Freely Available, Optimally Tuned Iterative Thresholding Algorithms for Compressed Sensing

Arian Maleki, David L. Donoho

Freely Available, Optimally Tuned Iterative Thresholding Algorithms for Compressed Sensing

Arian Maleki
Department of Electrical Engineering
Stanford University
arianm@stanford.edu

David L. Donoho
Department of Statistics
Stanford University
donoho@stanford.edu

***Abstract*— We conducted an extensive computational experiment, lasting multiple CPU-years, to optimally select parameters for important classes of algorithms for finding sparse solutions of underdetermined systems of linear equations. We make the optimally tuned implementations freely available at `sparselab.stanford.edu`; they can be used 'out of the box' with no user input: it is not necessary to select thresholds or know the likely degree of sparsity.**

Our class of algorithms includes iterative hard and soft thresholding with or without relaxation, as well as CoSaMP, Subspace Pursuit and some natural extensions. As a result, our optimally tuned algorithms dominate such proposals.

Our notion of optimality is defined in terms of phase transitions, i.e. we maximize the number of nonzeros at which the algorithm can successfully operate. We show that the phase transition is a well-defined quantity with our suite of random underdetermined linear systems. Our tuning gives the highest transition possible within each given class of algorithms. We verify by extensive computation the robustness of our recommendations to the amplitude distribution of the nonzero coefficients as well as the matrix ensemble defining the underdetermined system.

Several specific findings are established. (a) For all algorithms the worst amplitude distribution for nonzeros is generally the constant-amplitude random-sign distribution; where all nonzeros are the same size. (b) Various random matrix ensembles give the same phase transitions; random partial isometries give different transitions and require different tuning; (c) Optimally tuned Subspace Pursuit dominates optimally tuned CoSaMP, particularly so when the system is almost square. (d) For randomly decimated partial Fourier transform sampling, our recommended Iterative Soft Thresholding gives extremely good performance, making more complex algorithms like CoSaMP and Subspace Pursuit relatively pointless.

I. Introduction

A recent flood of publications offers practical algorithms for the task of obtaining sparse solutions of underdetermined systems of linear equations. Such algorithms have potential applications in fields ranging from medical imaging and wireless digital communications.

The interested user now has a bewildering variety of ideas and variations that might be used in applications, but this, paradoxically, creates uncertainty and can cause potential beneficiaries of such algorithms to avoid the topic entirely. In this note we announce a solution to this problem, in the form of freely available, optimally tuned algorithms ready for use 'out of the box'. Our tuning is based on a comprehensive study of parameter variations and options. It would have required several years to complete our study on a single modern desktop computer. Optimal tuning manages to make some very simple and unsexy ideas perform surprisingly well, reducing the need for more ambitious and impressive sounding ones (even if optimally tuned). Our tuning is based on quantitative principles, it can be used for other algorithms as well and implicitly establishes the 'current state of the art' which new methods ought to be compared to. It also generates insights previously unavailable about performance comparisons of methods and performance comparisons of different matrix ensembles.

II. Iterative Algorithms

Our problem setting will be described with the following notation. An unknown vector $x_0 \in \mathbb{R}^N$ is of interest; we have measurements $y = Ax_0$. Here A is an $n \times N$ matrix and $N > n$. Although the system is underdetermined, it has been shown that sufficient *sparsity* of x_0 may allow unique solution. We say that x_0 is k-sparse if it has at most k nonzeros. In many cases one can exactly recover such a sparse solution x_0 as the solution to

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

where $\|x\|_1$ denotes the ℓ_1 norm. This amounts to a large-scale linear programming problem. Unfortunately in some interesting potential applications [1], [2], the matrix A and vector x_0 may contain millions of entries and large-scale linear programming is too slow for those applications. Hence there is widespread interest in finding fast algorithms that work essentially as well; in particular lots of practical work by Jean-Luc Starck and co-authors and Elad and co-authors has shown that some very simple iterative algorithms can be strikingly successful on very large problems [3], [4]. In this note we consider two families of such iterative algorithms.

A. Simple Iterative Algorithms

The first family is inspired by the classical relaxation method for approximate solution of large linear systems. In classical relaxation, one iteratively applies A and A' to appropriate vectors and under appropriate conditions, the correct solution is obtained as a limit of the process. While the classical theory is inapplicable to underdetermined systems, it has been found both empirically and in theory that a sparsity-promoting variant of relaxation can correctly solve such systems, when they have sparse solutions.

$$x_{i+1} = \eta(x_i + \kappa \cdot (A' r_i)); \qquad r_i = y - Ax_i;$$

Here κ is a relaxation parameter $(0 < \kappa < 1)$ and η is a nonlinear thresholding rule; we consider both Hard thresholding – $\eta_t^H(y) = y1_{\{|y|>t\}}$ and Soft thresholding $\eta_t^S(y) = sgn(y)(|y| - t)_+$. Note that if we set $\eta(y) = y$ we would just have classical relaxation. The idea of iterative soft thresholding with a fixed threshold was first introduced in [5] for solving the Lasso problem. It was then extended to the ℓ_0 regularized problem which resulted in fixed threshold hard thresholding [6]. However the thresholding policies considered here may depend on the iteration and on other measured signal properties.

Such a thresholding scheme is easy to implement: it requires only two matrix vector products per iteration and some vector additions

and subtractions. For certain very large matrices we can rapidly apply A and A' without representing A as a full matrix – examples include partial Fourier and Hadamard transforms. In such settings, the work required scales very favorably with N eg as (eg $N \log(N)$ flops rather than $O(N^2)$).

Actually using such a scheme in practice requires choosing a parameter vector $\theta = (type, \kappa, t)$ here type = S or H depending as soft or hard thresholding is required; the other parameters are as earlier. Moreover t needs to vary from iteration to iteration. The general form in which such schemes are often discussed does not give a true ready-to-run algorithm. This is akin to presenting a recipe listing the ingredient for a dish, without listing the amounts; it keeps potential users from successfully exploiting the idea.

B. Composite Iterative Algorithms

In solving determined linear systems, relaxation can often be outperformed by other methods. Because of the similarity of relaxation to iterative soft/hard thersholding (IST/IHT) schemes, parallel improvements seem worth pursuing in the sparsity setting. A more sophisticated scheme – Two Stage Thresholding –uses exact solution of small linear systems combined with thresholding before and after this solution. In stage one, we screen for 'significant' nonzeros just as in IST and IHT:

$$v_i = \eta^1(x_i + \kappa A' r_i); \qquad r_i = Y - Ax_i;$$

We let I_i denote the combined support of v_i and x_i and we solve

$$w_i = (A'_{I_i} A_{I_i})^{-1} A'_{I_i} y$$

We then threshold a second time,

$$x_{i+1} = \eta^2(w_i),$$

producing a sparse vector. Here the threshold might be chosen differently in stages 1 and 2 and might depend on the iteration and on measured signal properties.

It seems that the use of explicit solutions to the smaller systems might yield improved performance, although at the cost of potentially much more expense per iteration . An important problem is user reticence. In the case of TST there are even more choices to be made than in the case to be made with IST/IHT. This scheme again presents the 'recipe ingredients without recipe amounts' obstacle: users may be turned off by the requirement to specify many such tunable parameters.

C. Threshold Choice

Effective choice of thresholds is a heavily-developed topic in statistical signal processing. We have focused on two families of tunable alternatives.

Interference heuristic. We pretend that the marginal histogram of $A'r$ at sites in the coefficient vector where $x_0(i) = 0$ is Gaussian, with common standard deviation. We estimate the standard deviation of $A'r$ robustly at a given iteration and set the threshold t as a fixed multiple $\lambda \cdot \sigma$. The underlying rationale for this approach is explained in [10] where its correctness was heavily tested. Under this heuristic, we control the threshold λ as in standard detection theory using the False Alarm Rate; thus $2 \cdot FAR = \Phi(\lambda)$ where Φ denotes the standard normal distribution function.

Oracle Heuristic. In the TST scheme imagine that an oracle tells us the true underlying k, and we scale the threshold adaptively at each iteration so that at stage 1 we yield $\alpha \cdot k$ nonzeros and at stage two $\beta \cdot k$ nonzeros. The method CoSaMP [7] corresponds to $\beta = 2$, $\alpha = 1$, while Subspace Pursuit [8] corresponds to $\beta = \alpha = 1$.

A problem with the Oracle Heuristic is that, in interesting applications, there is no such oracle, meaning that we wouldn't in practice ever know what k to use. A problem with the Interference heuristic is that the Gaussian model may not work when the matrix is not really 'random'.

III. Phase Transition

In the case of ℓ_1 minimization with A a random matrix there is a well-defined 'capacity': ℓ_1 can successfully recover the sparsest solution provided k is smaller than a certain definite fraction of n. Let $\delta = n/N$ be a normalized measure of problem indeterminacy and let $\rho = k/n$ be a normalized measure of the sparsity. We get a two-dimensional phase space $(\delta, \rho) \in [0,1]^2$ describing the type of problem. This phase space has an interesting two-phase structure.

Let A be a random matrix with iid Gaussian entries and let $y = Ax_0$ with x_0 k-sparse. The probability that (P_1) recovers the sparsest solution to $y = Ax$ tends to 1 or 0 with increasing system size according as k is above or below the curve $(\delta, \rho(\delta))$ [9].

Other algorithms also exhibit sharp phase transitions [10], [11]. Figure 1 displays behavior of IHT with FAR threshold selection, at a single fixed choice n/N with varying underlying number k of nonzeros. Below a certain threshold, the algorithm works well and above that threshold it fails; the transition zone is narrow, and gets better defined at large problem sizes N.

IV. Tuning Procedure

We conducted extensive computational experiments to evaluate the phase transitions of various algorithms. In all, we performed more than 90000000 reconstructions, using 38 servers at a commercial dedicated server facilities for one month. These calculations would have run more than 3 years on a single desktop computer.

For a fixed iterative scheme and a fixed tuning parameter θ, we considered in turn each of several problem suites $\mathcal{S} = (E, C)$, i.e. several random matrix ensembles E and several coefficient amplitude distributions C. At each combination we recorded metrics of algorithm success while varying the sparsity and indeterminacy of the problem.

In the tuning stage of our project we worked only with the *standard suite* $\mathcal{S}_0$ formed with the USE matrix ensemble and constant amplitude distribution on the nonzeros. In the later evaluation stage, other problem suites were used to test the robustness of the tuning.

For a fixed $N = 800$, we varied n and k through a grid of 900 δ and ρ combinations, with δ varying from .05 to 1 in 29 steps and ρ varying from .05 up to a ceiling value $\rho_{max} < 1$ in as many as 30 steps. At each grid point we solved $M = 100$ different random problem instances, obtaining the observed success fraction $Succ(\delta, \rho; \theta, \mathcal{S})$.

We measured the empirical phase transition by varying k with all other variables held fixed, searching for the 50% success point, and expressing k in proportional terms, we obtained

$$\rho^*(\delta; \theta) = \max\{k/n : Succ(n/N, \rho'; \theta) > .5, \forall \rho' < k/n\} \quad (1)$$

We chose the optimal parameter via

$$\theta^*(\delta) = \arg\max_{\theta} \rho^*(\delta; \theta). \quad (2)$$

V. Tuning Results

Figure 2 illustrates tuning results for IST on the standard suite $\mathcal{S}_0$. Here $\theta = (RelaxationParameter, FARParameter)$. Panel (a) shows the different optimized phase transitions available by tuning FAR to depend on δ while the relaxation parameter is fixed. Panel (b) shows the optimally tuned FAR parameters at each given δ and

choice of relaxation parameter. Figures 3 offer the same information for IHT.

Optimum performance of IST occurs at higher values of the false alarm rate than for IHT. Decreasing the relaxation parameter beyond the range shown here does not improve results for IST and IHT.

Figure 4 illustrates tuning results for the variant of TST based on FAR threshold selection. Again $\theta = (RelaxationParameter, FARParameter)$. The phase transitions correspond to different fixed choices of FAR at fixed levels of the relaxation parameter. The results depend very weakly on FAR.

Figure 5 illustrates performance of TST for different values of $\theta = (\alpha, \beta)$. Panel (a) shows the different optimized phase transitions available by tuning β at fixed $\alpha = 1$ and Panel (b) shows optimal phase transitions with $\alpha = \beta$ varying. Both displays point to the conclusion that $\alpha = \beta = 1$ dominates other choices. Hence Subspace Pursuit $(\alpha = 1, \beta = 1)$ dominates CoSaMP $(\alpha = 1, \beta = 2)$.

VI. Recommended Choices

We provide three versions of iterative algorithms based on our optimal tuning exercise: Recommended-IST, Recommended-IHT and Recommended-TST. They are implemented in Matlab and published at URL `sparselab.stanford.edu/ReadyToRun`.

In our recommended versions, there are no free parameters. The user specifies only the matrix A and the left-hand side y. In particular the user does not specify the expected sparsity level, which in most applications cannot be considered known.

These recommended algorithms are not the same as previously published algorithms. For example, Recommended TST has parameters $\alpha = 1$ and $\beta = 1$, so it initially seems identical to Subspace Pursuit [8]. However, Subspace Pursuit demands an *oracle* to inform the user of the true underlying sparsity of the vector. Recommended-TST has embedded in it a maximum value for the assumed sparsity level (see Table III). If the actual sparsity in x_0 is better than the maximin value, the algorithm still works, but if it is worse, the algorithm won't work anyway. The user does not need to know this number – it is baked into the code. In effect, we have de-oracle-ized the Subspace Pursuit method.

We remind the reader that these algorithms dominate other implementations in the same class. Thus, Recommended TST dominates CoSamp; this is particularly evident for $\delta > .5$ (see Figure 5).

A companion set of algorithms – described later – is available for the case where A is not an explicit matrix but instead a linear operator for which Av and $A'w$ can be computed without storing A as a matrix. Some differences in tuning for that case have been found to be valuable.

We record in the following tables a selection of the optimally tuned parameter values.

TABLE I
RECOMMENDED CHOICES OF FAR IN IST

δ	.05	.11	.21	.31	.41	.5	.6	.7	.8	.93
ρ	.124	.13	.16	.18	.2	.22	.23	.25	.27	.29
FAR	.02	.037	.07	.12	.16	.2	.25	.32	.37	.42

VII. Robustness

A *robust* choice of parameters offers a guaranteed level of performance across all situations. Such a choice can be made by solving

TABLE II
RECOMMENDED CHOICES OF FAR IN IHT

δ	.05	.11	.21	.41	.5	.6	.7	.8	.93
ρ	.12	.16	.18	.25	.28	.31	.34	.38	.41
FAR	.0015	.002	.004	.011	.015	.02	.027	.035	.043

TABLE III
RECOMMENDED CHOICES OF ρ TO BE USED IN TST

δ	.05	.11	.21	.31	.41	.5	.6	.7	.8	.93
ρ	.124	.17	.22	.26	.30	.33	.368	.4	.44	.48

the maximin problem

$$\theta^r(\delta) = \arg\max_{\theta} \min_{\mathcal{S}} \rho^*(\delta; \theta; \mathcal{S}).$$

The maximin is achieved at the *least-favorable* suite. Our tuning results were obtained at the standard suite $\mathcal{S}_0$, with constant amplitude, random-sign coefficients and matrices from USE. We considered other suites by varying the matrix ensemble C, including uniformly-distributed random projections and Bernoulli matrices with random signs. We also considered four coefficient ensembles C: in addition to the constant amplitude ensemble, we considered coefficients from the double exponential distribution, the Cauchy, and the uniform distribution on $[-1, 1]$.

Figures 6-7-8 display results for Recommended-IST, Recommended-IHT, and Recommended-TST. The constant amplitude, random sign ensemble is the least favorable input distribution for all three methods. Since we have tuned at that ensemble, our choice of tuning parameters can be said to be robust.

Figures 9-10-11 study Recommended-IST, Recommended-IHT, and Recommended-TST at three matrix ensembles. Results are similar for the Bernoulli and USE ensembles, and noticeably different – mostly better – for URP. Since we have tuned at USE our choice of tuning parameters can be called robust.

A surprising exception to the above pattern is described below in Section X.

VIII. Comparisons of Algorithms

Figure 12 compares our Recommended algorithms with ℓ_1. We have the following ordering

$$\ell_1 > Rec - TST > Rec - IHT > Rec - IST.$$

IX. Running Times

Algorithm running times were measured on an "Intel 2 Core Processor 2.13GHz" with 3GBytes RAM. All implementations are in Matlab, and in each case the iterative algorithms were run for 50 iterations.

X. Ensembles with Fast Operators

The matrix ensembles studied so far all used dense matrices with random elements. However, many applications of sparsity-seeking decompositions use linear operators which are never stored as matrices. An example is the partial Fourier ensemble [2]. Here the $n \times N$ matrix A has for its rows a random subset of the N rows in the standard Fourier transform matrix. Av and $A'w$ can both

TABLE IV

RUNNING TIMES (SEC) FOR 50 ITERATIONS OF RECOMMENDED ALGORITHMS. STANDARD SUITE.

N	n	IST	IHT	TST
4000	800	4.5	4.47	3.56
4000	1200	6.66	6.65	18.07
4000	1600	9.02	9.00	51.67
8000	1000	11.1	11.08	7.21
8000	2000	22.04	22.04	72.01
8000	4000	44.73	44.67	871.2

be computed in order $N \log(N)$ time; the comparable dense matrix vector products would cost order N^2 flops.

The Simple Iterative Algorithms IHT and IST are particularly suited for use in the FastOps setting, since they essentially amount to repetitive application of Av and $A'w$ interleaved with thresholding.

A version of our recommended algorithms is available for the FastOps case. We have found that a custom tuning in the Fourier case, can produce performance far exceeding anything seen in the 'random' matrix ensembles.

Recommended parameters are shown in Tables V-VI. The running time of the algorithms for partial Fourier are studied in Table VII

TABLE V

OPTIMAL VALUES ρ AND FAR FOR IST. PARTIAL FOURIER ENSEMBLE.

δ	.11	.21	.31	.41	.5	.6	.7	.8	.9
ρ	.32	.40	.47	.51	.56	.59	.62	.62	.76
FAR	.026	.063	.098	.13	.16	.19	.22	.25	.26

TABLE VI

OPTIMAL ρ AND FAR FOR IHT. PARTIAL FOURIER ENSEMBLE.

δ	.11	.21	.31	.41	.5	.6	.7	.8
ρ	.35	.4	.47	.5	.5	.5	.5	.5
FAR	.001	.0015	.002	.0025	.003	.0035	.004	.0045

TABLE VII

RUNNING TIMES (SEC) FOR 50 ITERATIONS OF RECOMMENDED ALGORITHMS. PARTIAL FOURIER ENSEMBLE.

N	n	IST, IHT
65536	12000	1.41
262144	24000	6.33
262144	50000	6.53

REFERENCES

[1] D. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, Vol. 52, pp. 489-509, April 2006.

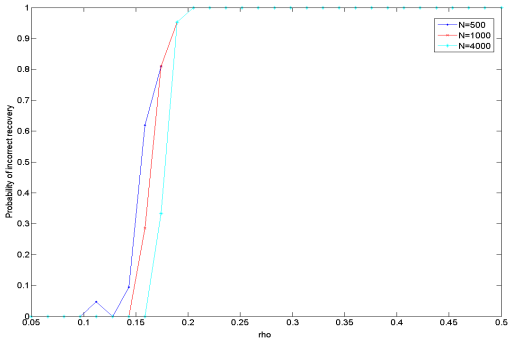

Fig. 1. Fraction of unsuccessful reconstructons by IHT. Here $\delta = .5$ and $\rho = k/n$ is varying. FAR parameter $= 10^{-3}$. Relaxation parameter = 1. At each unique parameter combination, twenty random problem instances were tried. Results are shown at 3 values of N: 500, 1000, 4000

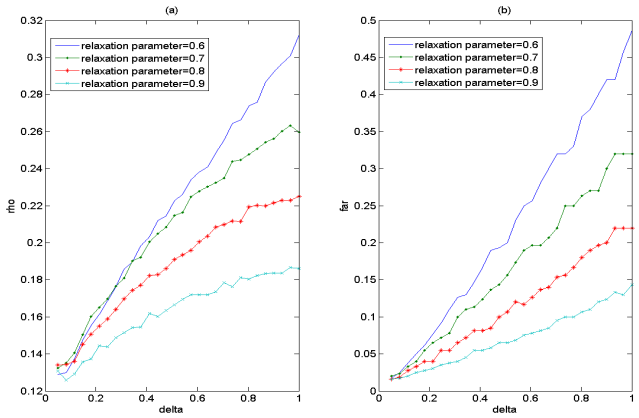

Fig. 2. (a) Optimum phase transitions for IST at fixed relaxation parameter (b) FAR parameter choice yielding the optimum

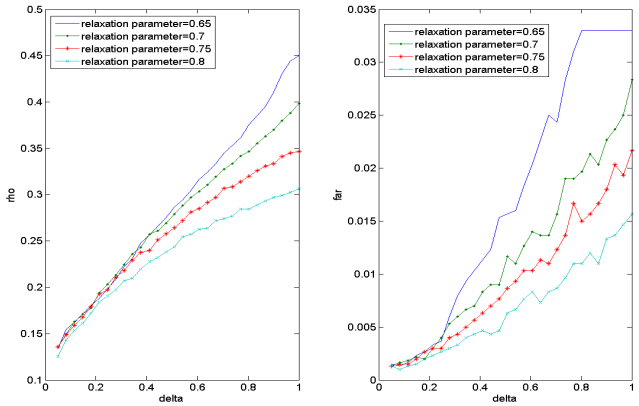

Fig. 3. (a) Optimum phase transitions for IHT at fixed relaxation parameter (b) FAR parameter choice yielding the optimum

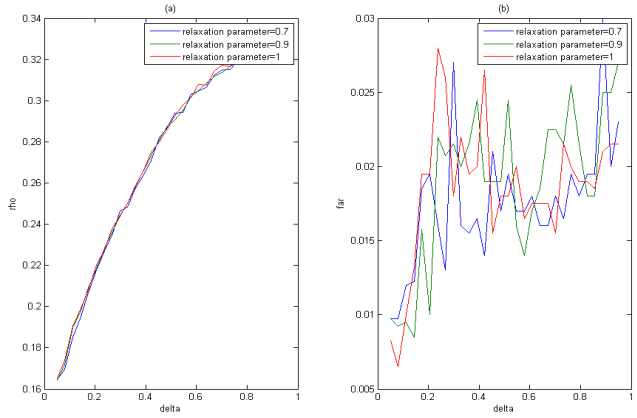

Fig. 4. (a) Optimum phase transitions for TST-FAR at fixed relaxation parameter (b) FAR parameter choice yielding the optimum

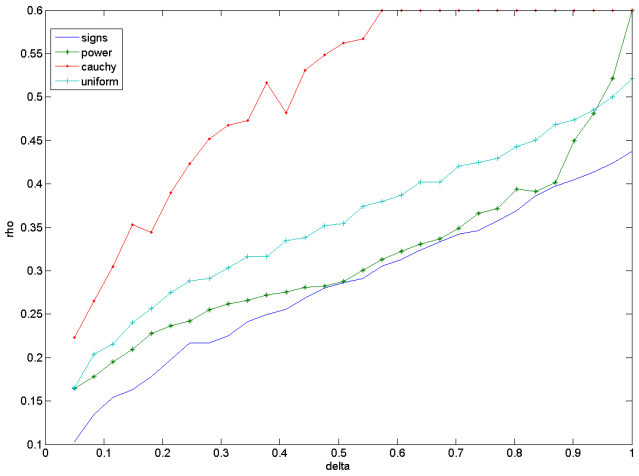

Fig. 7. Observed Phase Transition of Recommended IHT different coefficient ensembles..

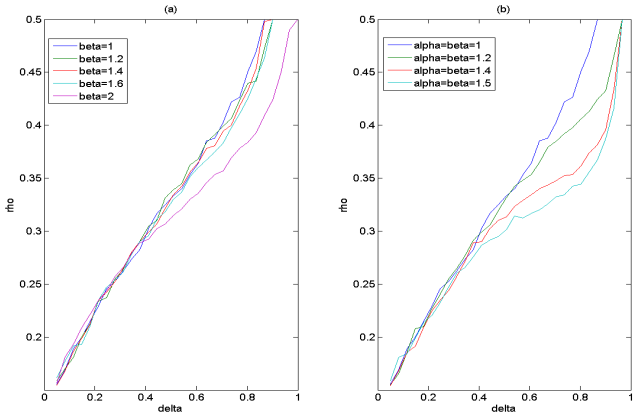

Fig. 5. (a) Empirical phase transitions of TST-(α, β) for $\alpha = 1$ and different values of β; (b) Empirical phase transitions when $\alpha = \beta$.

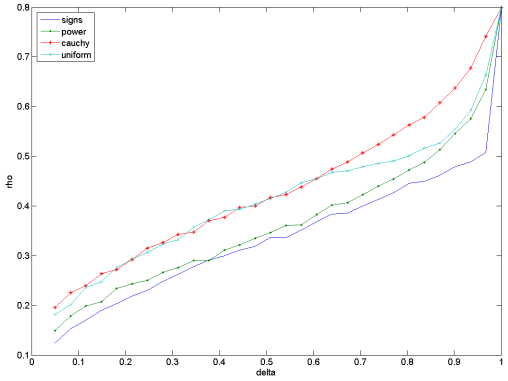

Fig. 8. Observed Phase Transition of Recommended TST for different coefficient ensembles..

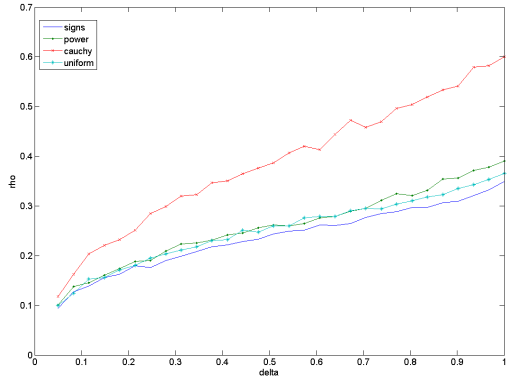

Fig. 6. Observed Phase Transitions of Recommended IST at different coefficient ensembles.

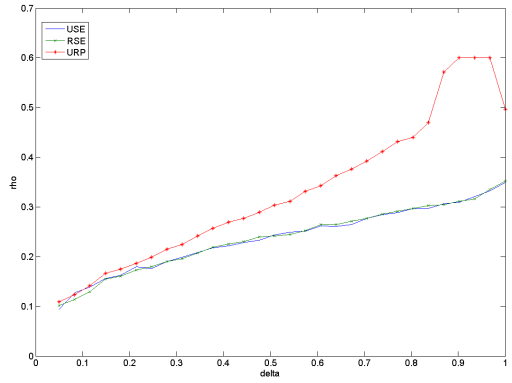

Fig. 9. Observed Phase Transition of Recommended IST for different matrix ensembles.

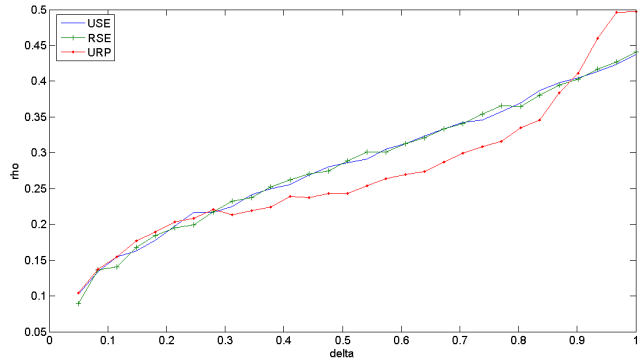

Fig. 10. Observed Phase Transition of Recommended IHT for different matrix ensembles.

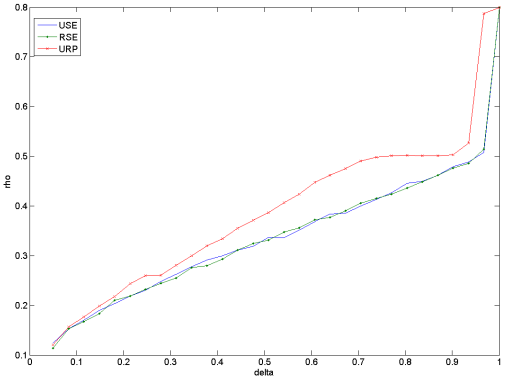

Fig. 11. Observed Phase Transition of Recommended TST for different matrix ensembles.

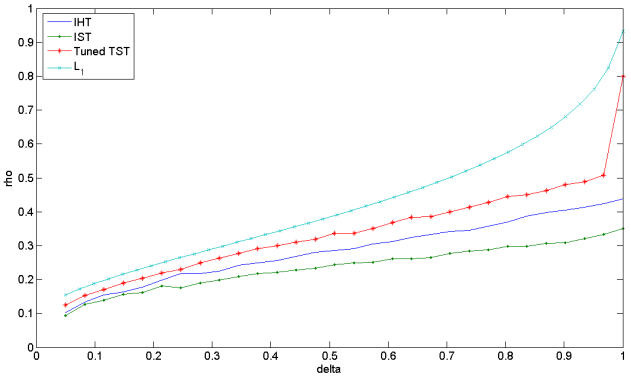

Fig. 12. Observed Phase Transitions of several algorithms at the standard suite.

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

[3] J. Bobin, J.-L. Starck, and R. Ottensamer,"Compressed Sensing in Astronomy" , *IEEE Journal of Selected Topics in Signal Processing* , Vol. 2, no. 5, pp. 718–726, 2008.

[4] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky, "A Wide-Angle View at Iterated Shrinkage Algorithms", SPIE (Wavelet XII) 2007, San-Diego CA, August 26-29, 2007.

[5] I. Daubechies, M. Defrise and C. De Mol "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, Vol. 75, pp. 1412-1457, 2004.

[6] T. Blumensath and M. Davies, "Iterative thresholding for sparse approximations," to appear in *Journal of Fourier Analysis and Applications, special issue on sparsity*, 2008.

[7] D. Needel, J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," Accepted to *Appl. Comp. Harmonic Anal*, 2008.

[8] W. Dai, O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction", submitted to *IEEE Transactions on Information Theory*.

[9] D. L. Donoho, J. Tanner, "Neighborliness of Randomly-Projected Simplices in High Dimensions," *Proceedings of the National Academy of Sciences*, Vol. 102(27),pp. 9452-9457, 2005.

[10] D. L. Donoho, I. Drori, Y. Tsaig, J. L. Starck, "Sparse Solution of Underdetermined Linear Equations by Stagewise Orthogonal Matching Pursuit", Submitted for publication.

[11] D. L. Donoho, Y. Tsaig, " Fast Solution of '1-norm Minimization Problems When the Solution May be Sparse", *IEEE Transactions on Information Theory*, November 2008.