Journal of Machine Learning Research 22 (2021) 1-47 Submitted 6/20; Revised 2/21; Published 3/21

Sparse Convex Optimization via Adaptively Regularized Hard Thresholding

Kyriakos Axiotis KAXIOTIS@MIT.EDU
Computer Science & Artificial Intelligence Laboratory (CSAIL)
Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA

Maxim Sviridenko SVIRI@VERIZONMEDIA.COM
Yahoo! Research
770 Broadway, New York, NY 10003, USA

Editor: Prateek Jain

Abstract

The goal of *Sparse Convex Optimization* is to optimize a convex function f under a sparsity constraint $s \le s^*\gamma$, where s^* is the target number of non-zero entries in a feasible solution (sparsity) and $\gamma \ge 1$ 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.

Keywords: sparse optimization, convex optimization, compressed sensing, iterative hard thresholding, orthogonal matching pursuit, convex regularization

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.

1.1 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 $f(x) \le f(x^*) + \epsilon$ and $s \le s^*\gamma$, where $\epsilon > 0$ is a desired accuracy and $\gamma \ge 1$ 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 and 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 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 $f(x) \le f(x^*) + O(\zeta^2/\rho^-)$. Somani et al. (2018) strengthens this bound to $f(x) \le f(x^*) + O(\zeta^2/\rho^+)$ 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.

Related work. Sparse convex optimization is closely related to the problem of optimizing a convex function under a rank constraint, which is a very general optimization problem encompassing matrix completion, robust principal component analysis, and others. Close

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

analogues of OMP and OMPR that give guarantees on the rank of the solution based on the condition number have been analyzed for this setting (Shalev-Shwartz et al., 2011; Axiotis and Sviridenko, 2021).

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).

1.2 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 and 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 $\left\|x - x^*\right\|_2^2 \leq C f(x^*)$, 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 and Lai, 2009; Cai et al., 2009; Foucart, 2010; Cai et al., 2010; Mo and Li, 2011; Andersson and 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 and Vershynin, 2009, 2010; Needell and Tropp, 2009; Blumensath and Davies, 2009; Jain et al., 2011; Foucart, 2011, 2012). See also the comprehensive surveys by Foucart and Rauhut (2017); Mousavi et al. (2019).

Needell and 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 and Rauhut (2017). Blumensath and 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

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*.

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 $\Omega\left(\frac{\sqrt{s+s^*}\|\nabla f(x^*)\|_\infty}{\rho^-_{s+s^*}}\right)$ (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 $O(s^*\kappa\log(s^*\kappa))$ in the statistical setting and Shen and 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.

1.3 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 Section 5.1 we briefly describe this example and also state a conjecture that it can be turned into an inapproximability result in Conjecture 27. Furthermore, in Section 5.2 we 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

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

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

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

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.

1.4 Comparison to Previous Work

In this section we compare our results with previous work on sparse optimization, solution, and support recovery.

1.4.1 Sparse Optimization

Our Algorithm 6 (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).

1.4.2 Sparse Solution Recovery

In Corollary 20, we show that the improved guarantees of Theorem 10 immediately imply that ARHT gives a bound of $\|x - x^*\|_2 \leq (2+\theta)\frac{\zeta}{\rho^-}$ 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 23 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 and 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.

1.4.3 Sparse Support Recovery

Corollary 21 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 and Li (2017b), the bounds on the superset size are incomparable in general, but our $|x^*_{\min}|$ condition is

Table 1: Compressed sensing tradeoffs implied by Theorem 23: Sparsity vs RIP condition

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

more relaxed, since $\sqrt{\kappa}\frac{\zeta}{\rho^-} = \Omega(\frac{\zeta}{\rho^-})$. Finally, Yuan et al. (2016) works under the condition $|x^*_{\min}| > \sqrt{\frac{2(f(x^*)-\min\limits_{\|x\|_0 \le s} f(x))}{\rho^-}}$, which is more relaxed since this quantity is always in $\left[\frac{1}{\sqrt{\kappa}}\frac{\zeta}{\rho^-}, \frac{\zeta}{\rho^-}\right]$, but uses a larger superset size of $O(s^*\kappa^2)$ instead of $O(s^*\kappa)$. 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.

1.5 Runtime discussion

ARHT has the advantage of being very simple to implement in practice. The runtime of Algorithm 6 (ARHT) is comparable to that of the most efficient greedy algorithms (e.g. OMP/OMPR), as it requires a single function minimization per iteration. On the other hand, Algorithm 4 (Exhaustive Local Search) is less efficient, as it requires $O(n)$ function minimizations in each iteration, although in practice one might be able to speed it up by exploiting the fact that the problems being solved in each iteration are very closely related.

1.6 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 $r = 1$ (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 and Davies (2007) regarding naming conventions.

Remark: 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

2.1 Definitions

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. $(\nabla f(x))_R$.

Definition 1 (ℓ_p Norms) *For any $p \in \mathbb{R}_+$, we define*

$$\|x\|_p = \left(\sum_i |x_i|^p \right)^{1/p} ,$$

as well as the special cases $\|x\|_0 = |\{i \ : \ x_i \neq 0\}|$ and $\|x\|_\infty = \max_i |x_i|$.

Definition 2 *For any $x \in \mathbb{R}^n$, we denote the* support *of x by $\mathrm{supp}(x) = \{i \ : \ x_i \neq 0\}$.*

Definition 3 (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 $s \geq 2$.*

The following lemma stems from the definitions of ρ_2^+, ρ_1^+ and can be used to relate ρ_2^+ with ρ_1^+

Lemma 4 *For any function f that has the RSC property at sparsity level ≥ 2 and RSS constants ρ_1^+, ρ_2^+ at sparsity levels 1 and 2 respectively, we have $\rho_2^+ \leq 2\rho_1^+$.*

Proof For any $x, y \in \mathbb{R}^n$ such that $|\mathrm{supp}(y - x)| \leq 2$, We will prove that

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{2\rho_1^+}{2} \|y - x\|_2^2 .$$

Let $y = x + \alpha\vec{1}_i + \beta\vec{1}_j$ for some $i, j \in [n]$ and $\alpha, \beta \in \mathbb{R}$. We assume $i \neq j$ and since otherwise the claim already follows from RSS at sparsity level 1. We apply the RSS property with sparsity level 1 to get the inequalities

$$f(x + 2\alpha\vec{1}_i) \le f(x) + 2\langle \nabla f(x), \alpha\vec{1}_i \rangle + 4\frac{\rho_1^+}{2} \left\| \alpha\vec{1}_i \right\|_2^2$$

and

$$f(x + 2\beta\vec{1}_j) \le f(x) + 2\langle \nabla f(x), \beta\vec{1}_j \rangle + 4\frac{\rho_1^+}{2} \left\| \beta\vec{1}_j \right\|_2^2 .$$

Now, by using convexity (more precisely restricted convexity at sparsity level 2 that is implied by RSC) we have

$$\begin{aligned} f(y) &= f(x + \alpha\vec{1}_i + \beta\vec{1}_j) \\ &\le \frac{1}{2}\left(f(x + 2\alpha\vec{1}_i) + f(x + 2\beta\vec{1}_j) \right) \\ &\le f(x) + \langle \nabla f(x), \alpha\vec{1}_i + \beta\vec{1}_j \rangle + \frac{2\rho_1^+}{2} \left\| \alpha\vec{1}_i + \beta\vec{1}_j \right\|_2^2 \\ &= f(x) + \langle \nabla f(x), y - x \rangle + \frac{2\rho_1^+}{2} \left\| y - x \right\|_2^2 . \end{aligned}$$

■

Definition 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 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*

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

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

Definition 7 (S-restricted minimizer) *Given $f : \mathbb{R}^n \to \mathbb{R}$, $x^* \in \mathbb{R}^n$, and $S \subseteq [n]$, we will call x^* an S-restricted minimizer of f if* $\mathrm{supp}(x^*) \subseteq S$ *and for all x such that* $\mathrm{supp}(x) \subseteq S$ *we have $f(x^*) \le f(x)$.*

In Lemma 8 we state a standard martingale concentration inequality that we will use. See also Chung and Lu (2006) for more on martingales.

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

Lemma 8 (Martingale concentration inequality (Chung and Lu, 2006)) *Let* $Y_0 = 0, Y_1, \dots, Y_n$ *be a martingale with respect to the sequence* $i_1, \dots, i_n$ *such that*

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

and

$$Y_{k-1} - Y_k \le M\,,$$

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

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

2.2 Algorithms

2.2.1 ℓ_1 optimization (LASSO)

The LASSO approach is to relax the ℓ_0 constraint into an ℓ_1 one, thus solving the following optimization problem:

$$\min_x f(x) + \lambda \|x\|_1\,, \tag{1}$$

for some parameter $\lambda > 0$.

2.2.2 Iterative Hard Thresholding (IHT):

Blumensath and Davies (2009) define the hard thresholding operator $H_r(x)$ as

$$[H_r(x)]_i = \begin{cases} x_i & \text{if } |x_i| \text{ is one of the } r \text{ entries of } x \\ & \text{with largest magnitude} \\ 0 & \text{otherwise} \end{cases}.$$

Using this, the algorithm is described in Algorithm 1.

Algorithm 1 Iterative Hard Thresholding (IHT)

function IHT(s, T)
  function to be minimized $f : \mathbb{R}^n \to \mathbb{R}$
  number of iterations T
  output sparsity s
  $S^0 \leftarrow \emptyset$
  $x^0 \leftarrow \vec{0}$
  for $t = 0 \dots T-1$ **do**
    $x^{t+1} \leftarrow H_s\left(x^t - \eta \nabla f(x^t)\right)$
  end for
  return x^T
end function

2.2.3 Orthogonal Matching Pursuit (Greedy/OMP/Fwd stepwise selection)

The algorithm is described in Algorithm 2.

Algorithm 2 Greedy/OMP/Fwd stepwise selection

function greedy(s)
  function to be minimized $f : \mathbb{R}^n \to \mathbb{R}$
  output sparsity s
  $x^0 \leftarrow \vec{0}$
  for $t = 0 \dots s-1$ **do**
    $i \leftarrow \text{argmax}\left\{|\nabla_i f(x^t)| \;\middle|\; i \in [n] \backslash S^t\right\}$
    $S^{t+1} \leftarrow S^t \cup \{i\}$
    $x^{t+1} \leftarrow \text{argmin}\left\{f(x) \;\middle|\; \text{supp}(x) \subseteq S^{t+1}\right\}$
  end for
  return x^s
end function

2.2.4 Orthogonal Matching Pursuit with Replacement (Local search/OMPR/Fwd stepwise selection with replacement steps)

The algorithm is described in Algorithm 3.

Algorithm 3 Orthogonal Matching Pursuit with Replacement

function OMPR(s)
  function to be minimized $f : \mathbb{R}^n \to \mathbb{R}$
  output sparsity s
  $S^0 \leftarrow [s]$
  $x^0 \leftarrow \text{argmin}\left\{f(x) \;\middle|\; \text{supp}(x) \subseteq S^0\right\}$
  $t \leftarrow 0$
  while true do
    $i \leftarrow \text{argmax}\left\{|\nabla_i f(x^t)| \;\middle|\; i \in [n] \backslash S^t\right\}$
    $j \leftarrow \text{argmin}\left\{|x_j^t| \;\middle|\; j \in S^t\right\}$
    $S^{t+1} \leftarrow S^t \cup \{i\} \backslash \{j\}$
    $x^{t+1} \leftarrow \text{argmin}\left\{f(x) \;\middle|\; \text{supp}(x) \subseteq S^{t+1}\right\}$
    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

2.2.5 Exhaustive Local Search

The algorithm in this section is similar to OMPR, in that it iteratively inserts a new element in the support while removing one from it at the same time. While, as in OMPR, the element to be removed is the minimum magnitude entry, the one to be inserted is chosen to be the one which results in the maximum decrease in the value of the objective. It is described in Algorithm 4.

Algorithm 4 Exhaustive Local Search

function to be minimized $f : \mathbb{R}^n \to \mathbb{R}$
target sparsity s
number of iterations T
$S^0 \leftarrow [s]$
$x^0 \leftarrow \mathrm{argmin}\{f(x) \mid \mathrm{supp}(x) \subseteq S^0\}$
for $t = 0 \dots T-1$ **do**
  $j \leftarrow \underset{j \in S^t}{\mathrm{argmin}}\ x_j^2$
  $i \leftarrow \underset{i \in [n] \backslash S^t}{\mathrm{argmin}} \left\{ \underset{x\,:\,\mathrm{supp}(x) \subseteq S^t \cup \{i\} \backslash \{j\}}{\min} f(x) \right\}$
  $S^{t+1} \leftarrow S^t \cup \{i\} \backslash \{j\}$
  $x^{t+1} \leftarrow \mathrm{argmin}\{f(x) \mid \mathrm{supp}(x) \subseteq S^{t+1}\}$
  if $f(x^{t+1}) \geq f(x^t)$ **then**
    return x^t
  end if
end for
return x^T

Remark 9 *In the following sections, we will denote the minimization objective by f, the RSS and RSC parameters ρ_2^+ and $\rho_{s+s^*}^-$ by ρ^+ and ρ^- respectively, as well as $\kappa = \frac{\rho_{s+s^*}^+}{\rho_{s+s^*}^-}$ and $\widetilde{\kappa} = \frac{\rho_2^+}{\rho_{s+s^*}^-}$. Note that the use of ρ_2^+ instead of ρ_1^+ used in some works is not restrictive. As shown in Lemma 4, $\rho_2^+ \leq 2\rho_1^+$ and so in all the bounds involving $\widetilde{\kappa}$, it can be replaced by $2\frac{\rho_1^+}{\rho_{s+s^*}^-}$, thus only losing a factor of 2. Furthermore, we state our results in terms of $\widetilde{\kappa}$ as opposed to κ. This is always more general since $\widetilde{\kappa} \leq \kappa$.*

In order to simplify the exposition, we will assume that $\min_x f(x) = 0$. This property can be ensured by adding a constant to f.

When no additional context is provided, we denote current solution by x and the target solution x^*, with respective support sets S and S^* and sparsities $s = |S|$ and $s^* = |S^*|$.

3. Adaptively Regularized Hard Thresholding (ARHT)

3.1 Overview and Main Theorem

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 $g(z) := f(z) + \frac{\rho_2^+}{2}\left\|z\right\|_2^2$. 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.

The core algorithm is presented in Algorithm 5. The full algorithm additionally requires some standard routines like binary search and is presented in Algorithm 6.

In the following, we will let $\widehat{f}^*$ denote a guess on the target value $f(x^*)$. Also, x^0 will denote the initial solution, which is an S^0-restricted minimizer for an arbitrary set $S^0 \subseteq [n]$ with $|S^0| = s$. In Algorithm 5, 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).

It should be noted that even though the value ρ_2^+ is used by the algorithm to define the regularizer, exact knowledge of ρ_2^+ is not required, and an upper bound U on it can be used. Of course, the final sparsity and runtime bound will then depend on $U/\rho_{s+s^*}^-$ instead of $\rho_2^+/\rho_{s+s^*}^-$. One such upper bound is $2\rho_1^+$. For linear and logistic regression where A is the (# examples) $\times$ (# features) data matrix, ρ_1^+ is the maximum ℓ_2 norm of a column of the data matrix A (so $\rho_1^+ = 1$ if the columns of A are normalized). More generally, getting such a bound would depend on the specific function we are trying to minimize. For example, if we are trying to minimize $\sum\limits_{i=1}^{m} L(Ax)_i$ where A is a data matrix with normalized columns and $L : \mathbb{R} \to \mathbb{R}$ is a 1-smooth loss function, then $\rho_1^+ \leq 1$ so 2 is a good upper bound for ρ_2^+. Another option is to run the algorithm for different candidate values for ρ_2^+ until we get the desired performance.

We are now ready for the main result of this section. It basically states that for any solution x^* with sparsity s^*, a solution x with $f(x) \leq f(x^*) + \epsilon$ and sparsity $O(\kappa s^*)$ can be recovered by Algorithm 6. In comparison, previously known analyses could guarantee either a sparsity $O(\kappa s^* \log \frac{f(\vec{0})}{\epsilon})$ (OMP) or $O(\kappa^2 s^*)$ (OMPR, IHT, LASSO). It is useful to

Algorithm 5 Adaptively Regularized Hard Thresholding core routine

function ARHT_core($s, \widehat{f}^*, \epsilon$)
  function to be minimized $f : \mathbb{R}^n \to \mathbb{R}$
  target sparsity s
  target value $\widehat{f}^*$ (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 $R \subseteq [n]$.
  $R^0 \leftarrow [n]$
  $S^0 \leftarrow [s]$
  $x^0 \leftarrow \underset{\operatorname{supp}(x)\subseteq S^0}{\operatorname{argmin}}\ g_{R^0}(x)$
  $T = 2s \log \frac{f(\vec{0}) - \min_x f(x)}{\epsilon}$ (number of iterations)
  for $t = 0 \ldots T-1$ **do**
    if $\underset{\operatorname{supp}(x)\subseteq S^t}{\min}\ f(x) \le \widehat{f}^*$ **then**
      return $\underset{\operatorname{supp}(x)\subseteq S^t}{\operatorname{argmin}}\ f(x)$
    end if
    $i \leftarrow \underset{i\in[n]}{\operatorname{argmax}}\ |\nabla_i g_{R^t}(x^t)|$
    $j \leftarrow \underset{j\in S^t}{\operatorname{argmin}}\ |x_j|$
    $S^{t+1} \leftarrow S^t \cup \{i\} \backslash \{j\}$
    $x^{t+1} \leftarrow \underset{\operatorname{supp}(x)\subseteq S^{t+1}}{\operatorname{argmin}}\ g_{R^t}(x)$
    if $g_{R^t}(x^t) - g_{R^t}(x^{t+1}) < \frac{1}{s}\left(g_{R^t}(x^t) - \widehat{f}^*\right)$ **then**
      $S^{t+1} \leftarrow S^t$
      Sample $i \in R^t$ proportional to $(x_i^t)^2$
      $R^{t+1} \leftarrow R^t \backslash \{i\}$
      $x^{t+1} \leftarrow \underset{\operatorname{supp}(x)\subseteq S^{t+1}}{\operatorname{argmin}}\ g_{R^{t+1}}(x)$
    end if
  end for
  return x^T
end function

note here that, because of the constant factor in front of κs^* (which is less than 20), this result cannot be directly used to obtain compressed sensing results with sparsity close to s^*, but is instead useful in the asymptotic regime (with relaxed sparsity $s \gg s^*$).

Theorem 10 *Given a function f and an (unknown) s^*-sparse solution x^*, with probability at least $1 - \frac{1}{n}$ Algorithm 6 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 $O\left(s \log^2 \frac{f(\vec{0})}{\epsilon} \log\left(n \log \frac{f(\vec{0})}{\epsilon}\right)\right)$.*

Algorithm 6 Adaptively Regularized Hard Thresholding

function ARHT_robust$(s, \widehat{f}^*, \epsilon)$
  function to be minimized $f : \mathbb{R}^n \to \mathbb{R}$
  $x^{\text{ret}} \leftarrow \vec{0}$
  for $z = 1 \dots 5 \log\left(6n \log \frac{f(\vec{0})}{\epsilon}\right)$ **do**
    $x \leftarrow$ ARHT_core$(s, \widehat{f}^*, \epsilon)$
    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 $f : \mathbb{R}^n \to \mathbb{R}$
  target sparsity s
  target error ϵ
  $l \leftarrow \vec{0}$
  $r \leftarrow f(\vec{0})$
  $b \leftarrow \vec{0}$
  while $r - l > \epsilon$ **do**
    $m \leftarrow \frac{l+r}{2}$
    $x \leftarrow$ ARHT_robust$(s, m, \epsilon/3)$
    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

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 11 (Theorem 10 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 6 is $O\left((G+M)s \log^2 \frac{f(\vec{0})}{\epsilon} \log\left(n \log \frac{f(\vec{0})}{\epsilon}\right)\right)$. If gradient descent is used for the implementation of the inner optimization problem, then $M = O\left(G\widetilde{\kappa} \log \frac{f(\vec{0})}{\epsilon}\right)$ and so the total runtime can be bounded by $O\left(Gs\widetilde{\kappa} \log^3 \frac{f(\vec{0})}{\epsilon} \log\left(n \log \frac{f(\vec{0})}{\epsilon}\right)\right)$.*

Before proving the above theorem, we provide the main components that are needed for its proof. It is important to split the iterations of Algorithm 5 into two categories: Those that make enough progress, i.e. for which the condition in Line 19 of Algorithm 5 is *false*, and those that don't, i.e. for which the condition in Line 19 is *true*. We call the former *Type 1* iterations and the latter *Type 2* iterations. Intuitively, Type 1 iterations signify that $g(x)$ is decreasing at a sufficient rate to achieve the desired convergence, while Type 2 iterations indicate a local minimum that should be dealt with. Our argument consists of two steps: Showing that as long as there are enough Type 1 iterations, a desired solution will be obtained (Lemma 12), and bounding the total number of Type 2 iterations with constant probability (Lemma 13).

Lemma 12 (Convergence rate) *If Algorithm 5 executes at least* $T_1 = s \log \frac{g(x^0) - \widehat{f}^*}{\epsilon}$ *Type 1 iterations, then* $f(x^T) \leq \widehat{f}^* + \epsilon$.

The proof of this lemma can be found in Appendix A.2.

Lemma 13 (Bounding Type 2 iterations) *If* $s \geq s^* \max\{4\widetilde{\kappa} + 7, 12\widetilde{\kappa} + 6\}$ *and* $\widehat{f}^* \geq f(x^*)$*, then with probability at least* 0.2 *the number of Type 2 iterations is at most* $(s^* - 1)(4\widetilde{\kappa} + 6)$.

The proof of this lemma appears in Section 3.2. These lemmas can now be directly used to obtain the following lemma, which states the performance guarantee of the ARHT core routine (Algorithm 5).

Lemma 14 (Algorithm 5 guarantee) *If* $s \geq s^* \max\{4\widetilde{\kappa} + 7, 12\widetilde{\kappa} + 6\}$ *and* $\widehat{f}^* \geq f(x^*)$*, with probability at least* 0.2 ARHT_core$(s, \widehat{f}^*, \epsilon)$ *returns an* s*-sparse solution* x *such that* $f(x) \leq \widehat{f}^* + \epsilon$.

Proof By Lemma 13, with probability at least 0.2 there will be at most $(s^* - 1)(4\widetilde{\kappa} + 6)$ Type 2 iterations. This means that the number of Type 1 iterations is at least

$$T - (s^* - 1)(4\widetilde{\kappa} + 6) \geq s \log \frac{f(\vec{0})}{\epsilon} \geq s \log \frac{g^0(x^0) - \widehat{f}^*}{\epsilon},$$

where the latter inequality follows from the fact that $f(\vec{0}) = g^0(\vec{0}) \geq g^0(x^0)$ and $\widehat{f}^* \geq f(x^*) \geq 0$. Lemma 12 then implies that $f(x^T) \leq \widehat{f}^* + \epsilon$. ■

In other words, as long as $\widehat{f}^* \geq f(x^*)$, a solution of value $\leq \widehat{f}^* + \epsilon$ will be found. As the value $f(x^*)$ is not known a priori, $\widehat{f}^*$ is just an estimate for it. We perform binary search over $\widehat{f}^*$, as described in Algorithm 6. The reason we need this estimate of $f(x^*)$ is line 19 of Algorithm 5. As our algorithm is randomized, we use this estimate to decide whether there was enough progress in one iteration (Type 1 iteration), or not (Type 2 iteration, in which case we perform the randomization step). If $\widehat{f}^*$ is smaller than $f(x^*)$, the algorithm might mistake a Type 1 iteration for a Type 2 iteration, thus performing the randomization step even though the algorithm makes enough progress. On the other hand, if $\widehat{f}^*$ is much larger than $f(x^*)$, the algorithm might terminate with a suboptimal solution of value much larger than $f(x^*)$.

The probability of success in the previous lemma can be boosted by repeating multiple times. Combining these arguments will lead us to the proof of Theorem 10. First, we turn the result of Lemma 14 into a high probability result by repeating multiple times:

Lemma 15 *If $s \geq s^* \max\{4\widetilde{\kappa}+7, 12\widetilde{\kappa}+6\}$ and $\widehat{f}^* \geq f(x^*)$,* ARHT_robust$(s, \widehat{f}^*, \epsilon)$ *returns an s-sparse solution x such that $f(x) \leq \widehat{f}^* + \epsilon$ with probability at least $1 - \frac{1}{6n \log \frac{f(\vec{0})}{\epsilon}}$.*

Proof From Lemma 14, the probability that a given call to ARHT_core fails is at most 0.8. Since this random experiment is executed $5 \log\left(6n \log \frac{f(\vec{0})}{\epsilon}\right)$ times independently, the probability that it never succeeds is at most $(0.8)^{5\log\left(6n\log\frac{f(\vec{0})}{\epsilon}\right)} < \frac{1}{6n\log\frac{f(\vec{0})}{\epsilon}}$, therefore the statement follows. ■

Lemma 16 *If $s \geq s^* \max\{4\widetilde{\kappa}+7, 12\widetilde{\kappa}+6\}$,* ARHT$(s, \epsilon)$ *(in Algorithm 6) returns an s-sparse solution x such that $f(x) \leq f(x^*) + \epsilon$. The algorithm succeeds with probability at least $1 - \frac{1}{n}$. and the number of calls to* ARHT_robust *is $\leq 6 \log \frac{f(\vec{0})}{\epsilon}$.*

Proof First we will bound the number of calls to ARHT_robust. Let L_k be the equal to $r - l$ before the k-th iteration in Line 21 of Algorithm 6. Then either $L_{k+1} = L_k/2$ (Line 25) or $L_{k+1} \leq L_k/2 + \epsilon/3 < 5L_k/6$ (Line 28). Therefore in any case we have $L_{k+1} < 5L_k/6$ which implies that after $T = 6\log\frac{f(\vec{0})}{\epsilon}$ iterations we will have $r - l \leq \epsilon$.

Now let us compute the probability that all the calls to ARHT_robust are successful. The number of such calls is at most $6\log\frac{f(\vec{0})}{\epsilon}$ and we know each one of them independently fails with probability less than $\frac{1}{6n\log\frac{f(\vec{0})}{\epsilon}}$, so by a union bound the probability that at least one call fails is less than $\frac{1}{n}$.

To prove correctness, note that by Lemma 15, for each $r \geq f(x^*)$ we have $f(\text{ARHT\_robust}(s, r, \epsilon/3)) \leq r + \epsilon/3$. After Line 20 of Algorithm 6, we will have $l = 0 \leq f(x^*)$. In the while construct, it is always true that $f(x^*) \geq l$. This is initially true, as we saw. For each m chosen in Line 22 and x in Line 23, note that if $f(x) > m + \epsilon/3$, then by Lemma 15 $f(x^*) > m$ and so the invariant that $f(x^*) \geq l$ stays true. On the other hand, it is always true that $f(b) \leq r$. Initially this is so because $f(\vec{0}) = r$, and when we decrease r to some $f(x)$ we also update $b = x$. This implies that in the end of the algorithm the returned solution will have the required property, since we will have $f(b) \leq r \leq l + \epsilon \leq f(x^*) + \epsilon$. ■

The proof Theorem 10 now easily follows.

Proof of Theorem 10. Lemma 16 already establishes the correctness of the algorithm with probability at least $1 - \frac{1}{n}$. For the runtime, note that ARHT_core takes $O\left(s \log\frac{f(\vec{0})}{\epsilon}\right)$ iterations, ARHT_robust takes $O\left(\log\left(n\log\frac{f(\vec{0})}{\epsilon}\right)\right)$ iterations, and ARHT takes $O\left(\log\frac{f(\vec{0})}{\epsilon}\right)$ iterations. In conclusion, the total number of iterations is $O\left(s\log^2\frac{f(\vec{0})}{\epsilon}\log\left(n\log\frac{f(\vec{0})}{\epsilon}\right)\right)$, each of which requires a constant number of minimizations of f. ■

3.2 Bounding Type 2 Iterations

When x has significant ℓ_2^2 mass in the target support, the regularization term $\frac{\rho_2^+}{2}\left\|x\right\|_2^2$ 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.

When referring to the t-th iteration of Algorithm 5, we let x^t be the current solution with support set S^t and $R^t \subseteq [n]$ the current regularization set as defined in the algorithm. For ease of notation, we will drop the subscript of the regularizer, i.e. $\Phi^t(z) := \frac{\rho_2^+}{2}\left\|z_{R^t}\right\|_2^2$ and of the regularized function, i.e. $g^t(z) := f(z) + \Phi^t(z)$. Note that by definition of the algorithm x^t is an S^t-restricted minimizer of g^t.

Let $(\rho_2^+)'$ and $(\rho_{s+s^*}^-)'$ be RSS and RSC parameters of g^t. We start with a lemma that relates $(\rho_2^+)'$ to ρ_2^+ and $(\rho_{s+s^*}^-)'$ to $\rho_{s+s^*}^-$, and is proved in Appendix A.1.

Lemma 17 (RSC, RSS of regularized function) $(\rho_2^+)' \leq 2\rho_2^+$ *and* $(\rho_{s+s^*}^-)' \geq \rho_{s+s^*}^-$

This states that the restricted smoothness and strong convexity constants of the regularized function are always within a constant factor of those of the original function, and thus we can make our statements in terms of the RSC, RSS of the original function. Next, we present a lemma that establishes a lower bound on the progress $g^t(x^t) - g^{t+1}(x^{t+1})$ in one iteration. This will be helpful in order to diagnose the cause of having insufficient progress in one iteration.

Lemma 18 (ARHT Progress Lemma) *If $\widehat{f}^* \geq f(x^*)$, for the progress $g^t(x^t) - g^t(x^{t+1})$ in Line 19 of Algorithm 5 it holds that*

$$\begin{aligned}
&g^t(x^t) - g^t(x^{t+1}) \\
&\geq \frac{\rho^-}{2|S^*\backslash S^t|\rho^+}\Big(f(x^t) - f(x^*) + \langle \nabla_{S^t\backslash S^*}\Phi^t(x^t), x^t_{S^t\backslash S^*}\rangle - \frac{1}{2\rho^-}\left\|\nabla_{S^t\cap S^*}\Phi^t(x^t)\right\|_2^2\Big) - \rho^+(x_j^t)^2\,.
\end{aligned}$$

Proof The proof will proceed as follows: We first use the smoothness of g^t to get a lower bound on the progress in one step, $g^t(x^t) - g^t(x^{t+1})$. This lower bound will depend on $\|\nabla_{S^*\backslash S^t} g^t(x^t)\|_2^2$, which is the norm of the gradient of g^t restricted to the set $S^*\backslash S^t$, as well as $(x_j^t)^2$, where j is the position of the minimum-magnitude entry of x^t. Then, we use the strong convexity of g^t to relate $\|\nabla_{S^*\backslash S^t} g^t(x^t)\|_2^2$ to the difference in function value $f(x^t) - f(x^*)$, plus some terms that come from the regularizer.

First of all, since the condition in Line 12 ("if $\min\limits_{\mathrm{supp}(x)\subseteq S^t} f(x) \leq \widehat{f}^*$") was not triggered, we have that $\min\limits_{\mathrm{supp}(x)\subseteq S^t} f(x) > \widehat{f}^* \geq f(x^*)$ and so $S^*\backslash S^t \neq \emptyset$. By Lemma 17 we have that

$(\rho^+)' \leq 2\rho^+$, therefore the decrease in g^t that is achieved is

$$\begin{aligned}
& g^t(x^t) - g^t(x^{t+1}) \\
& \geq \max_{\eta \in \mathbb{R}} \left\{ g^t(x^t) - g^t(x^t + \eta \vec{1}_i - x_j^t \vec{1}_j) \right\} \\
& \geq \max_{\eta \in \mathbb{R}} \left\{ -\langle \nabla g^t(x^t), \eta \vec{1}_i - x_j^t \vec{1}_j \rangle - \rho^+ \eta^2 - \rho^+ (x_j^t)^2 \right\} := B\,.
\end{aligned}$$

Note that, as defined by the algorithm, x^t is an S^t-restricted minimizer of g^t and since $j \in S^t$, we have $\nabla_j g^t(x^t) = 0$. Therefore

$$\begin{aligned}
B = & \max_{\eta \in \mathbb{R}} \{ -\langle \nabla g^t(x^t), \eta \vec{1}_i \rangle - \rho^+ \eta^2 - \rho^+ (x_j^t)^2 \} \\
= & \frac{\left[\nabla_i g^t(x^t)\right]^2}{4\rho^+} - \rho^+ (x_j^t)^2 \\
\geq & \max_{k \in S^* \backslash S} \frac{\left[\nabla_k g^t(x^t)\right]^2}{4\rho^+} - \rho^+ (x_j^t)^2 \\
\geq & \frac{\left\|\nabla_{S^* \backslash S^t} g^t(x^t)\right\|_2^2}{4|S^* \backslash S^t| \rho^+} - \rho^+ (x_j^t)^2 \,,
\end{aligned} \tag{2}$$

where we used the fact that i was picked to maximize $\left|\nabla_k g^t(x^t)\right|$. Now we would like to relate this to $g^t(x^t) - f(x^*)$ (and not $g^t(x^t) - g^t(x^*)$). By applying the Restricted Strong Convexity property,

$$\begin{aligned}
& f(x^*) - f(x^t) \\
& \geq \langle \nabla f(x^t), x^* - x^t \rangle + \frac{\rho^-}{2} \left\| x^t - x^* \right\|_2^2 \\
& \geq \langle \nabla f(x^t), x^* - x^t \rangle + \frac{\rho^-}{2} \left\| x^*_{S^* \backslash S^t} \right\|_2^2 + \frac{\rho^-}{2} \left\| (x^t - x^*)_{S^t \cap S^*} \right\|_2^2 \,.
\end{aligned}$$

Now note that $f(x^t) = g^t(x^t) - \Phi^t(x^t)$, $\nabla_{S^t} g^t(x^t) = \vec{0}$ (since x^t is an S^t-restricted minimizer of g^t), and $\nabla \Phi^t(x^t) = \nabla_{S^t} \Phi^t(x^t)$ therefore

$$\begin{aligned}
& \langle \nabla f(x^t), x^* - x^t \rangle \\
& = \langle \nabla g^t(x^t), x^* - x^t \rangle - \langle \nabla \Phi^t(x^t), x^* - x^t \rangle \\
& = \langle \nabla g^t_{S^* \backslash S^t}(x^t), x^*_{S^* \backslash S^t} \rangle + \langle \nabla_{S^t \backslash S^*} \Phi^t(x^t), x^t_{S^t \backslash S^*} \rangle + \langle \nabla_{S^t \cap S^*} \Phi^t(x^t), (x^t - x^*)_{S^t \cap S^*} \rangle \,.
\end{aligned}$$

Plugging this into the previous inequality, we get

$$\begin{aligned}
& f(x^*) - f(x^t) \\
& \geq \langle \nabla g^t_{S^* \backslash S^t}(x^t), x^*_{S^* \backslash S^t} \rangle + \frac{\rho^-}{2} \left\| x^*_{S^* \backslash S^t} \right\|_2^2 + \langle \nabla_{S^t \backslash S^*} \Phi^t(x^t), x^t_{S^t \backslash S^*} \rangle \\
& + \langle \nabla_{S^t \cap S^*} \Phi^t(x^t), (x^t - x^*)_{S^t \cap S^*} \rangle + \frac{\rho^-}{2} \left\| (x^t - x^*)_{S^t \cap S^*} \right\|_2^2 \\
& \geq -\frac{1}{2\rho^-} \left\| \nabla_{S^* \backslash S^t} g^t(x^t) \right\|_2^2 + \langle \nabla_{S^t \backslash S^*} \Phi^t(x^t), x^t_{S^t \backslash S^*} \rangle - \frac{1}{2\rho^-} \left\| \nabla_{S^t \cap S^*} \Phi^t(x^t) \right\|_2^2 \,,
\end{aligned}$$

where we twice used the inequality $\langle u, v\rangle + \frac{\lambda}{2}\left\|v\right\|_2^2 \geq -\frac{1}{2\lambda}\left\|u\right\|_2^2$ for any $\lambda > 0$. This inequality is derived by expanding $\frac{1}{2}\left\|\frac{1}{\sqrt{\lambda}}u + \sqrt{\lambda}v\right\|_2^2 \geq 0$. So plugging in $\left\|\nabla_{S^*\backslash S^t} g^t(x^t)\right\|_2^2$ into (2),

$$B$$
$$\geq \frac{\rho^-}{2|S^*\backslash S^t|\rho^+}\left(f(x^t) - f(x^*) + \langle \nabla_{S^t\backslash S^*}\Phi^t(x^t), x^t_{S^t\backslash S^*}\rangle - \frac{1}{2\rho^-}\left\|\nabla_{S^t\cap S^*}\Phi^t(x^t)\right\|_2^2\right) - \rho^+(x^t_j)^2 .$$

■

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 19

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

To give some intuition on this, ARHT owes its improved $\widetilde{\kappa}$ dependence on the regularizer $\frac{\rho^+}{2}\left\|x\right\|_2^2$. 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 19 states.

We can now proceed to show that, with constant probability, Algorithm 5 will only have $O(s^*\widetilde{\kappa})$ Type 2 iterations, which is the goal of this section.

Proof of Lemma 13. The idea of the proof is to use the progress bound in Lemma 18 to obtain a necessary condition under which the progress (i.e. the decrease of $g^t(x^t)$) is not sufficient in one iteration (thus, we have a Type 2 iteration). For our choice of regularizer, this condition implies that at least an $\Omega(\frac{1}{\widetilde{\kappa}})$ fraction of the ℓ_2^2 mass of x^t lies in the optimal support set S^*, which means that we can find an element in S^* with decent probability by appropriately sampling an element of x^t. We finally apply a probabilistic analysis over all iterations, to show that if each sampled element is unregularized, with constant probability the total number of Type 2 iterations cannot exceed $\Theta(\widetilde{\kappa}s^*)$.

We first observe some useful properties of our regularizer, which can be verified by simple substitution. The definition of $\Phi^t(x^t)$ implies that

$$\begin{aligned}
\langle \nabla_{S^t\backslash S^*}\Phi^t(x^t), x^t_{S^t\backslash S^*}\rangle &= \rho^+\langle x^t_{(R^t\cap S^t)\backslash S^*}, x^t_{S^t\backslash S^*}\rangle \\
&= \rho^+ \sum_{i\in(R^t\cap S^t)\backslash S^*} x_i^2 \\
&= \rho^+ \sum_{i\in R^t\backslash S^*} x_i^2 \\
&= \rho^+\left\|x^t_{R^t\backslash S^*}\right\|_2^2 ,
\end{aligned} \tag{3}$$

where the second-to-last equality follows because x^t is 0 outside of S^t. For the same reason, we also have

$$\left\|\nabla_{S^t\cap S^*}\Phi^t(x^t)\right\|_2^2 = (\rho^+)^2\left\|x^t_{R^t\cap S^*}\right\|_2^2 . \tag{4}$$

By combining (3) and (4) we get that

$$\Phi^t(x^t) = \frac{1}{2}\langle \nabla_{S^t\backslash S^*}\Phi^t(x^t), x^t_{S^t\backslash S^*}\rangle + \frac{1}{2\rho^+}\left\|\nabla_{S^t\cap S^*}\Phi^t(x^t)\right\|_2^2 . \tag{5}$$

Equations (3),(4), and (5) will be used later on. Now, before the first iteration we have $\left|R^0 \cap S^0\right| = \left|S^0\right| = s$. Since in each Type 2 iteration we have $\left|R^{t+1}\right| = \left|R^t\right| - 1$,

$$\left|R^t \cap S^t\right| \geq s - [\text{number of Type 2 iterations up to } t] .$$

This implies that for the first $(s^*-1)(4\widetilde{\kappa}+6)$ Type 2 iterations,

$$|R^t \cap S^t| \geq s - (s^*-1)(4\widetilde{\kappa}+6) \geq s^* \max\{1, 8\widetilde{\kappa}\} , \tag{6}$$

since $s \geq s^* \max\{4\widetilde{\kappa}+7, 12\widetilde{\kappa}+6\}$. From this it follows that

$$\begin{aligned}
|(R^t\cap S^t)\backslash S^*| &= |R^t\cap S^t| - |R^t\cap S^t\cap S^*|\\
&\geq s^*\max\{1,8\widetilde{\kappa}\} - |S^t\cap S^*|\\
&\geq |S^*\backslash S^t| 8\widetilde{\kappa}\\
&= |S^*\backslash S^t| 8\frac{\rho^+}{\rho^-}
\end{aligned}$$

and so

$$\begin{aligned}
(x_j^t)^2 &\leq \frac{1}{|(R^t\cap S^t)\backslash S^*|}\left\|x^t_{(R^t\cap S^t)\backslash S^*}\right\|_2^2\\
&\leq \frac{\rho^-}{8|S^*\backslash S^t|\rho^+}\left\|x^t_{R^t\backslash S^*}\right\|_2^2\\
&= \frac{\rho^-}{8|S^*\backslash S^t|(\rho^+)^2}\langle \nabla_{S^t\backslash S^*}\Phi^t(x^t), x^t_{S^t\backslash S^*}\rangle ,
\end{aligned}$$

where $j \in S^t$ is the element that the algorithm removes from S^t, and we used (3). Combining this inequality with the statement of Lemma 18 we have

$$\begin{aligned}
&g^t(x^t) - g^t(x^{t+1})\\
&\geq \frac{\rho^-}{2|S^*\backslash S^t|\rho^+}\left(f(x^t) - f(x^*) + \langle \nabla_{S^t\backslash S^*}\Phi^t(x^t), x^t_{S^t\backslash S^*}\rangle - \frac{1}{2\rho^-}\left\|\nabla_{S^t\cap S^*}\Phi^t(x^t)\right\|_2^2\right) - \rho^+(x_j^t)^2\\
&\geq \frac{\rho^-}{2|S^*\backslash S^t|\rho^+}\left(f(x^t) - f(x^*) + \frac{3}{4}\langle \nabla_{S^t\backslash S^*}\Phi^t(x^t), x^t_{S^t\backslash S^*}\rangle - \frac{1}{2\rho^-}\left\|\nabla_{S^t\cap S^*}\Phi^t(x^t)\right\|_2^2\right) .
\end{aligned} \tag{7}$$

By definition of a Type 2 iteration,

$$\begin{aligned}
g^t(x^t) - g^t(x^{t+1}) &< \frac{1}{s}\left(g^t(x^t) - \widehat{f}^*\right)\\
&\leq \frac{\rho^-}{2|S^*\backslash S^t|\rho^+}\left(g^t(x^t) - f(x^*)\right) \qquad ,\\
&= \frac{\rho^-}{2|S^*\backslash S^t|\rho^+}\left(f(x^t) - f(x^*) + \Phi^t(x^t)\right)
\end{aligned} \tag{8}$$

where we used the fact that $s \geq 2s^*\widetilde{\kappa} \geq 2|S^*\backslash S^t|\widetilde{\kappa}$ and $f(x^*) \leq \widehat{f}^*$. Combining inequalities (7) and (8) we get

$$\Phi^t(x^t) > \frac{3}{4}\langle \nabla_{S^t\backslash S^*}\Phi^t(x^t), x^t_{S^t\backslash S^*}\rangle - \frac{1}{2\rho^-}\left\|\nabla_{S^t\cap S^*}\Phi^t(x^t)\right\|_2^2,$$

or equivalently, by replacing $\Phi^t(x^t)$ from (5),

$$\frac{1}{2}\left(\frac{1}{\rho^-}+\frac{1}{\rho^+}\right)\left\|\nabla_{S^t\cap S^*}\Phi^t(x^t)\right\|_2^2 > \frac{1}{4}\langle \nabla_{S^t\backslash S^*}\Phi(x^t), x^t_{S^t\backslash S^*}\rangle .$$

Further applying (3) and (4), we equivalently get

$$2\left(1+\widetilde{\kappa}\right)\left\|x^t_{R^t\cap S^*}\right\|_2^2 > \left\|x^t_{R^t\backslash S^*}\right\|_2^2 . \tag{9}$$

Now, note that in Lines 21-22 the algorithm picks an element $i \in R^t$ with probability proportional to $(x_i^t)^2$ and unregularizes it, i.e. sets $R^{t+1} \leftarrow R^t\backslash\{i\}$. We denote this probability distribution over $i \in R^t$ by $\mathcal{D}$. From what we have established already in (9), we can lower bound the probability that i lies in the target support:

$$\begin{aligned}
\Pr_{i\sim\mathcal{D}}[i \in S^*] &= \frac{\left\|x^t_{R^t\cap S^*}\right\|_2^2}{\left\|x^t_{R^t\cap S^*}\right\|_2^2 + \left\|x^t_{R^t\backslash S^*}\right\|_2^2} \\
&> \frac{\frac{1}{2(1+\widetilde{\kappa})}}{1+\frac{1}{2(1+\widetilde{\kappa})}} \\
&= \frac{1}{2\widetilde{\kappa}+3} \\
&:= p .
\end{aligned} \tag{10}$$

Note that this event can happen at most once for each $i \in S^*$ during the whole execution of the algorithm, since each element can only be removed once from the set of regularized elements.

We will prove that with constant probability the number of Type 2 steps will be at most $(s^*-1)(4\widetilde{\kappa}+6) := b$. For $1 \leq k \leq b$, we define the following random variables:

- $i_k \in [n]$ is the index picked in the k-th Type 2 iteration, or $\perp$ if there are less than k Type 2 iterations.

- q_k is the probability of picking an index in the optimal support in the k-th Type 2 iteration (i.e. $i_k \in S^*$):

$$q_k = \begin{cases} \left\|x^{t_k}_{R^{t_k}\cap S^*}\right\|_2^2 / \left\|x^{t_k}_{R^{t_k}}\right\|_2^2 & \text{if } i_k \neq \perp \\ 0 & \text{otherwise} \end{cases},$$

  where $t_k \in [T]$ is the index of the k-th Type 2 iteration within all iterations of the algorithm. Note that, by (10), $q_k > 0$ implies $q_k \geq p$.

- X_k is 1 if the index picked in the k-th Type 2 step was in the optimal support:

$$X_k = \begin{cases} 1 & \text{with probability } q_k \\ 0 & \text{otherwise} \end{cases}$$

Our goal is to upper bound $\Pr\left[\sum_{k=1}^{b} X_k \le s^* - 1\right]$. This automatically implies the same upper bound on the probability that there will be more than b Type 2 iterations.

We define another sequence of random variables $Y_0, \dots, Y_b$, where $Y_0 = 0$, and

$$Y_k = \begin{cases} Y_{k-1} + \frac{p}{q_k} - p & \text{if } X_k = 1 \\ Y_{k-1} - p & \text{if } X_k = 0 \end{cases},$$

for $k \in [b]$. Since if $q_k > 0$ we have $\frac{p}{q_k} \le 1$, it is immediate that

$$Y_k - Y_{k-1} \le X_k - p$$

and so $Y_b \le \sum_{k=1}^{b} X_k - bp$. Furthermore,

$$\begin{aligned} \mathbb{E}\left[Y_k \mid i_1, \dots, i_{k-1}\right] &= Y_{k-1} + q_k\left(\frac{p}{q_k} - p\right) - (1 - q_k)\, p \\ &= Y_{k-1}\,, \end{aligned}$$

meaning that $Y_0, \dots, Y_b$ is a martingale with respect to $i_1, \dots, i_b$. We will apply the inequality from Lemma 8. We compute a bound on the differences

$$\begin{aligned} Y_{k-1} - Y_k &= \begin{cases} p - \frac{p}{q_k} & \text{if } X_k = 1 \\ p & \text{if } X_k = 0 \end{cases} \\ &\le p \end{aligned} \tag{11}$$

and the variance

$$\begin{aligned} \operatorname{Var}\left(Y_k \mid i_1, \dots, i_{k-1}\right) &= \mathbb{E}\left[\left(Y_k - \mathbb{E}\left[Y_k \mid i_1, \dots, i_{k-1}\right]\right)^2 \;\mid i_1, \dots, i_{k-1}\right] \\ &= \mathbb{E}\left[\left(Y_k - Y_{k-1}\right)^2 \;\mid i_1, \dots, i_{k-1}\right] \\ &= q_k \cdot \left(p - \frac{p}{q_k}\right)^2 + (1 - q_k) \cdot p^2 \\ &= q_k \cdot p^2\left(1 - \frac{2}{q_k} + \frac{1}{q_k^2}\right) + (1 - q_k) \cdot p^2 \\ &= p^2\left(\frac{1}{q_k} - 1\right) \\ &\le p\,, \end{aligned}$$

where we used (11) along with the fact that $q_k \geq p$. Using the concentration inequality from Lemma 8 we obtain

$$\begin{aligned}
\Pr\left[\sum_{k=1}^{b} X_k \leq s^* - 1\right] &\leq \Pr\left[Y_b \leq s^* - 1 - b \cdot p\right] \\
&\leq e^{-(bp - s^* + 1)^2/(2(b\cdot p + p\cdot(bp - s^* + 1)/3))} \\
&= e^{-(s^*-1)/(2(2+p/3))} \\
&\leq e^{-1/(2(2+1/9))} \\
&< 0.8\,,
\end{aligned}$$

where we used the fact that $bp = 2(s^* - 1)$, $s^* \geq 2$ (otherwise the problem is trivial), and $p = \frac{1}{2\widetilde{\kappa}+3} \leq \frac{1}{3}$. Therefore we conclude that the probability that we have not unregularized the whole set S^* after b steps is at most 0.8. Since we can only have a Type 2 step if there is a regularized element in S^* (this is immediate e.g. from (10)), this implies that with probability at least 0.2 the number of Type 2 steps is at most $b = (s^* - 1)(4\widetilde{\kappa} + 6)$.

■

3.3 Corollaries

As the first corollary of Theorem 10, 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 20 (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 6 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

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

Proof By strong convexity we have

$$\begin{aligned}\epsilon &\geq f(x) - f(x^*)\\ &\geq \langle \nabla f(x^*), x - x^* \rangle + \frac{\rho^-}{2} \|x - x^*\|_2^2\\ &\geq -\zeta \|x - x^*\|_2 + \frac{\rho^-}{2} \|x - x^*\|_2^2 \,,\end{aligned}$$

therefore

$$\frac{\rho^-}{2} \|x - x^*\|_2^2 - \zeta \|x - x^*\|_2 - \epsilon \leq 0 \,,$$

looking at which as a quadratic polynomial in $\|x - x^*\|_2$, it follows that

$$\begin{aligned}\|x - x^*\|_2 &\leq \frac{\zeta + \sqrt{\zeta^2 + 2\epsilon\rho^-}}{\rho^-}\\ &= \frac{\zeta}{\rho^-}\left(1 + \sqrt{1 + 2\epsilon\frac{\rho^-}{\zeta^2}}\right)\\ &= (2 + \theta)\frac{\zeta}{\rho^-} \,,\end{aligned}$$

by setting $\epsilon = \frac{\zeta^2}{\rho^-}\left(\theta + \frac{1}{2}\theta^2\right)$. ■

The next corollary shows that our Theorem 10 can be also used to obtain support recovery results under a "Signal-to-Noise" condition given as a lower bound to $|x^*_{\min}|$.

Corollary 21 (Support recovery) *As long as*

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

and $|x^*_{\min}| > \frac{\zeta}{\rho^-}$, *Algorithm 6 with* $\epsilon < -\frac{1}{2\rho^-}\zeta^2 + \frac{\rho^-}{2}(x^*_{\min})^2$ *returns a solution* x *with support* S *such that*

$$S^* \subseteq S \,.$$

Proof Let us suppose that $S^* \backslash S^t \neq \emptyset$. By restricted strong convexity we have

$$\begin{aligned}-\frac{1}{2\rho^-}\zeta^2 + \frac{\rho^-}{2}(x^*_{\min})^2 &> \epsilon\\ &\geq f(x) - f(x^*)\\ &\geq \langle \nabla f(x^*), x - x^* \rangle + \frac{\rho^-}{2} \|x - x^*\|_2^2\\ &\geq \langle \nabla f(x^*), x \rangle + \frac{\rho^-}{2} \|x_{S^t \backslash S^*}\|_2^2 + \frac{\rho^-}{2} \left\|x^*_{S^* \backslash S^t}\right\|_2^2\\ &\geq -\frac{1}{2\rho^-} \left\|\nabla_{S^t \backslash S^*} f(x^*)\right\|_2^2 + \frac{\rho^-}{2} \left\|x^*_{S^* \backslash S^t}\right\|_2^2\\ &\geq -\frac{1}{2\rho^-}\zeta^2 + \frac{\rho^-}{2}(x^*_{\min})^2\end{aligned}$$

a contradiction. Here we used the fact that by local optimality $\nabla_{S^*} f(x^*) = \vec{0}$, the inequality $\langle u, v\rangle + \frac{\lambda}{2} \|v\|_2^2 \geq -\frac{1}{2\lambda} \|u\|_2^2$ for any vectors u, v and scalar $\lambda > 0$, and the fact that $\left\|\nabla_{S^t \setminus S^*} f(x^*)\right\|_2^2 \leq \zeta^2$ by Definition 6. Therefore $S^* \subseteq S^t$. ■

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

4.1 Overview and Main Theorem

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. The algorithm description is in Algorithm 3.

For each iteration t of Algorithm 3, we will define a solution

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

to be the optimal solution supported on $S^t \cup S^*$. Furthermore, we let $\overline{x}^*$ be the optimal $(s+s^*)$-sparse solution, i.e.

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

By definition, the following chain of inequalities holds

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

We will denote $\mu = \sqrt{\frac{s^*}{s}}$. The following technical lemma is important for our approach, and roughly states that if there is significant ℓ_2 norm difference between x^t and x^*, at least one of x^t, x^* is significantly larger than $\widetilde{x}^t$ in function value. Its importance lies on the fact that instead of directly applying strong convexity between x^t and x^*, it gets a tighter bound by making use of $\widetilde{x}^t$.

Lemma 22 *For any function f with RSC constant ρ^- at sparsity level $s + s^*$ and any two solutions x^t,x^* with respective supports S^t, S^* and sparsity levels s, s^*, we have that*

$$\left(\sqrt{f(x^t) - f(\widetilde{x}^t)} + \sqrt{f(x^*) - f(\widetilde{x}^t)}\right)^2 \geq \frac{\rho^-}{2}\left(\left\|x^*_{S^* \setminus S^t}\right\|_2^2 + \left\|x^t_{S^t \setminus S^*}\right\|_2^2\right)\,.$$

The proof can be found in Appendix A.3.

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. It can be seen as a careful and general analysis of OMPR, and, in contrast to Theorem 10, the interesting part is not the asymptotic sparsity bound (which is $O(\kappa^2 s^*)$ and is known), but the constant factor in front of it, which allows its use in recovering a solution with sparsity close to s^*, under a RIP bound. It roughly states that for any solution x^* that is s^*-sparse, OMPR can be used to obtain a solution x

- with sparsity $\leq \kappa^2 s^*$ that comes arbitrarily close to x^* in function value, i.e. $f(x) \leq f(x^*) + \epsilon$
- with sparsity $\leq \kappa^2 s^*/4$ that approximates x^* in function value, and the approximation depends on how close x^* is to being a global optimum.

The latter can be used to obtain compressed sensing results, as it gives sparsity very close to s^* given that upper bounds on κ (equivalently, RIP upper bounds) are met. In comparison with previously known results, our work is the first to obtain compressed sensing RIP bounds for general functions f and for a wide range of sparsity levels from s^* to much larger than that.

Theorem 23 *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

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

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

4.2 Progress Lemma and Theorem Proof

The main ingredient needed to prove Theorem 23 is the following lemma, which bounds the progress of Algorithm 3 in one iteration.

Lemma 24 (OMPR Progress Lemma) *We can bound the progress of one step of the algorithm by distinguishing the following three cases:*
• *If $\mu\widetilde{\kappa} \leq 1$, then*

$$f(x^{t+1}) - f(x^*) \leq \left(f(x^t) - f(x^*)\right)\left(1 - \frac{\mu}{|S^*\backslash S^t|}\right)$$

• *If $\mu\widetilde{\kappa} > 1$ and $f(x^*) = f(\widetilde{x}^t)$, then*

$$f(x^{t+1}) - f(x^*) \leq \left(f(x^t) - f(x^*)\right) \cdot \left(1 - \frac{\mu}{|S^*\backslash S^t|}(2 - \mu\widetilde{\kappa})\right)$$

- *If $\mu\widetilde{\kappa} > 1$ and $f(x^*) > f(\widetilde{x}^t)$, then*

$$f(x^{t+1}) - f(x^*) \leq \left(f(x^t) - f(x^*)\right) \cdot \left(1 - \frac{\mu}{|S^*\backslash S^t|}\left(2 - \mu\widetilde{\kappa} - \frac{2(\mu\widetilde{\kappa} - 1)}{\sqrt{\frac{f(x^t)-f(\widetilde{x}^t)}{f(x^*)-f(\widetilde{x}^t)}} - 1}\right)\right)$$

Proof The proof will proceed as follows: We will use the smoothness and strong convexity of f to get a bound on the progress of one step of the algorithm in decreasing f, based on $f(x^t) - f(x^*)$. This progress will be offset by the ℓ_2 norm of x^* restricted to $S^*\backslash S^t$. We use Lemma 22 to upper bound this norm by a quantity that depends on $f(x^t) - f(\widetilde{x}^t)$ and $f(x^*) - f(\widetilde{x}^t)$, where $\widetilde{x}^t$ is the optimal solution in the joint support $S^t \cup S^*$. Finally, by a careful case analysis based on the value of μ, we obtain the three bullet points in the lemma statement.

First of all, if $S^* \subseteq S^t$ then, since x^t is an S^t-restricted minimizer, we have $f(x^T) \leq f(x^t) \leq f(x^*)$ and we are done. So suppose otherwise, i.e. $S^*\backslash S^t \neq \emptyset$ and $f(x^t) > f(x^*)$. Let $i = \underset{i \notin S^t}{\operatorname{argmax}} \left|\nabla_i f(x^t)\right|$ and $j = \underset{j \in S^t}{\operatorname{argmin}} \left|x_j^t\right|$. By definition of OMPR (Algorithm 3) and restricted smoothness of f, we have

$$\begin{aligned}
f(x^{t+1}) &\leq \min_{\eta \in \mathbb{R}} f(x^t + \eta\vec{1}_i - x_j^t\vec{1}_j) \\
&\leq \min_{\eta \in \mathbb{R}} f(x^t) + \langle \nabla f(x^t), \eta\vec{1}_i - x_j^t\vec{1}_j\rangle + \frac{\rho^+}{2}\left\|\eta\vec{1}_i - x_j^t\vec{1}_j\right\|_2^2 \\
&= \min_{\eta \in \mathbb{R}} f(x^t) + \eta\nabla_i f(x^t) + \frac{\rho^+}{2}\eta^2 + \frac{\rho^+}{2}(x_j^t)^2 \\
&= f(x^t) - \frac{\left(\nabla_i f(x^t)\right)^2}{2\rho^+} + \frac{\rho^+}{2}(x_j^t)^2 \\
&\leq f(x^t) - \frac{\left\|\nabla_{S^*\backslash S^t} f(x^t)\right\|_2^2}{2\rho^+|S^*\backslash S^t|} + \frac{\rho^+}{2|S^t\backslash S^*|}\left\|x_{S^t\backslash S^*}^t\right\|_2^2,
\end{aligned} \tag{12}$$

where the second to last equality follows from the fact that $\nabla_j f(x^t) = \vec{0}$, as x^t is an S^t-restricted minimizer of f, and the last inequality since

$$(x_j^t)^2 = \min_{j \in S^t\backslash S^*} (x_j^t)^2 \leq \frac{\left\|x_{S^t\backslash S^*}^t\right\|_2^2}{|S^t\backslash S^*|}.$$

Re-arranging (12), we get

$$|S^*\backslash S^t|(f(x^t) - f(x^{t+1})) \geq \frac{\left\|\nabla_{S^*\backslash S^t} f(x^t)\right\|_2^2}{2\rho^+} - \frac{\rho^+}{2}\frac{|S^*\backslash S^t|}{|S^t\backslash S^*|}\left\|x_{S^t\backslash S^*}^t\right\|_2^2. \tag{13}$$

On the other hand, by restricted strong convexity of f,

$$
\begin{aligned}
f(x^*) - f(x^t) &\ge \langle \nabla f(x^t), x^* - x^t \rangle + \frac{\rho^-}{2} \left\| x^* - x^t \right\|_2^2 \\
&= \langle \nabla_{S^* \setminus S^t} f(x^t), x^*_{S^* \setminus S^t} \rangle + \frac{\rho^-}{2} \left\| x^* - x^t \right\|_2^2 \\
&\ge \langle \nabla_{S^* \setminus S^t} f(x^t), x^*_{S^* \setminus S^t} \rangle + \frac{\rho^-}{2} \left\| x^*_{S^* \setminus S^t} \right\|_2^2 + \frac{\rho^-}{2} \left\| x^t_{S^t \setminus S^*} \right\|_2^2 \\
&\ge \langle \nabla_{S^* \setminus S^t} f(x^t), x^*_{S^* \setminus S^t} \rangle + \frac{\mu\rho^+}{2} \left\| x^*_{S^* \setminus S^t} \right\|_2^2 \\
&+ \frac{\rho^- - \mu\rho^+}{2} \left\| x^*_{S^* \setminus S^t} \right\|_2^2 + \frac{\rho^-}{2} \left\| x^t_{S^t \setminus S^*} \right\|_2^2 \\
&\ge -\frac{1}{2\mu\rho^+} \left\| \nabla_{S^* \setminus S^t} f(x^t) \right\|_2^2 + \frac{\rho^- - \mu\rho^+}{2} \left\| x^*_{S^* \setminus S^t} \right\|_2^2 + \frac{\rho^-}{2} \left\| x^t_{S^t \setminus S^*} \right\|_2^2 ,
\end{aligned}
\tag{14}
$$

where the first equality follows from the fact that $\nabla_{S^t} f(x^t) = \vec{0}$ as x^t is an S^t-restricted minimizer of f and the last inequality from using the fact that $\langle u, v \rangle + \frac{\lambda}{2} \|v\|_2^2 \ge -\frac{1}{2\lambda} \|u\|_2^2$ for any $\lambda > 0$.

Re-arranging (14), we get

$$
\frac{1}{2\mu\rho^+} \left\| \nabla_{S^* \setminus S^t} f(x^t) \right\|_2^2 \ge f(x^t) - f(x^*) - \frac{\mu\rho^+ - \rho^-}{2} \left\| x^*_{S^* \setminus S^t} \right\|_2^2 + \frac{\rho^-}{2} \left\| x^t_{S^t \setminus S^*} \right\|_2^2 .
$$

By substituting this into (13),

$$
\begin{aligned}
&|S^* \setminus S^t| (f(x^t) - f(x^{t+1})) \\
&\ge \mu \left(f(x^t) - f(x^*) \right) - \frac{\mu^2\rho^+ - \mu\rho^-}{2} \left\| x^*_{S^* \setminus S^t} \right\|_2^2 + \frac{\mu\rho^-}{2} \left\| x^t_{S^t \setminus S^*} \right\|_2^2 - \frac{\rho^+}{2} \frac{|S^* \setminus S^t|}{|S^t \setminus S^*|} \left\| x^t_{S^t \setminus S^*} \right\|_2^2 .
\end{aligned}
$$

Note that by our choice of μ and since $s^* \le s$,

$$
\mu^2 \rho^+ = \rho^+ \frac{s^*}{s} \ge \rho^+ \frac{s^* - |S^* \cap S^t|}{s - |S^* \cap S^t|} = \rho^+ \frac{|S^* \setminus S^t|}{|S^t \setminus S^*|}
$$

and so

$$
\begin{aligned}
&\mu \left(f(x^t) - f(x^*) \right) - \frac{\mu^2\rho^+ - \mu\rho^-}{2} \left\| x^*_{S^* \setminus S^t} \right\|_2^2 + \frac{\mu\rho^-}{2} \left\| x^t_{S^t \setminus S^*} \right\|_2^2 - \frac{\rho^+}{2} \frac{|S^* \setminus S^t|}{|S^t \setminus S^*|} \left\| x^t_{S^t \setminus S^*} \right\|_2^2 \\
&\ge \mu (f(x^t) - f(x^*)) - \frac{\mu}{2} \left(\mu\rho^+ - \rho^- \right) \left(\left\| x^*_{S^* \setminus S^t} \right\|_2^2 + \left\| x^t_{S^t \setminus S^*} \right\|_2^2 \right) ,
\end{aligned}
$$

concluding that

$$
|S^* \setminus S^t| (f(x^t) - f(x^{t+1})) \ge \mu (f(x^t) - f(x^*)) - \frac{\mu}{2} \left(\mu\rho^+ - \rho^- \right) \left(\left\| x^*_{S^* \setminus S^t} \right\|_2^2 + \left\| x^t_{S^t \setminus S^*} \right\|_2^2 \right) .
$$

For $\mu\widetilde{\kappa} \le 1 \Leftrightarrow \mu\rho^+ - \rho^- \le 0$, this automatically implies that

$$
\begin{aligned}
& f(x^t) - f(x^{t+1}) \ge \frac{\mu}{|S^* \setminus S^t|} \left(f(x^t) - f(x^*) \right) \\
\Leftrightarrow & f(x^{t+1}) - f(x^*) \le \left(1 - \frac{\mu}{|S^* \setminus S^t|} \right) \left(f(x^t) - f(x^*) \right) .
\end{aligned}
$$

On the other hand, if $\mu\widetilde{\kappa} > 1$ we have

$$\begin{aligned}
&|S^*\backslash S^t| \left(f(x^t) - f(x^{t+1})\right) \\
&\geq \mu(f(x^t) - f(x^*)) - \frac{\mu}{2}\left(\mu\rho^+ - \rho^-\right)\left(\left\|x^*_{S^*\backslash S^t}\right\|_2^2 + \left\|x^t_{S^t\backslash S^*}\right\|_2^2\right) \\
&\geq \mu(f(x^t) - f(x^*)) - \mu\left(\mu\widetilde{\kappa} - 1\right)\left(\sqrt{f(x^t) - f(\widetilde{x}^t)} + \sqrt{f(x^*) - f(\widetilde{x}^t)}\right)^2 ,
\end{aligned}$$

where we used Lemma 22. If $f(x^*) = f(\widetilde{x}^t)$ it is immediate that

$$f(x^{t+1}) - f(x^*) \leq \left(1 - \frac{\mu}{|S^*\backslash S^t|}\left(2 - \mu\widetilde{\kappa}\right)\right)\left(f(x^t) - f(x^*)\right) ,$$

so let us from now on assume that $f(x^*) > f(\widetilde{x}^t)$ and set $a = f(x^t) - f(\widetilde{x}^t)$, $a' = f(x^{t+1}) - f(\widetilde{x}^t)$, and $b = f(x^*) - f(\widetilde{x}^t)$. From what we have concluded before

$$|S^*\backslash S^t|\left(a - a'\right) \geq \mu(a - b) - \mu(\mu\widetilde{\kappa} - 1)\left(\sqrt{a} + \sqrt{b}\right)^2 ,$$

or equivalently

$$\begin{aligned}
a' - b &\leq \left(1 - \frac{\mu}{|S^*\backslash S^t|}\right)(a - b) + \frac{\mu}{|S^*\backslash S^t|}\left(\mu\widetilde{\kappa} - 1\right)\left(\sqrt{a} + \sqrt{b}\right)^2 \\
&= (a - b)\left(1 - \frac{\mu}{|S^*\backslash S^t|}\left(1 - (\mu\widetilde{\kappa} - 1)\frac{\left(\sqrt{a} + \sqrt{b}\right)^2}{a - b}\right)\right) \\
&= (a - b)\left(1 - \frac{\mu}{|S^*\backslash S^t|}\left(1 - (\mu\widetilde{\kappa} - 1)\frac{\sqrt{\frac{a}{b}} + 1}{\sqrt{\frac{a}{b}} - 1}\right)\right) \\
&= (a - b)\left(1 - \frac{\mu}{|S^*\backslash S^t|}\left(1 - (\mu\widetilde{\kappa} - 1)\left(1 + \frac{2}{\sqrt{\frac{a}{b}} - 1}\right)\right)\right) \\
&= (a - b)\left(1 - \frac{\mu}{|S^*\backslash S^t|}\left(2 - \mu\widetilde{\kappa} - \frac{2(\mu\widetilde{\kappa} - 1)}{\sqrt{\frac{a}{b}} - 1}\right)\right) ,
\end{aligned}$$

Replacing back a, a', b, the desired statement follows:

$$f(x^{t+1}) - f(x^*) \leq (f(x^t) - f(x^*)) \cdot \left(1 - \frac{\mu}{|S^*\backslash S^t|}\left(2 - \mu\widetilde{\kappa} - \frac{2(\mu\widetilde{\kappa} - 1)}{\sqrt{\frac{f(x^t) - f(\widetilde{x}^t)}{f(x^*) - f(\widetilde{x}^t)}} - 1}\right)\right) .$$

■

The proof of Theorem 23 now follows by appropriately applying Lemma 24.

Proof of Theorem 23. We will directly apply Lemma 24 over the course of multiple iterations. For the second bullet of the theorem statement, the progress bound of Lemma 24 also depends on $(f(x^t) - f(\widetilde{x}^t))/(f(x^*) - f(\widetilde{x}^t))$. We show that, as long as $f(x)$ is singificantly larger than $f(x^*)$, this can be lower bounded by a sufficiently large quantity, leading to fast convergence.

Case 1: $\mu\widetilde{\kappa} \leq 1$.

By Lemma 24, we have

$$\begin{aligned} f(x^T) - f(x^*) &\leq \left(f(x^{T-1}) - f(x^*)\right)\left(1 - \frac{\mu}{|S^*\backslash S^{T-1}|}\right) \\ &\leq \left(f(x^{T-1}) - f(x^*)\right)\left(1 - \frac{\mu}{s^*}\right) \\ &\leq \left(f(x^{T-1}) - f(x^*)\right)e^{-\frac{\mu}{s^*}} \\ &\leq \dots \\ &\leq \left(f(x^0) - f(x^*)\right)e^{-T\frac{\mu}{s^*}} \\ &\leq \epsilon\,, \end{aligned}$$

for our choice of $T = O\left(\sqrt{ss^*}\log\frac{f(x^0)-f(x^*)}{\epsilon}\right)$ and replacing $\mu = \sqrt{\frac{s^*}{s}}$.

Case 2: $\mu\widetilde{\kappa} > 1$.

Let $\mathcal{A}$ be the set of $0 \leq t \leq T-1$ such that $f(x^*) = f(\widetilde{x}^t)$ and $\mathcal{B}$ the set of $0 \leq t \leq T-1$ such that $f(x^*) > f(\widetilde{x}^t)$. By Lemma 24, for $t \in \mathcal{A}$ we then have

$$\begin{aligned} f(x^{t+1}) - f(x^*) &\leq \left(f(x^t) - f(x^*)\right)\left(1 - \frac{\mu}{|S^*\backslash S^t|}(2-\mu\widetilde{\kappa})\right) \\ &\leq \left(f(x^t) - f(x^*)\right)\left(1 - \frac{\mu}{s^*}(2-\mu\widetilde{\kappa})\right)\,. \end{aligned}$$

We now consider the case $t \in \mathcal{B}$. By Lemma 24,

$$f(x^{t+1}) - f(x^*) \leq \left(f(x^t) - f(x^*)\right)\cdot\left(1 - \frac{\mu}{|S^*\backslash S^t|}\left(2 - \mu\widetilde{\kappa} - \frac{2(\mu\widetilde{\kappa}-1)}{\sqrt{\frac{f(x^t)-f(\widetilde{x}^t)}{f(x^*)-f(\widetilde{x}^t)}} - 1}\right)\right)\,. \tag{15}$$

Let us suppose that the theorem statement is not true. This implies

$$\begin{aligned} f(x^t) - f(x^*) &\geq f(x^T) - f(x^*) \\ &> \epsilon + \frac{4(1-\theta)(\mu\widetilde{\kappa}-1)}{(2-\mu\widetilde{\kappa}-\theta)^2}(f(x^*) - f(\overline{x}^*)) \\ &\geq \epsilon + \frac{4(1-\theta)(\mu\widetilde{\kappa}-1)}{(2-\mu\widetilde{\kappa}-\theta)^2}(f(x^*) - f(\widetilde{x}^t)) \\ &\geq \frac{4(1-\theta)(\mu\widetilde{\kappa}-1)}{(2-\mu\widetilde{\kappa}-\theta)^2}(f(x^*) - f(\widetilde{x}^t))\,, \end{aligned} \tag{16}$$

for all $0 \leq t \leq T$. Therefore

$$\begin{aligned} f(x^t) - f(\widetilde{x}^t) &> \left(\frac{4(1-\theta)(\mu\widetilde{\kappa}-1)}{(2-\mu\widetilde{\kappa}-\theta)^2} + 1\right)(f(x^*) - f(\widetilde{x}^t)) \\ &= \left(\frac{4(1-\theta)(\mu\widetilde{\kappa}-1) + 4 + (\mu\widetilde{\kappa}+\theta)^2 - 4(\mu\widetilde{\kappa}+\theta)}{(2-\mu\widetilde{\kappa}-\theta)^2}\right)\cdot(f(x^*) - f(\widetilde{x}^t)) \\ &= \frac{(\mu\widetilde{\kappa}-\theta)^2}{(2-\mu\widetilde{\kappa}-\theta)^2}(f(x^*) - f(\widetilde{x}^t))\,, \end{aligned}$$

or equivalently for all $t \in \mathcal{B}$

$$\sqrt{\frac{f(x^t) - f(\widetilde{x}^t)}{f(x^*) - f(\widetilde{x}^t)}} - 1 > \frac{\mu\widetilde{\kappa} - \theta}{2 - \mu\widetilde{\kappa} - \theta} - 1 = \frac{2(\mu\widetilde{\kappa} - 1)}{2 - \mu\widetilde{\kappa} - \theta}\,.$$

Replacing this into (15), we get that for any $t \in \mathcal{B}$

$$\begin{aligned} f(x^{t+1}) - f(x^*) &\leq (f(x^t) - f(x^*)) \cdot \left(1 - \frac{\mu}{|S^* \backslash S^t|}\left(2 - \mu\widetilde{\kappa} - \frac{2(\mu\widetilde{\kappa} - 1)}{\sqrt{\frac{f(x^t) - f(\widetilde{x}^t)}{f(x^*) - f(\widetilde{x}^t)}} - 1}\right)\right) \\ &\leq (f(x^t) - f(x^*)) \left(1 - \frac{\mu}{|S^* \backslash S^t|}\theta\right) \end{aligned}$$

and so combining it with the case $t \in \mathcal{A}$ and using the fact that $\mu\widetilde{\kappa} < 2 - \theta \Leftrightarrow \theta < 2 - \mu\widetilde{\kappa}$,

$$\begin{aligned} f(x^T) - f(x^*) &\leq \left(f(x^{T-1}) - f(x^*)\right) \left(1 - \frac{\mu}{|S^* \backslash S^{T-1}|} \min\{2 - \mu\widetilde{\kappa}, \theta\}\right) \\ &\leq \left(f(x^{T-1}) - f(x^*)\right) \left(1 - \frac{\mu}{s^*}\theta\right) \\ &\leq \left(f(x^{T-1}) - f(x^*)\right) e^{-\frac{\mu}{s^*}\theta} \\ &\leq \dots \\ &\leq (f(x^0) - f(x^*)) e^{-T\frac{\mu}{s^*}\theta} \\ &= \epsilon + \frac{4(1-\theta)(\mu\widetilde{\kappa} - 1)}{(2 - \mu\widetilde{\kappa} - \theta)^2}(f(x^*) - f(\overline{x}^*))\,, \end{aligned}$$

where the last equality follows by our choice of

$$T = \frac{\sqrt{ss^*}}{\theta} \log \frac{f(x^0) - f(x^*)}{B}$$

and replacing $\mu = \sqrt{\frac{s^*}{s}}$. This is a contradiction. ■

4.3 Corollaries of Theorem 23

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. For $s = s^*$, it gives the same condition of $\widetilde{\kappa} < 2$ (or $\delta < 1/3$) as in Jain et al. (2011), while working for any function f. For $s > s^*$, it additionally gives a tradeoff between the sparsity and the RIP bound required for the algorithm.

Corollary 25 (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*

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

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 23 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. Similar results can be found e.g. in Candes (2008) using LASSO, albeit they only apply to case of linear regression ($f(x) = \frac{1}{2}\|Ax - b\|_2^2$) and do not offer a sparsity tradeoff.

Corollary 26 (ℓ_2 solution recovery) *Given any parameters $\epsilon > 0$ and $0 < \theta < 1$, the returned solution x of Algorithm 3 will satisfy*

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

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 $\frac{s}{s^*}$.

Proof By triangle inequality and strong convexity, and letting $\widetilde{x}^*$ be the optimal solution in $\mathrm{supp}(x) \cup \mathrm{supp}(x^*)$, we have

$$\begin{aligned}
\|x - x^*\|_2^2 &\leq 2\left(\|x - \widetilde{x}^*\|_2^2 + \|x^* - \widetilde{x}^*\|_2^2\right) \\
&\leq \frac{4}{1 - \delta_{s+s^*}}\left(f(x) - f(\widetilde{x}^*) + f(x^*) - f(\widetilde{x}^*)\right) \\
&= \frac{4}{1 - \delta_{s+s^*}}\left(f(x) - f(x^*) + 2(f(x^*) - f(\widetilde{x}^*))\right) \\
&= \frac{4}{1 - \delta_{s+s^*}}\left(f(x) - f(x^*) + 2(f(x^*) - \min_z f(z))\right).
\end{aligned}$$

Now, by applying Theorem 23 for some error tolerance $\widehat{\epsilon} > 0$, we get

$$f(x) \leq f(x^*) + \widehat{\epsilon} + \widehat{C}(f(x^*) - f(\overline{x}^*)) \leq f(x^*) + \widehat{\epsilon} + \widehat{C}(f(x^*) - \min_z f(z)),$$

for some $\widehat{C} > 0$, and so we conclude that

$$\|x - x^*\|_2^2 \leq \frac{4\widehat{\epsilon}}{1 - \delta_{s+s^*}} + \frac{4(\widehat{C} + 2)}{1 - \delta_{s+s^*}}(f(x^*) - \min_z f(z)).$$

The statement follows by setting $\epsilon = \frac{4\widehat{\epsilon}}{1-\delta_{s+s^*}}$ and $C = \frac{4(\widehat{C}+2)}{1-\delta_{s+s^*}}$. ■

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

5. Lower Bounds

5.1 $\Omega(s^*\kappa)$ Lower Bound due to Foster et al. (2015)

In Appendix B of Foster et al. (2015) a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$ are constructed and let us define $f(x) = \frac{1}{2}\|Ax-b\|_2^2$. If we let $\overline{S^*} = \{1,\dots,n-2\}$ and $S^* = \{n-1,n\}$, then f has the property that

$$\min_{\mathrm{supp}(x)\subseteq S^*} f(x) = \min_{\mathrm{supp}(x)\subseteq \overline{S^*}} f(x) = 0\,,$$

but for any $S \subset \overline{S^*}$,

$$\min_{\mathrm{supp}(x)\subseteq S} f(x) > 0\,.$$

Furthermore, for any $S \subset \overline{S^*}$ and $x = \underset{\mathrm{supp}(x)\subseteq S}{\mathrm{argmin}}\, f(x)$, it is true that

$$\max_{i\in S^*} |\nabla_i f(x)| < \min_{i\in \overline{S^*}\backslash S} |\nabla_i f(x)|\,.$$

This means that for any algorithm with an OMP-like criterion like Orthogonal Matching Pursuit, Orthogonal Matching Pursuit with Replacement, Iterative Hard Thresholding, and Partial Hard Thresholding, if the initial solution does not have an intersection with S^*, then it will never have, therefore implying that the sparsity returned by the algorithm is $|S| = n-2 = \Omega(n)$. As for this construction $\kappa = \frac{\rho_n^+}{\rho_n^-} = O\left(n\right)$, there exists a constant c such that the sparsity of the returned solution cannot be less than $cs^*\kappa$, since $s^*\kappa = O(n) = O(|S|)$. Therefore none of these algorithms can improve the bound $O(s^*\kappa)$ of Theorem 10 by more than a constant factor. This example also applies to ARHT and Exhaustive Local Search.

It seems difficult to get past this example and achieve sparsity $s = O(s^*\kappa^{1-\delta})$ for some $\delta > 0$. We conjecture that there might be a way to turn the above example into an inapproximability result:

Conjecture 27 *For any $\delta > 0$, there is no polynomial time algorithm that given a matrix $A \in \mathbb{R}^{m\times n}$, a vector $b \in \mathbb{R}^m$, a target sparsity $s^* \geq 1$, and a desired accuracy $\epsilon > 0$, returns an $s = O(s^*\kappa_{s+s^*}^{1-\delta})$-sparse solution x such that $\|Ax-b\|_2^2 \leq \min\limits_{\|x^*\|_0\leq s^*} \|Ax^*-b\|_2^2 + \epsilon$, if such a solution exists.*

5.2 $\Omega(s^*\kappa^2)$ Lower Bound for OMPR

The following lemma shows that, without regularization, OMPR requires sparsity $\Omega(s^*\kappa^2)$ in general, and therefore the sparsity upper bound is tight. We assume that the algorithm is run for a fixed T iterations, even when the solution stops improving, for a clearer presentation.

Lemma 28 *There is a function $f(x) = \frac{1}{2}\|Ax-b\|_2^2$ where $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ and a target solution x^* of f with sparsity s^*, as well as a set $S \subseteq [n]$ with $|S| = \Theta(s^*\kappa^2)$ such that OMPR initialized with support set S returns a solution x with $f(x) = f(x^*) + \Theta(s^*\kappa^2)$.*

Proof Without loss of generality we assume that κ is an even integer and set $n = s^*\left(1+\kappa+\kappa^2\right)$. We then partition $[n]$ into three intervals $I_1 = [1, s^*]$, $I_2 = [s^*+1, s^*(1+\kappa)]$, $I_3 = [s^*(1+\kappa)+1, s^*(1+\kappa+\kappa^2)]$. We define the diagonal matrix $A \in \mathbb{R}^{n\times n}$ such that

$$A_{ii} = \begin{cases} 1 & \text{if } i \in I_1 \\ \sqrt{\kappa} & \text{if } i \in I_2 \\ 1 & \text{if } i \in I_3 \end{cases}$$

and vector $b \in \mathbb{R}^n$ such that

$$b_i = \begin{cases} \kappa\sqrt{1-4\delta} & \text{if } i \in I_1 \\ \sqrt{\kappa}\sqrt{1-2\delta} & \text{if } i \in I_2 \\ 1 & \text{if } i \in I_3 \end{cases},$$

where $\delta > 0$ is a sufficiently small scalar used to avoid ties in the steps of the algorithm. The target solution is defined as

$$x_i^* = \begin{cases} \kappa(1-4\delta) & \text{if } i \in I_1 \\ 0 & \text{if } i \in I_2 \cup I_3 \end{cases}$$

and its value is $f(x^*) = s^*\kappa^2(1-\delta)$. Now consider any initial support set $S^0 \subset I_3$ such that $|S^0| = s^*\kappa^2/2$. The initial solution will then be

$$x_i^0 = \begin{cases} 0 & \text{if } i \in I_1 \cup I_2 \cup I_3 \backslash S^0 \\ 1 & \text{if } i \in S^0 \end{cases}$$

and its value $f(x^0) = s^*\kappa^2\left(\frac{5}{4} - 3\delta\right) = f(x^*) + \Theta(s^*\kappa^2)$. The gradient at x^0 is

$$\nabla_i f(x^0) = \begin{cases} -\kappa\sqrt{1-4\delta} & \text{if } i \in I_1 \\ -\kappa\sqrt{1-2\delta} & \text{if } i \in I_2 \\ -1 & \text{if } i \in I_3 \backslash S^0 \\ 0 & \text{if } i \in S^0 \end{cases},$$

therefore the algorithm will pick $S^1 = S^0 \cup \{i^0\} \backslash \{j^0\}$ for some $i^0 \in I_2$ and some $j^0 \in S^0$, since the gradient entries in I_2 have the largest magnitude among those in $[n]$. The new solution will be

$$x_i^1 = \begin{cases} 0 & \text{if } i \in I_1 \cup I_2 \cup I_3 \backslash S^1 \\ \sqrt{1-2\delta} & \text{if } i = i^0 \\ 1 & \text{if } i \in S^1 \backslash \{i^0\} \end{cases}$$

with value $f(x^1) = s^*\kappa^2\left(\frac{5}{4} - 3\delta\right) - \frac{1}{2}(\kappa(1-2\delta) - 1)$ and gradient

$$\nabla_i f(x^1) = \begin{cases} -\kappa\sqrt{1-4\delta} & \text{if } i \in I_1 \\ -\kappa\sqrt{1-2\delta} & \text{if } i \in I_2 \backslash S^1 \\ -1 & \text{if } i \in I_3 \backslash S^1 \\ 0 & \text{if } i \in S^1 \end{cases}$$

and therefore the algorithm will pick $S^2 = S^1 \cup \{i^1\} \backslash \{i^0\}$ for some $i^1 \in I_2$. i^0 will be the one to be removed from S^1 because x_{i^0} has the smallest magnitude out of all entries in S^1. Continuing this process, the algorithm will always have $S^t \cap I_2 = 1$ and $S^t \cap I_3 = |S^t| - 1$, and so $f(x^t) = s^* \kappa^2 \left(\frac{5}{4} - 3\delta\right) - \frac{1}{2}(\kappa(1-2\delta) - 1) = f(x^*) + \Theta(s^* \kappa^2)$ for $t \geq 1$. ■

6. Experiments

6.1 Overview

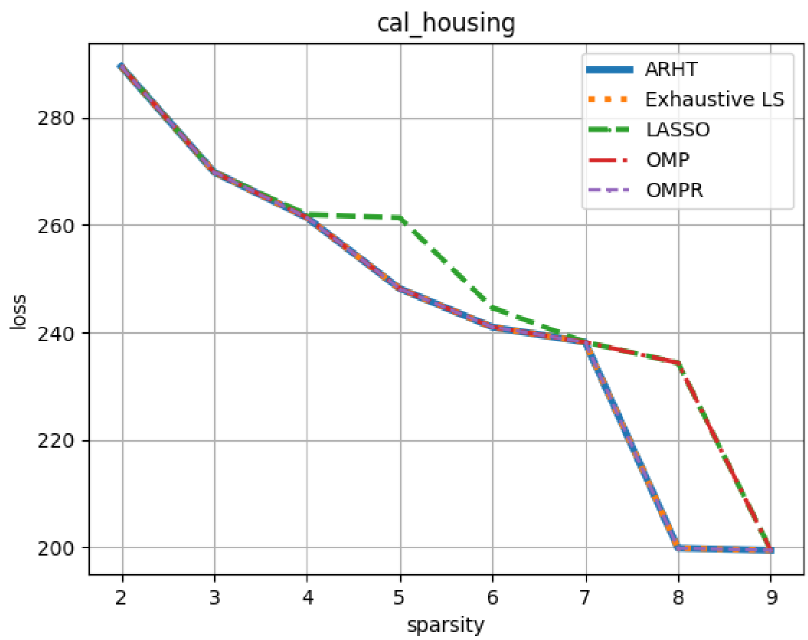
In this section we evaluate the training performance of different algorithms in the tasks of Linear Regression and Logistic Regression. More specifically, for each algorithm we are interested in how the *loss* over the training set (the quality of the solution) evolves as a function of the the *sparsity* 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 6), and *Exhaustive Local Search* (Algorithm 4). We run our experiments on publicly available regression and binary classification data sets, out of which we have presented those on which the algorithms have significantly different performance between each other. In some of the other data sets that we tested, we observed that all algorithms had similar performance. The results are presented in Figures 1, 2, 3, 4. Also, in Figure 5 we present a runtime comparison between ARHT and Exhaustive Local Search in the year and census datasets. 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 $p \leq 0.5$ 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.

We note that this section is not supposed to be a conclusive experimental evaluation of the aforementioned algorithms, but rather a preliminary set of experiments that gives partial evidence for their performance. A more extensive future experimental evaluation should focus on implementing runtime-optimized versions of these algorithms and generating sparsity vs loss and loss vs runtime plots for a larger collection of real datasets and with more datapoints and features.

These experiments suggest 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 give sparser solutions than other known methods in all the examples we tested. We leave a comprehensive comparison for future work. As a limitation, we observe that ARHT has inconsistent performance in some cases, oscillating between the Exhaustive Local Search and OMPR solutions.



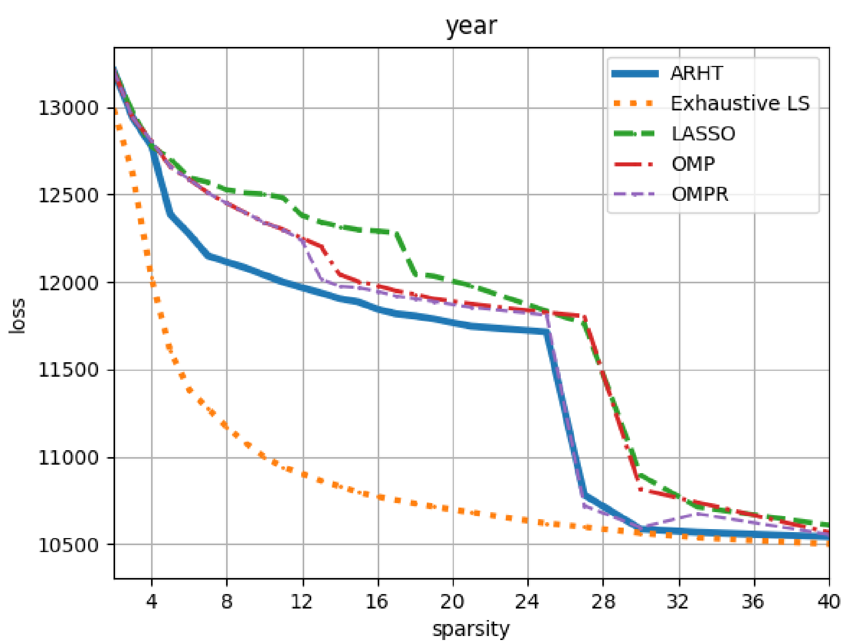


Figure 1: Comparison of different algorithms in the Regression data sets *cal_housing* and *year* using the Linear Regression loss.

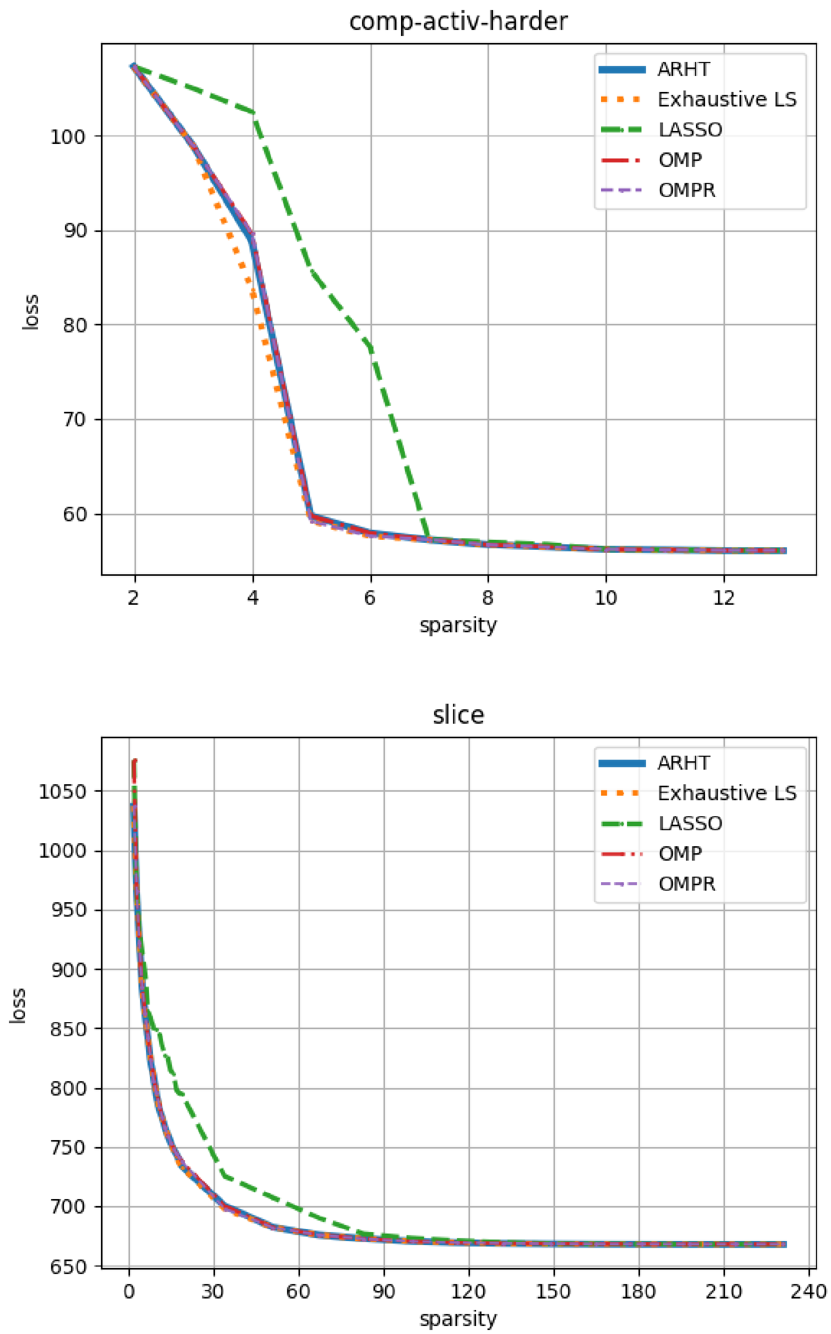


Figure 2: Comparison of different algorithms in the Regression data sets *comp-activ-harder* and *slice* using the Linear Regression loss.

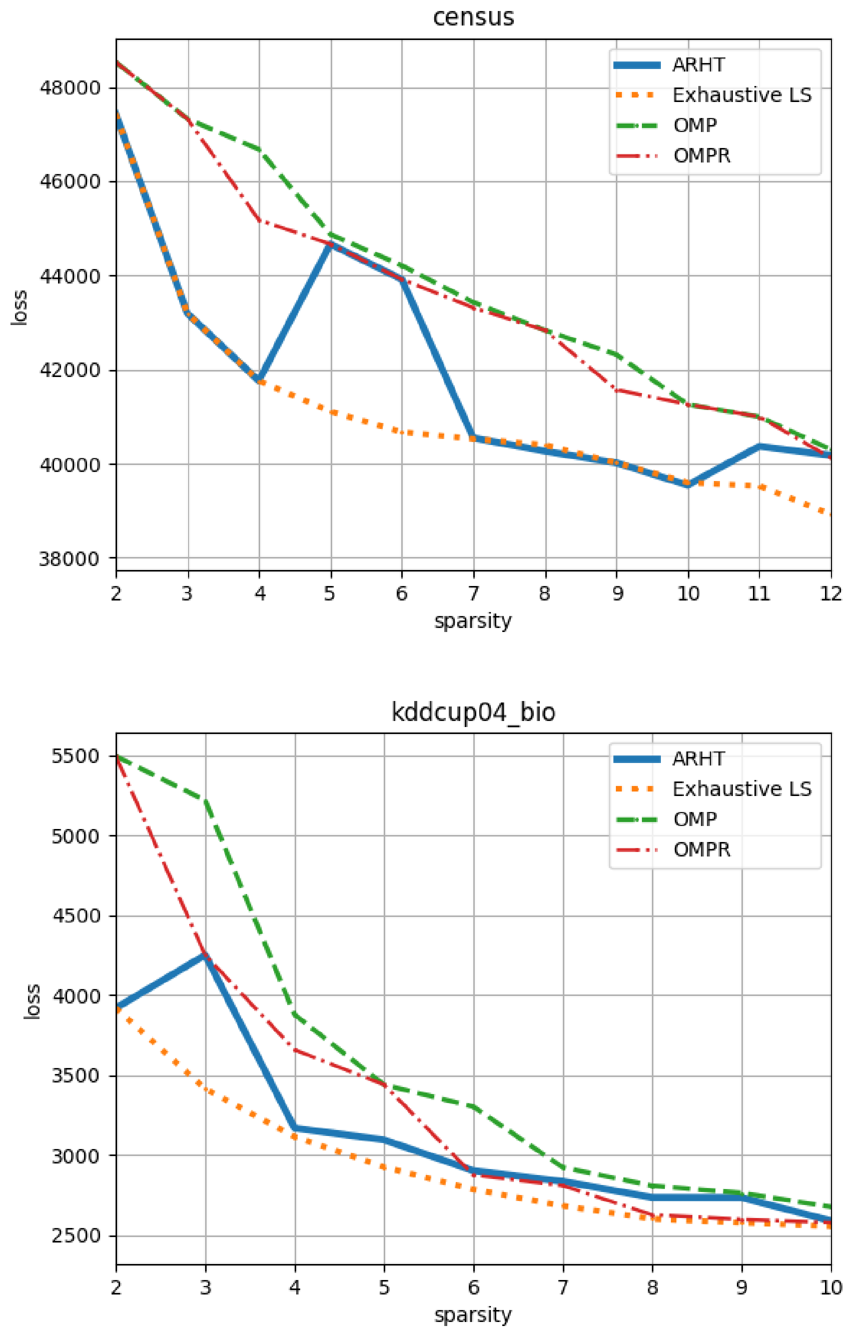


Figure 3: Comparison of different algorithms in the Binary classification data sets *census* and *kddcup04_bio* using the Logistic Regression loss.

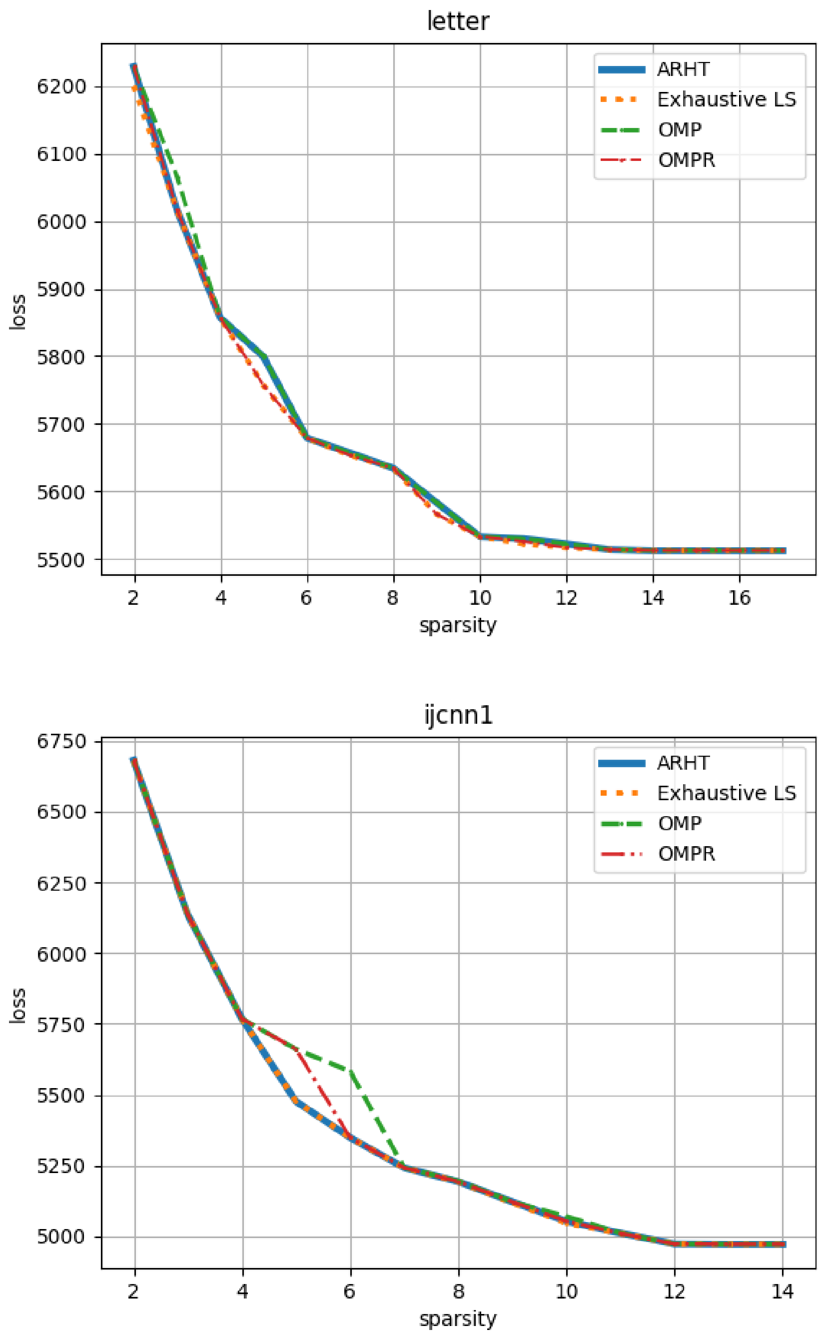


Figure 4: Comparison of different algorithms in the Binary classification data sets *letter* and *ijcnn1* using the Logistic Regression loss.

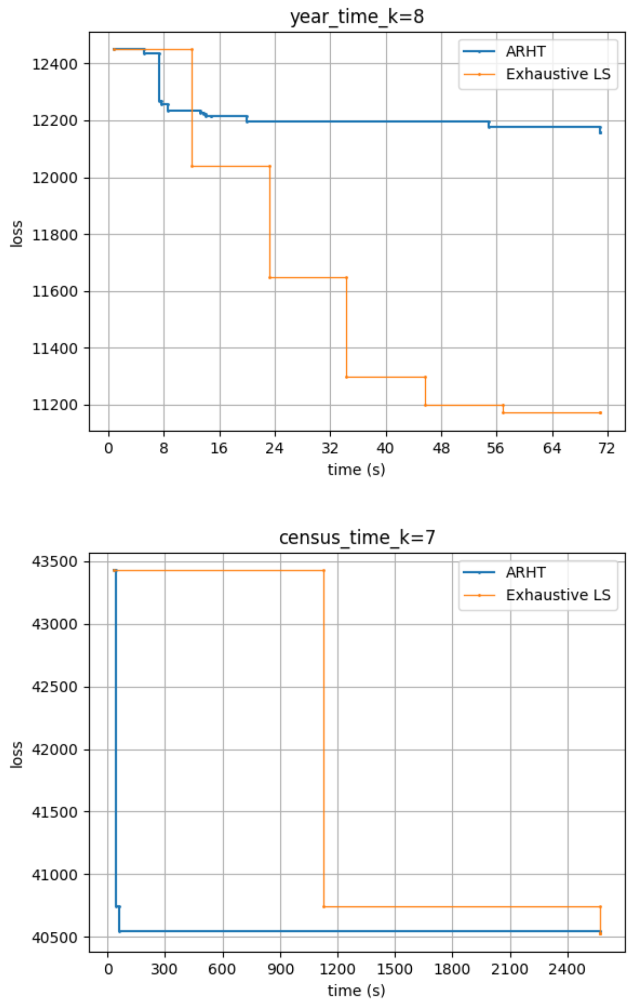


Figure 5: Comparison of the loss of a solution vs time elapsed to compute it, between ARHT and Exhaustive Local Search. The first experiment is on the *year* dataset (90 total features) with fixed sparsity 8 and the second is on the *census* dataset (401 total features) with fixed sparsity 7. We note that in the first case Exhaustive Local Search returns significantly sparser solutions without too significant time overhead compared to ARHT. However, in the second case ARHT computes a solution of similar loss to that of Exhaustive Local Search, but around 40 times faster. We attribute this to the fact that the Exhaustive Local Search has an extra n factor in the runtime, so it doesn't scale as well as ARHT as the number of features increases. Note: The reason there are "steps" in the plot is that the solution is improved at discrete time steps, i.e. whenever an insertion and removal from the solution support improves the solution.

For experimental evaluation we used well known and publicly available data sets. Their names and basic properties are outlined in Table 2.

Table 2: Data sets used for experimental evaluation. The columns are the data set name, the number of examples m, and the number of features n. The data sets can be downloaded here.

Name	n	d	problem
kddcup04_bio	145750	74	binary
cal_housing	20639	8	regression
census	299284	401	binary
comp-activ-harder	8191	12	regression
ijcnn1	24995	22	binary
letter	20000	16	binary
slice	53500	384	regression
year	463715	90	regression

6.2 Setup Details

6.2.1 Basic Definitions

The two quantities that take part in our experiments are the *sparsity* and the *loss* of a particular solution. We have already defined and discussed the former at length. The latter refers to the training loss for the problems of Linear Regression and Logistic Regression. We let m denote the number of examples and n the number of features in each example.

In the *Linear Regression* task we are given the data set (A, b), where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. The columns of A correspond to features and the rows to examples. The *(ℓ_2 Linear Regression) loss* of a solution $x \in \mathbb{R}^n$ is defined as $\ell_2\text{\_loss}(x) = \frac{1}{2}\left\|Ax - b\right\|_2^2$.

In the *Logistic Regression* task we are given the data set (A, b), where $A \in \mathbb{R}^{m\times n}$, $b \in \{0, 1\}^m$. The columns of A correspond to features and the rows to examples. The *(Logistic Regression) loss* of a solution $x \in \mathbb{R}^n$ is defined as

$$\text{logistic\_loss}(x) = \sum_{i\in[m]} \left(-b_i \log \sigma(Ax)_i - (1 - b_i) \log(1 - \sigma(Ax)_i)\right) ,$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ defined as $\sigma(t) = \frac{1}{1+e^{-t}}$ is the sigmoid function.

6.2.2 Data Pre-processing

We apply a very basic form of pre-processing to the data. More specifically, we use one-hot encoding to turn categorical features into numerical ones. Then, we discard any examples with missing data so that all the entries of A are defined. We also augment the matrix A with an extra all-ones column (i.e. $\vec{1}$) in order to encode the constant (y-intercept) term into A, and we scale all the columns of A so that their ℓ_2 norm is 1. Finally, for the case of ARHT we further augment A in order to encode the regularizer as well. We do this by adding an identity matrix as extra rows. In other words, $A \leftarrow \begin{pmatrix} A \\ I \end{pmatrix}$ and $b \leftarrow \begin{pmatrix} b \\ \vec{0} \end{pmatrix}$.

6.3 Implementation Details

The code has been implemented in *python3*, with libraries *numpy*, *sklearn*, and *scipy*.

6.3.1 Inner Optimization Problem

All the algorithms except for LASSO rely on an inner optimization routine in a restricted subset of coordinates in each step. The inner optimization problem consists of solving a standard Linear Regression or Logistic Regression problem using only a submatrix of A defined by a subset of s of its columns. For that, we use *LinearRegression* and *LogisticRegression* from *sklearn.linear_model*. For Logistic Regression we used an LBFGS solver with 1000 iterations.

6.3.2 Overall Algorithm

The LASSO solver we used is *Lasso* from *sklearn.linear_model* with 1000 iterations. As LASSO is not tuned in terms of a required sparsity s, but rather in terms of the regularization parameter α, for each sparsity level we applied binary search on α in order to find a parameter α that gives the required sparsity.

For ARHT, we used a fixed number of 20 iterations at Line 5 of Algorithm 6. In Line 19 of Algorithm 5 we slightly weaken the progress condition to

$$g_{R^t}(x^t) - g_{R^t}(x^{t+1}) \geq \frac{10^{-3}}{s}\left(g_{R^t}(x^t) - \mathrm{opt}\right) . \tag{17}$$

Furthermore, we do not perform a fixed number of iterations. Instead, we use a stopping criterion: If the progress condition (17) is not met and at least half the elements in x^t have already been unregularized, i.e. $\left|S^t \backslash R^t\right| \geq \frac{1}{2}\left|S^t\right|$, then we stop. If a desirable solution has not been found, it means that this might be an unsuccessful run, and early termination can be used to detect such runs early and re-start, thus improving the runtime. The routine which samples an index i proportional to x_i^2 was implementing by a standard sampling method that uses binary search on i and flips a random coin at each step. This requires computation of interval sums of x_i^2, which is done by computing partial sums.

Appendix A. Deferred Proofs

A.1 Proof of Lemma 17

Proof Φ^t is a quadratic restricted on R^t

$$\begin{aligned}
&\Phi(y) - \Phi(x) - \nabla\Phi(x)^T(y-x) \\
&= \frac{\rho_2^+}{2}\left(\|y_{R^t}\|_2^2 - \|x_{R^t}\|_2^2 - 2x_{R^t}^T(y_{R^t} - x_{R^t})\right) \\
&= \frac{\rho_2^+}{2}\|y_{R^t} - x_{R^t}\|_2^2 \\
&\in \left[0, \frac{\rho_2^+}{2}\|y-x\|_2^2\right]
\end{aligned}$$

and so for any x, y with $|\mathrm{supp}(y-x)| \leq s + s^*$ (resp. $|\mathrm{supp}(y-x)| \leq 1$) we have

$$\begin{aligned}
&g(y) - g(x) - \nabla g(x)^T(y-x) \\
&= f(y) - f(x) - \nabla f(x)^T(y-x) + \Phi(y) - \Phi(x) - \nabla\Phi(x)^T(y-x) \\
&\geq \frac{\rho_{s+s^*}^-}{2}\|y-x\|_2^2 \,(\text{resp.} \leq \rho_2^+\|y-x\|_2^2\,)\,.
\end{aligned}$$

■

A.2 Proof of Lemma 12

Proof By definition, and setting $\tau = \frac{1}{s}$, in each Type 1 iteration we have

$$\begin{aligned}
g(x^t) - g(x^{t+1}) &\geq \tau\left(g(x^t) - \mathrm{opt}\right) \\
\Rightarrow g(x^{t+1}) - \mathrm{opt} &\leq (1-\tau)(g(x^t) - \mathrm{opt})
\end{aligned}$$

and in each Type 2 iteration we have

$$g(x^{t+1}) - \mathrm{opt} \leq g(x^t) - \mathrm{opt}$$

(since g can only decrease when unregularizing), therefore

$$\begin{aligned}
f(x^T) - \mathrm{opt} &\leq g(x^T) - \mathrm{opt} \\
&\leq (1-\tau)^{T_1}(g(x^0) - \mathrm{opt}) \\
&\leq e^{-\tau T_1}(g(x^0) - \mathrm{opt}) \\
&\leq \epsilon\,,
\end{aligned}$$

where we used the fact that $T_1 = \frac{1}{\tau}\log\frac{g(x^0)-\mathrm{opt}}{\epsilon}$. ■

A.3 Proof of Lemma 22

Proof We have

$$\begin{aligned}\left(\sqrt{f(x^t)-f(\widetilde{x}^t)}+\sqrt{f(x^*)-f(\widetilde{x}^t)}\right)^2 &\geq \frac{\rho^-}{2}\left(\left\|x^t-\widetilde{x}^t\right\|_2+\left\|x^*-\widetilde{x}^t\right\|_2\right)^2\\ &\geq \frac{\rho^-}{2}\left\|x^t-x^*\right\|_2^2\\ &\geq \frac{\rho^-}{2}\left(\left\|x^*_{S^*\setminus S^t}\right\|_2^2+\left\|x^t_{S^t\setminus S^*}\right\|_2^2\right),\end{aligned}$$

where the first inequality follows by applying strong convexity to lower bound $f(x^t)-f(\widetilde{x}^t)$ and $f(x^*)-f(\widetilde{x}^t)$ combined with the fact that by definition of $\widetilde{x}^t$, $\nabla_{S^t\cup S^*}f(\widetilde{x}^t)=\vec{0}$, and the second is a triangle inequality. ■

References

Jason Altschuler, Aditya Bhaskara, Gang Fu, Vahab Mirrokni, Afshin Rostamizadeh, and Morteza Zadimoghaddam. Greedy column subset selection: New bounds and distributed algorithms. In *International Conference on Machine Learning*, pages 2539–2548, 2016.

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

Kyriakos Axiotis and Maxim Sviridenko. Local search algorithms for rank-constrained convex optimization. *arXiv preprint arXiv:2101.06262*, 2021.

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

Thomas Blumensath and Mike E Davies. On the difference between orthogonal matching pursuit and orthogonal least squares. 2007.

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

Holger Boche, Robert Calderbank, Gitta Kutyniok, and Jan Vybíral. A survey of compressed sensing. In *Compressed Sensing and its Applications*, pages 1–39. Springer, 2015.

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

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

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

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

Emmanuel J Candes, Justin K Romberg, and Terence Tao. 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.

Lin Chen, Moran Feldman, and Amin Karbasi. Weakly submodular maximization beyond cardinality constraints: Does randomization help greedy? In *International Conference on Machine Learning*, pages 804–813, 2018.

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

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

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

Ethan Elenberg, Alexandros G Dimakis, Moran Feldman, and Amin Karbasi. Streaming weak submodularity: Interpreting neural networks on the fly. In *Advances in Neural Information Processing Systems*, pages 4044–4054, 2017.

Dean Foster, Howard Karloff, and Justin Thaler. Variable selection is hard. In *Conference on Learning Theory*, pages 696–709, 2015.

Simon Foucart. A note on guaranteed sparse recovery via ℓ_1-minimization. *Applied and Computational Harmonic Analysis*, 29(1):97–103, 2010.

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

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

Simon Foucart and Ming-Jun Lai. Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$. *Applied and Computational Harmonic Analysis*, 26(3): 395–407, 2009.

Simon Foucart and Holger Rauhut. A mathematical introduction to compressive sensing. *Bulletin of the American Mathematical Society*, 54:151–165, 2017.

Prateek Jain, Ambuj Tewari, and Inderjit S Dhillon. Orthogonal matching pursuit with replacement. In *Advances in Neural Information Processing Systems*, pages 1215–1223, 2011.

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

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

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

Qun Mo and Song Li. New bounds on the restricted isometry constant δ_{2k}. *Applied and Computational Harmonic Analysis*, 31(3):460–468, 2011.

Seyedahmad Mousavi, Mohammad Mehdi Rezaee Taghiabadi, and Ramin Ayanzadeh. A survey on compressive sensing: Classical results and recent advancements. *arXiv preprint arXiv:1908.01014*, 2019.

Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

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

Deanna Needell and Roman Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of Computational Mathematics*, 9(3):317–334, 2009.

Deanna Needell and Roman Vershynin. 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.

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

Shai Shalev-Shwartz, Alon Gonen, and Ohad Shamir. Large-scale convex minimization with a low-rank constraint. In *International Conference on Machine Learning*, pages 329–336, 2011.

Jie Shen and Ping Li. On the iteration complexity of support recovery via hard thresholding pursuit. In *International Conference on Machine Learning*, pages 3115–3124. PMLR, 2017a.

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

Raghav Somani, Chirag Gupta, Prateek Jain, and Praneeth Netrapalli. Support recovery for orthogonal matching pursuit: upper and lower bounds. In *Advances in Neural Information Processing Systems*, pages 10814–10824, 2018.

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

Xiaotong Yuan, Ping Li, and Tong Zhang. Exact recovery of hard thresholding pursuit. In *Advances in Neural Information Processing Systems*, pages 3558–3566, 2016.

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

Le Zheng, Arian Maleki, Haolei Weng, Xiaodong Wang, and Teng Long. Does ℓ_p-minimization outperform ℓ_1-minimization? *IEEE Transactions on Information Theory*, 63(11):6896–6935, 2017.