

# Stagewise Weak Gradient Pursuits

## Part I: Fundamentals and Numerical Studies

Thomas Blumensath, *Member, IEEE*, Mike E. Davies, *Member, IEEE*

**Abstract**—Finding sparse solutions to underdetermined inverse problems is a fundamental challenge encountered in a wide range of signal processing applications, from signal acquisition to source separation. Recent theoretical advances in our understanding of this problem have further increased interest in their application to various domains. In many areas, such as for example medical imaging or geophysical data acquisition, it is necessary to find sparse solutions to very large underdetermined inverse problems. Fast methods have therefore to be developed.

In this paper, we promote a greedy approach. In each iteration, several new elements are selected. The selected coefficients are then updated using a conjugate update direction. This is an extension of the previously suggested Gradient Pursuit framework to allow an even greedier selection strategy.

A large set of numerical experiments, using artificial and real world data, demonstrate the performance of the method. It is found that the approach performs consistently better than other fast greedy approaches, such as Regularised Orthogonal Matching Pursuit and Stagewise Orthogonal Matching Pursuit and is competitive with other fast approaches, such as those based on  $\ell_1$  minimisation. It is also shown to have the unique property to allow a smooth trade-off between signal sparsity (or observation dimension) and computational complexity.

Theoretical properties of the method are studied in a companion paper [2].

**Index Terms**—Sparse Representations/Approximations, Orthogonal Matching Pursuit, Weak Matching Pursuit, Gradient Pursuit, Stagewise Selection, Compressed Sensing.

### I. INTRODUCTION

Sparse signal expansions are general signal models, applicable to a wide range of signals, that approximate a signal using a linear combination of a small number of elementary waveforms selected from a large collection. These models have over the last few years found applications in a wide range of areas, from source coding [3] to de-noising [4], source separation [5] and signal acquisition [6] (i.e. compressed sensing).

A sparse signal model is specified by a matrix  $\Phi \in \mathbb{R}^{M \times N}$  with typically more columns than rows, that is with  $M < N$ .  $\Phi$  is often known as the dictionary or the measurement matrix,

The authors are with IDCOM & Joint Research Institute for Signal and Image Processing, Edinburgh University, King's Buildings, Mayfield Road, Edinburgh EH9 3JL, UK (Tel.: +44(0)131 6505659, Fax.: +44(0)131 6506554, e-mail: thomas.blumensath@ed.ac.uk, mike.davies@ed.ac.uk).

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. The authors would like to thank Ian Marshall and Yuehui Tao for providing access to the MRI image data.

Some of the work presented here has previously been presented in [1].

© This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

depending on the application. The column vectors  $\phi_i$  of  $\Phi$  are sometimes called atoms and are here assumed to be of unit length unless stated otherwise. Given an observation  $\mathbf{x} \in \mathbb{R}^M$ , a sparse signal model approximates  $\mathbf{x}$  using a small subset of columns from  $\Phi$ , i.e.

$$\hat{\mathbf{x}} = \Phi \hat{\mathbf{y}},$$

where  $\hat{\mathbf{y}}$  is a vector with most of its elements being zero. If we allow for a non-zero error  $\mathbf{n} = \mathbf{x} - \hat{\mathbf{x}}$  we talk about a signal *approximation*, while if  $\hat{\mathbf{x}} = \mathbf{x}$  we have an exact signal *representation*.

If  $M < N$ , then there are an infinite number of  $\hat{\mathbf{y}}$  such that  $\mathbf{x} = \Phi \hat{\mathbf{y}}$ . The problem is then to find an estimate  $\hat{\mathbf{y}}$  that is sparse, such that the norm of  $\mathbf{n}$  is small. Whilst there are a range of ways in which sparsity could be measured, the most typical is probably to count the number of elements of  $\hat{\mathbf{y}}$  that are non zero. Even though this is not a norm, we denote this ‘numerosity’ as  $\|\cdot\|_0$ , informally called  $\ell_0$  norm.

The problem of finding a vector  $\hat{\mathbf{y}}$  with minimal  $\ell_0$  norm, under constraint on  $\|\mathbf{n}\|_2$  is known to be NP-hard in general [7], [8] and different sub-optimal strategies are used in practise. Commonly used strategies are typically based on convex relaxation, non-convex (often gradient based) local optimisation or greedy search strategies. Convex relaxation, one of the most popular strategies at the moment, is used in algorithms such as Basis Pursuit and Basis Pursuit De-Noising [9], the Least Absolute Shrinkage and Selection Operator (LASSO) and Least Angle Regression (LARS) [10]. Recently, fast algorithms solving the LASSO or the Basis Pursuit De-Noising problem have been suggested in [11], [12] and [13]. Non-convex local optimisation procedures include the Focal Underdetermined System Solver FOCUSS [14] and re-weighted  $\ell_1$  minimisation [15] while Bayesian approaches include the Relevance Vector Machine, also known as Sparse Bayesian Learning [16] [17] or Monte Carlo based approaches such as those in [18], [19] and [20]. Another very popular approach is to use greedy algorithms, the most important of which are Matching Pursuit (MP) [21], Orthogonal Matching Pursuit (OMP) [22] and Orthogonal Least Squares (OLS) [23], also often known as ORMP, OOMP or, in the regression literature, as forward selection. OMP typically shows greatly superior performance to MP, however, OMP is more costly in both computation time and storage requirement.

There are two main problems associated with the application of OMP to large data sets. On the one hand, the computation cost per iteration is high, both in terms of storage and computation. This problem was addressed in [24], where we have introduced a quite general framework for greedy algorithms, called collectively Gradient Pursuits (GP). Based on this

idea, we have developed two particular algorithms, with the computational complexity of MP, but with performance more akin to OMP. Compared to OMP, this has greatly reduced the computational and storage requirements per iteration, making the method applicable to large data sets. This approach is reviewed in section II-B.

However, another performance limitation of greedy methods such as MP, OMP as well as the Gradient Pursuits algorithms of [24], is that these methods select a single element per iteration. They have therefore to be run for at least as many iterations as there are non-zero elements in the solution. The remainder of this paper addresses this issue and develops a *Stagewise Weak* selection procedure that will allow several elements to be selected in each iteration. This, combined with the strategies from [24], will be shown to lead to very fast and efficient algorithms to solve the sparse signal modelling problem.

### A. Paper Overview

The developments in this paper are extensions of Orthogonal Matching Pursuit (OMP), which will be reviewed in subsection II-A, whilst subsection II-B reviews the Gradient Pursuit framework of [24].

Two previously suggested methods that select several elements per iteration are then discussed in section III. To overcome some of the problems associated with these methods, we devise what will be called a *Stagewise Weak* selection strategy. Using this method with OMP leads to the Stagewise Weak Orthogonal Matching Pursuit (SWOMP) algorithm. Experimental results in subsection V-A will highlight the advantage of this approach.

To further reduce the computational cost, we combine the Stagewise Weak selection with Gradient Pursuit in section IV. Section V presents numerical results that demonstrate the advantages of this approach when compared to other fast algorithms.

### B. Notation

The algorithms in this paper are iterative and the current iteration will be iteration  $n$ . The algorithms will keep track of a set  $\Gamma^{[n]}$  of indices, that will be grown in each iteration. These indices label a subset of columns from a matrix  $\Phi$  and, using the index set as a subscript, the matrix  $\Phi_{\Gamma^{[n]}}$  will be a sub-matrix of  $\Phi$  containing only those columns of  $\Phi$  with indices in  $\Gamma^{[n]}$ . The same convention is used for vectors. In general, the superscript in the subscript of  $\hat{\mathbf{y}}_{\Gamma^{[n]}}$  reminds us that we are in iteration  $n$ , on occasion, however, we resort to using additional superscripts (e.g.  $\hat{\mathbf{y}}^{[n]}$ ) to label the iteration. The gram matrix  $\mathbf{G}_{\Gamma^{[n]}} = \Phi_{\Gamma^{[n]}}^T \Phi_{\Gamma^{[n]}}$  will also be used frequently. In general, lower case bold face characters represent vectors while upper case bold face characters are used for matrices. Individual elements from a vector will be in standard type face with a subscript. For example  $\mathbf{g}$  will be used to refer to a negative gradient vector with  $g_i$  denoting the  $i^{th}$  element of this vector. Inner products between vectors will often be written using angled brackets, e.g.  $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y}$ . We will further use the hat  $\hat{\cdot}$  to distinguish an estimated quantity from the true quantity, which will be written without the hat.

## II. GREEDY PURSUITS

The algorithms discussed in this paper are generalisations of the Gradient Pursuit framework presented in [24], which in turn was developed as a generalisation of Orthogonal Matching Pursuit. We therefore review these two algorithms in this section.

### A. Orthogonal Matching Pursuit

Orthogonal Matching Pursuit (OMP) is an iterative algorithm that tries to find a ‘small’ set  $\Gamma$  and a vector  $\hat{\mathbf{y}}_{\Gamma}$  to approximate  $\mathbf{x}$  by

$$\hat{\mathbf{x}} = \Phi_{\Gamma} \hat{\mathbf{y}}_{\Gamma}.$$

OMP is initialised by setting the first residual  $\mathbf{r}^{[0]} = \mathbf{x}$ , setting  $\hat{\mathbf{y}}^{[0]} = \mathbf{0}$  and the set  $\Gamma^{[0]} = \emptyset$ . Each iteration updates these three quantities as follows:

- 1) Select a *single* column from  $\Phi$  based on the inner products  $\Phi^T \mathbf{r}^{[n]}$  and add its index to  $\Gamma^{[n-1]}$ .
- 2) Update the estimate of  $\hat{\mathbf{y}}$  by solving  $\hat{\mathbf{y}}_{\Gamma^{[n]}}^{[n]} = \min_{\hat{\mathbf{y}}_{\Gamma^{[n]}}} \|\mathbf{x} - \Phi_{\Gamma^{[n]}} \hat{\mathbf{y}}_{\Gamma^{[n]}}\|_2^2$ .
- 3) Update the residual  $\mathbf{r}^{[n]} = \mathbf{x} - \Phi_{\Gamma^{[n]}} \hat{\mathbf{y}}_{\Gamma^{[n]}}^{[n]}$ .

In step (1), new elements are selected based on the inner products between the column vectors of  $\Phi$  and the current residual. These inner products are calculated as

$$\mathbf{g}^{[n]} = \Phi^T \mathbf{r}^{[n-1]}. \quad (1)$$

OMP selects a *single* element and adds it to the index set

$$\Gamma^{[n]} = \Gamma^{[n-1]} \cup \arg_i \max |g_i^{[n]}|.$$

In an exact implementation of OMP elements will only be selected once, because the orthogonal projection used in the coefficient update (see below) ensures that the residual  $\mathbf{r}^{[n]}$  is orthogonal to all columns in  $\Phi_{\Gamma^{[n]}}$ . However, if this orthogonalisation is only approximated as, for example, in the Stagewise Orthogonal Matching Pursuit (StOMP) algorithm [25] (discussed below) or in Gradient Pursuit [24], it is advisable (both from theoretical arguments and empirical evidence) to allow the algorithm to re-select elements.

After updating the set  $\Gamma^{[n]}$  the coefficients  $\hat{\mathbf{y}}^{[n]}$  are updated using  $\hat{\mathbf{y}}_{\Gamma^{[n]}}^{[n]} = \Phi_{\Gamma^{[n]}}^{\dagger} \mathbf{x}$  where  $\Phi_{\Gamma^{[n]}}^{\dagger}$  is the pseudo inverse of  $\Phi_{\Gamma^{[n]}}$ . In an efficient implementation, the pseudo inverse is in general not calculated explicitly in each iteration. Instead, fast implementations of OMP either keep track of a QR factorisation of  $\Phi_{\Gamma^{[n]}}$ , which is updated efficiently in each iteration or, alternatively, keep track of a Cholesky factorisation of the Gram matrix  $\mathbf{G}_{\Gamma^{[n]}}^{[n]} = \Phi_{\Gamma^{[n]}}^T \Phi_{\Gamma^{[n]}}$  which is also updated from iteration to iteration. More details on these methods can be found in, for example, [24].

Finally, the residual is updated. The new residual is  $\mathbf{r}^{[n]} = \mathbf{x} - \Phi_{\Gamma^{[n]}} \hat{\mathbf{y}}_{\Gamma^{[n]}}^{[n]}$ , however, depending on the detailed implementation of the algorithm, the matrix vector product can often be replaced by a fast recursion.

For problems in which  $\Phi$  is large, two problems arise. Firstly, storage of  $\Phi$  can be problematic. Secondly, matrix vector products involving  $\Phi$  or its adjoint are costly. Therefore, in many applications,  $\Phi$  is designed with additional structure. This means that  $\Phi$  does not have to be stored explicitly and

that matrix vector products involving  $\Phi$  and its adjoint can be evaluated more efficiently. For example, if the fast Fourier transform is used, the computation time can often be reduced to be  $O(N \log M)$  instead of the  $O(MN)$  for unstructured matrices.

### B. Gradient Pursuit

To overcome several of the shortcomings of OMP discussed above, we have previously argued to approximate the orthogonal projection using directional optimisation, which can be done much more efficiently [24]. The Gradient Pursuit family of algorithms uses directional optimisation to update  $\hat{\mathbf{y}}^{[n-1]}$  in each iteration.

$$\mathbf{y}^{[n]} = \mathbf{y}^{[n-1]} + \beta \mathbf{d}^{[n]}.$$

Whilst the updates in Matching Pursuit and Orthogonal Matching Pursuits also fall into this framework, different directions  $\mathbf{d}^{[n]}$  might be beneficial. In [24], the gradient and an approximate conjugate gradient method were suggested. In the gradient based approach, the update direction is

$$\mathbf{d}_{\Gamma^{[n]}}^{[n]} = \mathbf{g}_{\Gamma^{[n]}}^{[n]} = \Phi_{\Gamma^{[n]}}(\mathbf{x} - \Phi_{\Gamma^{[n]}}\hat{\mathbf{y}}_{\Gamma^{[n]}}^{[n-1]}). \quad (2)$$

The (approximate)<sup>1</sup> conjugate gradient method uses the direction

$$\mathbf{d}_{\Gamma^{[n]}}^{[n]} = \mathbf{g}_{\Gamma^{[n]}}^{[n]} + \omega \mathbf{d}_{\Gamma^{[n]}}^{[n-1]}, \quad (3)$$

where  $\omega$  is calculated such that  $\langle \Phi_{\Gamma^{[n]}}\mathbf{d}_{\Gamma^{[n]}}^{[n]}, \Phi_{\Gamma^{[n]}}\mathbf{d}_{\Gamma^{[n]}}^{[n-1]} \rangle = 0$ , that is, consecutive update directions are *conjugate*.

Importantly, we have recently found an additional recursion, not given in [24], that allows the approximate conjugate gradient to be calculated more efficiently, so that the evaluation of the gradient and the approximate conjugate gradient have roughly equal computation costs. What is more, we have now also a proof of the convergence of the approximate conjugate gradient method that guarantees that using (3) reduces the residual's  $\ell_2$  norm more than using (2). Both, the new recursion as well as the convergence proof can be found in the companion paper [2]. Finally, extensive experimental evaluations [24] have shown that the approximate conjugate direction based approach outperforms the gradient based method in general.

When compared to the original OMP approach, the performance of this directional update step was shown [24] to offer significant advantages in terms of computational requirements, while on the other hand, leading to only minor decreases in performance. We therefore endorse the use of the approximate conjugate gradient approach, which for simplicity will be called Conjugate Gradient Pursuit (CGP) throughout this paper.

## III. GREEDIER PURSUITS: FROM THRESHOLDING TO OMP

Having thus significantly reduced the computational complexity of each iteration, we now turn to the problem of reducing the overall number of iterations. Before introducing

<sup>1</sup>Approximate, because the method only guarantees conjugacy of the current update direction to the previous update direction, but not to all previous update directions [24].

our proposed solution, we briefly review two other strategies that have recently been put forward to select several elements per iteration in an OMP type algorithm. Both of these methods use the inner products  $g_i = \phi_i^T \mathbf{r}^{[n-1]}$  to select new elements.

### A. Stagewise Orthogonal Matching Pursuit (StOMP)

The first approach, called Stagewise Orthogonal Matching Pursuit (StOMP) [25], calculates a threshold

$$\lambda_{stomp} = t \|\mathbf{r}^{[n-1]}\|_2 / \sqrt{M}.$$

All those elements whose inner products have a magnitude larger than this threshold are then selected. The set of indices is therefore updated as

$$\Gamma^{[n]} = \Gamma^{[n-1]} \cup \{i : |g_i| \geq \lambda_{stomp}\}. \quad (4)$$

As mentioned above, in [25], instead of calculating an exact orthogonalisation, i.e. instead of calculating  $\hat{\mathbf{y}}_{\Gamma^{[n]}}^{[n]} = \Phi_{\Gamma^{[n]}}^\dagger \mathbf{x}$  using QR or Cholesky factorisation, the authors suggested to approximate this orthogonalisation using a few iterations of a conjugate gradient solver. The motivation behind this was based on the assumption that sub-matrices of random matrices are well behaved as their condition number is small. See also our discussion regarding the convergence of gradient based algorithms presented in [2].

StOMP was developed explicitly for problems in which  $\Phi$  has been generated from a Uniform Spherical Ensemble, i.e. the columns of  $\Phi$  are drawn uniformly from the unit sphere. Performance guarantees for this method when applied to more general matrixes  $\Phi$  are therefore not available. Also, the selection of the parameter  $t$  required in the method is critical for its performance, but there do not seem to be any intuitive guidelines available for this other than the suggestion in [25] to use a value between 2 and 3. Furthermore, a problem when using the residual to define a threshold is that the algorithm might (and in our experience sometimes does) get 'stuck' when all inner products fall below the threshold. It would then be necessary to reduce the parameter  $t$ . In many of our own experiments, this approach has therefore shown mixed results.

### B. Regularised Orthogonal Matching Pursuit (ROMP)

In compressed sensing, one important direction of recent research was to study conditions on the size  $M$  of the observation  $\mathbf{x}$  that allow uniform performance guarantees for different algorithms, that is, guarantees under which, for a given  $\Phi$ , a given algorithm will reconstruct *all*  $K$  sparse signals. It has been shown that for many greedy algorithms such as OMP, the required conditions on  $M$  to allow such uniform guarantees are not as good as those for  $\ell_1$  based approaches. These theoretical considerations led to the introduction of the Regularised Orthogonal Matching Pursuit (ROMP) algorithm [26], [27]. In this strategy the inner products  $g_i$  are grouped into sets  $J_k$ , such that the elements in each set have a similar magnitude, i.e. they satisfy

$$|g_i| \leq \frac{1}{r} |g_j|, \text{ for all } i, j \in J_k.$$

ROMP selects the set  $J_k$  for which  $\|\mathbf{g}_{J_k}\|_2$  is largest.

In [26], [27],  $r$  was assumed to be 0.5. In this case, the algorithm was shown to have uniform performance guarantees similar to those of  $\ell_1$  based methods. Whilst these results indicate that, asymptotically for very large  $N$ , the performance of ROMP should be better than that of OMP, the particular constants involved in the theoretical guarantees are significantly smaller than those in the equivalent statements for  $\ell_1$  methods. Unfortunately, practice is often far from asymptotia. Also, in many practical situations, one might be more interested in average rather than uniform performance. In almost all practical applications we have studied, the average performance of ROMP was notably worse than that of OMP or StOMP.

### C. Stagewise Weak Orthogonal Matching Pursuit (SWOMP)

Instead of using the norm of the residual to define a threshold for element selection as done in StOMP, we propose the use of a threshold based on the maximum of  $|g_i|$ . This idea was inspired by the *Weak Matching Pursuit* algorithm [28]. Weak Matching Pursuit is a method developed for large or infinite dimensional problems in which not all inner products can be evaluated explicitly. To accomplish this, a weakness parameter  $\alpha \in (0, 1]$  is introduced into the selection criterion. Weak Matching Pursuit selects *any one* element such that

$$|g_j| \geq \alpha \max_i |g_i|.$$

Instead of selecting a *single* element satisfying the above condition, in applications in which all the inner products are available, one can select *all* element that satisfy this condition. This selection strategy will be call *stagewise weak* selection. Using this approach, the set of indices is updated as

$$\Gamma^{[n]} = \Gamma^{[n-1]} \cup \{i : |g_i| \geq \alpha \max_j |g_j|\}, \quad (5)$$

that is, we select *all* elements that come within a factor of  $\alpha$  of the largest inner product (in magnitude).

Let us briefly consider the relationship between the StOMP selection and this stagewise weak selection. Both algorithms select atoms by applying thresholding to  $|\Phi^T \mathbf{r}^{n-1}|$ . For the stagewise weak selection we have:

$$\lambda_{wss} = \alpha \|\Phi^T \mathbf{r}^{n-1}\|_\infty.$$

Using norm inequalities we can see that:

$$\frac{\alpha \sqrt{M}}{t \sqrt{N}} \sigma_M(\Phi) \lambda_{stomp} \leq \lambda_{wss} \leq \frac{\alpha}{t} \sqrt{M} \sigma_1(\Phi) \lambda_{stomp} \quad (6)$$

where  $\sigma_k(\Phi)$  denotes the  $k^{th}$  singular value of  $\Phi$ . We thus see that the two thresholds are similar, however the key difference lies in the fact that the stagewise weak threshold is a function of the correlation between the atoms and the residual rather than only a function of the residual. This allows us to extend OMP recovery results to stagewise weak algorithms. We present these in the companion paper [2].

When using the stagewise weak selection in OMP, we get a Stagewise Weak OMP (SWOMP) algorithm. An important property of such a strategy worth stressing is that by changing  $\alpha$ , SWOMP interpolates between two well known methods for

sparse approximation. A thresholding algorithm is obtained with small  $\alpha$ , whilst for  $\alpha = 1$  SWOMP becomes standard OMP.

### IV. FASTER AND GREEDIER: PUTTING IT ALL TOGETHER

Whilst it might seem intuitive that using fewer iterations will lead to a faster algorithm, this is not necessarily true when using a stagewise selection strategies with traditional implementations of OMP. For example, in the QR or Cholesky factorisation based implementations of OMP, the QR or Cholesky factorisations have to be updated for *each* of the newly selected elements. Overall, there will be as many of these updates as there are elements to be selected. As the updates dominate the computation cost, the computational advantage of using stagewise selection strategies with OMP are therefore small.

Instead, StOMP [25] used a small number of conjugate gradient steps in each iteration to approximate the required orthogonalisation. We promote the use of the Conjugate Gradient Pursuit algorithm to do the required approximate orthogonalisation. The selection step in the Conjugate Gradient Pursuit algorithm of section II-B is replaced by the stagewise weak selection step in (5) such that the overall number of iterations is potentially reduced significantly, while the computational complexity of each iterations remains the same as that of the standard Conjugate Gradient Pursuit method. This approach will be referred to as a Stagewise Weak Conjugate Gradient algorithm (SWCGP).

The weak selection strategy has now been incorporated into the implementation of CGP in the sparsify matlab toolbox to be found on the first authors web-page. The algorithm is accessible through the call to function `greed_nomp` (`nomp` for Nearly Orthogonal Matching Pursuit).

### V. NUMERICAL EVALUATION

Let us now turn to a numerical evaluation of the approaches discussed in this paper. We start by comparing the three stagewise selection strategies. In this experiment, we use these together with the full orthogonalisation of OMP. Whilst this combination does not offer significant computational advantages, it allows a study of the selection steps in isolation, without any effects introduced by the approximation procedure used for the orthogonalisation. The other experiments then look at the combination of the stagewise weak strategy with Conjugate Gradient Pursuit.

#### A. Experimental Comparison of Stagewise OMP algorithms

In the first set of experiments we evaluate the performance of the different selection strategies when used in conjunction with the exact orthogonalisation to estimate the coefficients. We here used a QR factorisation based approach to calculate this orthogonalisation.

The data for the experiments was produced as follows.  $\Phi \in \mathbb{R}^{128 \times 256}$  was generated by drawing each column independently and uniformly from the unit sphere.  $\mathbf{y}$  was generated by drawing the first  $K$  elements from i.i.d. normal distributions. The observations were generated as  $\mathbf{x} = \Phi \mathbf{y} + \mathbf{n}$

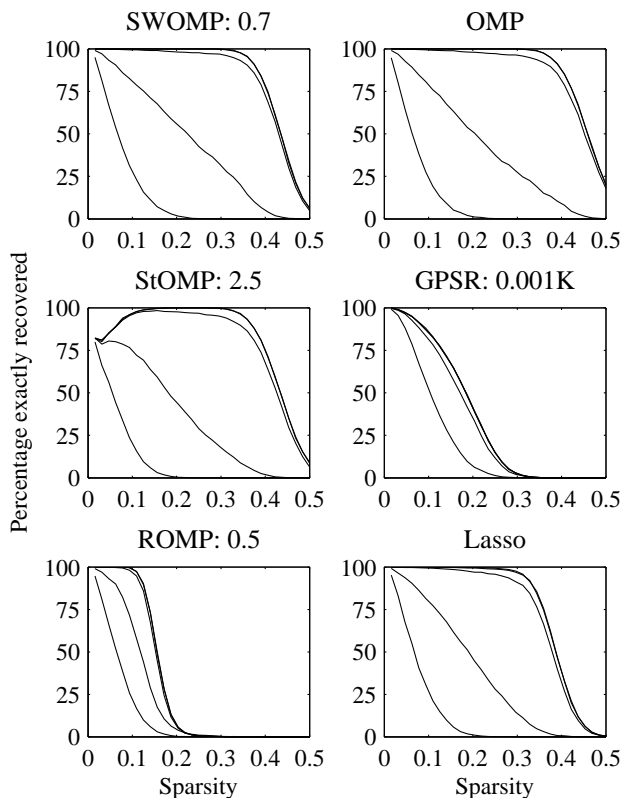


Figure 1: Normal distributed non-zero coefficients. Exact recovery performance of SWOMP, StOMP, ROMP, OMP, GPSR and Lasso for different observation SNR values (from top right to bottom left in performance SNR = 120dB, 60dB, 30dB, 15dB (Note, the 120dB result are the same as those found with no noise, i.e.  $\infty$ dB.)). The abscissa shows the ration between non-zero elements  $K$  and the observation dimension  $M$ . All results averaged over 10 000 realisations.

where  $\mathbf{n}$  was i.i.d. Gaussian noise, which was normalised so that the observations had a specified signal to noise ratio (SNR). We varied the number of non-zero elements  $K$  and the SNR ( $\infty$ dB, 120dB, 60dB, 30dB, and 15dB). All results are averaged over 10 000 problem realisations.

We compared OMP, StOMP, ROMP<sup>2</sup> and our stagewise weak OMP algorithm (SWOMP). We further show the results obtained by optimising the cost function  $\|\mathbf{x} - \Phi\mathbf{y}\|_2^2 + \lambda\|\mathbf{y}\|_1$  with  $\lambda = 0.001K$  (followed by an orthogonal projection onto the non-zero elements). We here used the GPSR algorithm of [12]. In addition we also optimised the LASSO cost function  $\|\mathbf{y}\|_1$  under the constraint that  $\|\mathbf{x} - \Phi\mathbf{y}\|_2 \leq \epsilon$  using the homotopy based algorithm of [10] as implemented in the SparseLab<sup>3</sup> toolbox (available at <http://sparselab.stanford.edu/>).

The important difference between GPSR and LASSO is that

<sup>2</sup>Note that, in all of our experiments, we used the same selection strategy for ROMP used in the code provided by the authors of [26]. In line with the theory developed in [26], this selection strategy only considers disjoint subsets for elements selection and not all subsets. This is faster but reduces the empirical performance of the method somewhat.

<sup>3</sup>Note also that the Lasso solution can also be found with the algorithm in [13], which in our experience is faster than the SparseLab implementation, however, we found it to be slower than the GPSR method.

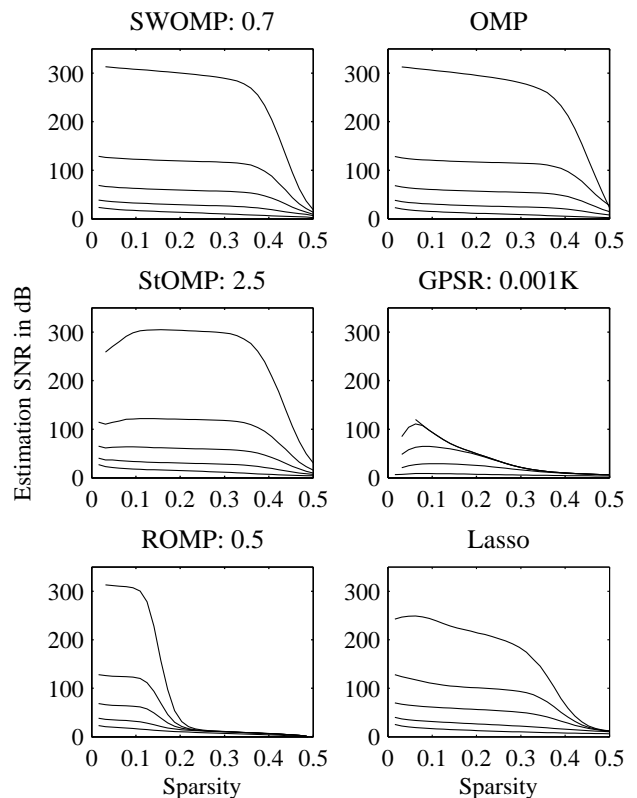


Figure 2: Normal distributed non-zero coefficients. Average SNR of recovered coefficient vector for SWOMP, StOMP, ROMP, OMP, GPSR and Lasso for different observation SNR values (from top right to bottom left in performance SNR =  $\infty$ dB, 120dB, 60dB, 30dB, 15dB). The abscissa shows the ration between non-zero elements  $K$  and the observation dimension  $M$ . All results averaged over 10 000 realisations.

in GPSR we have to specify  $\lambda$  whilst for LASSO we specify  $\epsilon$  (which in our experiments was set to  $\|\mathbf{n}\|_2$ ). In general, if we would have the ‘correct’ choice of  $\lambda$  for each problem instance, the algorithm would perform as well as LASSO. However, finding this  $\lambda$  is non-trivial as it is not just a function of  $K$ , but depends also on  $\mathbf{n}$  and  $\mathbf{y}$ . Our choice of  $\lambda$  was therefore a compromise such that the algorithm chose on average as many elements as the greedy strategies.

OMP, StOMP, ROMP and SWOMP, were run until they had selected *twice* as many elements as were used to generate the observations<sup>4</sup>. For StOMP, ROMP and SWOMP, we used the parameters  $t = 2.5$ ,  $r = 0.5$  and  $\alpha = 0.7$ .

Figure 1 compares the algorithms in terms of their performance in identifying the exact location of the non-zero elements in the coefficient vector used to generate the signal. The abscissa shows the ratio between the number of non-zero elements  $K$  used to generate the signal and the observation dimension  $M$ .

<sup>4</sup>Note that in the noiseless case and for random  $\Phi$ , if there is a  $K$ -sparse vector  $\mathbf{y}$ , such that  $\mathbf{x} = \Phi\mathbf{y}$  and if  $K/M < 0.5$ , this vector will be unique almost surely as any other vector  $\mathbf{y}$  s.t.  $\mathbf{x} = \Phi\mathbf{y}$  will have more than  $2K$  non-zero elements. Therefore, any  $2K$ -sparse vector that satisfies  $\mathbf{x} = \Phi\mathbf{y}$  has to be equal to the unique  $K$ -sparse vector [26].

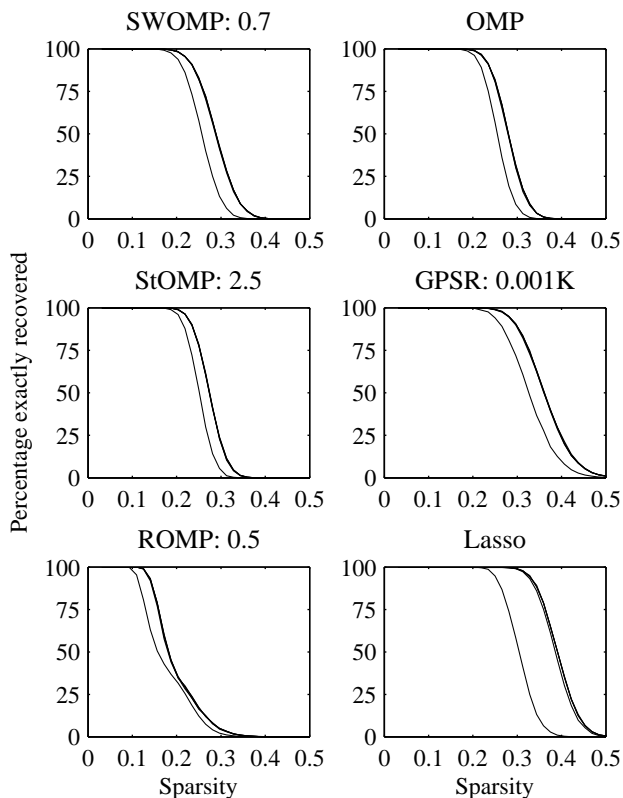


Figure 3: Bernoulli distributed non-zero coefficients. Exact recovery performance of SWOMP, StOMP, ROMP, OMP, GPSR and Lasso for different observation SNR values (the top right curves overlap and are the results found for a SNR of  $\infty$ dB, 120dB, 60dB and 30dB, whilst the bottom left curve is the result for 15dB). The abscissa shows the ration between non-zero elements  $K$  and the observation dimension  $M$ . All results averaged over 10 000 realisations.

We classified the coefficients to be exactly recovered whenever the largest (in magnitude)  $K$  coefficients in the estimate  $\hat{y}$  coincided with the locations of the non-zero elements in the true coefficient vector  $y$ . Due to the selection criterion in StOMP, this algorithm sometimes terminated before it had selected  $K$  elements. Similarly, the Lasso and GPSR algorithms also on occasions selected less than  $K$  elements. In these cases, we classified the results as not exactly recovered.

One important aspect of the experiment was that we here recovered twice as many non-zero elements as were used to generate the signal. Using this, it can be seen that OMP outperforms the Lasso algorithm, at least for normally distributed non-zero coefficients (see however the results below for Bernoulli coefficients). More importantly, the SWOMP algorithm, with a weakness factor of 0.7, also performs better than Lasso. Whilst the theoretical results for greedy strategies are typically worse than those for  $\ell_1$  based approaches like Lasso, these results suggest that there might be better results possible for greedy strategies at least on average. However, it seems to help to allow the algorithm to select several incorrect elements as done here. When running the greedy algorithms until they had selected as many elements as there were non-

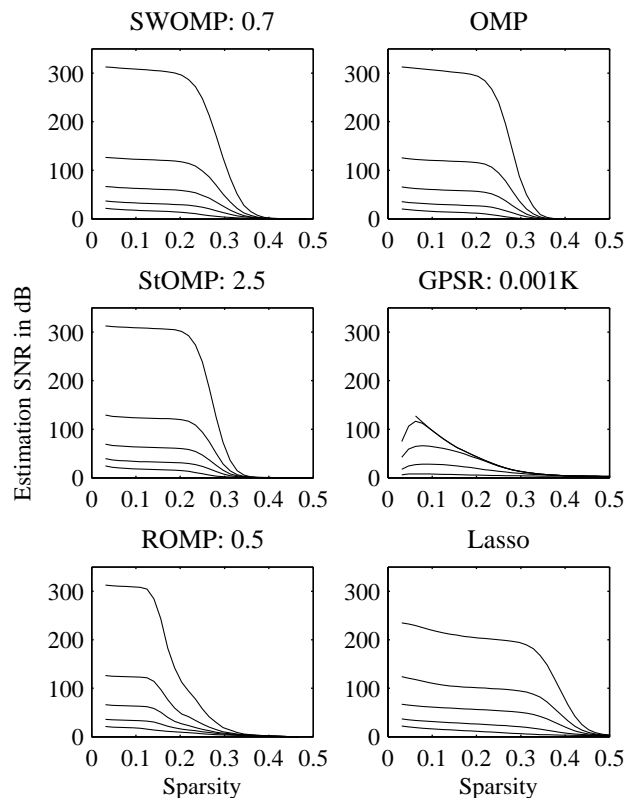


Figure 4: Bernoulli distributed non-zero coefficients. Average SNR of recovered coefficient vector for SWOMP, StOMP, ROMP, OMP, GPSR and Lasso for different observation SNR values (from top right to bottom left in performance SNR =  $\infty$ dB, 120dB, 60dB, 30dB, 15dB ). The abscissa shows the ration between non-zero elements  $K$  and the observation dimension  $M$ . All results averaged over 10 000 realisations.

zero elements, the performance was somewhat worse than those shown here, however, OMP was still comparable in performance to the Lasso method.

Exact recovery is not the only story, especially for noisy observations, where it might be more instructive to look at the signal to noise ratio (SNR) of the estimate  $\hat{y}$ . This is done in figure 2, where we show the averaged SNR values in dB<sup>5</sup>.

For comparison, we repeated the experiment using Bernoulli distributed non-zero coefficients. The results are shown in figure 3 and figure 4 respectively. It is known that the  $\ell_1$  methods are insensitive to different distributions of the non-zero coefficients, whilst greedy approaches such as OMP are typically assumed to perform worse if the non-zero coefficients are all of similar magnitude. This can also be observed here. It is also worth noting that the apparent improvement in the performance of GPSR in terms of exact recovery is here mainly due to the fact that with the value used for the regularisation parameter  $\lambda$ , which we set to the same value as in the other experiment, the method typically recovered far more non-zero elements than were required. There was therefore no improvement in terms of the SNR performance.

<sup>5</sup>We here first converted the SNR values to dB, and then averaged.

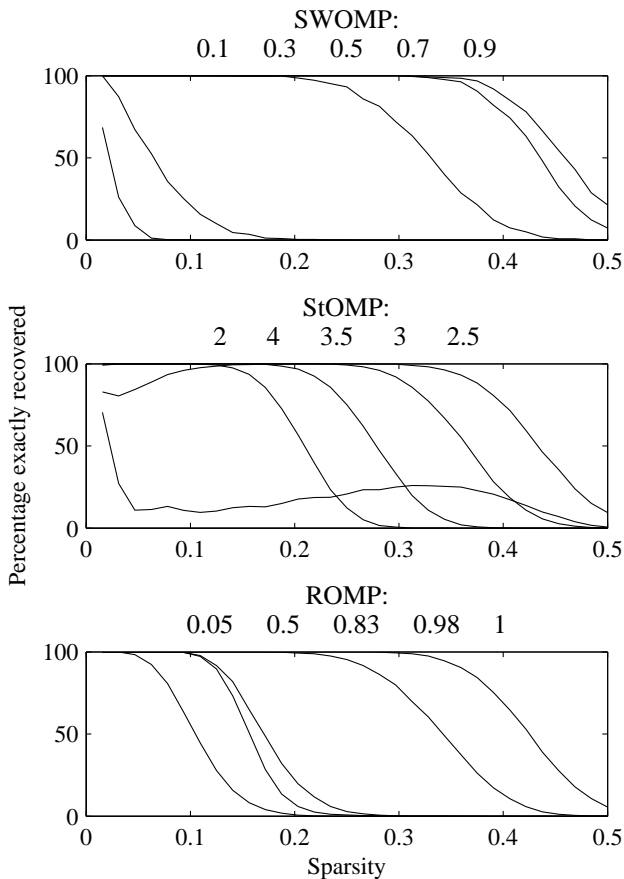


Figure 5: Normal distributed non-zero coefficients. Comparison between SWOMP, StOMP, ROMP for different parameters. The parameters used are shown above each panel (for SWOMP, from left to right,  $\alpha = 0.1, 0.3, 0.5, 0.7$  and  $0.9$ , for StOMP, from left to right,  $t = 2, 4, 3.5, 3$  and  $2.5$  and for ROMP, from left to right,  $r = 0.05, 0.5, 0.83, 0.98$  and  $1$ ). The abscissa shows the ration between non-zero elements  $K$  and the observation dimension  $M$ . All results averaged over 10 000 realisations.

It is also interesting to note that with Bernoulli coefficients, the recovery performance of all approaches seems much more robust against noise, so much in fact that the curves for SNR values of  $\infty$ dB, 120dB, 60dB and 30dB basically lay on top of each other, whilst only the curve for an observation SNR of 15dB is markedly different.

In a second set of experiments we evaluated the influence of the parameters  $t$ ,  $r$  and  $\alpha$  on the performance of the fast selection strategies. The experimental setup was here the same as in the first experiment above. It should be noted that in the development of ROMP in [26],  $r$  was assumed to be 0.5. However, this parameter can be set to a more general value between 0 and 1. In this case the theoretic bounds in [26] do not hold in general<sup>6</sup>. The results, again in terms of exact recovery, are shown in figure 5. Note that in this figure, the

<sup>6</sup>Interestingly, using  $r = 1$ , the algorithm is virtually identical to OMP. In this case, the theoretic results in [26] would qualitatively reduce to those available for OMP [29].

parameters are shown above each of the panels. Importantly, (in particular for StOMP) the parameters are shown in that order in which the associated graph appears in the panel. While for ROMP and SWOMP, a decrease in the parameter leads to a decrease in performance. This is not true for StOMP. Here, a parameter of 2.5 works better than larger or smaller values in general. However, very sparse signals are often not recovered with this parameter as the algorithm often stopped before it had selected  $K$  elements.

Whilst StOMP and SWOMP can both perform well, the influence of the parameter  $t$  in StOMP is more complicated than the smooth decay in performance observed for SWOMP. ROMP, even though it has certain nice theoretic properties when  $r = 0.5$ , does show significantly worse average performance when compared with SWOMP and StOMP. When increasing  $r$  and  $\alpha$  to one, both ROMP and SWOMP are effectively OMP, it is therefore not surprising that they both have comparable performance in this limit. However, the number of iterations both methods used when, for example  $\alpha = 0.5$  and  $r = 0.98$  (notice the performance of these methods is similar in this case) were quite different. In this case for a sparseness  $K/M = 0.2$ , ROMP used on average nearly four times as many iterations than SWOMP. As the cost for each iteration is the same, SWOMP was significant faster than ROMP when the parameter was adjusted such that both methods had a similar recovery performance.

### B. SWCGP vs OMP and MP

The next step is to evaluate the influence of replacing the exact orthogonalisation with the conjugate gradient update in (3) and to evaluate the influence of varying  $\alpha$ . We therefore repeated the above experiment using the SWCGP algorithm. We here only run the method until it had selected  $K$  elements. Figure 6 studies the influence of the weakness parameter  $\alpha$ . For  $\alpha = 1$ , the method is equivalent to CGP. For comparison, also shown are the results obtained with OMP and Matching Pursuit (MP).

It is clear that weakening the selection criterion reduces (in a controlled manner) the recovery performance. The advantage of this is a reduction in computational cost. This is shown in figure 7. Here the curves correspond to (from top to bottom)  $\alpha$  decreasing from 1 in steps of 0.05. The top curve indicates that the computational cost for CGP (SWCGP with  $\alpha = 1.0$ ) grows linearly with the number of non-zero coefficients. In contrast for  $\alpha < 1.0$  the computational cost grows much more slowly. It should be noted here that these figures do not fully capture the performance of SWCGP since the dictionaries used do not have a fast implementation. However they do provide a fair relative comparison between different values of  $\alpha$ .

### C. Evaluation on different signal processing problems

In order to compare and showcase the performance on a set of different problems often addressed with sparse approximation techniques, we choose six diverse problems from the SPARCO matlab toolbox (available at <http://www.cs.ubc.ca/labs/scl/sparco/>).

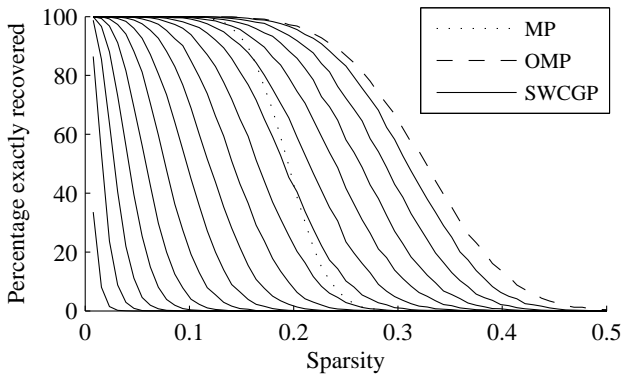


Figure 6: Comparison between Matching Pursuit (dotted), Orthogonal Matching Pursuit (dashed) and Stagewise Conjugate Gradient Pursuit (solid) in terms of exactly recovering the original coefficients. The ordinate shows the fraction of runs in which the algorithms exactly recovered the index set  $\Gamma$  used to generate the data while the abscissa shows the ratio of the size of  $\Gamma$  to the dimension of  $\mathbf{x}$ . Results averaged over 10 000 runs. The solid lines correspond to (from left to right):  $\alpha = 0.25$  to 1.0 in steps of 0.05.

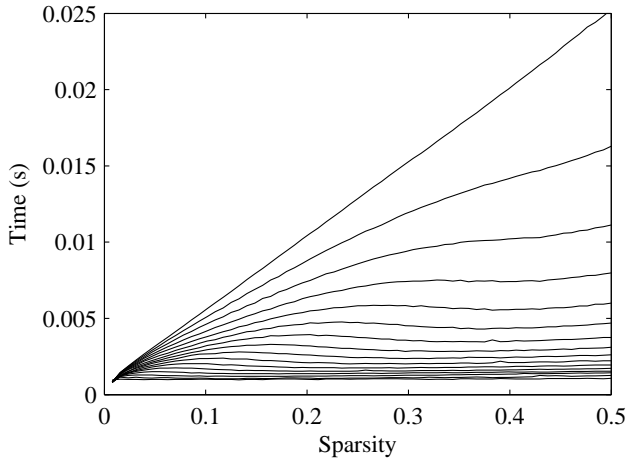
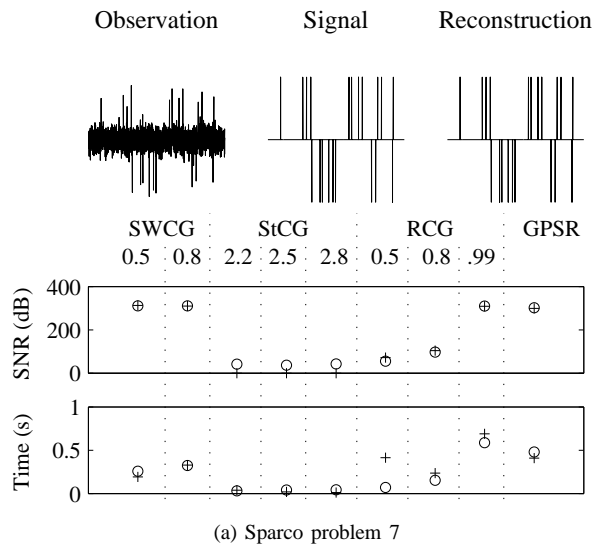


Figure 7: Comparison of the computation time for SWCGP with the different values of alpha as in figure 6. The curves correspond to (going from top to bottom):  $\alpha = 1$  to 0.25 in steps of 0.05.

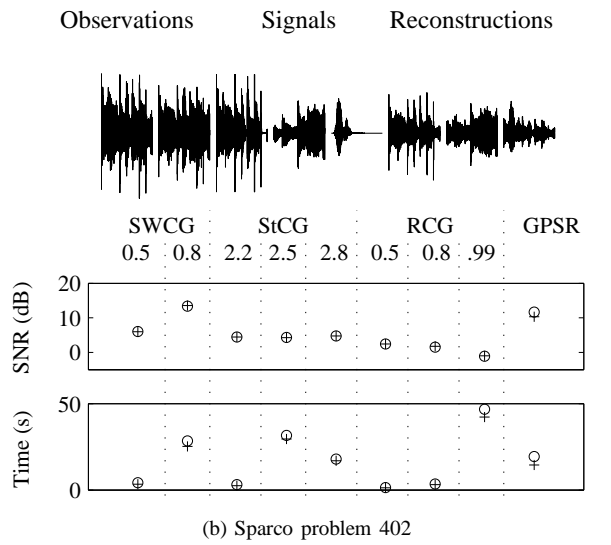
Problem 7 is a compressed sensing problem in which a signal with non-zero elements all having the same magnitude is to be recovered from a small number of Gaussian random measurements. The problem size<sup>7</sup> is  $M=600$ ,  $N=2,560$ ,  $K=198$ .

Problem 402 is a source separation problem. Three audio sources are mixed using an instantaneous mixing system to give two observations. To invert the underdetermined mixing system and separate the sources, the original audio is assumed to be sparse in a localised discrete cosine transform basis. The problem size is  $M=29,166$ ,  $N=86,016$ ,  $K=14,583$ .

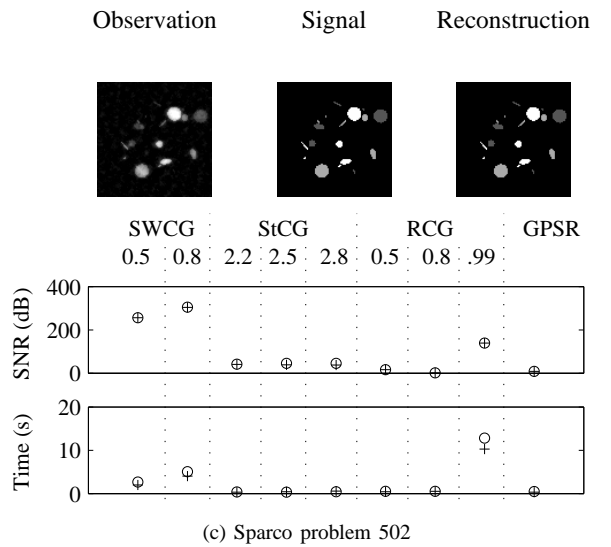
<sup>7</sup>The sparsity  $K$  is the number of non-zero elements extracted with the greedy algorithms.



(a) Sparco problem 7



(b) Sparco problem 402



(c) Sparco problem 502

Figure 8: Observed signal, original signal and reconstruction using SWCGP ( $\alpha = 0.5$ ) above SNR in dB of estimate calculated with different approaches and computation time. With (+) and without (o) normalisation of columns.

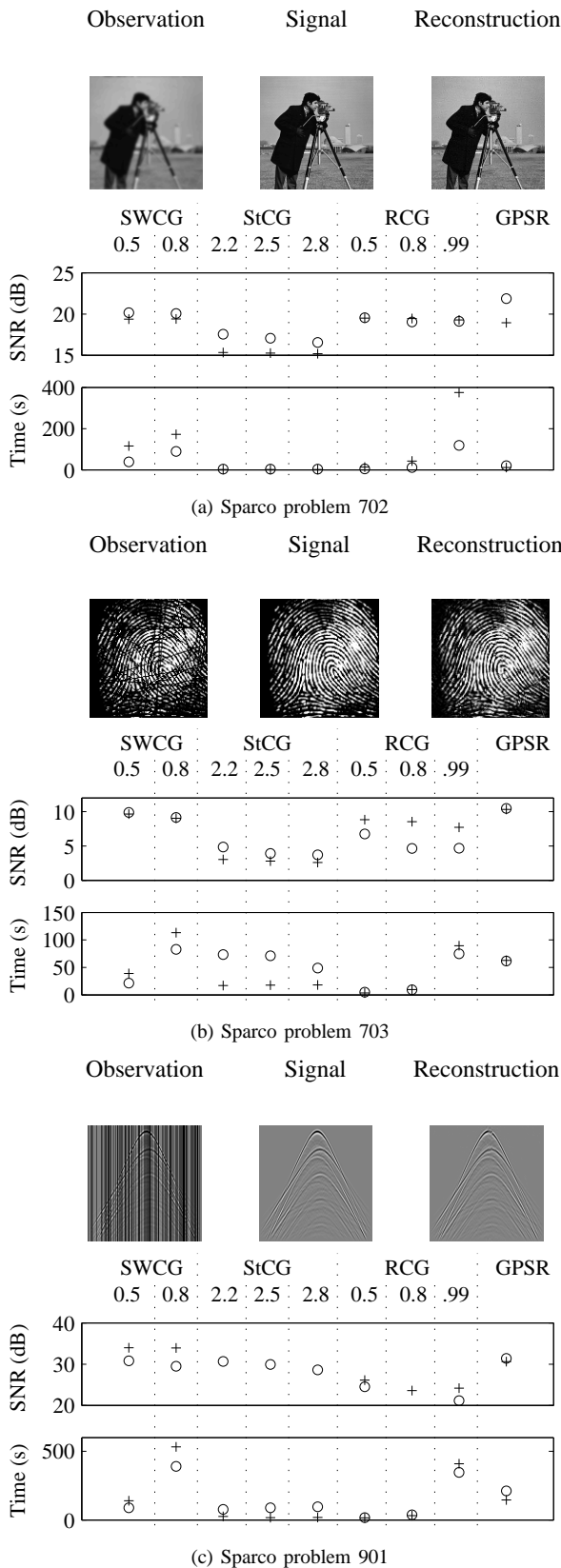


Figure 9: Observed signal, original signal and reconstruction using SWCGP ( $\alpha = 0.5$ ) above SNR in dB of estimate calculated with different approaches and computation time. With (+) and without (o) normalisation of columns.

Problem 502 is another compressed sensing problem<sup>8</sup> in which an artificial angiogram (sparse in the spatial domain) is to be reconstructed from a subset of measurements taken in the 2D Fourier domain. The problem size is  $M=10,000$ ,  $N=10,000$ ,  $K=4,000$ .

Problem 701 is an image de-blurring example. The image is assumed to be sparse in the wavelet domain. The problem size is  $M=65,536$ ,  $N=65,536$ ,  $K=9,000$ .

Problem 703 is a missing data problem in which scratches are to be removed from a fingerprint image, which is assumed to be sparse in the 2D curvelet domain. The problem size is  $M=11,013$ ,  $N=125,385$ ,  $K=5,000$ .

Problem 901 is another missing data problem in which missing traces from a seismic recording are to be recovered, again assuming sparsity in the 2D curvelet domain. The problem size is  $M=41,472$ ,  $N=480,617$ ,  $K=8,000$ .

For each problem, the observation (possibly mapped back using a linear projection into the signal space), the signal and the signal estimate calculated with the Stagewise Weak Conjugate Gradient (SW) algorithm with  $\alpha = 0.5$  are shown at the top of each panel in figures 8 and 9. Below this we show the error in estimating the signal (SNR in dB) above the computation time required by the different methods in seconds (all simulations were run in matlab on a Macintosh 2.5Ghz quad G5 computer). We here compare the Stagewise Weak (SW) CGP algorithm, ( $\alpha \in \{0.5, 0.8\}$ ), the Stagewise (St) CGP algorithm ( $t \in \{2.2, 2.5, 2.8\}$ ), the regularised (R) CGP algorithm ( $r \in \{0.5, 0.8, 0.99\}$ ) and the GPSR algorithm. The Stagewise Conjugate Gradient algorithm used the selection strategy in (4) together with the update strategy of the conjugate gradient pursuit algorithm. Similarly, regularised CGP used the selection strategy of ROMP.

For all algorithms we selected the stopping criterion as well as the regularisation parameter required in GPSR by trial and error until the observed SNR was optimal. Whilst this is not possible in practice, it allows a more or less fair comparison of the methods. All greedy algorithms used the approximate conjugate gradient update step and differed only in the element selection step. The difference in the computation time observed with these methods is therefore purely due to the different number of iterations used.

The matrices  $\Phi$  available in the SPARCO toolbox have columns of different norm. As the algorithms compared here will favour columns of  $\Phi$  that do have a larger  $\ell_2$  norm, it is in general desirable to design the measurement system with equal norm columns. Otherwise, it is often possible to pre-calculate the norm of the columns of  $\Phi$ . However, if this is also not feasible, a possibly sub-optimal approach would ignore the difference in norm. To study the influence of this normalisation, the results shown in figures 8 and 9 were therefore calculated with (+) and without (o) normalisation.

Comparing the SNR results for the different greedy strategies, it is evident that SWCGP performs consistently better than the other greedy approaches. GPSR on the other hand

<sup>8</sup>Note that GPSR did diverge on this problem (whatever the used parameters for the algorithm). The results shown here are therefore those obtained after a single iteration. The algorithm from [13] did not diverge and obtained an SNR value comparable to the SWCGP method.

can be seen to rival SWCGP in terms of SNR as well as computation time. SWCGP therefore seems to offer a competitive alternative to  $\ell_1$  based approaches that is applicable in a diverse range of settings to solve a large range of signal processing challenges.

#### D. Medical Imaging example

Compressed sensing [6] is a recent development based on sparse signal modelling ideas. One particularly promising application domain of this technique is Magnetic Resonance Imaging (MRI) [30] and we take our next example from this area using the Shepp-Logan phantom.

Acquiring MRI images is equivalent to taking one dimensional slices from the 2-dimension Fourier domain of the image. For rapid MR imaging it is desirable to take only a subset of these slices. For example one could take a reduced number of radial lines of the Fourier domain data. In order to reconstruct the original image, we utilize the fact that the image has a sparse representation in the Haar wavelet transform.<sup>9</sup> For this particular image of size  $256 \times 256$ , it was observed that the original image is well approximated (over 300 dB peak signal to noise ratio) using only 4000 of the wavelet coefficients.

The Shepp-Logan phantom, its sparse representation, and both the fully sampled measurement data and the data subsampled at approximately 15% of Nyquist (42 radial lines in the Fourier domain) are shown in figure 10. As shown in [24], this is roughly the limit for OMP to be able to fully reconstruct this image.

In [24] the performance of gradient pursuits with updates (2) and (3), as well as those of OMP and various  $L_1$  methods was reported for this problem. Here we examine the speed and performance of SWCGP for  $\alpha$  between 0.5 and 1.0. The results are presented in table I. In each case the algorithm was stopped once at least 4000 atoms were selected.<sup>10</sup> Notice that for this data it is possible to obtain an approximate speed up of 80 times using the stagewise algorithm instead of the stepwise version. Even using a relatively conservative value for  $\alpha$ , of 0.9, gave an 8 times reduction in computation time.

These improvements suggest that SWCGP should be a good candidate for tackling very large-scale problems such as those encountered in dynamic MRI imaging.

To test this, we used a subsampled version of a fully sampled MRI image sequence of a beating mouse heart to simulate rapid imaging. The sequence consisted of 8 consecutive

<sup>9</sup>It is important to note that we here use a Haar wavelet basis as our sparse representation and not a total variation based constraint as used for example in [6].

<sup>10</sup>These simulations were performed using Matlab running on a 2GHz Pentium PC.

I: Influence of  $\alpha$  on: number of iterations; approximate computation time and; PSNR performance (dB).

$\alpha$	0.5	0.6	0.7	0.8	0.9	1.0
No. of iterations	51	81	214	293	474	4087
computation time (sec.)	19.4	33.3	82	114	182	1562
PSNR (dB)	59	79	311	311	309	301

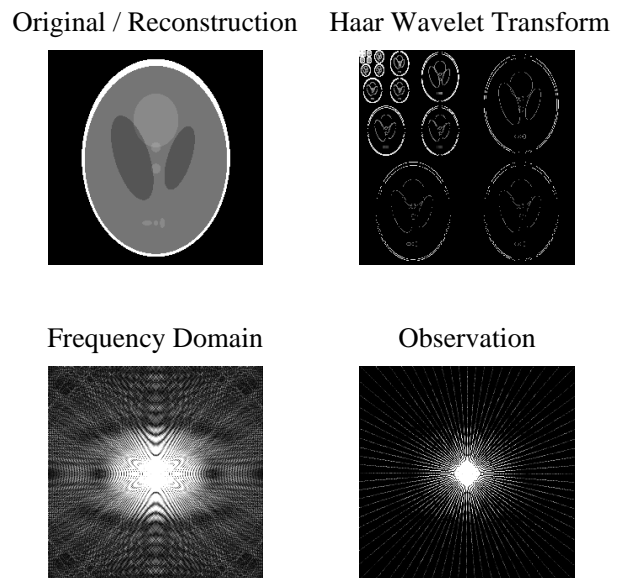


Figure 10: Magnetic Resonance Imaging (MRI) example. Original phantom image (top left), Fourier domain representation (bottom left), observation of 15% of the frequency coefficients sampled along 42 radial lines (bottom right) and sparse representation in Haar wavelet domain (top right).

$256 \times 256$  images of the heart. We here used a 3-dimensional Haar basis as the sparse representation. As in the previous example, measurements were taken using equally spaced radial lines in the spatial Fourier domain for each image. To add a degree of randomness the orientation of the lines was selected uniformly at random for each image.

Figure 11 shows a plot of the original image sequence as well as the reconstructed image sequence using SWCGP with  $\alpha = 0.7$  (stopped after 20,000 atoms were selected). The overall PSNR of the reconstruction was 31.3dB. Furthermore the reconstruction took 50 minutes (44 iterations), which is a speed up of approximately 450 times (based on iteration count) compared with the stepwise algorithm!

## VI. DISCUSSION AND CONCLUSION

Underdetermined inverse problems with a sparsity constraint on the solution are found in many areas of modern signal processing. In this paper, we have introduced a greedy strategy, that in each iteration selects several new elements. The coefficients are then updated using a directional optimisation step in which the update direction is conjugate to the previous update direction. This procedure addressed two issues arising when Orthogonal Matching Pursuit is applied to large scale problems. On the one hand, the use of directional optimisation reduces the computational cost of each iteration. On the other hand, picking several elements in each iteration, reduces the overall number of iterations required.

The use of the conjugate update direction was introduced in an earlier paper [24]. The main focus in this paper was therefore on the selection strategy. We have discussed the prior art in this respect and highlighted several disadvantages of current approaches. To overcome these, we presented a

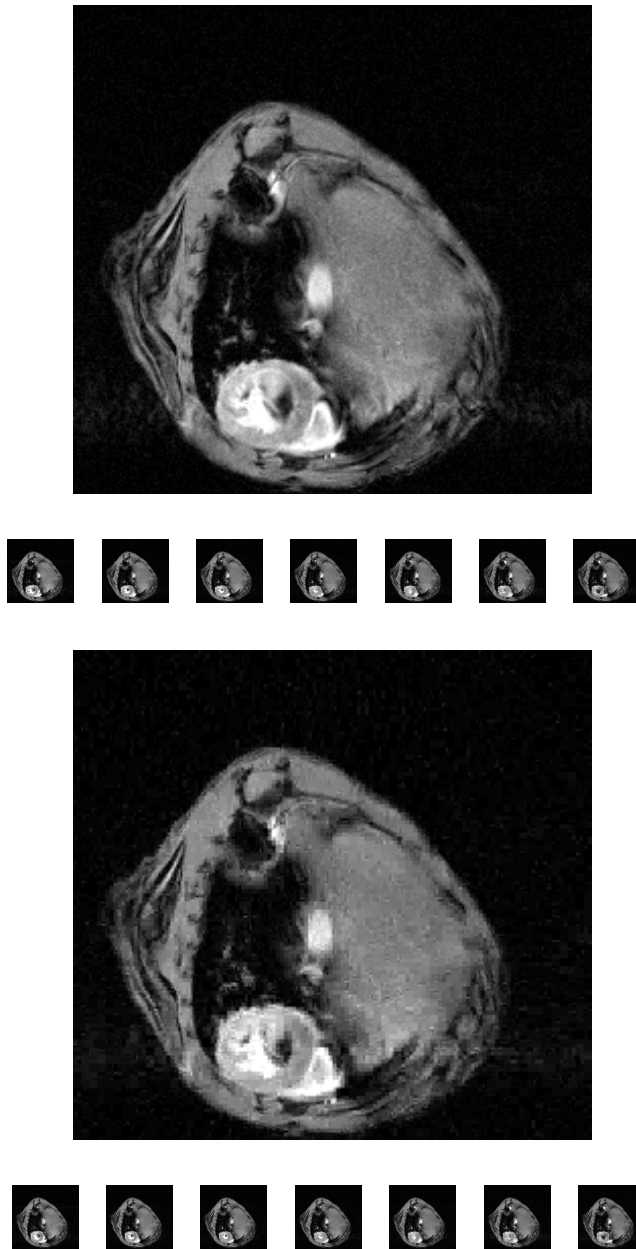


Figure 11: Dynamic MRI example. The original image sequence (top) and the image sequence reconstructed from 20% of the measurement data (bottom) using SWCGP with 44 iterations and  $\alpha = 0.7$ .

Stagewise Weak selection strategy. Using this strategy in OMP does not necessarily offer computational advantages, which were achieved only when combining the stagewise weak selection and the conjugate gradient update.

In this paper we presented a range of numerical experiments. Synthetic data highlighted several properties of the method and its good performance. In particular, the weakness parameter allowed a smooth trade-off between the sparsity  $K/M$  and computational complexity of the recovery problem. The application to a wide range of real world problems has shown that the approach is competitive with other state of the art approaches based on  $\ell_1$  minimisation, both in terms of speed and performance. Finally, the application to dynamic MRI

imaging demonstrated the applicability to very large data-sets.

A study of theoretical properties of the proposed approach can be found in the companion paper [2].

## REFERENCES

- [1] M. E. Davies and T. Blumensath, "Faster & greedier: algorithms for sparse reconstruction of large datasets," in *Proc. of the third IEEE International Symposium on Communications, Control, and Signal Processing*, (St Julians, Malta), March 2008.
- [2] T. Blumensath and M. E. Davies, "Stagewise weak gradient pursuits. Part II: Theoretical properties," *submitted for publication*, 2008.
- [3] V. K. Goyal, M. Vetterli, and N. T. Thao, "Quantized overcomplete expansions in  $R^N$ : Analysis, synthesis and algorithms," *IEEE Transactions on Information Theory*, vol. 44, pp. 16–31, Jan. 1998.
- [4] C. Fevotte and S. Godsill, "Sparse linear regression in unions of bases via bayesian variable selection," *IEEE Signal Processing Letters*, vol. 13, no. 7, pp. 441–444, 2006.
- [5] 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.
- [6] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, pp. 489–509, Feb 2006.
- [7] G. Davis, *Adaptive Nonlinear Approximations*. PhD thesis, New York University, 1994.
- [8] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, pp. 227–234, Apr 1995.
- [9] 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.
- [10] B. Efron, T. Hastie, I. Johnston, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.
- [11] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "A method for large-scale  $l_1$ -regularised least squares problems with applications in signal processing and statistics," *submitted*, 2007.
- [12] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *submitted manuscript*, 2007.
- [13] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," Tech. Rep. TR-200801, Department of Computer Science, University of British Columbia, 2008.
- [14] 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.
- [15] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted  $l_1$  minimization," tech. rep., California Institute of Technology, 2007.
- [16] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
- [17] D. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2153–2164, 2004.
- [18] R. E. McCulloch and E. I. George, "Approaches for Bayesian variable selection," *Statistica Sinica*, vol. 7, no. 2, pp. 339–374, 1997.
- [19] R. E. McCulloch and E. I. George, "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, pp. 881–889., September 1993.
- [20] J. Geweke, "Variable selection and model comparison in regression," in *Bayesian Statistics 5*. (J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, eds.), Oxford University Press, 1996.
- [21] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [22] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *27<sup>th</sup> Asilomar Conf. on Signals, Systems and Comput.*, Nov. 1993.
- [23] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification.," *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, 1989.
- [24] T. Blumensath and M. Davies, "Gradient pursuits," *IEEE Transactions on Signal Processing*, vol. 56, pp. 2370–2382, June 2008.

- [25] D. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solutions of underdetermined linear equations by stagewise orthogonal matching pursuit," 2006.
- [26] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *submitted*, 2007.
- [27] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *submitted*, 2008.
- [28] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
- [29] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [30] M. Lustig, D. L. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *submitted*, 2007.