Global and Quadratic Convergence of Newton Hard-Thresholding Pursuit

Shenglong Zhou SZ3G14@SOTON.AC.UK
School of Mathematical Sciences
University of Southampton
Southampton SO17 1BJ, UK

Naihua Xiu NHXIU@BJTU.EDU.CN
Department of Applied Mathematics
Beijing Jiaotong University
Beijing, China

Hou-Duo Qi HDQI@SOTON.AC.UK
School of Mathematical Sciences
University of Southampton
Southampton SO17 1BJ, UK

Abstract

Algorithms based on the hard thresholding principle have been well studied with sounding theoretical guarantees in the compressed sensing and more general sparsity-constrained optimization. It is widely observed in existing empirical studies that when a restricted Newton step was used (as the debiasing step), the hard-thresholding algorithms tend to meet halting conditions in a significantly low number of iterations and hence are very efficient. However, the thus obtained Newton hard-thresholding algorithms do not offer any better theoretical guarantees than their simple hard-thresholding counterparts. This annoying discrepancy between theory and empirical studies has been known for some time. This paper provides a theoretical justification for the use of the restricted Newton step. We build our theory and algorithm, Newton Hard-Thresholding Pursuit (`NHTP`), for the sparsity-constrained optimization. Our main result shows that `NHTP` is quadratically convergent under the standard assumption of restricted strong convexity and smoothness. We also establish its global convergence to a stationary point under a weaker assumption. In the special case of the compressive sensing, `NHTP` eventually reduces to some existing hard-thresholding algorithms with a Newton step. Consequently, our fast convergence result justifies why those algorithms perform better than without the Newton step. The efficiency of `NHTP` was demonstrated on both synthetic and real data in compressed sensing and sparse logistic regression.

Keywords: sparse optimization, stationary point, Newton's method, hard thresholding, global convergence, quadratic convergence rate

Running Title: Newton Hard-Thresholding Pursuit

1. Introduction

In this paper, we are mainly concerned with numerical methods for the sparsity constrained optimization

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad f(\mathbf{x}), \qquad \text{s.t.} \qquad \|\mathbf{x}\|_0 \le s, \tag{1}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable, $\|\mathbf{x}\|_0$ is the l_0 norm of $\mathbf{x}$, counting the number of nonzero elements in $\mathbf{x}$, and s is a given integer regulating the sparsity level in $\mathbf{x}$ (i.e., $\mathbf{x}$ is s-sparse). This problem has been well investigated by Bahmani et al. (2013) (from statistical learning perspective) and Beck and Eldar (2013) (from optimization perspective). Problem (1) includes the widely studied Compressive Sensing (CS) (see, e.g., Elad (2010); Zhao (2018)) as a special case:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad f(\mathbf{x}) = f_{\rm CS}(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x}-\mathbf{b}\|^2, \qquad \text{s.t.} \qquad \|\mathbf{x}\|_0 \le s, \tag{2}$$

where A is an $m\times n$ sensing matrix, $\mathbf{b}\in\mathbb{R}^n$ is the observation and $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^n$. Problem (1) has also been a major model in high-dimensional statistical recovery (Agarwal et al. (2010), Negahban et al. (2012)), nonlinear compressive sensing (Blumensath (2013)), and learning model-based sparsity (Bahmani et al. (2016)). An important class of algorithms makes use of the gradient information together with the hard-thresholding technique. We refer to Bahmani et al. (2013), Yuan et al. (2018) (for (1)) and Needell and Tropp (2009), Foucart (2011) (for CS) for excellent examples of such methods and their corresponding theoretical results. In terms of the numerical performance, it has been widely observed that whenever a restricted Newton step is used in the so-called debiasing step, those algorithms appear to take a significantly low number of iterations to converge, see Needell and Tropp (2009), Foucart (2011) and Blumensath (2012). Yet, their theoretical guarantee appears no better than their pure gradient-based counterparts. Hence, there exists an intriguing gap between the exceptional empirical experience and the best convergence theory. This paper aims to provide a theoretical justification for their efficiency by establishing the quadratic convergence of such methods under standard assumptions used in the literature.

In the following, we give a selective review of past work that directly motivated our research, followed by a brief explanation of our general framework that shares a similar structure with several existing algorithms.

1.1 A selective review of past work

There exists a large number of computational algorithms that can be applied to (1). For instance, many of them can be found in Google Scholar from the many papers citing Figueiredo et al. (2007), Needell and Tropp (2009), Elad (2010) and also in the latest book by Zhao (2018). We opt to conduct a bit technical review on a small number of papers that directly motivated our research. Those reviewed papers more or less suggest the following algorithmic framework that largely obeys the principle laid out in Needell and Tropp (2009) and follow the recipes for hard-thresholding methods in Kyrillidis and Cevher (2011). Given the kth iterate $\mathbf{x}^k$, update it to the next iterate $\mathbf{x}^{k+1}$ by the following steps:

$$\left\{\begin{array}{ll} \text{Step 1 (Support Identification Process)}: & T_k = \texttt{SIP}(h(\mathbf{x}^k)),\\ \text{Step 2 (Debiasing)}: & \widetilde{\mathbf{x}}^{k+1} = \arg\min\ \left\{q_k(\mathbf{x}): \quad \mathbf{x}|_{T_k^c} = 0\right\},\\ \text{Step 3 (Pruning)}: & \mathbf{x}^{k+1} \in \mathcal{P}_s(\widetilde{\mathbf{x}}^{k+1}). \end{array}\right. \tag{3}$$

We put the three steps in the perspectives of some existing algorithms and explain the notation involved. For the case of CS, the well-known `CoSaMP` (Compressive Sample Matching

Pursuit) of Needell and Tropp (2009) chose the identification function $h(\mathbf{x})$ to be the gradient function $\nabla f(\mathbf{x})$ and the support identification process `SIP` is chosen to be the union of the best $2s$ support of $h(\mathbf{x}^k)$ (i.e., the $2s$ indices that are from the $2s$ largest elements of $h(\mathbf{x}^k)$ in magnitude) and $\mathrm{supp}(\mathbf{x}^k)$, which are the indices of nonzero elements in $\mathbf{x}^k$. In this case, the number of indices in T_k is below $3s$ (i.e., $|T_k| \le 3s$). In the `HTP` (Hard Thresholding Pursuit) algorithm of Foucart (2011), $h(\mathbf{x})$ is set to be $(\mathbf{x} - \eta \nabla f(\mathbf{x}))$, where $\eta > 0$ is a step size. T_k is chosen to be the best s support of $h(\mathbf{x}^k)$. Hence, $|T_k| = s$. In `AIHT` of Blumensath (2012) (Accelerated Iterative Hard Thresholding), T_k is chosen as in `HTP`. For the general nonlinear function $f(\mathbf{x})$, the `GraSP` of Bahmani et al. (2013) (Gradient Support Pursuit) chose T_k as in `CoSaMP` so that $T_k| \le 3s$. The `GraHTP` of Yuan et al. (2018) (Gradient Hard Thresholding Pursuit) chose T_k as in `HTP` for CS.

Once T_k is chosen, Step 2 (debiasing step) attempts to provide a better estimate for the solution of (1) by solving an optimization problem within a restricted subspace obtained by setting all elements of $\mathbf{x}$ indexed by T_k^c to zero. Here T_k^c is the complementary set of T in $\{1, \ldots, n\}$. Step 3 (pruning step) simply applies the hard-thresholding operator, denoted as $\mathcal{P}_s$, to $\widetilde{\mathbf{x}}^{k+1}$ to retain the best s largest elements in magnitude from $\widetilde{\mathbf{x}}^{k+1}$ and set the rest to zero. The great flexibility in choosing T_k and the objective function $q_k(\mathbf{x})$ in the debiasing step makes it possible to derive various algorithms in literature. For instance, if we choose $T_k = \{1, \ldots, n\}$ (hence $T_k^c = \emptyset$) and $q_k(\mathbf{x})$ to be the first-order approximation of f with a proximal term at $\mathbf{x}^k$:

$$q_k(\mathbf{x}) := f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k),\ \mathbf{x} - \mathbf{x}^k \rangle + \frac{1}{2\eta} \|\mathbf{x} - \mathbf{x}^k\|^2,$$

then we will recover the popular (gradient) hard-thresholding algorithms, see, e.g., Blumensath and Davies (2008, 2009) and Beck and Eldar (2013) for the iterated hard-thresholding algorithms, and Bahmani et al. (2013) for the restricted gradient descent and Yuan et al. (2018) for `GraHTP`. The `CoSaMP` is recovered if T_k is chosen as in `CoSaMP` and $q_k(\mathbf{x}) = f_{\mathrm{CS}}(\mathbf{x})$. More existing methods can be interpreted this way and we omit the details here.

Instead, we focus on the algorithms that make use of the second order approximation in $q_k(\mathbf{x})$. Bahmani et al. (2013) proposed the restricted Newton step, which is equivalent to choosing $q_k(\mathbf{x})$ to be a restricted second-order approximation to $f(\mathbf{x})$ at $\mathbf{x}^k$:

$$q_k(\mathbf{x}) := f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k_{T_k}),\ \mathbf{x}_{T_k} - \mathbf{x}^k_{T_k} \rangle + \frac{1}{2} \langle \mathbf{x}_{T_k} - \mathbf{x}^k_{T_k},\ \nabla^2_{T_k} f(\mathbf{x}^k)(\mathbf{x}_{T_k} - \mathbf{x}^k_{T_k}) \rangle \tag{4}$$

where the notation $\mathbf{x}_{T_k}$ denotes the restriction of $\mathbf{x}$ to the indices in T_k, $\nabla f(\mathbf{x}^k_{T_k})$ is the (partial) gradient of $f(\mathbf{x})$ with respect to the variables indexed by T_k and evaluated at $\mathbf{x}^k_{T_k}$, and $\nabla^2_{T_k} f(\mathbf{x}^k)$ is the principle submatrix of the Hessian matrix $\nabla^2 f(\mathbf{x}^k)$ indexed by T_k. In the case of CS (2), the restricted Newton step is equivalent to minimizing $q_k(\mathbf{x}) = f_{\mathrm{CS}}(\mathbf{x})$ restricted on the subspace defined by $\mathbf{x}|_{T_k^c} = 0$. Hence, the restricted Newton step recovers `CoSaMP`. We note that in both cases, $|T_k| \le 3s$ (i.e., T_k is relatively large). In the `HTP` algorithm, Foucart (2011) managed to choose T_k of size s by making use of the hard thresholding technique, which is further investigated by Blumensath (2012) by the name of accelerated iterative hard-thresholding.

The benefit of using the Newton step has been particularly witnessed for the case of CS. Foucart (2011) compiled convincing numerical evidence that `HTP` took a significantly low number of iterations to converge when proper step-size η is used. Blumensath (2012) confirmed this accelerated performance via `AIHT` when the Newton step is obtained approximately by the conjugate gradient method. A general view from the two papers is that an adaptive step-size rule in the Newton step leads to more efficient performance. However, the existing theoretical guarantee for `HTP` and `AIHT` is no better than their greedy counterparts (e.g., simple iterative hard-thresholding algorithms (`IHT`)). That is, the theory ensures that the distance between each iterate to any given reference (sparse) point is bounded by the sum of two terms. The first term converges linearly and the second term is a fixed approximation error that depends on the choice of the reference point. We refer to the latest paper of Shen and Li (2018) for many of such a result, which is often called statistical error-bound guarantee. The discrepancy between being able to offer better empirical performance than many simple `IHT` algorithms and only sharing similar theoretical guarantee with them invites an intriguing question: why is it so? A positive answer will inevitably provide a deep understanding of the Newton-type HTP algorithms and lead to new powerful algorithms. This is exactly what we are going to achieve in this paper.

A different line of research for (1) was initiated by Beck and Eldar (2013) from an optimization perspective. The convergence results established were drastically contrasting to the statistical error bound result mentioned above. It is proved that any accumulation point of the generated sequence by the `IHT` method is one kind of stationary point (i.e., η-stationarity, to be defined later). In the particular case of CS, the whole sequence converges to an η stationary point under the s-regularity assumption of the sensing matrix A (note: s-regularity of A means that any s columns of A are linearly independent). It is known that $2s$-regularity is a minimal condition that any two s-sparse vectors can be distinguished and it is often assumed by a wide range of the restricted isometry property (RIP) of Candes and Tao (2005). The fact that the s-regularity is weaker than the $2s$-regularity means that many hard-thresholding algorithms actually converge to an η stationary point of (1). Hence, the quality of those algorithms can be measured not only by their statistical error bounds, but also by th quality of the η stationary point (e.g., whether a stationary point is optimal). We refer to Beck and Eldar (2013); Beck and Hallak (2015) for more discussion on the η stationarity in relation to the global optimality.

Similar convergence results to Beck and Eldar (2013) have also been established in the literature of CS. Blumensath and Davies (2010) showed that the normalized `IHT` with an adaptive step-size rule converges to a local minimum of (2) provided that the s-regularity holds. This leads us to ask the following question: when the Newton step is used in combination with `IHT` (such as `HTP` algorithm of Foucart (2011)), whether it is possible to establish the following quadratic convergence result:

$$\mathbf{x}^k \;\to\; \mathbf{x}^* \qquad \text{and} \qquad \|\mathbf{x}^{k+1}-\mathbf{x}^*\| \le c\|\mathbf{x}^k-\mathbf{x}^*\|^2 \quad \text{for sufficiently large } k, \tag{5}$$

where c is a constant solely dependent on the objective function f (independent of the iterates $\mathbf{x}^k$ and its limit $\mathbf{x}^*$). This fast convergence result would justify the stronger numerical performance of various Newton-type methods reviewed in the first part of the subsection. Although, it is expected in optimization that Newton's method (Nocedal and Wright (1999)) will usually lead to quadratic convergence, the problem (1) is not a standard optimization problem and it has a combinatorial nature. Hence, quadratic convergence does not follow from any existing theory from optimization. We also note that both Bahmani et al.

(2013) and Yuan et al. (2018) listed the restricted Newton step as a possible variant for the debiasing step, but it was not theoretically investigated.

We finish this brief review by noticing that there are researches that exclusively studied the role of Newton's method for (1) (see, e.g., Dai and Milenkovic (2009), Yuan and Liu (2017), Chen and Gu (2017)). However, as before, they do not offer any better theoretical guarantees than their simple greedy counterparts. Furthermore, their algorithms do not follow the general framework of (3) and hence their results cannot be used to explain the efficiency of Newton's method that follow (3). In this paper, we will design an algorithm, that also makes uses of a restricted Newton step in the debiasing step (Step 2) and analyse its role in convergence. We will show that our algorithm enjoys the quadratic convergence (5) as well as others. We will particularly relate it to `HTP` of Foucart (2011) so as to justify the strong empirical performance of similar algorithms.

1.2 Our approach and main contributions

The first departure of our proposed Newton step from the one of Bahmani et al. (2013) is that we employ a different quadratic function, denoted as $q_k^N(\mathbf{x})$:

$$\begin{aligned} q_k^N(\mathbf{x}) \;\; &:= \;\; \text{the second-order Taylor expansion of } f(\mathbf{x}) \text{ at } \mathbf{x}^k, \text{ then set } \mathbf{x}|_{T_k^c} = 0 \qquad (6)\\ &= \;\; \langle \nabla_{T_k} f(\mathbf{x}^k),\ \mathbf{x}_{T_k} - \mathbf{x}^k_{T_k}\rangle + \frac{1}{2}\langle \mathbf{x}_{T_k} - \mathbf{x}^k_{T_k},\ \nabla^2_{T_k} f(\mathbf{x}^k)(\mathbf{x}_{T_k} - \mathbf{x}^k_{T_k})\rangle \\ &\quad\; - \langle \mathbf{x}_{T_k},\ \nabla^2_{T_k, T_k^c} f(\mathbf{x}^k)(\mathbf{x}^k_{T_k^c})\rangle + \ \text{(constant term independent of } x), \end{aligned}$$

where $\nabla_{T_k} f(\mathbf{x}^k) := (\nabla f(\mathbf{x}^k))_{T_k}$ and $\nabla^2_{T_k,T_k^c} f(\mathbf{x}^k)$ is the submatrix whose rows and columns are from the Hessian matrix $\nabla^2 f(\mathbf{x}^k)$ indexed by T_k and T_k^c respectively. For the case of CS problem (2), it is straightforward to verify that $q_k^N(\mathbf{x}) = q_k(\mathbf{x})$ in (4). Therefore, the Newton step will become the one used in `CoSaMP` or `HTP` depending on how T_k is selected. In this paper, we choose T_k to be the best s support of $\mathbf{x}^k - \eta\nabla f(\mathbf{x}^k)$. That is, T_k contains a set of indices that define the s largest absolute values in $\mathbf{x}^k - \eta\nabla f(\mathbf{x}^k)$ with η being steplength. For the case of CS, it is the same as that in the algorithm $\texttt{HTP}^\mu$ of Foucart (2011). For the general nonlinear function f, however, $q_k^N(\mathbf{x})$ and $q_k(\mathbf{x})$ are different. The function $q_k(\mathbf{x})$ in (4) is obtained in such a way that we first restrict $f(\mathbf{x})$ to the subspace $\mathbf{x}|_{T_k^c} = 0$ and then approximate it by the second-order Taylor expansion (i.e., restriction and approximation). In contrast, the function $q_k^N(\mathbf{x})$ is obtained in the opposite way. We first approximate $f(\mathbf{x})$ by its second-order Taylor expansion and then restrict the approximation to the subspace $\mathbf{x}_{T_k^c} = 0$ (i.e., approximation and restriction). We will see that our way of construction will allow us quantitatively bound the error $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|$ in terms of $\|\mathbf{x}^k - \mathbf{x}^*\|$, eventually leading to the quadratic convergence in (5).

Our second innovation is to cast the Newton step as a Newton iteration for a nonlinear equation:

$$F_\eta(\mathbf{x},\ T_k) = 0, \qquad (7)$$

where $F_\eta(\cdot, T_k) : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a function reformulated from the η stationarity condition. We defer its technical definition to the next section. A crucial point we would like to make is that this new interpretation of the Newton step offers a fresh angle to examine it and will allow us to develop new analytical tools mainly from optimization perspective and eventually establish the promised quadratic convergence.

It is known that Newton's method is a local method. A commonly used technique for globalization is the line search strategy, which is adopted in this paper. Therefore, we will have a Newton iterate with varying step-size. This agrees with the empirical observation that adaptive step-size in `HTP` often works more efficiently than other variants. Putting together the three techniques (quadratic approximation $q_k^N(\mathbf{x})$, nonlinear equation (7), and the line search strategy) will result in our proposed algorithm termed as Newton Hard-Thresholding Pursuit (`NHTP`) due to the Newton-step and the way how T_k is selected being the two major components of the algorithm. We finish this section by summarizing our major contributions.

(i) We develop the new algorithm `NHTP`, which largely follows the general framework of (3) with Step 3 (pruning) to be replaced by a globalization step. The new step is achieved through the Armijo line search. We will establish its global convergence to an η stationary point under the restricted strong smoothness of f.

(ii) If f is further assumed to be restricted strongly convex at the one of the accumulation points of `NHTP`, the Armijo line search steplength will eventually becomes 1. Consequently, `NHTP` will become the restricted Newton method and leads to its convergence at a quadratic rate. This result successfully extends the classical quadratic convergence result of Newton's method to the sparse case. For the case of CS, `NHTP` reduces to some known algorithms including the `HTP` family of Foucart (2011), with properly chosen step-sizes. The quadratic convergence result resolves the discrepancy between the strong numerical performance of `HTP` (and its alike) and its existing linear convergence guarantee.

(iii) Rigorously establishing the quadratic convergence of `NHTP` is a major contribution of the paper. As far as we know, it is the first paper that establishes both the global and the quadratic convergence for an algorithm that employs both the Newton step and the gradient step (through the hard thresholding operator) for (1). The developed framework of analysis is innovative and will open possibility to prove that other Newton-type HTP methods may also enjoy the quadratic convergence. In our final contribution of this paper, we show experimental results in CS and the logistic regression, with both synthetic and real data, to illustrate the way `NHTP` works.

1.3 Organization

In the next section, we will describe the basic assumptions on the objective function f and their implications. We will also develop a theoretical foundation for the Newton method to be used in a way that it also solves a system of nonlinear equations. Section 3 includes the detailed description of `NHTP` and its global and quadratic convergence analysis. We will particularly discuss its implication to the CS problem and compare with the methods of `HTP` family Foucart (2011). Since some of the proofs are quite technical, we move all of the proofs to the appendices in order to avoid interrupting presentation of the main results. We report our numerical experiments in Section 4 and conclude the paper in Section 5.

2. Assumptions, Stationarity and Interpretation of Newton's Step

2.1 Notation

For easy reference, we list some commonly used notation below. We use ":=" to mean "define". Vectors are treated as column vectors and hence $\mathbf{x}^\top$ is a row vector. For an index set T, let $|T|$ be the number of elements of T and $T^c := \{1, 2, \ldots, \} \setminus T$ be its complementary set. We denote $\text{supp}(\mathbf{x})$ as the support set of $\mathbf{x}$, namely, the set of indices of nonzero elements of $\mathbf{x}$, and $\mathbf{x}_T \in \mathbb{R}^{|T|}$ as the sub vector of $\mathbf{x}$ containing elements indexed on T. Based on this, denote $\nabla_T f(\mathbf{x}) := (\nabla f(\mathbf{x}))_T$. For the Hessian matrix $\nabla^2 f(\mathbf{x})$, denote $\nabla^2_{T,J} f(\mathbf{x})$ its submatrix whose rows and columns are respectively indexed on T and J. In particular, we write $\nabla^2_T f(\mathbf{x})$ as its principle submatrix indexed on T, namely $\nabla^2_T f(\mathbf{x}) := \nabla^2_{T,T} f(\mathbf{x})$. Moreover, we let $\nabla^2_{T:} f(\mathbf{x})$ be its submatrix containing rows indexed by T. We denote $x_{(i)}$ the ith largest (in absolute) element of $\mathbf{x}$. Let $\|\cdot\|$ an $\|\cdot\|_\infty$ respectively denote the Euclidean norm and the infinity norm for a vector. We recall that $\mathcal{P}_s(\mathbf{x})$ denotes the set of all s best supports of $\mathbf{x}$ and it can be obtained by picking the s largest absolute values in $\mathbf{x}$ and setting the remaining to zero. It is important to note that for a given $\mathbf{x}$, it may have multiple s best supports. For example, for $\mathbf{x}^\top = (1, 2, -1, 0)$ and $s = 2$, $\mathcal{P}_s(\mathbf{x})$ contains two s best supports: $(1, 2, 0, 0)$ and $(0, 2, -1, 0)$. In this paper, any of the s best supports would be fine for our analysis. Finally, $\langle \mathbf{x}, \mathbf{y} \rangle$ denotes the standard inner product for $\mathbf{x}, \mathbf{y} \in \Re^n$.

2.2 Basic assumptions and stationarity

In order to study the convergence of various algorithms for the problem (1), some kind of regularities needs to be assumed. They are more or less analogous to the RIP for CS (see Candes and Tao (2005)). Those regularities often share the property of strong restricted convexity/smoothness, see Agarwal et al. (2010), Shalev-Shwartz et al. (2010), Jalali et al. (2011), Negahban et al. (2012), Bahmani et al. (2013), Blumensath (2013), and Yuan et al. (2018). We state the assumptions below in a way that is conducive to our technical proofs.

Definition 1 *(Restricted strongly convex and smooth) Suppose that $f : \mathbb{R}^n \mapsto \mathbb{R}$ is a twice continuously differentiable function whose Hessian is denoted by $\nabla^2 f(\cdot)$. Define*

$$M_{2s}(\mathbf{x}) := \sup_{\mathbf{y} \in \mathbb{R}^n} \left\{ \langle \mathbf{y},\ \nabla^2 f(\mathbf{x}) \mathbf{y} \rangle \ \middle|\ |\text{supp}(\mathbf{x}) \cup \text{supp}(\mathbf{y})| \le 2\text{s},\ \|\mathbf{y}\| = 1 \right\}$$

and

$$m_{2s}(\mathbf{x}) := \inf_{\mathbf{y} \in \mathbb{R}^n} \left\{ \langle \mathbf{y},\ \nabla^2 f(\mathbf{x}) \mathbf{y} \rangle \ \middle|\ |\text{supp}(\mathbf{x}) \cup \text{supp}(\mathbf{y})| \le 2\text{s},\ \|\mathbf{y}\| = 1 \right\}$$

for all s-sparse vectors $\mathbf{x}$.

(i) We say f is restricted strongly smooth (RSS) if there exists a constant $M_{2s} > 0$ such that $M_{2s}(\mathbf{x}) \le M_{2s}$ for all s-sparse vectors $\mathbf{x}$. In this case, we say f is M_{2s}-RSS. f is said to be locally RSS at $\mathbf{x}$ if $M_{2s}(\mathbf{z}) \le M_{2s}$ only holds for those s-sparse vectors $\mathbf{z}$ in a neighborhood of $\mathbf{x}$.

(ii) We say f is restricted strongly convex (RSC) if there exists a constant $m_{2s} > 0$ such that $m_{2s}(\mathbf{x}) \ge m_{2s}$ for all s-sparse vectors $\mathbf{x}$. In this case, we say f is m_{2s}-RSC. f is

said to be locally RSC at $\mathbf{x}$ *if* $m_{2s}(\mathbf{z}) \geq m_{2s}$ *only holds for those* s*-sparse vectors* $\mathbf{z}$ *in a neighborhood of* $\mathbf{x}$*.*

(iii) We say that f *is locally restricted Hessian Lipschitz continuous at an* s*-sparse vector* $\mathbf{x}$ *if there exists a Lipschitz constant* L_f *and a neighborhood* $\mathcal{N}(\mathbf{x})$ *such that*

$$\|\nabla^2_{T:} f(\mathbf{y}) - \nabla^2_{T:} f(\mathbf{z})\| \leq L_f \|\mathbf{y} - \mathbf{z}\|, \quad \forall \ \mathbf{y}, \ \mathbf{z} \in \mathcal{N}(\mathbf{x})$$

for any index set T *with* $|T| \leq s$ *and* $T \supseteq \mathrm{supp}(\mathbf{x})$*.*

Remark 1. We note that the definition of $M_{2s}(\mathbf{x})$ and $m_{2s}(\mathbf{x})$ is taken from the definition of the restricted stable Hessian (RSH) of (Bahmani et al., 2013, Def. 1). If $m_{2s}(\mathbf{x})$ is bounded away from zero, the RSH is equivalent to the RSC and RSS putting together. Under the assumption of twice differentiability, RSS and RSC become that of Negahban et al. (2009), Shalev-Shwartz et al. (2010) and (Yuan et al., 2018, Def. 1). By a standard calculus argument, M_{2s}-RSS implies

$$\left\{ \begin{array}{ll} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq M_{2s} \|\mathbf{x} - \mathbf{y}\|, & \forall \ \mathbf{x}, \mathbf{y}, \ |\mathrm{supp}(\mathbf{x})| \leq s \\ f(\mathbf{x}) - f(\mathbf{y}) - \langle \nabla f(\mathbf{x}), \ \mathbf{x} - \mathbf{y} \rangle \leq \frac{M_{2s}}{2} \|\mathbf{x} - \mathbf{y}\|^2, & |\mathrm{supp}(\mathbf{x}) \cup \mathrm{supp}(\mathbf{y})| \leq 2s. \end{array} \right. \tag{8}$$

The properties in (8) ensure that any optimal solution of (1) must be an η-stationary point, which is a major concept introduced to the spares optimization (1) by Beck and Eldar (2013). An s-sparse vector $\mathbf{x}^*$ is called an η-stationary point of (1) if it satisfies the following relation

$$\mathbf{x}^* \in \mathcal{P}_s(\mathbf{x}^* - \eta \nabla f(\mathbf{x}^*)).$$

Beck and Eldar (2013) called it the L-stationary point because η is very much related to the Lipschitz constant M_{2s} defined in (8). Lemma 2.2 in (Beck and Eldar, 2013) states that an s-sparse vector $\mathbf{x}^*$ is an η-stationary point if and only if

$$\nabla_\Gamma f(\mathbf{x}^*) = 0, \quad \|\nabla_{\Gamma^c} f(\mathbf{x}^*)\|_\infty \leq x^*_{(s)}/\eta. \tag{9}$$

where $\Gamma := \mathrm{supp}(\mathbf{x}^*)$. By invoking the proofs of (Beck and Eldar, 2013, Lemma 2.4 and Thm. 2.2) under the condition of (8), the existence of η-stationary point is ensured.

Theorem 2 *(Existence of* η*-stationary point) (Beck and Eldar, 2013, Thm. 2.2). Suppose that there exists a constant* $M_{2s} > 0$ *such that (8) holds. Let* $\eta < 1/M_{2s}$ *and* $\mathbf{x}^*$ *be an optimal solution of (1). Then*

(i) $\mathbf{x}^*$ *is an* η*-stationary point;*

(ii) $\mathcal{P}_s(\mathbf{x}^* - \eta \nabla f(\mathbf{x}^*))$ *contains exactly one element.*

Consequently, we have

$$\mathbf{x}^* = \mathcal{P}_s(\mathbf{x}^* - \eta \nabla f(\mathbf{x}^*)). \tag{10}$$

We would like to make a few remarks on the significance of Thm. 2.

Remark 2. The characterization of the optimal solution $\mathbf{x}^*$ as a solution of the fixed-point equation (10) immediately suggests a simple iterative procedure:

$$\mathbf{x}^{k+1} \in \mathcal{P}_s(\mathbf{x}^k - \eta \nabla f(\mathbf{x}^k)), \quad k = 0, 1, 2, \ldots.$$

Indeed, for the special case of CS with $\|A\|_2 < 1$, the choice of $\eta = 1 < 1/\|A\|_2 \le 1/M_{2s}$ recovers the `IHT` of Blumensath and Davies (2008). Moreover, any stationary point of $\{\mathbf{x}^k\}$ is an η-stationary point and satisfies the fixed-point equation (10). For the case $\|A\|_2 \ge 1$, the same conclusion holds as long as $\eta < 1/\|A\|_2$, see (Beck and Eldar, 2013, Remark 2).

Remark 3. The fixed-point equation characterization also measures how far an s-sparse point $\mathbf{x}$ is from being an η-stationary point (and hence a possible candidate for an optimal solution of (1)) by computing

$$h(\mathbf{x}, \eta) := \mathrm{dist}(\mathbf{x},\ \mathcal{P}_s(\mathbf{x} - \eta \nabla f(\mathbf{x}))), \tag{11}$$

which defines the shortest Euclidean distance from $\mathbf{x}$ to the set $\mathcal{P}_s(\mathbf{x} - \eta \nabla f(\mathbf{x}))$. If $h(\mathbf{x}, \eta)$ is below a certain tolerance level (e.g., small enough), we may stop at $\mathbf{x}$. This halting criterion is different from those commonly used in CS literature such as in `CoSaMP`, `GraSP`, and `HTP`.

Our third remark is about a differentiable nonlinear equation reformulation of the fixed-point equation (10) and it will give rise to a nice interpretation of the Newton step obtained from minimizing $q_k^N(\mathbf{x})$ in (6). This remark is the main content of the next subsection.

2.3 Nonlinear equations and new interpretation of Newton's step

Given a point $\mathbf{x} \in \mathbb{R}^n$ and $\eta > 0$, we define the collection of all index sets of best s-support of the vector $\mathbf{y} = (\mathbf{x} - \eta \nabla f(\mathbf{x}))$ by

$$\mathcal{T}(\mathbf{x}; \eta) := \left\{ T \subset \{1, \ldots, n\} \;\middle|\; |T| = s, \quad \begin{array}{c} \mathbf{y} := \mathbf{x} - \eta \nabla f(\mathbf{x}) \\ |y_i| \ge y_{(s)} \ \ \forall\, i \in T \end{array} \right\}. \tag{12}$$

That is, each T in $\mathcal{T}(\mathbf{x}; \eta)$ includes s indices that define the locations of the s largest absolute values among the elements of $(\mathbf{x} - \eta f(\mathbf{x}))$. For any given $T \in \mathcal{T}(\mathbf{x}; \eta)$, we define the corresponding nonlinear equation:

$$F_\eta(\mathbf{x}; T) := \begin{bmatrix} \nabla_T f(\mathbf{x}) \\ \mathbf{x}_{T^c} \end{bmatrix} = 0. \tag{13}$$

One advantage of defining the function $F_\eta(\mathbf{x}; T)$ is that it is continuously differentiable with respect to $\mathbf{x}$ once T is selected. Moreover, we have the following characterization of the fixed-point equation (10) in terms of F_η.

Lemma 3 *Suppose $\eta > 0$ is given. A given point $\mathbf{x} \in \mathbb{R}^n$ satisfies the fixed point equation (10) if and only if*

$$F_\eta(\mathbf{x}; T) = 0, \qquad \forall\, T \in \mathcal{T}(\mathbf{x}; \eta).$$

This result is instrumental and crucial to our algorithmic design. Bearing in mind that it is impossible to solve all the nonlinear equations associated with all possible $T \in \mathcal{T}(\mathbf{x}; \eta)$, our hope is that solving one such equation would lead to our desired results. Suppose $\mathbf{x}$ satisfies the equation (13) with one particular choice of T. Under the additional condition

$$|\nabla_i f(\mathbf{x})| \le \frac{1}{\eta} x_{(s)}, \qquad \forall\ i \in T^c \tag{14}$$

we must have

$$\mathbf{x} - \eta \nabla f(\mathbf{x}) = \left[\begin{array}{c} \mathbf{x}_T - \eta \nabla_T f(\mathbf{x}) \\ \mathbf{x}_{T^c} - \eta \nabla_{T^c} f(\mathbf{x}) \end{array} \right] = \left[\begin{array}{c} \mathbf{x}_T \\ -\eta \nabla_{T^c} f(\mathbf{x}) \end{array} \right]$$

and consequently $\mathbf{x}$ is an η-stationary point:

$$\mathbf{x} = \left[\begin{array}{c} \mathbf{x}_T \\ 0 \end{array} \right] \in \mathcal{P}_s(\mathbf{x} - \eta \nabla f(\mathbf{x})).$$

In other words, if we can find a point $\mathbf{x}$ that simultaneously satisfies the conditions (13) and (14) with one particular choice $T \in \mathcal{T}(\mathbf{x}; \eta)$, then we would have reached an η-stationary point, which is the best result we can expect of an optimization algorithm (unless more information is available). Therefore, a natural tolerance function to measure how close $\mathbf{x}$ is from being an η-stationary point is

$$\mathrm{Tol}_\eta(\mathbf{x};\ T) := \|F_\eta(\mathbf{x}; T)\| + \max_{i \in T^c} \Big\{ \max\Big(|\nabla_i f(\mathbf{x})| - x_{(s)}/\eta,\ 0 \Big) \Big\}. \tag{15}$$

It is easy to see that the halting function $h(\mathbf{x}, \eta) = 0$ in (11) implies that there exists $T \in \mathcal{T}(\mathbf{x}; \eta)$ such that $\mathrm{Tol}(\mathbf{x};\ T) = 0$ and vice verse. Our purpose is to quickly find this correct T.

We now turn our attention to the solution methods for (13). Suppose $\mathbf{x}^k$ is the current approximation to a solution of (13) and T_k is chosen from $\mathcal{T}(\mathbf{x}^k; \eta)$. Then Newton's method for the nonlinear equation (7) takes the following form to get the next iterate $\widetilde{\mathbf{x}}^{k+1}$:

$$F'_\eta(\mathbf{x}^k; T_k)(\widetilde{\mathbf{x}}^{k+1} - \mathbf{x}^k) = -F_\eta(\mathbf{x}^k; T), \tag{16}$$

where $F'_\eta(\mathbf{x}^k; T_k)$ is the Jacobian of $F_\eta(\mathbf{x}; T_k)$ at $\mathbf{x}^k$ and it assumes the following form:

$$F'_\eta(\mathbf{x}^k; T_k) = \left[\begin{array}{cc} \nabla^2_{T_k} f(\mathbf{x}^k) & \nabla^2_{T_k, T_k^c} f(\mathbf{x}^k) \\ 0 & I_{n-s} \end{array} \right]. \tag{17}$$

Let $\mathbf{d}_N^k := \widetilde{\mathbf{x}}^{k+1} - \mathbf{x}^k$ be the Newton direction. Substituting (17) into (16) yields

$$\left\{ \begin{array}{rcl} \nabla^2_{T_k} f(\mathbf{x}^k)(\mathbf{d}_N^k)_{T_k} & = & \nabla^2_{T_k, T_k^c} f(\mathbf{x}^k)\mathbf{x}^k_{T_k^c} - \nabla_{T_k} f(\mathbf{x}^k) \\ (\mathbf{d}_N^k)_{T_k^c} & = & -\mathbf{x}^k_{T_k^c}. \end{array} \right. \tag{18}$$

At this point, it is interesting to observe that the next iterate ($\widetilde{\mathbf{x}}^{k+1} = \mathbf{x}^k + \mathbf{d}_N^k$) is exactly the one we would get for the restricted Newton step from minimizing the restricted quadratic function $q_k^N(\mathbf{x})$ in (6). It is because of this exact interpretation of the restricted Newton step that it also drives the equation (13) to be eventually satisfied. In this way, we establish the global convergence to the η-stationarity. However, there are still a number of technical hurdles to overcome. We will tackle those difficulties in the next section.

3. Newton Hard-Thresholding Pursuit and Its Convergence

In this main section, we present our Newton Hard-Thresholding Pursuit (`NHTP`) algorithm, which largely follows the general framework (3), but with distinctive features. We already discussed the choice of T_k (Step 1 in (3)) and the quadratic approximation function q_k^N in (6) (Step 2 in (3)). Since $|T_k| = s$ and $\widetilde{\mathbf{x}}^{k+1}$ obtained is restricted to the subspace $\mathbf{x}|_{T_k^c} = 0$, hence $\mathrm{supp}(\widetilde{\mathbf{x}}^{k+1}) \subseteq T_k$ and the pruning step is not necessary. Instead, we replace it with the globalization step:

$$\text{Step 3' (globalization)} \quad \left\{ \begin{array}{l} \mathbf{x}^{k+1} = \mathcal{G}(\widetilde{\mathbf{x}}^{k+1}) \text{ such that} \\ \mathrm{supp}(\mathbf{x}^{k+1}) \subseteq T_k \quad \text{and} \quad f(\mathbf{x}^{k+1}) \le f(\mathbf{x}^k), \end{array} \right. \tag{19}$$

where $\mathcal{G}$ symbolically represents a globalization process to generate $\mathbf{x}^{k+1}$. We will choose the Armijo line search strategy (see Nocedal and Wright (1999)) to realize this process.

The rest of the section is to consolidate those three steps. We first examine how good is the restricted Newton direction (18) as well as the restricted gradient direction. We note that both directions were proposed in Bahmani et al. (2013). But as far as we know, they are not theoretically studied. We then describe our `NHTP` algorithm and present its global and quadratic convergence under the restricted strong convexity and smoothness.

3.1 Descent properties of the restricted Newton and gradient directions

Our first task is to answer whether the restricted Newton direction $\mathbf{d}_N^k$ from (18) provides a "good" descent direction for $f(\mathbf{x})$ on the restricted subspace $\mathbf{x}|_{T_k^c} = 0$. We have the following result.

Lemma 4 *(Descent inequality of the Newton direction) Suppose $f(\mathbf{x})$ is m_{2s}-restricted strongly convex and M_{2s}-restricted strongly smooth. Given a constant $\gamma \le m_{2s}$ and the step-size $\eta \le 1/(4M_{2s})$, we then have*

$$\left\langle \nabla_{T_k} f(\mathbf{x}^k),\ (\mathbf{d}_N^k)_{T_k} \right\rangle \le -\gamma \|\mathbf{d}_N^k\|^2 + \frac{1}{4\eta} \|\mathbf{x}_{T_k^c}^k\|^2. \tag{20}$$

We note that T_k will eventually identify the true support and $\mathbf{x}_{T_k^c}^k$ should be close to zero when this happens. Hence, the positive term $\|\mathbf{x}_{T_k^c}^k\|^2/(4\eta)$ is eventually negligible. When this happens the restricted Newton direction is able to provide a reasonably good descent direction on the subspace $\mathbf{x}_{T_k^c} = 0$. But in general (e.g., $f(\mathbf{x})$ is not restricted strongly convex), the inequality (20) may not hold and hence $\mathbf{d}_N^k$ may not provide a good descent direction at all. In this case, we opt for the restricted gradient direction (denoted by $\mathbf{d}_g^k$ to distinguish it from $\mathbf{d}_N^k$):

$$\mathbf{d}_g^k := \left[\begin{array}{c} (\mathbf{d}_g^k)_{T_k} \\ (\mathbf{d}_g^k)_{T_k^c} \end{array} \right] = \left[\begin{array}{c} -\nabla_{T_k} f(\mathbf{x}^k) \\ -\mathbf{x}_{T_k^c} \end{array} \right]. \tag{21}$$

This strategy of switching to the gradient direction whenever the Newton direction is not good enough (by certain measure) appears to have been implicitly used by Blumensath

(2012) in the accelerated IHT, where the conjugate gradient (CG) method was used to solve its Newton equation. It was suggested there that using just 3 CG steps would give best numerical performance. We note that the direction obtained by 3 CG steps is a direction between Newton's direction and the gradient direction. but is close to the latter because of the small number of CG steps used. Moreover, this strategy appears very popular and practical in optimization, see, e.g., Nocedal and Wright (1999); Sun et al. (2002); Qi et al. (2003); Qi and Sun (2006); Zhao et al. (2010). Therefore, our search direction $\mathbf{d}^k$ for the globalization step (S3') is defined as follows:

$$\mathbf{d}^k := \begin{cases} \mathbf{d}_N^k, & \text{if the condition (20) is satisfied} \\ \mathbf{d}_g^k, & \text{otherwise.} \end{cases} \tag{22}$$

It is important to note that the choice of γ and η in Lemma 4 is just sufficient for the Newton direction to be used. The inequality (20) may also hold if γ and η violate the required bounds. This has been experienced in our numerical experiments.

Our next result further shows that the search direction $\mathbf{d}^k$ is actually a descent direction for $f(\mathbf{x})$ at $\mathbf{x}^k$ with respect to the full space $\mathbb{R}^n$ provided that η is properly chosen. Suppose we have three constants γ, σ and β such that

$$0 < \gamma \le \min\{1,\ 2M_{2s}\}, \quad 0 < \sigma < 1/2, \quad \text{and} \quad 0 < \beta < 1. \tag{23}$$

They will be used in our `NHTP` algorithm. We note that this choice implies $M_{2s}/\gamma > \sigma$. Define two more constants based on them:

$$\overline{\alpha} := \min\left\{ \frac{1-2\sigma}{M_{2s}/\gamma - \sigma},\ 1 \right\} \quad \text{and} \quad \overline{\eta} := \min\left\{ \frac{\gamma(\overline{\alpha}\beta)}{M_{2s}^2},\ \overline{\alpha}\beta,\ \frac{1}{4M_{2s}} \right\}. \tag{24}$$

Lemma 5 *(Descent property of $\mathbf{d}^k$) Suppose $f(\mathbf{x})$ is M_{2s}-restricted strongly smooth. Let γ, σ and β be chosen as in (23). Suppose $\eta < \overline{\eta}$ and supp$(\mathbf{x}^k) \subseteq T_{k-1}$ (this will be automatically ensured by our algorithm). We then have*

$$\langle \nabla f(\mathbf{x}^k),\ \mathbf{d}^k \rangle \le -\rho \|\mathbf{d}^k\|^2 - \frac{\eta}{2} \|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2, \tag{25}$$

where $\rho > 0$ is given by

$$\rho := \min\left\{ \frac{\gamma - \eta M_{2s}^2}{2},\ \frac{2-\eta}{2} \right\}.$$

Lemma 5 will ensure that our algorithm `NHTP` is well defined.

3.2 `NHTP` and its convergence

Having settled that $\mathbf{d}^k$ is a descent direction of $f(\mathbf{x})$ at $\mathbf{x}^k$, we compute the next iterate along the direction $\mathbf{d}^k$ but restricted to the subspace $\mathbf{x}|_{T_k} = 0$: $\mathbf{x}^{k+1} = \mathbf{x}^k(\alpha_k)$ with α_k being calculated through the Armijo line search and

$$\mathbf{x}^k(\alpha) := \begin{bmatrix} \mathbf{x}_{T_k}^k + \alpha \mathbf{d}_{T_k}^k \\ \mathbf{x}_{T_k^c}^k + \mathbf{d}_{T_k^c}^k \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{T_k}^k + \alpha \mathbf{d}_{T_k}^k \\ 0 \end{bmatrix}, \qquad \alpha > 0. \tag{26}$$

Our algorithm is described in Table 1.

Table 1: Framework of `NHTP`

`NHTP`: Newton Hard-Thresholding Pursuit	
Step 0	Initialize $\mathbf{x}^0$. Choose $\eta, \gamma > 0, \sigma \in (0, 1/2), \beta \in (0, 1)$. Set $k \Leftarrow 0$.
Step 1	Choose $T_k \in \mathcal{T}(\mathbf{x}^k; \eta)$.
Step 2	If $\texttt{Tol}_\eta(\mathbf{x}^k; T_k) = 0$, then stop. Otherwise, go to **Step 3**.
Step 3	Compute the search direction $\mathbf{d}^k$ by (22).
Step 4	Find the smallest integer $\ell = 0, 1, \ldots$ such that $$f(\mathbf{x}^k(\beta^\ell)) \le f(\mathbf{x}^k) + \sigma\beta^\ell \langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle. \quad (27)$$ Set $\alpha_k = \beta^\ell$, $\mathbf{x}^{k+1} = \mathbf{x}^k(\alpha_k)$ and $k \Leftarrow k + 1$, go to **Step 1**.

Remark 4. We will see that `NHTP` has a fast computational performance because of two factors. One is that it terminates in a low number of iterations due to the quadratic convergence (to be proved) and this has been experienced in our numerical experiments. The other is the low computational complexity of each step. For example, for both CS and sparse logistic regression problems, the computational complexity of each step is $\mathcal{O}(s^3 + ms^2 + mn + ms\ell)$, where ℓ is the smallest integer satisfying (27) and it often assumes the value 1. By the way how $\mathbf{x}^k(\alpha)$ is defined, it is guaranteed that supp$(\mathbf{x}^{k+1}) \subseteq T_k$ for all $k = 0, 1, \ldots,$. If $\texttt{Tol}_\eta(\mathbf{x}^k; T_k) = 0$, then $\mathbf{x}^k$ is already an η-stationary point and we should terminate the algorithm. Without loss of any generality, we assume that `NHTP` generates an infinite sequence $\{\mathbf{x}^k\}$ and we will analyse its convergence properties. The line search condition (27) is known as the Armijo line search and ensures a sufficient decrease from $f(\mathbf{x}^k)$ to $f(\mathbf{x}^{k+1})$. Therefore, the two properties in the globalization step (19) is guaranteed, provided that the line search in (27) is successful. This is the main claim of the following result.

Lemma 6 *(Existence and boundedness of α_k) Suppose $f(\mathbf{x})$ is M_{2s}-restricted strongly smooth. Let the parameters γ, σ and β satisfy the conditions in (23) and $\overline{\alpha}$ and $\overline{\eta}$ be defined in (24). Suppose* $\texttt{Tol}_\eta(\mathbf{x}^k; T_k) \neq 0$. *For any α and η satisfying*

$$0 < \alpha \le \overline{\alpha} \qquad \text{and} \qquad 0 < \eta < \min\left\{ \frac{\alpha\gamma}{M_{2s}^2},\ \alpha,\ \frac{1}{4M_{2s}} \right\},$$

it holds

$$f(\mathbf{x}^k(\alpha)) \le f(\mathbf{x}^k) + \sigma\alpha\langle \nabla f(\mathbf{x}^k),\ \mathbf{d}^k \rangle. \tag{28}$$

Consequently, if we further assume that $\eta \le \overline{\eta}$, we have

$$\alpha_k \ge \beta\overline{\alpha} \qquad \forall\ k = 0, 1, \ldots, .$$

It is worth noting that the objective function is only assumed to be restricted strongly smooth (not necessarily to be restricted strongly convex). Lemma 6 does not only ensure the existence of α_k that satisfies the line search condition (27), but also guarantees that α_k is always bounded away from zero by a positive margin $\beta\overline{\alpha}$. This boundedness property will in turn ensure that `NHTP` will converge. Our first result on convergence is about a few quantities approaching zero.

Lemma 7 *(Converging quantities) Suppose $f(\mathbf{x})$ is M_{2s}-restricted strongly smooth. Let the parameters γ, σ and β satisfy the conditions in (23) and $\overline{\eta}$ be defined in (24). We further assume that $\eta \le \overline{\eta}$. Then the following hold.*

(i) $\{f(\mathbf{x}^k)\}$ *is a nonincreasing sequence and if* $\mathbf{x}^{k+1} \neq \mathbf{x}^k$*, then* $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$.

(ii) $\|\mathbf{x}^{k+1} - \mathbf{x}^k\| \to 0$;

(iii) $\|F_\eta(\mathbf{x}^k;\ T_k)\| \to 0$;

(iv) $\|\nabla_{T_k} f(\mathbf{x}^k)\| \to 0$ *and* $\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\| \to 0$.

Those converging quantities are the basis for our main results below. They also justify the halting conditions that we will use in our numerical experiments.

Theorem 8 *(Global convergence) Suppose $f(\mathbf{x})$ is M_{2s}-restricted strongly smooth. Let the parameters γ, σ and β satisfy the conditions in (23) and $\overline{\eta}$ be defined in (24). We further assume that $\eta \le \overline{\eta}$. Then the following hold.*

(i) Any accumulation point, say $\mathbf{x}^*$, of the sequence $\{\mathbf{x}^k\}$ is an η-stationary point of (1). If f is a convex function, then for any given reference point $\mathbf{x}$ we have

$$f(\mathbf{x}^*) \le f(\mathbf{x}) + \frac{x^*_{(s)}}{\eta}\|\mathbf{x}_{\Gamma^c_*}\|_1, \tag{29}$$

where $\Gamma_* := supp(\mathbf{x}^*)$.

(ii) If $\mathbf{x}^*$ is isolated, then the whole sequence converges to $\mathbf{x}^*$. Moreover, we have the following characterization on the support of $\mathbf{x}^*$.

(a) If $\|\mathbf{x}^*\|_0 = s$, then

$$supp(\mathbf{x}^*) = supp(\mathbf{x}^k) = T_k \quad \textit{for all sufficiently large } k.$$

(b) If $\|\mathbf{x}^*\|_0 < s$, then

$$supp(\mathbf{x}^*) \subseteq supp(\mathbf{x}^k) \cap T_k \quad \textit{for all sufficiently large } k.$$

Remark 5. Under the assumption of f being restricted strongly smooth, `NHTP` shares the most desirable convergence property (i.e., to η-stationary point) of the iterative hard-thresholding algorithm of Beck and Eldar (2013). If f is assumed to be convex, then (29) implies that for any given $\epsilon > 0$, there exists neighborhood $\mathcal{N}(\mathbf{x}^*)$ of $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \le f(\mathbf{x}) + \epsilon$ for any $\mathbf{x} \in \mathcal{N}(\mathbf{x}^*)$. In particular, if $\|\mathbf{x}^*\|_0 = s$, then $\mathbf{x}^*$ is a local minimum of (1). If $\|\mathbf{x}^*\|_0 < s$ (so that $x^*_{(s)} = 0$), then $\mathbf{x}^*$ is a global optimum of (1).

It achieves more. If the generated sequence converges to $\mathbf{x}^*$, the support of $\mathbf{x}^*$ is eventually identified as T_k provided that the sparse level of $\mathbf{x}^*$ is s. If $\|\mathbf{x}^*\|_0 < s$, its support would be eventually included in T_k. When specialized to the CS problem (2) with s-regularity, the whole sequence $\{\mathbf{x}^k\}$ will convergence to one point $\mathbf{x}^*$. This is because that any η-stationary point of the CS problem under the s-regularity is isolated, see (Beck and Eldar, 2013, Lemma 2.1 and Corollary 2.1). Our next result implies that under the $2s$-regularity, the whole sequence $\{\mathbf{x}^k\}$ converges to $\mathbf{x}^*$ at a quadratic rate.

Theorem 9 *(Quadratic convergence) Suppose all conditions as in Thm. 8 hold. Let $\mathbf{x}^*$ be one of the accumulation points of $\{\mathbf{x}^k\}$. We further assume $f(\mathbf{x})$ is m_{2s}-restricted strongly convex in a neighborhood of $\mathbf{x}^*$. If $\gamma \le \min\{1,\ m_{2s}\}$ and $\eta \le \overline{\eta}$, then the following hold.*

(i) The whole sequence $\{\mathbf{x}^k\}$ converges to $\mathbf{x}^*$, which is necessarily an η-stationary point.

(ii) The Newton direction is accepted for sufficiently large k.

(iii) If we further assume that f is locally restricted Hessian Lipschitz continuous at $\mathbf{x}^*$ with the Lipschitz constant L_f. The line search steplength becomes unity eventually and the convergence rate of $\{\mathbf{x}^k\}$ to $\mathbf{x}^*$ is quadratic. That is, there exists an iteration index k_0 such that

$$\alpha_k \equiv 1, \qquad \|\mathbf{x}^{k+1} - \mathbf{x}^*\| \le \frac{L_f}{2m_{2s}} \|\mathbf{x}^k - \mathbf{x}^*\|^2, \qquad \forall\ k \ge k_0. \tag{30}$$

Moreover, for sufficiently large k, we have

$$\|F_\eta(\mathbf{x}^{k+1};\ T_{k+1})\| \le \frac{2m_{2s}^3}{L_f\sqrt{M_{2s}^2+1}} \|F_\eta(\mathbf{x}^k;\ T_k)\|^2.$$

Remark 6. Taking into account of Lemma 7(iii) that $\|F_\eta(\mathbf{x}^k;\ T_k)\|$ converges to 0, Thm. 9(iii) asserts that it converges at a quadratic rate. Compared with the quadratic convergence (30), the quadratic convergence in $\|F_\eta(\mathbf{x}^k;\ T_k)\|$ has the advantage that it is computationally verifiable. The quantity is also a major part of our stopping criterion in monitoring $\mathrm{Tol}_\eta(\mathbf{x}^k;\ T_k)$ of (15), see Sect. 4.

3.3 The case of CS

We use this part to demonstrate the application and implication of our main convergence results to the CS problem (2). We will also discuss the similarities to and differences from the existing algorithms, in particular the `HTP` family of Foucart (2011). The purpose is to show that there is a wide range of choices for the parameters that will lead to quadratic convergence. This is best done in terms of the restricted isometry constant (RSC) of the sensing matrix A. We recall from Candes and Tao (2005) that RSC δ_s is the smallest $\delta \ge 0$ such that

$$(1-\delta)\|\mathbf{x}\|^2 \le \|A\mathbf{x}\|^2 \le (1+\delta)\|\mathbf{x}\|^2 \qquad \forall\ \|\mathbf{x}\|_0 \le s.$$

We will use δ_{2s}, which is assumed to be positive throughout. For this setting, we have

$$m_{2s} = 1 - \delta_{2s}, \quad M_{2s} = 1 + \delta_{2s}, \quad \text{and} \quad \mu_{2s} := \frac{M_{2s}}{m_{2s}},$$

where, μ_{2s} is known as the $2s$-restricted stable Hessian coefficient of A in Bahmani et al. (2013).

For simplicity, we choose a particular set of parameters used in our `NHTP` to illustrate our results (many other choices are also possible). Let

$$\beta = \frac{1}{4}, \quad \gamma = m_{2s}, \quad \sigma = r(w)\mu_{2s} \ \text{ with } \ r(w) := \frac{1-w}{2\mu_{2s} - w}, \quad 0 < w < 1.$$

It is elementary to verify that the function $r(w)$ is strictly decreasing over the interval $0 \le w \le 1$. Hence, $r(w) < r(0) = 1/(2\mu_{2s})$. Therefore, $\sigma < 1/2$ for any choice w between 0 and 1. This set of parameter choices certainly satisfies the condition (23). We now calculate $\overline{\alpha}$ and $\overline{\eta}$ defined in (24). Let

$$\phi(r) := \frac{1-2\sigma}{M_{2s}/\gamma - \sigma} = \frac{1 - 2r\mu_{2s}}{(1-r)\mu_{2s}} = 2 + \frac{1-2\mu_{2s}}{(1-r)\mu_{2s}}.$$

Clearly, $\phi(r)$ is strictly decreasing in r and we have $\phi(r) < \phi(0) = 1/\mu_{2s} \le 1$. The definition of $\overline{\alpha}$ chooses

$$\overline{\alpha} = \phi(r) = \frac{2 - 2r(w)\mu_{2s}}{(1-r(w))\mu_{2s}} = \frac{w}{\mu_{2s}}.$$

We now turn our attention to $\overline{\eta}$. Since $\gamma/M_{2s}^2 = 1/(\mu_{2s}M_{2s}) < 1$ and $\beta = 1/4$, we have

$$\begin{aligned} \overline{\eta} &= \frac{\gamma}{M_{2s}^2}\overline{\alpha}\beta = \frac{1}{4} \times \frac{1}{\mu_{2s}} \times \frac{1}{M_{2s}} \times \overline{\alpha} \\ &\ge \frac{1}{4} \times \frac{1}{\mu_{2s}} \times \frac{1}{2} \times \frac{w}{\mu_{2s}} \qquad \text{(because } M_{2s} \le 2\text{)} \\ &= \frac{w}{8\mu_{2s}^2}. \end{aligned}$$

Direction application of Thm. 9 yields the following corollary.

Corollary 10 *Suppose the RIC $\delta_{2s} > 0$ and the parameters of `NHTP` are chosen as follows:*

$$\beta = \frac{1}{4}, \quad \sigma = r(w)\mu_{2s}, \quad \gamma = m_{2s}, \quad \eta \le \frac{w}{8\mu_{2s}^2} \quad \text{with } \ 0 < w < 1. \tag{31}$$

Then `NHTP` is well-defined. In particular, the Newton direction d_N^k is always accepted as the search direction in (22) at each iteration. Moreover, `NHTP` enjoys all the three convergence results in Thm. 9.

Remark 7. (On RIC conditions) In the literature of CS, a benchmark condition (for theoretical investigation) often takes the form $\delta_t \le \delta_*$ with t being an integer. Suppose $\delta_{2s} \le \delta_*$. It is easy to define and derive the following.

$$\begin{aligned} m_{2s}^* &:= 1 - \delta_* \le 1 - \delta_{2s} = m_{2s} \\ \mu_{2s}^* &:= \frac{1+\delta_*}{1-\delta_*} \ge \frac{1+\delta_{2s}}{1-\delta_{2s}} = \mu_{2s} \\ \overline{\eta}^* &:= \frac{w}{8(\mu_{2s}^*)^2} \le \frac{w}{8\mu_{2s}^2}. \end{aligned}$$

Therefore, in the selection of the parameters in (31), μ_{2s} and m_{2s} can be respectively replaced by μ^*_{2s} and m^*_{2s}, and η can be chosen to satisfy $\eta \le \overline{\eta}^*$. In the scenario of Garg and Khandekar (2009) where $\delta_* = 1/3$, with $w = 0.5$ we could choose the parameters as $\beta = 1/4$, $\gamma = 2/3$, $\sigma = 2/7$ and $\eta = 1/64$. This set of choices would ensure `NHTP` converges quadratically under the RIP condition $\delta_{2s} \le \delta_* = 1/3$.

Remark 8. (On Newton's direction) That the Newton direction is always accepted at each iteration is because the inequality (20) is always satisfied with the parameter selection in (31) (its proof can be patterned after that for Thm. 9(iii)). Therefore, the Newton direction $\mathbf{d}_N^k$ at each iteration takes the form:

$$
\begin{aligned}
\left(\mathbf{d}_N^k\right)_{T_k} &= \left(A_{T_k}^\top A_{T_k}\right)^{-1}\left(A_{T_k}^\top A_{T_k^c}\mathbf{x}^k_{T_k^c} - A_{T_k}^\top(A\mathbf{x}^k - \mathbf{b})\right) \\
&= \left(A_{T_k}^\top A_{T_k}\right)^{-1}\left(A_{T_k}^\top A_{T_k^c}\mathbf{x}^k_{T_k^c} - A_{T_k}^\top(A_{T_k}\mathbf{x}^k_{T_k} + A_{T_k^c}\mathbf{x}^k_{T_k^c} - \mathbf{b})\right) \\
&= -\mathbf{x}^k_{T_k} + \left(A_{T_k}^\top A_{T_k}\right)^{-1} A_{T_k}^\top \mathbf{b}.
\end{aligned}
$$

Since the unit line search steplength $\alpha_k = 1$ is always accepted for all k sufficiently large (say, $k \ge k_0$), we have

$$
\mathbf{x}^{k+1} = \mathbf{x}^k(\alpha_k) = \mathbf{x}^k(1) = \begin{bmatrix} \mathbf{x}^k_{T_k} + (\mathbf{d}^k_N)_{T_k} \\ 0 \end{bmatrix} = \begin{bmatrix} \left(A_{T_k}^\top A_{T_k}\right)^{-1} A_{T_k}^\top \mathbf{b} \\ 0 \end{bmatrix}
$$

Equivalently,

$$
\mathbf{x}^{k+1} = \arg\min\left\{\|\mathbf{b} - A\mathbf{z}\| : \quad \mathrm{supp}(\mathbf{z}) \subseteq T_k\right\}.
$$

Consequently, `NHTP` eventually (when $k \ge k_0$) becomes $\texttt{HTP}^\eta$ of Foucart (2011):

$$
\texttt{HTP}^\eta : \quad \begin{cases} T_k & = \left\{\text{the best } s \text{ support of } (\mathbf{x}^k - \eta\nabla f(\mathbf{x}^k)\right\}, \quad (\text{i.e., } T_k \in \mathcal{T}(\mathbf{x}^k;\eta)) \\ \mathbf{x}^{k+1} & = \arg\min\left\{\|\mathbf{b} - A\mathbf{z}\| : \quad \mathrm{supp}(\mathbf{z}) \subseteq T_k\right\}. \end{cases}
$$

(Foucart, 2011, Prop. 3.2) states that $\texttt{HTP}^\eta$ will converge provided that $\eta\|A\|^2 < 1$, which is ensured when $\eta < 1/M_{2s}$. Our choice $\eta \le w/(8\mu_{2s}^2)$ apparently satisfies this condition. Hence, `NHTP` eventually enjoys all the good properties stated for $\texttt{HTP}^\eta$ under the same conditions assumed in Foucart (2011) as long as the η (note: μ is used in Foucart (2011) instead of η) used there does not clash with our choice.

Since the Newton direction is always accepted as the search direction every iteration, one may wonder why we did not just use the unit steplength $\alpha_k = 1$. We note that `NHTP` does not just seek for the next iterate satisfying $f(\mathbf{x}^{k+1}) \le f(\mathbf{x}^k)$, it also requires it to deduce a sufficient decrease by the quantity $\alpha_k\sigma\langle\nabla f(\mathbf{x}^k),\ \mathbf{d}^k\rangle$, which is proportional to the steplength α_k. Newton's direction $\mathbf{d}_N^k$ with the unit steplength may not provide this proportional decrease and hence the unit steplength cannot be accepted in this case (but the unit steplength will be eventually accepted). In contrast, the `HTP` family algorithms of Foucart (2011) only require a decrease $f(\mathbf{x}^{k+1}) \le f(\mathbf{x}^k)$. It is interesting to note that, in optimization, one of the guidelines in designing a descent algorithm is to ensure it deduces a sufficient decrease every iteration (see Nocedal and Wright (1999)) in order to achieve desirable convergence properties.

Remark 9. (On the gradient direction) When the information on μ_{2s} and m_{2s} is difficult to estimate, the choice of (31) may not be possible. On the one hand, those are the sufficient conditions for the Newton direction to be accepted. Numerical experiments show that Newton's direction is often accepted with a wide range of parameter choices. On the other hand, we have the restricted gradient direction to rescue if the Newton direction is not deemed to be good enough in terms of the condition (20). The resulting algorithm still enjoys the global convergence in Thm. 8 even if all search directions are of gradients. It is interesting to note that a restricted gradient method was also proposed in Foucart (2011) and is referred to as fast `HTP`. We describe this algorithm (with just one gradient iteration each step) in terms of our technical terminologies.

$$\texttt{FHTP}^\eta: \quad \left\{ \begin{array}{lll} \widetilde{\mathbf{x}}^{k+1} & = & \mathcal{P}_s(\mathbf{x}^k - \eta \nabla f(\mathbf{x}^k)) \\ T_{k+1} & \in & \mathcal{T}(\widetilde{\mathbf{x}}^{k+1}; \eta) \\ \mathbf{x}^{k+1}_{T_{k+1}} & = & \left(\widetilde{\mathbf{x}}^{k+1} - t_{k+1} \nabla f(\widetilde{\mathbf{x}}^{k+1}) \right)_{T_{k+1}} \quad \text{and} \quad \mathbf{x}^{k+1}_{T^c_{k+1}} = 0, \end{array} \right.$$

where t_{k+1} can be set to 1 or chosen adaptively. Despite it being also shown to enjoy similar convergence properties as $\texttt{HTP}^\eta$ in Foucart (2011), it does not fall within the framework (3) and (19). A noticeable difference is that $\texttt{FHTP}^\eta$ solves two optimization problems each step: one for $\widetilde{\mathbf{x}}^{k+1}$ and the other for $\mathbf{x}^{k+1}_{T_{k+1}}$. It would be interesting to see how the convergence analysis conducted in this paper can be extended to $\texttt{FHTP}^\eta$.

4. Numerical Experiments

In this part, we show experimental results of `NHTP` in CS (Sect. 4.1) and sparse logistic regression (Sect. 4.2) on both synthetic and real data. A general conclusion is that `NHTP` is capable of producing solutions of high quality and is very fast when benchmarked against six leading solvers from compressed sensing and three solvers from sparse logistic regression. All experiments were conducted by using MATLAB (R2018a) on a desktop of 8GB memory and Inter(R) Core(TM) i5-4570 3.2Ghz CPU.

We first describe how `NHTP` was set up. We initialize `NHTP` with $\mathbf{x}^0 = 0$ if $\nabla f(0) \neq 0$ and $\mathbf{x}^0 = \mathbf{1}$ if $\nabla f(0) = 0$. Parameters are set as $\sigma = 10^{-4}/2, \beta = 0.5$. For γ, theoretically any positive $\gamma \leq m_{2s}$ is fine, but in practice to guarantee more steps using Newton directions, it is supposed to be relatively small (De Luca et al., 1996; Facchinei and Kanzow, 1997). Thus we choose $\gamma = \gamma_k$ with updating

$$\gamma_k = \left\{ \begin{array}{ll} 10^{-10}, & \text{if} \quad \mathbf{x}^k_{T^c_k} = 0, \\ 10^{-4}, & \text{if} \quad \mathbf{x}^k_{T^c_k} \neq 0. \end{array} \right.$$

For parameter η, in spite of that Theorem 9 has suggested to set $0 < \eta < \overline{\eta}$, it is still difficult to fix a proper one since M_{2s} is not easy to compute in general. Overall, we choose to update η adaptively. Typically, we use the following rule: starting η with a fixed scalar associated with the dimensions of a problem and then update it as,

$$\begin{aligned}\eta_0 &= \frac{10(1+s/n)}{\min\{10,\ln(n)\}} > 1,\\ \eta_{k+1} &= \begin{cases}\eta_k/1.05, & \text{if } \mathrm{mod}(k,10)=0 \text{ and } \|F_{\eta_k}(\mathbf{x}^k;T_k)\| > k^{-2},\\ 1.05\eta_k, & \text{if } \mathrm{mod}(k,10)=0 \text{ and } \|F_{\eta_k}(\mathbf{x}^k;T_k)\| \le k^{-2},\\ \eta_k, & \text{otherwise.}\end{cases}\end{aligned}$$

where mod $(k,10)=0$ means k is a multiple of 10. We terminate our method if at kth step it meets one of the following conditions:

- $\mathrm{Tol}_{\eta_k}(\mathbf{x}^k;\ T_k) \le 10^{-6}$, where $\mathrm{Tol}_\eta(\mathbf{x};\ T)$ is defined as (15);
- $|f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k)| < 10^{-6}(1+|f(\mathbf{x}^k)|)$.
- k reaches the maximum number (e.g., 2000) of iterations.

4.1 Compressed Sensing

Compressed sensing (CS) has seen revolutionary advances both in theory and algorithms over the past decade. Ground-breaking papers that pioneered the advances are (Donoho, 2006; Candès et al., 2006; Candes and Tao, 2005). The model is described as in (2)

a) Testing examples. We will focus on the exact recovery $\mathbf{b} = A\mathbf{x}$ by utilizing the sensing matrix A chosen as in (Yin et al., 2015; Zhou et al., 2016).

Example 1 (Gaussian matrix) *Let $A \in \mathbb{R}^{m\times n}$ be a random Gaussian matrix with each column $A_j, j \in N_n$ being identically and independently generated from the standard normal distribution. We then normalize each column such that $\|A_j\| = 1$. Finally, the 'ground truth' signal $\mathbf{x}^*$ and the measurement $\mathbf{b}$ are produced by the following pseudo Matlab codes:*

$$\mathbf{x}^* = \mathtt{zeros}(n,1), \quad \Gamma = \mathtt{randperm}(n), \quad \mathbf{x}^*(\Gamma(1:s)) = \mathtt{randn}(s,1), \quad \mathbf{b} = A\mathbf{x}^*. \tag{32}$$

Example 2 (Partial DCT matrix) *Let $A \in \mathbb{R}^{m\times n}$ be a random partial discrete cosine transform (DCT) matrix generated by*

$$A_{ij} = \cos(2\pi(j-1)\psi_i), \quad i = 1,\ldots,m, \quad j = 1,\ldots,n$$

where $\psi_i, i = 1,\ldots,m$ is uniformly and independently sampled from $[0,1]$. We then normalize each column such that $\|A_j\| = 1$ with $\mathbf{x}^*$ and $\mathbf{b}$ being generated the same way as in Example 1.

b) Benchmark methods. There exists a large number of numerical methods for the CS problem (2). It is beyond the scope of this paper to compare them all. We selected six state-of-the-art methods. They are `HTP` (Foucart, 2011)[1], `NIHT` (Blumensath and Davies, 2010)[2], `GP` (Blumensath and Davies, 2008)[2], `OMP` (Pati et al., 1993; Tropp and Gilbert,

---

[1] `HTP` is available at: *https://github.com/foucart/HTP.*

[2] `NIHT`, `GP` and `OMP` are available at *https://www.southampton.ac.uk/engineering/about/staff /tb1m08.page#software.* We use the version `sparsify_0_5` in which `NIHT`, `GP` and `OMP` are called `hard_l0_Mterm`, `greed_gp` and `greed_omp`.

2007)[2], `CoSaMP` (Needell and Tropp, 2009)[3] and `SP` (Dai and Milenkovic, 2009)[3]. For `HTP`, set `MaxNbIter`=1000 and `mu`=`'NHTP'`. For `NIHT`, the maximum iteration `'maxIter'` is set as 1000 and `M` $= s$. For `GP` and `OMP`, the `'stopTol'` is set as 1000. For `CoSaMP` and `SP`, set `tol`$= 10^{-6}$ and `maxiteration`$=$ 1000. Notice that the first three methods prefer solving sensing matrix A with unit columns, which is the reason for us to normalize each generated A in Example 1 and Example 2. Let $\mathbf{x}$ be the solution produced by a method. We say a recovery of this method is successful if $\|\mathbf{x} - \mathbf{x}^*\| < 0.01\|\mathbf{x}^*\|$.

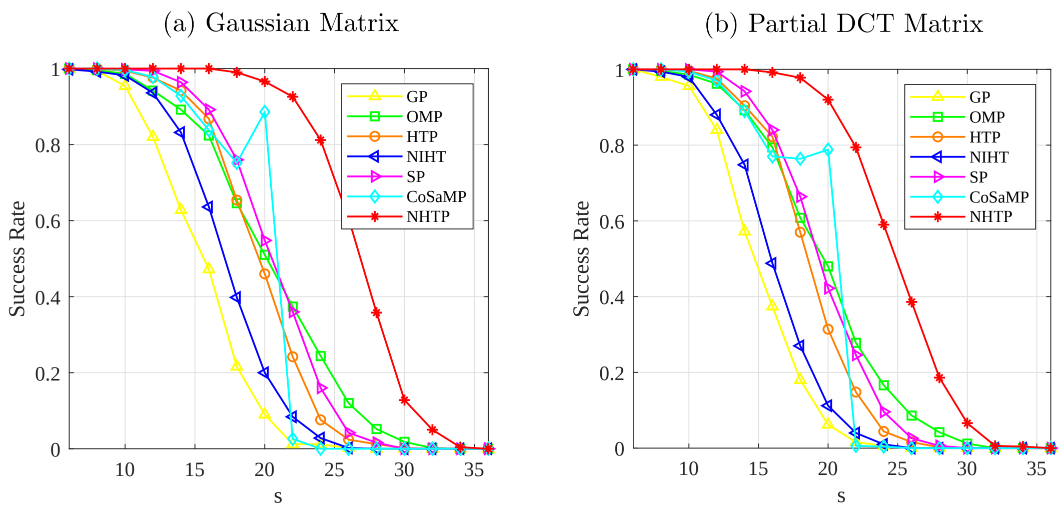

Figure 1: Success rates. $n = 256, m = \lceil n/4 \rceil, s \in \{6, 8, \cdots, 36\}$.

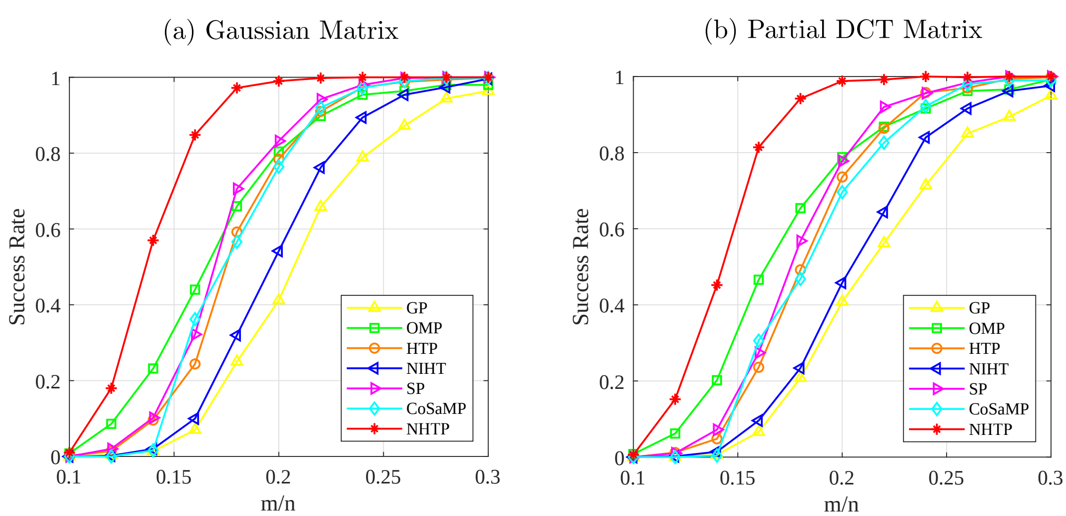

Figure 2: Success rates. $n = 256, s = \lceil 0.05n \rceil, m = \lceil rn \rceil$ with $r \in \{0.1, 0.12, \cdots, 0.3\}$.

[3] `CoSaMP` and `SP` are available at: *http://media.aau.dk/null_space_pursuits/2011/07/a-few-corrections-to-cosamp-and-sp-matlab.html.*

c) Numerical comparisons. We begin with running 500 independent trials with fixed $n = 256, m = \lceil n/4 \rceil$ and recording the corresponding success rates (which is defined by the percentage of the number of successful recoveries over all trails) at sparsity levels s from 6 to 36, where $\lceil a \rceil$ is the smallest integer that is no less than a. From Fig. 1, one can observe that for both Example 1 and Example 2, `NHTP` yielded the highest success rate for each s. For example, when $s = 22$ for Gaussian matrix, our method still obtained 90% successful recoveries while the other methods only guaranteed less than 40% successful ones. Moreover, `OMP`, `SP` and `HTP` generated similar results, and `GP` and `NIHT` always came the last. Next we run 500 independent trials with fixing $n = 256, s = \lceil 0.05n \rceil$ but varying $m = \lceil rn \rceil$ where $r \in \{0.1, 0.12, \cdots, 0.3\}$. It is clearly to be seen that the larger m is, the easier the problem becomes to be solved. This is illustrated by Fig. 2. Again `NHTP` outperformed the others due to highest success rate for each s, and `GP` and `NIHT` still came the last.

To see the accuracy of the solutions and the speed of these seven methods, we now run 50 trials for each kind of matrices with higher dimensions n increasing from 5000 to 25000 and keeping $m = \lceil n/4 \rceil, s = \lceil 0.01n \rceil, \lceil 0.05n \rceil$. Specific results produced by these seven methods are recorded in Tables 2 and 3. Our method `NHTP` always obtained the most accurate recovery, with accuracy order of 10^{-14} or higher, followed by `HTP`. `NIHT` was stable at achieving the solutions with accuracy of order 10^{-7}. Moreover, `GP` and `OMP` rendered solutions as accurate as those by `NHTP` when $s = \lceil 0.01n \rceil$, but yielded inaccurate ones when $s = \lceil 0.05n \rceil$, which means that these two methods worked well when the solution is very sparse. In contrast, `SP` and `CoSaMP` always generated results with worst accuracy. When it comes to the computational speed in Table 3, `NHTP` is the fastest for most of the cases. The fast convergence of `NHTP` becomes more superior in high dimensional data setting. For example, when $n = 25000$ and $s = \lceil 0.05n \rceil$, 6.58 seconds by `NHTP` against 36.93 seconds by `HTP`, which is the fastest method among the other five methods. `GP` and `OMP` always ran the slowest. In addition, we also compared seven algorithms on Example 1, but omitted all the related results since they were similar to those of Example 2

Table 2: Average absolute error $\|\mathbf{x} - \mathbf{x}^*\|$ for Example 2.

s	n	GP	OMP	HTP	NIHT	SP	CoSaMP	NHTP
	5000	2.78e-15	2.40e-15	2.97e-15	2.42e-7	1.12e-5	1.12e-5	4.59e-16
	10000	5.21e-15	4.75e-15	5.70e-15	3.26e-7	3.59e-5	3.59e-5	1.10e-15
$\lceil 0.01n \rceil$	15000	7.05e-15	7.07e-15	7.36e-15	4.28e-7	4.25e-5	4.25e-5	1.39e-15
	20000	9.49e-15	9.06e-15	9.47e-15	4.88e-7	6.56e-5	6.56e-5	1.88e-15
	25000	1.15e-14	1.12e-14	1.11e-14	5.32e-7	1.78e-4	1.78e-4	2.47e-15
	5000	1.28e-03	1.40e-03	1.26e-14	4.80e-7	9.07e-5	9.07e-5	5.94e-15
	10000	7.91e-04	3.56e-04	2.44e-14	6.86e-7	1.77e-4	1.77e-4	1.18e-14
$\lceil 0.05n \rceil$	15000	1.10e-03	6.20e-04	3.57e-14	8.54e-7	2.11e-4	2.11e-4	1.76e-14
	20000	9.43e-04	3.33e-04	4.87e-14	9.80e-7	3.53e-4	3.53e-4	2.39e-14
	25000	1.24e-03	5.57e-04	5.94e-14	1.01e-6	2.59e-4	2.59e-4	2.86e-14

Table 3: Average CPU time (in seconds) for Example 2.

s	n	GP	OMP	HTP	NIHT	SP	CoSaMP	NHTP
	5000	0.69	0.48	0.09	0.30	0.07	0.05	0.06
	10000	4.47	3.70	0.33	1.21	0.31	0.25	0.16
$\lceil 0.01n \rceil$	15000	14.57	13.41	0.74	2.96	0.96	0.86	0.37
	20000	32.70	30.46	1.34	5.53	2.30	2.00	0.65
	25000	68.94	67.13	2.49	37.03	20.11	4.18	1.13
	5000	3.52	3.22	0.23	1.29	0.90	1.43	0.28
	10000	19.84	23.55	1.52	4.63	6.02	15.56	0.79
$\lceil 0.05n \rceil$	15000	67.79	77.30	7.25	10.43	23.03	60.87	2.20
	20000	151.28	177.00	18.02	18.70	58.20	148.83	3.49
	25000	312.57	363.44	36.93	78.69	153.52	307.53	6.58

4.2 Sparse Logistic Regression

Sparse logistic regression (SLR) has drawn extensive attention since it was first proposed by Tibshirani (1996). Same as (Bahmani et al., 2013), we will address the so-called ℓ_2 norm regularized sparsity constrained logistic regression (SCLR) model, namely,

$$\min_{\|\mathbf{x}\|_0 \leq s} \ \ell(\mathbf{x}) + \mu\|\mathbf{x}\|_2^2 \qquad \text{with} \qquad \ell(\mathbf{x}) := \frac{1}{m}\sum_{i=1}^{m}\left\{\ln(1+e^{\langle \mathbf{a}_i, \mathbf{x}\rangle}) - b_i\langle \mathbf{a}_i, \mathbf{x}\rangle\right\}, \tag{33}$$

where $\mathbf{a}_i \in \mathbb{R}^n, b_i \in \{0,1\}, i = 1,\ldots,m$ are respectively given m features and responses/labels, and $\mu > 0$ (e.g. $\mu = 10^{-6}/m$). The employment of a regularization was well justified because otherwise *'one can achieve arbitrarily small loss values by tending the parameters to infinity along certain directions'* (see (Bahmani et al., 2013)). This is the reason why we will only focus on (33).

d) Testing examples. We will test three types of data sets. The first two are synthetic and the last one is from a real database. One synthetic data is adopted from (Lu and Zhang, 2013), (Pan et al., 2017) with the features $[\mathbf{a}_1 \ \cdots \ \mathbf{a}_m]$ being generated identically and independently. The other is the same as Agarwal et al. (2010) or Bahmani et al. (2013) who have considered independent features with each $\mathbf{a}_i$ being generated by an autoregressive process (Hamilton, 1994).

Example 3 (Independent Data (Lu and Zhang, 2013; Pan et al., 2017)) *To generate data labels $\mathbf{b} \in \{0,1\}^m$, we first randomly separate $\{1,\ldots,m\}$ into two parts I and I^c and set $b_i = 0$ for $i \in I$ and $b_i = 1$ for $i \in I^c$. Then the feature data is produced by*

$$\mathbf{a}_i = y_i v_i \mathbf{1} + \mathbf{w}_i, \qquad i = 1,\ldots,m$$

with $\mathbb{R} \ni \ v_i \sim \mathcal{N}(0,1)$, $\mathbb{R}^n \ni \mathbf{w}_i \sim \mathcal{N}(0,\mathcal{I}_n)$ and $\mathcal{N}(0,\mathcal{I}_n)$ is the normal distribution with zero mean and the identity covariance. Since the sparse parameter $\mathbf{x}^* \in \mathbb{R}^n$ is unknown, different sparsity levels will be tested.

Example 4 (Correlated Data (Agarwal et al., 2010; Bahmani et al., 2013)) *The sparse parameter* $\mathbf{x}^* \in \mathbb{R}^n$ *has* s *nonzero entries drawn independently from the standard Gaussian distribution. Each data sample* $\mathbf{a}_i = [a_{i1} \ \cdots \ a_{in}]^\top, i = 1, \ldots, m$ *is an independent instance of the random vector generated by an autoregressive process (see Hamilton, 1994)*

$$a_{i(j+1)} = \theta a_{ij} + \sqrt{1-\eta^2} v_{ij}, \quad j = 1, \ldots, n-1,$$

with $a_{i1} \sim \mathcal{N}(0,1)$, $v_{ij} \sim \mathcal{N}(0,1)$ *and* $\theta \in [0,1]$ *being the correlation parameter. The data labels* $\mathbf{y} \in \{0,1\}^m$ *are then drawn randomly according to the Bernoulli distribution with*

$$\Pr\{y_i = 0|\mathbf{a}_i\} = \left[1 + e^{\langle \mathbf{a}_i, \mathbf{x}^* \rangle}\right]^{-1}, \qquad i = 1, \ldots, m.$$

Example 5 (Real data) *This example comprises of seven real data sets for binary classification. They are* `colon-cancer`[1], `arcene`[2], `newsgroup`[3], `news20.binary`[1], `duke breast-cancer`[1], `leukemia`[1], `rcv1.binary`[1], *which are summarized in the following table, where the last three data sets have testing data. Moreover, as described in the website*[1], *for the four data with small sample sizes:* `colon-cancer`, `arcene`, `duke breast-cancer` *and* `leukemia`, *sample-wise normalization has been conducted so that each sample has mean zero and variance one, and then feature-wise normalization has been conducted so that each feature has mean zero and variance one. For the rest four data with larger sample sizes, they are feature-wisely scaled to* $[-1, 1]$. *All* -1*s in classes* $\mathbf{b}$ *are replaced by 0.*

Data name	m samples	n features	training size m_1	testing size m_2
`colon-cancer`	62	2000	62	0
`arcene`	100	10000	100	0
`newsgroup`	11314	777811	11314	0
`news20.binary`	19996	1355191	19996	0
`duke breast-cancer`	42	7129	38	4
`leukemia`	72	7129	38	34
`rcv1.binary`	40242	47236	20242	20000

e) Benchmark methods. Since there are numerous leading solvers that have been proposed to solve SLR problems, we again only focus on those dealing with the ℓ_2 norm regularized SCLR. We select three solvers: `GraSP` (Bahmani et al., 2013)[4], `NTGP` (Yuan and Liu, 2014) and `IIHT` (Pan et al., 2017). Notice that all those methods are used to solve ℓ_2 norm regularized SCLR model (33) with $\mu = 10^{-6}/m$. Except for `IIHT`, which only used the first order information such as objective values or gradients, the other three methods exploit second order information of the objective function. `NTGP` integrates Newton directions into some steps, and `GraSP` takes advantage of the Matlab built-in function: `minFunc` which calls a Quasi-Newton strategy. For `GraSP`, if we use its defaults parameters, it would be

[1] `https://www.csie.ntu.edu.tw/` $\sim$ `cjlin/libsvmtools/datasets/`
[2] `http://archive.ics.uci.edu/ml/index.php`
[3] `https://web.stanford.edu/` $\sim$ `hastie/glmnet_matlab/`
[4] `http://sbahmani.ece.gatech.edu/GraSP.html`

less likely to meet its stopping criteria before the number of iteration reaching the maximal one. Compared with other three methods, which all generate a sequence with decreasing objective function values, the objective function value at each iteration by `GraSP` fluctuated greatly. Therefore, we set an extra stopping criterion for `GraSP`: $f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) < 10^{-6}$. And if $f(\mathbf{x}^k) < f(\mathbf{x}^{k+1})$, then terminate it and output $\mathbf{x}^k$. For `NTGP`, to facilitate its computational speed, we set `maxIter`=20 for outer loops, and `maxIter_sub`=50 and `optTol_sub` $= 10^{-3}$ for inner loops. For `IIHT`, we keep its default parameters.

For both Example 3 and Example 4, we run 500 independent trials if $n < 10^3$ and 50 independent trials otherwise, and report the average logistic loss $\ell(\mathbf{x})$ and CPU time to demonstrate the performance of each method.

f) Numerical comparisons. For Example 3, we begin with testing each method for the case $n = 256$ and $m = \lceil n/5 \rceil$ with varying sparsity levels s from 10 to 30. From Fig. 3(a), one can observe that `IIHT` rendered the best $\ell(\mathbf{x})$ when $s = 10$ and `NHTP` performed the best $\ell(\mathbf{x})$ when $s > 10$. And importantly, the value $\ell(\mathbf{x})$ produced by `NHTP` for each instance is far smaller than others, with order about 10^{-6}. We then test the case $n = 256, s = \lceil 0.05n \rceil$ and $m = \lceil rn \rceil$ with varying $r \in \{0.05, 0.1, \cdots, 0.7\}$. From Fig. 3(b), $\ell(\mathbf{x})$ generated by `NHTP` is the lowest when the sample size was relatively small, and it gradually approached to the values similar to those obtained by the others. `IIHT` performed the best in terms of $\ell(\mathbf{x})$ when $m/n > 0.2$ and `GraSP` always rendered the highest loss.

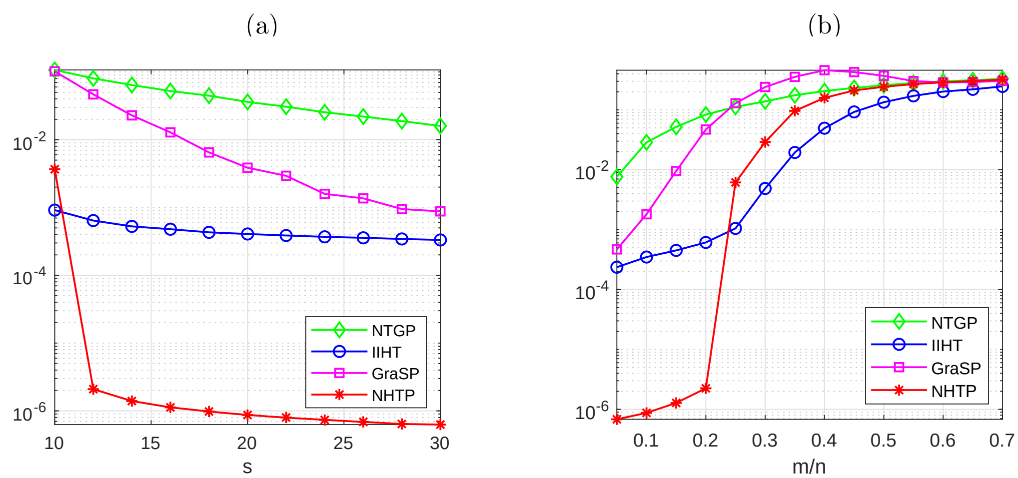

Figure 3: Average logistic loss $\ell(\mathbf{x})$ of four methods for Example 3.

When the size of example is becoming relatively large, the picture is significant different. Hence we now run 50 independent trials with higher dimensions n increasing from 10000 to 40000 and keeping $m = \lceil n/5 \rceil, s = \lceil 0.01n \rceil, \lceil 0.05n \rceil$. As presented in Table 4, when $s = \lceil 0.01n \rceil$, `IIHT` produced the lowest $\ell(\mathbf{x})$, followed by `NHTP` which was the fastest. But when $s = \lceil 0.05n \rceil$, `NHTP` outperformed others in terms of $\ell(\mathbf{x})$ with order of 10^{-7} which was much better than others. The time used by `NHTP` is also significantly less than the others, for example, 25.41s by `NHTP` vs. 619.5s by `GraSP` when $n = 40000$.

Table 4: Average logistic loss $\ell(\mathbf{x})$ and CPU time (in seconds) for Example 3.

s	n	$\ell(\mathbf{x})$				CPU Time			
		NTGP	IIHT	GraSP	NHTP	NTGP	IIHT	GraSP	NHTP
	10000	2.39e-1	1.43e-1	2.44e-1	2.26e-1	8.403	1.723	0.488	0.313
	15000	2.48e-1	1.37e-1	2.39e-1	2.28e-1	17.81	3.307	0.974	0.457
	20000	2.35e-1	1.36e-1	2.36e-1	2.20e-1	32.61	6.245	1.862	0.842
$\lceil 0.01n \rceil$	25000	2.25e-1	1.29e-1	2.30e-1	2.11e-1	52.99	8.913	3.006	1.372
	30000	2.24e-1	1.24e-1	2.30e-1	2.07e-1	76.31	14.15	4.309	2.140
	35000	2.21e-1	1.23e-1	2.29e-1	2.08e-1	149.7	21.84	16.08	2.875
	40000	2.18e-1	1.21e-1	2.32e-1	2.05e-1	466.1	29.12	804.2	3.923
	10000	4.58e-2	4.76e-4	4.97e-3	6.50e-7	9.931	3.094	1.795	0.987
	15000	4.05e-2	4.69e-4	7.77e-3	3.32e-7	26.34	6.218	4.069	2.442
	20000	4.10e-2	4.80e-4	8.24e-3	6.32e-7	51.29	10.69	5.695	4.315
$\lceil 0.05n \rceil$	25000	4.56e-2	4.90e-4	6.06e-3	4.77e-7	54.96	15.93	8.964	7.004
	30000	4.17e-2	4.92e-4	6.49e-3	6.89e-7	85.22	23.54	11.79	11.15
	35000	3.95e-2	4.89e-4	6.46e-3	6.65e-7	182.1	35.97	24.34	17.25
	40000	3.84e-2	4.92e-4	7.54e-3	5.81e-7	551.1	55.00	619.5	25.41

For Example 4, it is related to the parameter θ. We only report the results for $\theta = 1/2$ since the comparisons of all methods are similar for each fixed $\theta \in (0, 1)$. Again we first fix $n = 256, m = \lceil n/5 \rceil$ and vary sparsity levels s from 10 to 30. As shown in Fig. 4 (a), NHTP yielded the smallest logistic loss when $s > 12$, followed by IIHT. We then fix $n = 256, s = \lceil 0.05n \rceil$ and change the sample size $m = \lceil rn \rceil$, where $r \in \{0.05, 0.1, 0.15, \cdots, 0.7\}$. From Fig. 4(b), NHTP outperformed others when the sample size was relatively small such as $m/n < 0.2$, while IIHT performed best in terms of $\ell(\mathbf{x})$ when $m/n \geq 0.2$.

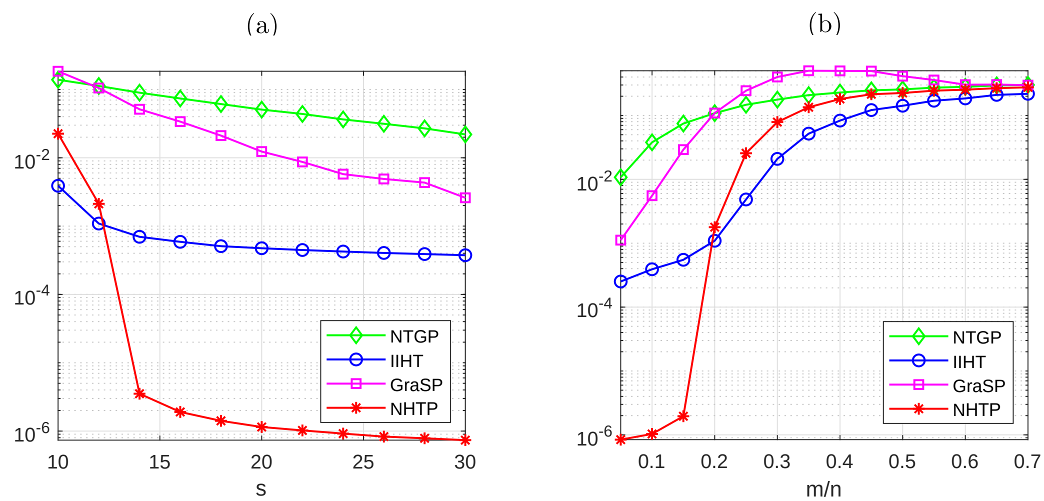

Figure 4: Average logistic loss $\ell(\mathbf{x})$ of four methods for Example 4.

When the size of the example is becoming relatively large, the picture again is significant different. We run 50 independent trials with higher dimensions n increasing from 10000 to 40000 and keeping $m = \lceil n/5 \rceil, s = \lceil 0.01n \rceil, \lceil 0.05n \rceil$. As presented in Table 5, when $s = \lceil 0.01n \rceil$ `IIHT` indeed provided the best logistic loss and comparable to ours. However, `NHTP` was significantly faster than `IIHT`. Clearly, under the case of $s = \lceil 0.05n \rceil$, `NHTP` offered the far lowest $\ell(\mathbf{x})$ with order of 10^{-6} and CPU time with 22.46 seconds against 721 seconds from `GraSP` when $n = 40000$.

Table 5: Average logistic loss $\ell(\mathbf{x})$ and CPU time (in seconds) for Example 4.

s	n	$\ell(\mathbf{x})$				CPU Time			
		`NTGP`	`IIHT`	`GraSP`	`NHTP`	`NTGP`	`IIHT`	`GraSP`	`NHTP`
	10000	1.87e-1	5.68e-2	1.93e-1	1.51e-1	8.338	4.394	0.471	0.245
	15000	1.81e-1	4.07e-2	1.73e-1	1.25e-1	19.72	7.156	1.403	0.702
	20000	1.61e-1	3.39e-2	1.64e-1	9.94e-2	36.68	10.74	2.370	1.194
$\lceil 0.01n \rceil$	25000	1.62e-1	2.62e-2	1.61e-1	9.84e-2	54.37	16.51	3.800	1.922
	30000	1.63e-1	2.75e-2	1.63e-1	9.59e-2	124.4	40.11	18.83	6.067
	35000	1.58e-1	2.09e-2	1.52e-1	8.73e-2	179.1	44.47	199.2	8.257
	40000	1.59e-1	2.14e-2	1.57e-1	8.87e-2	423.4	46.13	639.4	19.47
	10000	7.59e-2	6.02e-4	2.18e-2	1.54e-6	9.101	3.426	1.875	0.880
	15000	7.95e-2	6.15e-4	2.02e-2	1.67e-6	20.40	7.426	4.316	2.140
	20000	7.84e-2	5.93e-4	2.34e-2	1.55e-6	34.91	12.51	6.394	4.015
$\lceil 0.05n \rceil$	25000	7.96e-2	5.97e-4	2.44e-2	1.65e-6	54.41	19.03	8.921	6.590
	30000	7.76e-2	6.00e-4	2.04e-2	1.58e-6	107.2	29.95	16.57	10.09
	35000	7.74e-2	6.01e-4	2.18e-2	1.61e-6	137.3	45.71	26.05	16.10
	40000	7.89e-2	5.90e-4	2.41e-2	1.58e-6	305.8	70.83	721.0	22.46

Now we compare these four methods on solving real data in Example 5. For each method, we demonstrate its performance on instances with varying s. We first illustrate the performance of each method on solving those data without testing data sets. As presented in Fig. 5, we have the following observations:

- For `colon-cancer`, `NHTP` obtained the smallest $\ell(\mathbf{x})$ followed by `IIHT`. While `GraSP` ran the fastest and `NTGP` performed the slowest.

- For `arcene`, `IIHT` and `NHTP` generated best $\ell(\mathbf{x})$ when $s < 80$ and $s \geq 80$ respectively. And the latter consumed the smallest CPU time.

- For `newsgroup`, `NHTP` outperformed others in terms of the smallest $\ell(\mathbf{x})$ and CPU time. `NTGP` rendered the worst logistic loss and `IIHT` ran the slowest.

- For `news20.binary`, `GraSP` performed unstably, yet achieving best $\ell(\mathbf{x})$ for some cases such as $s \leq 1300$. `NTGP` still produced the highest logistic loss. As for computational speed, `NHTP` was the fastest and `IIHT` was the slowest.

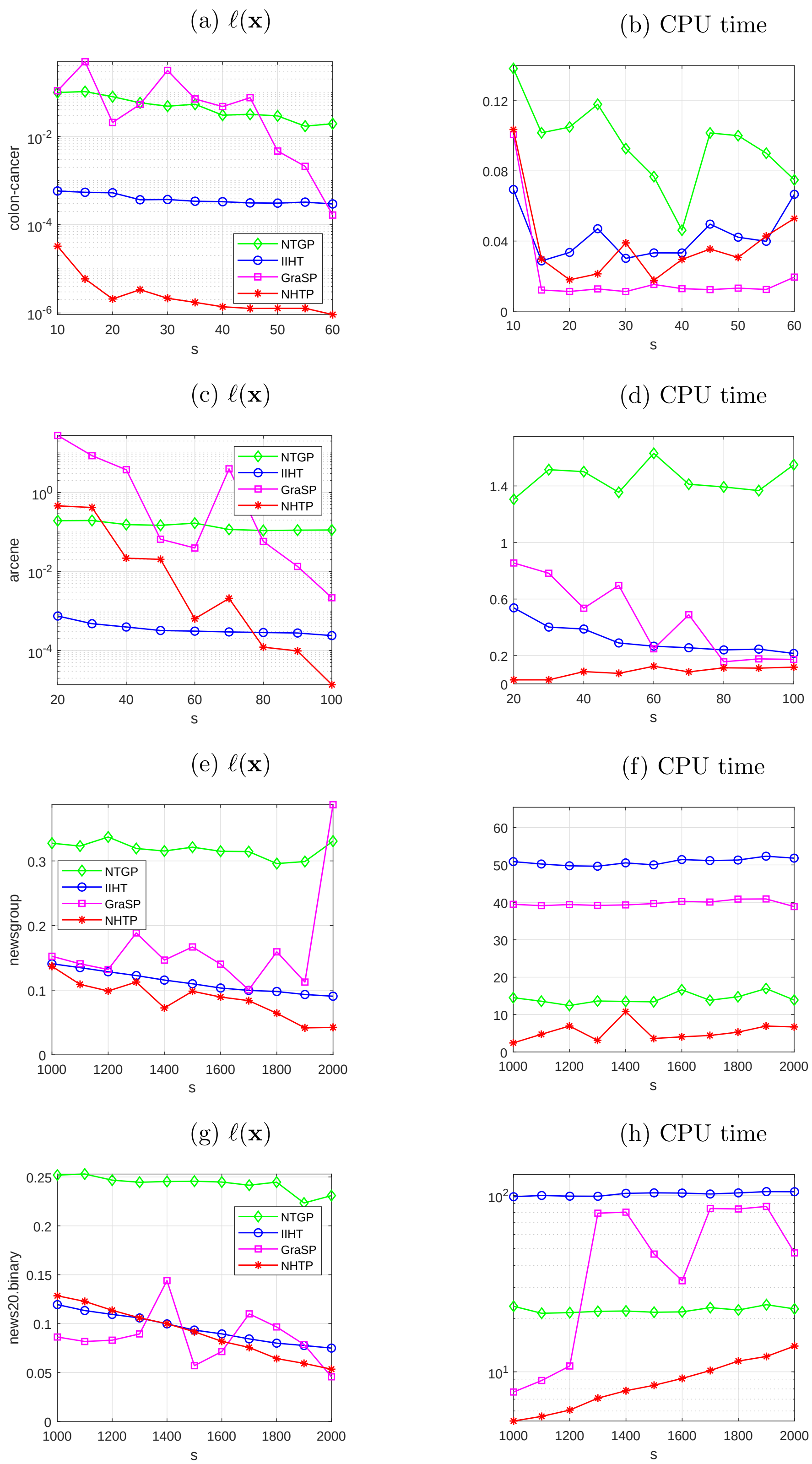

Figure 5: Logistic loss $\ell(\mathbf{x})$ and CPU time of four methods for Example 5.

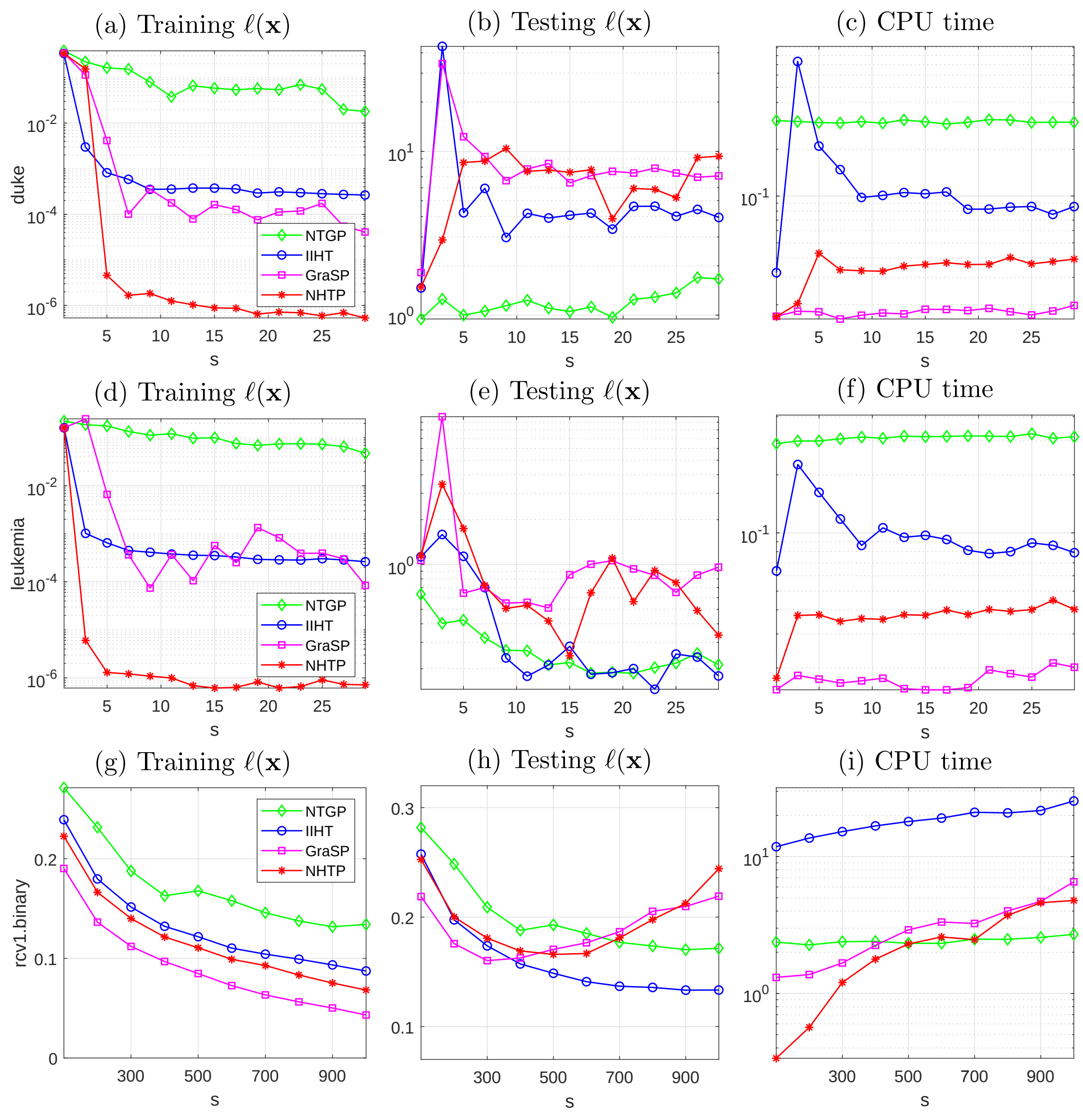

Figure 6: Logistic loss $\ell(\mathbf{x})$ and CPU time of four methods for Example 5.

Next we illustrate the performance of each method on solving those data with testing data sets. As shown in Fig. 6, some comments are able to be made as follows:

- For `duke breast-cancer`, along with increasing s, $\ell(\mathbf{x})$ on training data obtained by `NHTP` dropped significantly, with order 10^{-6}. By contrast, `NTGP` stabilized at above 10^{-2}. When it comes to the testing data, apparently `NTGP` yielded the best $\ell(\mathbf{x})$, followed by `IIHT`. It seems that the higher $\ell(\mathbf{x})$ on training data was solved by a method, the lower $\ell(\mathbf{x})$ on testing data would be provided. For CPU time, `GraSP` behaved the fastest, followed by `NHTP`, `IIHT` and `NTGP`.

- For `leukemia`, the performance of each method was similar to that on `duke breast-cancer` data. A slightly difference was that `NTGP` no more offered the best $\ell(\mathbf{x})$ on testing data as `IIHT` generated the best ones for some s.

- For `rcv1.binary`, `GraSP` performed the best $\ell(\mathbf{x})$ on training data, followed by our method. Again `NTGP` came the last. It is obvious that `IIHT` got the smallest $\ell(\mathbf{x})$ on testing data when $s \geq 400$, while `GraSP` produced the best ones otherwise. For CPU time, `NHTP` and `NTGP` was the most efficient when $s \geq 600$ and $s > 600$ respectively.

5. Conclusion

There exists numerous papers that use a restricted Newton step to accelerate methods belonging to hard-thresholding pursuits. This results in the method of Newton hard-thresholding pursuit. On the one hand, existing empirical experience shows significance acceleration when Newton's step is employed . On the other hand, existing theory for such methods does not offer any better statistical guarantee than the simple hard thresholding counterparts. The discrepancy between the superior empirical performance and the no-better theoretical guarantee has been well documented in the case of CS problem (2) and it invites further theory for justification.

In this paper, we develop a new `NHTP`, which makes use of the strategy "*approximation and restriction*" to obtain the truncated approximation within a subspace. This is in contrast to the popular strategy "*restriction and approximation*". We note that both strategies lead to the same Newton step in the case of CS. We further cast the resulting Newton step as a Newton iteration for a nonlinear equation. This new interpretation of the Newton step provides a new route for establishing its quadratic convergence. Finally, we used the Armijo line search to globalize the method. Extensive numerical experiments confirm the efficiency of the proposed method. The global and quadratic convergence theory for `NHTP` offers a theoretical justification why such methods are more efficient than their simple hard thresholding counterparts.

We expect that our algorithmic framework will make it possible to study quadratic convergence of existing `NHTP` based on the strategy of "*restriction and approximation*". A plausible approach would be to regard such method as an inexact version of our `NHTP`. Technically, it would involve quantifying/controlling the inexactness so as to ensure the quadratic convergence to hold. We leave the investigation in future work.

Appendix A. Identities and Inequalities for Proofs

Due to the restricted fashion of `NHTP`, we need to keep tracking the indices belonging to the subspace $\mathbf{x}|_{T_k} = 0$ and also those fall out of this subspace. To simplify our proofs, we will use a few more abbreviations and derive some identities and inequalities associated with the Newton direction $\mathbf{d}_N^k$. The sequence $\{\mathbf{x}^k\}$ used is generated by `NHTP`.

(a) Simplification of Newton's equation (18). We first define

$$J_k := T_{k-1} \setminus T_k, \qquad H_k := \nabla^2_{T_k} f(\mathbf{x}^k), \qquad G_k := \nabla^2_{T_k, J_k} f(\mathbf{x}^k). \tag{34}$$

We also have the following easy observation:

$$\mathrm{supp}(\mathbf{x}^k) \subseteq T_{k-1}, \quad |T_k| = |T_{k-1}| = s, \quad \text{and} \quad |T_k \setminus T_{k-1}| = |T_{k-1} \setminus T_k| = |J_k|. \tag{35}$$

It is important to note that $\mathbf{x}^k_{T_k^c}$ is also s-sparse. This is because for any $i \notin T_{k-1}$, $x_i^k = 0$ (because $\mathrm{supp}(\mathbf{x}^k) \subseteq T_{k-1}$),

$$\mathbf{x}^k_{T_k^c} = \begin{bmatrix} \mathbf{x}^k_{T_k^c \cap T_{k-1}} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{x}^k_{T_{k-1} \setminus T_k} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{x}^k_{J_k} \\ 0 \end{bmatrix}, \tag{36}$$

and $|J_k| \le |T_{k-1} \setminus T_k| \le |T_{k-1}| = s$. We emphasize that J_k captures all nonzero elements in $\mathbf{x}^k_{T_k^c}$. Therefore, we will see more J_k instead of T_k^c being used in our derivation below. This observation leads to the simplified Newton equation of (18):

$$\begin{cases} H_k(\mathbf{d}_N^k)_{T_k} &= G_k \mathbf{x}^k_{J_k} - \nabla_{T_k} f(\mathbf{x}^k) \\ (\mathbf{d}_N^k)_{T_k^c} &= -\mathbf{x}^k_{T_k^c} = -\begin{bmatrix} \mathbf{x}^k_{J_k} \\ 0 \end{bmatrix}. \end{cases} \tag{37}$$

An important feature to note is that the vectors $(\mathbf{d}_N^k)_{T_k}$, $(\mathbf{d}_N^k)_{T_k^c}$, $\mathbf{x}^k_{J_k}$ are all s-sparse. Putting together, at each iteration, we only involve vectors that do not exceed $2s$-sparsity. This is the reason why our assumptions are always on $2s$-restricted properties of f.

(b) An identity on the Newton direction. This involves a string of equalities as follows. We write $\mathbf{d}^k$ for $\mathbf{d}_N^k$ because there is no danger to cause any confusion.

$$\begin{aligned}
& \langle \mathbf{d}^k_{T_k \cup J_k}, \nabla^2_{T_k \cup J_k} f(\mathbf{x}^k) \mathbf{d}^k_{T_k \cup J_k} \rangle \qquad \text{(note } T_k \cap J_k = \emptyset\text{)} \\
= \; & (\mathbf{d}^k_{T_k \cup J_k})^\top \begin{bmatrix} H_k, & G_k \\ G_k^\top, & \nabla^2_{J_k} f(\mathbf{x}^k) \end{bmatrix} \mathbf{d}^k_{T_k \cup J_k} \\
= \; & (\mathbf{d}^k_{T_k \cup J_k})^\top \begin{bmatrix} H_k \mathbf{d}^k_{T_k} + G_k \mathbf{d}^k_{J_k} \\ \left(G_k^\top \, , \; \nabla^2_{J_k} f(\mathbf{x}^k) \right) \mathbf{d}^k_{T_k \cup J_k} \end{bmatrix} \\
\overset{(37)}{=} \; & (\mathbf{d}^k_{T_k \cup J_k})^\top \begin{bmatrix} -\nabla_{T_k} f(\mathbf{x}^k) \\ \left(G_k^\top, \; \nabla^2_{J_k} f(\mathbf{x}^k) \right) \mathbf{d}^k_{T_k \cup J_k} \end{bmatrix} \\
= \; & -\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}^k_{T_k} \rangle + \langle G_k \mathbf{d}^k_{J_k}, \mathbf{d}^k_{T_k} \rangle + \langle \mathbf{d}^k_{J_k}, \nabla^2_{J_k} f(\mathbf{x}^k) \mathbf{d}^k_{J_k} \rangle \\
\overset{(37)}{=} \; & -\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}^k_{T_k} \rangle - \langle H_k \mathbf{d}^k_{T_k} + \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}^k_{T_k} \rangle + \langle \mathbf{d}^k_{J_k}, \nabla^2_{J_k} f(\mathbf{x}^k) \mathbf{d}^k_{J_k} \rangle \\
= \; & -2\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}^k_{T_k} \rangle - \langle H_k \mathbf{d}^k_{T_k}, \mathbf{d}^k_{T_k} \rangle + \langle \mathbf{d}^k_{J_k}, \nabla^2_{J_k} f(\mathbf{x}^k) \mathbf{d}^k_{J_k} \rangle.
\end{aligned}$$

This leads to our identity:

$$\begin{aligned} 2\langle \nabla_{T_k} f(\mathbf{x}^k), \; \mathbf{d}^k_{T_k} \rangle \; = \; & -\langle \mathbf{d}^k_{T_k \cup J_k}, \nabla^2_{T_k \cup J_k} f(\mathbf{x}^k) \mathbf{d}^k_{T_k \cup J_k} \rangle \\ & -\langle H_k \mathbf{d}^k_{T_k}, \; \mathbf{d}^k_{T_k} \rangle + \langle \mathbf{d}^k_{J_k}, \; \nabla^2_{J_k} f(\mathbf{x}^k) \mathbf{d}^k_{J_k} \rangle. \end{aligned} \tag{38}$$

(c) An inequality on the gradient sequence. The role of T_k is like a working active set that is designed to identify the true support of an optimal solution. Its complementary set T_k^c is handled in such a way to make sure the next iterate $\mathbf{x}^{k+1}$ has zeros on T_k^c. To achieve this, in both the Newton direction $\mathbf{d}_N^k$ and the gradient direction $\mathbf{d}_g^k$ we set

$$\left(\mathbf{d}_N^k\right)_{T_k^c} = \left(\mathbf{d}_g^k\right)_{T_k^c} = -\mathbf{x}_{T_k^c}^k.$$

Let $\mathbf{d}^k$ be either $\mathbf{d}_N^k$ or $\mathbf{d}_g^k$. It follows from (36) that

$$\begin{aligned} &\|\mathbf{x}_{T_k^c}^k\| = \|\mathbf{x}_{J_k}^k\| = \|\mathbf{d}_{J_k}^k\| = \|\mathbf{d}_{T_k^c}^k\|, \quad \|\mathbf{d}^k\| = \|\mathbf{d}_{T_k\cup J_k}^k\| \\ &\langle \nabla_{T_k^c} f(\mathbf{x}^k),\ \mathbf{x}_{T_k^c}^k\rangle = \langle \nabla_{J_k} f(\mathbf{x}^k),\ \mathbf{x}_{J_k}^k\rangle \end{aligned} \tag{39}$$

By the definition of T_k and the fact, $x_i^k = 0$ for $i \in T_k \setminus T_{k-1}$, we have

$$|\eta\nabla_i f(\mathbf{x}^k)|^2 = |x_i^k - \eta\nabla_i f(\mathbf{x}^k)|^2 \geq |x_j^k - \eta\nabla_j f(\mathbf{x}^k)|^2, \quad \forall\ i \in T_k \setminus T_{k-1}, \quad j \in J_k.$$

The above inequality and the fact $|T_k \setminus T_{k-1}| = |J_k|$ in (35) imply

$$\begin{aligned} \eta^2\|\nabla_{T_k\setminus T_{k-1}} f(\mathbf{x}^k)\|^2 &= \sum_{i\in T_k\setminus T_{k-1}} |\eta\nabla_i f(\mathbf{x}^k)|^2 \geq \sum_{j\in J_k} |x_j^k - \eta\nabla_i f(\mathbf{x}^k)|^2 \\ \geq \quad & \|\mathbf{x}_{J_k}^k - \eta\nabla_{J_k} f(\mathbf{x}^k)\|^2 = \|\mathbf{x}_{J_k}^k\|^2 - 2\eta\langle \mathbf{x}_{J_k}^k,\ \nabla_{J_k} f(\mathbf{x}^k)\rangle + \eta^2\|\nabla_{J_k} f(\mathbf{x}^k)\|^2 \\ \overset{(39)}{=} \quad & \|\mathbf{x}_{T_k^c}^k\|^2 - 2\eta\langle \mathbf{x}_{J_k}^k,\ \nabla_{J_k} f(\mathbf{x}^k)\rangle + \eta^2\|\nabla_{J_k} f(\mathbf{x}^k)\|^2, \end{aligned}$$

which together with

$$\begin{aligned} \|\nabla_{T_k} f(\mathbf{x}^k)\|^2 &= \|\nabla_{T_k\cap T_{k-1}} f(\mathbf{x}^k)\|^2 + \|\nabla_{T_k\setminus T_{k-1}} f(\mathbf{x}^k)\|^2 \\ \|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2 &= \|\nabla_{T_k\cap T_{k-1}} f(\mathbf{x}^k)\|^2 + \|\nabla_{J_k} f(\mathbf{x}^k)\|^2 \end{aligned}$$

results in the following inequality on the gradient $\nabla f(\mathbf{x}^k)$

$$\eta\|\nabla_{T_k} f(\mathbf{x}^k)\|^2 - \eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2 - \|\mathbf{x}_{T_k^c}^k\|^2/\eta \geq -2\langle \mathbf{x}_{J_k}^k,\ \nabla_{J_k} f(\mathbf{x}^k)\rangle. \tag{40}$$

Appendix B. Proof of Lemma 3

Proof The proof for the "only if" part is straightforward. Suppose $\mathbf{x}$ satisfies (10). We have $\mathbf{x} = \mathcal{P}_s(\mathbf{x} - \eta\nabla f(\mathbf{x}))$. By the definition of $\mathcal{P}_s(\cdot)$ and $T \in \mathcal{T}(\mathbf{x}, \eta)$, we have $\mathbf{x}_{T^c} = 0$ and

$$\mathbf{x}_T = \Big(\mathcal{P}_s(\mathbf{x} - \eta\nabla f(\mathbf{x}))\Big)_T = (\mathbf{x} - \eta\nabla f(\mathbf{x}))_T = \mathbf{x}_T - \eta\nabla_T f(\mathbf{x}),$$

which implies $\nabla_T f(\mathbf{x}) = 0$.

We now prove the "if" part. Suppose we have $F_\eta(\mathbf{x}; T) = 0$ for all $T \in \mathcal{T}(\mathbf{x}; \eta)$, namely,

$$\nabla_T f(\mathbf{x}) = 0, \quad \mathbf{x}_{T^c} = 0. \tag{41}$$

We consider two cases. Case I: $\mathcal{T}(\mathbf{x}; \eta)$ is a singleton. By letting T be the only entry of $\mathcal{T}(\mathbf{x}; \eta)$, then

$$\mathbf{x} - \mathcal{P}_s(\mathbf{x} - \eta\nabla f(\mathbf{x})) = \begin{bmatrix} \mathbf{x}_T \\ \mathbf{x}_{T^c} \end{bmatrix} - \begin{bmatrix} \mathbf{x}_T - \eta\nabla_T f(\mathbf{x}) \\ 0 \end{bmatrix} \overset{(41)}{=} \begin{bmatrix} \mathbf{x}_T - \mathbf{x}_T \\ 0 - 0 \end{bmatrix} = 0,$$

which means $\mathbf{x}$ satisfies the fixed point equation (10).

Case II: $\mathcal{T}(\mathbf{x};\eta)$ has multiple elements. Then by the definition (12) of $\mathcal{T}(\mathbf{x};\eta)$ we have two claims:

$$(x-\eta\nabla f(\mathbf{x}))_{(s)} = (x-\eta\nabla f(\mathbf{x}))_{(s+1)} > 0 \quad \text{or} \quad (x-\eta\nabla f(\mathbf{x}))_{(s)} = 0.$$

Now we exclude the first claim. Without loss of any generality, we assume

$$|x_1-\eta\nabla_1 f(\mathbf{x})| \geq \cdots \geq |x_s-\eta\nabla_s f(\mathbf{x})| = |x_{s+1}-\eta\nabla_{s+1} f(\mathbf{x})| = (x-\eta\nabla f(\mathbf{x}))_{(s)}.$$

Let $T_1=\{1,2,\cdots,s\}$ and $T_2=\{1,2,\cdots,s-1,s+1\}$. Then $F_\eta(\mathbf{x};T_1)=F_\eta(\mathbf{x};T_2)=0$ imply that $\nabla_{T_1}f(\mathbf{x})=\nabla_{T_2}f(\mathbf{x})=0$ and $\mathbf{x}_{T_1^c}=\mathbf{x}_{T_2^c}=0$, which lead to

$$\begin{aligned} |x_1| &= |x_1-\eta\nabla_1 f(\mathbf{x})| \geq \cdots \geq |x_s| = |x_s-\eta\nabla_s f(\mathbf{x})| = \\ |x_{s+1}| &= |x_{s+1}-\eta\nabla_{s+1} f(\mathbf{x})| = (x-\eta\nabla f(\mathbf{x}))_{(s)} > 0. \end{aligned}$$

This is contradicted with $\mathbf{x}_{T_1^c}=0$ because of $(s+1)\in T_1^c$. Therefore, we have $(x-\eta\nabla f(\mathbf{x}))_{(s)}=0$. This together with the definition (12) of $\mathcal{T}(\mathbf{x};\eta)$ yields $0=(x-\eta\nabla f(\mathbf{x}))_{(s)} \geq |x_i-\eta\nabla_i f(\mathbf{x})| = |\eta\nabla_i f(\mathbf{x})|$ for any $i\in T^c$, which combining $\nabla_T f(\mathbf{x})=0$ renders $\nabla f(\mathbf{x})=0$. Hence $x_{(s)}=(x-\eta\nabla f(\mathbf{x}))_{(s)}=0$, yielding $\|\mathbf{x}\|_0<s$. Consequently, $\mathbf{x}=\mathbf{x}-\eta\nabla f(\mathbf{x})$ (because $\nabla f(\mathbf{x})=0$ and $\mathbf{x}=\mathcal{P}_s(\mathbf{x})=\mathcal{P}_s(\mathbf{x}-\eta\nabla f(\mathbf{x}))$ (because $\|\mathbf{x}\|_0<s$). That is $\mathbf{x}$ also satisfies the fixed point equation (10). ∎

Appendix C. Proof of Lemma 4

Proof For simplicity, we write $\mathbf{d}^k := \mathbf{d}_N^k$. Since $f(\mathbf{x})$ is m_{2s}-restricted strongly convex and M_{2s}-restricted strongly smooth. For any $\|\mathbf{x}\|_0 \leq s$, it follows from Definition 1 that

$$m_{2s}I_{2s} \preceq \nabla_T^2 f(\mathbf{x}) \preceq M_{2s}I_{2s} \qquad \text{for any } |T|\leq 2s. \tag{42}$$

Clearly, $|T_k\cup J_k|\leq 2s$ due to $|T_k|\leq s$ and $|J_k|\leq s$. This together with (38) implies

$$\begin{aligned} 2\left\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}_{T_k}^k \right\rangle &= -\left\langle \mathbf{d}_{T_k\cup J_k}^k, \nabla_{T_k\cup J_k}^2 f(\mathbf{x}^k)\mathbf{d}_{T_k\cup J_k}^k \right\rangle \\ &\quad -\left\langle H_k\mathbf{d}_{T_k}^k, \mathbf{d}_{T_k}^k \right\rangle + \left\langle \mathbf{d}_{J_k}^k, \nabla_{J_k}^2 f(\mathbf{x}^k)\mathbf{d}_{J_k}^k \right\rangle \\ &\leq -m_{2s}\left[\|\mathbf{d}_{T_k\cup J_k}^k\|^2 + \|\mathbf{d}_{T_k}^k\|^2\right] + M_{2s}\|\mathbf{x}_{T_k^c}^k\|^2 \\ &= -m_{2s}\left[\|\mathbf{d}_{T_k\cup J_k}^k\|^2 + \|\mathbf{d}_{T_k}^k\|^2 + \|\mathbf{d}_{J_k}^k\|^2 - \|\mathbf{d}_{J_k}^k\|^2\right] + M_{2s}\|\mathbf{x}_{T_k^c}^k\|^2 \\ &\overset{(39)}{=} -2m_{2s}\|\mathbf{d}^k\|^2 + m_{2s}\|\mathbf{x}_{T_k^c}^k\|^2 + M_{2s}\|\mathbf{x}_{T_k^c}^k\|^2 \\ &\leq -2m_{2s}\|\mathbf{d}^k\|^2 + 2M_{2s}\|\mathbf{x}_{T_k^c}^k\|^2 \\ &\leq -2\gamma\|\mathbf{d}^k\|^2 + \|\mathbf{x}_{T_k^c}^k\|^2/(2\eta), \end{aligned} \tag{43}$$

where the last inequality is owing to that $\gamma\leq m_{2s}$ and $\eta\leq 1/(4M_{2s})$. ∎

Appendix D. Proof of Lemma 5

Proof It follows from the fact $\eta < \overline{\eta}$ that

$$\eta < \overline{\eta} \leq \min\left\{\frac{\gamma(\overline{\alpha}\beta)}{M_{2s}^2},\ \overline{\alpha}\beta\right\} < \min\left\{\frac{\gamma}{M_{2s}^2},\ 1\right\},$$

where the last strict inequality used $\overline{\alpha} \leq 1$ and $\beta < 1$. Therefore, ρ is well defined and $\rho > 0$. Since $\mathrm{supp}(\mathbf{x}^k) \subseteq T_{k-1}$, the relationships in (35)-(40) all hold. We now prove the claim by two cases.

Case 1: If $\mathbf{d}^k = \mathbf{d}_N^k$, then it follows from (22) that

$$2\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}_{T_k}^k\rangle \ \ \leq \ \ -2\gamma\|\mathbf{d}^k\|^2 + \|\mathbf{x}_{T_k^c}^k\|^2/(2\eta). \tag{44}$$

In addition,

$$\begin{aligned}
\|\nabla_{T_k} f(\mathbf{x}^k)\|^2 &\overset{(37)}{=} \|H_k \mathbf{d}_{T_k}^k - G_k \mathbf{x}_{J_k}^k\|^2 \overset{(37)}{=} \|[H_k,\ G_k]\mathbf{d}_{T_k\cup J_k}^k\|^2 &(45)\\
&\leq M_{2s}^2\|\mathbf{d}_{T_k\cup J_k}^k\|^2 \overset{(39)}{=} M_{2s}^2\|\mathbf{d}^k\|^2, &(46)
\end{aligned}$$

where the inequality holds because $\|[H_k,\ G_k]\|_2 \leq \|\nabla^2_{T_k\cup J_k} f(\mathbf{x}^k)\|_2$ due to f being M_{2s}-restricted strongly smooth and $|T_k \cup J_k| \leq 2s$. This together with (40) derives

$$-2\langle \mathbf{x}_{J_k}^k, \nabla_{J_k} f(\mathbf{x}^k)\rangle \leq \eta M_{2s}^2\|\mathbf{d}^k\|^2 - \|\mathbf{x}_{T_k^c}^k\|^2/\eta - \eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2. \tag{47}$$

Direct calculation yields the following chain of inequalities,

$$\begin{aligned}
2\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle &= 2\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}_{T_k}^k\rangle - 2\langle \nabla_{T_k^c} f(\mathbf{x}^k), \mathbf{x}_{T_k^c}^k\rangle\\
&\overset{(39)}{=} 2\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}_{T_k}^k\rangle - 2\langle \nabla_{J_k} f(\mathbf{x}^k), \mathbf{x}_{J_k}^k\rangle\\
&\overset{(44,47)}{\leq} -\left[2\gamma - \eta M_{2s}^2\right]\|\mathbf{d}^k\|^2 - \|\mathbf{x}_{T_k^c}^k\|^2/(2\eta) - \eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2\\
&\leq -2\rho\|\mathbf{d}^k\|^2 - \eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2
\end{aligned}$$

Case 2: If $\mathbf{d}^k = \mathbf{d}_g^k$, then it follows from (21) (namely, $\mathbf{d}_{T_k}^k = -\nabla_{T_k} f(\mathbf{x}^k)$) that

$$\begin{aligned}
2\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle &= 2\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}_{T_k}^k\rangle - 2\langle \nabla_{T_k^c} f(\mathbf{x}^k), \mathbf{x}_{T_k^c}^k\rangle\\
&\overset{(39,40)}{\leq} -2\|\mathbf{d}_{T_k}^k\|^2 + \eta\|\nabla_{T_k} f(\mathbf{x}^k)\|^2 - \|\mathbf{x}_{T_k^c}^k\|^2/\eta - \eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2\\
&= -(2-\eta)\|\mathbf{d}_{T_k}^k\|^2 - \|\mathbf{d}_{T_k^c}^k\|^2/\eta - \eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2\\
&\leq -(2-\eta)(\|\mathbf{d}_{T_k}^k\|^2 + \|\mathbf{d}_{T_k^c}^k\|^2) - \eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2\\
&\leq -2\rho\|\mathbf{d}^k\|^2 - \eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2,
\end{aligned}$$

where the second inequality used the fact $\eta(2-\eta) \leq 1$. This finishes the proof. ■

Appendix E. Proof of Lemma 6

Proof If $0 < \alpha \leq \overline{\alpha}$ and $0 < \gamma \leq \min\{1, 2M_{2s}\}$, we have

$$\alpha \leq \frac{1-2\sigma}{M_{2s}/\gamma - \sigma} \leq \frac{1-2\sigma}{M_{2s} - \sigma}.$$

Since f is M_{2s}-restricted strongly smooth, we have

$$\begin{array}{rcl} 2f(\mathbf{x}^k(\alpha)) - 2f(\mathbf{x}^k) & \overset{(8)}{\leq} & 2\langle \nabla f(\mathbf{x}^k), \mathbf{x}^k(\alpha) - \mathbf{x}^k \rangle + M_{2s}\|\mathbf{x}^k(\alpha) - \mathbf{x}^k\|^2 \\ & = & 2\langle \nabla f(\mathbf{x}^k), \mathbf{x}^k(\alpha) - \mathbf{x}^k \rangle + M_{2s}\|\mathbf{x}^k(\alpha) - \mathbf{x}^k\|^2 \\ & & -\, 2\alpha\sigma\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle + 2\alpha\sigma\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle \\ & \overset{(26)}{=} & \alpha(1-\sigma)2\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}^k_{T_k} \rangle - (1-\alpha\sigma)2\langle \nabla_{T_k^c} f(\mathbf{x}^k), \mathbf{x}^k_{T_k^c} \rangle \\ & & +\, M_{2s}\left[\alpha^2\|\mathbf{d}^k_{T_k}\|^2 + \|\mathbf{x}^k_{T_k^c}\|^2\right] + 2\alpha\sigma\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle \\ & \overset{(39)}{=} & \Delta + 2\alpha\sigma\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k \rangle, \end{array}$$

where

$$\Delta := \alpha(1-\sigma)2\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}^k_{T_k} \rangle - (1-\alpha\sigma)2\langle \nabla_{J_k} f(\mathbf{x}^k), \mathbf{x}^k_{J_k} \rangle + \ M_{2s}\left[\alpha^2\|\mathbf{d}^k_{T_k}\|^2 + \|\mathbf{x}^k_{T_k^c}\|^2\right]$$

To conclude the conclusion, we only need to show $\Delta \leq 0$. We prove it by two cases.

Case 1: If $\mathbf{d}^k := \mathbf{d}^k_N$, then combining (44) and (47) yields that

$$\begin{array}{rcl} \Delta & \leq & \alpha(1-\sigma)2\langle \nabla_{T_k} f(\mathbf{x}^k), \mathbf{d}^k_{T_k} \rangle - (1-\alpha\sigma)2\langle \nabla_{J_k} f(\mathbf{x}^k), \mathbf{x}^k_{J_k} \rangle + M_{2s}\left[\alpha^2\|\mathbf{d}^k\|^2 + \|\mathbf{x}^k_{T_k^c}\|^2\right] \\ & \leq & c_1\|\mathbf{d}^k\|^2 + c_2\|\mathbf{x}^k_{T_k^c}\|^2 - (1-\alpha\sigma)\eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2, \end{array}$$

where

$$\begin{array}{rcll} c_1 & := & -\alpha(1-\sigma)2\gamma + (1-\alpha\sigma)\eta M_{2s}^2 + M_{2s}\alpha^2, & \\ & \leq & -\alpha(1-\sigma)2\gamma + (1-\alpha\sigma)\gamma\alpha + M_{2s}\alpha^2 & \text{because of } \alpha \leq 1, \sigma \leq \dfrac{1}{2}, \eta \leq \dfrac{\alpha\gamma}{M_{2s}^2} \\ & = & \alpha\left[(M_{2s} - \sigma\delta)\alpha - (1-2\sigma)\gamma\right] \leq 0, & \text{because of } \sigma\gamma \leq M_{2s}, \alpha \leq \dfrac{1-2\sigma}{M_{2s}/\gamma - \sigma} \\ c_2 & := & \alpha(1-\sigma)/(2\eta) - (1-\alpha\sigma)/\eta + M_{2s} & \\ & \leq & (1-\alpha\sigma)/(2\eta) - (1-\alpha\sigma)/\eta + M_{2s} & \text{because of } \alpha \leq 1 \\ & \leq & -(1-\alpha\sigma)/(2\eta) + M_{2s} \leq 0, & \text{because of } \alpha \leq 1, \sigma \leq \dfrac{1}{2}, \eta \leq \dfrac{1}{4M_{2s}}. \end{array}$$

Case 2: If $\mathbf{d}^k := \mathbf{d}^k_g$, then combining (21) that $\mathbf{d}^k_{T_k} = -\nabla_{T_k} f(\mathbf{x}^k)$ and (40) suffices to

$$\Delta \ \leq \ c_3\|\mathbf{d}^k_{T_k}\|^2 + c_4\|\mathbf{x}^k_{T_k^c}\|^2 - (1-\alpha\sigma)\eta\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2,$$

where

$$
\begin{aligned}
c_3 &:= -2\alpha(1-\sigma) + (1-\alpha\sigma)\eta + M_{2s}\alpha^2 \\
&\le \alpha\left[(M_{2s}-\sigma)\alpha - (1-2\sigma)\right] && \text{because of } \alpha \le 1, \sigma \le \frac{1}{2}, \eta \le \alpha \\
&\le 0. && \text{because of } \alpha \le \frac{1-2\sigma}{M_{2s}-\sigma} \\
c_4 &:= -(1-\alpha\sigma)/\eta + M_{2s} \\
&\le -1/(2\eta) + M_{2s} \le 0, && \text{because of } \alpha \le 1, \sigma \le \frac{1}{2}, \eta \le \frac{1}{4M_{2s}}
\end{aligned}
$$

which finishes proving the first claim. If $\eta \in (0, \overline{\eta})$ where $\overline{\eta}$ is defined as (24), then for any $\beta\overline{\alpha} \le \alpha \le \overline{\alpha}$, we have

$$
0 < \eta < \min\left\{\frac{\overline{\alpha}\gamma\beta}{M_{2s}^2}\ \ \overline{\alpha}\beta,\ \frac{1}{4M_{2s}}\right\} \le \min\left\{\frac{\alpha\gamma}{M_{2s}^2},\ \alpha,\ \frac{1}{4M_{2s}}\right\}.
$$

This together with (28), namely, $f(\mathbf{x}^k(\alpha)) - f(\mathbf{x}^k) \le \sigma\alpha\langle\nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle$, and the Armijo-type step size rule means that $\{\alpha_k\}$ is bounded from below by a positive constant, that is,

$$
\inf_{k\ge 0}\{\alpha_k\} \ge \beta\overline{\alpha} > 0. \tag{48}
$$

which finishes the whole proof. ■

Appendix F. Proof of Lemma 7

Proof Lemma 6 shows the existence of α_k, then (27) in NHTP (namely, (28)) provides

$$
\begin{aligned}
f(\mathbf{x}^{k+1}) - f(\mathbf{x}^k) \le \sigma\alpha_k\langle\nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle &\overset{(25)}{\le} -\sigma\alpha_k\left[\rho\|\mathbf{d}^k\|^2 - (\eta/2)\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2\right] \\
&\overset{(48)}{\le} -\sigma\overline{\alpha}\beta\left[\rho\|\mathbf{d}^k\|^2 - (\eta/2)\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2\right].
\end{aligned}
$$

Thus $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$ if $\mathbf{x}^{k+1} \ne \mathbf{x}^k$. Then it follows from above inequality that

$$
\begin{aligned}
&\sigma\overline{\alpha}\beta\max\left\{\rho\sum_{k=0}^{\infty}\|\mathbf{d}^k\|^2,\ \ \frac{\eta}{2}\sum_{k=0}^{\infty}\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\|^2\right\} \\
\le\ &\sum_{k=0}^{\infty}\left[f(\mathbf{x}^k) - f(\mathbf{x}^{k+1})\right] < \left[f(\mathbf{x}^0) - \lim_{k\to+\infty} f(\mathbf{x}^k)\right] < +\infty,
\end{aligned}
$$

where the last inequality is due to f being bounded from below. Hence

$$
\lim_{k\to\infty}\|\mathbf{d}^k\| = \lim_{k\to\infty}\|\nabla_{T_{k-1}} f(\mathbf{x}^k)\| = 0
$$

which suffices to $\lim_{k\to\infty}\|\mathbf{x}^{k+1} - \mathbf{x}^k\| = 0$ because of

$$
\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \overset{(26)}{=} \alpha_k^2\|\mathbf{d}^k_{T_k}\|^2 + \|\mathbf{x}^k_{T_k^c}\|^2 \le \|\mathbf{d}^k_{T_k}\|^2 + \|\mathbf{d}^k_{T_k^c}\|^2 = \|\mathbf{d}^k\|^2
$$

If $\mathbf{d}^k = \mathbf{d}_N^k$, then it follows from (46) that $\|\nabla_{T_k} f(\mathbf{x}^k)\| \le M_{2s}\|\mathbf{d}^k\|$. If $\mathbf{d}^k = \mathbf{d}_g^k$, it follows from (21) that $\|\nabla_{T_k} f(\mathbf{x}^k)\| = \|\mathbf{d}_{T_k}^k\|$. Those suffice to $\lim_{k\to\infty}\|\nabla_{T_k} f(\mathbf{x}^k)\| = 0$. Finally, (13) allows us to derive

$$\|F_\eta(\mathbf{x}^k; T_k)\|^2 = \|\nabla_{T_k} f(\mathbf{x}^k)\|^2 + \|\mathbf{x}_{T_k^c}^k\|^2 \le (M_{2s}^2 + 1)\|\mathbf{d}^k\|^2,$$

which is also able to claim $\lim_{k\to\infty}\|F_\eta(\mathbf{x}^k; T_k)\| = 0$. ■

Appendix G. Proof of Theorem 8

Proof (i) We prove in Lemma 7 (iv) that

$$\lim_{k\to\infty} \nabla_{T_k} f(\mathbf{x}^{k+1}) = 0. \tag{49}$$

Let $\{\mathbf{x}^{k_\ell}\}$ be the convergent subsequence of $\{\mathbf{x}^k\}$ that converges to $\mathbf{x}^*$. Since there are only finitely many choices for T_k, (re-subsequencing if necessary) we may without loss of any generality assume that the sequence of the index sets $\{\{T_{k_\ell - 1}\}\}$ shares a same index set, denoted as T_∞. That is

$$T_{k_\ell - 1} = T_{k_{\ell+1} - 1} = \cdots = T_\infty. \tag{50}$$

Since $\mathbf{x}^{k_\ell} \to \mathbf{x}^*$, $\mathrm{supp}(\mathbf{x}^{k_\ell}) \subseteq \mathrm{T}_{k_\ell - 1} = \mathrm{T}_\infty$, we must have

$$T_\infty \begin{cases} = \Gamma_* := \mathrm{supp}(\mathbf{x}^*), & \text{if } \|\mathbf{x}^*\|_0 = s, \\ \supset \Gamma_*, & \text{if } \|\mathbf{x}^*\|_0 < s. \end{cases}$$

which implies

$$\nabla_{T_\infty} f(\mathbf{x}^*) = \lim_{k_\ell \to \infty} \nabla_{T_{k_\ell - 1}} f(\mathbf{x}^{k_\ell}) \overset{(49)}{=} 0. \tag{51}$$

In addition, the definition (12) of $\mathcal{T}(\mathbf{x}^k, \eta)$ means

$$|\mathbf{x}_i^k - \eta \nabla_i f(\mathbf{x}^k)| \ge |\mathbf{x}_j^k - \eta \nabla_j f(\mathbf{x}^k)|, \quad \forall\ i \in T_k,\ \forall\ j \in T_k^c \tag{52}$$

Again by Lemma 7 (ii) that $\lim_{k\to\infty} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| = 0$, we obtain $\lim_{k_\ell\to\infty} \mathbf{x}^{k_\ell - 1} = \mathbf{x}^*$ due to $\lim_{k_\ell\to\infty} \mathbf{x}^{k_\ell} = \mathbf{x}^*$. Now we have the following chain of inequalities for any $i \in T_{k_\ell - 1} \overset{(50)}{=} T_\infty,\ j \in T_{k_\ell - 1}^c \overset{(50)}{=} T_\infty^c$

$$\begin{aligned} |\mathbf{x}_i^*| &\overset{(51)}{=} |\mathbf{x}_i^* - \eta \nabla_i f(\mathbf{x}^*)| = \lim_{k_\ell\to\infty} |\mathbf{x}_i^{k_\ell - 1} - \eta \nabla_i f(\mathbf{x}^{k_\ell - 1})| \\ &\overset{(52)}{\ge} \lim_{k_\ell\to\infty} |\mathbf{x}_j^{k_\ell - 1} - \eta \nabla_j f(\mathbf{x}^{k_\ell - 1})| = |\mathbf{x}_j^* - \eta \nabla_j f(\mathbf{x}^*)| = \eta |\nabla_j f(\mathbf{x}^*)|, \end{aligned}$$

which leads to

$$\mathbf{x}_{(s)}^* = \min_{i \in T_\infty} |\mathbf{x}_i^*| \ge \eta |\nabla_j f(\mathbf{x}^*)|, \quad \forall\ \ j \in T_\infty^c, \tag{53}$$

If $\|\mathbf{x}^*\|_0 = s$, then $T_\infty = \Gamma_*$. Consequently, $x^*_{(s)} \geq \eta|\nabla_j f(\mathbf{x}^*)|, \forall\, j \in \Gamma^c_*$. If $\|\mathbf{x}^*\|_0 < s$, then $x^*_{(s)} = 0$ and $\nabla f(\mathbf{x}^*) = 0$ from (53) and (51). Those together with (9) enable us to show that $\mathbf{x}^*$ is an η-stationary point.

If $f(\mathbf{x})$ is convex, letting $\Gamma_* := \mathrm{supp}(\mathbf{x}^*)$, then

$$\begin{aligned}
f(\mathbf{x}) &\geq f(\mathbf{x}^*) + \langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \\
&> f(\mathbf{x}^*) + \sum_{i\in\Gamma_*} \nabla_i f(\mathbf{x}^*)(x_i - x_i^*) + \sum_{i\notin\Gamma_*} \nabla_i f(\mathbf{x}^*)(x_i - x_i^*) \\
&\overset{(9)}{=} f(\mathbf{x}^*) + \sum_{i\notin\Gamma_*} \nabla_i f(\mathbf{x}^*) x_i \geq f(\mathbf{x}^*) - \sum_{i\notin\Gamma_*} |\nabla_i f(\mathbf{x}^*)||x_i| \\
&\overset{(9)}{\geq} f(\mathbf{x}^*) - (\mathbf{x}^*_{(s)}/\eta) \sum_{i\notin\Gamma_*} |x_i| \\
&= f(\mathbf{x}^*) - (\mathbf{x}^*_{(s)}/\eta)\|x_{\Gamma^c_*}\|_1.
\end{aligned}$$

(ii) The whole sequence converges because of (Lemma 4.10, Moré and Sorensen, 1983) and $\lim_{k\to\infty} \|\mathbf{x}^{k+1} - \mathbf{x}^k\| = 0$ from Lemma 7 (ii). If $\mathbf{x}^* = 0$, then the conclusion holds clearly due to $\mathrm{supp}(\mathbf{x}^*) = \emptyset$. We consider $\mathbf{x}^* \neq 0$. Since $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{x}^*$, the for sufficiently large k we must have

$$\|\mathbf{x}^k - \mathbf{x}^*\| < \min_{i\in\mathrm{supp}(\mathbf{x}^*)} |x_i^*| =: t^*.$$

If $\mathrm{supp}(\mathbf{x}^*) \nsubseteq \mathrm{supp}(\mathbf{x}^{\mathrm{k}})$, then there is an $i_0 \in \mathrm{supp}(\mathbf{x}^*) \setminus \mathrm{supp}(\mathbf{x}^{\mathrm{k}})$ such that

$$t^* > \|\mathbf{x}^k - \mathbf{x}^*\| \geq |x^k_{i_0} - x^*_{i_0}| = |x^*_{i_0}| \geq t^*,$$

which is a contradiction. Therefore, $\mathrm{supp}(\mathbf{x}^*) \subseteq \mathrm{supp}(\mathbf{x}^{\mathrm{k}})$. By the updating rule (26), we have $\mathrm{supp}(\mathbf{x}^{\mathrm{k}}) \subseteq \mathrm{T}_{\mathrm{k}-1}$, where $|T_{k-1}| = s$ by (12). Therefore, if $\|\mathbf{x}^*\|_0 = s$ then $\mathrm{supp}(\mathbf{x}^*) \equiv \mathrm{supp}(\mathbf{x}^{\mathrm{k}}) \equiv \mathrm{supp}(\mathbf{x}^{\mathrm{k+1}}) \equiv \mathrm{T}_{\mathrm{k}}$. If $\|\mathbf{x}^*\|_0 < s$ then $\mathrm{supp}(\mathbf{x}^*) \subseteq \mathrm{supp}(\mathbf{x}^{\mathrm{k}}), \mathrm{supp}(\mathbf{x}^*) \subseteq \mathrm{supp}(\mathbf{x}^{\mathrm{k+1}}) \subseteq \mathrm{T}_{\mathrm{k}}$. The whole proof is finished. ■

Appendix H. Proof of Theorem 9

Proof (i) We have proved in Theorem 8 (i) that any limit $\mathbf{x}^*$ of $\{\mathbf{x}^k\}$ is an η-stationary point. If $f(\mathbf{x})$ is m_{2s}-restricted strongly convex in a neighborhood of $\mathbf{x}^*$, then we can conclude that $\mathbf{x}^*$ is a strictly local minimizer of (1). In fact,

$$\begin{array}{rcl} f(\mathbf{x}) & \geq & f(\mathbf{x}^*) + \langle \nabla f(\mathbf{x}^*), \mathbf{x}-\mathbf{x}^*\rangle + (m_{2s}/2)\|\mathbf{x}-\mathbf{x}^*\|^2 \\ & > & f(\mathbf{x}^*) + \sum\limits_{i\in \mathrm{supp}(\mathbf{x}^*)} \nabla_i f(\mathbf{x}^*)(x_i - x_i^*) + \sum\limits_{i\notin \mathrm{supp}(\mathbf{x}^*)} \nabla_i f(\mathbf{x}^*)(x_i - x_i^*) \\ & \overset{(9)}{=} & f(\mathbf{x}^*) + \sum\limits_{i\notin \mathrm{supp}(\mathbf{x}^*)} \nabla_i f(\mathbf{x}^*) x_i \\ & \overset{(9)}{=} & f(\mathbf{x}^*) + \left\{ \begin{array}{ll} \sum_{i\notin \mathrm{supp}(\mathbf{x}^*)} \nabla_i f(\mathbf{x}^*) \times 0, & \|\mathbf{x}^*\|_0 = s \\ \sum_{i\notin \mathrm{supp}(\mathbf{x}^*)} 0 \times x_i, & \|\mathbf{x}^*\|_0 < s. \end{array} \right. \\ & = & f(\mathbf{x}^*) \end{array}$$

for any s-sparse vector $\mathbf{x}$, where the first inequality is from the m_{2s}-restricted strongly convexity. This also shows $\mathbf{x}^*$ is isolated and thus the whole sequence tends to $\mathbf{x}^*$ by Theorem 8 (ii).

(ii) The fact that $f(\mathbf{x})$ is m_{2s}-restricted strongly convex in a neighborhood of $\mathbf{x}^*$ and $\lim_{k\to\infty}\mathbf{x}^k = \mathbf{x}^*$ implies that $f(\mathbf{x})$ is also m_{2s}-restricted strongly convex in a neighborhood of $\mathbf{x}^k$ for sufficiently large k. By invoking Lemma 4, we see that the Newton direction $\mathbf{d}_N^k$ always satisfies the condition (20) and hence is accepted as the search direction when k is sufficiently large.

(iii) By $\mathrm{supp}(\mathbf{x}^*) \subseteq \mathrm{T_k}$ for sufficiently large k from 8 (ii) and $\mathbf{x}^*$ is an η-stationary point, it follows from (9) that

$$\mathbf{x}^*_{T_k^c} = 0 \quad \text{and} \quad \left\{ \begin{array}{rll} \nabla_{T_k} f(\mathbf{x}^*) = \nabla_{\mathrm{supp}(\mathbf{x}^*)} f(\mathbf{x}^*) & = 0 & \text{if } \|\mathbf{x}^*\|_0 = s, \\ \nabla f(\mathbf{x}^*) & = 0 & \text{if } \|\mathbf{x}^*\|_0 < s. \end{array} \right. \tag{54}$$

For any $0 \leq t \leq 1$, denote $\mathbf{x}(t) := \mathbf{x}^* + t(\mathbf{x}^k - \mathbf{x}^*)$. Clearly, as $\mathbf{x}^k$, $\mathbf{x}(t)$ is also in the neighbour of $\mathbf{x}^*$. So f being locally restricted Hessian Lipschitz continuous at $\mathbf{x}^*$ with the Lipschitz constant L_f gives rise to

$$\|\nabla^2_{T_k:} f(\mathbf{x}^k) - \nabla^2_{T_k:} f(\mathbf{x}(t))\| \leq L_f \|\mathbf{x}^k - \mathbf{x}(t)\| = (1-t)L_f\|\mathbf{x}^k - \mathbf{x}^*\|. \tag{55}$$

Moreover, by the Taylor expansion, we have

$$\nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^*) = \int_0^1 \nabla^2 f(\mathbf{x}(t))(\mathbf{x}^k - \mathbf{x}^*)dt. \tag{56}$$

We also have the following chain of inequalities

$$\begin{array}{rcll} \|\mathbf{x}^{k+1} - \mathbf{x}^*\| & = & \left[\|\mathbf{x}^{k+1}_{T_k} - \mathbf{x}^*_{T_k}\|^2 + \|\mathbf{x}^{k+1}_{T_k^c} - \mathbf{x}^*_{T_k^c}\|^2\right]^{1/2} & \\ & = & \|\mathbf{x}^{k+1}_{T_k} - \mathbf{x}^*_{T_k}\| \overset{(26)}{=} \|\mathbf{x}^k_{T_k} - \mathbf{x}^*_{T_k} + \alpha_k \mathbf{d}^k_{T_k}\| & \\ & \leq & (1-\alpha_k)\|\mathbf{x}^k_{T_k} - \mathbf{x}^*_{T_k}\| + \alpha_k \|\mathbf{x}^k_{T_k} - \mathbf{x}^*_{T_k} + \mathbf{d}^k_{T_k}\| & (57) \\ & \overset{(48)}{\leq} & (1-\overline{\alpha}\beta)\|\mathbf{x}^k - \mathbf{x}^*\| + \overline{\alpha}\|\mathbf{x}^k_{T_k} - \mathbf{x}^*_{T_k} + \mathbf{d}^k_{T_k}\|, & (58) \end{array}$$

where the second equality used the fact (26) and $\text{supp}(\mathbf{x}^{k+1}) \subseteq T_k$. Since $\mathbf{d}^k = \mathbf{d}_N^k$, we have

$$\begin{aligned}
& \|\mathbf{x}_{T_k}^k - \mathbf{x}_{T_k}^* + \mathbf{d}_{T_k}^k\| \qquad (59)\\
\overset{(18)}{=} \quad & \left\| H_k^{-1}\left(\nabla^2_{T_k,T_k^c} f(\mathbf{x}^k)\mathbf{x}^k_{T_k^c} - \nabla_{T_k} f(\mathbf{x}^k)\right) + \mathbf{x}_{T_k}^k - \mathbf{x}_{T_k}^*\right\| \\
= \quad & \left\| H_k^{-1}\left(\nabla^2_{T_k,T_k^c} f(\mathbf{x}^k)\mathbf{x}^k_{T_k^c} - \nabla_{T_k} f(\mathbf{x}^k) + \nabla^2_{T_k} f(\mathbf{x}^k)\mathbf{x}^k_{T_k} - \nabla^2_{T_k} f(\mathbf{x}^k)\mathbf{x}^*_{T_k}\right)\right\| \\
\leq \quad & \frac{1}{m_{2s}} \left\| \nabla^2_{T_k:} f(\mathbf{x}^k)\mathbf{x}^k - \nabla_{T_k} f(\mathbf{x}^k) - \nabla^2_{T_k} f(\mathbf{x}^k)\mathbf{x}^*_{T_k}\right\| \\
\overset{(54)}{=} \quad & \frac{1}{m_{2s}} \left\| \nabla^2_{T_k:} f(\mathbf{x}^k)\mathbf{x}^k - \nabla_{T_k} f(\mathbf{x}^k) - \nabla^2_{T_k:} f(\mathbf{x}^k)\mathbf{x}^* + \nabla_{T_k} f(\mathbf{x}^*)\right\| \\
\overset{(56)}{=} \quad & \frac{1}{m_{2s}} \left\| \nabla^2_{T_k:} f(\mathbf{x}^k)(\mathbf{x}^k - \mathbf{x}^*) - \int_0^1 \nabla^2_{T_k:} f(\mathbf{x}(t))(\mathbf{x}^k - \mathbf{x}^*)dt\right\| \\
= \quad & \frac{1}{m_{2s}} \left\| \int_0^1 \left[\nabla^2_{T_k:} f(\mathbf{x}^k) - \nabla^2_{T_k:} f(\mathbf{x}(t))\right](\mathbf{x}^k - \mathbf{x}^*)dt\right\| \\
\leq \quad & \frac{1}{m_{2s}} \int_0^1 \left\| \nabla^2_{T_k:} f(\mathbf{x}^k) - \nabla^2_{T_k:} f(\mathbf{x}(t))\right\| \|\mathbf{x}^k - \mathbf{x}^*\| dt \\
\overset{(55)}{\leq} \quad & \frac{L_f}{m_{2s}} \|\mathbf{x}^k - \mathbf{x}^*\|^2 \int_0^1 (1-t)dt \\
\leq \quad & L_f/(2m_{2s})\|\mathbf{x}^k - \mathbf{x}^*\|^2. \qquad (60)
\end{aligned}$$

Now, we have obtained (fact 1) $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{x}^*$, (fact 2) $\langle \nabla f(\mathbf{x}^k), \mathbf{d}^k\rangle \leq -\rho\|\mathbf{d}^k\|^2$ from Lemma 5 and (fact 3)

$$\lim_{k\to\infty} \frac{\|\mathbf{x}^k + \mathbf{d}^k - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} = \lim_{k\to\infty} \frac{\|\mathbf{x}^k_{T_k} + \mathbf{d}^k_{T_k} - \mathbf{x}^*_{T_k}\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} \overset{(60)}{\leq} \lim_{k\to\infty} \frac{L_f\|\mathbf{x}^k - \mathbf{x}^*\|^2}{2m_{2s}\|\mathbf{x}^k - \mathbf{x}^*\|} = 0,$$

where the first equality is because of $\mathbf{d}^k_{T_k^c} = -\mathbf{x}^k_{T_k^c}$ and (54). These three facts are exactly the same assumptions used in (Theorem 3.3, Facchinei, 1995), which establishes that eventually the step size α_k in the Armijo rule has to be 1, namely $\alpha_k \equiv 1$. Therefore, for sufficiently large k, it follows from (57) that

$$\begin{aligned}
\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \quad & \leq \quad \alpha_k\|\mathbf{x}_{T_k}^k - \mathbf{x}_{T_k}^* + \mathbf{d}_{T_k}^k\| + (1-\alpha_k)\|\mathbf{x}_{T_k}^k - \mathbf{x}_{T_k}^*\| \\
& = \quad \|\mathbf{x}_{T_k}^k - \mathbf{x}_{T_k}^* + \mathbf{d}_{T_k}^k\| \\
& \overset{(60)}{\leq} \quad (L_f/2m_{2s})\|\mathbf{x}^k - \mathbf{x}^*\|^2. \qquad (61)
\end{aligned}$$

That is, we have proved that the sequence has a quadratic convergence rate. Finally, for sufficiently large k, it follows

$$\begin{aligned}
\|F_\eta(\mathbf{x}^{k+1}; T_{k+1})\|^2 \quad & \overset{(13)}{=} \quad \|\nabla_{T_{k+1}} f(\mathbf{x}^{k+1})\|^2 + \|\mathbf{x}^{k+1}_{T^c_{k+1}}\|^2 \\
& \overset{(54)}{=} \quad \|\nabla_{T_{k+1}} f(\mathbf{x}^{k+1}) - \nabla_{T_{k+1}} f(\mathbf{x}^*)\|^2 + \|\mathbf{x}^k_{T^c_{k+1}} - \mathbf{x}^*_{T^c_{k+1}}\|^2 \\
& \overset{(8)}{\leq} \quad (M_{2s}^2 + 1)\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \\
& \overset{(61)}{\leq} \quad (M_{2s}^2 + 1)(L_f/2m_{2s})^2\|\mathbf{x}^k - \mathbf{x}^*\|^4. \qquad (62)
\end{aligned}$$

Since $f(\mathbf{x})$ is m_{2s}-restricted strongly convex in a neighborhood of $\mathbf{x}^*$, then $f(\mathbf{x})$ is also m_{2s}-restricted strongly convex in a neighborhood of $\mathbf{x}^k$ due to $\mathbf{x}^k$ being in neighborhood of $\mathbf{x}^*$ for sufficiently large k. Then

$$\begin{aligned} f(\mathbf{x}^k) &\geq f(\mathbf{x}^*) + \langle \nabla f(\mathbf{x}^*), \mathbf{x}^k - \mathbf{x}^* \rangle + (m_{2s}/2)\|\mathbf{x} - \mathbf{x}^*\|^2 \\ f(\mathbf{x}^*) &\geq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^k \rangle + (m_{2s}/2)\|\mathbf{x}^k - \mathbf{x}^*\|^2. \end{aligned}$$

Adding them yields that $\langle \nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^*), \mathbf{x}^k - \mathbf{x}^* \rangle \geq m_{2s}\|\mathbf{x}^k - \mathbf{x}^*\|^2$, which implies

$$\begin{aligned} m_{2s}\|\mathbf{x}^k - \mathbf{x}^*\|^2 &\leq \langle \nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^*), \mathbf{x}^k - \mathbf{x}^* \rangle \\ &= \langle \nabla_{T_k} f(\mathbf{x}^k) - \nabla_{T_k} f(\mathbf{x}^*), \mathbf{x}^k_{T_k} - \mathbf{x}^*_{T_k} \rangle \\ &\leq \|\nabla_{T_k} f(\mathbf{x}^k) - \nabla_{T_k} f(\mathbf{x}^*)\| \|\mathbf{x}^k_{T_k} - \mathbf{x}^*_{T_k}\| \\ &= \|\nabla_{T_k} f(\mathbf{x}^k) - \nabla_{T_k} f(\mathbf{x}^*)\| \|\mathbf{x}^k - \mathbf{x}^*\|, \end{aligned}$$

where two equalities hold due to $\mathbf{x}^k_{T_k^c} = \mathbf{x}^*_{T_k^c} \overset{(54)}{=} 0$. This proves

$$m_{2s}\|\mathbf{x}^k - \mathbf{x}^*\| \leq \|\nabla_{T_k} f(\mathbf{x}^k) - \nabla_{T_k} f(\mathbf{x}^*)\|,$$

which allows us to prove that

$$\begin{aligned} \|F_\eta(\mathbf{x}^k; T_k)\|^2 &\overset{(13)}{=} \|\nabla_{T_k} f(\mathbf{x}^k)\|^2 + \|\mathbf{x}^k_{T_k^c}\|^2 \quad \geq \|\nabla_{T_k} f(\mathbf{x}^k)\|^2 \\ &\overset{(54)}{=} \|\nabla_{T_k} f(\mathbf{x}^k) - \nabla_{T_k} f(\mathbf{x}^*)\|^2 \geq m_{2s}^2\|\mathbf{x}^k - \mathbf{x}^*\|^2 \\ &\overset{(62)}{\geq} \frac{2m_{2s}^3}{L_f\sqrt{M_{2s}^2 + 1}}\|F_\eta(\mathbf{x}^{k+1}; T_{k+1})\|. \end{aligned}$$

This completes the whole proof. ■

References

Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, pages 37–45, 2010.

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

Sohail Bahmani, Petros T. Boufounos, and Bhiksha Raj. Learning model-based sparsity via projected gradient descent. *IEEE Transactions on Information Theory*, 62(4):2092–2099, 2016.

Amir Beck and Yonina C Eldar. Sparsity constrained nonlinear optimization: Optimality conditions and algorithms. *SIAM Journal on Optimization*, 23(3):1480–1509, 2013.

Amir Beck and Nadav Hallak. On the minimization over sparse symmetric sets: projections, optimality conditions, and algorithms. *Mathematics of Operations Research*, 41(1):196–223, 2015.

Thomas Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, 92:752–756, 2012.

Thomas Blumensath. Compressed sensing with nonlinear observations and related nonlinear optimization problems. *IEEE Transactions on Information Theory*, 59(6):3466–3474, 2013.

Thomas Blumensath and Mike E Davies. Gradient pursuits. *IEEE Transactions on Signal Processing*, 56(6):2370–2382, 2008.

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

Thomas Blumensath and Mike E Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of selected topics in signal processing*, 4 (2):298–309, 2010.

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

Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

Jinghui Chen and Quanquan Gu. Fast newton hard thresholding pursuit for sparsity constrained nonconvex optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 757–766. ACM, 2017.

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

Tecla De Luca, Francisco Facchinei, and Christian Kanzow. A semismooth equation approach to the solution of nonlinear complementarity problems. *Mathematical programming*, 75(3):407–439, 1996.

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

Michael Elad. *Sparse and Redundant Representations*. Springer, 2010.

Francisco Facchinei. Minimization of sc1 functions and the maratos effect. *Operations Research Letters*, 17(3):131–138, 1995.

Francisco Facchinei and Christian Kanzow. A nonsmooth inexact newton method for the solution of large-scale nonlinear complementarity problems. *Mathematical Programming*, 76(3):493–512, 1997.

Mário A.T. Figueiredo, Robert D. Nowak, and Stephen J. Wright. Graident projection for psarse reconstruction: application to compressed sensing and other inverse problems. *IEEE J. Selected Topics in Signal Processing*, 1:586–597, 2007.

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

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

James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, NJ, 1994.

Ali Jalali, Christopher C Johnson, and Pradeep K Ravikumar. On learning discrete graphical models using greedy methods. In *Advances in Neural Information Processing Systems*, pages 1935–1943, 2011.

Anastasios Kyrillidis and Volkan Cevher. Recipes on hard thresholing methods. In *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 353–356. IEEE, 2011.

Zhaosong Lu and Yong Zhang. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization*, 23(4):2448–2478, 2013.

Jorge J Moré and Danny C Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

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.

Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999.

Lili Pan, Shenglong Zhou, Naihua Xiu, and Hou-Duo Qi. A convergent iterative hard thresholding for nonnegative sparsity optimization. *Pacific Journal of Optimization*, 13 (2):325–353, 2017.

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

Houduo Qi and Defeng Sun. A quadratically convergent newton method for computing the nearest correlation matrix. *SIAM journal on matrix analysis and applications*, 28(2): 360–385, 2006.

Houduo Qi, Liqun Qi, and Defeng Sun. Solving karush–kuhn–tucker systems via the trust region and the conjugate gradient methods. *SIAM journal on optimization*, 14(2):439–463, 2003.

Shai Shalev-Shwartz, Nathan Seebro, and Tong Zhang. Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM J. Optim.*, 20:2807–2832, 2010.

Jie Shen and Ping Li. A tight bound of hrad thresholding. *Journal of Machine Learning Research*, 18:1–42, 2018.

Defeng Sun, Robert S Womersley, and Houduo Qi. A feasible semismooth asymptotically newton method for mixed complementarity problems. *Mathematical Programming*, 94(1): 167–187, 2002.

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

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

Penghang Yin, Yifei Lou, Qi He, and Jack Xin. Minimization of 1-2 for compressed sensing. *SIAM Journal on Scientific Computing*, 37(1):A536–A563, 2015.

Xiao-Tong Yuan and Qingshan Liu. Newton greedy pursuit: A quadratic approximation method for sparsity-constrained optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4122–4129, 2014.

Xiao-Tong Yuan and Qingshan Liu. Newton-type greedy selection methods for ℓ_0-constrained minimization. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2437–2450, 2017.

XiaoTong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit. *Journal of Machine Learning Research*, 18:1–43, 2018.

Xin-Yuan Zhao, Defeng Sun, and Kim-Chuan Toh. A newton-cg augmented lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4):1737–1765, 2010.

Yunbinl Zhao. *Sparse Optimization: Theorey and Methods.* CRC Press/Taylor & Francis Group, 2018.

Shenglong Zhou, Naihua Xiu, Yingnan Wang, Lingchen Kong, and Hou-Duo Qi. A null-space-based weighted l 1 minimization approach to compressed sensing. *Information and Inference: A Journal of the IMA*, 5(1):76–102, 2016.