Accelerated Iterative Hard Thresholding

Thomas Blumensath
The University of Oxford, Centre for Functional MRI of the Brain, Oxford, OX3 9DU,
tblumens@fmrib.ox.ac.uk.

Abstract

The iterative hard thresholding algorithm (IHT) is a powerful and versatile algorithm for compressed sensing and other sparse inverse problems. The standard IHT implementation faces several challenges when applied to practical problems. The step-size and sparsity parameters have to be chosen appropriately and, as IHT is based on a gradient descend strategy, convergence is only linear. Whilst the choice of the step-size can be done adaptively as suggested previously, this letter studies the use of acceleration methods to improve convergence speed. Based on recent suggestions in the literature, we show that a host of acceleration methods are also applicable to IHT. Importantly, we show that these modifications not only significantly increase the observed speed of the method, but also satisfy the same strong performance guarantees enjoyed by the original IHT method.

Index Terms

Compressed Sensing, Iterative Hard Thresholding.

1. Introduction

Compressed Sensing (CS) [1] [2] is a sub-Nyquist sampling strategy in which a sparse or approximately sparse signal $\mathbf{x} \in \mathbb{R}^N$ is sampled with a linear sampling operator $\mathbf{\Phi}$. The samples $\mathbf{y} \in \mathbb{R}^M$ are potentially corrupted by observation noise $\mathbf{e} \in \mathbb{R}^M$, so that

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{e}. \tag{1}$$

In CS $M << N$, so that we need to exploit the sparsity of $\mathbf{x}$ to be able to recover $\mathbf{x}$ given only $\mathbf{y}$ and $\mathbf{\Phi}$.

Conceptually, we would want to find the sparsest estimate $\widehat{\mathbf{x}}$, that is a vector $\mathbf{x}$ in which only a small number of elements are non-zero, such that $\|\mathbf{y} - \mathbf{\Phi}\widehat{\mathbf{x}}\|_2$ is smaller than some tolerance. Unfortunately,

T. Blumensath acknowledges support of his previous position from the School of Mathematics at the University of Southampton.

due to the combinatorial nature of the sparsity constraint, this is an NP hard computational problem. Instead, CS reconstruction is typically solved using either a convex relaxation of the recovery problem [1] such as

$$\min_{\widehat{\mathbf{x}}} \|\widehat{\mathbf{x}}\|_1 \ : \ \|\mathbf{y} - \mathbf{\Phi}\widehat{\mathbf{x}}\|_2 \leq \epsilon, \tag{2}$$

or a greedy algorithm such as the Compressive Sampling Matching Pursuit (CoSaMP) [3] or the Iterative Hard Thresholding (IHT) algorithm [4], [9].

The IHT algorithm is an iterative method

$$\mathbf{x}^{n+1} = H_k(\mathbf{x}^n + \mu\mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n)), \tag{3}$$

where H_k is the hard thresholding operator that sets all but the k largest (in magnitude) elements[1] in a vector to zero.

IHT is a very simple algorithm and yet, it can be shown that, under certain conditions, IHT can recover sparse and approximately sparse vectors with near optimal accuracy [4]. However, in practice, there are two issues with this simple scheme. 1) The step-size μ has to be chosen appropriately to avoid instability of the method and 2) IHT has only a linear rate of convergence.

Recently, several approaches have been proposed to address these issues ([5], [6], [7] [8]). In [10], a normalised IHT (NIHT) algorithm was suggested in which μ is choosen adaptively in each iteration. This was shown to guarantee the stability of NIHT. In [10], the step-size is set to

$$\mu = \frac{\|\mathbf{\Phi}_{\Gamma^n}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n)\|_2^2}{\|\mathbf{\Phi}_{\Gamma^n}\mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n)\|_2^2} \tag{4}$$

in each iteration, where Γ^n is the support set of $\mathbf{x}^n$. Whilst this is sufficient to guarantee convergence under certain RIP conditions [10], if these conditions fail, then an additional line search was proposed in [10] to guarantee stability. A similar approach was suggested in [8], where again μ is calculated as in (4), but this time, the set Γ is the union of the support of $\mathbf{x}^n$ and the support of $H_k(\mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n))$, which again guarantees stability and performance under RIP conditions. Maleki [6] suggested the use of a projection onto the selected subspace within the IHT framework, whilst Qiu and Dogandzic [5] proposed the Expectation-Conditional Maximisation Either (ECME) algorithm, which replaces the IHT step-size with the pseudo-inverse of $\mathbf{\Phi}$.

$$\widetilde{\mathbf{x}}^{n+1} = H_k(\mathbf{x}^n + \mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n)). \tag{5}$$

[1] In case the k largest elements are not defined uniquely, H_k is allowed to choose from the offending elements in an arbitrary way. For example, it can use a respecified ordering or random selection.

This is guaranteed to converge, as the use of the inverse matrix $(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}$ guarantees stability, thus circumventing the need to tune μ. Importantly, as pointed out in [5], if $(\mathbf{\Phi}\mathbf{\Phi}^T)$ is the identity matrix (that is, if the rows of $\mathbf{\Phi}$ are orthonormal), then the ECME algorithm is identical to the IHT algorithm with $\mu = 1$. Thus, if $\mathbf{\Phi}$ has orthonormal rows, then the IHT algorithm with $\mu = 1$ is guaranteed to be stable (that is, the automatic step-size selection step in IHT is not required in this case). However, if $(\mathbf{\Phi}\mathbf{\Phi}^T)$ is not diagonal, then the ECME algorithm requires the pre-computation and storage of the inverse matrix $(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}$, which might not be feasible for certain large scale problems. More importantly, for many problems, $\mathbf{\Phi}$ itself is structured so that $\mathbf{\Phi}$ and $\mathbf{\Phi}^T$ can be applied using fast algorithms, whilst no such fast methods are guaranteed to exist for $\mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}$. The NIHT algorithm therefore remains an important alternative.

Qiu and Dogandzic further suggested a double over-relaxation scheme [5] to address the convergence speed issue. After calculating an update $\widetilde{\mathbf{x}}$ as in (5), $\widetilde{\mathbf{x}}$ is combined with the two previous estimates $\mathbf{x}^n$ and $\mathbf{x}^{n-1}$ to reduce a specific cost function. The new estimate is then again thresholded. If this newly thresholded estimate has a lower cost than $\widetilde{\mathbf{x}}^{n+1}$ itself, then this new estimate is accepted, whilst $\widetilde{\mathbf{x}}^{n+1}$ is used otherwise. This double relaxation approach (abbreviated DORE) led to a significant improvement in the convergence speed of the method as compared to the IHT algorithm.

Furthermore, Qiu and Dogandzic [5] provided a performance bound for sparse recovery under a '$2k$-sparse subspace quotient condition'[2]

$$\rho_{2k} = \min_{\mathbf{x}:\|\mathbf{x}\|_0 \leq 2k} \frac{\|\mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}\mathbf{\Phi}\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} > 0.5. \tag{6}$$

Inspired by these results and related work in [7] and [8], this letter studies the use of similar acceleration schemes in IHT. Our main contribution is to analyse the accelerated IHT algorithms based on the Restricted Isometry Property commonly used in CS theory. Importantly, we can show that the accelerated IHT algorithms have exactly the same strong, near optimal recovery results enjoyed by standard IHT. This result is a direct generalisation of a similar result by Foucart derived in [7].

2. ACCELERATION OF IHT

The accelerated IHT algorithm (AIHT) is any method that calculates an initial update

$$\widetilde{\mathbf{x}}^{n+1} = H_k(\mathbf{x}^n + \mu\mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{x}^n)), \tag{7}$$

[2] Here and throughout, $\|\mathbf{x}\|_0$ denotes the number of non-zero elements in the vector $\mathbf{x}$.

where μ can either be fixed or chosen adaptively using (4). However, instead of continuing the iterative process with $\widetilde{\mathbf{x}}^{n+1}$, following the same reasoning as in [5], we suggest the use of a strategy that tries to find an estimate $\mathbf{x}^{n+1}$, that satisfies two conditions:

1) $\mathbf{x}^{n+1}$ is k-sparse,
2) $\mathbf{x}^{n+1}$ satisfies $\|\mathbf{y} - \mathbf{\Phi}\mathbf{x}^{n+1}\|_2 \leq \|\mathbf{y} - \mathbf{\Phi}\widetilde{\mathbf{x}}^{n+1}\|_2$.

Any algorithm that calculates such an estimate will be called an accelerated IHT algorithm[3].

One can use different approaches to update $\widetilde{\mathbf{x}}^{n+1}$. These can be roughly split into two categories, methods that only update the non-zero elements in $\widetilde{\mathbf{x}}^{n+1}$ and methods that are allowed to update all elements of $\widetilde{\mathbf{x}}^{n+1}$ but which use a second thresholding step to guarantee the new estimate is k-sparse.

The first type of approach is conceptually the simplest. For example, assume the set of non-zero elements in $\widetilde{\mathbf{x}}^{n+1}$ is $\widetilde{\Gamma}$. If $\mathbf{\Phi}_{\widetilde{\Gamma}}$ is the matrix with the columns not in the set $\widetilde{\Gamma}$ removed and if $\widetilde{\mathbf{x}}_{\widetilde{\Gamma}}^{n+1}$ is defined similarly, then all we need to do is to optimise the cost function $\|\mathbf{y} - \mathbf{\Phi}_{\widetilde{\Gamma}}\widetilde{\mathbf{x}}\|_2^2$. This approach has first been proposed and analysed by Foucart in [7]. This optimisation can be done for example with a gradient (as in [7]) or conjugate gradient algorithm, which when initialised with $\widetilde{\mathbf{x}}_{\widetilde{\Gamma}}^{n+1}$, will always produce estimates that satisfy the condition 2) above. Importantly, in practice, it is advisable to only run a small number of gradient or conjugate gradient steps in each IHT iteration so not to spend too much time in optimising the cost function in the inner loop (see below).

The double-overrelaxation approach of [5] falls into the second category of approaches and uses two relaxation steps

$$\widetilde{\mathbf{x}}_1^{n+1} = \widetilde{\mathbf{x}}^{n+1} + a_1(\widetilde{\mathbf{x}}^{n+1} - \mathbf{x}^n), \tag{8}$$

$$\widetilde{\mathbf{x}}_2^{n+1} = \widetilde{\mathbf{x}}_1^{n+1} + a_2(\widetilde{\mathbf{x}}_1^{n+1} - \mathbf{x}^{n-1}), \tag{9}$$

where, for the AIHT algorithm, the line search parameters a_1 and a_2 can be calculated in closed form to minimise the quadratic cost function $\|\mathbf{y} - \mathbf{\Phi}\widetilde{\mathbf{x}}_1^{n+1}\|_2^2$ and $\|\mathbf{y} - \mathbf{\Phi}\widetilde{\mathbf{x}}_2^{n+1}\|_2^2$ respectively. With this approach, $\widetilde{\mathbf{x}}_2^{n+1}$ is no longer guaranteed to be k-sparse, so that the optimisation step needs to be followed by an additional thresholding step, which in turn is likely to increase the quadratic cost. It can thus happen that $\|\mathbf{y} - \mathbf{\Phi}H_k(\widetilde{\mathbf{x}}_2^{n+1})\|_2^2 > \|\mathbf{y} - \mathbf{\Phi}\widetilde{\mathbf{x}}^{n+1}\|_2^2$, which would violate our second condition. Thus, if $\|\mathbf{y} - \mathbf{\Phi}H_k(\widetilde{\mathbf{x}}_2^{n+1})\|_2^2 > \|\mathbf{y} - \mathbf{\Phi}\widetilde{\mathbf{x}}^{n+1}\|_2^2$, we set $\mathbf{x}^{n+1} = \widetilde{\mathbf{x}}^{n+1}$ (and thus waste the computations of the relaxation step) whilst we use $\mathbf{x}^{n+1} = H_k(\widetilde{\mathbf{x}}_2^{n+1})$ otherwise.

[3] Note however that, just because an algorithm satisfies the two conditions does not necessary mean it will automatically be faster than traditional IHT. Nevertheless, we use the term accelerated IHT here for this general class of methods to highlight the fact that we are in general primarily interested in algorithms of this class that demonstrate clear computational advantages.

3. Theoretical analysis of AIHT

The advantage of AIHT methods is that, as long as each estimate $\mathbf{x}^{n+1}$ satisfies the two conditions given above, then AIHT has the same recovery guarantees as IHT itself. In CS, these guarantees are typically stated in terms of the Restricted Isometry Property (RIP). For a given matrix $\mathbf{\Phi}$, the Restricted Isometry Constants (RIC) of order $2k$ are the largest α_{2k} and smallest β_{2k}, such that

$$\alpha_{2k}\|\mathbf{x}_1+\mathbf{x}_2\|_2^2 \leq \|\mathbf{\Phi}(\mathbf{x}_1+\mathbf{x}_2)\|_2^2 \leq \beta_{2k}\|\mathbf{x}_1+\mathbf{x}_2\|_2^2 \tag{10}$$

holds for all k sparse vectors $\mathbf{x}_1$ and $\mathbf{x}_2$.

AIHT satisfies the following performance bound that states that, as long as $\mathbf{\Phi}$ has RICs that are not too different, then AIHT can recover any signal $\mathbf{x}$ to near optimal accuracy.

Theorem 1: For arbitrary $\mathbf{x}$, given $\mathbf{y}=\mathbf{\Phi}\mathbf{x}+\mathbf{e}$ where $\mathbf{\Phi}$ satisfies the RIP with $\beta_{2k} \leq \mu^{-1} < 1.5\alpha_{2k}$, after

$$n^\star = \left\lceil 2\frac{\log(\|\widetilde{\mathbf{e}}\|_2/\|\mathbf{x}_k\|_2)}{\log(2/(\mu\alpha_{2k})-2)}\right\rceil \tag{11}$$

iterations[4], the AIHT algorithm calculates a solution $\mathbf{x}^{n^\star}$ satisfying

$$\|\mathbf{x}-\mathbf{x}^{n^\star}\|_2 \leq (1+c\sqrt{\beta_{2k}})\|\mathbf{x}-\mathbf{x}_k\|_2 + c\sqrt{\beta_{2k}}\frac{\|\mathbf{x}-\mathbf{x}_k\|_1}{\sqrt{k}} + c\|\mathbf{e}\|_2. \tag{12}$$

where $c \leq \sqrt{\frac{4}{3\alpha_{2k}-2\mu}}+1$, $\widetilde{\mathbf{e}} = \mathbf{\Phi}(\mathbf{x}-\mathbf{x}_k)+\mathbf{e}$ and $\mathbf{x}_k$ is the best k-term approximation to $\mathbf{x}$.
Note that the above theorem holds for both, fixed and variable step sizes, as long as the condition on μ remains satisfied. In situations in which the constants α_{2k} and β_{2k} are unknown, the step-size selection in (4) can be used. For this setting, theoretical results can be derived mirroring the approaches in [10] and [8].

Proof: The proof is an extension of the proof in [11] and establishes an upper bound on $\|\mathbf{x}-\mathbf{x}^{n+1}\|_2$. We here only summarise the main steps, concentrating on those areas that differ from [11]. As in [11], we have

$$\|\mathbf{x}-\mathbf{x}^{n+1}\|_2 \leq \|\mathbf{x}_k-\mathbf{x}\|_2 + \sqrt{\frac{2}{\alpha_{2k}}(\|\mathbf{y}-\mathbf{\Phi}\mathbf{x}^{n+1}\|_2^2+\|\widetilde{\mathbf{e}}\|_2^2)}, \tag{13}$$

where $\widetilde{\mathbf{e}} = \mathbf{\Phi}(\mathbf{x}-\mathbf{x}_k)+\mathbf{e}$.

The proof of [11] is modified by realising that, by the second condition of the acceleration scheme, any AIHT algorithm satisfies

$$\|\mathbf{y}-\mathbf{\Phi}\mathbf{x}^{n+1}\|_2^2 \leq \|\mathbf{y}-\mathbf{\Phi}\widetilde{\mathbf{x}}^{n+1}\|_2^2. \tag{14}$$

[4] Note that for $\|\widetilde{\mathbf{e}}\|_2 < \|\mathbf{x}_k\|_2$, then $\log(\|\widetilde{\mathbf{e}}\|_2/\|\mathbf{x}_k\|_2)$ is negative and, for our condition of μ and α_{2k}, so is $\log(2/(\mu\alpha_{2k})-2)$. Therefore, the algorithms requires a positive number of iterations, unless $\|\widetilde{\mathbf{e}}\|_2 > \|\mathbf{x}_k\|_2$, in which case the the error bound is already fulfilled trivially by $\mathbf{x}^0=\mathbf{0}$, so that no iterations are required.

It is thus sufficient to bound $\|\mathbf{y}-\mathbf{\Phi}\widetilde{\mathbf{x}}^{n+1}\|_2^2$, which can be done as follows (where $\mathbf{g}=2\mathbf{\Phi}^T(\mathbf{y}-\mathbf{\Phi}\mathbf{x}^n)$).

$$\begin{aligned}
& \|\mathbf{y}-\mathbf{\Phi}\widetilde{\mathbf{x}}^{n+1}\|_2^2-\|\mathbf{y}-\mathbf{\Phi}\mathbf{x}^n\|_2^2 \\
= \;& -\langle(\widetilde{\mathbf{x}}^{n+1}-\mathbf{x}^n),\mathbf{g}\rangle+\|\mathbf{\Phi}(\widetilde{\mathbf{x}}^{n+1}-\mathbf{x}^n)\|_2^2 \\
\leq \;& -\frac{2}{\mu}\langle(\widetilde{\mathbf{x}}^{n+1}-\mathbf{x}^n),\frac{\mu}{2}\mathbf{g}\rangle+\frac{1}{\mu}\|\widetilde{\mathbf{x}}^{n+1}-\mathbf{x}^n\|_2^2 \\
= \;& \frac{1}{\mu}\left[\|\widetilde{\mathbf{x}}^{n+1}-\mathbf{x}^n-\frac{\mu}{2}\mathbf{g}\|_2^2-\|\frac{\mu}{2}\mathbf{g}\|_2^2\right] \\
= \;& \frac{1}{\mu}\left[\inf_{\mathbf{x}:\|\mathbf{x}\|_0\leq k}\|\mathbf{x}-\mathbf{x}^n-\frac{\mu}{2}\mathbf{g}\|_2^2-\|\frac{\mu}{2}\mathbf{g}\|_2^2\right] \\
= \;& \inf_{\mathbf{x}:\|\mathbf{x}\|_0\leq k}\left[-\langle(\mathbf{x}-\mathbf{x}^n),\mathbf{g}\rangle+\frac{1}{\mu}\|\mathbf{x}-\mathbf{x}^n\|_2^2\right] \\
\leq \;& -\langle(\mathbf{x}_k-\mathbf{x}^n),\mathbf{g}\rangle+\frac{1}{\mu}\|\mathbf{x}_k-\mathbf{x}^n\|_2^2 \\
= \;& -2\langle(\mathbf{x}_k-\mathbf{x}^n),\mathbf{\Phi}^T(\mathbf{y}-\mathbf{\Phi}\mathbf{x}^n)\rangle+\alpha\|\mathbf{x}_k-\mathbf{x}^n\|_2^2+(\frac{1}{\mu}-\alpha)\|\mathbf{x}_k-\mathbf{x}^n\|_2^2 \\
\leq \;& -2\langle(\mathbf{x}_k-\mathbf{x}^n),\mathbf{\Phi}^T(\mathbf{y}-\mathbf{\Phi}\mathbf{x}^n)\rangle+\|\mathbf{\Phi}(\mathbf{x}_k-\mathbf{x}^n)\|_2^2+(\frac{1}{\mu}-\alpha)\|\mathbf{x}_k-\mathbf{x}^n\|_2^2 \\
= \;& \|\mathbf{y}-\mathbf{\Phi}\mathbf{x}_k\|_2^2-\|\mathbf{y}-\mathbf{\Phi}\mathbf{x}^n\|_2^2+(\frac{1}{\mu}-\alpha)\|\mathbf{x}_k-\mathbf{x}^n\|_2^2 \\
= \;& \|\widetilde{\mathbf{e}}\|_2^2-\|\mathbf{y}-\mathbf{\Phi}\mathbf{x}^n\|_2^2+(\frac{1}{\mu}-\alpha)\|\mathbf{x}_k-\mathbf{x}^n\|_2^2.
\end{aligned}$$

The inequalities are due to (from top to bottom) 1) the RIP condition and the choice of $\beta\leq\frac{1}{\mu}$, 2) the fact that $\mathbf{x}_k$ is k-sparse and 3) the RIP condition again. The third equality is due to the definition of $\widetilde{\mathbf{x}}^{n+1}=H_k(\mathbf{x}^n+\frac{\mu}{2}\mathbf{g})$.

Thus, wrapping up as in [11], we get the bound

$$\begin{aligned}
\|\mathbf{x}-\mathbf{x}^{n+1}\|_2^2 \;\leq\;& \sqrt{\left(\frac{2}{\mu\alpha_{2k}}-2\right)\|(\mathbf{x}_k-\mathbf{x}^n)\|_2^2+\frac{4}{\alpha_{2k}}\|\widetilde{\mathbf{e}}\|_2^2} \\
& +\|\mathbf{x}_k-\mathbf{x}\|_2 \qquad (15)
\end{aligned}$$

Therefore, the condition $2(\frac{1}{\mu\alpha_{2k}}-1)<1$ implies that

$$\|\mathbf{x}-\mathbf{x}^n\|_2\leq\left(\frac{2}{\mu\alpha_{2k}}-2\right)^{n/2}\|\mathbf{x}_k\|_2+\sqrt{\frac{4}{3\alpha_{2k}-2/\mu}}\|\widetilde{\mathbf{e}}\|_2+\|\mathbf{x}_k-\mathbf{x}\|_2,$$

so that the theorem follows using Lemma 6.1 in [3]. ■

4. Numerical Simulations

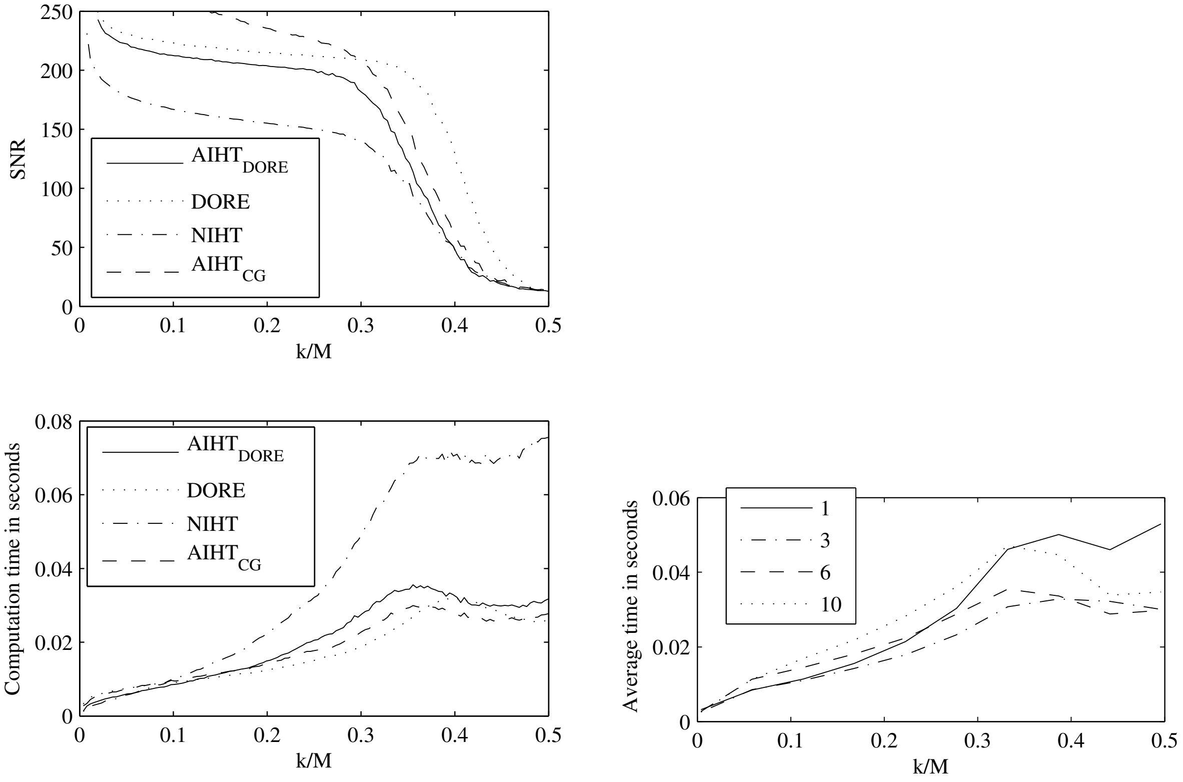
Three experiments were conducted in which two accelerated IHT approaches were compared. In the first approach three conjugate gradient steps were used per inner iteration (AIHT$_{CG}$), whilst the other approach used the double-over-relaxation method of [5] (AIHT$_{DORE}$). The normalised IHT algorithm (NIHT) and the ECME algorithm with the double-over-relaxation (DORE) as proposed in [5] were also used. Both the AIHT as well as the IHT methods used the automatic step-size selection approach which we slightly modified here to reduce the number of line searches. In each iteration, the current proposed step-size was compared to the previously used step-size and the smaller of the two was used. We also relaxed the line search criterion in [10] so that a line search was only initialised when the proposed step-size $\mu > 1.5(\|\mathbf{x}^{n+1} - \mathbf{x}^n\|_2^2)/(\|\mathbf{\Phi}(\mathbf{x}^{n+1} - \mathbf{x}^n)\|_2^2)$. For the ECME algorithm, the matrix inverse was precomputed, the cost of which was *not* counted toward the computation time shown (All simulations were run on a 8GB 2.8 GHz Intel Core i7 Macbook Pro computer.). All algorithms were stopped once $\|\mathbf{x}^{n+1} - \mathbf{x}^n\|_2^2/N < 10^{-9}$. The code for the simulations is available on the author's webpage.

In the first experiment, random matrices $\mathbf{\Phi} \in \mathbb{R}^{256\times512}$ were created with i.i.d. normal entries followed by normalisation of the columns of $\mathbf{\Phi}$. For each sparsity k in the interval from 1 to 128, 1000 matrices were generated and k-sparse vectors $\mathbf{x}$ were drawn with the k non-zero entries also drawn from the unit variance normal distribution. No noise was added. Figure 1(a) shows the average Signal to Noise Ratio (SNR) (top panel) and the average computation time in seconds (lower panel) for each sparsity level k/M. It is clear that both acceleration methods work well with IHT. Both significantly improve the convergence speed of the method, however, ECME is still somewhat faster in this example and also works somewhat better in terms of signal recovery when $k/M \approx 0.35$.

Figure 1(b), which shows the average computation time for the above experiment when AIHT uses 1, 3, 5 and 10 conjugate gradient steps, demonstrates that it is advisable to only use a small number of such steps.

To show a problem instance in which ECME (DORE) has a clear disadvantage, figure 2 shows the computation times for a larger problem where $\mathbf{\Phi}$ was a sparse matrix. Here $\mathbf{\Phi} \in \mathbb{R}^{2560\times5120}$ had only 52 non-zero elements in each row so that multiplication by $\mathbf{\Phi}$ and $\mathbf{\Phi}^T$ can be done fast, whilst multiplication by the dense matrix $(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}$ as used in ECME is costly. The non-zero entries were drawn from an i.i.d. normal and the columns were then normalised to unit length. Results are averaged over 10 problem instances and $k \in \{250, 500, 750, 1000\}$. All other parameters were the same as in the first experiment. Note again that the computation time shown here does not include the time required to calculate the matrix inverse, which here took about 20 seconds on average, which is two orders of magnitude larger than the computation time required by for the IHT type methods!

The third example used the Shepp-Logan image of size 512×512 (see figure 2), where between 50 to 70 radial slices were sampled from the 2D-Fourier transform of the image which were then used as the

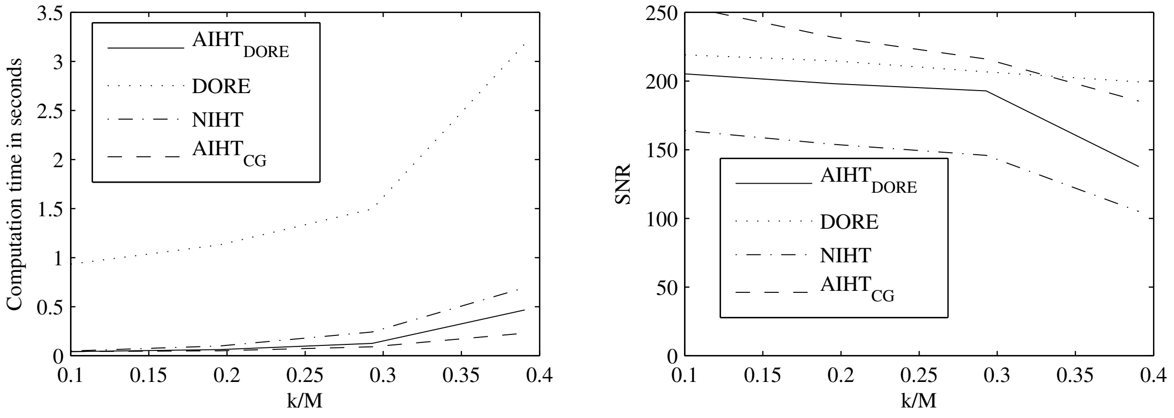


(a) (b)

Fig. 1. Panel (a) shows average SNR ($20 \log_{10} \|\mathbf{x}\|_2 / \|\mathbf{x} - \hat{\mathbf{x}}\|_2$) and computation time (seconds) for the four algorithms using $\mathbf{\Phi} \in \mathbb{R}^{256 \times 512}$ with i.i.d. normal entries and normalised rows with $\mathbf{x}$ only k non-zero entries drawn form a unit variance normal distribution. Panel (b) shows a comparison of average computation times for AIHT with 1, 3, 6 and 10 conjugate gradient steps.

measurements $\mathbf{y}$. The image was assumed to be k sparse in the Haar Wavelet domain with $k = 3769$. The algorithms were run with the same parameters as before but stopped once $\|\mathbf{x}^{n+1} - \mathbf{x}^n\|_2^2 / N < 10^{-16}$.

Figure 3, which also gives the results obtained by back-projection, shows the Peak Signal to Noise Ratio (PSNR) for each estimate as well as the computation time. NIHT is seen to be significantly slower than the other approaches. In contrast, using three iterations of a conjugate gradient solver per iteration to accelerate the NIHT algorithm not only significantly reduces the computation time but also lead to significantly better PSNR values. The DORE algorithm, which in this example does not have to use matrix inversion due to the orthogonality of the observation matrix (and is thus identical to our AIHT$_{DORE}$ method), shows comparable performance.



(a) (b)

Fig. 2. Average computation time (in seconds) (a) and SNR (b) for the four algorithms using $\mathbf{\Phi} \in \mathbb{R}^{2560 \times 5120}$, where each row of $\mathbf{\Phi}$ has only 52 non-zero entries (at random locations and with i.i.d. Gaussian values) and normalised rows. $\mathbf{x}$ had $k \in \{250, 500, 750, 1000\}$ non-zero entries drawn form a unit variance normal distribution.

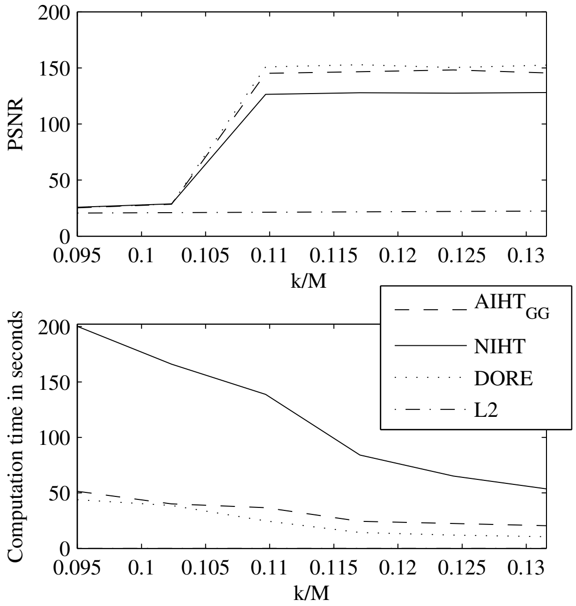


Fig. 3. Reconstruction accuracy and computation time for the 512×512 Shepp-Logan phantom image which was sampled taking between 50 to 70 equally spaced radial slices from the 2D Fourier transform of the image and reconstructed assuming sparsity in the Haar wavelet domain. Shown are the PSNR ($20 \log_{10} \|\mathbf{x}\|_\infty / \|\mathbf{x} - \hat{\mathbf{x}}\|_2$) and the computation time in seconds for different ratios of sparsity (k) to number of observations (M).

5. Discussion and conclusion

The Iterative Hard Thresholding algorithms is a simple yet powerful tool to reconstruct sparse signals. Not only does it give near optimal recovery guarantees under the RIP, it is also very versatile and can be easily adapted to a range of constraint sets [11] as well as to non-linear measurement systems [12]. Inspired by the recently developed ECME algorithm, we have here introduced and analysed an accelerated IHT framework. We have in particular looked at two acceleration strategies, the use of a conjugate gradient method and the use of the double-over-relaxation approach of [5], though other approaches can equally well be slotted into the AIHT algorithm. Our main contribution was to show that, if done correctly, then any accelerated IHT algorithm inherits the strong recovery bounds from the IHT algorithm. Furthermore, combining these acceleration methods with NIHT significantly increased the algorithm's empirical convergence speed, making the accelerated NIHT algorithm a strong competitor to the ECME method. Importantly, the accelerated NIHT method is extremely simple to implement and does not require the computation, storage and repeated use of matrix inverses. This is an advantage in many compressed sensing applications where the measurement matrix is often sparse or based on fast transforms such as the wavelet and Fourier transforms.

REFERENCES

[1] E. Candès and J. Romberg, "Practical signal recovery from random projections," in *Proc. SPIE Conf., Wavelet Applications in Signal and Image Processing XI*, Jan. 2005.

[2] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[3] Needell D, Tropp JA. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis. 2008 May;26(3):301–321.

[4] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3,. pp. 265–274 2009.

[5] K. Qiu and A. Dogandzic, "ECME Thresholding Methods for Sparse Signal Reconstruction," *arXiv*, no. 1004.4880v3.

[6] A. Maleki "Coherence Analysis of Iterative Thresholding Algorithms," in *Proc. of the 47th Annual Allerton Conference on Communication, Control, and Computing*, pp. 236–241, 2009

[7] S. Foucart "Hard thresholding pursuit: an algorithm for compressive sensing," submitted

[8] V. Cevher "On Accelerated Hard Thresholding Methods for Sparse Approximation," *EPFL Tec Rep.* , 2011.

[9] T. Blumensath and M.E. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, 2008.

[10] T. Blumensath and M.E. Davies, "Normalised Iterative Hard Thresholding; guaranteed stability and performance," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 298–309, 2010.

[11] T. Blumensath, "Sampling and reconstructing signals from a union of linear subspaces," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4660–4671, 2011.

[12] T. Blumensath, "Compressed Sensing with Nonlinear Observations," submitted, 2011.