HARD THRESHOLDING PURSUIT: AN ALGORITHM FOR COMPRESSIVE SENSING*

SIMON FOUCART†

Abstract. We introduce a new iterative algorithm to find sparse solutions of underdetermined linear systems. The algorithm, a simple combination of the Iterative Hard Thresholding algorithm and of the Compressive Sampling Matching Pursuit or Subspace Pursuit algorithms, is called Hard Thresholding Pursuit. We study its general convergence, and notice in particular that only a finite number of iterations are required. We then show that, under a certain condition on the restricted isometry constant of the matrix of the system, the Hard Thresholding Pursuit algorithm indeed finds all s-sparse solutions. This condition, which reads $\delta_{3s} < 1/\sqrt{3}$, is heuristically better than the sufficient conditions currently available for other Compressive Sensing algorithms. It applies to fast versions of the algorithm, too, including the Iterative Hard Thresholding algorithm. Stability with respect to sparsity defect and robustness with respect to measurement error are also guaranteed under the condition $\delta_{3s} < 1/\sqrt{3}$. We conclude with some numerical experiments to demonstrate the good empirical performance and the low complexity of the Hard Thresholding Pursuit algorithm.

Key words. compressive sensing, sparse recovery, iterative algorithms, thresholding

AMS subject classifications. 65F10, 65J20, 15A29, 94A12

1. Introduction.

In many engineering problems, high-dimensional signals, modeled by vectors in $\mathbb{C}^N$, are observed via low-dimensional measurements, modeled by vectors $\mathbf{y} \in \mathbb{C}^m$ with $m \ll N$, and one wishes to reconstruct the signals from the measurements. Without any prior assumption on the signals, this cannot be hoped for, but the maturing field of Compressive Sensing has shown the feasibility of such a program when the signals are sparse. The objective is then to find suitable measurement matrices $A \in \mathbb{C}^{m \times N}$ and efficient reconstruction algorithms in order to solve underdetermined systems of linear equations $A\mathbf{x} = \mathbf{y}$ when the solutions have only few nonzero components. To date, measurement matrices allowing the reconstruction of s-sparse vectors — vectors with at most s nonzero components — with the minimal number $m \approx c\, s \ln(N/s)$ of measurements have only been constructed probabilistically. As for the reconstruction procedure, a very popular strategy consists in solving the following ℓ_1-minimization problem, known as Basis Pursuit,

$$\underset{\mathbf{z} \in \mathbb{C}^N}{\text{minimize}}\ \|\mathbf{z}\|_1 \qquad \text{subject to} \quad A\mathbf{z} = \mathbf{y}. \tag{BP}$$

The goal of this paper is to propose an alternative strategy that combines two existing iterative algorithms. The new algorithm, called Hard Thresholding Pursuit, together with some variants, is formally introduced in Section 2 after a few intuitive justifications. In Section 3, we analyze the theoretical performance of the algorithm. In particular, we show that the Hard Thresholding Pursuit algorithm allows stable and robust reconstruction of sparse vectors if the measurement matrix satisfies some restricted isometry conditions that, heuristically, are the best available so far. Finally, the numerical experiments of Section 4 show that the algorithm also performs well in practice.

*This work was carried out when the author was with the Université Pierre et Marie Curie in Paris. It was supported by the French National Research Agency (ANR) through the project ECHANGE (ANR-08-EMER-006).

†Drexel University, Philadelphia

2. The algorithm's rationale. In this section, we outline two families of iterative algorithms for Compressive Sensing. We then combine their premises to create the new family of Hard Thresholding Pursuit algorithms.

2.1. Iterative Hard Thresholding. The Iterative Hard Thresholding (IHT) algorithm was first introduced for sparse recovery problems by Blumensath and Davies in [2]. Elementary analyses, in particular the one in [11], show the good theoretical guarantees of this algorithm. It is built on the simple intuitions: that solving the rectangular system $A\mathbf{x} = \mathbf{y}$ amounts to solving the square system $A^*A\mathbf{x} = A^*\mathbf{y}$; that classical iterative methods suggest to define a sequence $(\mathbf{x}^n)$ by the recursion $\mathbf{x}^{n+1} = (I - A^*A)\mathbf{x}^n + A^*\mathbf{y}$; and that, since sparse vectors are desired, each step should involve the hard thresholding operator H_s that keeps s largest (in modulus) components of a vector and sets the other ones to zero.[1] This is all the more justified by the fact that the restricted isometry property — see Section 3 — ensures that the matrix A^*A behaves like the identity when its domain and range are restricted to small support sets. Thus, the contributions to $\mathbf{x}^{n+1}$ of the terms $(I - A^*A)\mathbf{x}^n$ and $A^*\mathbf{y}$ are roughly $\mathbf{x}^n - \mathbf{x}^n = 0$ and $\mathbf{x}$, so that $\mathbf{x}^{n+1}$ appears as a good approximation to $\mathbf{x}$. This yields the following algorithm, whose inputs — like all algorithms below — are the measurement vector $\mathbf{y}$, the measurement matrix A, and the anticipated sparsity s.[2] One may also prescribe the number of iterations.
Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$, typically $\mathbf{x}^0 = 0$, and iterate the scheme

$$\mathbf{x}^{n+1} = H_s(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)) \tag{IHT}$$

until a stopping criterion is met.
One may be slightly more general and consider an algorithm (IHT$^\mu$) by allowing a factor $\mu \neq 1$ in front of $A^*(\mathbf{y} - A\mathbf{x}^n)$ — this was called Gradient Descent with Sparsification (GDS) in [12].
Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$, typically $\mathbf{x}^0 = 0$, and iterate the scheme

$$\mathbf{x}^{n+1} = H_s(\mathbf{x}^n + \mu A^*(\mathbf{y} - A\mathbf{x}^n)) \tag{IHT$^\mu$}$$

until a stopping criterion is met.
One may even allow the factor μ to depend on the iteration, hence considering the Normalized Iterative Hard Thresholding (NIHT) algorithm described as follows.
Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$, typically $\mathbf{x}^0 = 0$, and iterate the scheme

$$\mathbf{x}^{n+1} = H_s(\mathbf{x}^n + \mu^n A^*(\mathbf{y} - A\mathbf{x}^n)) \tag{NIHT}$$

until a stopping criterion is met.
The original terminology of Normalized Iterative Hard Thresholding used in [4] corresponds to the specific choice[3]

$$\mu^n = \frac{\|\big(A^*(\mathbf{y} - A\mathbf{x}^n)\big)_{S^n}\|_2^2}{\|A\big(\big(A^*(\mathbf{y} - A\mathbf{x}^n)\big)_{S^n}\big)\|_2^2}, \qquad S^n := \mathrm{supp}(\mathbf{x}^n),$$

[1] In case s largest components are not uniquely define, we select the smallest possible indices.

[2] The algorithms require a prior estimation of the sparsity level. Although Basis Pursuit does not, it generally requires a prior estimation of the measurement error.

[3] For a vector $\mathbf{z} \in \mathbb{C}^N$ and a subset T of $\{1, \ldots, N\}$, the vector $\mathbf{z}_T \in \mathbb{C}^N$ denotes the vector equal to $\mathbf{z}$ on T and to zero outside of T.

unless, fixing a prescribed $0 < \eta < 1$, one has $\mu^n > \eta \|\mathbf{x}^n - \mathbf{x}^{n+1}\|_2^2 / \|A(\mathbf{x}^n - \mathbf{x}^{n+1})\|_2^2$ — the reasons for this will become apparent in (3.2). In this case, the factor μ^n is halved until the exception vanishes.

2.2. Compressive Sensing Matching Pursuit. The Compressive Sampling Matching Pursuit (CoSaMP) algorithm proposed by Needell and Tropp in [16] and the Subspace Pursuit (SP) algorithm proposed by Dai and Milenkovic in [8] do not provide better theoretical guarantees than the simple IHT algorithm, but they do offer better empirical performances. These two algorithms, grouped in a family called Compressive Sensing Matching Pursuit (CSMP) for convenience here, were devised to enhance the Orthogonal Matching Pursuit (OMP) algorithm initially proposed in [15]. The basic idea consists in chasing a good candidate for the support, and then finding the vector with this support that best fits the measurements. The algorithm can be expressed in the following way, with $t = s$ for SP and $t = 2s$ for CoSaMP.
Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$, typically $\mathbf{x}^0 = 0$, and iterate the scheme

$$U^n = \mathrm{supp}(\mathbf{x}^n) \cup \big\{\text{indices of } t \text{ largest entries of } A^*(\mathbf{y} - A\mathbf{x}^n)\big\}, \qquad (\mathrm{CSMP}_1)$$

$$\mathbf{u}^n = \mathrm{argmin}\big\{\|\mathbf{y} - A\mathbf{z}\|_2, \mathrm{supp}(\mathbf{z}) \subseteq U^n\big\}, \qquad (\mathrm{CSMP}_2)$$

$$\mathbf{x}^{n+1} = H_s(\mathbf{u}^n), \qquad (\mathrm{CSMP}_3)$$

until a stopping criterion is met.[4]
In a sense, the present approach defies intuition, because the candidate for the support uses the largest components of the vector $A^*A(\mathbf{x} - \mathbf{x}^n) \approx \mathbf{x} - \mathbf{x}^n$, and not of a vector close to $\mathbf{x}$.

2.3. Hard Thresholding Pursuit. Sticking to the basic idea of chasing a good candidate for the support then finding the vector with this support that best fits the measurements, but inspired by intuition from the IHT algorithm, it seems natural to select instead the s largest components of $\mathbf{x}^n + A^*A(\mathbf{x} - \mathbf{x}^n) \approx \mathbf{x}$. This combination of the IHT and CSMP algorithms leads to the Hard Thresholding Pursuit (HTP) algorithm described as follows.
Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$, typically $\mathbf{x}^0 = 0$, and iterate the scheme [5]

$$S^{n+1} = \big\{\text{indices of } s \text{ largest entries of } \mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n)\big\}, \qquad (\mathrm{HTP}_1)$$

$$\mathbf{x}^{n+1} = \mathrm{argmin}\big\{\|\mathbf{y} - A\mathbf{z}\|_2, \mathrm{supp}(\mathbf{z}) \subseteq S^{n+1}\big\}, \qquad (\mathrm{HTP}_2)$$

until a stopping criterion is met. A natural criterion here is $S^{n+1} = S^n$, since then $\mathbf{x}^k = \mathbf{x}^n$ for all $k \geq n$, although there is no guarantee that this should occur. As with the Iterative Hard Thresholding algorithms, we may be more general and consider an algorithm (HTP^μ) by allowing a factor $\mu \neq 1$ as follows.
Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$, typically $\mathbf{x}^0 = 0$, and iterate the scheme

$$S^{n+1} = \big\{\text{indices of } s \text{ largest entries of } \mathbf{x}^n + \mu A^*(\mathbf{y} - A\mathbf{x}^n)\big\}, \qquad (\mathrm{HTP}_1^\mu)$$

$$\mathbf{x}^{n+1} = \mathrm{argmin}\big\{\|\mathbf{y} - A\mathbf{z}\|_2, \mathrm{supp}(\mathbf{z}) \subseteq S^{n+1}\big\}, \qquad (\mathrm{HTP}_2^\mu)$$

until a stopping criterion is met.

[4] Strictly speaking, the SP algorithm performs one last projection step after the stopping criterion.
[5] In HTP and other algorithms, the vector $A^*\mathbf{y}$ and the matrix A^*A need not be calculated at each iteration.

By allowing the factor μ to depend on the iteration according to the specific choice

$$\mu^n = \frac{\|\big(A^*(\mathbf{y} - A\mathbf{x}^n)\big)_{S^n}\|_2^2}{\|A\big(\big(A^*(\mathbf{y} - A\mathbf{x}^n)\big)_{S^n}\big)\|_2^2},$$

we may also consider the Normalized Iterative Hard Thresholding algorithm described as follows.
Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$, typically $\mathbf{x}^0 = 0$, and iterate the scheme

$$S^{n+1} = \big\{\text{indices of } s \text{ largest entries of } \mathbf{x}^n + \mu^n A^*(\mathbf{y} - A\mathbf{x}^n)\big\}, \qquad (\mathrm{NHTP}_1)$$
$$\mathbf{x}^{n+1} = \mathrm{argmin}\big\{\|\mathbf{y} - A\mathbf{z}\|_2, \mathrm{supp}(\mathbf{z}) \subseteq S^{n+1}\big\}. \qquad (\mathrm{NHTP}_2)$$

until a stopping criterion is met.

In all the above algorithms, the second steps require to solve the $s \times s$ system of normal equations $A^*_{S^{n+1}} A_{S^{n+1}} \mathbf{x}^{n+1} = A^*_{S^{n+1}} \mathbf{y}$. If these steps are judged too costly, we may consider instead a fast version of the Hard Thresholding Pursuit algorithms, where the orthogonal projection is replaced by a certain number k of gradient descent iterations. This leads for instance to the algorithm (FHTP$^\mu$) described below. In the special case $\mu = 1$, we call the algorithm Fast Hard Thresholding Pursuit (FHTP) — note that $k = 0$ corresponds to the classical IHT algorithm and that $k = \infty$ corresponds to the HTP algorithm.
Start with an s-sparse $\mathbf{x}^0 \in \mathbb{C}^N$, typically $\mathbf{x}^0 = 0$, and iterate the scheme

$$S^{n+1} = \mathrm{supp}(\mathbf{u}^{n+1,1}), \qquad \mathbf{u}^{n+1,1} := H_s(\mathbf{x}^n + \mu A^*(\mathbf{y} - A\mathbf{x}^n)), \qquad (\mathrm{FHTP}^\mu_1)$$
$$\mathbf{x}^{n+1} = \mathbf{u}^{n+1,k+1}, \qquad \mathbf{u}^{n+1,\ell+1} := \big(\mathbf{u}^{n+1,\ell} + t^{n+1,\ell} A^*(\mathbf{y} - A\mathbf{u}^{n+1,\ell})\big)_{S^{n+1}}, \qquad (\mathrm{FHTP}^\mu_2)$$

until a stopping criterion is met.
A simple choice for $t^{n+1,\ell}$ is simply $t^{n+1,\ell} = 1$, while a wiser choice, corresponding to a steepest descent, is

$$t^{n+1,\ell} = \frac{\|\big(A^*(\mathbf{y} - A\mathbf{u}^{n+1,\ell})\big)_{S^{n+1}}\|_2^2}{\|A\big(\big(A^*(\mathbf{y} - A\mathbf{u}^{n+1,\ell})\big)_{S^{n+1}}\big)\|_2^2}. \qquad (2.1)$$

3. Theoretical justification. In this section, we analyze the theoretical performances of the proposed algorithms. We first show the convergence of the algorithms under some conditions on the measurement matrix A, precisely on its operator norm then on its restricted isometry constants, which are introduced along the way. Next, we study the exact recovery of sparse vectors as outputs of the proposed algorithms using perfect measurements. Sufficient conditions for successful recovery are given in terms of restricted isometry constants, and we heuristically argue that these conditions are the best available so far. Finally, we prove that these sufficient conditions also guarantee a stable and robust recovery with respect to sparsity defect and to measurement error.

3.1. Convergence. First and foremost, we make a simple observation about the HTP algorithm — or HTP$^\mu$ and NHTP, for that matter. Namely, since there is only a finite number of subsets of $\{1, \ldots, N\}$ with size s, there exist integers $n, p \geq 1$ such that $S^{n+p} = S^n$, so that (HTP$_2$) and (HTP$_1$) yield $\mathbf{x}^{n+p+1} = \mathbf{x}^{n+1}$ and $S^{n+p+1} = S^{n+1}$, and so on until $\mathbf{x}^{n+2p} = \mathbf{x}^{n+p}$ and $S^{n+2p} = S^{n+p} = S^n$. Thus, one actually shows recursively that $\mathbf{x}^{n+kp+r} = \mathbf{x}^{n+r}$ for all $k \geq 1$ and $1 \leq r \leq p$. Simply stated, this takes the following form.

LEMMA 3.1. *The sequences defined by (HTP), (HTP$^\mu$), and (NHTP) are eventually periodic.*

The importance of this observation lies in the fact that, as soon as the convergence of one of these algorithms is established, then we can certify that the limit is exactly achieved after a finite number of iterations. For instance, we establish below the convergence of the HTP algorithm under a condition on the operator norm $\|A\|_{2\to 2} := \sup_{\mathbf{x}\neq 0} \|A\mathbf{x}\|_2/\|\mathbf{x}\|_2$ of the matrix A. This parallels a result of [2], where the convergence of (IHT) was also proved under the condition $\|A\|_{2\to 2} < 1$. Our proof uses the same strategy, based on the decrease along the iterations of the quantity $\|\mathbf{y} - A\mathbf{x}^n\|_2$ (the 'cost'; note that the auxiliary 'surrogate cost' is not mentioned here).

PROPOSITION 3.2. *The sequence* $(\mathbf{x}^n)$ *defined by (HTP$^\mu$) converges in a finite number of iterations provided* $\mu\|A\|_{2\to 2}^2 < 1$.

Proof. Let us consider the vector supported on S^{n+1} defined by

$$\mathbf{u}^{n+1} := H_s(\mathbf{x}^n + \mu A^*(\mathbf{y} - A\mathbf{x}^n)).$$

According to the definition of $A\mathbf{x}^{n+1}$, we have $\|\mathbf{y} - A\mathbf{x}^{n+1}\|_2^2 \le \|\mathbf{y} - A\mathbf{u}^{n+1}\|_2^2$, and it follows that

$$\begin{aligned} \|\mathbf{y}-A\mathbf{x}^{n+1}\|_2^2 - \|\mathbf{y}-A\mathbf{x}^n\|_2^2 &\le \|\mathbf{y}-A\mathbf{u}^{n+1}\|_2^2 - \|\mathbf{y}-A\mathbf{x}^n\|_2^2 \\ &= \|A(\mathbf{x}^n-\mathbf{u}^{n+1}) + \mathbf{y}-A\mathbf{x}^n\|_2^2 - \|\mathbf{y}-A\mathbf{x}^n\|_2^2 \\ &= \|A(\mathbf{x}^n-\mathbf{u}^{n+1})\|_2^2 + 2\Re\langle A(\mathbf{x}^n-\mathbf{u}^{n+1}), \mathbf{y}-A\mathbf{x}^n\rangle. \end{aligned} \tag{3.1}$$

We now observe that $\mathbf{u}^{n+1}$ is a better s-term approximation to $\mathbf{x}^n + \mu A^*(\mathbf{y} - A\mathbf{x}^n)$ than $\mathbf{x}^n$ is, so that

$$\|\mathbf{x}^n + \mu A^*(\mathbf{y} - A\mathbf{x}^n) - \mathbf{u}^{n+1}\|_2^2 \le \|\mu A^*(\mathbf{y} - A\mathbf{x}^n)\|_2^2.$$

After expanding the squares, we obtain

$$2\mu\Re\langle \mathbf{x}^n - \mathbf{u}^{n+1}, A^*(\mathbf{y} - A\mathbf{x}^n)\rangle \le -\|\mathbf{x}^n - \mathbf{u}^{n+1}\|_2^2.$$

Substituting this into (3.1), we derive

$$\|\mathbf{y} - A\mathbf{x}^{n+1}\|_2^2 - \|\mathbf{y} - A\mathbf{x}^n\|_2^2 \le \|A(\mathbf{x}^n - \mathbf{u}^{n+1})\|_2^2 - \frac{1}{\mu}\|\mathbf{x}^n - \mathbf{u}^{n+1}\|_2^2. \tag{3.2}$$

We use the simple inequality

$$\|A(\mathbf{x}^n - \mathbf{u}^{n+1})\|_2^2 \le \|A\|_{2\to 2}^2 \, \|\mathbf{x}^n - \mathbf{u}^{n+1}\|_2^2, \tag{3.3}$$

and the hypothesis that $\|A\|_{2\to 2}^2 < 1/\mu$ to deduce that

$$\|\mathbf{y} - A\mathbf{x}^{n+1}\|_2^2 - \|\mathbf{y} - A\mathbf{x}^n\|_2^2 \le -c\,\|\mathbf{x}^n - \mathbf{u}^{n+1}\|_2^2, \tag{3.4}$$

where $c := 1/\mu - \|A\|_{2\to 2}^2$ is a positive constant. This proves that the nonnegative sequence $(\|\mathbf{y} - A\mathbf{x}^n\|_2)$ is nonincreasing, hence it is convergent. Since it is also eventually periodic, it must be eventually constant. In view of (3.4), we deduce that $\mathbf{u}^{n+1} = \mathbf{x}^n$, and in particular that $S^{n+1} = S^n$, for n large enough. This implies that $\mathbf{x}^{n+1} = \mathbf{x}^n$ for n large enough, which implies the required result. □

As seen in (3.3), it is not really the norm of A that matters, but rather its 'norm on sparse vectors'. This point motivates the introduction of the sth order restricted isometry constant $\delta_s = \delta_s(A)$ of a matrix $A \in \mathbb{C}^{m \times N}$. We recall that these are defined as the smallest $\delta \geq 0$ such that

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|A\mathbf{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2 \qquad \text{for all } s\text{-sparse vectors } \mathbf{x} \in \mathbb{C}^N. \tag{3.5}$$

Replacing (3.3) by $\|A(\mathbf{x}^n - \mathbf{u}^{n+1})\|_2^2 \leq (1+\delta_{2s})\,\|\mathbf{x}^n - \mathbf{u}^{n+1}\|_2^2$ in the previous proof immediately yields the following result.

THEOREM 3.3. *The sequence $(\mathbf{x}^n)$ defined by (HTP$^\mu$) converges in a finite number of iterations provided $\mu(1+\delta_{2s}) < 1$.*

We close this subsection with analogs of Proposition 3.2 and Theorem 3.3 for the fast versions of the Hard Thresholding Pursuit algorithm.

THEOREM 3.4. *For any $k \geq 0$, with $\mu \geq 1/2$ and with $t^{n+1,\ell}$ equal to 1 or given by* (2.1)*, the sequence $(\mathbf{x}^n)$ defined by (FTHP$^\mu$) converges provided $\mu\|A\|_{2\to2}^2 < 1$ or $\mu(1+\delta_{2s}) < 1$.*

Proof. Keeping in mind the proof of Proposition 3.2, we see that it is enough to establish the inequality $\|\mathbf{y} - A\mathbf{x}^{n+1}\|_2 \leq \|\mathbf{y} - A\mathbf{u}^{n+1}\|_2$. Since $\mathbf{x}^{n+1} = \mathbf{u}^{n+1,k+1}$ and $\mathbf{u}^{n+1} = \mathbf{u}^{n+1,1}$, we just need to prove that, for any $1 \leq \ell \leq k$,

$$\|\mathbf{y} - A\mathbf{u}^{n+1,\ell+1}\|_2 \leq \|\mathbf{y} - A\mathbf{u}^{n+1,\ell}\|_2. \tag{3.6}$$

Let $A_{S^{n+1}}$ denote the submatrix of A obtained by keeping the columns indexed by S^{n+1} and let $\mathbf{v}^{n+1,\ell+1}, \mathbf{v}^{n+1,\ell} \in \mathbb{C}^s$ denote the subvectors of $\mathbf{u}^{n+1,\ell+1}, \mathbf{u}^{n+1,\ell} \in \mathbb{C}^N$ obtained by keeping the entries indexed by S^{n+1}. With $t^{n+1,\ell} = 1$, we have

$$\begin{aligned}\|\mathbf{y} - A\mathbf{u}^{n+1,\ell+1}\|_2 &= \|\mathbf{y} - A_{S^{n+1}}\mathbf{v}^{n+1,\ell+1}\|_2 = \|(I - A_{S^{n+1}}A_{S^{n+1}}^*)(\mathbf{y} - A_{S^{n+1}}\mathbf{v}^{n+1,\ell})\|_2\\ &\leq \|\mathbf{y} - A_{S^{n+1}}\mathbf{v}^{n+1,\ell}\|_2 = \|\mathbf{y} - A\mathbf{u}^{n+1,\ell}\|_2,\end{aligned}$$

where the inequality is justified because the hermitian matrix $A_{S^{n+1}}A_{S^{n+1}}^*$ has eigenvalues in $[0, \|A\|_{2\to2}^2]$ or $[0, 1+\delta_{2s}]$, hence in $[0, 1/\mu] \subseteq [0, 2]$. Thus (3.6) holds with $t^{n+1,\ell} = 1$. With $t^{n+1,\ell}$ given by (2.1), it holds because this is actually the value that minimizes over $t = t^{n+1,\ell}$ the quadratic expression

$$\|\mathbf{y} - A_{S^{n+1}}\mathbf{v}^{n+1,\ell+1}\|_2^2 = \|\mathbf{y} - A_{S^{n+1}}\mathbf{v}^{n+1,\ell} - t\, A_{S^{n+1}}A_{S^{n+1}}^*(\mathbf{y} - A_{S^{n+1}}\mathbf{v}^{n+1,\ell+1})\|_2^2,$$

as one can easily verify. □

3.2. Exact recovery of sparse vectors from accurate measurements. We place ourselves in the ideal case where the vectors to be recovered are exactly sparse and are measured with infinite precision. Although the main result of this subsection, namely Theorem 3.5, is a particular instance of Theorem 3.8, we isolate it because its proof is especially elegant in this simple case and sheds light on the more involved proof of Theorem 3.8. Theorem 3.5 guarantees the recovery of s-sparse vectors via Hard Thresholding Pursuit under a condition on the $3s$th restricted isometry constant of the measurement matrix. Sufficient conditions of this kind, which often read $\delta_t < \delta_*$ for some integer t related to s and for some specific value δ_*, have become a benchmark for theoretical investigations, because they are pertinent in the analysis a wide range of algorithms. Note that the condition $\delta_t < \delta_*$ is mainly known to be satisfied for random matrices provided the number of measurements scales like

$$m \approx c\,\frac{t}{\delta_*^2}\,\ln(N/t).$$

Since we want to make as few measurements as possible, we may heuristically assess a sufficient condition by the smallness of the ratio t/δ_*^2. In this respect, the sufficient condition $\delta_{3s} < 1/\sqrt{3}$ of this paper — valid not only for HTP but also for FTHP (in particular for IHT, too) — is currently the best available, as shown in the following table[6]. More careful investigations should be carried out in the framework of phase transition, see e.g. [9].

Algorithm	IHT	GDS	CoSaMP	(F)HTP	BP
Reference	[11]	[12]	[11]	this paper	[10]
$\delta_t < \delta_*$	$\delta_{3s} < 0.5$	$\delta_{2s} < 0.333$	$\delta_{4s} < 0.384$	$\delta_{3s} < 0.577$	$\delta_{2s} < 0.465$
Ratio t/δ_*^2	12	18	27.08	**9**	9.243

Before turning to the main result of this subsection, we point out a less common, but sometimes preferable, expression of the restricted isometry constant, i.e.,

$$\delta_s = \max_{|S| \le s} \|A_S^* A_S - \mathrm{Id}\|_{2\to 2},$$

where A_S denotes the submatrix of A obtained by keeping the columns indexed by S. This enables to observe easily that

$$|\langle \mathbf{u}, (\mathrm{Id} - A^*A)\mathbf{v}\rangle| \le \delta_t \|\mathbf{u}\|_2 \|\mathbf{v}\|_2 \qquad \text{whenever } |\mathrm{supp}(\mathbf{u}) \cup \mathrm{supp}(\mathbf{v})| \le t. \tag{3.7}$$

Indeed, setting $T := \mathrm{supp}(\mathbf{u}) \cup \mathrm{supp}(\mathbf{v})$ and denoting by $\mathbf{u}_T$ and $\mathbf{v}_T$ the subvectors of $\mathbf{u}$ and $\mathbf{v}$ obtained by only keeping the components indexed by T,[7] we have

$$\begin{aligned} |\langle \mathbf{u}, (\mathrm{Id} - A^*A)\mathbf{v}\rangle| &= |\langle \mathbf{u}, \mathbf{v}\rangle - \langle A\mathbf{u}, A\mathbf{v}\rangle| = |\langle \mathbf{u}_T, \mathbf{v}_T\rangle - \langle A_T\mathbf{u}_T, A_T\mathbf{v}_T\rangle| \\ &= |\langle \mathbf{u}_T, (\mathrm{Id} - A_T^*A_T)\mathbf{v}_T| \le \|\mathbf{u}_T\|_2 \, \|(\mathrm{Id} - A_T^*A_T)\mathbf{v}_T\|_2 \\ &\le \|\mathbf{u}_T\|_2 \, \delta_t \|\mathbf{v}_T\|_2 = \delta_t \|\mathbf{u}\|_2 \|\mathbf{v}\|_2. \end{aligned}$$

THEOREM 3.5. *Suppose that the $3s$th order restricted isometry constant of the measurement matrix $A \in \mathbb{C}^{m\times N}$ satisfies*

$$\delta_{3s} < \frac{1}{\sqrt{3}} \approx 0.57735.$$

Then, for any s-sparse $\mathbf{x} \in \mathbb{C}^N$, the sequence $(\mathbf{x}^n)$ defined by (HTP) with $\mathbf{y} = A\mathbf{x}$ converges towards $\mathbf{x}$ at a geometric rate given by

$$\|\mathbf{x}^n - \mathbf{x}\|_2 \le \rho^n \, \|\mathbf{x}^0 - \mathbf{x}\|_2, \qquad \rho := \sqrt{\frac{2\delta_{3s}^2}{1 - \delta_{2s}^2}} < 1. \tag{3.8}$$

Proof. The first step of the proof is a consequence of (HTP_2). We notice that $A\mathbf{x}^{n+1}$ is the best ℓ_2-approximation to $\mathbf{y}$ from the space $\{A\mathbf{z}, \mathrm{supp}(\mathbf{z}) \subseteq S^{n+1}\}$, hence it is characterized by the orthogonality condition

$$\langle A\mathbf{x}^{n+1} - \mathbf{y}, A\mathbf{z}\rangle = 0 \qquad \text{whenever } \mathrm{supp}(\mathbf{z}) \subseteq S^{n+1}. \tag{3.9}$$

---

[6] We did not include the sufficient conditions $\delta_{2s} < 0.473$ and $\delta_{3s} < 0.535$ of [10] and [5], giving the ratios 8.924 and 10.44, because they are only valid for large s.

[7] This notation is in slight conflict with the one used elsewhere in the paper, where $\mathbf{u}_T$ and $\mathbf{v}_T$ would be vectors in $\mathbb{C}^N$

Since $\mathbf{y} = A\mathbf{x}$, this may be rewritten as

$$\langle \mathbf{x}^{n+1} - \mathbf{x}, A^* A \mathbf{z} \rangle = 0 \qquad \text{whenever } \operatorname{supp}(\mathbf{z}) \subseteq S^{n+1}.$$

We derive in particular

$$\begin{aligned}
\|(\mathbf{x}^{n+1} - \mathbf{x})_{S^{n+1}}\|_2^2 &= \langle \mathbf{x}^{n+1} - \mathbf{x}, (\mathbf{x}^{n+1} - \mathbf{x})_{S^{n+1}} \rangle \\
&= \langle \mathbf{x}^{n+1} - \mathbf{x}, (I - A^* A)\big((\mathbf{x}^{n+1} - \mathbf{x})_{S^{n+1}}\big) \rangle \\
&\le \delta_{2s} \|\mathbf{x}^{n+1} - \mathbf{x}\|_2 \|(\mathbf{x}^{n+1} - \mathbf{x})_{S^{n+1}}\|_2.
\end{aligned}$$

After simplification, we have $\|(\mathbf{x}^{n+1} - \mathbf{x})_{S^{n+1}}\|_2 \le \delta_{2s} \|\mathbf{x}^{n+1} - \mathbf{x}\|_2$. It follows that

$$\begin{aligned}
\|\mathbf{x}^{n+1} - \mathbf{x}\|_2^2 &= \|(\mathbf{x}^{n+1} - \mathbf{x})_{\overline{S^{n+1}}}\|_2^2 + \|(\mathbf{x}^{n+1} - \mathbf{x})_{S^{n+1}}\|_2^2 \\
&\le \|(\mathbf{x}^{n+1} - \mathbf{x})_{\overline{S^{n+1}}}\|_2^2 + \delta_{2s}^2 \|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2^2.
\end{aligned}$$

After a rearrangement, we obtain

$$\|\mathbf{x}^{n+1} - \mathbf{x}\|_2^2 \le \frac{1}{1 - \delta_{2s}^2} \|(\mathbf{x}^{n+1} - \mathbf{x})_{\overline{S^{n+1}}}\|_2^2. \tag{3.10}$$

The second step of the proof is as a consequence of (HTP$_1$). With $S := \operatorname{supp}(\mathbf{x})$, we notice that

$$\|(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))_S\|_2^2 \le \|(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))_{S^{n+1}}\|_2^2.$$

Eliminating the contribution on $S \cap S^{n+1}$, we derive

$$\|(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))_{S \setminus S^{n+1}}\|_2 \le \|(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))_{S^{n+1} \setminus S}\|_2. \tag{3.11}$$

For the right-hand side, we have

$$\|(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))_{S^{n+1} \setminus S}\|_2 = \|\big((I - A^* A)(\mathbf{x}^n - \mathbf{x})\big)_{S^{n+1} \setminus S}\|_2.$$

As for the left-hand side, we have

$$\begin{aligned}
\|(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))_{S \setminus S^{n+1}}\|_2 &= \|(\mathbf{x} - \mathbf{x}^{n+1})_{\overline{S^{n+1}}} + \big((I - A^* A)(\mathbf{x}^n - \mathbf{x})\big)_{S \setminus S^{n+1}}\|_2 \\
&\ge \|(\mathbf{x} - \mathbf{x}^{n+1})_{\overline{S^{n+1}}}\|_2 - \|\big((I - A^* A)(\mathbf{x}^n - \mathbf{x})\big)_{S \setminus S^{n+1}}\|_2.
\end{aligned}$$

It follows that

$$\begin{aligned}
\|(\mathbf{x} - \mathbf{x}^{n+1})_{\overline{S^{n+1}}}\|_2 &\le \|\big((I - A^* A)(\mathbf{x}^n - \mathbf{x})\big)_{S \setminus S^{n+1}}\|_2 + \|\big((I - A^* A)(\mathbf{x}^n - \mathbf{x})\big)_{S^{n+1} \setminus S}\|_2 \\
&\le \sqrt{2}\, \|\big((I - A^* A)(\mathbf{x}^n - \mathbf{x})\big)_{S \Delta S^{n+1}}\|_2 \le \sqrt{2}\, \delta_{3s} \|\mathbf{x}^n - \mathbf{x}\|_2. \qquad (3.12)
\end{aligned}$$

As a final step, we put (3.10) and (3.12) together to obtain

$$\|\mathbf{x}^{n+1} - \mathbf{x}\|_2 \le \sqrt{\frac{2\delta_{3s}^2}{1 - \delta_{2s}^2}}\, \|\mathbf{x}^n - \mathbf{x}\|_2.$$

The estimate (3.8) immediately follows. We point out that the multiplicative coefficient $\rho := \sqrt{2\delta_{3s}^2/(1 - \delta_{2s}^2)}$ is less than one as soon as $2\delta_{3s}^2 < 1 - \delta_{2s}^2$. Since $\delta_{2s} \le \delta_{3s}$, this occurs as soon as $\delta_{3s} < 1/\sqrt{3}$. □

As noticed earlier, the convergence requires a finite number of iterations, which can be estimated as follows.

COROLLARY 3.6. *Suppose that the matrix $A \in \mathbb{C}^{m \times N}$ satisfies $\delta_{3s} < 1/\sqrt{3}$. Then any s-sparse vector $\mathbf{x} \in \mathbb{C}^N$ is recovered by (HTP) with $\mathbf{y} = A\mathbf{x}$ in at most*

$$\left\lceil \frac{\ln\left(\sqrt{2/3}\,\|\mathbf{x}^0 - \mathbf{x}\|_2/\xi\right)}{\ln\left(1/\rho\right)} \right\rceil + 1 \quad \textit{iterations}, \tag{3.13}$$

where $\rho := \sqrt{2\delta_{3s}^2/(1-\delta_{2s}^2)}$ and ξ is the smallest nonzero entry of $\mathbf{x}$ in modulus.

Proof. We need to determine an integer n such that $S^n = S$, since then (HTP_2) implies $\mathbf{x}^n = \mathbf{x}$. According to the definition of S^n, this occurs if, for all $j \in S$ and all $\ell \in \overline{S}$, we have

$$|(\mathbf{x}^{n-1} + A^*A(\mathbf{x} - \mathbf{x}^{n-1}))_j| > |(\mathbf{x}^{n-1} + A^*A(\mathbf{x} - \mathbf{x}^{n-1}))_\ell|. \tag{3.14}$$

We observe that

$$|(\mathbf{x}^{n-1}+A^*A(\mathbf{x}-\mathbf{x}^{n-1}))_j| = |\mathbf{x}_j + \big((I-A^*A)(\mathbf{x}^{n-1}-\mathbf{x})\big)_j| \geq \xi - |\big((I-A^*A)(\mathbf{x}^{n-1}-\mathbf{x})\big)_j|,$$

and that

$$|(\mathbf{x}^{n-1} + A^*A(\mathbf{x} - \mathbf{x}^{n-1}))_\ell| = |\big((I - A^*A)(\mathbf{x}^{n-1} - \mathbf{x})\big)_\ell|.$$

Then, in view of

$$\begin{aligned}
|\big((I - A^*A)(\mathbf{x}^{n-1} - \mathbf{x})\big)_j| &+ |\big((I - A^*A)(\mathbf{x}^{n-1} - \mathbf{x})\big)_\ell| \\
&\leq \sqrt{2}\,\|\big((I - A^*A)(\mathbf{x}^{n-1} - \mathbf{x})\big)_{\{j,\ell\}}\|_2 \leq \sqrt{2}\,\delta_{3s}\|\mathbf{x}^{n-1} - \mathbf{x}\|_2 \\
&< \sqrt{2/3}\,\rho^{n-1}\|\mathbf{x}^0 - \mathbf{x}\|_2,
\end{aligned}$$

we see that (3.14) is satisfied as soon as

$$\xi \geq \sqrt{2/3}\,\rho^{n-1}\|\mathbf{x}^0 - \mathbf{x}\|_2.$$

The smallest such integer n is the one given by (3.13). □

Turning our attention to the fast version of the Hard Thresholding Pursuit algorithm, it is interesting to notice that s-sparse recovery via (FHTP) is also guaranteed by the condition $\delta_{3s} < 1/\sqrt{3}$, independently on the number k of descent iterations used in (FHTP_2). Note that here we do not make the default choice for $t^{n+1,\ell}$ given by (2.1), but we simply choose $t^{n+1,\ell} = 1$, which in practice is not optimal. With $k = 0$, the result means that the classical IHT algorithm also allows s-sparse recovery as soon as $\delta_{3s} < 1/\sqrt{3}$, which incidentally improves the best condition [11] found in the current literature.

THEOREM 3.7. *Suppose that the $3s$th order restricted isometry constant of the measurement matrix $A \in \mathbb{C}^{m \times N}$ satisfies*

$$\delta_{3s} < \frac{1}{\sqrt{3}} \approx 0.57735.$$

Then, for any s-sparse $\mathbf{x} \in \mathbb{C}^N$, the sequence $(\mathbf{x}^n)$ defined by (FHTP) with $\mathbf{y} = A\mathbf{x}$, $k \geq 0$, and $t^{n+1,\ell} = 1$ converges towards $\mathbf{x}$ at a geometric rate given by

$$\|\mathbf{x}^n - \mathbf{x}\|_2 \leq \rho^n\,\|\mathbf{x}^0 - \mathbf{x}\|_2, \qquad \rho := \sqrt{\frac{\delta_{3s}^{2k+2}(1 - 3\delta_{3s}^2) + 2\delta_{3s}^2}{1 - \delta_{3s}^2}} < 1. \tag{3.15}$$

Proof. Exactly as in the proof of Theorem 3.5, we can still derive (3.12) from ($FHTP_1$), i.e.,

$$\|(\mathbf{x} - \mathbf{x}^{n+1})_{\overline{S^{n+1}}}\|_2^2 \le 2\delta_{3s}^2 \|\mathbf{x} - \mathbf{x}^n\|_2^2. \tag{3.16}$$

Let us now examine the consequence of ($FHTP_2$). With $\mathbf{u}^{n+1,0} := \mathbf{x}^n$, we can write, for each $0 \le \ell \le k$,

$$\begin{aligned}
\|\mathbf{x} - \mathbf{u}^{n+1,\ell+1}\|_2^2 &= \|(\mathbf{x} - \mathbf{u}^{n+1,\ell+1})_{S^{n+1}}\|_2^2 + \|(\mathbf{x} - \mathbf{u}^{n+1,\ell+1})_{\overline{S^{n+1}}}\|_2^2 \\
&= \|\big((I - A^*A)(\mathbf{x} - \mathbf{u}^{n+1,\ell})\big)_{S^{n+1}}\|_2^2 + \|\mathbf{x}_{\overline{S^{n+1}}}\|_2^2 \\
&\le \delta_{2s}^2 \|\mathbf{x} - \mathbf{u}^{n+1,\ell}\|_2^2 + \|\mathbf{x}_{\overline{S^{n+1}}}\|_2^2.
\end{aligned}$$

This yields, by immediate induction on ℓ,

$$\|\mathbf{x} - \mathbf{u}^{n+1,k+1}\|_2^2 \le \delta_{2s}^{2k+2} \|\mathbf{x} - \mathbf{u}^{n+1,0}\|_2^2 + (\delta_{2s}^{2k} + \cdots + \delta_{2s}^2 + 1)\|\mathbf{x}_{\overline{S^{n+1}}}\|_2^2.$$

In other words, we have

$$\|\mathbf{x} - \mathbf{x}^{n+1}\|_2^2 \le \delta_{2s}^{2k+2} \|\mathbf{x} - \mathbf{x}^n\|_2^2 + \frac{1 - \delta_{2s}^{2k+2}}{1 - \delta_{2s}^2} \|(\mathbf{x} - \mathbf{x}^{n+1})_{\overline{S^{n+1}}}\|_2^2. \tag{3.17}$$

From (3.16), (3.17), and the simple inequality $\delta_{2s} \le \delta_{3s}$, we derive

$$\|\mathbf{x} - \mathbf{x}^{n+1}\|_2^2 \le \frac{\delta_{3s}^{2k+2}(1 - 3\delta_{3s}^2) + 2\delta_{3s}^2}{1 - \delta_{3s}^2} \|\mathbf{x} - \mathbf{x}^n\|_2^2.$$

The estimate (3.15) immediately follows. We point out that, for any $k \ge 0$, the multiplicative coefficient ρ is less than one as soon as $\delta_{3s} < 1/\sqrt{3}$. □

Note that, although the sequence $(\mathbf{x}^n)$ does not converge towards x in a finite number of iterations, we can still estimate the number of iterations needed to approximate x with an ℓ_2-error not exceeding ϵ as

$$n_\epsilon = \left\lceil \frac{\ln\big(\|\mathbf{x}^0 - \mathbf{x}\|_2/\epsilon\big)}{\ln\big(1/\rho\big)} \right\rceil.$$

3.3. Approximate recovery of vectors from flawed measurements. In this section, we extend the previous results to the case of vectors that are not exactly sparse and that are not measured with perfect precision. Precisely, we prove that the HTP algorithm is stable and robust with respect to sparsity defect and to measurement error under the same sufficient condition on δ_{3s}. To this end, we need to remark that, for any $\mathbf{e} \in \mathbb{C}^N$,

$$\|(A^*\mathbf{e})_S\|_2 \le \sqrt{1 + \delta_s}\, \|\mathbf{e}\|_2 \qquad \text{whenever } |S| \le s. \tag{3.18}$$

To see this, we write

$$\begin{aligned}
\|(A^*\mathbf{e})_S\|_2^2 = \langle A^*\mathbf{e}, (A^*\mathbf{e})_S\rangle = \langle \mathbf{e}, A\big((A^*\mathbf{e})_S\big)\rangle &\le \|\mathbf{e}\|_2\, \|A\big((A^*\mathbf{e})_S\big)\|_2 \\
&\le \|\mathbf{e}\|_2 \sqrt{1 + \delta_s}\, \|(A^*\mathbf{e})_S\|_2,
\end{aligned}$$

and we simplify by $\|(A^*\mathbf{e})_S\|_2$. Let us now state the main result of this subsection.

THEOREM 3.8. *Suppose that the $3s$th order Restricted Isometry Constant of the measurement matrix $A \in \mathbb{C}^{m\times N}$ satisfies*

$$\delta_{3s} < \frac{1}{\sqrt{3}} \approx 0.57735.$$

Then, for any $\mathbf{x} \in \mathbb{C}^N$ and any $\mathbf{e} \in \mathbb{C}^m$, if S denotes an index set of s largest (in modulus) entries of $\mathbf{x}$, the sequence $(\mathbf{x}^n)$ defined by (HTP) with $\mathbf{y} = A\mathbf{x}+\mathbf{e}$ satisfies

$$\|\mathbf{x}^n - \mathbf{x}_S\|_2 \le \rho^n \|\mathbf{x}^0 - \mathbf{x}_S\|_2 + \tau \, \frac{1-\rho^n}{1-\rho} \, \|A\mathbf{x}_{\overline{S}} + \mathbf{e}\|_2, \qquad \textit{all } n \ge 0, \tag{3.19}$$

where

$$\rho := \sqrt{\frac{2\delta_{3s}^2}{1-\delta_{2s}^2}} < 1 \qquad \textit{and} \qquad \tau := \frac{\sqrt{2(1-\delta_{2s})} + \sqrt{1+\delta_s}}{1-\delta_{2s}} \le 5.15.$$

Proof. The proof follows the proof of Theorem 3.5 closely, starting with a consequence of (HTP_2) and continuing with a consequence of (HTP_1). We notice first that the orthogonality characterization (3.9) of $A\mathbf{x}^{n+1}$ is still valid, so that, writing $\mathbf{y} = A\mathbf{x}_S + \mathbf{e}'$ with $\mathbf{e}' := A\mathbf{x}_{\overline{S}} + \mathbf{e}$, we have

$$\langle \mathbf{x}^{n+1} - \mathbf{x}_S, A^*A\mathbf{z}\rangle = \langle \mathbf{e}', A\mathbf{z}\rangle \qquad \text{whenever } \operatorname{supp}(\mathbf{z}) \subseteq S^{n+1}.$$

We derive in particular

$$\begin{aligned}
\|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{S^{n+1}}\|_2^2 &= \langle \mathbf{x}^{n+1} - \mathbf{x}_S, (\mathbf{x}^{n+1} - \mathbf{x}_S)_{S^{n+1}}\rangle \\
&= \langle \mathbf{x}^{n+1} - \mathbf{x}_S, (I - A^*A)\big((\mathbf{x}^{n+1} - \mathbf{x}_S)_{S^{n+1}}\big)\rangle + \langle \mathbf{e}', A\big((\mathbf{x}^{n+1} - \mathbf{x}_S)_{S^{n+1}}\big)\rangle \\
&\underset{(3.7)}{\le} \delta_{2s}\|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2 \|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{S^{n+1}}\|_2 + \|\mathbf{e}'\|_2 \, \sqrt{1+\delta_s} \, \|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{S^{n+1}}\|_2.
\end{aligned}$$

After simplification, we have $\|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{S^{n+1}}\|_2 \le \delta_{2s}\|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2 + \sqrt{1+\delta_s}\,\|\mathbf{e}'\|_2$. It follows that

$$\begin{aligned}
\|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2^2 &= \|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{\overline{S^{n+1}}}\|_2^2 + \|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{S^{n+1}}\|_2^2 \\
&\le \|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{\overline{S^{n+1}}}\|_2^2 + \big(\delta_{2s}\|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2 + \sqrt{1+\delta_s}\,\|\mathbf{e}'\|_2\big)^2.
\end{aligned}$$

This reads $P(\|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2) \le 0$ for the quadratic polynomial defined by

$$P(t) := (1-\delta_{2s}^2)\, t^2 - (2\delta_{2s}\sqrt{1+\delta_s}\,\|\mathbf{e}'\|_2)\, t - (\|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{\overline{S^{n+1}}}\|_2^2 + (1+\delta_s)\|\mathbf{e}'\|_2^2).$$

Hence $\|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2$ is bounded by the largest root of P, i.e.,

$$\|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2 \le \frac{\delta_{2s}\sqrt{1+\delta_s}\,\|\mathbf{e}'\|_2 + \sqrt{(1-\delta_{2s}^2)\|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{\overline{S^{n+1}}}\|_2 + (1+\delta_s)\|\mathbf{e}'\|_2}}{1-\delta_{2s}^2}.$$

Using the fact that $\sqrt{a^2+b^2} \le a+b$ for $a, b \ge 0$, we obtain

$$\|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2 \le \frac{1}{\sqrt{1-\delta_{2s}^2}}\|(\mathbf{x}^{n+1} - \mathbf{x}_S)_{\overline{S^{n+1}}}\|_2 + \frac{\sqrt{1+\delta_s}}{1-\delta_{2s}}\|\mathbf{e}'\|_2. \tag{3.20}$$

We now notice that (HTP_1) still implies (3.11). For the right-hand side of (3.11), we have

$$\begin{aligned}\|(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))_{S^{n+1}\setminus S}\|_2 &= \|\big((I - A^*A)(\mathbf{x}^n - \mathbf{x}_S) + A^*\mathbf{e}'\big)_{S^{n+1}\setminus S}\|_2\\ &\le \|\big((I - A^*A)(\mathbf{x}^n - \mathbf{x}_S)\big)_{S^{n+1}\setminus S}\|_2 + \|(A^*\mathbf{e}')_{S^{n+1}\setminus S}\|_2.\end{aligned}$$

As for the left-hand side of (3.11), we have

$$\begin{aligned}\|(\mathbf{x}^n + A^*(\mathbf{y} - A\mathbf{x}^n))_{S\setminus S^{n+1}}\|_2 &= \|\big(\mathbf{x}_S + (I - A^*A)(\mathbf{x}^n - \mathbf{x}_S) + A^*\mathbf{e}'\big)_{S\setminus S^{n+1}}\|_2\\ &\ge \|(\mathbf{x}_S - \mathbf{x}^{n+1})_{\overline{S^{n+1}}}\|_2 - \|\big((I - A^*A)(\mathbf{x}^n - \mathbf{x}_S)\big)_{S\setminus S^{n+1}}\|_2 - \|(A^*\mathbf{e}')_{S\setminus S^{n+1}}\|_2.\end{aligned}$$

It follows that

$$\begin{aligned}\|(\mathbf{x}_S - \mathbf{x}^{n+1})_{\overline{S^{n+1}}}\|_2 &\le \|\big((I - A^*A)(\mathbf{x}^n - \mathbf{x}_S)\big)_{S\setminus S^{n+1}}\|_2 + \|\big((I - A^*A)(\mathbf{x}^n - \mathbf{x}_S)\big)_{S^{n+1}\setminus S}\|_2\\ &\quad + \|(A^*\mathbf{e}')_{S\setminus S^{n+1}}\|_2 + \|(A^*\mathbf{e}')_{S^{n+1}\setminus S}\|_2\\ &\le \sqrt{2}\,\big[\|\big((I - A^*A)(\mathbf{x}^n - \mathbf{x}_S)\big)_{S\Delta S^{n+1}}\|_2 + \|(A^*\mathbf{e}')_{S\Delta S^{n+1}}\|_2\big]\\ &\underset{(3.7)-(3.18)}{\le} \sqrt{2}\,\big[\delta_{3s}\|\mathbf{x}^n - \mathbf{x}_S\|_2 + \sqrt{1+\delta_{2s}}\,\|\mathbf{e}'\|_2\big]. \qquad (3.21)\end{aligned}$$

As a final step, we put (3.20) and (3.21) together to obtain

$$\|\mathbf{x}^{n+1} - \mathbf{x}_S\|_2 \le \sqrt{\frac{2\delta_{3s}^2}{1 - \delta_{2s}^2}}\,\|\mathbf{x}^n - \mathbf{x}_S\|_2 + \frac{\sqrt{2(1-\delta_{2s})} + \sqrt{1+\delta_s}}{1 - \delta_{2s}}\,\|\mathbf{e}'\|_2.$$

Again, we point out that the multiplicative coefficient $\rho := \sqrt{2\delta_{3s}^2/(1-\delta_{2s}^2)}$ is less than one as soon as $\delta_{3s} < \sqrt{3}$. In this case, the estimate (3.19) easily follows. □

We can deduce from Theorem 3.8 some error estimates that are comparable to the ones available for Basis Pursuit. We include the argument here because it does not seem to be standard, although somewhat known, see [16, Remark 2.3] and [1, p. 87]. It actually applies to any algorithm producing $(c\,s)$-sparse vectors for which the estimates (3.19) are available, hence also to IHT and to CSMP. A key inequality in the proof goes back to Stechkin, and reads, for $p \ge 1$,

$$\sigma_s(\mathbf{x})_p \le \frac{1}{s^{1-1/p}}\|\mathbf{x}\|_1, \qquad \mathbf{x} \in \mathbb{C}^N. \qquad (3.22)$$

An improved inequality can be found in [13] for $p = 2$, and a sharp and more general inequality can be found in [11] for any $p \ge 1$.

COROLLARY 3.9. *Suppose that the matrix $A \in \mathbb{C}^{m\times N}$ satisfies $\delta_{6s} < 1/\sqrt{3}$. Then, for any $\mathbf{x} \in \mathbb{C}^N$ and any $\mathbf{e} \in \mathbb{C}^m$, every cluster point $\mathbf{x}^\star$ of the sequence $(\mathbf{x}^n)$ defined by (HTP) with s replaced by $2s$ and with $\mathbf{y} = A\mathbf{x} + \mathbf{e}$ satisfies*

$$\|\mathbf{x} - \mathbf{x}^\star\|_p \le \frac{C}{s^{1-1/p}}\sigma_s(\mathbf{x})_1 + D s^{1/2-1/p}\|\mathbf{e}\|_2, \qquad 1 \le p \le 2,$$

where the constants C and D depend only on δ_{6s}.

Proof. Let S_0 be an index set of s largest components of $\mathbf{x}$, S_1 an index set of s next largest components of $\mathbf{x}$, etc. It is classical to notice that, for $k \ge 1$,

$$\|\mathbf{x}_{S_k}\|_2 \le \frac{\|\mathbf{x}_{S_{k-1}}\|_1}{\sqrt{s}}. \qquad (3.23)$$

We then obtain

$$\begin{aligned}
\|\mathbf{x}-\mathbf{x}^\star\|_p &\le \|\mathbf{x}_{\overline{S_0\cup S_1}}\|_p + \|\mathbf{x}^\star - \mathbf{x}_{S_0\cup S_1}\|_p \le \|\mathbf{x}_{\overline{S_0\cup S_1}}\|_p + (4s)^{1/p-1/2}\|\mathbf{x}^\star - \mathbf{x}_{S_0\cup S_1}\|_2 \\
&\underset{(3.19)}{\le} \|\mathbf{x}_{\overline{S_0\cup S_1}}\|_p + (4s)^{1/p-1/2}\frac{\tau}{1-\rho}\|A\mathbf{x}_{\overline{S_0\cup S_1}} + \mathbf{e}\|_2 \\
&\underset{(3.22)}{\le} \frac{1}{s^{1-1/p}}\|\mathbf{x}_{\overline{S_0}}\|_1 + (4s)^{1/p-1/2}\frac{\tau}{1-\rho}\|A\mathbf{x}_{\overline{S_0\cup S_1}} + \mathbf{e}\|_2 \\
&\le \frac{1}{s^{1-1/p}}\sigma_s(\mathbf{x})_1 + (4s)^{1/p-1/2}\frac{\tau}{1-\rho}\Big(\|A\mathbf{x}_{S_2}\|_2 + \|A\mathbf{x}_{S_3}\|_2 + \cdots + \|\mathbf{e}\|_2\Big) \\
&\le \frac{1}{s^{1-1/p}}\sigma_s(\mathbf{x})_1 + (4s)^{1/p-1/2}\frac{\tau}{1-\rho}\Big(\sqrt{1+\delta_s}\,(\|\mathbf{x}_{S_2}\|_2 + \|\mathbf{x}_{S_3}\|_2 + \cdots) + \|\mathbf{e}\|_2\Big) \\
&\underset{(3.23)}{\le} \frac{1}{s^{1-1/p}}\sigma_s(\mathbf{x})_1 + (4s)^{1/p-1/2}\frac{\tau}{1-\rho}\Big(\sqrt{1+\delta_s}\,\frac{\|\mathbf{x}_{\overline{S_0}}\|_1}{\sqrt{s}} + \|\mathbf{e}\|_2\Big) \\
&= \frac{C}{s^{1-1/p}}\sigma_s(\mathbf{x})_1 + D s^{1/2-1/p}\|\mathbf{e}\|_2,
\end{aligned}$$

where $C := 1 + 4^{1/p-1/2}\tau\sqrt{1+\delta_{6s}}/(1-\rho)$ and $D := 4^{1/p-1/2}\tau/(1-\rho)$. □

Remark: From (3.19) with $n = \infty$, it is tempting to derive an ℓ_2-instance optimal estimate of the type

$$\|\mathbf{x}-\mathbf{x}^\star\|_2 \le C\,\sigma_s(\mathbf{x})_2 + D\|\mathbf{e}\|_2, \qquad C = C'\|A\|_{2\to 2}.$$

However, it is known that ℓ_2-instance optimal estimates are only possible for $m \ge cN$, see [7]. This implies that, if $\delta_{3s} < 1/\sqrt{3}$, then $\|A\|_{2\to 2}$ cannot be bounded independently of m and N, unless $m \ge cN$. Therefore, the convergence of (HTP) — or of (HTP$^\mu$) — for random matrices is not explained by Proposition 3.2. Note that it is not explained by Theorem 3.3 either.

4. Computational investigations. Although we have obtained better theoretical guarantees for the Hard Thresholding Pursuit algorithms than for other algorithms, this does not say much about empirical performances, because there is no way to verify conditions of the type $\delta_t \le \delta_*$. This section now aims at supporting the claim that the proposed algorithms are also competitive in practice — at least in the situations considered. Of course, one has to be cautious about the validity of the conclusions drawn from our computational investigations: our numerical experiments only involved the dimensions $N = 1000$ and $m = 200$, they assess average situations while theoretical considerations assess worst-case situation, etc. Nonetheless, they allow to raise intriguing questions. The interested readers are invited to make up their own opinion based on the codes for HTP and FHTP available on the author's web page. It should also be mentioned that these experiments used the author's nonoptimal implementation — and usage — of the NIHT, CoSaMP, SP, ℓ_1-magic, and NESTA algorithms, which may not do them justice.

4.1. On the number of iterations. The first issue we investigated concerns the number of iterations needed in the HTP algorithm, since we know that, assuming its convergence, only a finite number of iterations are required. The experiment consisted in running the HTP algorithm 100 times — 25 realizations of Gaussian matrices A and 4 realizations of s-sparse Gaussian vectors x per matrix realization — with inputs s, A, and $\mathbf{y} = A$x. For each $45 \le s \le 85$, we

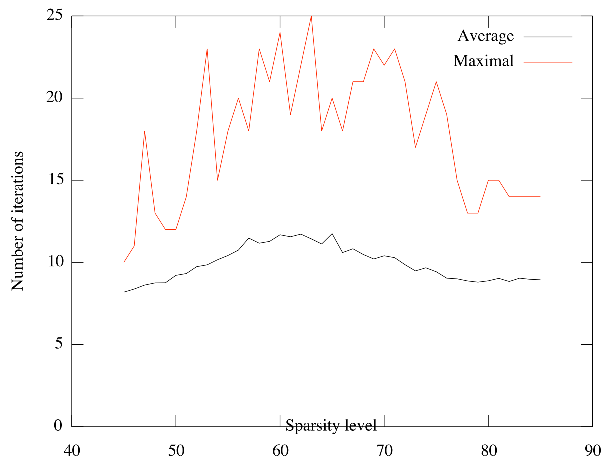

FIG. 4.1. *Average and maximal number of iterations in the HTP algorithm*

have recorded the average and maximal number of iterations performed until the stopping criterion was reached. The results are reported on Figure 4.1. We see that the number of iterations never exceeded 25, and that its average fluctuated around 10. This partly explains the observed speed of the HTP algorithm. Let us point out that, in this experiment, we have not encountered a case of nonconvergence of the algorithm (although the limit was not always the original sparse vector $\mathbf{x}$), but that nonconvergence was occasionally detected for measurement vectors $\mathbf{y}$ not of the form $A\mathbf{x}$ for some sparse vector $\mathbf{x}$. This raises the question of finding conditions that guarantee the convergence of the HTP algorithm. The remark at the end of Subsection 3.3 has to be kept in mind in this respect.

4.2. On various thresholding algorithms. The second issue we investigated concerns the empirical performances of different algorithms in the IHT and HTP families. Precisely, we tested the classical Iterative Hard Thresholding, the Iterative Hard Thresholding with parameter $\mu = 1/3$ (as advocated in [12]), the Normalized Iterative Hard Thresholding of [4], the Normalized Hard Thresholding Pursuit, the Fast Hard Thresholding Pursuit with default parameters $\mu = 1$, $k = 3$, and $t^{n+1,\ell}$ given by (2.1), and finally the Hard Thresholding Pursuit. For a sparsity level in the range $1 \leq s \leq 80$, each of these algorithms was run 100 times — 50 realizations of Gaussian matrices A and 2 realizations of s-sparse Gaussian vectors $\mathbf{x}$ per matrix realization — with inputs s, A, and $\mathbf{y} = A\mathbf{x}$ to produce outputs $\mathbf{x}^\star$. We have recorded a successful recovery if $\|\mathbf{x} - \mathbf{x}^\star\|_2 < 10^{-3}$. For each of the algorithms, the number of successful recoveries is shown as a function of the sparsity level in Figure 4.2. The results confirm the observations that it is better to use a parameter $\mu < 1$ in IHT$^\mu$ (although more iterations are required) and even better to use NIHT. However, the latter is outperformed by the family of HTP algorithms, and it is interesting to notice how drastically the step (HTP$_2$) improves the performance of IHT. Moreover, we observe that it is not beneficial to use the normalized version of HTP, and that the fast version of HTP performs reasonably well, even with only 3 descent iterations to replace the projection step. We have also kept track of the time consumed by each algorithm

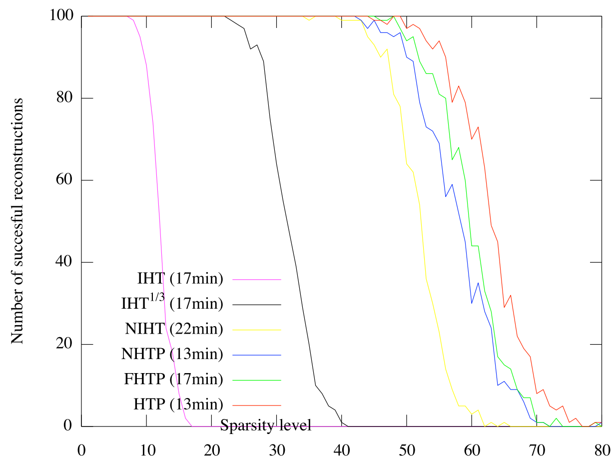

FIG. 4.2. *Comparison of various thresholding algorithms*

in the sparsity range $20 \leq s \leq 80$. The NHTP and HTP algorithms appeared to be the fastest. This is probably due to the fact that the stopping criteria were reached relatively quickly (see Subsection 4.1), while the stopping criterion for FHTP, namely $\|\mathbf{y} - A\mathbf{x}^n\|_2 < 10^{-4}$, required more iterations to be reached. No stopping criteria were set for IHT, IHT$^{1/3}$, and NIHT. Instead, 80 iterations were systematically carried out, which explains the relative slowness of these algorithms here.

4.3. On the parameter μ. The third issue we investigated concerns the dependence of the HTP$^\mu$ algorithm on μ. For the sparsity range $30 \leq s \leq 90$, and for values of μ equal to 0.4, 0.7, 1, 1.3, 1.6, we run the corresponding Hard Thresholding Pursuit algorithms 100 times — 50 realizations of Gaussian matrices A and 2 realizations of s-sparse Gaussian vectors $\mathbf{x}$ per matrix realization — with inputs s, A, and $\mathbf{y} = A\mathbf{x}$. Figure 4.3 shows the number of successful recoveries as a function of the sparsity level. It is quite surprising to observe a behavior contrasting with the one for the IHT$^\mu$ algorithm. Namely, it is better to use a parameter $\mu > 1$ (although more iterations are required). This raises the question of understanding the fundamental difference in the structure of the algorithms in order to tune the choice of μ.

4.4. On the compared performances of classical algorithms. The last but not least issue we investigated concerns the relative performance of the HTP algorithm compared with other classical algorithms, namely the IHT, CoSaMP, SP, and BP algorithms. We have been more thorough in this experiment than in the previous ones by testing not only Gaussian matrices and Gaussian vectors, but also Bernoulli and partial Fourier matrices and Bernoulli vectors. Again, all the algorithms were run 100 times — 4 vector realizations for each of the 25 matrix realizations. The results are summarized in Figure 4.4, where the first row corresponds to Gaussian matrices, the second row to Bernoulli matrices, and the third row to partial Fourier matrices, while the first column corresponds to sparse Gaussian vectors and the second column to sparse Bernoulli vectors. Even

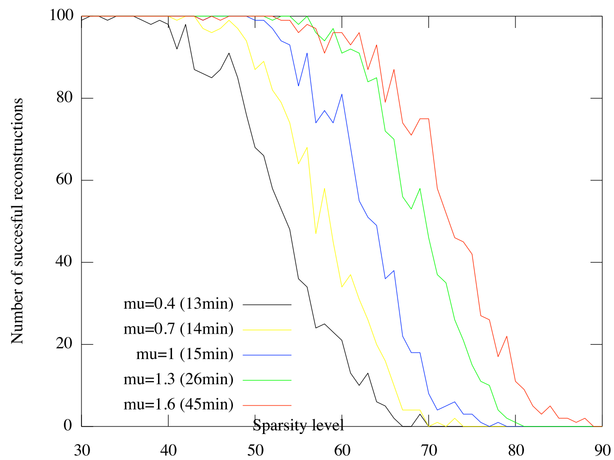

FIG. 4.3. *Influence of the factor μ in the HTP$^\mu$ algorithm*

with the nonoptimal choice $\mu = 1$, we observe that the HTP algorithm genuinely outperforms other algorithms for Gaussian vectors, but not for Bernoulli vectors. Such a phenomenon was already observed in [8] when comparing the SP and BP algorithms. Note that the BP algorithm behaves similarly for Gaussian or Bernoulli vectors, which is consistent with the theoretical observation that the recovery of a sparse vector via ℓ_1-minimization depends (in the real setting) only on the sign pattern of this vector. For partial Fourier matrices, we notice that the HTP algorithm becomes competitive again with Bernoulli vectors. We finally point out that we have kept track of the time consumed by each algorithm in the sparsity range delimited by the two $|$ in each plot (e.g. the range $35 \le s \le 85$ for Gaussian matrices and Gaussian vectors), but that HTP is actually faster than indicated (for Gaussian matrices and Gaussian vectors, the time should not really be the 37 minutes of Figure 4.4, but in fact less than the 15 minutes of Figure 4.3). This is because, for lack of a clear-cut stopping criterion for all iterative algorithms, we systematically used 50 iterations. Even in this case, the HTP algorithm appears at least twice as fast as any other algorithm — IHT excluded. This confirms speed as a key asset of the algorithm.

5. Conclusion. We have introduced a new iterative algorithm, called Hard Thresholding Pursuit, designed to find sparse solutions of underdetermined linear systems. One of the strong features of the algorithm is its speed, which is accounted for by its simplicity and by the fact that convergence only takes a finite number of iterations. We have also given an elegant proof of its good theoretical performance, as we have shown that the Hard Thresholding Pursuit algorithm finds any s-sparse solution of a linear system (in a stable and robust way) if the restricted isometry constant of the matrix of the system satisfies $\delta_{3s} < 1/\sqrt{3}$. We have finally conducted some numerical experiments with random linear systems to illustrate the fine empirical performance of the algorithm compared to other classical algorithms. Experiments on realistic situations are now needed to truly validate the practical performance of the Hard Thresholding Pursuit algorithm.

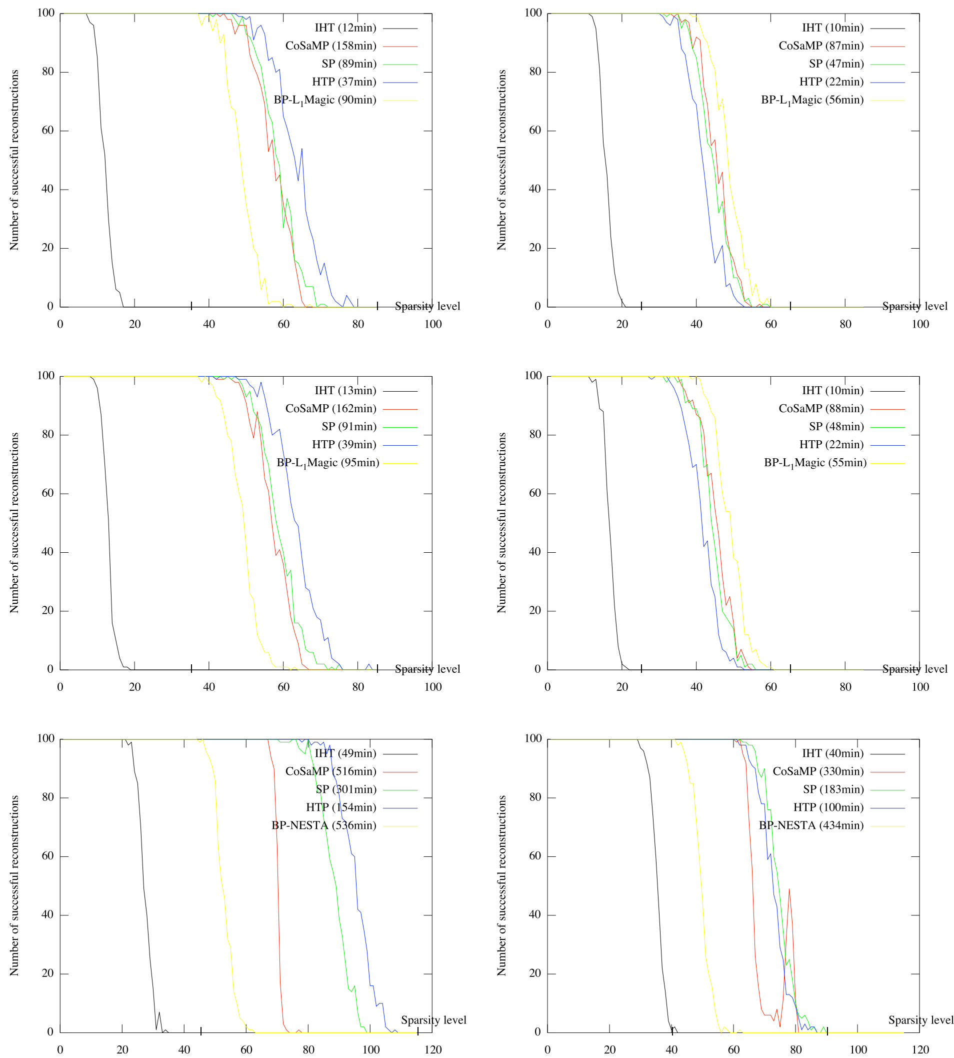

FIG. 4.4. *Comparison of different algorithms (top: Gauss, middle: Bernoulli, bottom: Fourier) with Gaussian (left) and Bernoulli (right) sparse vectors*

Acknowledgments. The author wishes to thank Albert Cohen and Rémi Gribonval for a discussion on an early version of this manuscript, and more generally for the opportunity to work on the project ECHANGE. Some preliminary results of this article were presented during a workshop in Oberwolfach, and the author wishes to thank Rachel Ward for calling the preprint [14] to his attention. This preprint considered an algorithm similar to HTP$^\mu$ with $\mu = 1/(1+\delta_{2s})$, which was called Hybrid Iterative Thresholding. Despite the drawback that a prior knowledge of δ_{2s} is needed, theoretical results analog to Theorems 3.5 and 3.8 were established under the more demanding conditions $\delta_{2s} < 1/3$ and $\delta_{2s} < \sqrt{5} - 2$, respectively.

REFERENCES

[1] R. BERINDE, *Advances in sparse signal recovery methods*, Masters thesis, 2009.
[2] T. BLUMENSATH AND M.E. DAVIES, *Iterative Thresholding for Sparse Approximations*, J. Fourier Anal. Appl., 14 (2008), pp.629–654.
[3] T. BLUMENSATH AND M.E. DAVIES, *Iterative hard thresholding for compressed sensing*, Appl. Comput. Harmon. Anal., 27 (2009), pp.265–274.
[4] T. BLUMENSATH AND M.E. DAVIES, *Normalized iterative hard thresholding: guaranteed stability and performance*, IEEE Journal of selected topics in signal processing, 4 (2010), pp.298–309.
[5] T.T. CAI, L. WANG, AND G. XU, *Shifting inequality and recovery of sparse signals*, IEEE Transactions on Signal Processing, 58 (2010), pp.1300–1308.
[6] E. CANDÈS AND T. TAO, *Decoding by linear programing*, IEEE Trans. Inf. Theory, 51 (2005), pp.4203–4215.
[7] A. COHEN, W. DAHMEN, AND R.A. DEVORE, *Compressed sensing and best k-term approximation*, J. Amer. Math. Soc., 22 (2009), pp.211–231.
[8] W. DAI AND O. MILENKOVIC, *Subspace pursuit for compressive sensing signal reconstruction*, IEEE Trans. Inform. Theory, 55 (2009), pp.2230–2249.
[9] D. DONOHO AND J. TANNER, *Precise undersampling theorems*, Proceedings of the IEEE, 98 (2010), pp.913–924.
[10] S. FOUCART, *A note on guaranteed sparse recovery via ℓ_1-minimization*, Applied and Comput. Harmonic Analysis, 29 (2010), pp.97–103.
[11] S. FOUCART, *Sparse Recovery Algorithms: Sufficient conditions in terms of restricted isometry constants*, Preprint.
[12] R. GARG AND R. KHANDEKAR, *Gradient descent with sparsification: An iterative algorithm for sparse recovery with restricted isometry property*, in Proceedings Of The 26th International Conference on Machine Learning, L. Bottou and M. Littman, eds., pp.337–344.
[13] A. GILBERT, M. STRAUSS, J. TROPP, AND R. VERSHYNIN, *One sketch for all: Fast algorithms for compressed sensing*, in Proceedings Of The 39th ACM Symp. Theory Of Computing, San Diego, 2007.
[14] S. HUANG AND J. ZHU *Hybrid iterative thresholding for sparse signal recovery from incomplete measurements*, Preprint.
[15] S.G MALLAT AND Z. ZHANG, *Matching pursuits with time-frequency dictionaries*, IEEE Trans. Signal Process., 41 (1993), pp.3397–3415.
[16] D. NEEDELL AND J.A. TROPP, *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*, Appl. Comput. Harmon. Anal., 26 (2009), pp.301–32.