The Journal of Fourier Analysis and Applications

Iterative Thresholding for Sparse Approximations

Thomas Blumensath and Mike E. Davies

Communicated by ...

ABSTRACT. Sparse signal expansions are signal expansions that represent or approximate a signal using a small number of elements from a large collection of elementary waveforms. Finding the optimum sparse expansion is known to be NP hard in general and non-optimal strategies such as Matching Pursuit, Orthogonal Matching Pursuit, Basis Pursuit and Basis Pursuit De-noising are often called upon. These methods show good performance in practical situations, however, they often do not operate on the cost functions that are at the heart of the problem. In this paper we study two iterative algorithms that are minimising the cost functions of interest. Furthermore, these strategies have a comparable computational cost per iteration to a single Matching Pursuit iteration, making the methods applicable to many real world problems. However, the non-convexity of the optimisation problem means that these strategies are only guaranteed to find local solutions and good initialisation becomes paramount. To guarantee good performance, we study two approaches. The first approach uses the proposed algorithms to refine the solutions found with other methods, replacing the typically used conjugate gradient solver. The second strategy adapts the algorithms and we show that this adaptation can be used to achieve results that lie between those obtained with Matching Pursuit and those found with Orthogonal Matching Pursuit, while retaining the computational complexity of the Matching Pursuit algorithm. Numerical studies demonstrate the performance of the two approaches.

1. Introduction

Sparse signal approximations have over the last decade gained in popularity in mathematics, statistics and engineering. For example, a wide range of engineering applications such as source coding [1], [2], denoising [3], source

Math Subject Classifications. 65D15, 41A45
Keywords and Phrases. Sparse Approximations, Iterative Thresholding, L_0 Regularisation, Subset Selection.

ISSN 1069-5869 DOI: ITA-JFAA - April 3, 2007

separation [4] and pattern analysis [5] have benefited from progress made in this area. A sparse signal approximation is a solution to the linear equation:

$$\mathbf{x} = \sum_i \phi_i y_i + \mathbf{e},$$

where $\mathbf{x}$ is the signal of interest and $\{\phi_i\}$ a set of elements, commonly called the dictionary. In this paper we are dealing primarily with sparse approximations as opposed to sparse representations, i.e. we allow for a non-zero error $\mathbf{e}$ in the above signal model. In this paper we restrict our attention to finite length vectors[1] $\mathbf{x}$. We therefore introduce the matrix Φ and vector $\mathbf{y}$ such that $\Phi\mathbf{y} = \sum_i \phi_i y_i$, i.e. the vectors ϕ_i are the column vectors in Φ. In general, the dimension of $\mathbf{y}$ is allowed to be larger than that of $\mathbf{x}$.

1.1 Useful Dictionary Properties

Before proceeding, we introduce some useful concepts and properties of dictionaries, which will help us in the development below. The *spark* of a dictionary is defined in [6] as

Definition 1 spark***.*** *spark*(Φ) is the size of the smallest subset of columns of Φ such that the elements in this subset are linearly dependent.

For example if all M dimensional subsets of column vectors from Φ are linearly independent, but there exist a subset of size $M+1$ in which the columns are dependent, then the spark of Φ is $M+1$.

We also need the following definitions:

Definition 2 Cumulative Coherence. The Cumulative Coherence or Babel function [7] is defined as:

$$\mu_1(m) := \max_{|\Gamma|=m} \max_{\omega \notin \Gamma} \sum_{\gamma \in \Gamma} |\langle \phi_\omega, \phi_\gamma \rangle|. \tag{1.1}$$

A useful bound on the cumulative coherence is given in terms of the coherence:

Definition 3 Coherence.

$$\mu_1(1) = \mu_0 := \max_{\omega \neq \gamma} |\langle \phi_\omega, \phi_\gamma \rangle|. \tag{1.2}$$

A bound on the Cumulative Coherence is then:

$$\mu_1(m) \leq m\mu_0. \tag{1.3}$$

[1] Many of the results derived in this paper are valid in a more general Hilbert space and hold for more general operators, however, definitions 1 to 3 below and the results base on them require a more restrictive setting.

See for example [7]. Note, that contrary to common practice, in the above definitions and throughout this paper we do not assume that $\|\phi_i\|_2 = 1$. However, on occasion we use the weaker condition that $\|\phi_i\|_2 = c$ for all i.

Finally we introduce:

Definition 4 maxcor(x). $\operatorname{maxcor}(x) = \frac{\|\Phi^H \mathbf{x}\|_\infty}{\|\mathbf{x}\|_2}$.

1.2 Problem Formulation

In this paper we look at two incarnations of the sparse approximation problem. The first we call the L_0 regularised problem. This problem is defined as a solution to the following optimisation problem. For a given $\mathbf{x}$ and Φ the L_0 regularised sparse approximation problem is to find coefficient vector $\mathbf{y}$ minimising the cost function:

$$C_{L_0}(\mathbf{y}) = \|\mathbf{x} - \Phi\mathbf{y}\|_2^2 + \lambda\|\mathbf{y}\|_0, \tag{1.4}$$

where $\|\mathbf{y}\|_0$ is defined as $|\Gamma_1(\mathbf{y})|$ and throughout this paper we use $\Gamma_1(\mathbf{y}) = \{y_i : y_1 \neq 0\}$ as the set of non-zero coefficients. $|\Gamma_1(\mathbf{y})|$ is the size of this set so that $\|\mathbf{y}\|_0$ counts the number of non-zero coefficients.

The other problem will be called the M-sparse problem, this is a constraint optimisation problem of the form:

$$\min_{\mathbf{y}} \|\mathbf{x} - \Phi\mathbf{y}\|_2^2 \text{ subject to } \|\mathbf{y}\|_0 \leq M, \tag{1.5}$$

i.e now we constrain the number of non-zero coefficients to be below a certain value[2].

The problem of minimising equation (1.5) for general $\mathbf{x}$ and Φ is known to be an NP-hard optimisation problem [8, 9]. Therefore, two common themes have been adopted to approximately solve the problem, greedy optimisation strategies and relaxation of the cost function. Greedy strategies, such as Matching Pursuit (MP) type algorithms [10], are iterative procedures, which are often relatively fast and which have therefore been used extensively in practical applications. The performance of these methods is however not guaranteed in general and only under very strict conditions can they be shown to optimise the above cost function [11] [12]. Relaxation methods replace the $\|\mathbf{y}\|_0$ constraint by an almost everywhere differentiable and often convex cost function, such as the FOCUSS algorithm [13] or the

[2] It is worth pointing out that the two problems are related in that there exist a λ, which depends on $\mathbf{x}$ and M, such that the solution to the L_0 regularised problem is the same as that to the M-sparse problem. However, it is also important to realise that the algorithms derived here can have quite different performance, even though they have a very similar structure.

Basis Pursuit De-noising method [14]. These often offer better performance, but are computationally demanding.

Basis Pursuit De-noising relaxes the L_0 penalty and replaces it with the convex L_1 penalty. This leads to the convex optimisation problem:

$$\min_{\mathbf{y}} \|\mathbf{x} - \Phi\mathbf{y}\|_2^2 + \lambda\|\mathbf{y}\|_1. \tag{1.6}$$

Recently, iterative thresholding algorithms have been proposed to solve this problem as in [15], [16], [17] and [18]. A similar algorithm to directly solve the L_0 regularised optimisation problem had previously been put forward by Kingsbury in [19] and more recently, a slight variation of this was used in [20].

1.3 Paper Overview

In section 2 we derive the algorithm used in [20] using ideas from [15]. Importantly, we present the following novel results with regard to this algorithm.

- We give a simple condition guaranteeing the convergence of the method to a local optimum of the cost function 1.4.
- We give conditions specifying the fixed points of the algorithm.
- We show that the algorithm is guaranteed not to increase the cost function 1.4.
- We give bounds on the error and sparsity of the solution.
- We analyse the convergence speed of the method.

In section 3 a novel variation of the algorithm is derived to solve the M-sparse problem. We show the following properties of this algorithm.

- We give a simple condition guaranteeing the convergence of the method to a local optimum of the cost function 1.5.
- We give conditions specifying the fixed points of the algorithm.
- We show that the algorithm is guaranteed not to increase the cost function 1.5.
- We give bounds on the error and sparsity of the solution.
- We analyse the convergence speed of the method.

These two algorithms work directly on the cost functions 1.4 and 1.5. As these functions are non-convex, we find that the algorithms only converge to local optima. Even worse, we find that the fixed points of the first algorithm do not have any guarantee of sparsity, in fact, adding any element from the null-space of Φ to any fixed point will give another fixed point. Numerical studies also confirm, initialisation of the method is important and when initialised with zero vectors, the algorithms were found to perform worse than for example the Matching Pursuit algorithm. We therefore suggest two strategies for a successful application of the methods. The first strategy is to use the methods in conjunction with other methods such as Matching Pursuit or Basis Pursuit De-noising. The solutions found with these algorithms are in general not even local optimal solutions to the problems 1.4 and 1.5. In fact, the solutions can always be improved by orthogonally projecting the signal onto the space of the non-zero components chosen with these methods. This projection is typically done using a conjugate gradient algorithm. By replacing the conjugate gradient algorithm with the methods proposed here one does not only calculate such a projection, more importantly, the proposed algorithms can also change the support set of the solution, while at the same time guaranteeing to improve the solution. This is shown numerically in subsections 4.1 and 4.2.

The other suggested approach is a slight modification of the algorithms. This method varies the number of retained coefficients in each iteration, starting with a single coefficient and adding additional coefficients as the algorithm progresses. If the number of non-zero coefficients is increased in each iteration, the performance is comparable to Matching pursuit, whilst by increasing the number of non-zero coefficients more slowly allows us to improve the performance. This approach is studied in subsection 4.3.

This paper can be read on three different levels, the casual reader, interested in the algorithms and their properties, but less interested in more formal statements of these properties nor in the exact derivation of the algorithms can read the digest subsections given at the beginning of the next two sections. More formal derivations of the algorithms and statements of the main theorems and lemmata comprise the rest of the next two sections. The keen reader, interested also in the proofs, is referred to the appendices.

2. An iterative algorithm for the L_0 regularised problem

2.1 Digest: the Iterative Hard-Thresholding Algorithm

To solve the optimisation problem:

$$\min_{\mathbf{y}} \|\mathbf{x} - \Phi\mathbf{y}\|_2^2 + \lambda\|\mathbf{y}\|_0, \tag{2.1}$$

we derive the following iterative algorithm:

$$\mathbf{y}^{n+1} = H_{\lambda^{0.5}}(\mathbf{y}^n + \Phi^H(\mathbf{x} - \Phi\mathbf{y}^n)), \tag{2.2}$$

where $H_{\lambda^{0.5}}$ is the element wise hard thresholding operation:

$$H_{\lambda^{0.5}}(y_i) \begin{cases} 0 & \text{if } |y_i| \leq \lambda^{0.5} \\ y_i & \text{if } |y_i| > \lambda^{0.5}. \end{cases} \tag{2.3}$$

This algorithm will be called the *iterative hard-thresholding algorithm.* We show that under the assumption that $\|\Phi\|_2 < 1$ the algorithm is guaranteed not to increase equation (1.4) and in fact converges to a local minimum of equation (1.4). Furthermore, the asymptotic convergence rate is linear and at any fixed point $\mathbf{y}^\star$ the error satisfies the bound:

$$\|x - \Phi\mathbf{y}^\star\|_2 \leq \lambda^{0.5}. \tag{2.4}$$

2.2 Optimisation Transfer

Instead of optimising equation (1.4), let us introduce a surrogate objective function, as proposed in [21]:

$$C^S_{L_0}(\mathbf{y},\mathbf{z}) = \|\mathbf{x} - \Phi\mathbf{y}\|_2^2 + \lambda\|\mathbf{y}\|_0 - \|\Phi\mathbf{y} - \Phi\mathbf{z}\|_2^2 + \|\mathbf{y} - \mathbf{z}\|_2^2. \tag{2.5}$$

Note that $C_{L_0}(\mathbf{y}) = C^S_{L_0}(\mathbf{y},\mathbf{y})$. Equation (2.5) can be rewritten as:

$$C^S L_0(\mathbf{y},\mathbf{z}) = \sum_i [y_i^2 - 2y_i(z_i + \phi_i^H\mathbf{x} - \phi_i^H\Phi\mathbf{z}) + \lambda|y_i|^0] \\ + \|\mathbf{x}\|_2^2 + \|\mathbf{z}\|_2^2 - \|\Phi\mathbf{z}\|_2^2,$$

where $|y_i|^0$ is one if $y_i \neq 0$ and zero otherwise. Now the y_i are decoupled. Therefore, the minimum of equation (2.5) can be calculated by minimising with respect to each y_i individually. To derive the minimum, we distinguish two cases, $y_i = 0$ and $y_i \neq 0$. In the first case, the element wise cost is (ignoring the constant terms) λ. In the second case the cost is (again ignoring the constant terms):

$$y_i^2 - 2y_i(z_i + \phi_i^H\mathbf{x} - \phi_i^H\Phi\mathbf{z}),$$

the minimum of which is achieved at

$$y_i^\star = z_i + \phi_i^H\mathbf{x} - \phi_i^H\Phi\mathbf{z}.$$

Comparing the cost for both cases (i.e $y_i = 0$ and $y_i = z_i + \phi_i^H\mathbf{x} - \phi_i^H\Phi\mathbf{z}$) we see that the minimum of equation (2.5) is attained at:

$$\mathbf{y} = H_{\lambda^{0.5}}(\mathbf{z} + \Phi^H(\mathbf{x} - \Phi\mathbf{z})),$$

where we use the element-wise hard thresholding operator given above in equation (2.3). Note that the minimum need not be unique whenever $z_i + \phi_i^H\mathbf{x} - \phi_i^H\Phi\mathbf{z} = \lambda^{0.5}$. However, using a strict inequality in the definition of the thresholding operator as done here guarantees a unique update.

The iterative hard thresholding algorithm is now defined as:

$$\mathbf{y}^{n+1} = H_{\lambda^{0.5}}(\mathbf{y}^n + \Phi^H(\mathbf{x} - \Phi\mathbf{y}^n)). \tag{2.6}$$

In the rest of this section we will often simplify the notation and introduce the non-linear operator $T\mathbf{y} = H_\theta(\mathbf{y} + \Phi^H(\mathbf{x} - \Phi\mathbf{y}))$. This is a thresholded version of the well known Landweber iteration [22]. This is the same algorithm suggested in [20], however, we found that this algorithm is not stable in general. In this paper we show that a sufficient requirement for the above algorithm to converge is that the eigenvalues of the linear operator $(\mathbf{I} - \Phi^H\Phi)$ are $0 < eig(\mathbf{I} - \Phi^H\Phi) \leq 1$. Using the singular value decomposition of $\Phi = USV^H$ we can write $(\mathbf{I} - \Phi^H\Phi) = (\mathbf{I} - VS^HSV^H) = (V(I - S^HS)V^H)$, so that we can express the above requirement as a restriction on the singular values $\sigma(\Phi)$ of Φ, i.e. $\sigma(\Phi) < 1$.

2.3 Relationship Between Optimisation of the Surrogate Function and the Original Cost Function

In this subsection we give an important lemma:

Lemma 1. *Let $\mathbf{y}^{n+1} = H_\theta(\mathbf{y}^n + \Phi^H(\mathbf{x} - \Phi\mathbf{y}^n))$. The sequences $(C_{L_0}(\mathbf{y}^n))_n$ and $(C^S_{L_0}(\mathbf{y}^{n+1}, \mathbf{y}^n))_n$ are non-increasing.*

The proof of this lemma can be found in appendix A.

This lemma states that the cost function 1.4 does not increase from iteration to iteration, or, more bluntly, using the algorithm cannot lead to worse results than not using the algorithm.

2.4 Specifying the Fixed Points

As mentioned above, the algorithm does not have a unique solution for different initialisations. It is therefore important to specify conditions on the fixed points of the algorithm.

Lemma 2. *Let ϕ_i^H be the i^{th} row of Φ^H and define the sets $\Gamma_0 = \{i : y_i^\star = 0\}$ and $\Gamma_1 = \{i : y_i^\star > \lambda^{0.5}\}$. Then at a fixed point of algorithm 2.2, i.e at points such that $\mathbf{y}^\star = T(\mathbf{y}^\star)$ we have*

$$|\phi_i^H(\mathbf{x} - \Phi\mathbf{y}^\star)| \begin{cases} = 0 & \text{if } i \in \Gamma_1 \\ \leq \lambda^{0.5} & \text{if } i \in \Gamma_0. \end{cases}$$

The proof of this result is straight forward and we give it here, as it has some merit in itself.

Proof. A fixed point is any $\mathbf{y}^\star$ such that $\mathbf{y}^\star = T(\mathbf{y}^\star)$. Looking at this equality element wise and inserting the algorithm we have:

$$y_i^\star = H_{\lambda^{0.5}}(y_i^\star + \phi_i^H(\mathbf{x} - \Phi\mathbf{y}^\star)),$$

If $y_i^\star = 0$, then $|\phi_i^H(\mathbf{x} - \Phi\mathbf{y}^\star)| \leq \lambda^{0.5}$. Similarly for $i \in \Gamma_1$ we have :

$$y_i^\star = y_i^\star + \phi_i^H(\mathbf{x} - \Phi\mathbf{y}^\star),$$

where we have dropped the thresholding operator, as $y_i^\star \neq 0$. □

One of the main results in this section relates the fixed points with the cost function 1.4:

Lemma 3. *A fixed point* $\mathbf{y}^\star = T\mathbf{y}^\star$ *is a local minimum of equation (1.4).*

The proof of this lemma is a bit more involved and can be found in appendix B. By a local minima we mean here that perturbing $\mathbf{y}^\star$ by an infinitesimal small amount (in any direction) will not decrease the cost function.

Before proceeding we answer another important question, whether the set of fixed points includes the optimal solution. To show that this is in fact true we appeal to theorem 12 in [7], which we enhance here by adding an additional property.

Theorem 1 (Tropp [7]: Theorem 12). *For an input signal* $\mathbf{x}$ *and a threshold* $\lambda^{0.5}$*, denote by* $\mathbf{y}^{opt}$ *the global minimum of the optimisation problem 1.4. Define* $\Gamma_0 = \{\gamma : y_\gamma^{opt} = 0\}$ *and* $\Gamma_1 = \{\gamma : y_\gamma^{opt} \neq 0\}$ *Then:*

- $\forall \gamma \in \Gamma_1,\ |y_\gamma^{opt}| \geq \lambda^{0.5}$
- $\forall \gamma \in \Gamma_0,\ \phi_\gamma^H(\mathbf{x} - \Phi\mathbf{y}^{opt}) \leq \lambda^{0.5}$
- $\forall \gamma \in \Gamma_1,\ \phi_\gamma^H(\mathbf{x} - \Phi\mathbf{y}^{opt}) = 0,$

The third condition implies that the error is orthogonal to the atoms $\phi_i^H{}_\gamma$ when $\gamma \in \Gamma_1$. In other words, the signal is projected orthogonally onto the space spanned by these atoms. This condition is not given in [7] but is easily verified. For any fixed subset of atoms, the cost function 1.4 is minimised by orthogonal projection onto these atoms. A formal proof is omitted here.

Comparing the sufficient conditions in theorem 1 to the fixed point conditions of the algorithm in lemma 2 we have:

Theorem 2. *The optimal solution to the optimisation problem 1.4 belongs to the fixed points of the iterative algorithm defined by equation (2.2).*

2.5 Convergence

We have shown in the previous section that the iterative hard thresholding algorithm is guaranteed not to increase the cost function in equation (1.4). In this subsection we state an even more important property of the algorithm, namely, the algorithm converges to a local minimum of equation (1.4).

More formally, we have the following theorem:

Theorem 3. *Assume* $\mathbf{y} \in \mathcal{H}$*, where* $\mathcal{H}$ *is a Hilbert space. If* $C_{L_0}(\mathbf{y}^0) < \infty$ *and if the eigenvalues of the operator* $(\mathbf{I} - \Phi^H\Phi)$ *obey* $0 < eig(\mathbf{I} - \Phi^H\Phi) \leq 1$*, then the sequence* $(\mathbf{y}^n)_n$ *defined by the iterative procedure in equation (2.2)* *converges to a local minimum of equation (1.4).*

Note that the condition $C_{L_0}(\mathbf{y}^0) < \infty$, which we use in lemma C.1 is only really of importance in infinite dimensional spaces, where it implies that only a finite number of y_i^0 are non-zero. The proof of the above theorem is given in appendix C.

2.6 Bounds on error and sparsity

Lemma 2 is a necessary and sufficient condition and the fixed points of the algorithm do only depend on the inner product between the reconstruction error and the dictionary elements. Adding any element from the null-space of Φ to $\mathbf{y}^\star$ will not change the reconstruction error. The overcompletness of Φ implies the existence of a null-space. Elements of the null-space are in general not sparse and therefore, the algorithm does not guarantee a sparse solution.

Note also that the algorithm is guaranteed to find a local minima of the cost function. This implies that the local minima of the cost function are not required to be sparse either.

Clearly, having no guarantee on the sparsity of the solution is not in general a desirable property of an algorithm to find sparse representations. One would therefore want to modify the algorithm to impose constraints on the number of non-zero coefficients of the solution. Such an approach is studied in the next section, however, in the rest of this section we will present a bound on the reconstruction error achieved with the proposed method.

A bound on the approximation error is stated in

Lemma 4. *Assume that $\|\phi_i\|_2 \leq 1$, then a tight bound for the approximation error $\|x - \Phi\mathbf{y}^\star\|_2$ achieved at a fixed point $\mathbf{y}^\star$ is*

$$\|x - \Phi\mathbf{y}^\star\|_2 \leq \lambda^{0.5}. \tag{2.7}$$

The proof can be found in appendix D. Note that the condition on the norm of the columns of Φ is typically satisfied if we scale a dictionary with unit norm columns such that it fulfils the condition $\|\Phi\|_2 < 1$ required for the convergence of the iterative method.

2.7 Speed of convergence

The convergence proof of the algorithm relied on the fact that after a finite number of iterations, the algorithm does not change the selected subset anymore. Then, the algorithm simplifies to the standard Landweber iteration [22]. Therefore the asymptotic convergence speed is the linear convergence of the Landweber algorithm [22] given by:

$$\|\mathbf{y}^n - \mathbf{y}^m\|_2 \leq \|1 - \Phi_{\Gamma_1}^H \Phi_{\Gamma_1}\|_2^{(n-m)} \|\mathbf{y}^m\|_2 \tag{2.8}$$

Note that we have expressed this result in terms of the sub-matrices Φ_{Γ_1}. Assuming that $\|\phi_i\|_2 = c$ for all i and that $c - \mu_1(M-1) > 0$, where M is the size of the set Γ_1, we can use results form [11] to bound the eigenvalues of $1 - \Phi_{\Gamma_1}^H \Phi_{\Gamma_1}$ with the cumulative coherence, leading to

$$\|\mathbf{y}^n - \mathbf{y}^m\|_2 \leq [1 - (c - \mu_1(M-1))]^{\frac{n-m}{2}} \|\mathbf{y}^m\|_2. \tag{2.9}$$

Because the cumulative coherence is an increasing function of M, it can be seen that the bound decreases, the smaller the selected sub-dictionary.

3. An iterative algorithm for the M-sparse problem

3.1 Digest: the M-Sparse Algorithm

In this section we turn to the M-sparse problem:

$$\min_{\mathbf{y}} \|\mathbf{x} - \Phi\mathbf{y}\|_2^2 \text{ subject to } \|\mathbf{y}\|_0 \leq M, \tag{3.1}$$

and derive the following iterative algorithm:

$$\mathbf{y}^{n+1} = H_M(\mathbf{y}^n + \Phi^H(\mathbf{x} - \Phi\mathbf{y}^n)), \tag{3.2}$$

where H_M is now a non-linear operator that only retains the M coefficients with the largest absolute magnitude:

$$H_M(y_i) \begin{cases} 0 & \text{if } |y_i| < \lambda_M^{0.5}(\mathbf{y}) \\ y_i & \text{if } |y_i| \geq \lambda_M^{0.5}(\mathbf{y}). \end{cases} \tag{3.3}$$

The threshold $\lambda_M^{0.5}(\mathbf{y})$ is set to the M^{th} largest absolute value of $\mathbf{y}^n + \Phi^H(\mathbf{x} - \Phi\mathbf{y}^n)$, if less than M values are non-zero we define $\lambda_M^{0.5}(\mathbf{y})$ to be the smallest absolute value of the non-zero coefficients. We call this algorithm the *M-sparse algorithm.*

In addition to the requirement that $\|\Phi\|_2 < 1$ we now also use the assumption that $M <$*spark*(Φ). Under these two assumptions the algorithm is guaranteed not to increase equation (1.5) and also converges to a local minimum of equation (1.5). As before, the asymptotic convergence rate is linear and if $\|\phi_i\|_2 = c$, for all i, then at the fixed point $\mathbf{y}^\star$ the error satisfies the bound:

$$\|x - \Phi\mathbf{y}^\star\|_2 \leq \frac{\|x\|_2 \text{ maxcor}(x)}{c^2 - \mu_1(M-1)}. \tag{3.4}$$

3.2 Optimisation Transfer

We again use optimisation transfer to derive the iterative algorithm. The surrogate objective function is now:

$$C_M^S(\mathbf{y},\mathbf{z}) = \|\mathbf{x}-\Phi\mathbf{y}\|_2^2 - \|\Phi\mathbf{y}-\Phi\mathbf{z}\|_2^2 + \|\mathbf{y}-\mathbf{z}\|_2^2. \tag{3.5}$$

Note that we do not use a regularisation term here, however, in the minimisation of the surrogate cost function we now require that the constraint $\|\mathbf{y}\|_0 \leq M$ is satisfied. We again have $C_M(\mathbf{y}) = C_M^S(\mathbf{y},\mathbf{y})$. As in the previous section we write equation (3.5) as:

$$C_M^S(\mathbf{y},\mathbf{z}) \propto \sum_i [y_i^2 - 2y_i(z_i + \phi_i^H\mathbf{x} - \phi_i^H\Phi\mathbf{z})].$$

This again de-couples the y_i. If we ignore the constraint on the number of non-zero coefficients we would get the standard Landweber minimum of:

$$y_i^\star = z_i + \phi_i^H\mathbf{x} - \phi_i^H\Phi\mathbf{z}.$$

At this minimum, the cost function would be

$$C_M^S(\mathbf{y}^\star,\mathbf{z}) \propto \sum_i [y_i^{\star 2} - 2y_i^\star(z_i + \phi_i^H\mathbf{x} - \phi_i^H\Phi\mathbf{z})] = \sum_i -y_i^{\star 2}.$$

The *constraint* minimum of the surrogate cost function is then acheived by chosing the M largest (in absolute value) coefficients $y_i^\star$.

The minimum of equation (2.5) is then attained at:

$$\mathbf{y} = H_M(\mathbf{z} + \Phi^H(\mathbf{x}-\Phi\mathbf{z})),$$

where now the thresholding operator H_M choses the threshold depending on its argument.

The iterative M-sparse algorithm is then:

$$\mathbf{y}^{n+1} = H_M(\mathbf{y}^n + \Phi^H(\mathbf{x}-\Phi\mathbf{y}^n)). \tag{3.6}$$

Again we show that a sufficient requirement for the above algorithm to converge is that the eigenvalues of the linear operator $(\mathbf{I}-\Phi^H\Phi)$ are $0 < eig(\mathbf{I}-\Phi^H\Phi) \leq 1$ or that the singular values $\sigma(\Phi)$ of Φ, obey $\sigma(\Phi) < 1$.

3.3 Relationship Between Optimisation of the Surrogate Function and the Original Cost Function

Again we can show that the algorithm reduces the cost function $C_M(\mathbf{y}^n)$. It is important to stress that we here require $\|\mathbf{y}^0\|_0 \leq M$, which is guaranteed by the fact that we choose only the largest M coefficients in each iteration.

(In cases where there are more than one coefficient with equal magnitude, such that the M largest coefficients are not uniquely defined, we assume that the algorithm selects randomly from the offending coefficients.) The equivalent to lemma 1 also holds:

Lemma 5. *Let* $\mathbf{y}^{n+1} = H_M(\mathbf{y}^n + \Phi^H(\mathbf{x} - \Phi\mathbf{y}^n))$*. The sequences* $(C_M(\mathbf{y}^n))_n$ *and* $(C_M^S(\mathbf{y}^{n+1}, \mathbf{y}^n))_n$ *are non-increasing.*

The proof to this lemma is exactly the same as that for lemma 1 given in appendix A, with the cost functions chosen appropriately.

3.4 Specifying the Fixed Points

It is again important to analyse the fixed points of this algorithm. This again follows a similar approach taken above. Note, however, we have now a slightly different lemma. First, we have one extra condition, i.e. we have the restriction $\|\mathbf{y}\|_0 \leq M < N$, where N is the dimension of $\mathbf{x}$ and secondly, the threshold in the condition is not fixed anymore, but depends on the smallest non-zero element in $\mathbf{y}^\star$

Lemma 6. *Let* ϕ_i^H *be the* i^{th} *row of* Φ^H *and define the sets* $\Gamma_0 = \{i : y_i^\star = 0\}$ *and* $\Gamma_1 = \{i : y_i^\star > \lambda^{0.5}\}$*. Then at a fixed point of algorithm (3.2), i.e at points such that* $\mathbf{y}^\star = T(\mathbf{y}^\star)$ *we have*

$$|\phi_i^H(\mathbf{x} - \Phi\mathbf{y}^\star)| \begin{cases} = 0 & \text{if } i \in \Gamma_1 \\ \leq \lambda_M^{0.5}(\mathbf{y}^\star) & \text{if } i \in \Gamma_0. \end{cases}$$

and $\|y\|_0 \leq M$*. We can actually say something more, in fact* $\|\mathbf{y}^\star\|_0 = M$ *unless* $\mathbf{x} - \Phi\mathbf{y}^\star = 0$*, in which case* $C_M(\mathbf{y}^\star) = 0$*.*

Again, these are necessary and sufficient conditions.

Proof. A fixed point is defined as $\mathbf{y}^\star = T(\mathbf{y}^\star)$ and after insertion of the algorithm we get:

$$y_i^\star = H_M(y_i^\star + \phi_i^H(\mathbf{x} - \Phi\mathbf{y}^\star)),$$

$\|y\|_0 \leq M$ follows necessarily from the enforcement of the constraint, however, the threshold $\lambda_M^{0.5}(\mathbf{y})$ in the algorithm now depends on $\phi_i^H(\mathbf{x} - \Phi\mathbf{y}^\star)$. For $y_i^\star = 0$ we require $|\phi_i^H(\mathbf{x} - \Phi\mathbf{y}^\star)| < \lambda_M^{0.5}(\mathbf{y})$ whilst for $i \in \Gamma_1$ we have:

$$y_i^\star = y_i^\star + \phi_i^H(\mathbf{x} - \Phi\mathbf{y}^\star).$$

Also, assume that $C_M(\mathbf{y}^\star) \neq 0$ and that $\|\mathbf{y}^\star\|_0 = M - 1$. Then $\mathbf{x} - \Phi\mathbf{y} \neq 0$. Assuming Φ spans the signal space and assuming $M <$*spark*(Φ), then there must be a ϕ_i^H for $i \in \Gamma_0$ such that $|\phi_i^H(\mathbf{x} - \Phi\mathbf{y})| > 0$. However, then also $\lambda_M^{0.5}(\mathbf{y}) > 0$ and the algorithm would have chosen M terms. The proof for $\|\mathbf{y}^\star\|_0 < M - 1$ follows by induction. □

We can further show that:

Lemma 7. *If M <spark(Φ), then a fixed point $\mathbf{y}^\star = T\mathbf{y}^\star$ is a local minimum of the constraint cost function 1.5.*

The proof can be found in appendix E.

3.5 Convergence

We also have a convergence proof for the M-sparse algorithm:

Theorem 4. *Assume $\mathbf{y} \in \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space. If $C_M(\mathbf{y}^0) < \infty$ and if the eigenvalues of the operator $(\mathbf{I} - \Phi^H\Phi)$ obey $0 < eig(\mathbf{I} - \Phi^H\Phi) \leq 1$, then the sequence $(\mathbf{y}^n)_n$ defined by the iterative M-sparse algorithm 3.2 converges to a local minimum of equation (1.5).*

The proof can be found in appendix F.

3.6 Bounds on error and sparsity

The sparsity of the fixed points of the algorithm is naturally given by the parameter M.

A similar bound on the approximation error as given in lemma 4 can be found:

Lemma 8. *Assume that $\|\phi_\gamma\|_2 \leq 1$, then a tight bound for the approximation error $\|x - \Phi\mathbf{y}^\star\|_2$ achieved at a fixed point $\mathbf{y}^\star$ is*

$$\|x - \Phi\mathbf{y}^\star\| \leq \lambda^{0.5}(\mathbf{y}^\star). \tag{3.7}$$

The proof is virtually identical to the proof of lemma 4. The difference is now that $\lambda^{0.5}(\mathbf{y}^\star)$ is a function of the fixed point itself. We therefore need a lemma bounding $\lambda^{0.5}(\mathbf{y}^\star)$. This is done in

Lemma 9. *Assume that for all i, $\|\phi_i\|_2 = c$. If $\mu_1(M-1) < 1$, then we have the following bound:*

$$\lambda^{0.5}(\mathbf{y}^\star) \leq \frac{\|x\|_2 \ \mathrm{maxcor}(x)}{c^2 - \mu_1(M-1)}. \tag{3.8}$$

The proof of this lemma can be found in appendix G.

3.7 Speed of convergence

Again, appealing to the fact that the algorithm will after a certain number of iterations (say m), not change the subset (see proof of theorem 4), we have the same convergence properties as for the algorithm of section 2:

$$\|\mathbf{y}^n - \mathbf{y}^m\|_2 \leq \|1 - \Phi_{\Gamma_1}^H \Phi_{\Gamma_1}\|_2^{(n-m)} \|\mathbf{y}^m\|_2 \tag{3.9}$$

and, under the same conditions as in subsection 2.7

$$\|\mathbf{y}^n - \mathbf{y}^m\|_2 \leq [1 - (c - \mu_1(M-1))]^{\frac{n-m}{2}} \|\mathbf{y}^m\|_2. \tag{3.10}$$

4. Numerical Studies

4.1 Improving the cost function

In this subsection we study the ability of the algorithms to improve on results calculated with Matching Pursuit, which is reviewed in appendix H. We chose this algorithm for comparison as it is relatively fast and therefore used in many applications. We show that the use of the above algorithms in conjunction with Matching Pursuit often leads to an improvement in the results. Note that Matching Pursuit does not give the minimum squared error solution achievable with the selected subset, which is known to be achieved by an orthogonal projection of the signal onto the selected elements. We therefore compare our results here to those found with Matching Pursuit followed by an orthogonal projection. We here do this projection using the pseudo-inverse, however, in most situations it would be more efficient to use conjugate gradient type algorithms.

The results are shown in Figures 1 and 2 for the iterative hard-thresholoding algorithm and the M-sparse algorithm respectively.

The results in Figure 1 were calculated as follows. We randomly generate 1 000 dictionaries of size 128 × 256 with elements distributed uniformly on the unit sphere. From each of these we randomly selected 128 elements. The coefficients were generated by drawing i.i.d. zero mean and unit variance Gaussian variables, however, values with a magnitude below $\lambda^{0.5} = \sqrt{2}\ \mathrm{erf}^{-1}(M/128)$ were set to zero. This threshold ensures that on average, only M of the coefficients were non-zero. We choose $M \in \{2, 11, 20, 29, 38, 47, 56, 65, 74, 83, 92, 101, 110, 119, 128\}$. This procedure ensured that we used the same average number of non-zero coefficients as in the experiment below while ensuring that the coefficients are above $\lambda^{0.5}$ as required by lemma 1. We repeated this procedure four times and added different levels of zero mean Gaussian noise giving a Signal to Noise Ratio (SNR) of 120 dB, 80 dB, 40 dB and 0 dB.

We then run the Matching Pursuit algorithm stopping when the minimum in the cost function $\|\mathbf{x} - \Phi\mathbf{y}\|_2^2 + \lambda\|y\|_0$ was reached. These coefficients were then used to initialise the iterative hard-thresholding algorithm. In Figure 1 the stars are the cost function averages (averaged over the 1 000 dictionaries) for the cost function after projecting the Matching Pursuit results. The squares are the results for the iterative hard-thresholding algorithm. The lower panels show the difference between the results. We here show the cost (as well as the difference in the cost) in dB.

The results in Figure 2 where calculated similarly, the only difference being that the coefficients were generated by directly choosing M elements at random using Gaussian coefficients. This time the coefficient values were not restricted in magnitude. We then run the Matching Pursuit algorithm stopping after M elements had been selected. We used these results to

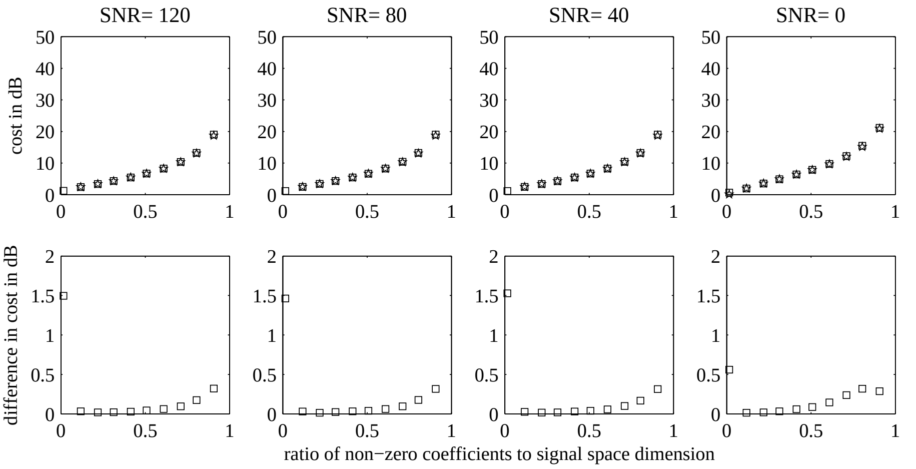

FIGURE 1 The top panels show a comparison between Matching Pursuit followed by orthogonal projection (stars) and additional use of the iterative hard-thresholding algorithm (squares) for different amounts of noise added to the signal. The y-axis shows the cost function $\|\mathbf{x} - \Phi\mathbf{y}\|_2^2 + \lambda\|y\|_0$ expressed in dB and the x-axis shows the ratio of non-zero elements used to generate the signal to the signal dimension. The lower panels show the difference between the cost functions in the upper panels.

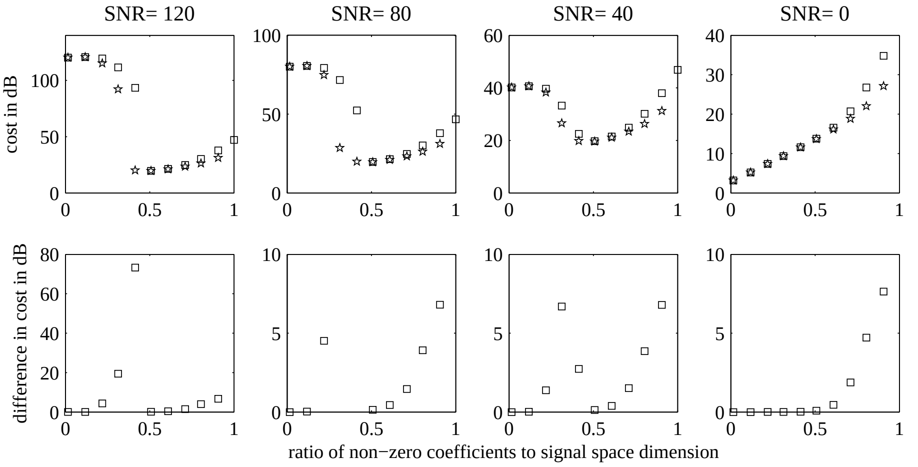

FIGURE 2 The top panels show a comparison between Matching Pursuit followed by orthogonal projection (stars) and additional use of the M-sparse algorithm (squares) for different amounts of noise added to the signal. The y-axis shows the cost function $\|\mathbf{x} - \Phi\mathbf{y}\|_2^2$ expressed in dB and the x-axis shows the ratio of non-zero elements used to generate the signal to the signal dimension. The lower panels show the difference between the cost functions in the upper panels.

initialise the M-sparse algorithm. In Figure 2 the stars are the cost function averages for the projected Matching Pursuit results. The squares are the results for the thresholding algorithm. The lower panels again show the difference between the results. We here show the cost $\|x - \Phi\mathbf{y}\|_2^2$, again expressed in dB.

From these results we can draw the following conclusions:

- When working with the regularised cost function we see that the iterative hard-thresholding algorithm can improve performance only marginally.
- For the M-sparse problem, we see that the M-sparse algorithms can lead to significantly better results than using Matching Pursuit.
- In the low noise case, both Matching pursuit and the M-sparse algorithm often recover exactly the elements used to generate the signal whenever the signal was generated with very few non-zero elements. Importantly, the M-sparse algorithm is able to find the exact representations even in cases in which Matching Pursuit alone fails.
- Apart from the increased ability to find the exact representation, the M-sparse algorithm also improves the SNR by several dB for only mildly sparse representations.

4.2 Exact Recovery

In some applications, such as compressive sampling [17], it is desirable to exactly recover the elements used to construct the observation $\mathbf{x}$. In order to analyse this we conducted experiments similar to those above. We generated M-sparse signals with $M \in \{30, 40, 50, 60\}$ using Gaussian coefficients, however, this time we did not add noise to the signals.

We then repeated the first experiment reported above for a range of $\lambda^{0.5}$ values. The second experiment was also a repetition of the second experiment in the previous subsection, this time using a range of different values M for each algorithm.

We then calculated the number of correctly identified elements (True Positives) and the non-zero elements estimated which were not used to generate the particular observation (False Positives). These quantities are here normalised by dividing them by the number of non-zero elements and the number of zero elements in the original data-generating coefficient vector respectively.

The averaged results for the two algorithms are shown in Figures 3 and 4 respectively. The solid lines are the results from the iterative algorithms proposed in this paper, while the dotted lines are the results found with Matching Pursuit alone. The four different lines in each panel are the results calculated for the signals generated with different numbers of non-zero coefficients, with lines from top left to bottom right having decreasing sparsity.

We can draw the following conclusions:

- Both algorithms improve the ability to correctly identify elements used to generate the signal.

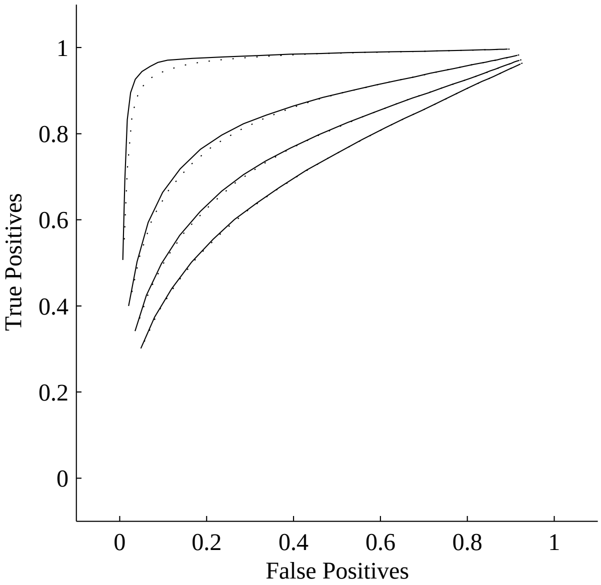

FIGURE 3 Comparison of the iterative hard-thresholding algorithm and matching pursuit in terms of the number of correctly identified elements used to generate the test signal and the number of elements identified not used to generate the test signal. We here show the results for different sparsities, i.e. (from top left to bottom right) $M \in \{30, 40, 50, 60\}$. The solid lines are the results for the iterative hard-thresholding algorithm and the dotted lines are the results for Matching Pursuit.

- For less sparse signals this advantage becomes smaller.

4.3 Stepping through lambda vs. MP

Instead of calculating the pseudo inverse in each iteration in orthogonal Matching Pursuit, an iterative method could be envisaged to approximate the pseudo inverse solution. This idea is similar to a strategy suggested in [23], which can also be used for the Landweber based algorithms of this paper. The M sparse algorithm can be run for different values of M, starting from $M = 1$. After S iterations M is increased by one. If the algorithm is initialised with a zero vector and if S is large enough such that the algorithm converges to the minimum error solution for each M, then this strategy is very similar to orthogonal Matching Pursuit. The difference with orthogonal Matching Pursuit is that the M sparse algorithm does not necessarily use the same subset of elements in each stage.

The other extreme would be to set $S = 1$. Then the algorithm is similar to Matching Pursuit with a similar computational complexity, however, previously selected atoms are now updated in subsequent iterations. Another difference is that if the columns of Φ are not of unit length, then the value of a newly selected atom is not the same for both algorithms.

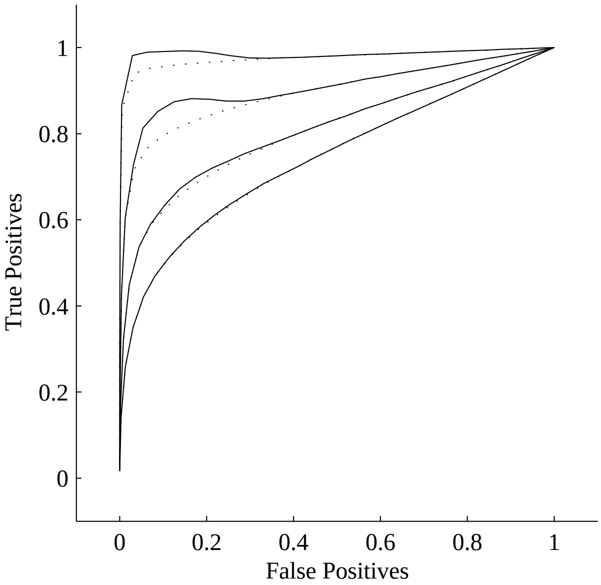

FIGURE 4 Comparison of the M-sparse algorithm and matching pursuit in terms of the number of correctly identified elements used to generate the test signal and the number of elements identified not used to generate the test signal. We here show the results for different sparsities, i.e. (from top left to bottom right) $M \in \{30, 40, 50, 60\}$. The solid lines are the results for the M-sparse algorithm and the dotted lines are the results for Matching Pursuit.

One could also use the iterative hard thresholding algorithm with a threshold depending on the current residual norm. Such a strategy would be similar to the StOMP algorithm proposed in [24], but again with the difference that we would not necessarily calculate the exact orthogonal projection for each threshold and that we allow the set of selected elements to change from iteration to iteration.

To test these idea we generated 1 000 signals $\mathbf{x}$ by randomly choosing 64 elements from Φ generated as above again without added noise. We then averaged the performance of Matching Pursuit, Orthogonal Matching Pursuit and the strategy in which the number of retained elements is increased by one every $S \in \{1, 2, 5, 10, 50\}$ iterations. The approximation error in dB is shown in Figure 5 for approximations with varying number of non-zero coefficients.

The following observations can be made:

- For $S = 1$ the proposed method shows marginally worse performance than Matching Pursuit if less than 30 coefficients are non-zero. If more than 30 coefficients are non-zero, the proposed method outperforms Matching Pursuit.
- Increasing S increases the performance and for $S = 2$ the method outperformes Matching Pursuit if more than 18 elements are non-

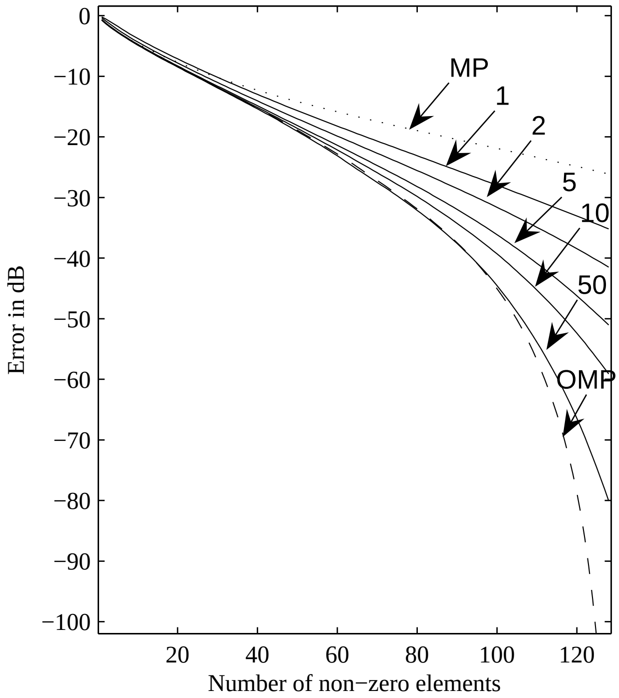

FIGURE 5 Signal to Noise Ratio in dB for different numbers of non-zero elements. The dotted line are the results with the matching pursuit algorithm and the dashed line are the results obtained with orthogonal matching pursuit. The solid lines are the results of the iterative approach in which the number of retained coefficients is increased stepwise any S iterations. The numbers in the figure indicate which S was used for each curve. The results are averaged over 100 randomly generated dictionaries $\Phi \in \mathbb{R}^{128 \times 256}$ and signals with 64 non-zero gaussian coefficients.

zero.

- Using $S = 50$ we see that the proposed algorithm outperformes even orthogonal Matching Pursuit if the number of non-zero values is between 25 and 94. This is a sign that the algorithm is not just an iterative orthogonal Matching Pursuit implementation.

5. Conclusion

In this paper we derived two algorithms that operate directly on the L_0 regularised cost function and the M-sparse constraint cost function respectively. To our knowledge, these are the only algorithms, (apart from exhaustive search) that have this property[3]. We have derived extensive theoretical

[3] Though it is now well known that under certain conditions, other algorithms can also find the optimal solutions to the problems studied here.

results for the methods. These results reveal that the algorithms have multiple fixed points making a straightforward application difficult. However, we here argued for the use of the algorithms in two contexts. Firstly, the algorithms can be used to improve the results calculated with other methods such as Matching Pursuit. In this case we have shown that the algorithms offer benefits that cannot be explained by orthogonal projection onto the selected elements alone, i.e. the methods often also discover better sets of elements to describe the signal. The improved performance was shown to be both, in terms of the cost function of interest as well as in terms of identification of the elements that were used to generate the signal. Secondly, by running the M-sparse algorithm for increasing values of M, we have shown that the method can be used on its own, with performance ranging from the performance of Matching pursuit to that of orthogonal Matching Pursuit and even beating the latter in certain circumstances.

Most importantly, the methods have similar computational requirements as Matching Pursuit. Furthermore, many of the fast computational techniques suggested for Matching Pursuit [25] can be used also for the proposed algorithms. The proposed algorithms are therefore applicable to very large signals and dictionaries so that they can potentially be used in many real-world applications.

A. Proof of lemma 1 and lemma 5

Proof. Define the operator $L = \sqrt{\mathbf{I} - \Phi^H\Phi}$. Then:

$$\begin{aligned}
C(\mathbf{y}^{n+1}) &\leq C(\mathbf{y}^{n+1}) + \|L(\mathbf{y}^{n+1} - \mathbf{y}^n)\|_2^2 \\
&= C^S(\mathbf{y}^{n+1}, \mathbf{y}^n) \\
&\leq C^S(\mathbf{y}^n, \mathbf{y}^n) \\
&= C(\mathbf{y}^n) \\
&\leq C(\mathbf{y}^n) + \|L(\mathbf{y}^n - \mathbf{y}^{n-1})\|_2^2 \\
&= C^S(\mathbf{y}^n, \mathbf{y}^{n-1}),
\end{aligned}$$

where the first equality is the definition of C^S and the second inequality is due the fact the $\mathbf{y}^{n+1}$ is the minimiser of $C^S(\mathbf{y}, \mathbf{y}^n)$. □

B. Proof of lemma 3

Proof. Given a fixed point $\mathbf{y}^\star = T\mathbf{y}^\star$ and any small perturbation $|\partial h_i| < \epsilon$, for some $\epsilon > 0$. We show that $C_{L_0}(\mathbf{y}^\star + \partial h) > C_{L_0}(\mathbf{y}^\star)$. However, we first show $\exists\ \epsilon > 0 : \forall\ \|\partial h\| < \epsilon$ the following inequality holds:

$$C^S_{L_0}(\mathbf{y}^\star + \partial h, \mathbf{y}^\star) \geq C^S_{L_0}(\mathbf{y}^\star, \mathbf{y}^\star) + \|\partial h\|_2^2.$$

$$\begin{aligned} &C^S_{L_0}(\mathbf{y}^\star + \partial h, \mathbf{y}^\star) - C^S_{L_0}(\mathbf{y}^\star, \mathbf{y}^\star) = \\ &\qquad \sum_i (y_i + \partial h_i)^2 - 2(y_i + \partial h_i)y_i - 2(y_i + \partial h_i)(\Phi^H \mathbf{x} - \Phi^H \Phi \mathbf{y}^\star)_i \\ &\qquad\qquad - y_i^2 + 2y_i^2 + 2y_i(\Phi^H \mathbf{x} - \Phi^H \Phi \mathbf{y}^\star)_i - \lambda |y_i|^0 + \lambda |y_i + \partial h_i|^0. \end{aligned}$$

After simplification of the above equation, we split the summation into two parts, one for $\Gamma_0 = \{i : y_i = 0\}$ and one for $\Gamma_1 = \{i : y_i \neq 0\}$. We get:

$$\begin{aligned} &C^S_{L_0}(\mathbf{y}^\star + \partial h, \mathbf{y}^\star) - C^S_{L_0}(\mathbf{y}^\star, \mathbf{y}^\star) = \\ &\qquad \|\partial h\|_2^2 + \sum_{\Gamma_0} \lambda |\partial h_i|^0 - 2\partial h_i(\Phi^H \mathbf{x} - \Phi^H \Phi \mathbf{y}^\star) \\ &\qquad\qquad + \sum_{\Gamma_1} -2\partial h_i(\Phi^H \mathbf{x} - \Phi^H \Phi \mathbf{y}^\star) \end{aligned}$$

For a fixed point $\mathbf{y}^\star$ the last line is zero as stated in lemma 2. For the summation over Γ_0 we have to consider two cases, if $\partial h_i = 0$, then this term is zero. If $\partial h_i \neq 0$, then choosing $|\partial h_i| \leq |\frac{\lambda}{2(\phi_i^H(\mathbf{x} - \Phi \mathbf{y}))}|$ guarantees the non-negativity of this term. Note that we also need the condition that $|\partial h_i| \leq \mathbf{y}_i$ for all $i \in \Gamma_1$ such that $y_i - h_i \neq 0$. This condition is required when splitting the cost function $|y_i - \partial h_i|^0$. Therefore $\exists\ \epsilon : \forall\ \partial h, |\partial h_i| \leq \epsilon, C^S_{L_0}(\mathbf{y}^\star + \partial h, \mathbf{y}^\star) \geq C^S_{L_0}(\mathbf{y}^\star, \mathbf{y}^\star) + \|\partial h\|_2^2$. Using this we get:

$$\begin{aligned} C_{L_0}(\mathbf{y}^\star + \partial h) &= C^S(\mathbf{y}^\star + \partial h, \mathbf{y}^\star) - \|L \partial h\|_2^2 \\ &\geq C^S_{L_0}(\mathbf{y}^\star + \partial h, \mathbf{y}^\star) - \|\partial h\|_2^2 \geq C^S_{L_0}(\mathbf{y}^\star, \mathbf{y}^\star) = C_{L_0}(\mathbf{y}^\star) \end{aligned}$$

□

C. Convergence Proof of the Iterative Hard Thresholding Algorithm

To prove theorem 3 we need one more lemma.

Lemma C.1. $\forall\ \epsilon > 0,\ \exists\ N$ *such that* $\forall\ n > N,\ \|\mathbf{y}^{n+1} - \mathbf{y}^n\|_2^2 \leq \epsilon.$

Proof. We show that $\sum_n^N \|\mathbf{y}^{n+1} - \mathbf{y}^n\|_2^2$ converges, which implies the lemma [26, Theorem 3.23]. This is done by showing that $\sum_n^N \|\mathbf{y}^{n+1} - \mathbf{y}^n\|_2^2$ is monotonically increasing and bounded. We have monotonicity by:

$$\sum_{n=1}^{N-1} \|\mathbf{y}^{n+1} - \mathbf{y}^n\|_2^2 + \|\mathbf{y}^{N+1} - \mathbf{y}^N\|_2^2 \geq \sum_{n=1}^{N-1} \|\mathbf{y}^{n+1} - \mathbf{y}^n\|_2^2.$$

and boundedness follows from:

$$\begin{aligned}\sum_{n=0}^{N}\|\mathbf{y}^{n+1}-\mathbf{y}^{n}\|_2^2 &\leq \frac{1}{c}\sum_{n=0}^{N}\|L(\mathbf{y}^{n+1}-\mathbf{y}^{n})\|_2^2 \qquad \text{(C.1)}\\ &\leq \frac{1}{c}\sum_{n=0}^{N}[C_{L_0}(\mathbf{y}^{n})-C_{L_0}(\mathbf{y}^{n+1})]\\ &= \frac{1}{c}(C_{L_0}(\mathbf{y}^{0})-C_{L_0}(\mathbf{y}^{N+1}))\\ &\leq \frac{1}{c}C_{L_0}(\mathbf{y}^{0}),\end{aligned}$$

$$\text{(C.2)}$$

where c is a lower bound on the spectrum of the linear operator $L^H L$ where we use $L=\sqrt{\mathbf{I}-\Phi^H\Phi}$, which by assumption is strictly greater than zero. $\|L(\mathbf{y}^{n+1}-\mathbf{y}^{n})\|_2^2 \leq C_{L_0}(\mathbf{y}^{n})-C_{L_0}(\mathbf{y}^{n+1})$ (see proof of Lemma 1) is here used to derive the second inequality. □

Proof of theorem 3. In lemma C.1 take $\epsilon<\lambda$. If $y_i^n>\lambda^{0.5}$ and $y_i^{n+1}=0$, then $\|\mathbf{y}^{N+1}-\mathbf{y}^{N}\|_2^2\geq\lambda$, which by lemma C.1 is impossible for $n>N$ for some N. Therefore, for large N, the set of zero and non-zero coefficients will not change and $|y_i^n|>\lambda^{0.5},\forall i\in\Gamma_1,n>N$. For $y_i^n,i\in\Gamma_1$ the algorithm then reduces to the standard Landweber algorithm with guaranteed convergence [22]. Note that the largest (smallest) eigenvalue of $(I-\Phi^H\Phi)$ will not increase (decrease) if we delete columns from Φ ensuring that the eigenvalue constraint required for the Landweber convergence is satisfied.

Also, by lemma 3 the fixed point is a local minimum of equation (1.4).
□

D. Proof of lemma 4

Proof. From lemma 2 we have:

$$\|\Phi^H(x-\Phi\mathbf{y}^\star)\|_\infty\leq\lambda^{0.5}. \qquad \text{(D.1)}$$

Define $\text{maxcor}(x)=\frac{\|\Phi^H\mathbf{x}\|_\infty}{\|\mathbf{x}\|_2}$, then $\text{maxcor}(x-\Phi\mathbf{y}^\star)\|(x-\Phi\mathbf{y}^\star)\|_2=\|\Phi^H(x-\Phi\mathbf{y}^\star)\|_\infty\leq\lambda^{0.5}$. The condition on the l_2 norm of the columns of Φ ensures that $\text{maxcor}(x)\leq 1$, from which the lemma follows. □

E. Proof of lemma 7

Proof. Again $\|\mathbf{y}^\star\|_0\leq M$ due to the constraint. We want to show that:

$$C_M(\mathbf{y}^\star+\partial h)\geq C_M(\mathbf{y}^\star),$$

for any small perturbation $|\partial h_i| < \epsilon$, for some $\epsilon > 0$. If we restric the solution to the support of $\mathbf{y}^\star$ then $\mathbf{y}^\star$ is the minimum squared error solution [22]. Now the support of ∂h might contain elements that are not in the support of $\mathbf{y}$. If $|\mathbf{y}^\star| < M$, then by lemma 6 $C(\mathbf{y}^\star) = 0$, which is obviously the global minimum. Otherwise $|\mathbf{y}^\star| = M$, therefore, if ∂h contains an element not in the support of $\mathbf{y}$ it also needs to include an element within the support of $\mathbf{y}^\star$ with opposite value to that in $\mathbf{y}^\star$, say $y_i^\star$. This value is strictly larger than zero, so we can chose $y^\star > 2_i\epsilon > 0$. Therefore, we can always chose ϵ small enough such that a perturbation of a fixed point with radius smaller than ϵ does not reach any other fixed point satisfying the constraint on $\|\mathbf{y}\|_0$. □

F. Proof of Theorem 4

First note that lemma C.1 also holds for the M-sparse algorithm. We further need:

Lemma F.1. *For all* $\mathbf{y} : \|\mathbf{y}\|_0 \le M < N, \mathbf{y} \neq 0$, *there exists a* $\lambda_M^\star > 0$ *such that* $\lambda_M^\star < \lambda_M(\mathbf{y})$.

Proof. If $\mathbf{x} \neq 0$, and $M \neq 0$, then $\mathbf{y} + \Phi^H(x - \Phi\mathbf{y})$ has at least one non-zero element, for all $\mathbf{y}$ and $\mathbf{x} \neq 0$. If $\mathbf{y} = 0$, then we have $\Phi^H x$, which by the assumption of full rank of Φ^H cannot be zero for $x \neq 0$. Also for $\mathbf{y} \neq 0$, $\mathbf{y} + \Phi^H(x - \Phi\mathbf{y}) = 0$ would imply that $\mathbf{y} = -\Phi^H(x - \Phi\mathbf{y})$. This requires that the null-space of Φ^H, $null(\Phi^H) < K - M$, where K is the dimension of $\mathbf{y}$. But $null(\Phi^H) + rank(\Phi^H) = K$, and $rank(\Phi^H) = N > M$. Therefore $null(\Phi^H) = K - N < K - M$. Because $\mathbf{y} + \Phi^H(x - \Phi\mathbf{y})$ has at least one non-zero element for all $\mathbf{y} : \|\mathbf{y}\|_0 \le M$, $\lambda_M(\mathbf{y})$ cannot be zero. □

Proof of theorem 4. Again it is sufficient to show that the algorithm will converge to a fixed subset. Once the set of non-zero coefficients does not change anymore, the convergence is guaranteed using the Landweber convergence proof in the fixed subspace [22]. By lemma F.1, there exists a $\lambda_M^\star > 0$ such that $\lambda_M^\star < \lambda_M(\mathbf{y})$ for all $\mathbf{y} : \|\mathbf{y}\|_0 \le M, \mathbf{y} \neq 0$. For $M \neq 0$ and $\mathbf{x} \neq 0$ $\mathbf{y} = 0$ is not a fixed point of the algorithm so we can assume $\mathbf{y} \neq 0$. Using $\lambda_M^\star$ we can follow the proof for the hard thresholding algorithm with fixed λ. In lemma C.1 take $\epsilon < \lambda_M^\star$. If $y_i^n > \lambda^{0.5}$ and $y_i^{n+1} = 0$, then $\|\mathbf{y}^{n+1} - \mathbf{y}^n\|_2^2 \ge \lambda_M(\mathbf{y}^N)$, which by lemma C.1 is impossible for $n > N$ for some N. Therefore, for large N, the set of zero and non-zero coefficients will not change and $|y_i^n| > \lambda_M^\star, \forall i \in \Gamma_1, n > N$.

□

G. Proof of lemma 9

Let us first proof

Lemma G.1. *Assume that for all i, $\|\phi_i\|_2 = c$. Let $G = \Phi_\Gamma^H \Phi_\Gamma$ for $|\Gamma| \leq M \leq N$ and suppose that $\mu_1(M-1) < c^2$. Then:*

$$\|G^{-1}\|_\infty \leq \frac{1}{c^2 - \mu_1(M-1)}. \tag{G.1}$$

Proof. Write $\hat{G} = \frac{1}{c^2}G$. Then

$$\begin{aligned} \|G^{-1}\|_\infty = \frac{1}{c^2}\|\hat{G}^{-1}\|_\infty = \frac{1}{c^2}\|\sum_{k=0}^{\infty}(\mathbf{I}-\hat{G})^k\|_\infty \\ \leq \frac{1}{c^2}\sum_{k=0}^{\infty}\|(\hat{G}-\mathbf{I})\|_\infty^k = \frac{1}{c^2 - c^2\|(\hat{G}-\mathbf{I})\|_\infty}. \end{aligned} \tag{G.2}$$

We now recognise that $c^2\|(\hat{G}-\mathbf{I})\|_\infty = \|(G - c^2\mathbf{I})\|_\infty = \mu_1(M-1)$, completing the proof. □

We can now give

Proof of lemma 9. We have

$$\begin{aligned} \lambda^{0.5}(\mathbf{y}^\star) &\leq \|(\Phi_\Gamma^H\Phi_\Gamma)^{-1}\Phi_\Gamma^H\mathbf{x}\|_\infty \\ &\leq \|(\Phi_\Gamma^H\Phi_\Gamma)^{-1}\|_\infty\|\Phi_\Gamma^H\mathbf{x}\|_\infty \\ &\leq \|(\Phi_\Gamma^H\Phi_\Gamma)^{-1}\|_\infty\|\mathbf{x}\|_2 \text{ maxcor}(x). \end{aligned} \tag{G.3}$$

Proposition G.1 then bounds the first term on the right proving the lemma. □

H. Matching Pursuit and Orthogonal Matching Pursuit.

Matching Pursuit (MP) [10] is a greedy iterative algorithm that calculates a sparse approximation using the following steps:

1. Initialise $R^0 = \mathbf{x}, \mathbf{y}^0 = 0$

2. $\alpha_i = \langle R^n, \phi_i \rangle$

3. $i_{max} = \arg_i \max |\alpha_i|$

4. $y_{i_{max}}^n = y_{i_{max}}^{n-1} + \alpha_{i_{max}}$

5. $R^n = R^{n-1} - \phi_i \alpha_{i_{max}}$

6. iterate from 2 until stopping criterion is fulfilled.

Orthogonal Matching Pursuit (OMP) [27] is a variation of MP in which in each iteration, the coefficient vector is the orthogonal projection of the signal onto the dictionary elements selected up to this iteration:

```
1. Initialise $R^0 = \mathbf{x}, \mathbf{y}^0 = 0, \Gamma_1^0 = \oslash$
2. $\alpha_i = \langle R^n, \phi_i \rangle$
3. $i_{max} = \arg_i \max |\alpha_i|$
4. $\Gamma_1^n = \Gamma_1^{n-1} \cup i_{max}$
5. $\mathbf{y}^n = \Phi^\dagger_{\Gamma_1^n} \mathbf{x}$
6. $R^n = \mathbf{x} - \Phi \mathbf{y}^n$
7. iterate from 2 until stopping criterion is fulfilled.
```

Here $\Phi^\dagger_{\Gamma_1^n}$ is the pseudo-inverse of the sub-dictionary $\Phi_{\Gamma_1^n}$.

Orthogonal matching pursuit is normally implemented using QR factorisation of the selected subset of elements and is computationally more demanding than Matching Pursuit. However, the projection ensures that the algorithm selects a new element in each iteration and that for the currently selected set of elements, the error is minimal.

Acknowledgments

This research was supported by EPSRC grant D000246/1. MED acknowledges support of his position from the Scottish Funding Council and their support of the Joint Research Institute with the Heriot-Watt University as a component part of the Edinburgh Research Partnership.

References

[1] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Transactions on Information Theory*, vol. 44, pp. 2435–2476, Oct. 1998.

[2] S. Mallat and F. Falzon, "Analysis of low bit rate image transform coding," *IEEE Transactions on Signal Processing*, vol. 46, pp. 1027–1042, Apr. 1998.

[3] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[4] M. Davies and N. Mitianoudis, "A simple mixture model for sparse overcomplete ICA," *IEE Proc.-Vision, Image and Signal Processing*, vol. 151, pp. 35–43, August 2004.

[5] T. Blumensath and M. Davies, "Sparse and shift-invariant representations of music," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, pp. 50–57, Jan 2006.

[6] D. L. Donoho and M. Elad, "Optimally-sparse representation in general (non-orthogonal) dictionaries via l1 minimization," *Proc. Nat. Aca. Sci.*, vol. 100, pp. 2197–2202, 2003.

[7] J. Tropp, "Just relax: Convex programming methods for identifying sparse signals

in noise," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 1031–1051, 2006.

[8] G. Davis, *Adaptive Nonlinear Approximations.* PhD thesis, New York University, 1994.

[9] B. K. Nataratjan, "Sparse approximat solutions to linear systems," *SIAM Journal on Computing*, vol. 24, pp. 227–234, Apr 1995.

[10] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[11] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[12] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 255–261, 2006.

[13] J. F. Murray and K. Kreutz-Delgado, "An improved FOCUSS-based learning algorithm for solving sparse linear inverse problems," in *Conf. Record of the Thirty-Fifth Asilomar Conf. on Signals, Systems and Computers*, pp. 347–351, 2001.

[14] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal of Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[15] I. Daubechies, M. Defries, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Pure and applied Mathematics*, vol. 57, pp. 1413–1457, 2004.

[16] M. Elad, "Why simple shrinkage is still relevant for redundant representation," *to appear in IEEE Transactions on Information Theory.*

[17] E. Candes and J. Romberg, "Signal recovery from random projections." *Proceedings of SPIE*, vol. 5674, pp. 76-86, 2005.

[18] K. Bredies and D. A. Lorenz, "Iterated hard shrinkage for minimization problems with sparsity constraints," *Preprint Nr. 144 of the DFG Schwerpunktprogramm 1114*, June 2006.

[19] N. G. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Journal of Applied and Computational Harmonic Analysis*, vol. 10, pp. 234–253, May 2001.

[20] K. K. Herrity, A. C. Gilbert, and J. A. Tropp, "Sparse approximation via iterative thresholding.," in *Proceedings of the Int. Conf. on Acoustics, Speech and Signal Processing*, 2006.

[21] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 1–20, Mar. 2006.

[22] L. Landwebe, "An iterative formula for fredholm integrals of the first kind," *American Journal of Mathematics*, vol. 73, pp. 615–624, Jul 1951.

[23] N. G. Kingsbury and T. H. Reeves, "Iterative image coding with overcomplete complex wavelet transforms," in *Proc. Conf. on Visual Communications and Image Processing*, 2003.

[24] D. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solutions of underdetermined linear equations by stagewise orthogonal matching pursuit.," *manuscript* 2006.

[25] S. Krustulovic and R. Gribonval, "MPTK: Matching pursuit made tractable," in *Proc. Int. Conf. on Acoustic Speech and Signal Processing*, (Toulouse, France), May 2006.

[26] W. Rudin, *Principles of Mathematical Analysis.* McGraw-Hill, 1976.

[27] S. Mallat, *A Wavelet Tour of Signal Processing.* Academic Press, 1999.

Received ...

Revision received ...

IDCOM & Joint Research Institute for Signal and Image Processing

Edinburgh University, King's Buildings, Mayfield Road
Edinburgh EH9 3JL, UK
e-mail: thomas.blumensath@ed.ac.uk

IDCOM & Joint Research Institute for Signal and Image Processing
Edinburgh University, King's Buildings, Mayfield Road
Edinburgh EH9 3JL, UK
e-mail: mike.davies@ed.ac.uk