On the Iteration Complexity of Support Recovery via Hard Thresholding Pursuit

Jie Shen [1] **Ping Li** [1]

Abstract

Recovering the support of a sparse signal from its compressed samples has been one of the most important problems in high dimensional statistics. In this paper, we present a novel analysis for the hard thresholding pursuit (HTP) algorithm, showing that it exactly recovers the support of an *arbitrary* s-sparse signal within $\mathcal{O}(s\kappa \log \kappa)$ iterations via a properly chosen proxy function, where κ is the condition number of the problem. In stark contrast to the theoretical results in the literature, the iteration complexity we obtained holds without assuming the restricted isometry property, or relaxing the sparsity, or utilizing the optimality of the underlying signal. We further extend our result to a more challenging scenario, where the subproblem involved in HTP cannot be solved exactly. We prove that even in this setting, support recovery is possible and the computational complexity of HTP is established. Numerical study substantiates our theoretical results.

1. Introduction

In the last two decades, pursuing a sparse representation for high dimensional data has become one of the most significant problems in machine learning. To seek a sparse solution, a large body of work is devoted to efficient methods, including the convex formulation, for instance, basis pursuit (Chen et al., 1998) and the Lasso (Tibshirani, 1996), as well as greedy pursuits, e.g., orthogonal matching pursuit (Pati et al., 1993), iterative hard thresholding (Daubechies et al., 2004) and hard thresholding pursuit (HTP) (Foucart, 2011), along with elegant theoretical understanding on parameter estimation and support recovery in either ideal setting or noisy scenario (Candès & Tao, 2005; Wainwright, 2009; Tropp & Gilbert, 2007; Cai et al., 2010; Blumensath & Davies, 2009; Bouchot et al., 2016).

Compared to parameter estimation, i.e., bounding the ℓ_2 distance between the solution and the desired sparse signal, support recovery is a much more challenging task and it usually requires more stringent conditions. See Tropp (2004); Zhao & Yu (2006); Yuan & Lin (2007); Zhang (2009) for some early results and Nguyen & Tran (2013); Loh & Wainwright (2014) for more recent developments. Nevertheless, if the support of a signal can be predicted by a method, then the solution returned by the method immediately enjoys the oracle property, i.e., with optimal statistical rate (Wainwright, 2009). Thereby, support recovery has received broad attention in recent years (Osher et al., 2016; Wang et al., 2016; Bouchot et al., 2016).

In this work, we follow the research line with a particular interest in the hard thresholding pursuit algorithm, which exhibits encouraging performance among many machine learning applications. The algorithm was originally presented by Foucart (2011) for recovering the true signal in compressed sensing (Donoho, 2006). Yuan et al. (2014) suggested using the HTP algorithm for general sparsity-constrained machine learning problems, and they showed that the solution obtained from HTP converges with a geometric rate. Very recently, a rigorous theoretical analysis on when HTP guarantees support recovery was independently carried out by Bouchot et al. (2016) and Yuan et al. (2016). In Bouchot et al. (2016), they considered the compressed sensing problem and illustrated that HTP recovers the support of the true signal in finite iterations if the restricted isometry property (RIP) condition holds (Candès & Tao, 2005). Yuan et al. (2016) showed that in some situations, HTP eventually terminates and guarantees support recovery without assuming the RIP condition.

Although these appealing theoretical results characterize the behavior of HTP in particular regimes, it turns out that a thorough understanding on when HTP identifies the support of an *arbitrary* sparse signal is missing in the literature. To be more precise, the RIP condition used in Bouchot et al. (2016) amounts to imposing a small condition number for the underlying problem, which may not be practical for machine learning applications where the condition number usually grows with the sample size. To guar-

[1] Rutgers University, Piscataway, New Jersey, USA. Jie Shen: js2007@rutgers.edu, Ping Li: pingli@stat.rutgers.edu.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

antee the support recovery of an s-sparse signal, Yuan et al. (2016) required that the signal of interest is the unique global minimizer of a sparsity-constrained program (which invokes the RIP condition), or that HTP maintains denser iterates. This poses an interesting question of whether HTP is able to recover the support without the RIP assumption, or the optimality of the signal, or the relaxed sparsity.

In addition, an insightful analysis on the performance of HTP in a realistic scenario is missing. For concreteness, recall that HTP proceeds as follows:

$$\begin{aligned}
&\text{(HTP1)} \quad \boldsymbol{b}^{t+1} = \boldsymbol{x}^t - \eta \nabla F(\boldsymbol{x}^t),\\
&\text{(HTP2)} \quad S^{t+1} = \operatorname{supp}\left(\boldsymbol{b}^{t+1}, k\right),\\
&\text{(HTP3)} \quad \boldsymbol{x}^{t+1} = \mathop{\arg\min}_{\operatorname{supp}(\boldsymbol{x}) \subset S^{t+1}} F(\boldsymbol{x}),
\end{aligned}$$

where $\eta > 0$ is a step size, $\operatorname{supp}\left(\boldsymbol{b}^{t+1}, k\right)$ denotes the support of the k largest absolute elements of $\boldsymbol{b}^{t+1}$ and $F(\boldsymbol{x})$ is a properly chosen function. For general machine learning problems, we are only guaranteed with ϵ-approximate solutions in the third step, i.e., for all $t \geq 0$,

$$F(\boldsymbol{x}^{t+1}) - F(\boldsymbol{x}_*^{t+1}) \leq \epsilon,$$

where $\boldsymbol{x}_*^{t+1}$ is the global minimizer of $F(\boldsymbol{x})$ restricted on S^{t+1}. Related to the inexact solutions, a natural question to ask is how the accuracy parameter ϵ affects the recovery performance of HTP, additively or progressively.

Another issue coming up with the inexact iterates is that the usually employed stopping criterion $S^{t+1} = S^t$ may not be valid, which makes part of the analysis in Yuan et al. (2016) not applicable to this setting. Note that when exact solutions are available, HTP becomes stationary as soon as the detected support does not change, since the solutions are entirely determined by the support. Yuan et al. (2016) made use of this feature to establish theoretical guarantee for HTP. However, allowing approximate iterates quickly changes the premise because many stochastic solvers, e.g., stochastic gradient descent, introduce randomness, rendering (HTP3) outputs different results even restricted on the same support set.

1.1. Contribution

We make the following contribution in this paper. First, suppose that (HTP3) has exact solutions, we show that under very mild conditions, HTP either terminates early or guarantees support recovery of an arbitrary s-sparse signal within $\mathcal{O}\left(s\kappa \log \kappa\right)$ iterations. Then we move on to the inexact case, and prove that under the RIP condition or using a relaxed sparsity, support recovery with the same iteration complexity holds provided that the optimization error ϵ is small compared to the magnitude of the target signal. As a consequence, we present the first bound on the computational complexity of HTP. For concreteness, we relate our deterministic results to two prevalent statistical models, and show that the conditions involved in our theorems can be met with high probability.

We also revisit the role of $F(\boldsymbol{x})$ of the HTP algorithm. Previous work, for example, Jain et al. (2014), tends to treat $F(\boldsymbol{x})$ as an objective function, the choice of which depends on the underlying problem and the signal, and views HTP as an optimization procedure towards the optimal solution. Interestingly, we find that $F(\boldsymbol{x})$ behaves more like a *proxy function* that guides HTP to the target signal. Hence, to recover a signal, we have many more choices of $F(\boldsymbol{x})$ as far as it satisfies the conditions to be present (see Section 4).

From a high level, the paper shares the same merit of Bouchot et al. (2016); Yuan et al. (2016), i.e., recovering a sparse signal. Hence, part of our proof is inspired by their work. Yet, we establish novel RIP-free results based on a more careful analysis for the problem structure. See a detailed comparison in Section 3.

1.2. Notation

Throughout the paper, we use bold lowercase letters, e.g., $\boldsymbol{v}$, to denote a column vector. The support of a vector $\boldsymbol{v}$ is denoted by $\operatorname{supp}(\boldsymbol{v})$, whereas that of the largest k absolute elements is denoted by $\operatorname{supp}(\boldsymbol{v}, k)$. Both $\|\boldsymbol{v}\|_0$ and $|\operatorname{supp}(\boldsymbol{v})|$ are used to count the non-zeros in $\boldsymbol{v}$. Suppose that $\Omega \subset \{1, 2, \dots, d\}$ is an index set, then for $\boldsymbol{v} \in \mathbb{R}^d$, $\boldsymbol{v}_\Omega$ can either be explained as an $|\Omega|$-dimensional vector or a d-dimensional vector with the elements outside of Ω set to zero. The Euclidean norm of a vector $\boldsymbol{v}$ is denoted by $\|\boldsymbol{v}\|$. We write boldface capital letters, e.g., $\boldsymbol{A}$, for matrices, and the transpose is denoted by $\boldsymbol{A}^\top$.

The s-sparse vector $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ is the target signal we aim to recover, and we reserve the capital letter S for its support. We define $\bar{\boldsymbol{x}}_{\min} > 0$ as the absolute value of the smallest element (in magnitude) of $\bar{\boldsymbol{x}}_S \in \mathbb{R}^s$. With a slight abuse of the notation, $\nabla_k F(\bar{\boldsymbol{x}})$ should be explained as the vector consisting of the top k elements (in magnitude) of $\nabla F(\bar{\boldsymbol{x}})$ rather than the kth component of $\nabla F(\bar{\boldsymbol{x}})$.

1.3. Roadmap

The remainder of the paper is organized as follows. Section 2 introduces the problem setting and some preliminary results that the main theorems build on. Section 3 presents the main results of this paper with a detailed comparison to closely related work. In Section 4, we specialize our results to two concrete statistical models. A proof sketch of the main results is given in Section 5. Next, we verify our theoretical results with extensive numerical study in Section 6 and Section 7 concludes the paper. Technical lemmas and the full proof are deferred to the appendix (see the supplementary file).

2. Problem Setup and Preliminary Results

In this section, we introduce the problem setting and some preliminary consequences on which our main results build. To be clear, the target signal $\bar{\boldsymbol{x}} \in \mathbb{R}^d$ we consider in this paper is only endowed with sparsity.

Our analysis depends on the following two properties of the function $F(\boldsymbol{x})$.

Definition 1. A differentiable function $F(\boldsymbol{x})$ is said to be restricted strongly convex (RSC) with parameter $m_K > 0$, if for all vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ with $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq K$,

$$F(\boldsymbol{x}) - F(\boldsymbol{x}') - \langle \nabla F(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle \geq \frac{m_K}{2} \|\boldsymbol{x} - \boldsymbol{x}'\|^2 .$$

Definition 2. A differentiable function $F(\boldsymbol{x})$ is said to be restricted smooth (RSS) with parameter $M_K > 0$, if for all vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$ with $\|\boldsymbol{x} - \boldsymbol{x}'\|_0 \leq K$,

$$F(\boldsymbol{x}) - F(\boldsymbol{x}') - \langle \nabla F(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle \leq \frac{M_K}{2} \|\boldsymbol{x} - \boldsymbol{x}'\|^2 .$$

In particular, we require that the RSC condition holds at sparsity level $k + s$ and the RSS condition holds at sparsity level $2k$, respectively. That is,

$(A1)$ $F(\boldsymbol{x})$ is m_{k+s}-restricted strongly convex;

$(A2)$ $F(\boldsymbol{x})$ is M_{2k}-restricted smooth.

Note that the RSC and RSS conditions are now standard and are widely utilized for establishing performance guarantees for a variety of popular algorithms. See, for example, Negahban et al. (2009); Agarwal et al. (2012); Jain et al. (2014) and Loh & Wainwright (2014). For simplicity, throughout the paper we write $m := m_{k+s}$ and $M := M_{2k}$. We also denote $\kappa = M/m$ which is actually the (restricted) condition number of the problem.

The first result states that if (HTP3) outputs exact solutions, then HTP decreases the function value with a geometric rate before the stopping criterion (i.e., $S^{t+1} = S^t$) is met. Formally, we have the following proposition.

Proposition 1. *Consider the HTP algorithm with exact solutions in (HTP3). Assume (A1) and (A2), pick $\eta < 1/M$ in (HTP1) and set $k = s$ in (HTP2). Then before HTP terminates, it holds that for all $t \geq 0$,*

$$F(\boldsymbol{x}^{t+1}) - F(\bar{\boldsymbol{x}}) \leq \mu \left(F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \right),$$

where

$$\mu = 1 - \frac{2\eta m(1 - \eta M)}{1 + s} \in (0, 1).$$

Remark. Note that we did not assume the optimality of $\bar{\boldsymbol{x}}$ with respect to the function $F(\boldsymbol{x})$. In other words, Prop. 1 holds even for $F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) < 0$. It is also worth mentioning that by the proposition, we can deduce

$$F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}}) \leq \mu^t \left(F(\boldsymbol{x}^0) - F(\bar{\boldsymbol{x}}) \right).$$

However, the above inequality does not imply the convergence of $\{F(\boldsymbol{x}^t)\}_{t \geq 0}$, since $F(\boldsymbol{x}^t) - F(\bar{\boldsymbol{x}})$ is not bounded from below. Rather, it is invoked to establish parameter estimation for HTP.

The following proposition shows that when the conditions in Prop. 1 are satisfied, we have an accurate estimate on the signal in the ℓ_2 metric.

Proposition 2. *Assume same conditions as in Prop. 1. Then before HTP terminates, the following holds for $t \geq 0$:*

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \sqrt{2\kappa} (\sqrt{\mu})^t \left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{3}{m} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\|,$$

where μ is given in Prop. 1.

In the literature, a variety of work has established theoretical guarantees on parameter estimation, either under the RIP condition (Bouchot et al., 2016) or by relaxing the sparsity (Yuan et al., 2016). In contrast, neither of the conditions are assumed in Prop. 2, owing to a careful analysis on the connection between $\nabla F(\boldsymbol{x}^t)$ and $\bar{\boldsymbol{x}}$. See the supplementary file for the proof. However, we point out that such an appealing behavior is not guaranteed if (HTP3) does not output exact solutions, and in this case, we have to relax the sparsity or use the RIP condition. In particular, let

$$\boldsymbol{x}_*^t = \operatorname*{arg\,min}_{\mathrm{supp}(\boldsymbol{x}) \subset S^t} F(\boldsymbol{x}),$$

and consider that (HTP3) outputs $\boldsymbol{x}^t$ obeying

$$\mathrm{supp}\left(\boldsymbol{x}^t\right) \subset S^t,\ F(\boldsymbol{x}^t) - F(\boldsymbol{x}_*^t) \leq \epsilon. \tag{1}$$

Note that this is a realistic scenario because even for simple functions, e.g., $F(\boldsymbol{x})$ is the logistic loss, convex solvers only ensure ϵ-approximate solutions. The major issue coming up with the ϵ-approximate solutions is that the gradient of $F(\boldsymbol{x})$ evaluated at $\boldsymbol{x}^t$ does not vanish on the support S^t, which makes our technical analysis of Prop. 2 invalid. Yet, we can still bound it under proper conditions.

Lemma 3. *Assume (A2) and (1). Then at any iteration $t \geq 0$, we have*

$$\left\|\nabla_{S^t} F(\boldsymbol{x}^t)\right\| \leq \sqrt{2M\epsilon}.$$

Based on the lemma, we show the following RIP-based result for parameter estimation.

Proposition 4. *Consider the HTP algorithm with inexact solutions (1). Suppose that the condition number $\kappa < 1.25$*

and set $k = s$ in (HTP2). Then picking $\eta = \eta'/M$ with $\kappa - 0.25 < \eta' < 1$ guarantees

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq (\sqrt{2}(\kappa - \eta'))^t \left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{6\kappa}{m} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\| + \frac{4\sqrt{M\epsilon}}{m}.$$

As the RIP condition is hard to fulfill for many machine learning problems, Jain et al. (2014) proposed to relax the sparsity parameter $k = \mathcal{O}\left(\kappa^2 s\right)$ in order to alleviate it. Shen & Li (2016) further showed that by relaxing the sparsity, a stochastic solver is able to produce an accurate solution for sparsity-constrained programs. Inspired by their interesting work, we derive the following result for HTP.

Proposition 5. *Consider the HTP algorithm with inexact solutions (1). Pick $\eta < 1/M$ and let $k \geq 2s + \frac{8s}{\eta^2 m^2}$. Then*

$$\left\|\boldsymbol{x}^t - \bar{\boldsymbol{x}}\right\| \leq \sqrt{2\kappa}(\sqrt{\mu})^t \left\|\boldsymbol{x}^0 - \bar{\boldsymbol{x}}\right\| + \frac{3}{m} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\| + \sqrt{\frac{4\epsilon}{m(1-\mu)}},$$

where

$$\mu = 1 - \frac{\eta m(1 - \eta M)}{2}.$$

3. Main Results

This section is dedicated to a deterministic analysis on the performance of HTP. We first treat the exact case, i.e., (HTP3) outputs exact solutions, along with a detailed comparison with previous work in the literature. Then we demonstrate that even when (HTP3) is solved approximately up to an ϵ-accuracy, support recovery is still possible provided that ϵ is small enough compared to the magnitude of the target signal.

The following theorem is one of the main results in the paper. It justifies that under proper conditions, HTP recovers the support of $\bar{\boldsymbol{x}}$ using finite iterations.

Theorem 6. *Consider the HTP algorithm with exact solutions in (HTP3). Assume (A1) and (A2). Pick $\eta < 1/M$ in (HTP1) and $k = s$ in (HTP2). Then HTP either terminates early, or recovers the support of $\bar{\boldsymbol{x}}$ using at most*

$$t_{\max} = \left(\frac{3\log\kappa}{\log(1/\mu)} + \frac{2\log(2/(1-\lambda))}{\log(1/\mu)} + 2\right) \left\|\bar{\boldsymbol{x}}\right\|_0 \quad (2)$$

iterations, provided that for some constant $\lambda \in (0, 1)$

$$\bar{\boldsymbol{x}}_{\min} \geq \frac{2\sqrt{2} + \sqrt{\kappa}}{m\lambda} \left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\|. \quad (3)$$

Above, the quantity μ is given by

$$\mu = 1 - \frac{2m\eta(1 - \eta M)}{1 + s} \in (0, 1).$$

In the theorem, we recall that $\bar{\boldsymbol{x}}_{\min}$ is the minimum absolute value of the non-zeros of $\bar{\boldsymbol{x}}$. Below we discuss the important messages conveyed by the theorem and contrast our result to prior work. For ease of exposition, we write $\eta = \eta'/M$ for some constant $\eta' \in (0, 1)$, and it quickly indicates that $\mu = 1 - \mathcal{O}\left(1/\kappa\right)$.

Iteration complexity. We remind that the first term in (2) plays the most crucial role, since it upper bounds the other two for sufficiently large κ. In the regime where κ itself is bounded by a constant from above, the iteration complexity is simply explained as $\mathcal{O}\left(\left\|\bar{\boldsymbol{x}}\right\|_0\right)$. Asymptotically, we can show that the iteration complexity is dominated by $\kappa\log\kappa$ as κ tends to infinity, that is,

$$t_{\max} = \mathcal{O}\left(\left\|\bar{\boldsymbol{x}}\right\|_0 \kappa\log\kappa\right).$$

This follows from a simple calculation on the Taylor expansion of $\log(1/\mu)$ at the point $x = 1$, with μ being replaced with $1 - \mathcal{O}\left(1/\kappa\right)$. Note that the number of iterations we obtained for support recovery is as few as that for accurate parameter estimation (see Prop. 2). It is also worth mentioning that the linear dependency on the sparsity of $\bar{\boldsymbol{x}}$ is nearly optimal, because in the worst case HTP may take several steps to pick only one correct support.

Conditions. We also emphasize that the condition (3) is now ubiquitous for analyzing the support recovery performance. The quantity $\bar{\boldsymbol{x}}_{\min}$ involved is natural, because a signal with large magnitude is easier to recover than those with small or vanishing components. To see why $\left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\|$ is used to lower bound the magnitude of $\bar{\boldsymbol{x}}$, let us consider the compressed sensing problem as an example. Suppose that we observe the response vector $\boldsymbol{y}$, which obeys $\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e}$ for a given design matrix $\boldsymbol{A}$ and some noise $\boldsymbol{e}$. In order to recover the true parameter $\bar{\boldsymbol{x}}$, we may choose $F(\boldsymbol{x})$ as the least-squares, of which the derivative evaluated at $\boldsymbol{x} = \bar{\boldsymbol{x}}$ is given by

$$\nabla F(\bar{\boldsymbol{x}}) = \boldsymbol{A}^\top \left(\boldsymbol{A}\bar{\boldsymbol{x}} - \boldsymbol{y}\right) = -\boldsymbol{A}^\top \boldsymbol{e}.$$

Then the RIP condition asserts that

$$\left\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\right\| \geq \sqrt{1 - \delta_{k+s}} \left\|\boldsymbol{e}\right\|,$$

where $\delta_{k+s} \in (0, 1)$ is the $(k + s)$-th restricted isometry constant (Candès & Tao, 2005). Therefore, imposing the condition (3) amounts to distinguishing the true signal from the observation noise.

Comparison to prior work. We contrast our result to the state-of-the-art work of Yuan et al. (2016). To recover a sparse signal $\bar{\boldsymbol{x}}$, Yuan et al. (2016) required the condition number $\kappa < 1.14$, which might be too restrictive to general machine learning problems where the condition number grows with sample size. In addition, support recovery was established only for a carefully chosen $F(\boldsymbol{x})$, i.e., $\bar{\boldsymbol{x}}$

Table 1. **Comparison to previous work on HTP-style algorithm. We present the first support recovery guarantee for an arbitrary sparse signal without assuming the RIP condition or relaxing the sparsity.**

Result	Target sparse signal	RIP-free	No sparsity relaxation	Support recovery
Foucart (2011)	true signal	✗	✓	✗
Yuan et al. (2014)	arbitrary	✗	✗	✗
Jain et al. (2014)	optimal solution	✓	✗	✗
Bouchot et al. (2016)	true signal	✗	✓	✓
Yuan et al. (2016, Theorem 1)	optimal solution	✗	✓	✓
Yuan et al. (2016, Theorem 3)	arbitrary	✓	✗	✓
Proposed Theorem 6	arbitrary	✓	✓	✓

must be the unique global minimizer of $F(\boldsymbol{x})$ subject to a sparsity constraint (see Theorem 1 therein). Such a requirement dramatically excludes many popular and simple choices of $F(\boldsymbol{x})$. For example, let us again examine the compressed sensing problem. With the presence of noise, it is almost impossible for $\bar{\boldsymbol{x}}$ to be the global optimum of $F(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2$. Hence, one cannot apply the theoretical result of Yuan et al. (2016) to justify the performance of HTP. In comparison, our theorem ensures that support recovery is possible as far as the selected $F(\boldsymbol{x})$ fulfills the condition (3). Though Theorem 3 in Yuan et al. (2016) does not assume the RIP condition or the optimality of $\bar{\boldsymbol{x}}$ with respect to $F(\boldsymbol{x})$, it requires a relaxed sparsity parameter $k = \mathcal{O}\left(\kappa^2 s\right)$, whereas the proposed Theorem 6 asserts that $k = s$ suffices. We also note that iteration complexity was not provided by Yuan et al. (2016) in the relaxed sparsity case, whereas we clearly state the dependency on all the parameters.

Compared with Bouchot et al. (2016), it is not hard to see that the problem considered here is more general, since we aim to recover an arbitrary sparse signal while they targeted the true parameter of compressed sensing. Bouchot et al. (2016) also imposed the RIP condition that is not invoked here. Jain et al. (2011; 2014) presented HTP-style algorithms with analysis on parameter estimation, but a guarantee on support recovery was not considered. We summarize the comparison in Table 1.

Weakness. We remark that though Theorem 6 is free of the RIP condition and the relaxed sparsity, it implicitly requires that HTP should not terminate too early. Otherwise, HTP may fail to recover the support. We believe that it is a very interesting future direction to give a lower bound on the iteration complexity of HTP. In the sequel, we strengthen our result by providing sufficient conditions which prevent HTP from early stopping.

In particular, we move on to the practical scenario where the results to be established also apply to the exact case. As a reminder, due to the assumption (*A*1), (HTP3) is virtually solving a convex program. Yet, since $F(\boldsymbol{x})$ is a general function, (HTP3) can only be solved approximately by, e.g., gradient descent (Nesterov, 2004), stochastic gradient descent (Bottou & Bousquet, 2007), or the more recent variance reduced variant (Johnson & Zhang, 2013). The question to ask is, whether support recovery is possible under such a "noisy" setting, and how the optimization accuracy ϵ enters the conditions for this end.

The following theorem presents an affirmative answer, though the RIP condition is assumed.

Theorem 7. *Consider the HTP algorithm with ϵ-approximate solutions in (HTP3). Assume (A1) and (A2). Suppose that the condition number $\kappa < 1.25$. Pick $\eta = \eta'/M$ with $\kappa - 0.25 < \eta' < 1$ and set $k = s$ in (HTP2). Then HTP recovers the support of $\bar{\boldsymbol{x}}$ using at most*

$$t_{\max} = \left(\frac{\log \kappa}{\log(1/\mu)} + \frac{\log(\sqrt{2}/(1-\lambda))}{\log(1/\mu)} + 2 \right) \|\bar{\boldsymbol{x}}\|_0$$

iterations, provided that for some constant $\lambda \in (0, 1)$

$$\bar{\boldsymbol{x}}_{\min} \geq \frac{\sqrt{2} + 3\sqrt{2}\kappa}{m\lambda} \|\nabla_{k+s} F(\bar{\boldsymbol{x}})\| + \frac{4}{m\lambda}\sqrt{M\epsilon}. \tag{4}$$

Above, the quantity μ is given by

$$\mu = \sqrt{2}(\kappa - \eta') \in (0, \sqrt{2}/4).$$

Since the condition number is assumed to be well bounded, it follows that the iteration complexity is a constant multiple of the sparsity, i.e., $\mathcal{O}\left(\|\bar{\boldsymbol{x}}\|_0\right)$. By examining the $\bar{\boldsymbol{x}}_{\min}$ condition (4), we find that the optimization error ϵ does not propagate in a progressive manner. Rather, it enters the condition as an additive error. By comparing (4) to (3), the exact case, one may argue that (4) is more stringent because it requires $\bar{\boldsymbol{x}}_{\min} \geq \mathcal{O}\left(\kappa\right) \|\nabla_{k+s} F(\bar{\boldsymbol{x}})\|$ while (3) imposes $\bar{\boldsymbol{x}}_{\min} \geq \mathcal{O}\left(\sqrt{\kappa}\right) \|\nabla_{k+s} F(\bar{\boldsymbol{x}})\|$. Yet, we point out that Theorem 7 is based on the RIP condition, i.e., $\kappa < 1.25$. So it is not appropriate to examine the asymptotic behavior for the condition (4).

Finally, we study under which RIP-free conditions can HTP guarantee support recovery in the face of approximate solutions. We have the following result.

Theorem 8. *Consider the HTP algorithm with ϵ-approximate solutions in (HTP3). Assume (A1) and (A2). Pick $\eta < 1/M$ and let $k \geq 2s + \frac{8s}{\eta^2 m^2}$ in (HTP2). Then HTP recovers the support of $\bar{\boldsymbol{x}}$ using at most*

$$t_{\max} = \left(\frac{3 \log \kappa}{\log(1/\mu)} + \frac{4 \log(\sqrt{2}/(1-\lambda))}{\log(1/\mu)} + 2 \right) \|\bar{\boldsymbol{x}}\|_0$$

iterations, provided that for some constant $\lambda \in (0,1)$

$$\begin{aligned} \bar{\boldsymbol{x}}_{\min} \geq & \frac{2\sqrt{2} + \sqrt{\kappa}}{m\lambda} \|\nabla_{k+s} F(\bar{\boldsymbol{x}})\| \\ & + \lambda^{-1} \left(\sqrt{\frac{2}{m(1-\mu)}} + \sqrt{\frac{2}{m}\kappa} \right) \sqrt{\epsilon}. \end{aligned} \tag{5}$$

Above, the quantity μ is given by

$$\mu = 1 - \frac{\eta m(1-\eta M)}{2} \in (0,1).$$

To be clear, due to sparsity relaxation, Theorem 8 only ensures support inclusion, i.e., $S \subset S^{t_{\max}}$. In Yuan et al. (2016), they showed that under the condition

$$\bar{\boldsymbol{x}}_{\min} > 1.62 \sqrt{\frac{2(F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^*))}{m}},$$

HTP terminates with output $\boldsymbol{x}^t$ satisfying $\operatorname{supp}(\boldsymbol{x}^t, s) = S$. However, the iteration number t was not given. Either, it is not clear how large the difference $F(\bar{\boldsymbol{x}}) - F(\boldsymbol{x}^*)$ is, where $\boldsymbol{x}^*$ is the global s-sparse minimizer of $F(\boldsymbol{x})$ and we recall that $\bar{\boldsymbol{x}}$ is an arbitrary signal.

In contrast to Theorem 7, the quantity $\sqrt{\epsilon}$ here is multiplied by the condition number κ, which will consume more computational resources in order to fulfill the condition. This is not surprising because enlarging the support increases the chance of detecting the support but as a price, it also introduces more noise. Fortunately, under the RSC and RSS assumptions, first order solvers converges linearly. For instance, after $\mathcal{O}(\kappa \log(1/\epsilon))$ steps, gradient descent guarantees an ϵ-approximate solution.

In view of the existing results from convex optimization (Nesterov, 2004), together with Theorem 8, we can show that the total computational complexity of HTP is

$$\left(d + \kappa^2 s \log d + \kappa^3 s \log(1/\epsilon)\right) s\kappa \log \kappa. \tag{6}$$

To see this, note that (HTP1) consumes $\mathcal{O}(d)$ operations and (HTP2) costs $\mathcal{O}(k \log d)$. Using gradient descent to solve (HTP3) results in a complexity $\mathcal{O}(k\kappa \log(1/\epsilon))$. Combining them together and noting $k = \mathcal{O}(\kappa^2 s)$, we obtain the above.

We point out that though Theorem 6 and Theorem 7 need to know the sparsity s, one can set k to be a quantity smaller than s. In this case, it follows from our analysis that HTP recovers the support of the top-k elements. Interested readers may refer to Lemma 19 for more details.

4. Statistical Results

In this section, we relate our main results, Theorem 6 to Theorem 8, to concrete statistical models. In particular, we study two prevalent models: the sparse linear regression and the sparse logistic regression.

The sparse linear regression model is in essence the one considered in the compressed sensing community. It assumes that the given response vector $\boldsymbol{y}$ obeys $\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{e}$, for a known design matrix $\boldsymbol{A}$, a true sparse parameter $\bar{\boldsymbol{x}}$ (to be estimated) and an unknown noise $\boldsymbol{e}$. In order to estimate the signal $\bar{\boldsymbol{x}}$, many researchers (e.g., Jain et al. (2014)) considered the following formulation:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} F(\boldsymbol{x}) := \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2, \text{ s.t. } \|\boldsymbol{x}\|_0 \leq s,$$

and attempted to prove that the (near) optimal solution of the above program is close enough to $\bar{\boldsymbol{x}}$. Yet, it turns out that we can use more flexible functions $F(\boldsymbol{x})$, e.g., $F(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|^2 + \alpha \|\boldsymbol{x}\|^2$. To see this, by standard results (e.g., Vershynin (2010); Shen & Li (2016)), we are guaranteed that when the entries of $\boldsymbol{A}$ and those of $\boldsymbol{e}$ are i.i.d. sub-gaussian,

$$\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\| \leq \mathcal{O}\left(\sqrt{N^{-1}(k+s)\log d}\right) + \alpha \|\bar{\boldsymbol{x}}\|$$

holds with high probability, where N is the sample size. Hence, by picking $\alpha = \mathcal{O}\left(\sqrt{N^{-1}(k+s)\log d}\right)$, we have $\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\|$ vanishes as N increases. In light of such an observation and our theorems (specifically the $\bar{\boldsymbol{x}}_{\min}$ conditions), we find that it is not the sparsity-constrained program matters. Rather, it is a properly chosen $F(\boldsymbol{x})$ that guides HTP to the target signal.

The logistic regression model is used for binary classification. It has been shown in a number of work (see, e.g., Yuan et al. (2014)) that $\|\nabla_{k+s} F(\bar{\boldsymbol{x}})\|$ is bounded from above by $\mathcal{O}\left(\sqrt{N^{-1}(k+s)\log d}\right)$ with high probability, assuming the data is i.i.d. sub-gaussian. Again, we can add an ℓ_2 regularizer to the logistic loss to make it strongly convex, without loss of the support recovery guarantee.

Relating these statistical results to our theorems, we conclude that the $\bar{\boldsymbol{x}}_{\min}$ conditions involved can be satisfied with high probability as soon as the sample size N grows with $(k+s)\log d$. Moreover, under the same conditions, the condition number κ is well bounded from above, say $\kappa < 9$, implying a constant iteration complexity $\mathcal{O}(\|\bar{\boldsymbol{x}}\|_0)$ and a fast computation (see the complexity in (6)). We also remark that in light of the many more choices of $F(\boldsymbol{x})$, the function $F(\boldsymbol{x})$ essentially acts as a proxy that guides HTP to the target signal, rather than an objective function being optimized by HTP.

5. Proof Sketch

Our main results, Theorem 6 to Theorem 8, are proved by mathematical induction. The key idea is partitioning the support set S into several disjoint subsets S_1, S_2, …, S_K according to the magnitude of the elements (Zhang, 2011; Bouchot et al., 2016). Then we show that after a few iterations, say n_1, HTP identifies the first subset, i.e., $S_1 \subset S^{n_1}$. Given this, we further examine how many iterations are needed to include the first two subsets. And we inductively show that after $n_1 + n_2 \cdots + n_i$ steps, the support set produced by HTP contains the first i number of subsets, i.e., $S_1 \cup S_2 \cdots \cup S_i \subset S^{n_1+n_2\cdots+n_i}$. We then show that each n_i is small, and the sum of them is upper bounded by a multiple of $\|\bar{\boldsymbol{x}}\|_0$. Hence, two components are important to this end. First, we need to construct the subsets properly, and second, we need to offer an estimate on the n_i's which should be small enough.

Without loss of generality, suppose that the elements of $\bar{\boldsymbol{x}}$ are arranged in descending order. Then each subset S_i is inductively constructed as follows:

$$S_i = \{s_{i-1}+1, \dots, s_i\},\ 1 \leq i \leq K,$$

where $s_0 = 0$ and for all $1 \leq i \leq K$, s_i is defined as the largest index such that

$$|\bar{\boldsymbol{x}}_{s_i}| > \frac{1}{\sqrt{2}} \left|\bar{\boldsymbol{x}}_{s_{i-1}+1}\right|.$$

Note that the constant $1/\sqrt{2}$ can be replaced with any other quantity smaller than 1. Since s_i is the largest one, it follows that

$$|\bar{\boldsymbol{x}}_{s_i+1}| \leq \frac{1}{\sqrt{2}} \left|\bar{\boldsymbol{x}}_{s_{i-1}+1}\right|,$$

which immediately implies

$$\left\|\bar{\boldsymbol{x}}_{\{s_{i-1}+1,\dots,s\}}\right\|^2 \leq 2(\bar{\boldsymbol{x}}_{s_i})^2 S_{i:K},$$

where

$$S_{i:K} := \sum_{j=0}^{K-i} 2^{-j} \left|S_{i+j}\right|.$$

Then we show that given the above and the condition $S_1 \cup S_2 \cdots \cup S_{i-1} \subset S^{n_1+n_2\cdots+n_{i-1}}$, as soon as HTP decreases the distance to $\bar{\boldsymbol{x}}$ with a geometric rate (which is the theme of Section 2), we are guaranteed that $S_1 \cup S_2 \cdots \cup S_i \subset S^{n_1+n_2\cdots+n_i}$. Here, n_i is given by

$$|\bar{\boldsymbol{x}}_{s_i}| > \alpha \cdot \beta^{n_i} \sqrt{S_{i:K}} + \theta,$$

for some parameters α, β and θ. Now assuming $\theta < \bar{\boldsymbol{x}}_{\min} \leq |\bar{\boldsymbol{x}}_{s_i}|$ implies that n_i is as small as the logarithm of $S_{i:K}$. Thus,

$$t_{\max} = \sum_{i=1}^{K} n_i \leq \sum_{i=1}^{K} \log S_{i:K} \leq K \log \frac{1}{K} \sum_{i=1}^{K} S_{i:K}.$$

The result follows by doing some calculation on the sum of $S_{i:K}$'s. See the full proof in the supplementary file.

6. Numerical Study

The HTP algorithm has been studied for several years and has found plenty of successful applications. There is also a large volume of empirical study, e.g., Bouchot et al. (2016), showing that HTP performs better in terms of computational efficiency and parameter estimation than compressive sampling matching pursuit (Needell & Tropp, 2009), subspace pursuit (Dai & Milenkovic, 2009), iterative hard thresholding (Blumensath & Davies, 2009), to name a few. Hence, the focus of our numerical study is to verify the theoretical findings in Section 3.

Data. In order to investigate the performance of HTP with both the exact and inexact solutions, we consider the linear regression model $\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + \sigma \boldsymbol{e}$, where $\bar{\boldsymbol{x}}$ is a 100-dimensional vector with a tunable sparsity s. The elements in the design matrix $\boldsymbol{A}$ and the noise $\boldsymbol{e}$ are i.i.d. normal variables. The response $\boldsymbol{y}$ is an N-dimensional vector. For a certain sparsity level s, the support of $\bar{\boldsymbol{x}}$ is chosen uniformly and the non-zero components of $\bar{\boldsymbol{x}}$ are i.i.d. normal variables. If not specified, we set $N = 100$ and $\sigma = 0.01$.

Evaluation metric. In the experiments, we are mainly interested in examining the percentage of successful support recovery and the iteration number that guarantees it. We mark a trial as success if before HTP terminates, there is a solution $\boldsymbol{x}^t$ satisfying $\mathrm{supp}(\boldsymbol{x}^t) = \mathrm{supp}(\bar{\boldsymbol{x}})$. Otherwise, we mark it as failure. The iteration number is counted only for those success trials and we report the averaged result.

Solvers. We choose the least-squares loss as the proxy function $F(\boldsymbol{x})$, for which an exact solution can be computed in (HTP3). We also implement the gradient descent (GD) algorithm to approximately solve (HTP3). In order to produce solutions with different accuracy ϵ, we run the GD algorithm with a various number of gradient oracle calls. In this way, we are able to examine how ϵ affects support recovery through the number of oracle calls.

Other settings. The step size η in HTP is fixed as $\eta = 1$. We use the true sparsity for the sparsity parameter k in (HTP2). For each configuration of sparsity, we generate 100 independent copies of $\bar{\boldsymbol{x}}$. Hence, all the experiments are performed with 100 trials.

A notable aspect of our theoretical results is that after $\mathcal{O}(s\kappa \log \kappa)$ iterations, HTP captures the support. For the purpose of justification, we vary the sparsity s from 1 to 50,

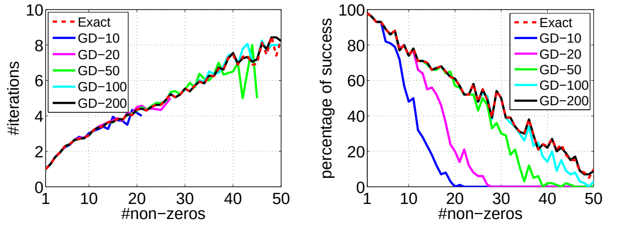

Figure 1. **Iteration number and percentage of success against the sparsity.** The number of measurements $N = 100$. GD–"T" means we run the gradient descent algorithm for T steps. As predicted by our theorem, the iteration number is nearly proportional to the sparsity (left panel). Note that using approximate solutions does not affect the iteration complexity. From the right panel, we observe that gradient descent with 50 steps already ensures comparable performance to the exact solution, possibly due to the geometric convergence rate of gradient descent.

and plot the curve of the iteration number used to identify the support against the true sparsity s. Note that we use the same design matrix for all trials, hence a fixed condition number κ. The result is recorded in the left panel of Figure 1. As predicted by our theorem, the iteration number is (almost) linear with the sparsity. Interestingly, we also find that HTP uses far fewer steps than expected. For example, to recover the support of a 20-sparse signal, 4 iterations suffice in average, suggesting possible improvement of our theorems in special cases. Also note that for a given sparsity level, applying an inexact solver for (HTP3) does not increase the iteration number of HTP. This is not surprising since our theorem states that the optimization error in (HTP3) only enters the $\bar{\boldsymbol{x}}_{\min}$ condition. In other words, it only affects the percentage of success as shown in the right panel of Figure 1. Thanks to the linear convergence of gradient descent, it turns out that using 50 calls of gradient oracle guarantees an appealing performance.

Next, we tune the number of measurements N from 1 to 100, and study the support recovery performance against the choice of N. Here, the sparsity level s is fixed to $s = 5$. With the sub-gaussian design, standard result shows that the condition number can be upper bounded by $(\mathrm{C}_1 N + s \log d)/(\mathrm{C}_2 N - s \log d)$. See, for example, Jain et al. (2014). This indicates that the condition number is inversely proportional to N after a proper shifting, and hence the iteration number. The curves on the left panel of Figure 2 matches our assertion. In the right panel, a phase transition emerges (Donoho & Tanner, 2010). That is, above a certain threshold (here the threshold is 20), support recovery is guaranteed with high probability while below that threshold, we have no hope to estimate the signal. We also find that when sufficient measurements are available, running GD with 10 gradient oracle calls already brings desirable performance.

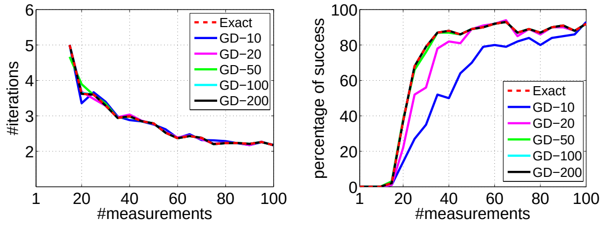

Figure 2. **Iteration number and percentage of success against the number of measurements.** The sparsity $s = 5$. GD–"T" means we run the gradient descent algorithms for T steps. The left panel shows that the more measurements we have, the faster we detect the support. The rationale is that the condition number becomes smaller with additional measurements, and by our theorem, we need fewer iterations. The right panel shows a phase transition phenomenon: when we have 20 or more measurements, HTP guarantees support recovery with high probability while support recovery is impossible if we do not have sufficient samples. Again, running GD with 50 gradient oracle calls produces similar result with the exact solution.

We remind that in Figure 1 and Figure 2, some values of #iterations are not plotted. For example, we do not have the iteration number for GD–50 in Figure 1 when $s \geq 45$. This is simply because all the trials are marked as failure. See the associated percentage of success curve.

Now let us return to the $\bar{\boldsymbol{x}}_{\min}$ condition of Theorem 8, i.e., Eq. (5). From Figure 1 and Figure 2, we conclude that as far as the optimization error is small enough, HTP with inexact iterates behaves comparably to that with exact solutions. For example, the "GD–200" curve (black solid) and the "Exact" curve (red dashed) in these two figures actually lie on top of each other even the RIP condition is not met (small N or large s). This suggests that the relaxed sparsity condition in Theorem 8 may not be vital.

7. Conclusion and Future Work

In this paper, we have studied the iteration complexity of the hard thresholding pursuit algorithm for recovering the support of an arbitrary s-sparse signal. We have shown that if the iterates of HTP are exact solutions, HTP recovers the support within $\mathcal{O}(s\kappa \log \kappa)$ iterations where κ is the condition number. In a more practical machine learning setting, we have proved that even with inexact solutions, support recovery is still possible with the same iteration bound. We have also investigated two popular statistical models, and have established probabilistic arguments under the standard sub-gaussian design. The numerical study has confirmed the correctness of our theoretical findings.

Orthogonal to the present work, an interesting direction for future study is establishing a lower bound on the iteration complexity of HTP for support recovery. It is also interesting to investigate the performance on realistic datasets.

Acknowledgements

The work is supported in part by NSF-Bigdata-1419210 and NSF-III-1360971. We thank the anonymous reviewers for pointing out several related work and for suggesting improvement on the proof sketch and the experiments. We also thank Jian Wang and Jing Wang for valuable discussion on the work.

References

Agarwal, Alekh, Negahban, Sahand, and Wainwright, Martin J. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 2012.

Blumensath, Thomas and Davies, Mike E. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

Bottou, Léon and Bousquet, Olivier. The tradeoffs of large scale learning. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems*, pp. 161–168, 2007.

Bouchot, Jean-Luc, Foucart, Simon, and Hitczenko, Pawel. Hard thresholding pursuit algorithms: number of iterations. *Applied and Computational Harmonic Analysis*, 41(2):412–435, 2016.

Cai, Tony T., Wang, Lie, and Xu, Guangwu. New bounds for restricted isometry constants. *IEEE Trans. Information Theory*, 56(9):4388–4394, 2010.

Candès, Emmanuel J. and Tao, Terence. Decoding by linear programming. *IEEE Trans. Information Theory*, 51(12): 4203–4215, 2005.

Chen, Scott Shaobing, Donoho, David L., and Saunders, Michael A. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

Dai, Wei and Milenkovic, Olgica. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. Information Theory*, 55(5):2230–2249, 2009.

Daubechies, Ingrid, Defrise, Michel, and Mol, Christine De. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

Donoho, David L. Compressed sensing. *IEEE Trans. Information Theory*, 52(4):1289–1306, 2006.

Donoho, David L. and Tanner, Jared. Precise undersampling theorems. *Proceedings of the IEEE*, 98(6):913–924, 2010.

Foucart, Simon. Hard thresholding pursuit: An algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

Jain, Prateek, Tewari, Ambuj, and Dhillon, Inderjit S. Orthogonal matching pursuit with replacement. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems*, pp. 1215–1223, 2011.

Jain, Prateek, Tewari, Ambuj, and Kar, Purushottam. On iterative hard thresholding methods for high-dimensional M-estimation. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pp. 685–693, 2014.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pp. 315–323, 2013.

Loh, Po-Ling and Wainwright, Martin J. Support recovery without incoherence: A case for nonconvex regularization. *CoRR*, abs/1412.5632, 2014.

Needell, Deanna and Tropp, Joel A. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3): 301–321, 2009.

Negahban, Sahand, Ravikumar, Pradeep, Wainwright, Martin J., and Yu, Bin. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pp. 1348–1356, 2009.

Nesterov, Yurii. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

Nguyen, Nam H. and Tran, Trac D. Robust lasso with missing and grossly corrupted observations. *IEEE Trans. Information Theory*, 59(4):2036–2058, 2013.

Nguyen, Nam H., Needell, Deanna, and Woolf, Tina. Linear convergence of stochastic iterative greedy algorithms with sparse constraints. *CoRR*, abs/1407.0088, 2014.

Osher, Stanley, Ruan, Feng, Xiong, Jiechao, Yao, Yuan, and Yin, Wotao. Sparse recovery via differential inclusions. *Applied and Computational Harmonic Analysis*, 41(2):436–469, 2016.

Pati, Yagyensh C., Rezaiifar, Ramin, and Krishnaprasad, Perinkulam S. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pp. 40–44. IEEE, 1993.

Shen, Jie and Li, Ping. A tight bound of hard thresholding. *CoRR*, abs/1605.01656, 2016.

Tibshirani, Robert. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, pp. 267–288, 1996.

Tropp, Joel A. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Information Theory*, 50(10):2231–2242, 2004.

Tropp, Joel A. and Gilbert, Anna C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Information Theory*, 53(12):4655–4666, 2007.

Vershynin, Roman. Introduction to the non-asymptotic analysis of random matrices. *CoRR*, abs/1011.3027, 2010.

Wainwright, Martin J. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Trans. Information Theory*, 55(5):2183–2202, 2009.

Wang, Jian, Kwon, Suhyuk, Li, Ping, and Shim, Byonghyo. Recovery of sparse signals via generalized orthogonal matching pursuit: A new analysis. *IEEE Trans. Signal Processing*, 64(4):1076–1089, 2016.

Yuan, Ming and Lin, Yi. On the non-negative garrotte estimator. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):143–161, 2007.

Yuan, Xiao-Tong, Li, Ping, and Zhang, Tong. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 127–135, 2014.

Yuan, Xiao-Tong, Li, Ping, and Zhang, Tong. Exact recovery of hard thresholding pursuit. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pp. 3558–3566, 2016.

Zhang, Tong. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009.

Zhang, Tong. Sparse recovery with orthogonal matching pursuit under RIP. *IEEE Trans. Information Theory*, 57 (9):6215–6221, 2011.

Zhao, Peng and Yu, Bin. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.