

FAST GREEDY ALGORITHMS FOR LARGE SPARSE INVERSE PROBLEMS

Thomas Blumensath and Mike E. Davies

IDCOM & Joint Research Institute for Signal and Image Processing
The University of Edinburgh, King's Buildings, Mayfield Road, Edinburgh, EH9 3JL, UK

ABSTRACT

In this paper we introduce and study algorithms to solve *large* linear equations, in which the solution is assumed to be sparse. We concentrate on greedy strategies and in particular study extensions to the recently introduced Gradient Pursuit framework. Different approaches are compared that select several elements per iteration, thereby significantly decreasing the computational complexity of the method. We argue in particular for the use of a so called stage-wise weak selection strategy. Experiments with a range of sparse inverse problems show that a Gradient Pursuit algorithm with such a selection strategy is competitive to and in many cases outperforms other fast algorithms, both in terms of estimation accuracy and in terms of computation speed.

1. INTRODUCTION

Given a vector $\mathbf{x} \in \mathbb{R}^M$ and a matrix $\Phi \in \mathbb{R}^{M \times N}$ ($N > M$), whose columns, denoted by ϕ_i , are assumed to have unit length, the problem addressed in this paper is to estimate a *sparse* vector $\hat{\mathbf{y}} \in \mathbb{R}^N$ and approximate \mathbf{x} as

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

The approximation error is denoted as $\hat{\mathbf{x}} - \mathbf{x} = \mathbf{e}$. One typically tries to find the sparsest vector $\hat{\mathbf{y}}$, subject to a constraint on the ℓ_2 norm of \mathbf{e} , however, alternative formulations place the constraint on the sparsity and optimise the ℓ_2 error or use a regularised cost.

Such sparse signal models are of interest in several areas of signal processing. One can roughly split the applications into two main categories, sparse signal *approximation* and sparse signal *estimation*, depending on whether \mathbf{x} or \mathbf{y} is the actual signal of interest. In the first case, the focus is on modelling a signal \mathbf{x} , with the error of interest being $\mathbf{x} - \hat{\mathbf{x}}$, while in the second case, the focus is on *estimating* a vector \mathbf{y} with the error defined in the coefficient domain, that is the error of interest is $\mathbf{y} - \hat{\mathbf{y}}$. Applications falling into the first category include, for example, source coding [1], [2], de-noising [3] and pattern analysis [4], while the second category includes, for example, source separation [5] and compressed sensing [6, 7].

Unfortunately, the optimisation problem as set out above is known to be NP hard in general [8, 9] and sub-optimal¹ strategies have to be used. Whilst there is now an abundance of different approaches available, two classes of methods are currently among the most popular. The first approach replaces the sparsity measure to be optimised with the 'convex' ℓ_1 penalty [12]. The other approach is to use a greedy strategy, such as Matching Pursuit (MP) or Orthogonal Matching Pursuit (OMP) [13]. The latter approach will be adopted here.

2. ORTHOGONAL MATCHING PURSUIT

The algorithms in this paper are fast variants of Orthogonal Matching Pursuit (OMP), which is an iterative algorithm that in each iteration approximates \mathbf{x} using a subset of columns from Φ . These

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.

¹Note, however, that under certain conditions, linear programming methods and greedy algorithms are guaranteed to recover the optimum [10, 11].

sub-matrixes will be denoted by Φ_Γ , where Γ is a set of indices that specify which columns of the matrix Φ are retained. A similar notation is used throughout to denote sub-vectors of \mathbf{y} . Therefore, in iteration n , we have an estimate $\mathbf{y}_{\Gamma^{[n]}}^{[n]}$. The algorithm also keeps track of the residual $\mathbf{r}^{[n]} = \mathbf{x} - \Phi_{\Gamma^{[n]}} \mathbf{y}_{\Gamma^{[n]}}^{[n]}$.

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

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

The selection step is based on the inner products between the column vectors of Φ and the current residual. The vector of inner products is calculated as

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

The selection step then searches through the vector $\mathbf{g}^{[n]}$ and selects the *single* element with the largest magnitude.

$$i^{[n]} = \arg_i \max |g_i^{[n]}|,$$

which is added to the set of selected elements

$$\Gamma^{[n]} = \Gamma^{[n-1]} \cup i^{[n]}.$$

Note that an exact implementation of OMP will not select elements more than once. This is due to the orthogonal projection used in the coefficient update (see below), which 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 [14] or in Gradient Pursuit [15], it is advisable in general to allow the algorithm to re-select elements.

Once the set of elements $\Gamma^{[n]}$ has been updated, the coefficients $\mathbf{y}^{[n]}$ have to be estimated. In OMP, this is done by solving

$$\mathbf{y}_{\Gamma^{[n]}}^{[n]} = \arg \min_{\mathbf{y}_{\Gamma^{[n]}}} \|\mathbf{x} - \Phi_{\Gamma^{[n]}} \mathbf{y}_{\Gamma^{[n]}}\|_2^2,$$

i.e., $\mathbf{y}_{\Gamma^{[n]}}^{[n]} = \Phi_{\Gamma^{[n]}}^\dagger \mathbf{x}$ where $\Phi_{\Gamma^{[n]}}^\dagger$ is the pseudo inverse of $\Phi_{\Gamma^{[n]}}$. However, the explicit calculation of the pseudo inverse in each iteration is only feasible for relatively small problems and fast implementations keep track of a QR or Cholesky factorisation, which is updated from iteration to iteration. More details on this can be found in, for example, [15].

The final step updates the residual $\mathbf{r}^{[n]} = \mathbf{x} - \Phi_{\Gamma^{[n]}} \mathbf{y}_{\Gamma^{[n]}}^{[n]}$. Depending on the detailed implementation of the algorithm, faster recursions are available for this update using only vector addition and scalar multiplication.

For very large problems, the calculation of the inner products in equation (1) can be costly. Therefore, in many practical applications, Φ is designed to have additional structure so that the inner products can be evaluated efficiently. In this case, Φ does not have to be stored explicitly. The main bottleneck of the algorithm is then the update of $\mathbf{y}^{[n]}$. Even if fast methods are used, this requires several matrix vector products as well as the storage of a QR or Cholesky factorisation, which can be costly in terms of storage requirement. To overcome this limitation, we have previously argued to approximate the orthogonal projection using directional optimisation, which can be done much more efficiently [15].

3. GRADIENT PURSUIT: APPROXIMATING THE ORTHOGONALISATION STEP

Gradient Pursuit, introduced in [15], replaces the exact orthogonalisation in OMP by a directional update step

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

Different directions $\mathbf{d}^{[n]}$ can be used. In [15], the gradient and an approximate conjugate gradient method were suggested. Importantly, we have recently found an additional recursion [16], which allows the approximate conjugate gradient to be calculated recursively, so that the evaluation of the gradient and the approximate conjugate gradient have roughly equal computation costs. Crucially, extensive experimental evaluations have shown that the approximate conjugate direction based approach outperforms the gradient based method. We therefore endorse the use of the approximate conjugate gradient approach, which for simplicity will be called Conjugate Gradient Pursuit (CGP) throughout this paper.

The conjugate gradient is calculated as a combination of the current gradient and the previous update direction and is chosen such that the current update direction $\mathbf{d}^{[n]}$ is *conjugate*² to the previous update direction $\mathbf{d}^{[n-1]}$.

As mentioned above, if using this strategy, the residual $\mathbf{r}^{[n]}$ is no longer guaranteed to be orthogonal to the selected columns of Φ and it is important to let the algorithm re-select elements from Φ .

The performance of this directional update step was shown in [15] to offer significant advantages in terms of computational requirements, while on the other hand, leading to only minor decreases in performance when compared to the original OMP approach.

It should also be noted that this approach is fundamentally different from a strategy that uses a conjugate gradient solver in each iteration of OMP as is done, for example, in StOMP [14]. While our conjugate gradient method tries to retain conjugacy between OMP iterations, the method used in StOMP starts a new conjugate gradient optimisation in each iteration.

4. STAGewise SELECTION: REPLACING THE SELECTION STRATEGY

Having thus substantially increased the computation speed of each iteration, we now turn to a reduction in the total number of iterations required. In OMP and standard Gradient Pursuit, only a single element is selected in each iteration, so that the algorithm has to be run for at least as many iterations as there are non-zero elements to be estimated. This can only be avoided in a greedy algorithm by selecting more than a single element in each iteration. So far two strategies have been put forward that select several elements in each OMP step. Stagewise OMP (StOMP), proposed in [14], calculates a threshold

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

and then selects all elements, whose inner products have a magnitude larger than this threshold

$$\Gamma^{[n]} = \Gamma^{[n-1]} \cup \{i : |g_i| \geq \lambda\}.$$

²Conjugate here means that $(\mathbf{d}^{[n]})^T \Phi_{\Gamma^{[n]}}^T \Phi_{\Gamma^{[n]}} \mathbf{d}^{[n-1]} = \mathbf{0}$.

A different strategy, which groups the g_i into sets J_k , such that the elements in each set have a similar magnitude, i.e. they satisfy

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

is called ROMP, where the R stands for regularised. This algorithm selects the set J_k for which $\|\mathbf{y}_{J_k}\|_2$ is largest. This approach has been introduced in [17] in which $r = 0.5$ was used.

While ROMP has some nice theoretical properties, our own practical experiments, some of which can be found below, have found the approach not to work as well as StOMP. StOMP on the other hand was developed explicitly for problems in which Φ has been generated from random entries. Furthermore, 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. Another problem with using the residual to define a threshold is that the algorithm might get ‘stuck’ when all inner products fall below the threshold. In our experience this approach has shown mixed results.

We propose the use of yet another selection criterion, similar to StOMP, but based on ideas from *weak Matching Pursuit*. Weak Matching Pursuit is a method developed for large or infinite dimensional problems in which not all inner products can be evaluated explicitly. The selection criterion in this approach therefore introduces a weakness parameter α and selects *any one* element such that

$$|g_j| \geq \alpha \max_i |g_i|. \quad (2)$$

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. We call such a strategy *stagewise weak* selection. The set of selected elements is defined formally as

$$\Gamma^{[n]} = \Gamma^{[n-1]} \cup \{i : |g_i| \geq \alpha \max_j |g_j|\}, \quad (3)$$

i.e. we select all elements that come within a factor of α of the largest inner product.

Conditions guaranteeing that weak Matching Pursuit will select a correct element must also carry over if all elements are selected that satisfy the above condition. This is explored in a companion paper [16].

Whilst it is possible to use all of the fast selection methods in conjunction with the exact orthogonalisation as used in OMP, it should be noted that, when using efficient implementations of this orthogonalisation based on QR or Cholesky factorisation, the advantages of selecting several elements at once are minimal. This is because updating the factorisations itself has to be done for *any* newly selected element. As this update dominates the computation cost, there is not much difference in the computation time between these fast selection methods and original OMP. The real advantage in terms of computation speed offered by these fast selection methods therefore only appears in conjunction with fast approximate orthogonalisation methods as those used in Gradient Pursuit or StOMP [14].

4.1 Greed or relaxation

In this paper we have derived a new fast variant to OMP. Another alternative to solve large linear inverse problems with a sparsity constraint are methods optimising an ℓ_1 regularised cost function such as, for example, the GPSR algorithm [18].

All of these algorithms require the selection of parameters and stopping criteria. For example, the GPSR algorithm requires the specification of a regularisation parameter. Selecting this parameter is not trivial in general and often requires the application of the method with a range of different settings. The greedy approaches in this paper all came equipped with a single parameter. While the regularised approach and our stagewise weak approach are akin in

practice to OMP whenever the parameter is set to 1, both algorithms behaved quite differently when changing the parameter. In particular, we found the performance of ROMP to deteriorate very rapidly when decreasing r . On the other hand the stagewise weak approach allows a more controlled trade-off between available signal sparsity and computation performance (see also [16]). The StOMP selection strategy has shown a more complicated relationship between performance and the choice of the parameter.

In selecting the stopping criterion, greedy methods are fundamentally different to methods based on the optimisation of a convex cost function, such as the GPSR approach. While greedy methods select new elements iteratively, it is possible to monitor the sparsity and approximation error until an acceptable compromise is reached. This trade-off is specified in a non-trivial manner by the regularisation parameter in the GPSR algorithm. The stopping criterion is then only concerned with the closeness of the solution to the minimum of the used cost function. Hennenfent and Herrmann [20] have therefore proposed an annealing approach to the choice of the regularisation parameter, however, this approach requires the specification of an annealing schedule, which in itself might be non-trivial.

4.2 Software

The weak stagewise selection step is now also available in the implementation of the approximate conjugate gradient algorithm of the sparsify matlab toolbox, available from the first authors webpage.

5. EXPERIMENTAL EVALUATION

5.1 Comparison of fast selection strategies

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. Whilst this combination does not offer any 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 use of the methods in conjunction with the conjugate gradient pursuit update is studied in the next subsection.

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 $\mathbf{x} = \Phi\mathbf{y} + \mathbf{e}$ where \mathbf{e} 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 observation SNR. All results are averaged over 10 000 problem realisations.

We first compared the performance of OMP, StOMP, ROMP and our stagewise weak OMP algorithm (SWOMP). For comparison with non-greedy algorithms, we also run the GPSR algorithm (with de-biasing) [18], which optimises the cost function $\|\mathbf{x} - \Phi\mathbf{y}\|_2^2 + \lambda\|\mathbf{y}\|_1$ (followed by an orthogonal projection onto the non-zero elements) as well as the slower LASSO algorithm implemented in the SparseLab toolbox (available at <http://sparselab.stanford.edu/>).

The greedy algorithms OMP, StOMP, ROMP and SWOMP, were run until they had selected twice as many elements as were used to generate the observations. For StOMP, ROMP and SWOMP, we used the parameters $t = 2.5$, $r = 0.5$ and $\alpha = 0.7$. In GPSR we set $\lambda = 0.001 * K$. The LASSO algorithm was stopped once the residual error had the same ℓ_2 norm as the noise \mathbf{e} .

We then evaluated whether the algorithms identified the exact location of the non-zero elements in the coefficient vector used to generate the signal. As we allowed the greedy algorithms to select twice³ as many elements as were used to generate the signal,

we classified the coefficients to be exactly recovered whenever the largest (in magnitude) K coefficients in the estimate $\hat{\mathbf{y}}$ coincided with the locations of the non-zero elements in the true coefficient vector \mathbf{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 did classify the results as not exactly recovered. The results are shown in figure 1 for SNR values of ∞ dB, 120dB, 60dB, 30dB, and 15dB. The abscissa shows the ratio between the number of non-zero elements K used to generate the signal and the observation dimension M .

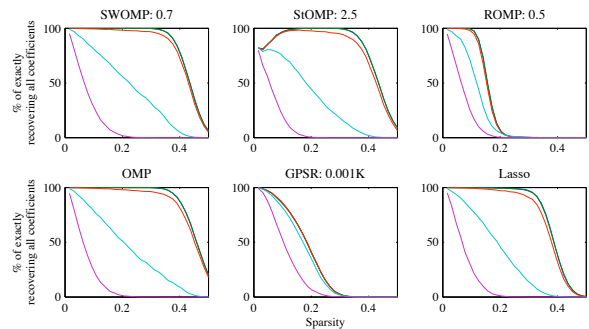


Figure 1: Comparison between SWOMP, StOMP, ROMP, OMP, GPSR and Lasso for different observation SNR values (from top right to bottom left in performance SNR = ∞ dB, 120dB, 60dB, 30dB, 15dB). The abscissa shows the ratio between non-zero elements K and the observation dimension M . All results averaged over 10 000 realisations.

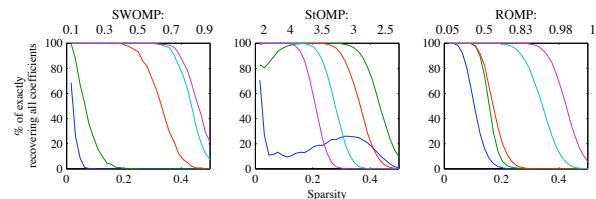


Figure 2: Comparison between SWOMP, StOMP, ROMP for different parameters. The parameters used are shown above each panel. Note that (in particular for StOMP) the parameters are shown in that order in which the associated graph appears in the panel. The abscissa shows the ratio between non-zero elements K and the observation dimension M . All results averaged over 10 000 realisations.

In a second set of experiments we evaluated the influence of the parameters t , r and α on the performance of the fast selection strategies. It should be noted that in the development of ROMP in [17], r was assumed to be 0.5. However, this parameter can be set to a more general value between 0 and 1. The results, again in terms of exact recovery, are shown in figure 2. Note that in this figure, the 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, which was due to the algorithm stopping often before it had selected K elements.

almost surely and 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 [17].

³Note that in the noiseless case and for random Φ , 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

5.2 Fast Gradient Pursuits

In this subsection we analyse the performance of the different fast selection strategies in conjunction with the approximate conjugate gradient update step. We here used the default problems 7, 402, 502, 701, 703 and 901 from the SPARCO matlab toolbox (available at <http://www.cs.ubc.ca/labs/scl/sparco/>) to compare the different approaches. Due to space constraints, the reader is referred to [19] for a detailed description of these problems. Each of these problems defines a signal s , a synthesis mapping \mathbf{D} in which this signal is assumed to be sparse and an observation mapping \mathbf{M} , that maps the signal to an observation \mathbf{x} . In our notation, we have $\Phi = \mathbf{MD}$. Assume we are only given \mathbf{x} , \mathbf{D} and \mathbf{M} . Each of the problems can then be solved by finding a sparse vector $\hat{\mathbf{y}}$, such that $\mathbf{x} \approx \Phi \hat{\mathbf{y}}$. The signal s is then estimated as $\hat{s} = \mathbf{D} \hat{\mathbf{y}}$. The signal s , the observation \mathbf{x} (possibly mapped back using a linear projection into the signal space) and the signal estimate $\mathbf{D} \hat{\mathbf{y}}$ (where $\hat{\mathbf{y}}$ was calculated with the Stagewise Weak Conjugate Gradient (SW) algorithm with $\alpha = 0.5$) are shown at the top of figures 3 to 4. Below this we show the error in estimating s (SNR in dB) above the computation time required by the different methods in seconds (all simulations were programmed in matlab and run on a Macintosh 2.5Ghz quad G5 computer). We here compare the Stagewise Weak (SW) algorithm, ($\alpha \in \{0.5, 0.8\}$), the Stagewise (St) algorithm ($t \in \{2.2, 2.5, 2.8\}$), the regularised (R) algorithm ($r \in \{0.5, 0.8, 0.99\}$) and the GPSR⁴ algorithm. 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.

If the columns in Φ are not of the same length, greedy algorithms as well as ℓ_1 based methods will favour columns of Φ that do have a larger ℓ_2 norm. If the measurement system cannot be designed with equal norm columns, it is often possible to pre-calculate the norm of the columns of Φ . However, if this is not feasible, a possibly sub-optimal approach would ignore the difference in norm. The matrixes Φ available in the SPARCO toolbox *do* have columns of different norm in general. The results shown in figures 3 to 4 were 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 approaches. RCGP comes close to the SNR performance of SWCGP only in problem 7, but here RCGP (using $r = 0.99$) requires more time. StCGP is competitive to SWCGP only on problem 901 and possibly problem 402. GPSR on the other hand was found to often rival with SWCGP in terms of SNR (apart from problem 502) as well as computation time. It therefore seems that the choice between greedy and relaxed methods is highly problem dependent. Nevertheless, our SWCGP approach gave consistently good results on all problems.

6. CONCLUSIONS

Many problems in signal processing, from compressed sensing to source coding, require the solution to underdetermined inverse problems under a sparsity constraint. The ever increasing size of such problems in many real world applications makes the development of computationally efficient algorithms paramount.

In this paper we have proposed the use of a stagewise weak greedy element selection strategy together with a previously introduced approximate conjugate direction optimisation step. Theoretic justification for this approach has been given in a companion paper [16] and we have here concentrated on empirical evidence, both in favour of the stagewise weak element selection strategy per se and

⁴It should be mentioned that GPSR did diverge on problem 502 (whatever the used parameters for the algorithm). The results shown here are therefore those obtained after a single iteration.

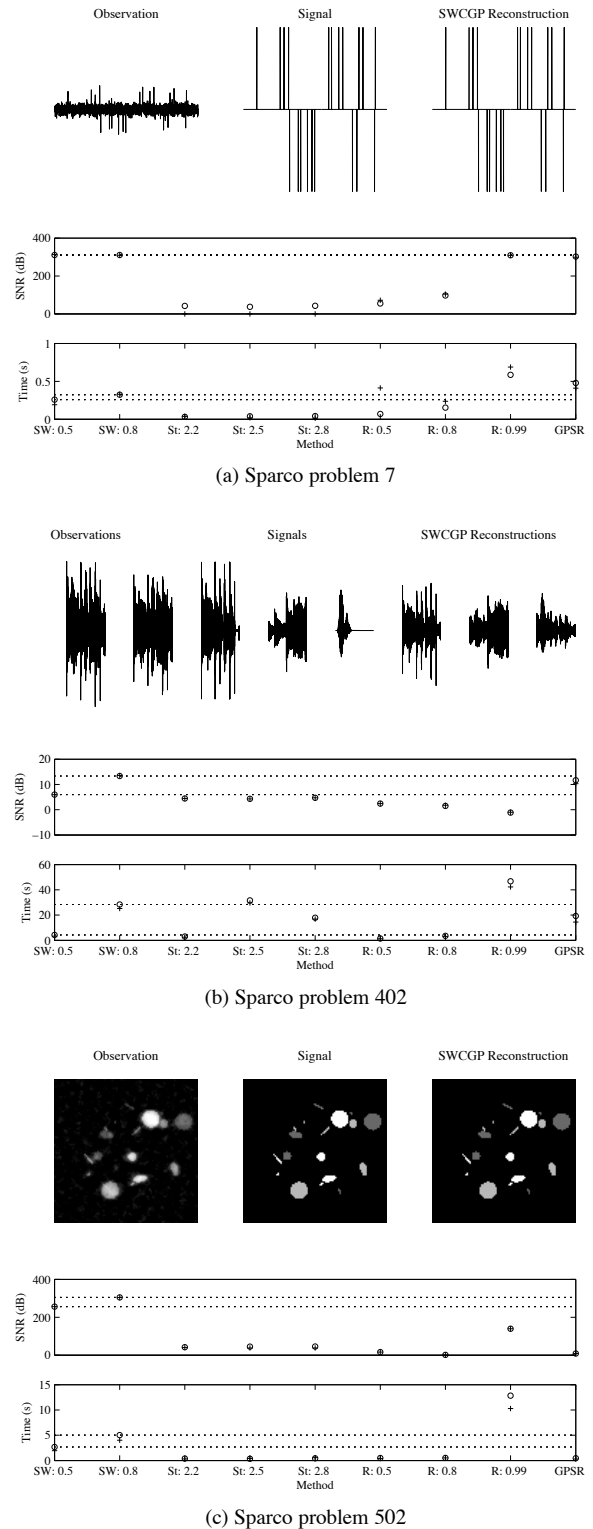
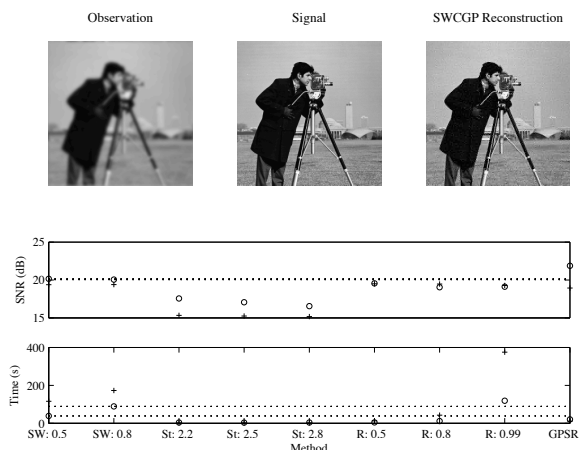
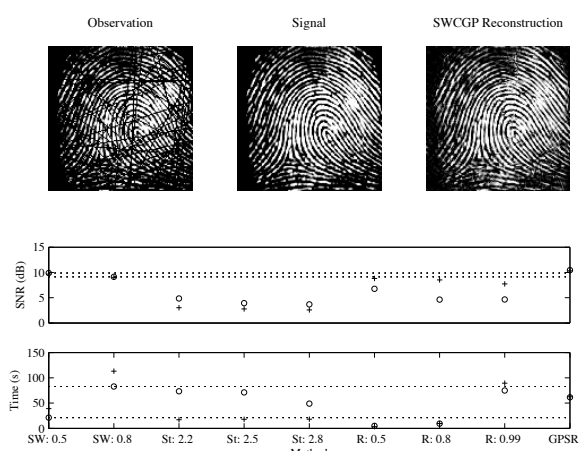


Figure 3: Observed signal, original signal and reconstruction using SWCGP ($\alpha = 0.5$) above SNR in dB of estimate calculated with different approaches and computation time. With (+) and without (o) normalisation of columns.

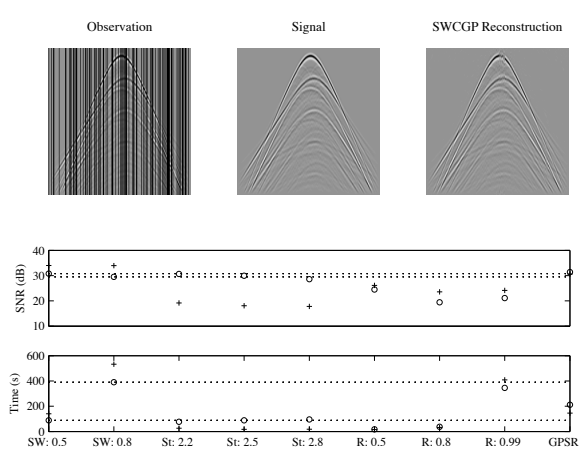
for its combination with the fast directional optimisation approach. The results show that the approach is not only competitive on a range of problems, in many cases the method outperformed other



(a) Sparco problem 701



(b) Sparco problem 703



(c) Sparco problem 901

Figure 4: 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.

fast algorithms, such as the GPSR algorithm (both, in terms of estimation accuracy and in terms of computation time).

REFERENCES

- [1] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Trans. 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 Trans. on Signal Processing*, vol. 46, pp. 1027–1042, Apr. 1998.
- [3] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.
- [4] T. Blumensath and M. Davies, "Sparse and shift-invariant representations of music," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, pp. 50–57, Jan 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, "Compressive sampling," in *Proceedings of the International Congress of Mathematics*, (Madrid, Spain), 2006.
- [7] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [8] G. Davis, *Adaptive Nonlinear Approximations*. PhD thesis, New York University, 1994.
- [9] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, pp. 227–234, Apr 1995.
- [10] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution," *Comm. Pure and Applied Math.*, vol. 59, no. 6, pp. 797–829, 2006.
- [11] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [12] 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.
- [13] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [14] D. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solutions of underdetermined linear equations by stagewise orthogonal matching pursuit," Tech. Rep. 2006-02, Department of Statistics, Stanford University, 2006.
- [15] T. Blumensath and M. Davies, "Gradient pursuits," to appear in *IEEE Trans. on Signal Processing*, 2008.
- [16] M. E. Davies and T. Blumensath, "Faster & greedier: algorithms for sparse reconstruction of large datasets," in *Proceedings of the third International Symposium on Communications, Control and Signal Processing (ISCCSP)*, (Malta), March 2008.
- [17] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *submitted*, 2007.
- [18] 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.
- [19] E. v. Berg, M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab, and O. Yılmaz, "Sparco: A testing framework for sparse reconstruction," Tech. Rep. TR-2007-20, Dept. Computer Science, University of British Columbia, Vancouver, October 2007.
- [20] Hennenfent and F. J. Herrmann, "Application of stable signal recovery to seismic interpolation," in *SEG International Exposition and 76th Annual Meeting*, 2006.