On Accelerated Hard Thresholding Methods for Sparse Approximation

Volkan Cevher

Swiss Institute of Technology, Lausanne, Switzerland.

ABSTRACT

We propose and analyze acceleration schemes for hard thresholding methods with applications to sparse approximation in linear inverse systems. Our acceleration schemes fuse combinatorial, sparse projection algorithms with convex optimization algebra to provide computationally efficient and robust sparse recovery methods. We compare and contrast the (dis)advantages of the proposed schemes with the state-of-the-art, not only within hard thresholding methods, but also within convex sparse recovery algorithms.

Keywords: sparse recovery, hard thresholding methods, algebraic pursuits, model based compressive sensing.

1. INTRODUCTION

Given a regression matrix $A \in \mathbb{R}^{M \times N}$ ($M < N$), a vector $x^* \in \Sigma_K^N$, suppose we observe $u \in \mathbb{R}^M$ via

$$u = Ax^* + n, \tag{1}$$

where n is an additive noise, and $\Sigma_K^N \subset \mathbb{R}^N$ denotes a union-of-subspaces model with at most K-nonzero entries in N-dimensions ($K \ll N$).[1] To determine x^* from u, we propose to solve the following minimization problem:

$$\min_{x : x \in \Sigma_K^N} f(x), \quad f(x) = \|u - Ax\|^2. \tag{2}$$

The combinatorial problem, as defined by (2), is an instance of sparse approximation—a topic of great interest in *underdetermined linear regression* (i.e., $M < N$), where sparsity is the *de facto* regularization standard to obtain "good" solutions; examples include learning sparse subsets of features in classification,[2] learning sparse graphical models in statistical inference,[3] and compressive sensing.[4]

In this paper, we focus on the class of hard thresholding methods for sparse approximation; c.f.,[5,6] for a review of existing methods and their applications. Typically, these methods iteratively refine a putative solution with a correction term, followed by a combinatorial projection to satisfy the sparsity constraint. For instance, the iterative hard thresholding (IHT) algorithm with step size μ has the following recursion:

$$x_{i+1} = H_K \left(x_i + \mu A^t(u - Ax_i) \right), \tag{3}$$

where i is the iteration number, and H_K is the combinatorial projection onto Σ_K^N:

$$H_K(y) = \mathrm{argmin}_{x : x \in \Sigma_K^N} \|x - y\|, \tag{4}$$

whose action amounts to hard thresholding.

While the solution of (2) is NP-Hard in general, we can establish the correctness of the hard thresholding methods when A satisfies the so-called restricted isometry property (RIP). When Σ_K^N is modulo isomorphic (i.e., if $x_i \in \Sigma_{K_i}^N$ ($i = 1, 2$), then $(x_1 + x_2) \in \Sigma_{K_1+K_2}^N$), the RIP implies that the linear system is bi-Lipschitz:

$$(1 - \delta_K)\|x\|^2 \leq \|Ax\|^2 \leq (1 + \delta_K)\|x\|^2, \forall x \in \Sigma_K^N, \tag{5}$$

where δ_K is the minimum among the isometry constants of A on the set Σ_K^N. Assuming the RIP, the recursion of several hard thresholding methods satisfies $\|x^* - x_i\| \leq \rho^i \|x^* - x_0\| + C\|n\|$, where $x^* \in \Sigma_K^N$ and n are related to u as in (1), C is a constant, and $|\rho| < 1$ depends on δ_{cK}, where $c = 2, 3, 4$.[*]

Per iteration complexity of the hard thresholding methods are dominated by two main factors: the combinatorial projection onto Σ_K^N, and the application of A (and its adjoint A^t). Depending on the problem (e.g., N or the definition of the set Σ_K^N), these operations can have different relative costs; hence, hard thresholding methods with low iteration counts and the flexibility to trade-off these operations are desired.

To obtain the desidera, several well-known ideas from convex optimization are applied to create different variants of hard thresholding methods:[5] analyze the IHT algorithm in the context of the gradient descent method and propose to use $\mu = 1/(1 + \delta_{2K})$ as the step size.[7] proposes an involved line-search method to adaptively select the step size per iteration.,[8],[7] and[9] propose multi-stage approaches, which also minimize $f(x')$–exactly or approximately–restricted to the non-zero coefficients of the putative solution.

A major alternative to the hard thresholding methods for sparse approximation is based on convex optimization with sparsity inducing, convex norms.[6, 10] Once the sparse approximation problem is *convexified*, decades of experience in convex optimization methods can be leveraged. In the high-dimensional scaling of (2), first-order methods, such as accelerated Nesterov, augmented Lagrangian, and operator splitting, are the *modus operandi* of convexified sparse approximation. Unsurprisingly, we can also establish the correctness of these methods by assuming RIP.[6] Albeit lacking convergence guarantees, another promising alternative to hard thresholding methods is called the approximate message passing (AMP).[11]

Contributions: We propose and analyze three acceleration schemes, broadly applicable to the class of hard thresholding methods for sparse approximation. The first scheme is a computationally efficient, one shot step size selection procedure that exploits the structure of the sparse approximation problem. Inspired by Nesterov's accelerated first-order methods, the second scheme incorporates a momentum term based on the previous iterate of hard thresholding methods. Inspired by the AMP algorithm, the third scheme incorporates a weighted sum of thresholded gradients for acceleration. We compare and contrast the (dis)advantages of the proposed schemes with the state-of-the-art, not only within hard thresholding methods, but also within the convex approaches. We also provide a loose RIP analysis of the proposed algorithms for completeness.

2. PRELIMINARIES

Notation: We assume Σ_K^N is modulo isomorphic (or has the nested approximation property[12]) along with the RIP, as in (5).

We use the ℓ_2-norm $\|\cdot\|$ throughout, unless otherwise stated. The bracket notation $\langle x, y \rangle = x^t y$ refers to the inner product, where ^{t} is the transpose operation. By objective function, we specifically mean the ℓ_2-observation error: $f(x) = \|u - Ax\|^2$, where $\|x\| = \left(\sum_{i=1}^N |[x]_i|\right)^{1/2}$, and $[x]_i$ refers to the i-th element of the vector x. We use $\nabla f(x) = -2A^t(u - Ax)$ to denote the gradient of the objective $f(x)$.

The support supp(x) of a vector x is defined as the index set of its non-zero coefficients. The set difference operator is denoted as $\backslash$. Given an index set $\mathcal{S} \subseteq \mathcal{I} = \{1, 2, \ldots, N\}$, the notation $\nabla_\mathcal{S} f(x)$ means that $[\nabla_\mathcal{S} f(x)]_i = [\nabla f(x)]_i$, whenever $i \in \mathcal{S}$, and $[\nabla_\mathcal{S} f(x)]_i = 0$, otherwise.

Structure of the objective function: We highlight two key properties for the objective function, which are used in establishing method guarantees.

PROPERTY 1 (QUADRATIC SURROGATES). *Define the Bregman distance based on the objective function:*

---

[*]A great deal of research therefore revolves around bounding the isometry constant for convergence and estimation guarantees. While the isometry constant is typically unknown *a priori*, a larger M leads to a better (or smaller) δ, as a rule of thumb. For instance, for random matrices with sub-Gaussian entries, $M = \mathcal{O}\left(\log |\Sigma_K^N|\right)$ is sufficient to provide a desired level of isometry, where $\left|\Sigma_K^N\right|$ is the cardinality of Σ_K^N.[1]

$B(y, x) = f(y) - f(x) - \langle \nabla f(x), y - x \rangle$. Then, $B(y, x)$ satisfies the following ($j = 1, 2$):

$$
\begin{array}{llll}
(1) & B(x_2, x_1) & = & \|A(x_2 - x_1)\|^2, \qquad \forall x_j \in \mathbb{R}^N; \\
(2) & B(x_2, x_1) & \leq & (1 + \delta_{K'})\|x_2 - x_1\|^2, \quad \forall x_j \in \Sigma_{K_j}; \\
(3) & B(x_2, x_1) & \geq & (1 - \delta_{K'})\|x_2 - x_1\|^2, \quad \forall x_j \in \Sigma_{K_j};
\end{array}
\tag{6}
$$

where $K' = K_1 + K_2$. These expressions follow from simple linear algebra and the RIP assumption in (5).

PROPERTY 2 (HARD THRESHOLDING DISTANCE). Let $b \in \Sigma_B^N$, where $B > K$, and $\bar{b} = H_K(b)$. Then, given $x^* \in \Sigma_K^N$, the following inequalities hold ($j = 1, 2$):

$$
\|x^* - \bar{b}\| \leq 2\|x^* - b\| \tag{7}
$$

$$
\leq \frac{2}{\sqrt{1 - \delta_{K'}}} \left(\|u - Ab\| + \|n\| \right), \tag{8}
$$

where $K' = K + B$. Defining $\kappa = 2\sqrt{\frac{1 + \delta_{2K}}{1 - \delta_{K'}}}$, whenever $f(\bar{b}) \geq \|n\|^2$, we also have

$$
\|u - A\bar{b}\| \leq \kappa \|u - Ab\| + (1 + \kappa)\|n\|. \tag{9}
$$

A proof of this property is in the Appendix.

Distance mapping: For many hard thresholding methods, it is easier to track the evolution of the objective values than to track the distance to x^*. The following lemma shows that a small objective value implies proximity to x^*, which is proved in the Appendix.

LEMMA 2.1 (DISTANCE MAPPING). Let $\|u - Aa\| \leq c\|n\|$ for some $c > 0$. If $a \in \Sigma_K^N$, then

$$
\|x^* - a\| \leq \frac{c + 1}{\sqrt{1 - \delta_{2K}}} \|n\|. \tag{10}
$$

3. ACCELERATION VIA STEP SIZE SELECTION

Motivation: Step size selection is a natural way of improving the convergence speed of hard thresholding methods. Existing approaches broadly fall into two categories: constant and adaptive step size selection.

Among the constant step sizes, $\mu^* = 1/(1 + \delta_{2K})$ of GraDes[5] is theoretically optimal. To see this, it is instructive to view the IHT algorithm (3) in the context of proximal algorithms, where the quadratic surrogate in Property 1(2) is used as a majorizing function to $f(x)$ around x_i to obtain

$$
\mathrm{argmin}_{x \in \Sigma_K^N} f(x_i) + \langle \nabla f(x_i), x - x_i \rangle + (1 + \delta_{2K})\|x - x_i\|^2 = H_K \left(x_i + 1/(1 + \delta_{2K})A^t(u - Ax_i) \right).
$$

As δ_{2K} is the minimum over all the isometry constants of A on Σ_{2K}^N, any μ larger than μ^* can violate the RIP assumption during method execution; this potentially leads to instability. Unfortunately, unless A has a special structure (e.g., randomized), calculation of μ^* is a hefty task.[13]

There is limited work on the adaptive step size selection for hard thresholding methods. To the best of our knowledge,[7,14] are the only studies that attempt line searching in this context. The main disadvantage of these approaches is computational: they require several combinatorial projections and function evaluations to calculate an iteration dependent step size μ_i while guaranteeing sufficient descent and stability.

In contrast, our acceleration scheme is based on a one-shot step size selection procedure, and empirically outperforms the approaches above, as demonstrated in Section 6. Our approach relies on a key observation:

REMARK 1. *Suppose an oracle provides us the largest μ_i^* at iteration i, which does not violate a relaxed RIP assumption, given that $\mathcal{X}_i = \mathrm{supp}(x_i)$ is fixed. Based on this knowledge, we obtain $x_{*,i+1} = H_K \left(x_i + \mu_i^* A^t(u - Ax_i) \right)$. It then holds that $\mathrm{supp}(x_{*,i+1})$ is necessarily included in the index set $\mathcal{S}_i$ with cardinality $|\mathcal{X}_i| + K$, where*

$$
\mathcal{S}_i = \mathcal{X}_i \cup \mathrm{supp} \left(H_K \left(\nabla_{\mathcal{I} \setminus \mathcal{X}_i} f(x_i) \right) \right). \tag{11}
$$

The proof is straightforward as $\mathcal{S}_i$ contains $\mathrm{supp}(x_{i+1})$ for any μ, and is left to the reader. While $\mathrm{supp}(x_{*,i+1})$ is unknown, we obtain the smallest set $\mathcal{S}_i$ that contains it at the cost of one combinatorial projection.

Algorithm 1 Template for memoryless IHT methods

    Input: u, A, x_0, ϵ, and MaxIterations;
    repeat
        Determine $\mathcal{S}_i$ via (11).
        if SolveNewtonb=1 **then**
            Solve $b = \operatorname{argmin}_{v:\operatorname{supp}(v)=\mathcal{S}_i} \|\nabla_{\mathcal{S}_i} f(v)\|$.
        else
            Calculate b via (12) and (13).
        end if
        Set $x_{i+1} = H_K(b)$ and $\mathcal{X}_{i+1} = \operatorname{supp}(x_{i+1})$.
        if GradientDescentx=1 **then**
            Calculate $\nabla_{\mathcal{X}_{i+1}} f(x_{i+1})$; Set $[x_{i+1}]_{\mathcal{X}_{i+1}} =$
            $[x_{i+1}]_{\mathcal{X}_{i+1}} + \frac{\|\nabla_{\mathcal{X}_{i+1}} f(x_{i+1})\|^2 \nabla_{\mathcal{X}_{i+1}} f(x_{i+1})}{2\|A\nabla_{\mathcal{X}_{i+1}} f(x_{i+1})\|^2}$
        else if SolveNewtonx=1 **then**
            $x_{i+1} = \operatorname{argmin}_{v:\operatorname{supp}(v)=\mathcal{X}_{i+1}} \|\nabla_{\mathcal{X}_{i+1}} f(v)\|$
        end if
    until $\|x_i - x_{i+1}\| \leq \epsilon \|x_{i+1}\|$ or MaxIterations.

Main idea: We propose to calculate a step-size $\bar{\mu}_i$ that first takes x_i to a proxy-vector $b \in \Sigma_{2K}^N$, whose support is restricted to $\mathcal{S}_i$, that best minimizes $f(x)$ via

$$b = x_i - 0.5\bar{\mu}_i \nabla_{\mathcal{S}_i} f(x_i), \text{ where} \tag{12}$$

$$\bar{\mu}_i = \operatorname{argmin}_\mu f(x_i - 0.5\mu \nabla_{\mathcal{S}_i} f(x_i)) = \frac{\|\nabla_{\mathcal{S}_i} f(x_i)\|^2}{\|A\nabla_{\mathcal{S}_i} f(x_i)\|^2}. \tag{13}$$

Note that $1 - \delta_{2K} \leq \bar{\mu}_i^{-1} \leq 1 + \delta_{2K}$ due to RIP. Proposition 1, whose proof is in the Appendix, characterizes a variant of the IHT algorithm with this approach:

PROPOSITION 1. *The vector $b \in \Sigma_{2K}^N$ in (12) satisfies*

$$\|u - Ab\| \leq \sqrt{2\delta_{2K}}\|x^* - x_i\| + \|n\|. \tag{14}$$

Moreover, if we use $x_{i+1} = H_K(b)$, then

$$\|x^* - x_{i+1}\| \leq \rho\|x^* - x_i\| + \frac{4}{\sqrt{1 - \delta_{3K}}}\|n\|, \tag{15}$$

where $\rho = 2\sqrt{\frac{2\delta_{2K}}{1 - \delta_{3K}}}$. If $\rho < 1$, then we have

$$\|x^* - x_i\| \leq \rho^i \|x^* - x_0\| + \frac{4(1 - \rho)^{-1}}{\sqrt{1 - \delta_{3K}}}\|n\|. \tag{16}$$

    While the resulting RIP requirement for the algorithm is more stringent than the IHT methods in[7, 9, 14] this variant of IHT empirically outperforms the alternative methods.[†]

A template for memoryless IHT methods: We describe how to incorporate our step size selection scheme into the class of memoryless hard thresholding methods. By memoryless, we mean the class of methods that does not keep track of the previous solutions.

    Algorithm 1 provides a template with three options that trade-off the number of combinatorial projections with the applications of A and A^t. The SolveNewton() options correspond to solving the Newton system

---

[†]MATLAB implementations of all the IHT methods proposed in this paper can be found at http://lions.epfl.ch/ALPS.

restricted to a sparse support, which can be efficiently computed via conjugate gradients. For instance, setting (SolveNewtonb=1) has the same flavor as the subspace pursuit algorithm[15] (but it is not quite the same, since the support selection steps are different), whereas setting (SolveNewtonx=1) is akin to hard thresholding pursuit in.[9] The GradientDescentx switch enables a single gradient update on $\bar{b}$ restricted to its support with line search, which is similar to fast hard thresholding pursuit in.[9]

PROPOSITION 2. *All variants of Algorithm 1 satisfy*

$$\|u - Ax_i\| \leq \rho^i \|u - Ax_0\| + C\|n\|, \tag{17}$$

where $\rho = 2\sqrt{\frac{2\delta_{2K}(1+\delta_{2K})}{(1-\delta_{2K})(1-\delta_{3K})}}$, and C is a constant. A proof of this statement is in the Appendix.

According to Proposition 2, it is possible to reduce the objective to $C(1-\rho)^{-1}\|n\|$ by iterating on the template defined by Algorithm 1. We then invoke Lemma 2.1 to obtain the final estimation guarantee. To obtain tighter RIP guarantees, we can do further analysis on individual algorithm variants, such as.[16]

4. ACCELERATION VIA 1-MEMORY

Motivation: To introduce the new acceleration scheme, consider the following convexified version of (2):

$$\min_{x:x\in\mathbb{R}^N} f'(x), \quad f'(x) = \|u - Ax\|^2 + \lambda\|x\|_1, \tag{18}$$

where we replace the set constraint Σ_K^N by 1-norm regularization ($\|x\|_1 = \sum_{i=1}^N |[x]_i|$). The parameter $\lambda > 0$ is a constant. The classical iterative soft thresholding (IST) algorithm is a popular method to solve (18):

$$x_{i+1} = T_{\lambda\mu/2}\left(x_i + \mu A^t(u - Ax_i)\right), \tag{19}$$

where $[T_\alpha(x)]_i = (|[x]_i| - \alpha)_+\text{sign}([x]_i)$. Theoretically, the IST algorithm has a sublinear convergence rate of $f(x_i) - f(x^*) \approx \mathcal{O}(1/\sqrt{i})$, where x^* is the minimizer of $f'(x)$.[17] However, it is possible to improve this rate to $f(y_i) - f(x^*) \approx \mathcal{O}(1/i)$ by

$$\begin{aligned} y_i &= T_{\lambda\mu/2}\left(x_i + \mu A^t(u - Ax_i)\right), \\ x_{i+1} &= y_i + \frac{a_i - 1}{a_{i+1}}(y_i - y_{i-1}), \\ a_{i+1} &= \left(1 + \sqrt{1 + 4a_i^2}\right)/2; \end{aligned} \tag{20}$$

where $a_1 = 1$. The recursion in (20) is proposed as the fast iterative shrinkage and thresholding algorithm (FISTA) by Beck and Teboulle[17] in the light of Nesterov's work on accelerated gradient methods.[18]

Main idea: Based on a similar momentum term, we propose the following hard thresholding method:

$$y_i = H_K(b), \quad x_{i+1} = y_i + \tau_i(y_i - y_{i-1}); \tag{21}$$

where $\tau_i \in (0, 1]$, and b is calculated using (12). When μ^* is known, we set $y_{i+1} = H_K\left(x_i + \mu^* A^t(u - Ax_i)\right)$. Proposition 3, whose proof is given in the Appendix, characterizes the convergence of the hard thresholding method with 1-memory in (21).

PROPOSITION 3. *Let $c = 2\sqrt{\frac{2\delta_{3K}}{1-\delta_{4K}}} < 1/3$, $\tau_0 = 0$, and $\tau_i \leq 1$. The output y_i of (21) satisfies the following:*

$$\|x^* - y_i\| \leq C_1\rho_+^i + C_2\rho_-^i + \frac{4(1-3c)^{-1}}{\sqrt{1-\delta_{4K}}}\|n\|, \tag{22}$$

where $\rho_\pm = c \pm \sqrt{c^2 + c}$; and, C_1 and C_2 are constants.

We further improve the RIP requirements of the 1-memory methods and provide further enhancements in.[16]
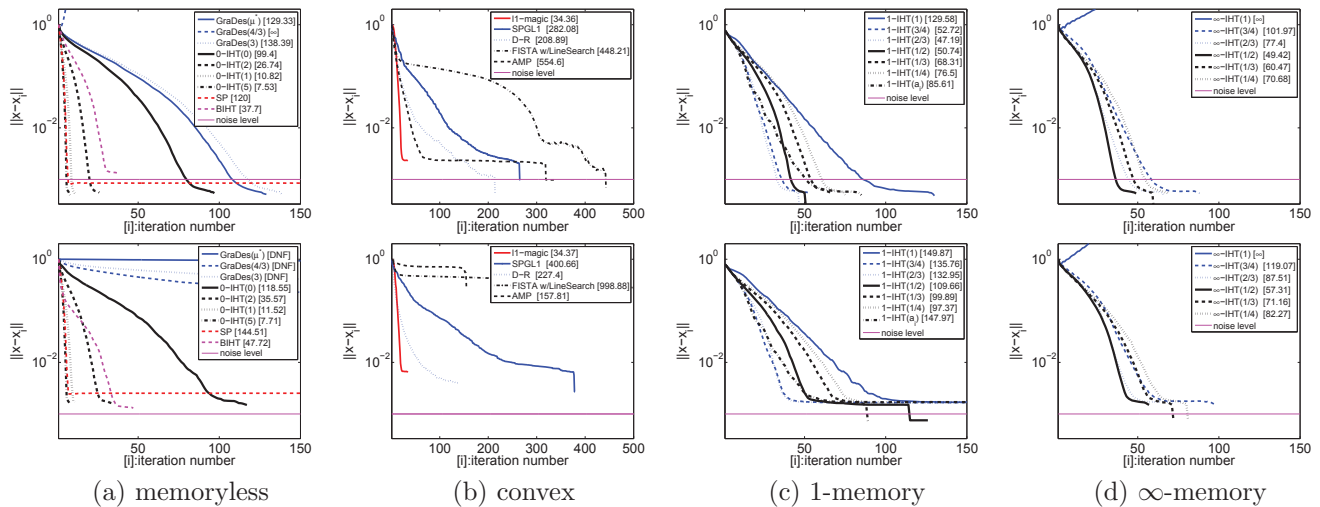
Figure 1. Top/bottom row corresponds to dense/sparse matrix case.

(a) memoryless	(b) convex	(c) 1-memory	(d) ∞-memory

5. ACCELERATION VIA ∞-MEMORY

Motivation: The approximate message passing (AMP) algorithm leverages a heuristic, called the Onsager correction, from statistical physics to improve the IST algorithm.[11] The AMP recursion is

$$x_{i+1} = T_{\lambda_i}\left(x_i + A^t z_i\right), \quad z_i = r_i + z_{i-1}\frac{\|x_{i-1}\|_0}{M}; \tag{23}$$

where $\lambda_i = \|z_i\|/\sqrt{M}$, $r_i = u - Ax_i$, and $\|x\|_0$ counts the number of non-zero entries of x.

Main idea: We propose the following hard thresholding version of AMP based on our step size selection scheme

$$x_{i+1} = H_K\left(x_i + y_i\right), \quad y_i = -0.5\bar{\mu}_i\nabla_{\mathcal{S}_i}f(x_i) + \tau_i y_{i-1}; \tag{24}$$

where $\tau_i \in (0,1)$ controls the momentum (e.g., $\tau_i = K/M$ based on (23)). We categorize the algorithm in (24) as an ∞-memory method since it uses a weighted sum of previous gradients (e.g., if $\tau_i = \tau$, then $y_i = -0.5\sum_{j=1}^{i}\bar{\mu}_j\tau^{i-j}\nabla_{\mathcal{S}_i}f(x_i)$ with $y_0 = 0$).

PROPOSITION 4. *Let $\tau_i = 1/4$ and $c = \sqrt{\frac{2\delta_{2K}}{1-\delta_{3K}}} < 1/8$. The output x_i of (24) satisfies the following:*

$$\|x^* - x_i\| \leq D_1\rho_+^i + D_2\rho_-^i + \gamma\|n\|, \tag{25}$$

where $\rho_{\pm} = (1/8 + c) \pm \sqrt{(1/8 + c)^2 + 1/2}$, and γ, D_1, and D_2 are constants.

We provide a proof for Proposition 4 in the Appendix.

6. EXPERIMENTS

Prologue: We refer to the memoryless IHT algorithms as 0-IHT(#), where # is the decimal representation of the binary number, generated by the options (SolveNewtonb, GradientDescentx, SolveNewtonx). We refer to the 1-memory IHT algorithm as 1-IHT(τ_i), and explicitly specify the parameter. Similarly, we refer to the ∞-memory algorithm as ∞-IHT(τ_i).

In this paper, we only provide experiments with a restricted set of the options (# = 0, 1, 2, 5) for 0-IHT methods. We also do not consider other variants of the 1- and ∞-memory algorithms, as in Algorithm 1, which can provide other computational trade-offs.

Table 1. Per iteration cost of the proposed methods

$$
\begin{bmatrix}
\text{0-IHT(0)} \\
\text{0-IHT(1)} \\
\text{0-IHT(2)} \\
\text{0-IHT(5)} \\
\text{1-IHT} \\
\infty\text{-IHT}
\end{bmatrix}
=
\begin{bmatrix}
\mathbb{H}_N + \mathbb{H}_{2K} + \nabla_N \\
\mathbb{H}_N + \mathbb{H}_{2K} + \nabla_N + \mathbb{N}_K \\
\mathbb{H}_N + \mathbb{H}_{2K} + \nabla_N + \nabla_K \\
\mathbb{H}_N + \mathbb{H}_{2K} + \nabla_N + \mathbb{N}_{2K} + \mathbb{N}_K \\
\mathbb{H}_N + \mathbb{H}_{3K} + \nabla_N \\
2\mathbb{H}_N + \nabla_N
\end{bmatrix}
$$

Caveat emptor: We mainly focus on the iteration count of the hard thresholding methods to illustrate the acceleration due to the schemes we propose. In order to better estimate the total computational complexity, we first provide an analysis of complexity per iteration. Below is a description of the basic operations:

$H_K(\cdot)$: We denote the computational cost of this operation as $\mathbb{H}_d$, where the subscript d refers to the effective dimension of the combinatorial projection. For instance, $d = N$ when we calculate $\mathcal{S}$ in (11), whereas $d = |\mathcal{S}|$ when we calculate $H_K(b)$. For many interesting structured sparsity models, the combinatorial projection is manageable. For instance, when Σ_K^N is the set of all K-sparse signals, this operation amounts to sorting with $\mathbb{H}_N = \mathcal{O}(N \log N)$ complexity. When Σ_K^N is the set of K-tree sparse signals, a dynamic program can obtain the combinatorial projection with $\mathbb{H}_N = \mathcal{O}(N \log N)$ complexity.[12]

$\nabla f(\cdot)$: We denote the computational cost of this operation as ∇_d, where the subscript d refers to the effective dimension of the needed gradient. $A^t u$ can be precalculated. Assuming $A^t A$ can be stored, the computational cost is then dominated by $A^t A$ applied to a K-sparse vector. Then, $\nabla_N = \mathcal{O}(KN)$ and $\nabla_K = \mathcal{O}(K^2)$ (GradientDescentx=1) for general matrices.

SolveNewton(): We denote the computational cost of this operation as $\mathbb{N}_d$, where d is the effective dimension of the Newton system. The effective dimension is $d = |\mathcal{S}|$ when (SolveNewtonb=1), whereas $d = K$ when (SolveNewtonx=1). The main complexity of this operation is dominated by the solution of (presumably) well-conditioned $d \times d$ symmetric linear system of equations. We use conjugate gradients for the solution of this problem, where $L = 50$ is the upper-bound on the number of iterations, leading to $\mathbb{N}_d = \mathcal{O}(Ld^2)$.
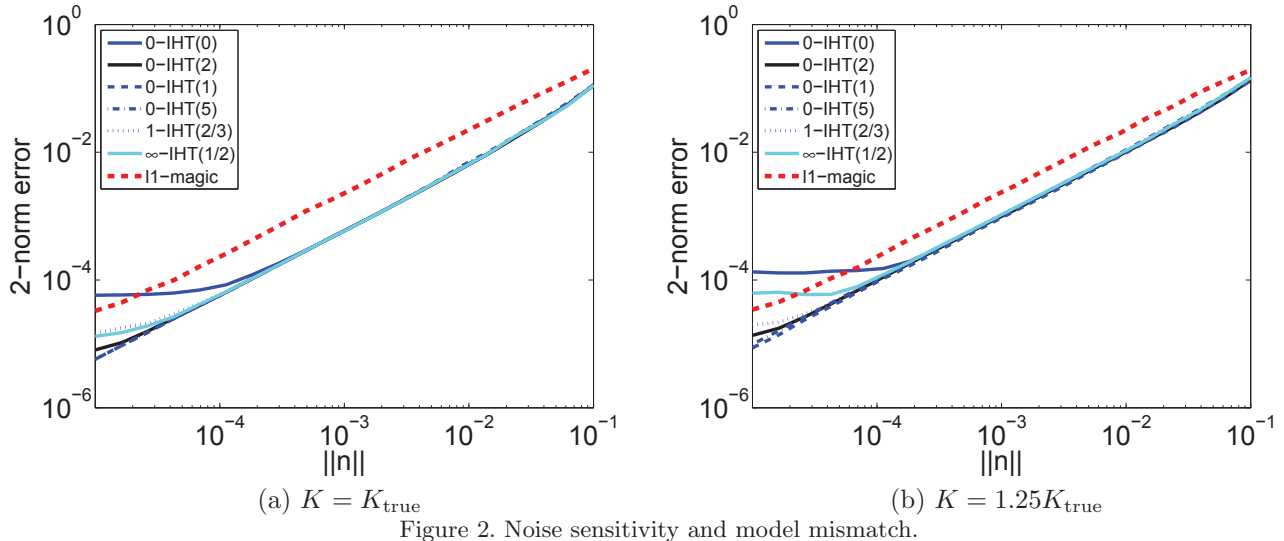
Table 1 provides a summary.

The competition: To illustrate the effectiveness of our acceleration schemes, we also test the following algorithms: ℓ_1-magic (an interior point algorithm), which uses conjugate gradients for solution of the Newton system ($L = 200$ by default); SPGL1 (spectral gradient method), which on the average requires one multiplication by A and two by A^t per iteration;[19] and Douglas-Rachford (D-R) splitting,[20] which is a monotone operator splitting technique, requiring one multiplication by A and one by A^t, if A is a tight frame (otherwise a constant factor more by each).

To solve (18), we use FISTA with line search (a simplified version is discussed in Section 4), and the AMP algorithm. The AMP algorithm requires one multiplication by A and A^t each at every iteration. FISTA's base requirements are the same with a constant factor increase for the line search steps.

We also compare against Blumensath's most recent accelerated IHT method (BIHT) that use adaptive step size strategy,[7] subspace pursuit (SP)[15] as well as GraDes[5] for which we calculate the optimal step-size μ^*, using concentration-of-measures.

Set-up: We test the algorithms in two distinct regression matrix settings. Case 1[dense matrix]: A is a random matrix whose entries are iid Gaussian with zero mean and variance $1/M$. For such matrices, it is possible to show that $\mu^* = 1 + \delta_{2K'} \lesssim (1 + \sqrt{2K/M} + t)^2$ with probability $1 - \exp\left(-Mt^2/2\right)$. We use $Mt^2/2 = 10$ for GraDes(μ^*). Case 2[sparse matrix]: A is the normalized adjacency matrix of an unbalanced 8-regular expander graph. Such matrices have the RIP in the 1-norm, which corresponds to $\|Ax\|_2 \leq \|Ax\|_1 \leq \|x\|_1 \leq \sqrt{K}\|x\|_2$. Hence, we use $\mu^* = 2K$ for GraDes(μ^*) and also use the algorithm's suggested settings, where $\mu = 4/3$ and $\mu = 3$.

(a) $K = K_{\text{true}}$ (b) $K = 1.25K_{\text{true}}$

Figure 2. Noise sensitivity and model mismatch.

To demonstrate the convergence speeds, we generate 100 realizations of $K = 100$ sparse signal in $N = 1000$-dimensions with unit norm, whose nonzero coefficients are iid Gaussian. We pick $M = 400 = 4K$. We then add Gaussian noise to the observations, whose norm is $\|n\| = 10^{-3}$. We provide the hard thresholding methods with the true sparsity, the convex optimization methods with the correct noise and the soft-thresholding values. All the algorithms are tested for the same signal-matrix-noise realizations. All the algorithms use the same convergence tolerance $\epsilon = 10^{-5}$.

Performance summary: Figure 1 illustrates effectiveness of the proposed acceleration schemes on dense and sparse matrix settings. In the figure, the error curves are the median values across realizations over each iteration. We also indicate the *average* number of iterations each algorithm takes to reach the convergence tolerance, next to the algorithm names.

The results provide good empirical support for our step size selection procedure. For instance, 0-IHT(0), which only use the adaptive step size selection procedure, converges faster than GraDes(μ^*), since μ^* is a conservative value that is valid for all Σ_K^N. 1-memory and inf-memory methods also accelerate the convergence of 0-IHT(0) algorithm. The algorithm 1-IHT(a_i) use the weights a_i in (20). The results favor 0-IHT(2) algorithm over the other alternatives when H_K is cheap. Otherwise, 0-IHT(1) is preferred since it has a smaller iteration count, and it needs to solve a smaller Newton system.

While SP and AMP quickly reach a "good" solution in the tests, they did not reach the desired accuracy in many cases and continued iterating until MaxIterations. Moreover, the AMP algorithm performed poorly with sparse matrices (we believe that it requires a different soft thresholding rule).

Robustness summary: Figure 2 illustrates the robustness of the accelerated IHT methods vs. the linear programming approach. For this test, we vary the noise variance, repeat the above test 100 times, and record the average reconstruction errors. Moreover, we also input a target sparsity K, which is not the true target sparsity K_{true}, for the hard thresholding methods. The accelerated methods appear insensitive to the input value K as long as it overestimates the true sparsity, and the (K, M)-pair for the input K is on the phase transition curve.[21] If the sparsity is underestimated, then the reconstruction error grows proportional to the mismatch. As all the methods were limited by 150 iterations, the method 0-IHT(0) tapers off at low noise as it needs more iterations to reach the high accuracy solution.

7. CONCLUSIONS

We derive acceleration schemes that provide salient computational trade-offs for the class of hard thresholding methods for sparse approximation. Our approach in essence reinterprets the convex optimization algebra specifically for sparse sets. Hence, the proposed IHT methods, as they iterate, optimally exploit the sparse scaffold

on which the approximation problem resides. This leads to convergence speed and computational advantages over the convex sparse recovery algorithms (e.g., based on soft-thresholding), which have to iterate over dense putative solutions until they reach a sparse solution. Empirical results demonstrate that our acceleration schemes are quite effective without sacrificing the robustness.

APPENDIX A. PROOFS OF KEY RESULTS

Proof. [Property 2] To establish (7), we use the triangle inequality $\|x^* - \bar{b}\| \leq \|\bar{b} - b\| + \|x^* - b\|$, and note that b is closer to $\bar{b}$ than to x^*. To prove (7), we first leverage the RIP property: $\|x^* - b\| \leq \frac{1}{\sqrt{1-\delta_{K'}}}\|A(x^* - b)\|$, which is followed by another triangle inequality: $\|A(x^* - b) + n - n\| \leq \|u - Ab\| + \|n\|$. Note that depending on the support of x^* and b, K' is at most $K + B$. To obtain (9), we apply RIP on the left hand side of (7): $\frac{1}{\sqrt{1+\delta_{2K}}}\|A(x^* - \bar{b})\| \leq \|x^* - \bar{b}\|$. Now, $\|A(x^* - \bar{b}) + n - n\| \geq \|u - A\bar{b}\| - \|n\|$, whenever $f(\bar{b}) \geq \|n\|^2$. Combining this observation with (8), we obtain the final inequality. □

Proof. [Lemma 2.1] We first exploit Property 1(1): $\|A(x^* - a)\|^2 = f(a) - f(x^*) - \langle \nabla f(x^*), a - x^* \rangle$. Noting that $f(x^*) = \|n\|^2$, we have $\|A(x^* - a)\|^2 + \|n\|^2 \leq c^2\|n\|^2 + 2\langle A^T(u - Ax^*), a - x^* \rangle$. Applying Cauchy-Schwarz to the right hand side of this equation and rearranging, we obtain $\|A(x^* - a)\|^2 - 2\|n\|\|A(x^* - a)\| + \|n\|^2 \leq c^2\|n\|^2$. Taking the square root of both sides and applying RIP, we reach the desired. □

Proof. [Proposition 1] We first define $\mathcal{S}_i^* = \mathcal{X}^* \cup \mathcal{X}_i$ where $\mathcal{X}^* = \text{supp}(x^*)$, and note that $\|\nabla_{\mathcal{S}_i} f(x_i)\| \geq \|\nabla_{\mathcal{S}_i^*} f(x_i)\|$. This is because $\mathcal{S}_i \cup \mathcal{X}_i$, as defined in (11), includes the K-largest elements in magnitude of the gradient. Let $L^* = 2(1 + \delta_{2K})$ and $\tilde{b} = x_i - \frac{1}{L^*}\nabla_{\mathcal{S}_i} f(x_i)$. By using Property 1(2), we have

$$f(\tilde{b}) - f(x_i) - \left\langle \nabla f(x_i), \frac{-1}{L^*}\nabla_{\mathcal{S}_i} f(x_i) \right\rangle \leq \frac{L^*}{2}\left\| \frac{\nabla_{\mathcal{S}_i} f(x_i)}{L^*} \right\|^2$$

$$\Rightarrow f(\tilde{b}) - f(x_i) \leq -\frac{1}{2L^*}\|\nabla_{\mathcal{S}_i} f(x_i)\|^2 \leq -\frac{1}{2L^*}\|\nabla_{\mathcal{S}_i^*} f(x_i)\|^2$$

$$\leq \frac{L^*}{2}\left\| x^* - \left(x_i - \frac{1}{L^*}\nabla_{\mathcal{S}_i^*} f(x_i) \right) \right\|^2 - \frac{1}{2L^*}\|\nabla_{\mathcal{S}_i^*} f(x_i)\|^2$$

$$= \left\langle \nabla_{\mathcal{S}_i^*} f(x_i), x^* - x_i \right\rangle + \frac{L^*}{2}\|x^* - x_i\|^2$$

Via Property 1(3), we have the following bound

$$\left\langle \nabla_{\mathcal{S}_i^*} f(x_i), x^* - x_i \right\rangle \leq f(x^*) - f(x_i) - (1 - \delta_{2K})\|x^* - x_i\|^2,$$

when combined with the bound right above leads to

$$f(\tilde{b}) \leq f(x^*) + 2\delta_{2K}\|x^* - x_i\|^2. \tag{26}$$

Note that $f(b) \leq f(\tilde{b})$ as $\text{supp}(\tilde{b}) = \text{supp}(b) = \mathcal{S}_i$, and b is updated with a step size $\bar{\mu}_i$ that minimizes $f(x)$ on $\mathcal{S}_i$, as described in (12) and (13). Substituting $f(x^*) = \|n\|^2$ into (26), and leveraging the fact that $a_1^2 \leq a_2^2 + a_3^2 \Rightarrow a_1 \leq a_2 + a_3$ for $a_i \geq 0$, we obtain (14). To reach (15), we simply recall (8) in Property 2 and substitute (14) with $K' = 3K$.

To establish (16), we look at the single root of the characteristic equation of the series inequality defined by (15), which is given by $\rho > 0$, as defined in Proposition 1. Assuming $\rho < 1$, which defines the isometry requirements of the algorithm, the series is convergent. At the stationary point, we solve

$$\|x^* - x_\infty\| \leq \rho\|x^* - x_\infty\| + \frac{4}{\sqrt{1 - \delta_{3K}}}\|n\|, \tag{27}$$

to obtain the final result (16). It is easy to check that the first iteration of the algorithm satisfies the recursion (15), completing the proof. □

Proof. [Proposition 2] We revisit the proof of Proposition 1 at (26). Note that $f(b) \leq f(\tilde{b})$ is still satisfied for (SolveNewtonb=1) option. Let us define $\tilde{x} = H_K(b)$. Only this time, we apply the inequality (9) in Property (2) to obtain

$$\|u - A\tilde{x}\| \leq \kappa\sqrt{2\delta_{2K}}\|x^* - x_i\| + C_1\|n\|, \tag{28}$$

where κ is as defined in Property (2) with $K' = 3K$, and $C_1 = 1 + 2\kappa$. We apply RIP to obtain $\|x^* - x_i\| \leq \frac{\|u - Ax_i\| + \|n\|}{\sqrt{1 - \delta_{2K}}}$, when substituted into (28) leads to

$$\|u - A\tilde{x}\| \leq \rho\|u - Ax_i\| + C_2\|n\|, \tag{29}$$

where ρ is as defined in Proposition 2 and $C_2 = 1 + \rho + 2\kappa$. Note that the recursion in (29) is still satisfied if we "refine" $\tilde{x}$ by any operation, restricted to $\mathcal{S}_i$ that decreases $f(x)$. Therefore, (17) holds for all variants Algorithm 1 with $C = (1 - \rho)^{-1}C_2$. □

Proof. [Proposition 3] As $y_i = H_K(b)$, we first recycle (15) from Proposition 1. Only this time, we have $K' = 4K$ since x_i now has $2K$ sparsity: $\|x^* - y_i\| \leq c\|x^* - x_i\| + \frac{4}{\sqrt{1 - \delta_{4K}}}\|n\|$, where c is as defined in Proposition 3. Thanks to the triangle inequality and the new definition of x_{i+1}, we also have $\|x^* - x_{i+1}\| \leq \|x^* - y_i\| + \tau_i\|x^* - y_i\| + \tau_i\|x^* - y_{i-1}\|$. Combining these two expressions, we stumble upon the following second order difference inequality: $\|x^* - y_{i+1}\| \leq$

$$c(1 + \tau_i)\|x^* - y_i\| + c\tau_i\|x^* - y_{i-1}\| + \frac{4}{\sqrt{1 - \delta_{4K}}}\|n\|. \tag{30}$$

Assuming $\tau_i \leq 1$, (22) provides the homogeneous solution and the particular solution, where the roots $\rho_\pm$ are obtained from the characteristic polynomial. The values of C_1 and C_2 depend on the initial conditions. □

Proof. [Proposition 4] Let $\tau_i = \tau \in (0, 1)$. Using (7) from Property 2, we first note that $\|x^* - x_{i+1}\| \leq 2\|x^* - b_i\| + 2\tau\|y_{i-1}\|$, where b_i is calculated as in (12). Similarly, it is clear that $y_i = \sum_{j=1}^{i} \tau^{i-j}(b_j - x_j)$, which allows us to bound the norm of y_i via

$$\|y_{i-1}\| \quad \leq \sum_{j=1}^{i-1} \tau^{i-1-j}\left(\|x^* - b_j\| + \|x^* - x_j\|\right) \tag{31}$$

$$\leq (1 + c)\sum_{j=1}^{i-1} \tau^{i-1-j}\|x^* - x_j\| + C_1\|n\|, \tag{32}$$

where $c = \sqrt{\frac{2\delta_{2K}}{1 - \delta_{3K}}}$ and $C_1 = \frac{2(1-\tau)^{-1}}{\sqrt{1 - \delta_{3K}}}$. In (31), we apply RIP to first obtain $\|x^* - b_j\| \leq \frac{\|u - Ab_j\| + \|n\|}{\sqrt{1 - \delta_{3K}}}$, followed by the inequality (14) from Proposition 1. We then upperbound the summation $\sum_{j=1}^{i-1} \tau^{i-1-j} \leq (1 - \tau)^{-1}$ to obtain C_1.

Combining the statements above, we reach the following inequality for the iterations of (24): $\|x^* - x_{i+1}\| \leq$

$$2c\|x^* - x_i\| + 2(1 + c)\tau \sum_{j=1}^{i-1} \tau^{i-1-j}\|x^* - x_j\| + C\|n\|,$$

where $C = 2C_1$. Let us now suppose that the i-th iterate satisfies $\|x^* - x_i\| \leq D\rho^i + \gamma\|n\|$, for some constants D, $|\rho| < 1$, and γ. We now seek the conditions on c and τ to see if an induction argument can hold for the $(i+1)$-th iterate: $\|x^* - x_{i+1}\| \leq 2Dc\rho^i +$

$$2c\gamma\|n\| + 2(1 + c)\tau \sum_{j=1}^{i-1} \tau^{i-1-j}(D\rho^j + \gamma\|n\|) + C\|n\|.$$

At this juncture, let us assume $\tau = 1/4$. After some laborious algebra, it is possible to show that the induction hypothesis would be satisfied if $c < 1/8$ and $\gamma = 3(1/8 - c)^{-1}C$ with the two values of ρ, as stated in Proposition 4. It is easy to see that the induction hypothesis is satisfied for $i = 1$, completing the proof. □

REFERENCES

[1] Blumensath, T. and Davies, M., "Sampling theorems for signals from the union of finite-dimensional linear subspaces," *Information Theory, IEEE Trans. on* **55**(4), 1872–1882 (2009).

[2] Tibshirani, R., "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)* , 267–288 (1996).

[3] Banerjee, O., El Ghaoui, L., and d'Aspremont, A., "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *J. of Machine Learning Research* **9**, 485–516 (2008).

[4] Candès, E. and Wakin, M., "An introduction to compressive sampling," *Signal Processing Magazine, IEEE* **25**(2), 21–30 (2008).

[5] Garg, R. and Khandekar, R., "Gradient descent with sparsification: An iterative algorithm for sparse recovery with restricted isometry property," in [*ICML*], ACM (2009).

[6] Tropp, J. and Wright, S., "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE* **98**(6), 948–958 (2010).

[7] Blumensath, T., "Accelerated Iterative Hard Thresholding," preprint (2011).

[8] Needell, D. and Tropp, J., "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *ACHA* **26**(3), 301–321 (2009).

[9] Foucart, S., "Hard thresholding pursuit: An algorithm for compressive sensing," preprint (2010).

[10] Bach, F., "Structured sparsity-inducing norms through submodular functions," in [*NIPS*], (2010).

[11] Montanari, A., "Graphical Models Concepts in Compressed Sensing," *preprint, arXiv:1011.4328* (2010).

[12] Baraniuk, R., Cevher, V., Duarte, M., and Hegde, C., "Model-based compressive sensing," *Information Theory, IEEE Trans. on* **56**(4), 1982–2001 (2010).

[13] Juditsky, A. and Nemirovski, A., "On verifiable sufficient conditions for sparse signal recovery via l1 minimization," arXiv:0809.2650v2 (2008).

[14] Blumensath, T. and Davies, M., "Normalized iterative hard thresholding: Guaranteed stability and performance," *Selected Topics in Signal Processing, IEEE Journal of* **4**(2), 298–309 (2010).

[15] Dai, W. and Milenkovic, O., "Subspace pursuit for compressive sensing signal reconstruction," *Information Theory, IEEE Trans. on* **55**(5), 2230–2249 (2009).

[16] A. Kyrillidis, V. C., "Recipes on hard thresholding methods," in [*CAMSAP*], (2011).

[17] Beck, A. and Teboulle, M., "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences* **2**(1) (2009).

[18] Nesterov, Y., "A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$," in [*Soviet Mathematics Doklady*], **27**(2), 372–376 (1983).

[19] van den Berg, E. and Friedlander, M. P., "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing* **31**(2), 890–912 (2008).

[20] Fadili, M. and Starck, J., "Monotone operator splitting for fast sparse solutions of inverse problems," *SIAM J. on Imaging Sciences* , 2005–2006 (2009).

[21] Donoho, D. and Tanner, J., "Precise undersampling theorems," *Proceedings of the IEEE* **98**(6) (2010).