GRADIENT PURSUIT FOR NON-LINEAR SPARSE SIGNAL MODELLING

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 the linear sparse signal model is extended to allow more general, non-linear relationships and more general measures of approximation error. A greedy gradient based strategy is presented to estimate the sparse coefficients. This algorithm can be understood as a generalisation of the recently introduced Gradient Pursuit framework. Using the presented approach with the traditional linear model but with a different cost function is shown to outperform OMP in terms of recovery of the original sparse coefficients. A second set of experiments then shows that for the non-linear model studied and for highly sparse signals, recovery is still possible in at least a percentage of cases.

1. INTRODUCTION

A linear sparse signal model approximates an observation $\mathbf{x}$ from a vector or Hilbert space using a small number of elements selected from a set $\{\phi_i\}$ of elements from the same space. With each of the ϕ_i is associated a coefficient y_i. An estimate of $\mathbf{x}$ can then be written as

$$\hat{\mathbf{x}} = \sum_i \phi_i \hat{y}_i = \mathbf{\Phi}\hat{\mathbf{y}},$$

where $\mathbf{\Phi}$ can be thought of as a matrix with columns ϕ_i and where $\hat{\mathbf{y}}$ is the vector of coefficients $\hat{y}_i$. We further write the approximation error as $\hat{\mathbf{x}} - \mathbf{x} = \mathbf{e}$.

In the rest of this paper we will restrict the discussion to finite dimensional vector spaces, as these are the ones encountered in most applications. In particular, assume $\mathbf{e}, \mathbf{x}, \hat{\mathbf{x}} \in \mathbb{C}^M$ and $\hat{\mathbf{y}} \in \mathbb{C}^N$. If $N > M$, the estimation of $\mathbf{y}$ is underdetermined and additional constraints have to be used. Sparse models address this issue by the use of a sparsity measure. Furthermore, one also often allows for a non-zero error $\mathbf{e}$ to account for observation noise or model misfit. A general optimisation problem, using $C_\mathbf{e}$ and $C_\mathbf{y}$ to denote the measures used to quantify the approximation error and the coefficient sparsity, is then[1]

$$\mathbf{y}^{opt} = \min_{\hat{\mathbf{y}}} C_\mathbf{e}(\mathbf{x} - \mathbf{\Phi}\hat{\mathbf{y}}) + \lambda C_\mathbf{y}(\hat{\mathbf{y}}). \tag{1}$$

1.1 Applications

Sparse signal models have a range of different applications in signal processing. These can be roughly grouped into two categories, sparse *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 error defined in the coefficient domain, i.e. the error of interest is $\mathbf{y} - \hat{\mathbf{y}}$. Applications falling into the first category include, for example, source coding [1], [2], denoising [3] and pattern analysis [4], while the second category includes, for example, source separation [5] and compressed sensing [6, 7].

1.2 Algorithmic strategies

Whilst different measures of sparsity are available, one often counts the number of non-zero elements in $\mathbf{y}$. Even though this is not a norm, it is typical to denote this 'ℓ_0' measure using the notation $\|\mathbf{y}\|_0$. Also, in many applications, $C_\mathbf{e}$ is often the mean squared error. Using these cost functions, the optimisation problem as set out in (1) is known to be NP hard in general [8, 9] and sub-optimal[2] strategies have to be used. Whilst there are many possible strategies to be followed, a common method is to relax the ℓ_0 penalty and to use the 'convex' ℓ_1 penalty instead [12]. Another approach is to use a greedy strategy, such as Matching Pursuit (MP) or Orthogonal Matching Pursuit (OMP) [13].

The algorithm in this paper can be understood as a generalisation of the OMP paradigm. We therefore review OMP here in a bit more detail. Both, OMP and MP, are iterative algorithms, that, in each iteration, select a single new element from $\{\phi_i\}$ and then update the coefficients $\hat{\mathbf{y}}$ appropriately. If $\mathbf{r}^{[n-1]} = \mathbf{x} - \mathbf{\Phi}\hat{\mathbf{y}}^{[n-1]}$ is the approximation error at the beginning of iteration n, then the selection criterion is

$$i^{[n]} = \arg_i \max |\phi_i^H \mathbf{r}^{[n-1]}|.$$

To motivate the choice of the selection criterion in our algorithm developed below, we note that $\mathbf{\Phi}^H \mathbf{r}^{[n-1]}$ is the negative gradient of $\|\mathbf{x} - \mathbf{\Phi}\hat{\mathbf{y}}\|_2^2$ evaluated at the current coefficient estimate $\hat{\mathbf{y}}^{[n-1]}$ (See for example [14]).

Both, MP and OMP use the same selection strategy, but differ in the calculation of $\hat{\mathbf{y}}^{[n]}$. OMP estimates $\hat{\mathbf{y}}^{[n]}$ by minimising $\|\mathbf{x} - \mathbf{\Phi}\hat{\mathbf{y}}\|_2^2$ under the restriction that only those values in $\hat{\mathbf{y}}$ are allowed to be non-zero that have been selected by the algorithm up to this point. Let this set be denoted by $\Gamma^{[n]}$ and let $\Gamma^{[n]} = \Gamma^{[n-1]} \cup i^{[n]}$. If $\mathbf{\Phi}_{\Gamma^{[n]}}$ is the matrix $\mathbf{\Phi}$ with all columns removed apart from those indexed by $\Gamma^{[n]}$ and if $\hat{\mathbf{y}}_{\Gamma^{[n]}}$ is defined similarly, then

$$\hat{\mathbf{y}}_{\Gamma^{[n]}}^{[n]} = \mathbf{\Phi}_{\Gamma^{[n]}}^{\dagger} \mathbf{x},$$

where the dagger denotes the pseudo inverse.

2. NON-LINEAR SPARSE SIGNAL MODELS

In this section we generalise the linear sparse model of the previous section to allow for more general non-linear functions and more general measures of approximation error.

The applicability of a linear sparse signal model to a particular problem depends heavily on the ability to find a linear model $\mathbf{x} \approx \mathbf{\Phi}\hat{\mathbf{y}}$ in which $\mathbf{x}$ can be approximated with a sparse coefficient vector. In this paper, we therefore extend the linear model to a general non-linear model, allow for more general cost functions $C_\mathbf{e}$ and suggest a greedy algorithm to calculate the sparse coefficients. Let the approximation be defined as

$$\hat{\mathbf{x}} = f(\hat{\mathbf{y}}),$$

for some known non-linear function f. Again, assume we are interested in finding a sparse vector $\hat{\mathbf{y}}$ that either approximates $\mathbf{x}$ or

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.

[1] Instead of the regularised optimisation problem given here, an alternative would be to optimise either one of the above terms subject to a constraint on the other one.

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

estimates $\mathbf{y}$. As $\mathbf{y}$ is not available, we can only measure the error as a function of the observation $\mathbf{x}$ and the estimate $\hat{\mathbf{y}}$ using a general cost function $C_{\mathbf{e}}(\mathbf{x},\hat{\mathbf{y}})$. Sparsity is measured by $C_{\hat{\mathbf{y}}}$ so that the problem becomes

$$\mathbf{y}^{opt} = \min_{\hat{\mathbf{y}}} C_{\mathbf{e}}(\mathbf{x},\hat{\mathbf{y}}) + \lambda C_{\mathbf{y}}(\hat{\mathbf{y}}). \qquad (2)$$

To find a (possibly non-optimal) solution to the above problem, we derive a greedy algorithm that is inspired by OMP, but can be used with more general[3] functions f and costs $C_{\mathbf{e}}$.

To motivate the use of the above model, we introduce a real world example in which the non-linear sparse model might be of benefit. In a previous paper [15], we have presented preliminary results on the applicability of compressed sensing ideas to remote radar imaging. A synthetic aperture radar is mounted on a satellite. Constraints on the computing power available on-board mean that it is often too costly to fully process and code the data on-board the satellite. Instead, following the compressed sensing paradigm, the data, which is often available in the spatial Fourier domain, can be reduced by discarding random subsets of the frequency bins. The observed signal $\mathbf{x}$ is then a quantised (i.e. noisy) subset of the frequency domain. If the full frequency information were available, a complex valued image could be reconstructed using a two dimensional Fourier transform. Whilst in this example, the phase of the image is often not very structured, the magnitude (representing a picture of the imaged area) is often sparse in a wavelet domain. Splitting the complex image into the magnitude and phase part is a non-linear transform. Therefore, the linear model is no-longer applicable and a non-linear model has to be used. Note that this signal model is different from the complex valued linear model in which the complex image is modelled using complex wavelet coefficients with sparse magnitudes. In the radar imaging problem, the non-linear operation (taking the magnitude) is performed in the image domain (i.e. before taking the wavelet transform) and not in the wavelet domain (that is after the wavelet transform).

3. THE GREEDY GRADIENT ALGORITHM

In this section, we describe a greedy gradient based algorithm to calculate the coefficients in the sparse non-linear model introduced in section 2.

As in OMP, we iteratively build up an estimate of $\hat{\mathbf{y}}$ by selecting a single element in each iteration. This selection is based on the current estimate $\hat{\mathbf{y}}^{[n-1]}$. The elements selected up to the current iteration are collected into the set $\Gamma^{[n]}$ and $\hat{\mathbf{y}}^{[n]}$ is estimated by optimising $C_{\mathbf{e}}$ under the constraint that only those elements with indices in $\Gamma^{[n]}$ are non-zero.

3.1 Greedy selection strategy

As mentioned above, the selection step in OMP can be understood as follows. The gradient of the squared error is evaluated at the current coefficient estimate and OMP selects that coefficient for which the derivative (with respect to this coefficient) is largest in magnitude. Using this point of view, a generalisation to more general differentiable cost functions $C_{\mathbf{e}}$ and non-linear signal models is straight forward. Let

$$g_i^{[n]} = \left.\frac{dC_{\mathbf{e}}(\mathbf{x},\hat{\mathbf{y}})}{d\hat{y}_i}\right|_{\hat{\mathbf{y}}^{[n-1]}}$$

be the derivative of $C_{\mathbf{e}}$ with respect to the i^{th} coefficient $\hat{y}_i$, evaluated at the current estimate $\hat{\mathbf{y}}^{[n-1]}$. As in MP and OMP, selection is then based on the magnitude of $g_i^{[n]}$

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

In other words, we choose the coefficient that, if changed by a small amount, would give the largest benefit in terms of $C_{\mathbf{e}}$.

3.2 Optimisation

Also, as in OMP, the new estimate of $\mathbf{y}$ is found by adjusting those coefficients in $\hat{\mathbf{y}}$, whose indices are in the set $\Gamma^{[n]}$ of selected elements. The coefficients not in the set $\Gamma^{[n]}$ are left at zero. In OMP, the cost function $C_{\mathbf{e}}$ is quadratic and the constrained optimum is a minimum mean squared error estimate for which many efficient computational strategies are available.

In the more general case, there is no universal strategy for the required constrained optimisation. For general non-linear signal models $f(\hat{\mathbf{y}})$, for which the cost function $C_{\mathbf{e}}$ is differentiable, one could for example adopt gradient based optimisation strategies. The performance of any specific algorithm will depend on the particular non-linearity $f(\hat{\mathbf{y}})$ and cost $C_{\mathbf{e}}$. In this paper we therefore opt for the relatively simple gradient descent method, which is applicable in a wide range of contexts, though performance benefits might be available for any concrete example by a more refined algorithm choice, such as, for example, a conjugate gradient solver.

3.3 Pseudo-code

We summarise the *greedy gradient* algorithm as follows

```
1. Initialise $\hat{\mathbf{y}}^0 = \mathbf{0}, \Gamma^{[0]} = \emptyset$
2. for $n = 1; n := n+1$ till stopping criterion is met
   (a) Evaluate gradient of $C_{\mathbf{e}}$ at $\hat{\mathbf{y}}^{[n-1]}$ : $\mathbf{g}^{[n]} = \left.\frac{dC_{\mathbf{e}}}{d\hat{\mathbf{y}}}\right|_{\hat{\mathbf{y}}^{[n-1]}}$
   (b) $i^{[n]} = \arg_i \max \left|g_i^{[n]}\right|$
   (c) $\Gamma^{[n]} = \Gamma^{[n-1]} \cup i^{[n]}$
   (d) Optimise $C_{\mathbf{e}}$ under the constraint that $\hat{y}_i^{[n]} = 0$ for all $i \notin \Gamma^{[n]}$
3. Output $\hat{\mathbf{y}}^{[n]}$
```

An implementation of this strategy in Matlab will be incorporated into version 0.3 of the ‘sparsify’ Matlab toolbox, available on the first authors web-page.

3.4 Variations

There are now a wide range of variations on greedy algorithms, many of which are also applicable to the greedy gradient algorithm discussed here. Some possible extensions of the *greedy gradient* algorithm are listed below.

1. Often, prior information on the occurrence or importance of individual coefficients is available. In these circumstances, it might be beneficial to use pre-specified weights in the selection step to increase or decrease the probability of selecting individual elements.
2. Orthogonal Least Squares (OLS) [16][4] is a greedy algorithm similar to OMP, but with a slight variation on the element selection strategy. OMP is not guaranteed to select the element that will give the smallest residual error. OLS, on the other hand, uses a selection step that is guaranteed to lead to the smallest residual. A similar strategy might be feasible for the greedy gradient algorithm. However, this would require the estimation of the minimum of $C_{\mathbf{e}}$ for each possible choice of element to be selected, which would greatly increase the computational cost.
3. As mentioned above, the gradient optimisation suggested here is only one possibility and more advanced strategies might beavailable.
4. Different strategies have been proposed for OMP to select more than a single new element in each iteration [17] [18] [19]. Similar approaches might also be feasible for the non-linear problem discussed here.

[3] As the algorithm proposed here is based on the derivatives of $C_{\mathbf{e}}(\mathbf{x},\hat{\mathbf{y}})$ with respect to the coefficients $\hat{y}_i$, these should be well defined and non-zero at least for those points that are not locally optimal.

[4] also known as Order Recursive Matching Pursuit or Optimized Orthogonal Matching Pursuit

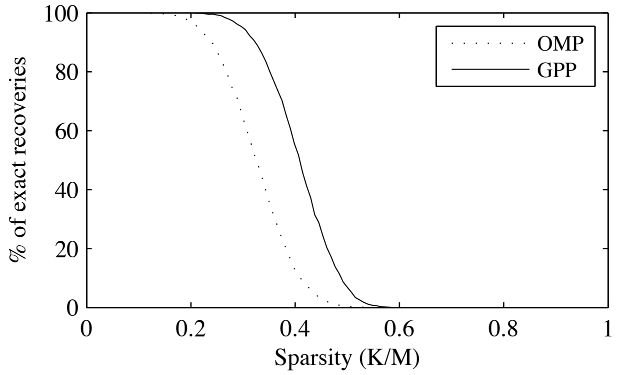

Figure 1: Exact recovery performance of OMP (dotted line) and Gradient Pursuit with cost function $\|\mathbf{\Phi}^\dagger(\mathbf{x}-\mathbf{\Phi}\mathbf{y})\|_2^2$ (GPP) (solid line).

5. Instead of calculating a (local) optimum in each iteration, it is possible to use a few gradient update steps in each iteration, before selecting a new element. This strategy was suggested for the gradient pursuit framework introduced in [14].
6. Instead of adding individual elements to the set $\Gamma^{[n]}$ in each iteration, it is also possible to select a new set $\Gamma^{[n]}$ in each iteration. Such a strategy is used by the M-sparse algorithm suggested for the linear sparse model in [20].
7. A strategy to remove elements from the set of selected elements might also be used. This could be done, for example, based on the difference in the cost (2) when one of the non-zero elements in $\mathbf{y}$ is set to zero.

4. EXPERIMENTAL EVALUATION

We present a small set of preliminary experiments to evaluate the proposed method. The first experiment looks at the performance gains achievable by optimising a different cost function in the standard linear model. The other two experiments are intended to give a flavour of the performance of the method under some possible non-linear models.

4.1 Optimising another cost function.

As mentioned above, the gradient based greedy algorithm proposed in this paper is not only applicable to non-linear mappings, it can also be applied to linear mappings in the same way as OMP, but with another measure for the approximation error. In this subsection, we study this application of the approach.

To motivate our particular example, we use a toy compressed sensing problem. Assume a signal $\mathbf{y} \in \mathbb{R}^N$ is known to be sparse. Instead of observing $\mathbf{y}$ directly, we only observe a shorter vector $\mathbf{x} \in \mathbb{R}^M$ with $M < N$. The observation model is linear, i.e. $\mathbf{x} = \mathbf{\Phi}\mathbf{y}$ for some known matrix $\mathbf{\Phi}$.

The OMP algorithm is built explicitly around the cost $\|\mathbf{x} - \mathbf{\Phi}\mathbf{y}\|_2^2$. This measure is in the observation space and determines how well our estimate $\mathbf{\Phi}\hat{\mathbf{y}}$ estimates $\mathbf{x}$. In compressed sensing one is interested in estimating $\mathbf{y}$ and not in approximating $\mathbf{x}$. Therefore, the error should be measured in the domain of $\mathbf{y}$ and not in that of $\mathbf{x}$ as done in OMP.

Unfortunately, $\mathbf{y}$ itself is not available. However, in the noiseless case, the observation model $\mathbf{x} = \mathbf{\Phi}\tilde{\mathbf{y}}$ defines a linear subspace of elements $\tilde{\mathbf{y}}$. A more appropriate error measure might therefore be to measure the distance of the estimate $\hat{\mathbf{y}}$ from the subspace defined by $\mathbf{x} = \mathbf{\Phi}\tilde{\mathbf{y}}$. Using an Euclidean distance, this can be measured using

$$\|\mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T)^{-1}(\mathbf{x}-\mathbf{\Phi}\hat{\mathbf{y}})\|_2^2. \quad (3)$$

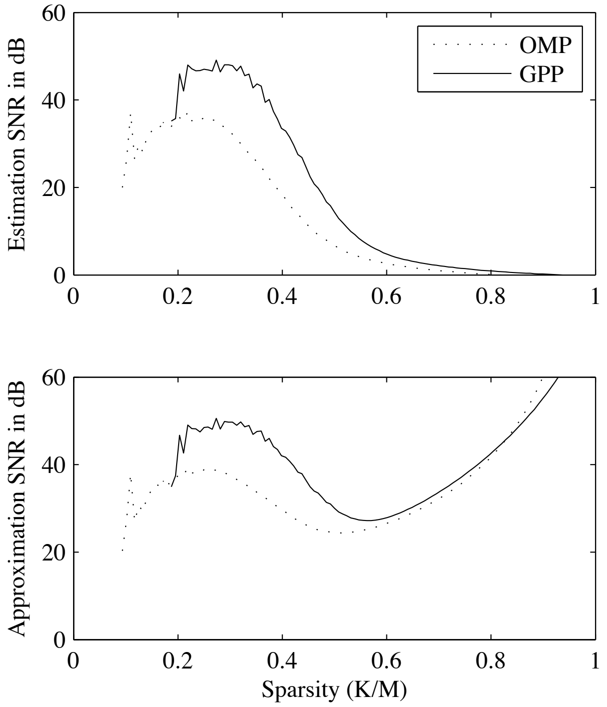

Figure 2: Error in estimating $\mathbf{y}$ top and approximating $\mathbf{x}$ bottom for OMP (dotted line) and Gradient Pursuit with cost function $\|\mathbf{\Phi}^\dagger(\mathbf{x}-\mathbf{\Phi}\mathbf{y})\|_2^2$ (GPP) (solid line). Shown are the averages over only those cases in which the exact support of $\mathbf{y}$ was not recovered.

Note that the use of the (re-scaled) pseudo-inverse as 'sensing matrix' has previously been suggested for the use in a thresholding algorithm [21].

The experiment reported here used matrices $\mathbf{\Phi} \in \mathbb{R}^{128\times 256}$ with columns drawn uniformly from the unit sphere. The first[5] K coefficients of $\mathbf{y}$ were then drawn from a unit variance normal distribution. K was varied from 1 to 128 and for each K, 10 000 realisations of the problem were generated. We then run standard OMP and our gradient based algorithm with the cost function (3), which we call the GPP[6] algorithm below. In each iteration, after the selection of an element based on the gradient of cost function (3), we used a single gradient descent step with the gradient restricted to the selected elements as proposed in [14]. We used the optimum step size which for this problem can be evaluated in closed form. Both methods were run until they had selected K elements.

Figure 1 looks at the percentage of cases in which each of the algorithms were able to recover the exact support set of elements in $\mathbf{y}$ used to generate the signal. From this graph it is evident that using the cost function suggested here leads to a better average performance in terms of estimation of $\mathbf{y}$.

In figure 2, the estimation error $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$ is shown in dB in the upper panel and the approximation error $\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$, again in dB in the lower panel. Because the error is negligible in the cases in which the algorithm identifies the correct atoms, the results are here averaged over those results in which the algorithms failed to recover the exact support set.

It is evident that even if the algorithms miss-specified the non-zero elements, the gradient based algorithm using cost function (3) outperforms OMP on average. However, it should also be noted that

[5] As the columns in $\mathbf{\Phi}$ have an i.i.d. distribution, it doesn't make a difference how the locations of the non-zero elements are chosen.

[6] The second P can stand for pseudo-inverse or alternatively for projection.

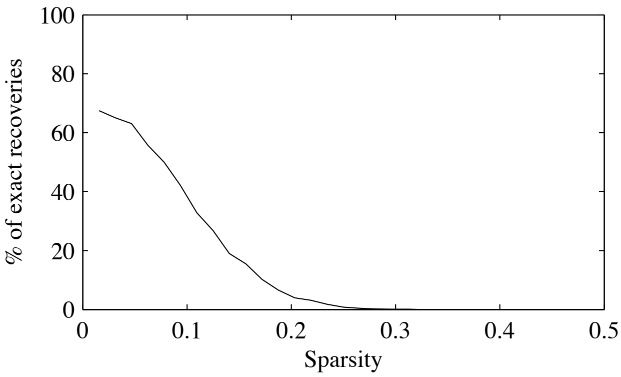

Figure 3: Exact recovery performance for different levels of sparsity in non-linear magnitude and phase estimation example.

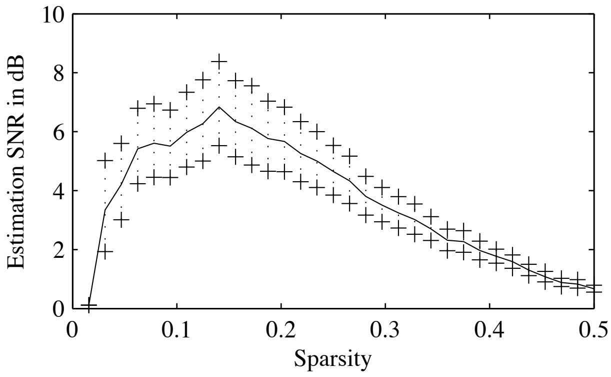

Figure 4: Error in estimating $\mathbf{y}$ top in the non-linear magnitude and phase estimation example. Shown are the averages over only those cases in which the exact support of $\mathbf{y}$ was not recovered. Also shown are the error bars.

calculating the gradient of cost function (3) requires the solution of a linear equation involving the pseudo inverse of $\mathbf{\Phi}$. We here evaluated and stored this pseudo inverse at the start of the algorithm.

4.2 Sparse magnitude and phase of complex variables.

This experiment was motivated by coherent imaging applications. Assume complex valued data is measured and the images of interest are the magnitude and phase of the complex data. Further assume that the magnitude and phase is sparse in some transform domain. In compressed sensing for imaging applications, instead of observing the complex valued image data, a subset of coefficients is observed in another domain, such as, for example, the Fourier domain.

We here look at an artificial example. We generated two vectors $\tilde{\mathbf{z}}$ and $\tilde{\mathbf{y}}$ of sparse coefficients with K non-zero elements, each drawn from a normal distribution. These were transformed with the inverse wavelet transform and were then used as the magnitude ($\mathbf{z}$) and phase ($\mathbf{y}$) of complex valued data[7]. The M observations were generated by multiplying this complex data with a matrix $\mathbf{\Phi}$ with real coefficients drawn from an i.i.d. normal distribution. The model is then

$$\mathbf{x} = \mathbf{\Phi}[z_i * e^{jy_i}]_i,$$

where $[\cdot]_i$ denotes a vector with elements indexed by i. We here fixed the dimension of the complex valued data to $N = 256$ and used $M = 128$ complex valued observations $\mathbf{x}$.

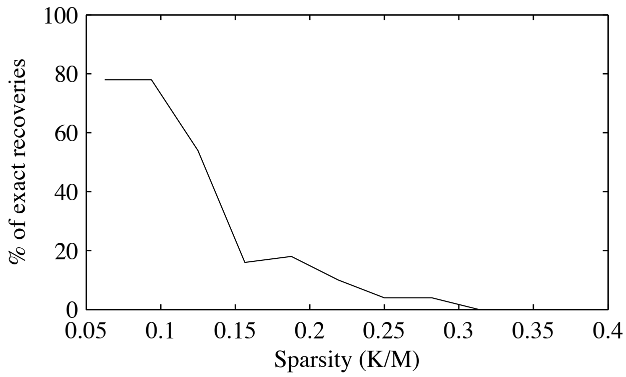

Figure 5: Exact recovery performance for different levels of sparsity in phase estimation example.

We repeated the following experiments for a range of sparsities, averaging the results over 1000 realisations. We run the algorithm derived in this paper to recover as many non-zero elements as there were non-zero elements in the original data. Figure 3 shows the percentage of cases in which the algorithm was able to exactly identify the coefficients that were used to generate the data. Figure 4 shows the average ratio of the signal energy to the error energy expressed in dB. We again average the error only over those cases for which the algorithm did not recover the exact support set. It is interesting to observe that for very sparse signals, if the algorithm does not recover the correct support set, then the estimation error is relatively large. This seems to be an artefact of only plotting the error for the cases in which the support set was not identified. For example, if there is only a single non-zero element to be recovered, if the incorrect element is identified, the error has to be large. On the other hand, if there are many non-zero elements, only a few elements might be incorrectly identified in which case the error will be small.

4.3 Sparse phase of a complex image with known magnitude.

The second example is similar to the previous one, but this time using two dimensional image data. To have more control over the problem, we use artificial two dimensional complex valued images. Each was generated using a phase that was sparse in the wavelet domain. We found the estimation of the phase to be more difficult in general. In this example, the magnitude was therefore assumed to be uniform and known. The task was to estimate the phase from observations of subsets of the Fourier coefficients of the complex image.

The image size was 64×64. We here look at the performance in terms of approximation of $\mathbf{x}$, estimation of $\mathbf{y}$ and recovery of the exact support.

In this example, we kept the number of non-zero elements K fixed but changed the number of observations M. Due to the high computation cost for this example, we here used only 50 experiments for each observation dimension. The percentage of cases in which the exact sparse support set was recovered is shown in figure 5. In figure 6 we show the average estimation error $\|\mathbf{y} - \hat{\mathbf{y}}\|_2^2 / \|\mathbf{y}\|_2^2$ in dB (upper panel) and the average approximation error $\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 / \|\mathbf{x}\|_2^2$ (in dB) in the lower panel. The results are again only averaged over those results in which the algorithm failed to recover the exact support set. Also shown are error bars.

Note that we here let the algorithm extract twice as many elements as were used to generate the signal. Because the observations were noiseless and allowed an exact sparse representation, if the set of identified elements included the correct element, the hope was that the other elements would be set to zero or negligible values. This approach was found to work well in practice.

[7] We here normalised the phase to be between $-\pi$ and π.

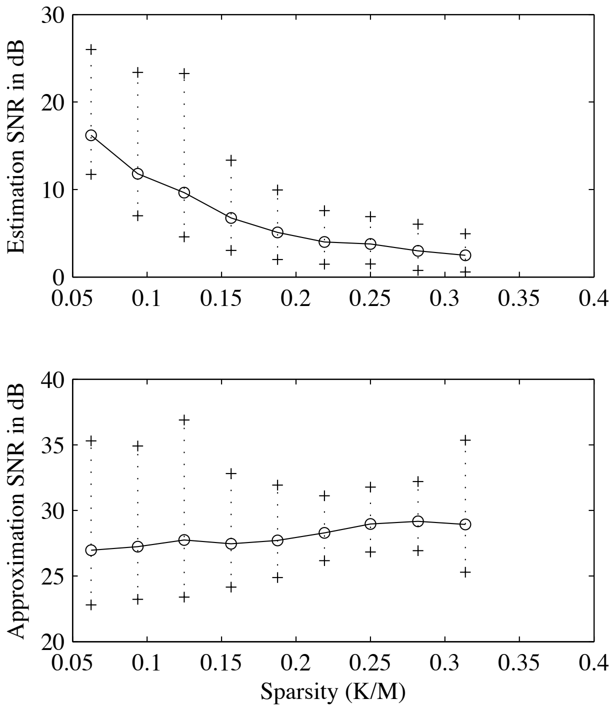

Figure 6: Error in estimating $\mathbf{y}$ and approximating $\mathbf{x}$ bottom in the phase estimation example. Shown are the averages over only those cases in which the exact support of $\mathbf{y}$ was not recovered. Also shown are the error bars.

5. CONCLUSIONS

In this paper we have looked at a general non-linear sparse modelling framework equipped with a quite general measure of approximation error and introduced a greedy gradient based algorithm for coefficient estimation. In the standard linear model, striking results were achieved with a cost function measuring the error in the coefficient domain. In this case, the greedy gradient algorithm was found to outperform OMP.

Solving the general non-linear problem is, however, more difficult and one could think of examples which do not even have unique solutions. Also, not only do we have to identify the set of non-zero coefficients, finding the optimal values of these coefficients is also not trivial as it might, for example, involve the minimisation of complicated cost functions with several local minima. It therefore seems clear that a general strategy that performs well on all problems is impossible.

Nevertheless, the results presented for non-linear models indicate that the proposed method can find the exact support set in certain cases. The performance intuitively depends on the non-linearity involved. In particular, the gradient of the cost function with respect to the individual coefficients has to be ‘well behaved’. For example, the selection criterion depends on this gradient to be able to indicate which coefficients to include. For highly non-linear models, this requirement is likely not to be fulfilled. A theoretical analysis of this remains to be undertaken. Also, whilst the preliminary experiments reported here are encouraging, we are currently evaluating the method on more challenging non-linear problems and are also planning the evaluation on real world data.

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] 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.

[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 Transactions 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 Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[14] T. Blumensath and M. Davies, “Gradient pursuits,” to appear in *IEEE Transactions on Signal Processing*, 2008.

[15] S. Bhattacharya, T. Blumensath, B. Mulgrew, and M. Davies, “Fast encoding of synthetic aperture radar raw data using compressed sensing,” in *IEEE Workshop on Statistical Signal Processing*, (Madison, USA), 2007.

[16] 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.

[17] D. Donoho, Y. Tsaig, I. Drori, and J. Starck, “Sparse solutions of underdetermined linear equations by stagewise orthogonal matching pursuit,” 2006.

[18] D. Needell and R. Vershynin, “Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit,” *submitted*, 2007.

[19] 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.

[20] T. Blumensath and M. Davies, “Iterative thresholding for sparse approximations,” to appear in *Journal of Fourier Analysis and Applications*, 2008.

[21] K. Schnass and P. Vandergheynst, “Average performance analysis for thresholding,” *IEEE Signal Processing Letters*, vol. 14, no. 11, pp. 828–831, 2007.