Randomized Iterative Hard Thresholding: A Fast Approximate MMSE Estimator for Sparse Approximations

Robert Crandall, Bin Dong, Ali Bilgin

Abstract—**Typical greedy algorithms for sparse reconstruction problems, such as orthogonal matching pursuit and iterative thresholding, seek strictly sparse solutions. Recent work in the literature suggests that given a priori knowledge of the distribution of the sparse signal coefficients, better results can be obtained by a weighted averaging of several sparse solutions. Such a combination of solutions, while not strictly sparse, approximates an MMSE estimator and can outperform strictly sparse solvers in terms of mean l^2 reconstruction error. Existing algorithms show promising results in improving performance based on approximate MMSE estimation, but can be prohibitively expensive for large-scale problems. We introduce a novel method for obtaining such an approximate MMSE estimator by replacing the deterministic thresholding operator of Iterative Hard Thresholding with a randomized version. This algorithm achieves the performance of the recently introduced RandOMP with much greater computational efficiency, suitable for application to large-scale problems.**

I. Introduction

TYPICAL greedy algorithms for sparse reconstruction problems seek to find solutions which are strictly sparse in some predefined dictionary. Recent work suggests that, given prior knowledge of the distribution of sparse signal coefficients, better results can be obtained by non-sparse estimates formed by a weighted average of several sparse candidate solutions. Such a combination of solutions is motivated by the Bayesian minimum mean-squared error (MMSE) estimator, which is formed using a weighted average of all possible sparse solutions. Since the number of possible sparse supports grows exponentially with signal length, computation of the MMSE estimator is combinatorially hard, and approximations must be used in practice.

Recent algorithms for approximating the sparse MMSE estimator include Randomized Orthogonal Matching Pursuit (RandOMP, [1]) and Fast Bayesian Matching Pursuit (FBMP, [2]). In this paper we introduce an efficient method for generating candidate sparse solutions using a novel randomized hard thresholding operation. By using this randomized thresholding in conjuction with a gradient descent step we develop the Randomized Iterative Hard Thresholding (RIHT) algorithm, which samples from possible sparse candidate solutions with significantly reduced computation times as compared with RandOMP and FBMP, making it suitable for large-scale problems. The randomized thresholding operation can be implemented with negligible increase in computation time, making a single pass of RIHT nearly as efficient as IHT while still delivering lower mean squared error; thus, we advocate the use of a randomized thresholding operation when sufficient prior signal knowledge is available. By running RIHT multiple times and combining the results, in an algorithm we call Aggregated Random Iterative Hard Thresholding (ARIHT), we obtain an approximate MMSE estimate and additional performance gains.

A. Outline of Paper

In Section II we discuss the regularization of linear inverse problems by assuming signal sparsity. In Section III we review iterative hard thresholding (IHT), a fast algorithm for recovering sparse signals. In Section IV we introduce probabilistic signal models that assume further prior knowledge beyond sparsity. In Section V we overview the recently introduced randomized orthogonal matching pursuit (RandOMP) algorithm for approximating an MMSE estimator to the models from Section IV; RandOMP gives improved performance over greedy methods such as OMP and IHT. This motivates our development of the new randomized iterative hard thresholding (RIHT) algorithm in Section VI, which randomizes IHT to give an efficient way of generating sparse candidate solutions. In the same section we also develop the aggregated randomized iterative hard thresholding (ARIHT) algorithm, which combines solutions generated by RIHT to approximate the MMSE estimate for probabilistic sparse signal models. Using ARIHT we achieve the performance of RandOMP with greater computational efficiency. In Section VII we demonstrate the performance of our algorithm in experiments and compare with existing algorithms. Discussion and conclusions are given in Section VIII.

II. Regularized Inverse Problems

A. Linear Inverse Problems

Consider the noisy linear inverse problem of recovering a signal x from a measurement given by an affine transformation of x:

$$y = Ax + e. \tag{1}$$

R. Crandall is with the Program in Applied Mathematics, University of Arizona, Tucson, AZ, 85719 USA e-mail: rcrandall@math.arizona.edu

B. Dong is with the Department of Mathematics, University of Arizona, Tucson, AZ, 85719 USA email: dongbin@math.arizona.edu

A. Bilgin is with the Department of Biomedical Engineering and Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, 85719 USA email: bilgin@email.arizona.edu

y is the measurement or data we observe, and x is the signal we wish to recover. A is a linear operator representing the measurement process through which we observe x, and e is a noise or error vector which corrupts our observation. We assume that $x \in \mathbb{R}^n$, and the measurement y lies in $\mathbb{R}^m$. The linear operator $A : \mathbb{R}^n \to \mathbb{R}^m$ can be represented by an $m \times n$ matrix $A \in \mathbb{R}^{m \times n}$, and the noise vector e lies in $\mathbb{R}^m$. We will assume throughout this paper that the matrix A has full rank, and that $m \leq n$; this covers many common signal processing applications including denoising, image deblurring, and compressed sensing.

Even in the absence of noise, there are infinitely many solutions to $y = Ax$, lying in an affine space of the same dimension as $null(A)$. In recovering x from (1) we must deal with both the additive noise and the inherent ambiguity of the underdetermined linear system. Both of these concerns are dealt with by adopting a priori assumptions about the structure of the signal x. A particularly useful assumption is that of *sparsity*, or the assumption that most of the coefficients of x are zero or nearly zero; this assumption makes the recovery of x feasible. Additional assumptions on the distribution of the coefficients of x can lead to more accurate reconstructions.

B. Regularization

A common approach to selecting from the space of solutions is to formulate the search for x as an optimization problem:

$$\hat{x} = \arg\min_x C(x)$$

where $C : \mathbb{R}^n \to \mathbb{R}$ is some appropriately chosen cost function that quantifies the quality of a particular solution in some way.

If $m < n$, then since A has full rank by assumption, there are infinitely many x solving $y = Ax$; the set of solutions is the affine space $x_0 + \text{null}(A)$ where x_0 is any particular solution. One approach to choosing among this space of solutions is to *regularize* the problem by imposing an additional constraint on the norm of the recovered solution. For instance, we can attempt to balance a data fidelity constraint with a weighted p-norm of the recovered solution; if $\mu \in \mathbb{R}^n_+$ is a vector of non-negative weights and $p \geq 1$, we can seek

$$\arg\min_x \Phi_{p,\mu}(x) = \|y - Ax\|_2^2 + \sum_{i=1}^n \mu_i |x_i|^p. \qquad (2)$$

Iterative thresholding methods combine a gradient descent step with a thresholding step to efficiently address this problem. See e.g. [3] for a comprehensive discussion of iterative thresholding methods for 2 for arbitrary $p \geq 1$.

C. Sparse Approximations and l^0 Regularization

We define a *sparse* vector to be one whose entries are mostly zero; to quantify this we use the operator $\|\cdot\|_0$ which counts the number of nonzero elements in a vector:

$$\|x\|_0 = \#\{i \in \{1, \ldots, n\} \text{ s.t. } x_i \neq 0\}.$$

The problem of recovering the sparsest vector satisfying some constraint is often called l^0-minimization [4].

The assumption of signal sparsity is useful in many problems. For example, natural images are often well approximated by a sparse vector in a particular basis. Common image compression algorithms such as JPEG [5] assume that an image x can be well approximated by $\Psi\alpha$, where α is sparse and Ψ is a linear transform. By "well approximated" we mean that $\|x - \Psi\alpha\|$ is small in some norm. The ubiquitousness of the JPEG standard is a testament to the efficacy of sparse representation methods for most natural images; the visually important information in images of interest to humans tends to be concentrated on a low dimensional subspace.

If we expect that signals of interest are sparse, then we can formulate an l^0-optimization problem such as

$$\text{minimize } \|y - Ax\|_2^2 + \lambda\|x\|_0, \qquad (3)$$

or the constrained version

$$\text{minimize } \|y - Ax\|_2^2 \text{ s.t. } \|x\|_0 \leq k. \qquad (4)$$

These problems are non-convex, and thus much more difficult to solve than (2). The non-convexity means that in general we cannot hope to find a global minimizer without resorting to a combinatorial search over all possible supports [4]. For an in-depth discussion of the local and global minimizers of (3) and the relationship between (3) and (4), the interested reader is referred to [6] and [7]. Note that the choice of k in (4) (or of λ in (3)) will depend on additional knowledge or assumptions about the sparsity level of the signals of interest; however, we will always assume here that $k < m$.

III. Iterative Hard Thresholding

The class of iterative soft thresholding (IST) methods can be used to solve problems of type (2), as described in [3]. These methods rely on continuous soft-thresholding operations to solve the convex problem (2). To solve the non-convex problem (4), we will focus on one method called *iterative hard thresholding*, or IHT ([8], [9], [10]), which is analogous to IST but with a discontinuous thresholding operation H_k. H_k is the constrained hard thresholding operator which zeros out all elements of its argument except the k with largest magnitude (contrast this with the unconstrained hard thresholding operator $H_{\lambda^{0.5}}$ which sets all elements whose magnitude is less than $\lambda^{0.5}$ to zero; this operator is used when solving (3)). If there is no unique set of k largest elements then we can choose randomly or in some prespecified order.

Algorithm 1 Iterative Hard Thresholding

```
Given $x^0$, iterate
    $$x^{\nu+1} = H_k(x^\nu + \mu_\nu A^*(y - Ax^\nu)) \qquad (5)$$
until either $\nu > N_{max}$ or $\|y - Ax^\nu\|_2 < \epsilon$.
```

Thus, the iteration consists of a step in the direction of the negative gradient of the discrepancy term $\|y - Ax\|_2^2$ in (4), then a greedy projection onto a sparse support to satisfy the constraint $\|x\|_0 \leq k$. The stepsize μ_ν can be selected adaptively to guarantee convergence under quite general conditions; see [11] for a description of stepsize selection for IHT.

If in addition A satisfies a restricted isometry property, we are guaranteed convergence to within a constant times the norm of the error $\|e\|_2$ of the best k-sparse approximation to the true signal x (see [9] for details). These guarantees, along with the simplicity of implementation and speed, make IHT and related algorithms quite attractive for large-scale sparse signal processing problems.

IV. Probabilistic Signal Models

The more information about the signal is available to us, the more accurately we should be able to reconstruct x. In solving an optimization problem such as (4), our only assumption is that x is k-sparse (or well approximated by a k-sparse vector). If in addition we have some probabilistic model of how these sparse coefficients are distributed, then we can move from the deterministic framework of (4) to statistical estimation techniques that take advantage of our prior knowledge. In particular, we will seek an estimator which is optimal in terms of the l^2 reconstruction error, which turns out to be a weighted combination of locally optimal solutions. Our goal is to exploit this additional prior knowledge to improve on the results obtainable by IHT.

A. A General Model for Sparse Signals

Suppose that the signal x and the noise e are random variables with known distributions. Let S denote the *support* of the signal x: $S \subset \{1, 2, \dots, n\}$ is the set $\{i : x_i \neq 0\}$ encoding the locations of the nonzero coefficients of x. The set of all possible supports (of which there are 2^n) is denoted Ω. We assume that S is a random variable with known discrete probability density $P(S)$ over Ω. The density of the signal x conditioned on a given support, $p(x|S)$, is also known; thus, we can think of x as being chosen by first drawing a support randomly from Ω with probability $P(S)$, then choosing the nonzero coefficients on S according to $p(x|S)$. The full prior density on x is then found by marginalizing over the possible supports:

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

The noise vector e has known density $p_e(e)$.

B. MAP and MMSE Estimators

Given these three distributions $P(S)$, $p(x|S)$, and $p_e(e)$ which describe our *a priori* knowledge of the system, we can compute two important estimators for x: the *maximum a-posteriori probability* (MAP) estimator, and the *minimum mean-squared error* (MMSE) estimator.

The MAP estimate is found by choosing the most probable support, then choosing the most probable solution on that support:

$$S_{\text{MAP}} = \arg\max_{\hat{S}} P(\hat{S}|y)$$

$$x_{\text{MAP}} = \arg\max_{\hat{x}} p(\hat{x}|S_{MAP}, y)$$

Finding the MAP estimate is related to solving the l^0 problem (3) or (4); that is, we can think of greedy algorithms such as IHT (5) or orthogonal matching pursuit [12] as approximate MAP estimators TODO CITE. To improve on these solutions, we look to the MMSE estimator.

The MMSE estimate minimizes the average l^2 error conditioned on the data y, and is defined as

$$x_{\text{MMSE}} = \arg\min_{\hat{x}} E(\|x - \hat{x}\|_2^2 | y). \tag{6}$$

It is well known that the MMSE estimator is given by the expected value of x conditioned on the measurement y:

$$x_{\text{MMSE}} = E(x|y)$$

or equivalently, for our sparse model,

$$x_{\text{MMSE}} = \sum_{S \in \Omega} E(x|y, S)P(S|y). \tag{7}$$

Note that each of the $E(x|y, S)$ is a constrained MMSE estimate, solving (6) with the constraint that the support is S. It is clear from (7) that the MMSE is a weighted average of the locally optimal solutions $E(x|y, S)$, weighted by the probability that a given support is the correct one. Even though the true solution x is known to be sparse, the MSE-optimal estimator is not.

The computational complexity of both the MAP and MMSE estimators is exponential in signal length, since there are up to 2^n possible supports for x; their computation is NP-hard for sparse systems [4]. This difficulty arises because the posterior $p(x|y) = \sum p(x|y, S)P(S)$ is multimodal even if the $p(x|S)$ are unimodal, so there are many local minimizers. Even so, we can hope to seek practical approximations to these estimates that outperform naive reconstructions (i.e., reconstructions which do not make use of $p(x)$ and $p_e(e)$). Approximations to the MAP estimator are discussed, for example, in [13], while the MMSE estimator for the sparse case has been discussed recently in [14] (for overdetermined systems) and in [1], [2] for the underdetermined case.

C. A Special Case: Gaussian Signal and Noise

For concreteness let us introduce a signal model which we will use in our numerical experiments. As in [1], we choose this model for our analysis because it is mathematically tractable; however, the procedures described in this paper could in principle be repeated for any distributions $P(S)$, $p_e(e)$, and $p(x|S)$, but obtaining simple closed-form expressions for the posterior such as (8) will not be feasible for many models.

Suppose that the support S has known distribution described by $P(S)$. For example if the support length is known to be to k, and all supports of length k are equally likely, then $P(S) = 0$ if $\#S \neq k$ and $P(S) = \frac{k!(n-k)!}{n!}$ if $\#S = k$. Once a support is selected, the nonzero elements of x are then chosen independently from a normal distribution $N(0, \Sigma_x)$, where Σ_x is diagonal with entries $\sigma_{x,i}^2$, and the elements of the noise vector are chosen independently from $N(0, \sigma_e^2)$. The MMSE estimator is then computed as follows. Define

$$Q_S = \frac{1}{\sigma_e^2} A_S^* A_S + \Sigma_x^2$$

where again A_S is the submatrix of A formed by selecting only the columns corresponding to the support S. Define

$$z_S = \frac{1}{\sigma_e^2} Q_S^{-1} A_S^* y,$$

which is the k-vector giving the values of the nonzero coefficients of $E(x|y,S)$. Then if we define I_S to be the $n \times k$ zero-fill matrix formed by choosing the k columns of the identity matrix corresponding to x, we have

$$E(x|y,S) = I_S z_S$$

There are 2^n such solutions, one for each possible support, which are local minimizers of $E(\|x - \hat{x}\|_2 | y)$ constrained to a particular support S.

The posterior support density conditioned on the measurement is

$$P(S|y) \propto \exp\left(\frac{z_S^* Q_S z_S}{2} + \frac{1}{2}\log(\det(Q_S^{-1}))\right) \tag{8}$$

where the constant of proportionality can be calculated using the normalization requirement $\sum_S P(S|y) = 1$. The MMSE estimator is then given by

$$x_{\text{MMSE}} = \sum_{S \in \Omega} P(S|y) I_S z_S,$$

and the MAP estimate is found by maximizing (8) with respect to S then computing $x_{\text{MAP}} = I_{S_{\text{MAP}}} z_{S_{\text{MAP}}}$. Thus, the MAP estimate is one of the local minimizers $E(x|y,S)$, while the MMSE is a weighted combination of all such local minimizers.

D. Approximate MMSE Estimation by Sampling

Equation (7) suggests a Monte Carlo procedure for approximating the MMSE estimator. Suppose that we can draw a random sample of supports $\{S_i\}_{i=1}^N$ from the distribution $P(S|y)$. From our sampled supports $\{S_i\}$ we generate a set of candidate solutions $\{z_{S_i}\}_{i=1}^N$, then combine them by simple averaging:

$$\hat{x} = \frac{1}{N}\sum_{i=1}^{N} I_{S_i} z_{S_i}. \tag{9}$$

Then as the number of samples $N \to \infty$, this estimate approaches the MMSE solution (6).

If most of the energy of the MMSE estimate is concentrated on a few highly probable supports, then we can hope to achieve a good estimate with a number of samples that is much fewer than the total number of possible supports. The difficulty, then, is in developing a procedure for sampling from $P(S|y)$ without performing any combinatorial search over all supports. An approximate Gibbs sampler is used in [1] to build up a support one atom at a time, and a Markov chain monte carlo method is developed in [2]. Our procedure will sample from $P(S|y)$ by randomizing the thresholding operation in IHT. This procedure is very efficient, generating an entire support in a single step and requiring only the sorting of a "key vector" of weights of length n.

V. RandOMP

To motivate our procedure for randomizing the thresholding operation in IHT we review the recently introduced Randomized Orthogonal Matching Pursuit (RandOMP [1]) algorithm. In general, sampling from $P(S|y)$ would require us to compute $P(S|y)$ for every possible support S. In [1] an approximate Gibbs sampler is proposed. Suppose that x is known to have only one non-zero element; that is, $\#S = k = 1$. Then $S = \{i\}$ for some $i \in \{1, ..., n\}$, Q_S becomes the scalar

$$Q_S \equiv q_i = \frac{\|a_i\|_2^2}{\sigma_e^2} + \frac{1}{\sigma_{x,i}^2},$$

and

$$z_S \equiv z_i = \frac{\sigma_{x,i}^2}{\sigma_{x,i}^2 \|a_i\|_2^2 + \sigma_e^2} a_i^* y.$$

Then (8) reduces to

$$P(S = \{i\}|y) \propto \exp\left(\frac{\sigma_{x,i}^2 |a_i^* y|^2}{2\sigma_e^2(\sigma_{x,i}^2 \|a_i\|_2^2 + \sigma_e^2)} - \frac{1}{2}\log(q_i)\right). \tag{10}$$

Now there are only n support probabilities to compute. If we choose a set of N supports at random based on (10), then the sum in (9) will approach the MMSE estimate as N approaches infinity.

So what do we do when $k > 1$? We can build up a support greedily, one atom at a time, by assuming that at each step the next element should be chosen from the density (10) (with the elements already chosen removed and the distribution renormalized at each step). This is the procedure for RandOMP described in [1]. Specifically, we start with an empty support $S^0 = \emptyset$ and the initial estimate $x^0 = 0$. At iteration ν we compute the residual $r^\nu = y - Ax^\nu$. Then we compute an updated set of support probabilities for $\{1, ..., n\} \backslash S^\nu$, conditioned on the ν elements already chosen, as

$$P(i|S^\nu, y) \propto \exp\left(\frac{\sigma_{x,i}^2 |a_i^* r^\nu|^2}{2\sigma_e^2(\sigma_{x,i}^2 \|a_i\|_2^2 + \sigma_e^2)} - \frac{1}{2}\log(q_i)\right). \tag{11}$$

Note the difference between (11) and (10): in (11) we correlate the columns of A with the residual r^ν, rather than the data vector y, to account for the support that has been selected so far. Also, the constant of proportionality is different, since there are $k - \nu$ atoms to choose from rather than k.

So, at each iteration we update the support by choosing i randomly with probability (11) and setting $S^{\nu+1} = S^\nu \cup \{i\}$. The estimate is then updated to $x^{\nu+1} = E(x|y, S^{\nu+1})$. This iteration is repeated until a predetermined support length k is reached, or until the norm of the residual $y - Ax^\nu$ falls below some chosen threshold ϵ. Once the stopping criterion is reached, we store x^ν as one of the samples in (9). This procedure is repeated N times, and the final solution is obtained by simple averaging. When $k = 1$ the result is an MMSE estimator in the limit of $N_{avg} \to \infty$. For $k > 1$ the sampling is inexact but works well in practice [1].

VI. Randomized Iterative Hard Thresholding

The RandOMP procedure introduced in [1] and summarized above gives improved performance over OMP by randomizing the support selection step in order to approximate the MMSE estimator instead of the MAP estimator. Inspired by the success of this algorithm, we propose a similar modification of IHT which achieves comparable performance but with significantly reduced computation time for large problems.

The primary advantage of IHT over OMP is improved computational efficiency. Each algorithm requires correlation of the current residual with the dictionary columns: $A^*(y-Ax^\nu)$. This computation is performed once per iteration in each algorithm. For OMP and RandOMP, the total number of iterations must be on the order of the signal sparsity level k, since one atom is added to the support at each iteration (a method for accelerating this procedure for RandOMP is examined in [15]). IHT, on the other hand, provides an estimate with support length k at each iteration, and the number of iterations depends on how long it takes the gradient descent to converge rather than how long it takes to build up the necessary support length, so IHT scales better with increasing k for large signals.

Furthermore, OMP requires computation of the pseudoinverse of a submatrix of A of the form $A_S^\dagger$ at each step where S is the current support estimate, while IHT requires only application of the transpose operation A^*. This is an advantage when the operation A and its transpose can be applied as a fast transform such as a fast Fourier- or wavelet transform; for very large systems computations of the form $(A_S^* A_S)^{-1}$ become impractical or slow. We will demonstrate in experiments the improved efficiency of our algorithm.

A. *Definition of RIHT*

Our randomized algorithm RIHT replaces the deterministic hard thresholding operation H_k with a *randomized hard thresholding* operator $H_{\tilde{P}}$, which selects a support S with probability $\tilde{P}(S)$ and sets to zero all elements on the complement of S. Note that in general $\tilde{P}(S)$ can be a function of the input x, and that in practice it will typically not be equal to the model distribution $P(S|y)$.

At each step we estimate an adaptive stepsize μ_ν using a procedure similar to the one described in [11]. We first estimate an initial stepsize μ_0 by performing a line search along the gradient direction restricted to the largest k components $H_k(A^*(y-Ax^\nu))$. We then form an intermediate estimate $\tilde{x}$, and apply the randomized hard thresholding operation $H_{\tilde{P}}(\tilde{x})$ to determine the current support $S_{\nu+1}$. We then recompute the stepsize μ_ν using a line search in the direction $H_{S_{\nu+1}}(\tilde{x})$, and compute the final result for the current step by projecting $x^\nu + \mu_\nu A^*(y-Ax^\nu)$ onto the chosen support $S_{\nu+1}$. The procedure is summarized in Algorithm 2.

A solution generated by RIHT is not an MMSE estimate but rather an approximation of $E(x|y,S)$, chosen with probability approximately $P(S|y)$. Thus, to form an MMSE estimate, we combine these *candidate solutions* $\{x_i\}_{i=1}^{N_{avg}}$ by simple averaging to find what we dub the Aggregated Randomized Iterative Hard Thresholding (ARIHT) solution $x = \frac{1}{N_{avg}}\sum_{i=1}^{N_{avg}} x_i$, which is an approximation to the MMSE estimate (7) of the form (9).

Algorithm 2 Randomized Iterative Hard Thresholding (RIHT)

```
Given $y$, $x_0$, and $\tilde{P}$. Set $\rho_0 = \|y\|_2$
Initialize iteration count to $\nu = 0$, result to $x_0$
While $abs(\|\rho_\nu\| - \|\rho_{\nu-1}\|) >$ tol
1. Compute residual $r_\nu = y - Ax_\nu$
2. Compute running average of residual norms
       $$\rho_\nu = \frac{1}{\nu}\sum_{i=1}^{\nu}\|r_\nu\|_2$$
2. Initial stepsize guess based on deterministic hard thresholding:
       $$\mu_0 = \frac{\|H_k(A^* r_\nu)\|_2^2}{\|AH_k(A^* r_\nu)\|_2^2}$$
3. Compute intermediate estimate $\tilde{x} = x_\nu + \mu_0 A^* r_\nu$
4. Compute support $S_{\nu+1}$ by randomized thresholding:
       $$S_{\nu+1} = supp(H_{\tilde{P}}(\tilde{x}))$$
5. Update stepsize using line search on chosen support:
       $$\mu_\nu = \frac{\|H_{S_{\nu+1}}(A^* r_\nu)\|_2^2}{\|AH_{S_{\nu+1}}(A^* r_\nu)\|_2^2}$$
6. Compute
       $$x_{\nu+1} = H_{S_{\nu+1}}(x_\nu + \mu_\nu A^*(y - Ax_\nu)) \qquad (12)$$
   where $H_{S_{\nu+1}}$ denotes projection onto the chosen support $S_{\nu+1}$.
8. Increment iteration count $\nu = \nu + 1$
```

Note that we have not yet specified how the randomized thresholding distribution $\tilde{P}$ is chosen. Later we will give a specific example for the case of the model described in IV-C.

B. *RIHT when $\tilde{P}(S) = P(S|y)$*

It is instructive to examine an idealized version of the RIHT algorithm where we assume we have no computational limitations. Suppose for the moment that we can choose the thresholding operator H such that it selects a support S with probability exactly $\tilde{P}(S) = P(S|y)$. In this idealized scenario the expected value of the sequence generated by the RIHT Algorithm 2 converges to a unique minimizer of an l^2-regularized problem. The expected value of the sequence generated by RIHT can also be thought of as the result of ARIHT in the limit $N_{avg} \to \infty$.

The probability of selecting a given support is independent of the iteration number since $P(S|y)$ depends only on the measurement y. We have

$$E(x_{\nu+1}|x_\nu) = E(H(x_\nu + A^*(y - Ax_\nu))),$$

where the expectation on the right hand side is over the support probabilities $P(S|y)$. It is easy to see that

$$E(x_{\nu+1}|x_\nu) = D(x_\nu + A^*(y - Ax_\nu))$$

where $D \in \mathbb{R}^{n\times n}$ is the diagonal matrix whose entries are $D_{ii} = P(i \in S|y)$. By integrating out the conditional over x_ν, we find

$$E(x_{\nu+1}) = D(E(x_\nu) + A^*(y - AE(x_\nu)));$$

that is, the expectation at step $\nu + 1$ can be computed recursively as a damped Landweber iteration using the expectation at step n, where each coefficient is scaled by its probability of appearing in the true support.

Algorithm 3 Aggregated Random Iterative Hard Thresholding (ARIHT)

```
Given $N_{avg}$, and $\tilde{P}$
for $i = 1$ to $N_{avg}$
1. Compute candidate solution $x_i$ using Algorithm 2
Combine candidate solutions by averaging to obtain final result
    $$x = \frac{1}{N_{avg}} \sum_{i=1}^{N_{avg}} x_i$$
```

Suppose for simplicity that $D_{ii} \neq 0$ (which will be true almost surely for the Gaussian signal model in Section IV-C), and that $\|A\|_2^2 < 2$. Then the sequence $E(x_\nu)$ converges to the unique minimizer of the functional (by theorem 3.1 in [3])

$$\begin{aligned}\Phi_{2,D}(x) &= \|y - Ax\|_2^2 + \sum_{i=1}^{n} \frac{1 - D_{ii}}{D_{ii}} |x_i|^2 \\ &= \|y - Ax\|_2^2 + \sum_{i=1}^{n} \frac{P(i \notin S|y)}{P(i \in S|y)} |x_i|^2 \qquad (13)\end{aligned}$$

given by

$$x^\star = (I - D(I - A^*A))^{-1} DA^*y. \qquad (14)$$

This functional heavily penalizes the squared magnitude of coefficients on improbable supports (as the ratio $\frac{P(i \notin S)}{P(i \in S)} \to \infty$), while coefficients on very probable supports are allowed to converge to whatever value best minimizes the discrepancy term $\|y - Ax\|_2$.

The minimizer to $\Phi_{2,D}$ given by (14) is *not* equivalent to the MMSE estimate in general, even though we assumed (unrealistically) that the true probabilities $P(S|y)$ were available to us. We observe that the minimizer depends only on the probabilities $P(i \in S|y)$ of individual coefficients appearing in the support, and that these probabilities are not sufficient to reconstruct the full support probabilities $P(S|y)$. For example, the $P(i \in S|y)$ do not tell us anything about correlations between different elemenets i, j appearing in a support S.

In the special case where the dictionary A is unitary, then (14) reduces to

$$x^\star = DA^*y.$$

which is a constant multiple of the unitary-dictionary MMSE estimator [16]

$$x^\star = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_e^2} DA^*y.$$

C. Random Thresholding by Weighted Random Sampling

For RIHT to be practical we must avoid combinatorial computations of the type required to compute $P(S|y)$ on every possible support. The difficulty, then, is in selecting a randomized thresholding operation that is both practical to implement on real problems, and useful in the sense that it outperforms IHT for the proposed model.

A particular class of randomized thresholding operators can be implemented by *weighted random sampling*. Given a set of n elements and a vector $w \in \mathbb{R}^n$ of non-negative weights, a weighted random sample (WRS) of length k is chosen as follows.

Algorithm 4 Weighted Random Sample

```
Given a set $\{1, 2, ..., n\}$ and a vector of non-negative weights $w \in \mathbb{R}^N$
Initialize $S = \emptyset$.
While $\#S < k$, iterate
  • Choose $i \in \{1, ..., n\}\backslash S$ with probability $\frac{w_i}{\sum_{j \in \{1,...,n\}\backslash S} w_j}$
  • Add $i$ to $S$: $S = S \cup \{i\}$
```

An efficient algorithm to generate a WRS given the weights w is described in [17]; we first generate a vector $U \in \mathbb{R}^n$ whose entries are drawn independently from the uniform distribution on $[0, 1]$, then compute a vector of "keys" $K_i = U_i^{1/w_i}$. We then sort these keys in descending order of magnitude, and the indices of the largest k keys will constitute a WRS as described in algorithm 4.

To cast RIHT as a weighted random sampling problem, we first examine the special case of a unitary dictionary A, for which a closed-form, non-combinatorial MMSE solution is given in [16].

D. A Special Case: Gaussian Signal, Unitary Dictionary

Consider the signal model given in IV-C, and suppose for now that the dictionary A is unitary. Then (8) reduces to

$$P(S|y) \propto \prod_{i \in S} \exp\left(\frac{1}{2\sigma_e^2} \frac{\sigma_{x,i}^2}{\sigma_{x,i}^2 + \sigma_e^2} |a_i^* y|^2\right) \qquad (15)$$

Achieving these support probabilities by sampling is difficult. A straightforward method would be to select a support S by sampling k elements from $\{1, 2, ..., n\}$ independently, *with replacement*, selecting element i to add to the support on a given step with probability $p_i \propto \exp\left(\frac{1}{2\sigma_e^2} \frac{\sigma_{x,i}^2}{\sigma_{x,i}^2 + \sigma_e^2} |a_i^* y|^2\right)$. If we end up with a support containing k unique indices, we are done; if there are duplicates, we consider the support invalid and repeat the same procedure until a valid support is selected. This method will not be practical, since we are sampling *with replacement* and the probability of selecting the same index twice will typically be very large.

In [16], the authors bypass this difficulty by computing the probabilities $P(i \in S|y)$ in a recursive way that does not require random sampling or combinatorial computation, and use this to derive a closed-form expression for the MMSE estimate when the dictionary is unitary. For our purposes, since we need to actually pick a support for RIHT to make sense, we will use the weighted random sampling technique mentioned above (sampling *without replacement*). In particular, let us select weights

$$w_i \propto \exp\left(\frac{1}{2\sigma_e^2}\frac{\sigma_{x,i}^2}{\sigma_{x,i}^2+\sigma_e^2}|a_i^* y|^2\right) \qquad (16)$$

Generating a WRS from these weights will result in support probabilities that approximate, but do not match exactly, those given in (15) (the probability of a support S being chosen will not be exactly proportional to the product of the weights w_i for $i \in S$). Since the dictionary is unitary, $x + A^*(y - Ax) = A^*y$, and only one iteration is necessary.

E. Selection of Thresholding Operator in Practice

In practice the dictionary A will not be unitary in applications such as compressed sensing where $m < n$, and we need an approximate sampling procedure. RandOMP uses the recursive probabilities (11) which depend on the residual r^ν of the estimate at iteration ν. For RandIHT, we will instead use the magnitudes of the coefficients of the intermediate estimate

$$\tilde{x}_{\nu+1} = x_\nu + \mu_\nu A^*(y - Ax_\nu),$$

and the support of $x_{\nu+1} = H(\tilde{x}_{\nu+1})$ is chosen by using the WRS algorithm above with weights

$$w_i \propto \exp\left(\frac{\sigma_{x,i}^2|\tilde{x}_{i\nu+1}|^2}{2\sigma_e^2(\sigma_{x,i}^2\|a_i\|_2^2+\sigma_e^2)} - \frac{1}{2}\log(q_i)\right). \qquad (17)$$

We are, in effect, crudely assuming that the columns of the matrix A are orthonormal when they are not; this allows us to make our thresholding decision based on the magnitudes of individual coefficients rather than on all the possible quadratic forms $z_S^* Q_S z_S$ that appear in the true support probability (8). In order for any sparse recovery algorithm to work, however, we expect a certain degree of *incoherence* in the dictionary A; this means that our assumption of orthonormality is not "too bad" in that we expect most of the energy of $z_S^* Q_S z_S$ to come from the diagonal entries of Q_S, so that the approximation (17) is close.

F. Stopping Criteria

The stepsize selection procedure of [11] guarantees that for normalized IHT the norm of the residual $y - Ax^\nu$ decreases at each step. No such guarantee is possible for RIHT, since the random thresholding operator has a nonzero probability of picking a sub-optimal support at each step. As such we propose a stopping criterion for the randomized algorithm based on the *running average* of the residual norm. In particular, let $\rho_\nu = \frac{1}{\nu}\sum_{i=1}^{\nu}\|y - Ax_i\|_2$ be the average of the residual norm of RIHT for iterations 1 through ν. Then we terminate the algorithm when $|\rho_\nu - \rho_{\nu-1}| < \epsilon$ for some tolerance ϵ; that is, we stop when the running average of the residual is not changing by much. In Figure 1 the residual norm for a single pass of RIHT is plotted as a function of iteration number, along with the running average of the residual norm; due to the random nature of the algorithm the residual norm does not converge, but the running average does converge empirically and can be used as a stopping criterion.

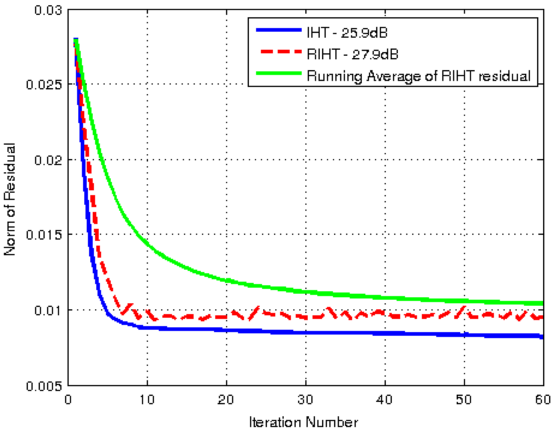

Fig. 1. Norm of residual $y - Ax_\nu$ as a function of iteration number ν for IHT (blue) and RIHT (red). The running average of the RIHT residual, which is used as a stopping criterion, is shown in green.

VII. Experiments

A. Recovery of Synthetic 1D Signals

We begin by examining the performance of RIHT and ARIHT on one-dimensional signals generated according to the model in Section IV-C. The support of the signal x is chosen uniformly at random from the set of supports of length k. The dictionary $A \in \mathbb{R}^{m\times n}$ is generated by drawing its entries from an i.i.d. normal distribution $\mathcal{N}(0,1)$ and then normalizing the columns to have unit l^2 length. The noise vector $e \in \mathbb{R}^m$ is i.i.d. Gaussian with known σ_e^2, and we form the measurement $y = Ax + e$. The nonzero entries of x are drawn from $\mathcal{N}(0,1)$. For ARIHT and RandOMP we set $N_{avg} = 10$.

We compare the proposed algorithms with IHT, OMP, RandOMP, and FBMP. We also include a comparison with a non-greedy method, namely the $l1$-magic toolbox [18]. OMP and IHT are both greedy methods that find local minima of (4), and they perform similarly on these examples. $l1$-magic finds a global minimizer of the convex functional (2) with $p = 1$. FBMP is another Bayesian method that approximates the MMSE estimator; this algorithm outperforms RIHT at low noise levels but not as well for larger σ_e.

An example signal and the approximations obtained by IHT, RIHT, and ARIHT are compared in Figure 2. The IHT and RIHT solutions are sparse, while the ARIHT solution is not; however, the ARIHT solution has the highest PSNR.

In Figure 3 we visualize the set of candidate solutions obtained by RIHT using multi-dimensional scaling (MDS, [19] [20]), a technique for visualizing high dimensional data sets in a way that approximately preserves distance relationships between points. We use a small system ($m = 20$, $n = 30$, $k = 2$) so that the MMSE solution and all of the 435 possible 2-sparse solutions can be computed exactly. The blue circles represent locally optimal solutions $E(x|y,S)$ on different 2-sparse supports. The candidate solutions generated by RIHT are represented as black dots; they fall approximately on the

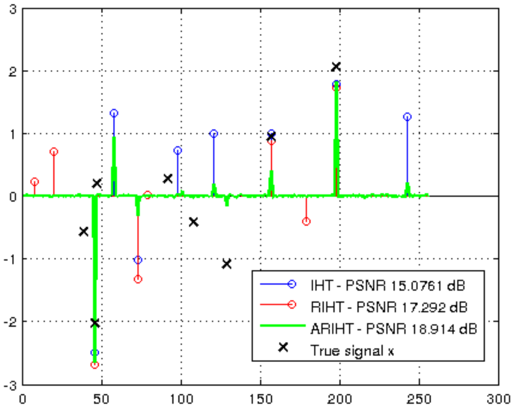

Fig. 2. Sample solutions from IHT, RIHT, and ARIHT

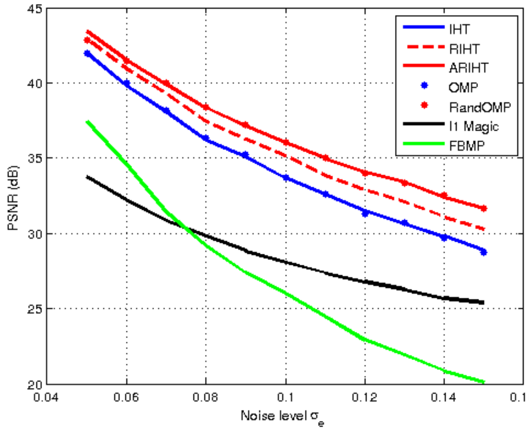

Fig. 4. Mean PSNR as a function of σ_e for synthetic 1D signals, averaged over 500 trials. $\sigma_x = 1$, $m = 128$, $n = 512$, $k = k_{true} = 6$

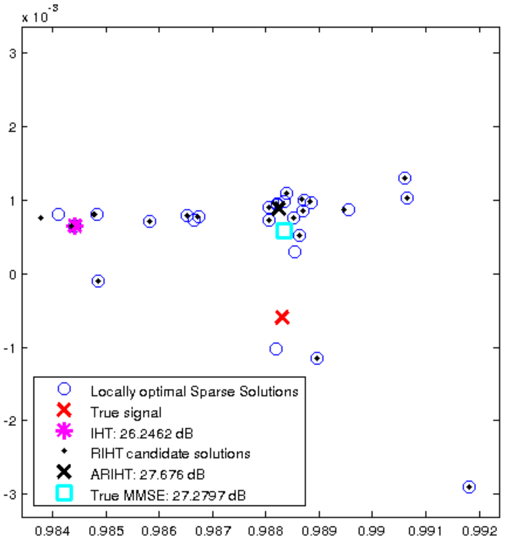

Fig. 3. Visualization of solutions obtained by IHT, RIHT, and ARIHT, for $m = 20$, $n = 30$, $k = 2$. Projected into two dimensions by multi-dimensional scaling (MDS). The true MMSE, as shown by the teal square, is computed by brute force. The ARIHT solution (black X) is obtained by averaging the RIHT solutions (black dots), and gives an approximation to the true MMSE.

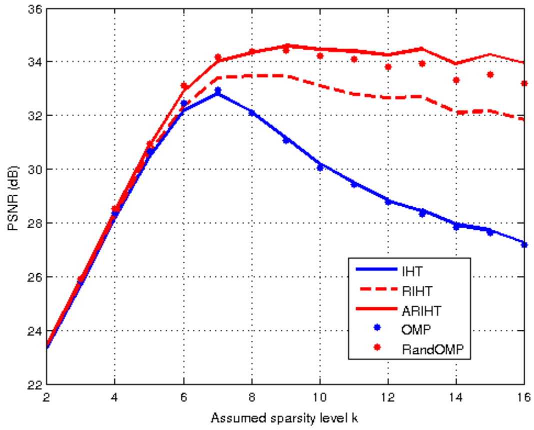

Fig. 5. Mean PSNR as a function of k for synthetic 1D signals, averaged over 500 trials. $\sigma_x = 1$,$\sigma_e = 0.15$, $m = 128$, $n = 256$, $k_{\text{true}} = 8$

locally optimal solutions. The approximate MMSE solution obtained by ARIHT (black X) is closer to the true MMSE (teal square) than the IHT solution.

In the first experiment we fix the sparsity level to $k = 6$ and vary the noise level σ_e. We assume that the sparsity level k is known exactly. The recovered signal PSNR is plotted against σ_e in Figure 4. Note that RIHT outperforms IHT and OMP; this is notable because the computational complexity of RIHT is similar to that of IHT, so we have achieved improved performance at little additional cost.

Next we fix the *true* sparsity level at $k_{\text{true}} = 8$ and the noise level at $\sigma_e = 0.15$, but vary the *assumed* sparsity level used in the recovery algorithms; PSNR is displayed as a function of assumed sparsity in Figure 5. This figure showcases an important feature of the randomized algorithms, namely that they are significantly more robust with respect to choice of assumed sparsity level; this is important since the optimal choice of sparsity is rarely known in practice.

Again it is important to note that, while ARIHT does better than RIHT, RIHT outperforms IHT. This is not the case for the RandOMP algorithm, which performs worse than OMP when the RandOMP solutions are not combined by averaging (when $N_{avg} = 1$). An important consequence is that RIHT can improve performance without adding computational complexity; additional improvement can be obtained by increasing N_{avg} and using ARIHT.

In Figure 7 we plot the reconstruction PSNR from ARIHT as a function of N_{avg} to demonstrate the improvement in performance obtained by averaging the individual candidate solutions generated by RIHT; as N_{avg} grows we form a better approximation of the MMSE estimate, but for this experiment we see diminishing returns beyond $N_{avg} = 10$.

B. Experiments on Computational Complexity

We now examine the computational complexity of RIHT and ARIHT and compare with the comparable existing al-

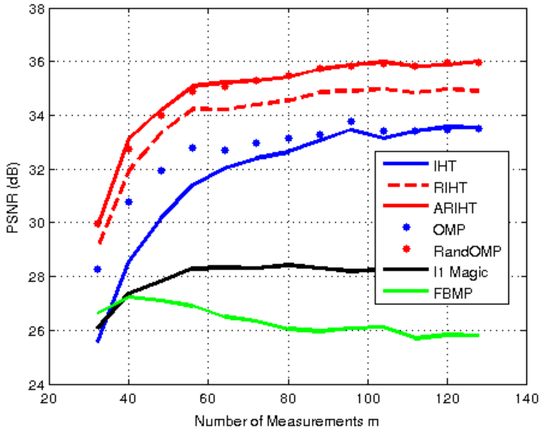

Fig. 6. Mean PSNR as a function of number of measurements for synthetic 1D signals, averaged over 500 trials. $\sigma_x = 1$, $m = 128$, $n = 512$, $k = k_{true} = 6$

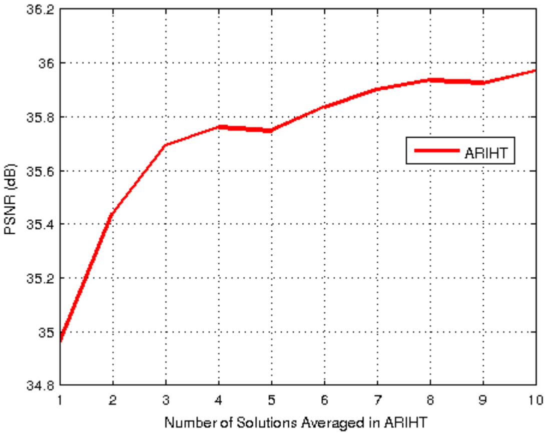

Fig. 7. Mean PSNR for RIHT as a function of N_{avg}

gorithms, OMP and RandOMP. The process of randomizing the hard thresholding operation adds negligible complexity, requiring only generation of the key vector K_i as described in VI-C, and sorting of the keys by magnitude. As such, the computation time required for Algorithm 2 is approximately the same as that required for IHT. The complexity of the aggregated Algorithm 3 scales linearly with N_{avg}. Since each candidate solution can be generated independently, this algorithm is easily parallelizable. The algorithm was not parallelized in these experiments.

Running times as a function of signal length are compared on the left side of Figure 8. The corresponding reconstruction PSNR for each algorithm is shown on the right. For these experiments we set $m = n/2$ and $k = 0.1m$, and increase the signal length n. RIHT run times are comparable to IHT, but RIHT gives better performance; additional performance gain can be obtained using ARIHT. For larger signals, ARIHT is faster than even the deterministic OMP.

VIII. Conclusions

In this work we introduced introduced the RIHT algorithm for solving sparse linear inverse problems. This algorithm modifies the classical IHT by introducing a randomized hard thresholding operator. The resulting Algorithm 2 (RIHT) exploits prior knowledge of the distribution of signal and noise coefficients to obtain lower mean-squared error as compared with IHT, with minimal additional computational complexity. Algorithm 3 (ARIHT) combines the random candidate solutions generated by RIHT to approximate an MMSE estimator and achieve additional performance gains, with a computational cost that increases linearly with the number of candidate solutions used in the average.

Since RIHT and ARIHT require only applications of the operator A and its adjoint, they can be implemented efficiently using fast transforms such as the FFT. Unlike competing approximate MMSE methods such as RandOMP, RIHT does not require inversion of matrices of the form $A_S^* A_S$, which becomes prohibitive for large-scale problems. Furthermore, the randomized thresholding operation immediately returns a full candidate support at each step, so the running time scales better with signal size and sparsity as compared with RandOMP. We have demonstrated that RIHT is practical for much larger-scale problems than RandOMP.

In this work we only investigated the application of a randomized thresholding operation to optimization problems of the form (4), but we expect that improved results could be obtained for other l_0-regularized problems by similarly replacing the deterministic hard thresholding with a randomized version.

IX. Acknowledgments

The authors gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Knowledge Enhanced Compressive Measurements (KECoM) project through contract #N66001-4079, and the University of Arizona Technology Research Initiative Fund (TRIF).

We also thank the anonymous reviewers of an earlier version of this manuscript for their constructive comments and suggestions.

Appendix
Boundedness of the RIHT Sequence

In the body of the paper we have not discussed convergence of the RIHT algorithm in a formal way. We have seen from our experimental results that RIHT converges in the sense of a running average of the residual norm $\|y - Ax\|$, and that the aggregated algorithm ARIHT appears to converge ergodically, in the sense that as $N_{avg} \to \infty$ and the iteration number $\nu \to \infty$ we see convergence. We prove here a practical result that guarantees that the sequence x_ν generated by RIHT is bounded given a condition on the spark of A.

The *spark* of a matrix, written $spark(A)$, is the smallest number of columns of A that can be combined to form a linearly dependent set. We will assume that the dictionary of interest A has $spark(A) > k$, where k is the sparsity of x (such matrices are abundant; they can be generated

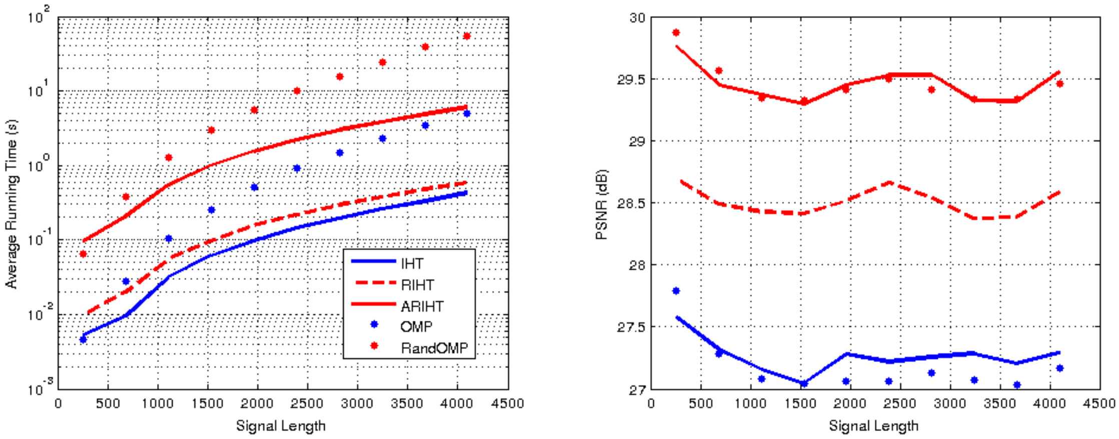

Fig. 8. Left: Comparison of average run time in seconds (on a log scale) as a function of signal length. Right: Corresponding reconstruction PSNR (dB). Number of measurements is set to $m = n/2$, and the sparsity level is set to $k = 0.1m$.

deterministically, and random Gaussian matrices are full-spark almost surely [21]) This property guarantees that any k-column submatrix of A has full rank, which is a necessary and sufficient condition for the local minimizers of (4) on a particular support S to be unique [6]. The non-zero entries of the minimizer restricted to a given support S are given by the k-vector

$$x^\star_S = (A^*_S A_S)^{-1} A^*_S y. \tag{18}$$

where A_S is the submatrix of A formed by taking only those columns corresponding to the support S.

The sequence x_ν generated by RIHT is bounded in $\mathbb{R}^n$, as long as $\|A\|_2^2 < 2$ and $spark(A) > k$ (recall that we make this assumption on the spark throughout this paper).

If we iterate Algorithm 2 for a given choice of $\tilde{P}$, then we have

$$x_{\nu+1} = D_{\nu+1}(I - A^*A)x_\nu + D_{\nu+1}A^*y$$

where $D_{\nu+1}$ is the diagonal matrix with ones on the entries corresponding to $supp(x_{\nu+1})$ and zeros elsewhere. For a given starting point x_0, then, we have

$$x_\nu = \left[\prod_{i=\nu}^{1} D_i(I - A^*A)\right] x_0 + \sum_{i=1}^{\nu} \left[\prod_{j=\nu}^{i+1} D_j(I - A^*A)\right] D_i A^* y \tag{19}$$

Here the matrix product with the larger index on bottom indicates that the matrices are multiplied from left to right starting with the largest index, i.e.

$$\prod_{i=\nu}^{1} D_i(I - A^*A) = D_\nu(I - A^*A) D_{\nu-1}(I - A^*A) ... D_1(I - A^*A).$$

D_ν is a random sequence of diagonal matrices with $\|D_\nu\|_2 = 1$. There are N_S such matrices, one for each allowed support S. The probability distribution of this random sequence depends on the starting point x_0, and the random thresholding probabilities $\tilde{P}$.

Suppose that the dictionary has been scaled to satisfy $\|A\|_2^2 < 2$. Then $\|(I - A^*A)x\|_2 \leq \|x\|_2$, with equality iff $x \in null(A)$. It follows immediately that the first term in (19) is bounded, since $\|D_i(I - A^*A)\|_2 \leq 1$ which implies

$$\|\prod_{i=\nu}^{1} D_i(I - A^*A)x_0\|_2 \leq \|x_0\|_2$$

for all ν.

By assumption, $spark(A) > k$, so there are no nonzero k-sparse vectors in $null(A)$. Then

$$\|(I - A^*A)D_i\|_2 < 1$$

for all i, and in particular, since there are only finitely many different matrices D_i (one for each support) we have

$$\|(I - A^*A)D_i\|_2 < C$$

for some constant $C < 1$ that is independent of i. Then

$$\|\sum_{i=1}^{\nu} \left(\prod_{j=\nu}^{i+1} D_j(I - A^*A)\right) D_i A^* y\|_2 \leq \sum_{i=1}^{\nu} C^{\nu-i}\|A^*y\|_2$$

$$= \frac{1 - C^\nu}{1 - C}\|A^*y\|_2.$$

Thus, the sequence x_ν is bounded with

$$\|x_\nu\|_2 \leq \|x_0\|_2 + \frac{1}{1 - C}\|A^*y\|_2.$$

We conjecture that the sequence $E(x^\nu)$ of expected states converges as $\nu \to \infty$, although we have not proven it. We do see ergodic convergence in practice in our numerical results, and are exploring proofs in ongoing research.

References

[1] Michael Elad and Irad Yavneh, "A plurality of sparse representations is better than the sparsest one alone," *Information Theory, IEEE Transactions on*, vol. 55, no. 10, pp. 4701–4714, 2009.

[2] Philip Schniter, Lee C Potter, and Justin Ziniel, "Fast bayesian matching pursuit," in *Information Theory and Applications Workshop, 2008*. IEEE, 2008, pp. 326–333.

[3] Ingrid Daubechies, Michel Defrise, and Christine De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[4] Balas Kausik Natarajan, “Sparse approximate solutions to linear systems,” *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[5] William B Pennebaker and Joan L Mitchell, *JPEG: Still image data compression standard*, Kluwer Academic Pub, 1992.

[6] M. Nikolova, “Description of the minimizers of least squares regularized with 0 norm. uniqueness of the global minimizer,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 934–937, 2013.

[7] Mila Nikolova, “Relationship between the optimal solutions of least squares regularized with l0-norm and constrained by k-sparsity,” 2014.

[8] Thomas Blumensath and Mike E Davies, “Iterative thresholding for sparse approximations,” *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 629–654, 2008.

[9] Thomas Blumensath and Mike E Davies, “Iterative hard thresholding for compressed sensing,” *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[10] Rahul Garg and Rohit Khandekar, “Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property,” in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 337–344.

[11] Thomas Blumensath and Mike E Davies, “Normalized iterative hard thresholding: Guaranteed stability and performance,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 298–309, 2010.

[12] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*. IEEE, 1993, pp. 40–44.

[13] Shihao Ji, Ya Xue, and Lawrence Carin, “Bayesian compressive sensing,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2346–2356, 2008.

[14] Erik G Larsson and Yngve Selén, “Linear regression with a sparse parameter vector,” *Signal Processing, IEEE Transactions on*, vol. 55, no. 2, pp. 451–460, 2007.

[15] Shutao Li and Leyuan Fang, “Signal denoising with random refined orthogonal matching pursuit,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 61, no. 1, pp. 26–34, 2012.

[16] Matan Protter, Irad Yavneh, and Michael Elad, “Closed-form mmse estimation for signal denoising under sparse representation modeling over a unitary dictionary,” *Signal Processing, IEEE Transactions on*, vol. 58, no. 7, pp. 3471–3484, 2010.

[17] Pavlos S Efraimidis and Paul G Spirakis, “Weighted random sampling with a reservoir,” *Information Processing Letters*, vol. 97, no. 5, pp. 181–185, 2006.

[18] Emmanuel Candes and Justin Romberg, “l1-magic: Recovery of sparse signals via convex programming,” *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, vol. 4, 2005.

[19] Joseph B Kruskal, “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis,” *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[20] Joseph B Kruskal and Myron Wish, *Multidimensional scaling*, vol. 11, Sage, 1978.

[21] Boris Alexeev, Jameson Cahill, and Dustin G Mixon, “Full spark frames,” *Journal of Fourier Analysis and Applications*, vol. 18, no. 6, pp. 1167–1194, 2012.