics*, 1995.

[5] Emmanuel Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, December 2005.

[6] Emmanuel J. Candès and Justin Romberg. Practical signal recovery from random projections. *Manuscript*, 2005.

[7] Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.

[8] Emmanuel J. Candès and Terence Tao. Stable signal recovery from incomplete and inaccurate observations. *Comm. Pure Appl. Math.*, 59(8):1207–1223, August 2006.

[9] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci Comp.*, 20(1):33–61, 1999.

[10] Sou Cheng Choi, David L. Donoho, Ana Georgina Flesia, Xiaoming Huo, Ofer Levi, and Danzhu Shi. Beamlab web site. http://www-stat.stanford.edu/~beamlab/.

[11] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–1541, August 2004.

[12] Geoff Davis, Stèphane Mallat, and Marco Avellaneda. Adaptive greedy approximations. *Journal of Constructive Approximation*, 13:57–98, 1997.

[13] David L. Donoho. Neighborly polytopes and sparse solution of underdetermined linear equations. Technical report, Dept. of Statistics, Stanford University, 2005.

[14] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[15] David L. Donoho. For most large underdetermined systems of linear equations, the minimal ℓ^1 solution is also the sparsest solution. *Comm. Pure Appl. Math.*, 59(7):907–934, July 2006.

[16] David L. Donoho. For most underdetermined systems of linear equations, the minimal ℓ^1-norm near-solution approximates the sparsest near-solution. *Comm. Pure Appl. Math.*, 59(6):797–829, June 2006.

[17] David L. Donoho. High-dimensional centrosymmetric polytopes with neighborliness proportional to dimension. *Discrete and Computational Geometry*, 35(4):617–652, May 2006.

[18] David L. Donoho and Michael Elad. Optimally sparse representation from overcomplete dictionaries via ℓ^1 norm minimization. *Proc. Natl. Acad. Sci. USA*, 100(5):2197–2002, March 2003.

[19] David L. Donoho, Michael Elad, and Vladimir N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, January 2006.

[20] David L. Donoho and Xiaoming Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Trans. Info. Theory*, 47(7):2845–2862, 2001.

[21] David L. Donoho and Benjamin F. Logan. Signal recovery and the large sieve. *SIAM J. Appl. Math.*, 52(2):577–591, 1992.

[22] David L. Donoho and Jared Tanner. Counting faces of randomly-projected polytopes when the projection radically lowers dimension, 2006.

[23] David L. Donoho, Yaakov Tsaig, Iddo Drori, and Jean-Luc Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. *Submitted*, 2006.

[24] Bradley Efron, Trevor Hastie, Iain M. Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

[25] M. Elad, B. Matalon, and M. Zibulevsky. Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization. *Appl. Comp. Harm. Anal.*, to appear.

[26] Michael Elad and Alfred M. Bruckstein. A generalized uncertainty principle and sparse representations in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, September 2002.

[27] Jean J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, June 2004.

[28] A. Y. Garnaev and E. D. Gluskin. On widths of the euclidean ball. *Soviet Mathematics – Doklady*, 30:200–203, 1984. (in English).

[29] Anna C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal sparse fourier representations via sampling. In *Proc. 34th ACM symposium on Theory of Computing*, pages 152–161. ACM Press, 2002.

[30] Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[31] Rémi Gribonval and Morten Nielsen. Sparse representations in unions of bases. *IEEE Trans. Info. Theory*, 49(12):1320–1325, 2003.

[32] Rémi Gribonval and Morten Nielsen. Highly sparse representations from dictionaries are unique and independent of the sparseness measure. *preprint*, 2006.

[33] M. Harwit and N.J.A. Sloane. *Hadamard Transform Optics*. Academic Press, 1979.

[34] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[35] Jeffrey C. Hoch and Alan Stern. *NMR Data Processing*. Wiley-Liss, 1996.

[36] Boris S. Kashin. Diameters of certain finite-dimensional sets in classes of smooth functions. *Izv. Akad. Nauk SSSR, Ser. Mat.*, 41(2):334–351, 1977.

[37] Michael Lustig, Jin H. Lee, David L. Donoho, and John M. Pauly. Faster imaging with randomly perturbed spirals and ℓ_1 reconstruction. In *Proc. of the 13th ISMRM*, 2004.

[38] Michael Lustig, J.M. Santos, David L. Donoho, and John M. Pauly. Rapid mr angiography with randomly under-sampled 3dft trajectories and non-linear reconstruction. In *Proc. of the 9th SCMR*, 2006.

[39] Dmitry M. Malioutov, Müjdat Çetin, and Alan S. Willsky. Homotopy continuation for sparse signal representation. In *IEEE Int. Conf. Acoustics, Speech and Signal Processing, Philadelphia, PA*, volume 5, pages 733–736, March 2005.

[40] Michael R. Osborne, Brett Presnell, and Berwin A. Turlach. A new approach to variable selection in least squares problems. *IMA J. Numerical Analysis*, 20:389–403, 2000.

[41] Michael R. Osborne, Brett Presnell, and Berwin A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9:319–337, 2000.

[42] M. D. Plumbley. Geometry and homotopy for l1 sparse representations. In *Proceedings of SPARS'05, Rennes, France*, pages 206–213, November 2005.

[43] M. D. Plumbley. Recovery of sparse representations by polytope faces pursuit. In *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Source Separation (ICA 2006), Charleston, SC, USA*, pages 206–213, March 2006.

[44] Mark Rudelson and Roman Vershynin. Geometric approach to error-correcting codes and reconstruction of signals. Technical report, Department of Mathematics, Univ. of California at Davis, 2005.

[45] Michael A. Saunders and Byunggyoo Kim. Pdco: Primal-dual interior method for convex objectives. http://www.stanford.edu/group/SOL/software/pdco.html.

[46] Peter Schmieder, Alan S. Stern, Gerhard Wagner, and Jeffrey C. Hoch. Application of nonlinear sampling schemes to cosy-type spectra. *Journal of Biomolecular NMR*, 3(134):569–576, 1993.

[47] Peter Schmieder, Alan S. Stern, Gerhard Wagner, and Jeffrey C. Hoch. Improved resolution in triple-resonance spectra by nonlinear sampling in the constant-time domain. *Journal of Biomolecular NMR*, 4(188):483–490, 1994.

[48] Jean-Luc Starck, Michael Elad, and David L. Donoho. Image decomposition: Separation of texture from piecewise smooth content. In *SPIE annual meeting, San Diego, California, USA*, December 2003.

[49] Jean-Luc Starck, Michael Elad, and David L. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Advances in Imaging and Electron Physics*, 132:287–348, 2004.

[50] Thomas Strohmer and Robert Heath Jr. Grassmannian frames with applications to coding and communcations. *Appl. Comp. Harm. Anal.*, 14(3):257–275, 2003.

[51] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

[52] Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(11):2231–2242, October 2004.

[53] Joel A. Tropp. Just relax: Convex programming methods for subset selection and sparse approximation. *IEEE Transactions on Information Theory*, 51(3):1030–1051, March 2006.

[54] Joel A. Tropp and Anna C. Gilbert. Signal recovery from partial information by orthogonal matching pursuit. *Manuscript*, 2005.

[55] Yaakov Tsaig and David L. Donoho. Breakdown of equivalence between the minimal ℓ^1-norm solution and the sparsest solution. *Signal Processing*, 86(3):533–548, 2006.

[56] Yaakov Tsaig and David L. Donoho. Extensions of compressed sensing. *Signal Processing*, 86(3):549–571, 2006.