Sparse Convex Optimization via Adaptively Regularized Hard Thresholding

Kyriakos Axiotis [1] **Maxim Sviridenko** [2]

Abstract

The goal of *Sparse Convex Optimization* is to optimize a convex function f under a sparsity constraint $s \leq s^*\gamma$, where s^* is the target number of non-zero entries in a feasible solution (sparsity) and $\gamma \geq 1$ is an approximation factor. There has been a lot of work to analyze the sparsity guarantees of various algorithms (LASSO, Orthogonal Matching Pursuit (OMP), Iterative Hard Thresholding (IHT)) in terms of the *Restricted Condition Number* κ. The best known algorithms guarantee to find an approximate solution of value $f(x^*)+\epsilon$ with the sparsity bound of $\gamma = O\left(\kappa \min\left\{\log \frac{f(x^0)-f(x^*)}{\epsilon}, \kappa\right\}\right)$, where x^* is the target solution. We present a new *Adaptively Regularized Hard Thresholding (ARHT)* algorithm that makes significant progress on this problem by bringing the bound down to $\gamma = O(\kappa)$, which has been shown to be tight for a general class of algorithms including LASSO, OMP, and IHT. This is achieved without significant sacrifice in the runtime efficiency compared to the fastest known algorithms. We also provide a new analysis of OMP with Replacement (OMPR) for general f, under the condition $s > s^* \frac{\kappa^2}{4}$, which yields Compressed Sensing bounds under the Restricted Isometry Property (RIP). When compared to other Compressed Sensing approaches, it has the advantage of providing a strong tradeoff between the RIP condition and the solution sparsity, while working for any general function f that meets the RIP condition.

1. Introduction

Sparse Convex Optimization is the problem of optimizing a convex objective, while constraining the sparsity of the solution (its number of non-zero entries). Variants and special cases of this problem have been studied for many years, and there have been countless applications in Machine Learning, Signal Processing, and Statistics. In Machine Learning it is used to regularize models by enforcing parameter sparsity, since a sparse set of parameters often leads to better model generalization. Furthermore, in a lot of large scale applications the number of parameters of a trained model is a significant factor in computational efficiency, thus improved sparsity can lead to improved time and memory performance. In applied statistics, a single extra feature translates to a real cost from increasing the number of samples. In Compressed Sensing, finding a sparse solution to a Linear Regression problem can be used to significantly reduce the sample size for the recovery of a target signal. In the context of these applications, decreasing sparsity by even a small amount while not increasing the accuracy can have a significant impact.

Sparse Optimization Given a function $f : \mathbb{R}^n \to \mathbb{R}$ and any s^*-sparse *(unknown) target solution* x^*, the Sparse Optimization problem is to find an s-sparse solution x, i.e. a solution with at most s non-zero entries, such that $f(x) \leq f(x^*) + \epsilon$ and $s \leq s^*\gamma$, where $\epsilon > 0$ is a desired accuracy and $\gamma \geq 1$ is an approximation factor for the target sparsity. Even if f is a convex function, the sparsity constraint makes this problem non-convex, and it has been shown that it is an intractable problem, even when $\gamma = O\left(2^{\log^{1-\delta} n}\right)$ and f is the Linear Regression objective (Natarajan, 1995; Foster et al., 2015). However, this worst-case behavior is not observed in practice, and so a large body of work has been devoted to the analysis of algorithms under the assumption that the *restricted condition number* $\kappa_{s+s^*} = \frac{\rho^+_{s+s^*}}{\rho^-_{s+s^*}}$ (or just $\kappa = \frac{\rho^+}{\rho^-}$) of f is bounded (Natarajan, 1995; Shalev-Shwartz et al., 2010; Zhang, 2011; Bahmani et al., 2013; Liu et al., 2014; Jain et al., 2014; Yuan et al., 2016; Shen & Li, 2017a;b; Jain et al., 2014; Somani et al., 2018). Note: Here, $\rho^+_{s+s^*}$ is the maximum smoothness constant of any restriction of f on an $(s+s^*)$-sparse subset of coordinates and $\rho^-_{s+s^*}$ is the minimum strong convexity constant of any restriction of f on an $(s+s^*)$-sparse subset of coordinates.

The first algorithm for this problem, often called *Orthogonal Matching Pursuit (OMP)* or *Greedy*, was analyzed by

[1]MIT [2]Yahoo! Research. Correspondence to: Kyriakos Axiotis <kaxiotis@mit.edu>, Maxim Sviridenko <sviri@verizonmedia.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

(Natarajan, 1995) for Linear Regression, and subsequently for general f by (Shalev-Shwartz et al., 2010), obtaining the guarantee that the sparsity of the returned solution is $O\left(s^*\kappa \log \frac{f(x^0)-f(x^*)}{\epsilon}\right)$ [1]. In applications where having low sparsity is crucial, the dependence of sparsity on the required accuracy ϵ is undesirable. The question of whether this dependence can be removed was answered positively (Shalev-Shwartz et al., 2010; Jain et al., 2014) giving a sparsity guarantee of $O(s^*\kappa^2)$. As remarked in (Shalev-Shwartz et al., 2010), this bound sacrifices the linear dependence on κ, while removing the dependence on ϵ and $f(x^0) - f(x^*)$.

Since then, there has been some work on improving these results by introducing non-trivial assumptions, such as the target solution x^* being close to globally optimal. More specifically, (Zhang, 2011) defines the *Restricted Gradient Optimal Constant (RGOC) at level* s, ζ_s (or just ζ) as the ℓ_2 norm of the top-s elements in $\nabla f(x^*)$ and analyzes an algorithm that gives sparsity $s = O\left(s^*\kappa \log\left(s^*\kappa\right)\right)$, and such that $f(x) \leq f(x^*) + O(\zeta^2/\rho^-)$. (Somani et al., 2018) strengthens this bound to $f(x) \leq f(x^*) + O(\zeta^2/\rho^+)$ with sparsity $s = O\left(s^*\kappa \log \kappa\right)$. However, this means that $f(x)$ might be much larger than $f(x^*) + \epsilon$ in general. To the best of our knowledge, no improvement has been made over the $O\left(s^* \min\left\{\kappa \frac{f(x^0)-f(x^*)}{\epsilon}, \kappa^2\right\}\right)$ bound in the general case.

Another line of work studies a maximization version of the sparse convex optimization problem as well as its generalizations for matroid constraints (Altschuler et al., 2016; Elenberg et al., 2017; Chen et al., 2018).

Sparse Solution and Support Recovery Often, as is the case in Compressed Sensing, one needs a guarantee on the closeness of the solution x to the target solution x^* in absolute terms, rather than in terms of the value of f. The goal is usually either to recover (a superset of) the target support, or to ensure that the returned solution is close to the target solution in ℓ_2 norm. The results for this problem either assume a constant upper bound on the *Restricted Isometry Property (RIP)* constant $\delta_r := \frac{\kappa_r-1}{\kappa_r+1}$ for some r (RIP-based recovery), or that x^* is close to being a global optimum (RIP-free recovery). This problem has been extensively studied and is an active research area in the vast Compressed Sensing literature. See also the survey by (Boche et al., 2015).

In the seminal papers of (Candes & Tao, 2005; Candes et al., 2006; Donoho, 2006; Candes, 2008) it was shown that for the Linear Regression problem when $\delta_{2s^*} < \sqrt{2}-1 \approx 0.41$, the LASSO algorithm (Tibshirani, 1996) can recover a solution with $\left\|x - x^*\right\|_2^2 \leq Cf(x^*)$, where C is a constant depending only on δ_{2s^*} and $f(x^*) = \frac{1}{2}\left\|Ax^* - b\right\|_2^2$ is the error of the target solution[2]. Since then, a multitude of results of similar flavor have appeared, either giving related guarantees for the LASSO algorithm while improving the RIP upper bound (Foucart & Lai, 2009; Cai et al., 2009; Foucart, 2010; Cai et al., 2010; Mo & Li, 2011; Andersson & Strömberg, 2014) which culminate in a bound of $\delta_{2s^*} < 0.6248$, or showing that similar guarantees can be obtained by greedy algorithms under more restricted RIP conditions, but that are typically faster than LASSO (Needell & Vershynin, 2009; 2010; Needell & Tropp, 2009; Blumensath & Davies, 2009; Jain et al., 2011; Foucart, 2011; 2012). See also the comprehensive surveys (Foucart & Rauhut, 2017; Mousavi et al., 2019).

(Needell & Tropp, 2009) presents a greedy algorithm called *CoSaMP* and shows that for Linear Regression it achieves a bound in the form of (Candes, 2008) while having a more efficient implementation. Their method works for the more restricted RIP upper bound of $\delta_{2s^*} < 0.025$, or $\delta_{4s^*} < 0.4782$ as improved by (Foucart & Rauhut, 2017). (Blumensath & Davies, 2009) proves that another greedy algorithm called *Iterative Hard Thresholding (IHT)* achieves a similar bound to that of CoSaMP for Linear Regression, with the condition $\delta_{3s^*} < 0.067$, which is improved to $\delta_{2s^*} < \frac{1}{3}$ by (Jain et al., 2011) and to $\delta_{3s^*} < 0.5774$ by (Foucart, 2011).

The RIP-free line of research has shown that strong results can be achieved without a RIP upper bound, given that the target solution is sufficiently close to being a global optimum. These results typically require that s is significantly larger than s^*. In particular, (Zhang, 2011) shows that if ζ is the RGOC of f it can be guaranteed that $\left\|x - x^*\right\|_2 \leq 2\sqrt{6}\frac{\zeta}{\rho^-}$ (or $(1+\sqrt{6})\frac{\zeta}{\rho^-}$ with a slightly tighter analysis). (Somani et al., 2018) strengthens this bound to $\left(1+\sqrt{1+\frac{5}{\kappa}}\right)\frac{\zeta}{\rho^-}$. Furthermore, it has been shown that as long as a "Signal-to-Noise" condition holds, one can actually recover a superset of the target support. Typically the condition is a lower bound on $|x^*_{\min}|$, the minimum magnitude non-zero entry of the target solution. Different lower bounds that have been devised include $\Omega\left(\frac{\sqrt{s+s^*}\left\|\nabla f(x^*)\right\|_\infty}{\rho^-_{s+s^*}}\right)$ (Jain et al., 2014), which was later improved to $\Omega\left(\sqrt{\frac{f(x^*)-f(\overline{x}^*)}{\rho^-_{2s}}}\right)$, where $\overline{x}^*$ is an optimal s-sparse solution (Yuan et al., 2016). Finally, (Somani et al., 2018) improves the sparsity bound to $O(s^*\kappa \log\left(s^*\kappa\right))$ in the statistical setting and (Shen & Li, 2017b) shows that the sparsity can be brought down to $s = s^* + O(\kappa^2)$ if a stronger lower bound of $\Omega\left(\sqrt{\kappa}\frac{\zeta}{\rho}\right)$ is assumed.

[1] Even though (Natarajan, 1995) states a less general result, this is what is implicitly proven.

[2] $f(x^*)$ is also commonly denoted as $\frac{1}{2}\|\eta\|_2^2$, where $Ax^* = b + \eta$, i.e. η is the measurement *noise*.

1.1. Our work

In this work we present a new algorithm called *Adaptively Regularized Hard Thresholding (ARHT)*, that closes the longstanding gap between the $O\left(s^*\kappa\frac{f(x^0)-f(x^*)}{\epsilon}\right)$ and $O\left(s^*\kappa^2\right)$ bounds by getting a sparsity of $O(s^*\kappa)$ and thus achieving the best of both worlds. As (Foster et al., 2015) shows that for a general class of algorithms (including greedy algorithms like OMP, IHT as well as LASSO) the linear dependence on κ is necessary even for the special case of Sparse Regression, our result is tight for this class of algorithms. In the supplementary material we briefly describe this example and also state a conjecture that it can be turned into an inapproximability result. Furthermore, there we also show that the $O(s^*\kappa^2)$ sparsity bound is tight for OMPR, thus highlighting the importance of regularization in our method. Our algorithm is efficient, as it requires roughly $O\left(s\log^3\frac{f(x^0)-f(x^*)}{\epsilon}\right)$ iterations, each of which includes one function minimization in a restricted support of size s and is simple to describe and implement. Furthermore, it directly implies non-trivial results in the area of Compressed Sensing.

We also provide a new analysis of OMPR (Jain et al., 2011) and show that under the condition that $s > s^*\frac{\kappa^2}{4}$, or equivalently under the RIP condition $\delta_{s+s^*} < \frac{2\sqrt{\frac{s}{s^*}}-1}{2\sqrt{\frac{s}{s^*}}+1}$, it is possible to approximately minimize the function f up to some error depending on the RIP constant and the closeness of x^* to global optimality. More specifically, we show that for any $\epsilon > 0$ OMPR returns a solution x such that

$$f(x) \le f(x^*) + \epsilon + C_1(f(x^*) - f(x^{\mathrm{opt}}))$$

where x^{opt} is the globally optimal solution, as well as

$$\|x - x^*\|_2^2 \le \epsilon + C_2(f(x^*) - f(x^{\mathrm{opt}}))$$

where C_1, C_2 are constants that only depend on $\frac{s}{s^*}$ and δ_{s+s^*}. An important feature of our approach is that it provides a meaningful tradeoff between the RIP constant upper bound and the sparsity of the solution, even when the sparsity s is arbitrarily close to s^*. In other words, one can relax the RIP condition at the expense of increasing the sparsity of the returned solution. Furthermore, our analysis applies to general functions with bounded RIP constant.

Experiments with real data suggest that ARHT and a variant of OMPR which we call *Exhaustive Local Search* achieve promising performance in recovering sparse solutions.

1.2. Comparison to previous work

Sparse Optimization Our Algorithm 2 (ARHT) returns a solution with $s = O(s^*\kappa)$ without any additional assumptions, thus significantly improving over the bound $O\left(s^*\min\left\{\kappa\frac{f(x^0)-f(x^*)}{\epsilon}, \kappa^2\right\}\right)$ that was known in previous work. This proves that neither any dependence on the required solution accuracy ϵ, nor the quadratic dependence on the condition number κ is necessary. Furthermore, no assumption on the function or the target solution is required to achieve this bound. Importantly, previous results imply that our bound is tight up to constants for a general class of algorithms, including Greedy-type algorithms and LASSO (Foster et al., 2015).

Table 1. Compressed Sensing tradeoffs implied by Theorem 3.7: Sparsity vs RIP condition

s	RIP CONDITION
s^*	$\delta_{2s^*} < 0.33$
$2s^*$	$\delta_{3s^*} < 0.47$
$3s^*$	$\delta_{4s^*} < 0.55$
$30s^*$	$\delta_{31s^*} < 0.83$

Sparse Solution Recovery In Corollary 3.5, we show that the improved guarantees of Theorem 3.3 immediately imply that ARHT gives a bound of $\|x - x^*\|_2 \le (2+\theta)\frac{\zeta}{\rho^-}$ for any $\theta > 0$, where ζ is the Restricted Gradient Optimal Constant. This improves the constant factor in front of the corresponding results of (Zhang, 2011; Somani et al., 2018).

As we saw, our Theorem 3.7 directly implies that OMPR gives an upper bound on $\|x - x^*\|_2^2$ of the same form as the RIP-based bounds in previous work, under the condition $\delta_{s+s^*} < \frac{2\sqrt{\frac{s}{s^*}}-1}{2\sqrt{\frac{s}{s^*}}+1}$. While previous results either concentrate on the case $s = s^*$, or $s \gg s^*$, our analysis provides a way to trade off increased sparsity for a more relaxed RIP bound, allowing for a whole family of RIP conditions where s is arbitrarily close to s^*. Specifically, if we set $s = s^*$ our work implies recovery for $\delta_{2s^*} < \frac{1}{3} \approx 0.33$, which matches the best known bound for any greedy algorithm (Jain et al., 2011), although it is a stricter condition than the $\delta_{2s^*} < 0.62$ required by LASSO (Foucart & Rauhut, 2017). Table 1 contains a few such RIP bounds implied by our analysis and shows that it readily surpasses the bounds for Subspace Pursuit $\delta_{3s^*} < 0.35$, CoSaMP $\delta_{4s^*} < 0.48$, and OMP $\delta_{31s^*} < 0.33$ (Jain et al., 2011; Zhang, 2011). Another important feature compared to previous work is that all our guarantees are not restricted to Linear Regression and are true for any function f, as long as it satisfies the required RIP condition, which makes the result more general.

Sparse Support Recovery Corollary 3.6 shows that as a direct consequence of our work, the condition $|x^*_{\min}| > \frac{\zeta}{\rho^-}$ suffices for our algorithm to recover a superset of the support with size $s = O(s^*\kappa)$. Compared to (Jain et al., 2014), we improve both the size of the superset, as well as the condition, since $\sqrt{s}\frac{\|\nabla f(x^*)\|_\infty}{\rho^-} \geq \sqrt{\frac{s}{s^*}}\frac{\zeta}{\rho^-} = \Omega\left(\frac{\zeta}{\rho^-}\right)$. Compared to (Shen & Li, 2017b), the bounds on the superset size are incomparable in general, but our $|x^*_{\min}|$ condition is more relaxed, since $\sqrt{\kappa}\frac{\zeta}{\rho^-} = \Omega(\frac{\zeta}{\rho^-})$. Finally, compared to (Yuan et al., 2016) we have a stricter lower bound on $|x^*_{\min}|$, but with a better bound on the superset size ($O(s^*\kappa)$ instead of $O(s^*\kappa^2)$). Although not explicitly stated, (Zhang, 2011; Somani et al., 2018) also give a similar lower bound of $\sqrt{1 + \frac{10}{\kappa}}\frac{\zeta}{\rho^-}$ which we improve by a constant factor.

Runtime discussion ARHT has the advantage of being very simple to implement in practice. The runtime of Algorithm 2 (ARHT) is comparable to that of the most efficient greedy algorithms (e.g. OMP/OMPR), as it requires a single function minimization per iteration.

Naming Conventions The algorithm that we call *Orthogonal Matching Pursuit (OMP)*, is also known as "Greedy" (Natarajan, 1995), "Fully Corrective Forward Greedy Selection" or just "Forward Selection". What we call *Orthogonal Matching Pursuit with Replacement (OMPR)* (Jain et al., 2011) is also known by various other names. It is referenced in (Shalev-Shwartz et al., 2010) as a simpler variant of their "Fully Corrective Forward Greedy Selection with Replacement" algorithm, or just Forward Selection with Replacement, or "Partial Hard Thresholding with parameter $r = 1$ ($\mathrm{PHT}(r)$ where $r = 1$)" (Jain et al., 2017) which is a generalization of Iterative Hard Thresholding. Finally, what we call *Exhaustive Local Search* is essentially a variant of "Orthogonal Least Squares" that includes replacement steps, and also appears in (Shalev-Shwartz et al., 2010) as "Fully Corrective Forward Greedy Selection with Replacement", or just "Forward Stepwise Selection with Replacement". See also (Blumensath & Davies, 2007) regarding naming conventions.

Remark 1.1. Most of the results in the literature either only apply to, or are only presented for the Linear Regression problem. Since all of our results apply to general function minimization, we present them as such.

2. Preliminaries

Remark 2.1. An addendum to this section can be found in the *Supplementary Material*.

We denote $[i] := \{1, 2, \ldots, i\}$. For any $x \in \mathbb{R}^n$ and $R \subseteq [n]$, we define $x_R \in \mathbb{R}^n$ as $(x_R)_i = \begin{cases} x_i & i \in R \\ 0 & \text{otherwise} \end{cases}$ Additionally, for any differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ with gradient $\nabla f(x)$, we will denote by $\nabla_R f(x)$ the restriction of $\nabla f(x)$ to R, i.e. $(\nabla f(x))_R$.

In Lemma 2.2 we state a standard martingale concentration inequality that we will use. See also (Chung & Lu, 2006) for more on martingales.

Lemma 2.2 (Martingale concentration inequality (Special case of Theorem 6.5 in (Chung & Lu, 2006))). Let $Y_0 = 0, Y_1, \ldots, Y_n$ be a martingale with respect to the sequence $i_1, \ldots, i_n$ such that

$$\mathrm{Var}\left(Y_k \mid i_1, \ldots, i_{k-1}\right) \leq \sigma^2$$

and

$$Y_{k-1} - Y_k \leq M$$

for all $k \in [n]$, then for any $\lambda > 0$,

$$\Pr\left[Y_n \leq -\lambda\right] \leq e^{-\lambda^2/\left(2\left(n\sigma^2 + M\lambda/3\right)\right)}$$

Definition 2.3. For any $x \in \mathbb{R}^n$, we denote the *support* of x by $\mathrm{supp}(x) = \{i \,:\, x_i \neq 0\}$

Definition 2.4 (Restricted Condition Number). Given a differentiable function f, the *Restricted ℓ_2-Smoothness (RSS)* constant, or just Restricted Smoothness constant, of f at sparsity level s is the minimum $\rho_s^+ \in \mathbb{R}$ such that

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{\rho_s^+}{2}\|y - x\|_2^2$$

for all $x, y \in \mathbb{R}^n$ with $|\mathrm{supp}(y - x)| \leq s$. Similarly, the *Restricted ℓ_2-Strong Convexity (RSC)* constant, or just Restricted Strong Convexity constant, of f at sparsity level s is the maximum $\rho_s^- \in \mathbb{R}_+$ such that

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\rho_s^-}{2}\|y - x\|_2^2$$

for any $x, y \in \mathbb{R}^n$ with $|\mathrm{supp}(y - x)| \leq s$. Given that $\rho_s^+, \rho_s^- > 0$, the *Restricted Condition Number* of f at sparsity level s is defined as $\kappa_s = \rho_s^+/\rho_s^-$. We will also make use of $\widetilde{\kappa}_s = \rho_2^+/\rho_s^-$ which is at most κ_s as long as $s \geq 2$.

Definition 2.5 (Restricted Isometry Property (RIP)). We say that a differentiable function f has the *Restricted Isometry Property* at sparsity level s if $\rho_s^+, \rho_s^- > 0$, and the *RIP constant* of f at sparsity level s is then defined as $\delta_s = \frac{\kappa_s - 1}{\kappa_s + 1}$.[3]

Definition 2.6 (Restricted Gradient Optimal Constant (RGOC)). Given a differentiable function f and a "target" solution x^*, the *Restricted Gradient Optimal Constant* (Zhang, 2011) at sparsity level s is the minimum $\zeta_s \in \mathbb{R}_+$ such that

$$|\langle \nabla f(x^*), y\rangle| \leq \zeta_s \|y\|_2$$

for all s-sparse y. Setting $y = \nabla_S f(x^*)$ for some set S with $|S| \leq s$, this implies that $\zeta_s \geq \|\nabla_S f(x^*)\|$. An alternative definition is that ζ_s is the ℓ_2 norm of the s elements of $\nabla f(x^*)$ with highest absolute value.

[3] We note that this is a more general definition than the one usually given for quadratic functions (i.e. Linear Regression).

Definition 2.7 (S-restricted minimizer)**.** Given $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $x^* \in \mathbb{R}^n$, and $S \subseteq [n]$, we will call x^* an S-restricted minimizer of f if $\text{supp}(x^*) \subseteq S$ and for all x such that $\text{supp}(x) \subseteq S$ we have $f(x^*) \leq f(x)$.

3. Theoretical Results

In this section we will call the current solution x and the target solution x^*, with respective supports S and S^* and sizes $s = |S|$ and $s^* = |S^*|$. The results in this section are stated in terms of $\widetilde{\kappa}$, but since $\widetilde{\kappa} = \frac{\rho_2^+}{\rho_{s+s^*}^-} \leq \frac{\rho_{s+s^*}^+}{\rho_{s+s^*}^-} = \kappa$, the same statements where $\widetilde{\kappa}$ is replaced by κ automatically follow.

3.1. Adaptively Regularized Hard Thresholding (ARHT)

Our algorithm is essentially a Hard Thresholding algorithm (and more specifically OMPR also known as PHT(1)) with the crucial novelty that it is applied on an adaptively regularized objective function.

Hard thresholding algorithms maintain a solution x supported on $S \subseteq [n]$, which they iteratively update by inserting new elements into the support set S and removing the same number of elements from it, in order to preserve the sparsity of x. More specifically, OMPR makes one insertion and one removal in each iteration. In order to evaluate the element i to be *inserted* into S, OMPR uses the fact that, because of smoothness, $\frac{(\nabla_i f(x))^2}{2\rho_2^+}$ is a lower bound on the decrease of $f(x)$ caused by inserting i into the support, and therefore picks i to maximize $|\nabla_i f(x)|$. Similarly, in order to evaluate the element j to be *removed* from S, OMPR uses the fact that $\frac{\rho_2^+}{2} x_j^2$ upper bounds the increase of $f(x)$ caused by setting $x_j = 0$, and therefore picks j to minimize $|x_j|$. However, the real worth of j might be as small as $\frac{\rho_2^-}{2} x_j^2$, so the upper bound can be loose by a factor of $\frac{\rho_2^+}{\rho_2^-} \geq \frac{\rho_2^+}{\rho_{s+s^*}^-} = \widetilde{\kappa}$.

We eliminate this discrepancy by running the algorithm on the regularized function $g(z) := f(z) + \frac{\rho_2^+}{2} \|z\|_2^2$. As the restricted condition number of g is now $O(1)$, the real worth of a removal candidate j matches the upper bound up to a constant factor.

However, even though g is now well conditioned, the analysis can only guarantee the quality of the solution in terms of the *original* objective f if the regularization is *not* applied on elements S^*, i.e. $\frac{\rho_2^+}{2} \left\|x_{R \setminus S^*}\right\|_2^2$ for some sufficiently large $R \subseteq [n]$; if this is the case, a solution with sparsity $O(s^* \widetilde{\kappa})$ can be recovered. Unfortunately, there is no way of knowing a priori which elements not to regularize, as this is equivalent to finding the target solution. As a result, the algorithm can get trapped in local minima, which are defined as states in which one iteration of the algorithm does not decrease $g(x)$, even though x is a suboptimal solution in terms of f (i.e. $f(x) > f(x^*)$).

The main contribution of this work is to characterize such local minima and devise a procedure that is able to successfully escape them, thus allowing x to converge to a desired solution for the original objective.

When x has significant ℓ_2^2 mass in the target support, the regularization term $\frac{\rho_2^+}{2} \|x\|_2^2$ might penalize the target solution too much, leading to a Type 2 iteration. In this case, we use random sampling to detect an element in the optimal support and unregularize it. This procedure escapes all local minima, thus leading to a bound in the total number of Type 2 iterations.

More concretely, we show that if at some iteration of the algorithm the value of $g(x)$ does not decrease sufficiently (Type 2 iteration), then roughly at least a $\frac{1}{\widetilde{\kappa}}$-fraction of the ℓ_2^2 mass of x lies in the target support S^*. We exploit this property by sampling an element i proportional to x_i^2 and removing its corresponding term from the regularizer (*unregularizing* it). We show that with constant probability this will happen at most $O(s^* \widetilde{\kappa})$ times, as after that all the elements in S^* will have been unregularized.

The core algorithm is presented in Algorithm 1. The full algorithm additionally requires some standard routines like binary search and is presented in Algorithm 2.

Let $R \subseteq [n]$ be the set of currently regularized elements. The following invariant is a crucial ingredient for bringing the sparsity from $O(s^* \widetilde{\kappa}^2)$ down to $O(s^* \widetilde{\kappa})$, and we intend to enforce it at all times. It essentially states that there will always be enough elements in the current solution that are being regularized.

Invariant 3.1.

$$|R \cap S| \geq s^* \max\{1, 8\widetilde{\kappa}\}$$

To give some intuition on this, ARHT owes its improved $\widetilde{\kappa}$ dependence on the regularizer $\frac{\rho^+}{2} \|x\|_2^2$. However, during the algorithm, some elements are being unregularized. Our analysis requires that the current solution support always contains $\Omega\left(s^* \widetilde{\kappa}\right)$ regularized elements, which is what Invariant 3.1 states.

In the following, we will let opt denote a guess on the target value $f(x^*)$. Also, x^0 will denote the initial solution, which is an S^0-restricted minimizer an arbitrary set $S^0 \subseteq [n]$ with $|S^0| = s$. In Algorithm 1, S^0 is defined explicitly as $[s]$, however in practice one might want to pick a better initial set (e.g. returned by running OMP).

Algorithm 1 ARHT core routine

function ARHT_core$(s, \text{opt}, \epsilon)$
  function to be minimized $f : \mathbb{R}^n \rightarrow \mathbb{R}$
  target sparsity s
  target value opt (current guess for the optimal value)
  target error ϵ
  Define $g_R(x) := f(x) + \frac{\rho_2^+}{2} \|x_R\|_2^2$ for all $R \subseteq [n]$.
  $R^0 \leftarrow [n]$
  $S^0 \leftarrow [s]$
  $x^0 \leftarrow \underset{\text{supp}(x) \subseteq S^0}{\text{argmin}} \; g_{R^0}(x)$
  $T = 2s \log \frac{g_{R^0}(x^0) - \min_x f(x)}{\epsilon}$ (number of iterations)
  for $t = 0 \ldots T-1$ **do**
    if $\min_{\text{supp}(x) \subseteq S^t} f(x) \le \text{opt}$ **then**
      return $\underset{\text{supp}(x) \subseteq S^t}{\text{argmin}} \; f(x)$
    end if
    $i \leftarrow \underset{i \in [n]}{\text{argmax}} \; |\nabla_i g_{R^t}(x^t)|$
    $j \leftarrow \underset{j \in S^t}{\text{argmin}} \; |x_j|$
    $S^{t+1} \leftarrow S^t \cup \{i\} \backslash \{j\}$
    $x^{t+1} \leftarrow \underset{\text{supp}(x) \subseteq S^{t+1}}{\text{argmin}} \; g_{R^t}(x)$
    if $g_{R^t}(x^t) - g_{R^t}(x^{t+1}) < \frac{1}{s}\left(g_{R^t}(x^t) - \text{opt}\right)$ **then**
      $S^{t+1} \leftarrow S^t$
      Sample $i \in R^t$ proportional to $(x_i^t)^2$
      $R^{t+1} \leftarrow R^t \backslash \{i\}$
      $x^{t+1} \leftarrow \underset{\text{supp}(x) \subseteq S^{t+1}}{\text{argmin}} \; g_{R^{t+1}}(x)$
    end if
  end for
  return x^T
end function

Algorithm 2 ARHT

function ARHT_robust$(s, \text{opt}, \epsilon, B)$
  function to be minimized $f : \mathbb{R}^n \rightarrow \mathbb{R}$
  lower bound on target value B
  $x^{\text{ret}} \leftarrow \vec{0}$
  for $z = 1 \ldots 5 \log \left(6n \log \frac{f(\vec{0}) - B}{\epsilon}\right)$ **do**
    $x \leftarrow \text{ARHT\_core}(s, \text{opt}, \epsilon)$
    if $f(x) < f(x^{\text{ret}})$ **then**
      $x^{\text{ret}} \leftarrow x$
    end if
  end for
  return x^{ret}
end function
function ARHT(s, ϵ)
  function to be minimized $f : \mathbb{R}^n \rightarrow \mathbb{R}$
  target sparsity s
  target error ϵ
  $B \leftarrow \min_x f(x)$
  $l \leftarrow B$
  $b \leftarrow \vec{0}$
  $r \leftarrow f(b)$
  while $r - l > \epsilon$ **do**
    $m \leftarrow \frac{l+r}{2}$
    $x \leftarrow \text{ARHT\_robust}(s, m, \epsilon/3, B)$
    if $f(x) > m + \epsilon/3$ **then**
      $l \leftarrow m$
    else
      $b \leftarrow x$
      $r \leftarrow f(x)$
    end if
  end while
  return b
end function

We begin with a lemma analyzing Algorithm 1, which is the core of our algorithm.

Lemma 3.2. If $s \ge s^* \max\{4\widetilde{\kappa} + 7, 12\widetilde{\kappa} + 6\}$ and $\text{opt} \ge f(x^*)$, with probability at least 0.2 $\text{ARHT\_core}(s, \text{opt}, \epsilon)$ returns an s-sparse solution x such that $f(x) \le \text{opt} + \epsilon$.

In other words, as long as $\text{opt} \ge f(x^*)$, a solution of value $\le \text{opt} + \epsilon$ will be found. As the value opt is not known a priori, we perform binary search on it, as described in Algorithm 2. Furthermore, the probability of success in the previous lemma can be boosted by repeating multiple times.

The following theorem encapsulates the main result of this section.

Theorem 3.3. Given a function f and an (unknown) s^*-sparse solution x^*, with probability at least $1 - \frac{1}{n}$ Algorithm 2 returns an s-sparse solution x with $f(x) \le f(x^*) + \epsilon$, as long as $s \ge s^* \max\{4\widetilde{\kappa} + 7, 12\widetilde{\kappa} + 6\}$. The number of iterations is $O\left(s \log^2 \frac{f(\vec{0}) - B}{\epsilon} \log\left(n \log \frac{f(\vec{0}) - B}{\epsilon}\right)\right)$ where $B = \min_x f(x)$.

The following corollary that bounds the total runtime can be immediately extracted. Note that in practice the total runtime heavily depends on the choice of f, and it can often be improved for various special cases (e.g. linear regression).

Corollary 3.4 (Theorem 3.3 runtime)**.** If we denote by G the time needed to compute ∇f and by M the time to minimize f in a restricted subset of $[n]$ of size s, the total runtime of Algorithm 2 is $O\left((G+M) s \log^2 \frac{f(\vec{0}) - B}{\epsilon} \log\left(n \log \frac{f(\vec{0}) - B}{\epsilon}\right)\right)$. If gradient descent is used for the implementation of the inner optimization problem, then $M = O\left(G \widetilde{\kappa} \log \frac{f(\vec{0}) - B}{\epsilon}\right)$ and so the total runtime can be bounded by $O\left(G s \widetilde{\kappa} \log^3 \frac{f(\vec{0}) - B}{\epsilon} \log\left(n \log \frac{f(\vec{0}) - B}{\epsilon}\right)\right)$.

As the first corollary of the above theorem, we show that it directly implies solution recovery bounds similar to those of (Zhang, 2011), while also improving the recovery bound by a constant factor.

Corollary 3.5 (Solution recovery)**.** Given a function f and an (unknown) s^*-sparse solution x^*, such that the Restricted Gradient Optimal Constant at sparsity level s is ζ, i.e.

$$|\langle \nabla f(x^*), y\rangle| \leq \zeta \left\|y\right\|_2$$

for all s-sparse y and as long as

$$s \geq s^* \max\left\{4\widetilde{\kappa}+7, 12\widetilde{\kappa}+6\right\}$$

Algorithm 2 ensures that

$$f(x) \leq f(x^*) + \epsilon$$

and

$$\left\|x - x^*\right\|_2 \leq \frac{\zeta}{\rho^-}\left(1 + \sqrt{1 + 2\epsilon\frac{\rho^-}{\zeta^2}}\right)$$

For any $\theta > 0$ and $\epsilon \leq \frac{\zeta^2}{\rho^-}\theta(1+\frac{\theta}{2})$, this implies that

$$\left\|x - x^*\right\|_2 \leq (2+\theta)\frac{\zeta}{\rho^-}$$

The next corollary shows that our Theorem 3.3 can be also used to obtain support recovery results under a "Signal-to-Noise" condition given as a lower bound to $|x^*_{\min}|$.

Corollary 3.6 (Support recovery)**.** As long as

$$s \geq s^* \max\left\{4\widetilde{\kappa}+7, 12\widetilde{\kappa}+6\right\}$$

and $|x^*_{\min}| > \frac{\zeta}{\rho^-}$, Algorithm 2 with $\epsilon < -\frac{1}{2\rho^-}\zeta^2 + \frac{\rho^-}{2}(x^*_{\min})^2$ returns a solution x with support S such that

$$S^* \subseteq S$$

3.2. Analysis of Orthogonal Matching Pursuit with Replacement (OMPR)

The OMPR algorithm was first described (under a different name) in (Shalev-Shwartz et al., 2010). It is an extension of OMP but after each iteration some element is removed from S^t so that the sparsity remains the same. It is described in Algorithm 3.

When x (with support S) and x^* (with support S^*) are clear from context, we will also define a solution

$$\widetilde{x} = \underset{\mathrm{supp}(z)\subseteq S\cup S^*}{\mathrm{argmin}}\ f(z)$$

Algorithm 3 Orthogonal Matching Pursuit with Replacement

function $\mathrm{OMPR}(s)$
  function to be minimized $f : \mathbb{R}^n \to \mathbb{R}$
  output sparsity s
  $S^0 \leftarrow [s]$
  $x^0 \leftarrow \mathrm{argmin}\left\{f(x) \mid \mathrm{supp}(x) \subseteq S^0\right\}$
  $t \leftarrow 0$
  while true do
    $i \leftarrow \mathrm{argmax}\left\{|\nabla_i f(x^t)| \mid i \in [n]\backslash S^t\right\}$
    $j \leftarrow \mathrm{argmin}\left\{|x^t_j| \mid j \in S^t\right\}$
    $S^{t+1} \leftarrow S^t \cup \{i\}\backslash\{j\}$
    $x^{t+1} \leftarrow \mathrm{argmin}\left\{f(x) \mid \mathrm{supp}(x) \subseteq S^{t+1}\right\}$
    if $f(x^{t+1}) \geq f(x^t)$ **then**
      break
    end if
    $t \leftarrow t+1$
  end while
  $T \leftarrow t$
  return x^T
end function

Furthermore, we will denote by $\overline{x}^*$ the optimal $(s+s^*)$-sparse solution, i.e.

$$\overline{x}^* = \underset{|\mathrm{supp}(z)|\leq s+s^*}{\mathrm{argmin}}\ f(z)$$

By definition, for any x, x^* the following chain of inequalities holds

$$\min_{z\in\mathbb{R}^n} f(z) \leq f(\overline{x}^*) \leq f(\widetilde{x}) \leq \min\{f(x), f(x^*)\}$$

The following theorem is the main result of this section. Its strength lies in its generality, and various useful corollaries can be directly extracted from it.

Theorem 3.7. Given a function f, an (unknown) s^*-sparse solution x^*, a desired solution sparsity level s, and error parameters $\epsilon > 0$ and $0 < \theta < 1$, Algorithm 3 returns an s-sparse solution x such that

- If $\widetilde{\kappa}\sqrt{\frac{s^*}{s}} \leq 1$, then

$$f(x) \leq f(x^*) + \epsilon$$

in $O\left(\sqrt{ss^*}\log\frac{f(x^0)-f(x^*)}{\epsilon}\right)$ iterations.

- If $1 < \widetilde{\kappa}\sqrt{\frac{s^*}{s}} < 2-\theta$, then

$$f(x) \leq f(x^*) + B$$

where

$$B = \epsilon + \frac{4(1-\theta)\left(\widetilde{\kappa}\sqrt{\frac{s^*}{s}}-1\right)}{\left(2-\widetilde{\kappa}\sqrt{\frac{s^*}{s}}-\theta\right)^2}\left(f(x^*) - f(\overline{x}^*)\right)$$

in $O\left(\frac{\sqrt{ss^*}}{\theta}\log\frac{f(x^0)-f(x^*)}{B}\right)$ iterations.

The first corollary states that in the "noiseless" case (i.e. when the target solution is globally optimal), the returned solution can reach arbitrarily close to the target solution:

Corollary 3.8 (Noiseless case)**.** If $\widetilde{\kappa}\sqrt{\frac{s^*}{s}} < 2$ and x^* is a globally optimal solution, i.e. $f(x^*) = \min_z f(z)$, Algorithm 3 returns a solution with

$$f(x) \le f(x^*) + \epsilon$$

in $O\left(\frac{\sqrt{ss^*}}{2-\widetilde{\kappa}\sqrt{\frac{s^*}{s}}}\log\frac{f(x^0)-f(x^*)}{\epsilon}\right)$ iterations.

Proof. We apply Theorem 3.7 with $\theta = \frac{1}{2}\left(2-\widetilde{\kappa}\sqrt{\frac{s^*}{s}}\right)$. □

The following result is in the usual form of sparse recovery results, which provide a bound on $\|x-x^*\|_2$ given a RIP constant upper bound. It provides a tradeoff between the RIP constant and the sparsity of the returned solution.

Corollary 3.9 (ℓ_2 solution recovery)**.** Given any parameters $\epsilon > 0$ and $0 < \theta < 1$, the returned solution x of Algorithm 3 will satisfy

$$\|x-x^*\|_2^2 \le \epsilon + C\left(f(x) - \min_z f(z)\right)$$

as long as

$$\delta_{s+s^*} < \frac{(2-\theta)\sqrt{\frac{s}{s^*}}-1}{(2-\theta)\sqrt{\frac{s}{s^*}}+1}$$

where C is a constant that depends only on θ, δ_{s+s^*}, and $\frac{s}{s^*}$.

In particular, for $s = s^*$, the above lemma implies recovery under the condition $\delta_{2s^*} < \frac{1}{3}$.

4. Experiments

Remark 4.1. More details about the datasets used for evaluation can be found in the *Supplementary material*.

In this section we evaluate the training performance of different algorithms in the tasks of Linear Regression and Logistic Regression. More specifically, for each algorithm we are interested in how the *loss* over the training set (the quality of the solution) evolves as a function of the the *sparsity* of the solution, i.e. the number of non-zeros.

The algorithms that we will consider are *LASSO*, *Orthogonal Matching Pursuit (OMP)*, *Orthogonal Matching Pursuit with Replacement (OMPR)*, *Adaptively Regularized Hard Thresholding (ARHT)* (Algorithm 2), and *Exhaustive Local Search* (which is a version of OMPR that examines *all* possible insertions/removals in each iteration). We run our experiments on publicly available regression and binary classification datasets, out of which we have presented those on which the algorithms have significantly different performance between each other. In some of the other datasets that we tested, we observed that all algorithms had similar performance. The results are presented in Figure 1 and Figure 2. Another relevant class of algorithms that we considered was ℓ_p *Appproximate Message Passing* algorithms (Donoho et al., 2009; Zheng et al., 2017). Brief experiments showed its performance in terms of sparsity for $p \le 0.5$ to be promising (on par with OMPR and ARHT although these had much faster runtimes), however a detailed comparison is left for future work.

In both types of objectives (linear and logistic) we include an intercept term, which is present in all solutions (i.e. it is always counted as $+1$ in the sparsity of the solution). For consistency, all greedy algorithms (OMPR, ARHT, Exhaustive Local Search) are initialized with the OMP solution of the same sparsity.

The experiments make it clear that Exhaustive Local Search outperforms the other algorithms. However, ARHT also has promising performance and it might be preferred because of better computational efficiency. As a general conclusion, however, both Exhaustive Local Search and ARHT offer an advantage compared to OMP and OMPR. As a limitation, we observe that ARHT has inconsistent performance in some cases, oscillating between the Exhaustive Local Search and OMPR solutions.

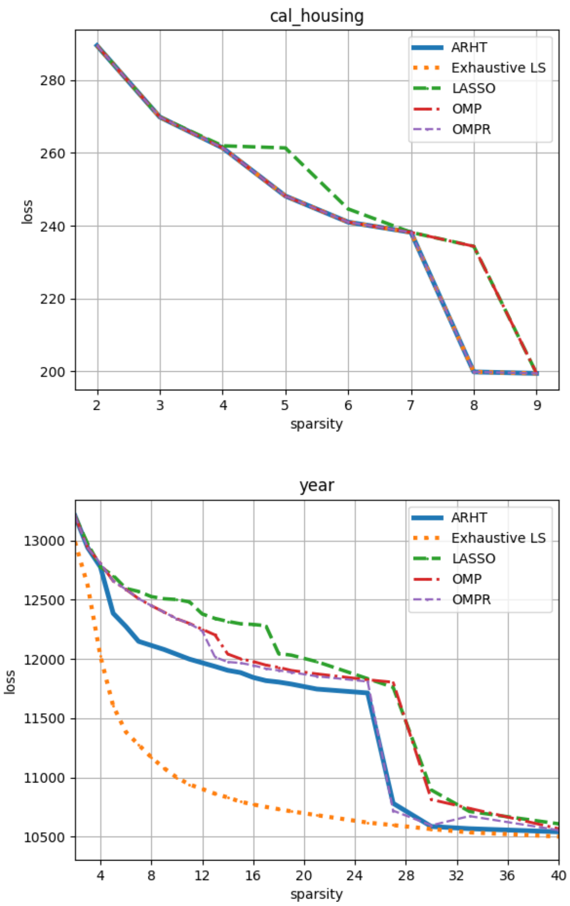

Figure 1. Comparison of different algorithms in the Regression datasets *cal_housing* and *year* using the Linear Regression loss.

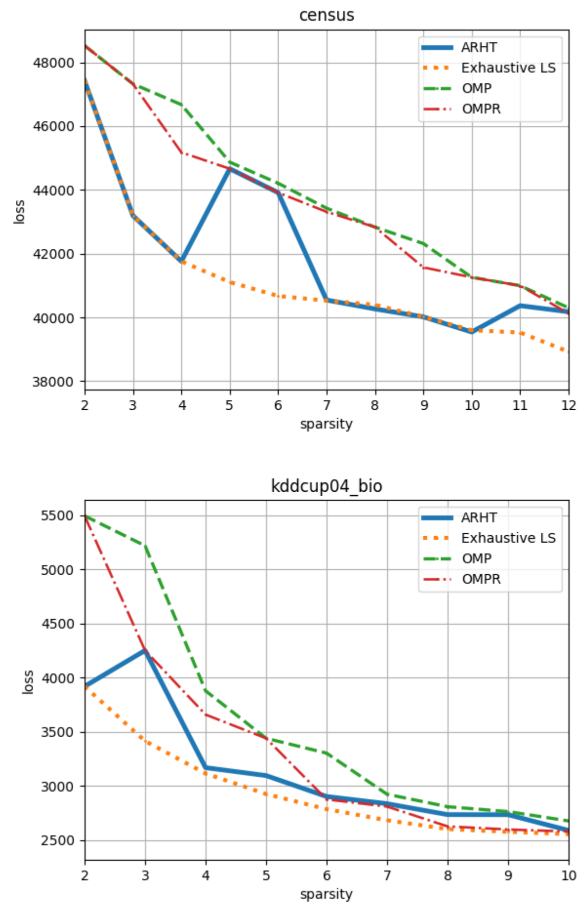

Figure 2. Comparison of different algorithms in the Binary classification datasets *census* and *kddcup04_bio* using the Logistic Regression loss.

References

Altschuler, J., Bhaskara, A., Fu, G., Mirrokni, V., Rostamizadeh, A., and Zadimoghaddam, M. Greedy column subset selection: New bounds and distributed algorithms. In *International Conference on Machine Learning*, pp. 2539–2548, 2016.

Andersson, J. and Strömberg, J.-O. On the theorem of uniform recovery of random sampling matrices. *IEEE Transactions on Information Theory*, 60(3):1700–1710, 2014.

Bahmani, S., Raj, B., and Boufounos, P. T. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14(Mar):807–841, 2013.

Blumensath, T. and Davies, M. E. On the difference between orthogonal matching pursuit and orthogonal least squares. 2007.

Blumensath, T. and Davies, M. E. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Boche, H., Calderbank, R., Kutyniok, G., and Vybíral, J. A survey of compressed sensing. In *Compressed sensing and its applications*, pp. 1–39. Springer, 2015.

Cai, T. T., Wang, L., and Xu, G. Shifting inequality and recovery of sparse signals. *IEEE Transactions on Signal processing*, 58(3):1300–1308, 2009.

Cai, T. T., Wang, L., and Xu, G. New bounds for restricted isometry constants. *IEEE Transactions on Information Theory*, 56(9):4388–4394, 2010.

Candes, E. J. The restricted isometry property and its implications for compressed sensing. *Comptes rendus mathematique*, 346(9-10):589–592, 2008.

Candes, E. J. and Tao, T. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

Candes, E. J., Romberg, J. K., and Tao, T. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

Chen, L., Feldman, M., and Karbasi, A. Weakly submodular maximization beyond cardinality constraints: Does randomization help greedy? In *International Conference on Machine Learning*, pp. 804–813, 2018.

Chung, F. and Lu, L. Concentration inequalities and martingale inequalities: a survey. *Internet Mathematics*, 3(1): 79–127, 2006.

Donoho, D. L. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

Donoho, D. L., Maleki, A., and Montanari, A. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

Elenberg, E., Dimakis, A. G., Feldman, M., and Karbasi, A. Streaming weak submodularity: Interpreting neural networks on the fly. In *Advances in Neural Information Processing Systems*, pp. 4044–4054, 2017.

Foster, D., Karloff, H., and Thaler, J. Variable selection is hard. In *Conference on Learning Theory*, pp. 696–709, 2015.

Foucart, S. A note on guaranteed sparse recovery via l1-minimization. *Applied and Computational Harmonic Analysis*, 29(1):97–103, 2010.

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

Foucart, S. Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. In *Approximation Theory XIII: San Antonio 2010*, pp. 65–77. Springer, 2012.

Foucart, S. and Lai, M.-J. Sparsest solutions of underdetermined linear systems via lq-minimization for 0¡ q¡= 1. *Applied and Computational Harmonic Analysis*, 26(3): 395–407, 2009.

Foucart, S. and Rauhut, H. A mathematical introduction to compressive sensing. *Bull. Am. Math*, 54:151–165, 2017.

Jain, P., Tewari, A., and Dhillon, I. S. Orthogonal matching pursuit with replacement. In *Advances in neural information processing systems*, pp. 1215–1223, 2011.

Jain, P., Tewari, A., and Kar, P. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, pp. 685–693, 2014.

Jain, P., Tewari, A., and Dhillon, I. S. Partial hard thresholding. *IEEE Transactions on Information Theory*, 63(5): 3029–3038, 2017.

Liu, J., Ye, J., and Fujimaki, R. Forward-backward greedy algorithms for general convex smooth functions over a cardinality constraint. In *International Conference on Machine Learning*, pp. 503–511, 2014.

Mo, Q. and Li, S. New bounds on the restricted isometry constant δ2k. *Applied and Computational Harmonic Analysis*, 31(3):460–468, 2011.

Mousavi, S., Taghiabadi, M. M. R., and Ayanzadeh, R. A survey on compressive sensing: Classical results and recent advancements. *arXiv preprint arXiv:1908.01014*, 2019.

Natarajan, B. K. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

Needell, D. and Tropp, J. A. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009.

Needell, D. and Vershynin, R. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of computational mathematics*, 9(3): 317–334, 2009.

Needell, D. and Vershynin, R. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of selected topics in signal processing*, 4(2):310–316, 2010.

Shalev-Shwartz, S., Srebro, N., and Zhang, T. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM Journal on Optimization*, 20(6): 2807–2832, 2010.

Shen, J. and Li, P. On the iteration complexity of support recovery via hard thresholding pursuit. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3115–3124. JMLR. org, 2017a.

Shen, J. and Li, P. Partial hard thresholding: Towards a principled analysis of support recovery. In *Advances in Neural Information Processing Systems*, pp. 3124–3134, 2017b.

Somani, R., Gupta, C., Jain, P., and Netrapalli, P. Support recovery for orthogonal matching pursuit: upper and lower bounds. In *Advances in Neural Information Processing Systems*, pp. 10814–10824, 2018.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Yuan, X., Li, P., and Zhang, T. Exact recovery of hard thresholding pursuit. In *Advances in Neural Information Processing Systems*, pp. 3558–3566, 2016.

Zhang, T. Sparse recovery with orthogonal matching pursuit under rip. *IEEE Transactions on Information Theory*, 57 (9):6215–6221, 2011.

Zheng, L., Maleki, A., Weng, H., Wang, X., and Long, T. Does ℓ_p-minimization outperform ℓ_1-minimization? *IEEE Transactions on Information Theory*, 63(11):6896–6935, 2017.