



HAL
open science

A Greedy Sparse Approximation Algorithm Based On L1-Norm Selection Rules

Ramzi Ben Mhenni, Sébastien Bourguignon, Jérôme Idier

► **To cite this version:**

Ramzi Ben Mhenni, Sébastien Bourguignon, Jérôme Idier. A Greedy Sparse Approximation Algorithm Based On L1-Norm Selection Rules. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), May 2020, Barcelona, Spain. pp.5390-5394, 10.1109/ICASSP40776.2020.9054670 . hal-02563553

HAL Id: hal-02563553

<https://hal.science/hal-02563553v1>

Submitted on 11 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

A GREEDY SPARSE APPROXIMATION ALGORITHM BASED ON ℓ_1 -NORM SELECTION RULES

Ramzi Ben Mhenni¹ Sébastien Bourguignon¹ Jérôme Idier¹

¹ École Centrale de Nantes – CNRS, LS2N, 1 rue de la Noë, F-44321 Nantes, France

ABSTRACT

We propose a new greedy sparse approximation algorithm, called SLS for *Single ℓ_1 Selection*, that addresses a least squares optimization problem under a cardinality constraint. The specificity and increased efficiency of SLS originate from the atom selection step, based on exploiting ℓ_1 -norm solutions. At each iteration, the regularization path of a least-squares criterion penalized by the ℓ_1 norm of the remaining variables is built. Then, the selected atom is chosen according to a scoring function defined over the solution path. Simulation results on difficult sparse deconvolution problems involving a highly correlated dictionary reveal the efficiency of the method, which outperforms popular greedy algorithms when the solution is sparse.

Index Terms— Sparse optimization, greedy algorithm, ℓ_1 -norm regularization, deconvolution.

1. INTRODUCTION

Sparse approximation is a very active research topic, in which we search for a sparse vector $\mathbf{x} \in \mathbb{R}^M$ (i.e., with a large number of zero components), to approximate data $\mathbf{y} \in \mathbb{R}^N$ as a linear combination of a small number of atoms, i.e., columns from a dictionary $\mathbf{H} \in \mathbb{R}^{N \times M}$. Applications range from sparsity-enhancing inverse problems in geophysics [1, 2] or nondestructive testing [3, 4], compression and denoising [5], or subset selection in Statistics [6].

The sparse approximation problem can be formulated by minimizing the least-squares approximation error and imposing an upper bound on the number of non-zero elements of the solution [7]:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \quad \text{subject to (s.t.)} \quad \|\mathbf{x}\|_0 \leq K, \quad (1)$$

where $\|\mathbf{x}\|_0$ denotes the ℓ_0 “norm”: $\|\mathbf{x}\|_0 := \text{Card}\{x_j | x_j \neq 0\}$. Finding the best K -sparse solution essentially resorts to combinatorial optimization, which is often not possible for high-dimensional problems—the problem (1) is NP-hard [8]. Therefore, many works in signal processing and statistics have proposed computationally efficient, suboptimal, methods, that can be classified into two categories [7].

On the one hand, *continuous relaxation* methods replace the ℓ_0 norm by a continuous surrogate function. Problem (1) is then converted into a continuous optimization problem. In particular, many works have considered the convex case with the ℓ_1 norm $\|\mathbf{x}\|_1 := \sum_n |x_n|$ [7].

On the other hand, *partial combinatorial exploration strategies* generate a sequence of sparse iterates by performing simple transitions on the support (the set of non-zero components) of \mathbf{x} . Among this category fall the well-known Matching Pursuit (MP) [9], Orthogonal Matching Pursuit (OMP) [10] and Orthogonal Least Squares (OLS) [11] greedy methods. However, the selection step

of a new atom in such methods is very sensitive to interferences between the different atoms, in particular in the case of highly correlated dictionaries [12]. More sophisticated strategies try to circumvent this issue (see for example [13]). Among them, the Single Best Replacement (SBR) algorithm [12] includes possible removal steps allowing to correct erroneously detected components, CoSAMP [14] considers including $2K$ components at each iteration and selects the K most likely ones, and A^* OMP [15] performs a partial tree-search exploration step for the selection step.

While the ℓ_1 -norm convex formulation exactly solves an approximate problem, the second class of methods returns a local optimum for the exact problem. In the compressed sensing framework [5], conditions on \mathbf{H} have been established, for which such approaches are ensured to solve the initial problem (1). However, in the case of inverse problems where \mathbf{H} is ill-conditioned, or if the mutual correlation of its columns is high, such guarantees are lost and, in practice, they often fail in finding the global optimum [16]. The relative performance of methods is then evaluated under a compromise criterion between the quality of the solution and its computational cost.

In this paper, we propose an algorithm which gathers advantages of the two classes of methods. It essentially consists of a greedy strategy, where the selection rule at each iteration is based on exploiting ℓ_1 -norm solutions. The number of iterations is then controlled by the sparsity level K of the searched solution, limiting the computational burden. Moreover, the selection of each new atom, based on solving a convex optimization problem, is expected to be more robust to interferences (high correlations) between the different atoms than standard greedy methods. More precisely, at a given iteration, an ℓ_1 -norm-penalized problem is built, where the ℓ_1 norm operates on the non-selected variables, and the regularization path is computed by an homotopy continuation algorithm [17, 18]. A heuristic selection rule is then proposed, which considers the predominant variable along the regularization path.

Section 2 presents classical greedy forward algorithms and discusses the limitations of their variable selection rules. In Section 3, we introduce our new selection rule, and implementation issues are addressed. The resulting algorithm, called SLS for Single ℓ_1 Selection, is then evaluated and compared to other sparse approximation algorithms in Section 4.

2. LIMITATIONS OF FORWARD GREEDY ALGORITHMS

Forward greedy methods start from an empty set and iteratively construct a sparse solution, by alternating between two steps: a new atom is selected by maximizing a score function, denoted by \mathcal{F} , and then the model is updated. Let \mathbb{S} denote the index set of the variables already selected (the current *support* of the solution), and let $\mathbf{H}_{\mathbb{S}}$ denote the matrix composed of the columns of \mathbf{H} indexed by \mathbb{S} . Similarly, $\bar{\mathbb{S}}$ indexes the remaining variables. We also denote the size

of \mathbb{S} (respectively, of $\bar{\mathbb{S}}$) by k (respectively, \bar{k}). The general principle of forward selection algorithms is given in Algorithm 1.

Algorithm 1: Forward Selection greedy algorithm

```

1 Initialization:  $\mathbb{S} = \emptyset$ 
2 while  $|\mathbb{S}| < K$  do
3   Variable selection:  $\hat{j} = \operatorname{argmax}_{j \in \bar{\mathbb{S}}} \mathcal{F}(j)$ ;
4   Support update:  $\mathbb{S} \leftarrow \mathbb{S} \cup \{\hat{j}\}$ ;
5   (If needed) update the estimate  $\mathbf{x}_{\mathbb{S}}$  and the residual
      $\mathbf{r} = \mathbf{y} - \mathbf{H}_{\mathbb{S}}\mathbf{x}_{\mathbb{S}}$ ;
6 end
Result: support  $\mathbb{S}$  and solution  $\mathbf{x}_{\mathbb{S}}$ 

```

In the sequel, we suppose that all columns in \mathbf{H} have unit norm. For both MP and OMP algorithms, the selected atom is the most correlated to the residual:

$$\mathcal{F}_{(\text{O})\text{MP}}(j) = |\mathbf{h}_j^T \mathbf{r}|, \quad j \in \mathbb{R}^M, \quad (2)$$

where \mathbf{h}_j is the j -th column of \mathbf{H} . OMP includes an additional orthogonalization step, which updates the current solution on its support by:

$$\mathbf{x}_{\mathbb{S}} = \operatorname{argmin}_{\mathbf{x}_{\mathbb{S}} \in \mathbb{R}^k} \|\mathbf{y} - \mathbf{H}_{\mathbb{S}}\mathbf{x}_{\mathbb{S}}\|_2^2 = (\mathbf{H}_{\mathbb{S}}^T \mathbf{H}_{\mathbb{S}})^{-1} \mathbf{H}_{\mathbb{S}}^T \mathbf{y},$$

such that the residual is orthogonal to the model $\mathbf{H}_{\mathbb{S}}\mathbf{x}_{\mathbb{S}}$. In this case, the score function (2) is computed only for $j \in \bar{\mathbb{S}}$.

For OLS [11], the approximation error is minimized among all possible supports including one new component:

$$\hat{j} = \operatorname{argmin}_{j \in \bar{\mathbb{S}}} \min_{\mathbf{x}_{\mathbb{S} \cup \{j\}} \in \mathbb{R}^k} \|\mathbf{y} - \mathbf{H}_{\mathbb{S} \cup \{j\}} \mathbf{x}_{\mathbb{S} \cup \{j\}}\|_2^2,$$

which amounts to

$$\mathcal{F}_{\text{OLS}}(j) = \|\mathbf{H}_{\bar{\mathbb{S}} \cup \{j\}} (\mathbf{H}_{\bar{\mathbb{S}} \cup \{j\}}^T \mathbf{H}_{\bar{\mathbb{S}} \cup \{j\}})^{-1} \mathbf{H}_{\bar{\mathbb{S}} \cup \{j\}}^T \mathbf{y}\|_2^2.$$

The atom selection stage of MP, OMP and OLS is a one-step procedure, which is a short-term vision of the selection issue. In the case of highly correlated dictionaries, this represents a major limitation of such greedy algorithms. For illustration purposes, let us consider a sparse deconvolution problem, where \mathbf{H} is composed of shifted versions of the impulse response of the filter, and adjacent atoms give very close contributions to the model. The toy example in Figure 1 (a) and (b) illustrates a situation where \mathbf{x} is composed of two close spikes, giving strongly overlapping echoes in the data \mathbf{y} . The score function for the first iteration of both OMP and OLS is $\mathcal{F}(j) = |\mathbf{h}_j^T \mathbf{y}|$, and is shown in Figure 1 (c). It is maximal for the index located in the middle of the two true indices, thus selecting a wrong atom – even in the noiseless case.

3. SINGLE L1 SELECTION (SLS)

In this Section, we build a new forward method, where a more accurate selection rule is proposed, based on a global, convex, formulation of the problem.

3.1. Selection rule based on ℓ_1 problems

Our selection rule exploits the solutions of ℓ_1 -norm-based problems. Recall the example in Figure 1. It is clear that a joint approach to the sparse estimation problem would be less sensitive to interferences

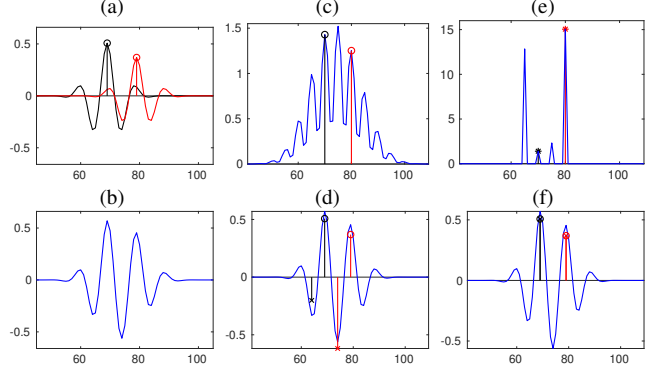


Fig. 1. a) Toy example where standard greedy selection fails. (a) Contribution of two atoms and (b) corresponding noise-free data. (c) Scoring function for OMP and OLS at first iteration and (d) OLS solution after two iterations. (e) Scoring function at the first iteration of the proposed SLS algorithm and (f) SLS solution after two iterations. Circles (resp. stars) locate the true (resp. estimated) spikes.

between the atom contributions, by allowing non-zero weights to the true atom locations. Therefore, we propose to base the selection of a new atom by considering the following ℓ_1 -norm optimization problem at each iteration of a greedy forward procedure:

$$\min_{\mathbf{x}_{\mathbb{S}} \in \mathbb{R}^k, \mathbf{x}_{\bar{\mathbb{S}}} \in \mathbb{R}^{\bar{k}}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}_{\mathbb{S}}\mathbf{x}_{\mathbb{S}} - \mathbf{H}_{\bar{\mathbb{S}}}\mathbf{x}_{\bar{\mathbb{S}}}\|_2^2 + \lambda \|\mathbf{x}_{\bar{\mathbb{S}}}\|_1, \quad (3)$$

Similarly to OLS, such a criterion allows the joint re-estimation of the amplitudes of previously selected components (whereas they are fixed in the residual for the selection rule of OMP). It also jointly estimates a sparse vector for the remaining ones, which is not restricted to a single non-zero component as in OLS.

We remark first that the problem in Eq. (3) can be recast as an optimization problem in variables $\mathbf{x}_{\bar{\mathbb{S}}}$ only: indeed, for a given $\mathbf{x}_{\bar{\mathbb{S}}}$, the solution in $\mathbf{x}_{\mathbb{S}}$ is explicit. Its expression can be inserted into the least-squares term in (3), so that the problem amounts to the following standard ℓ_1 -norm-penalized problem:

$$\mathcal{P}^\lambda : \min_{\mathbf{x}_{\bar{\mathbb{S}}} \in \mathbb{R}^{\bar{k}}} \frac{1}{2} \|\bar{\mathbf{y}} - \bar{\mathbf{H}}_{\bar{\mathbb{S}}}\mathbf{x}_{\bar{\mathbb{S}}}\|_2^2 + \lambda \|\mathbf{x}_{\bar{\mathbb{S}}}\|_1, \quad (4)$$

where $\bar{\mathbf{y}} := \mathbf{P}\mathbf{y}$, $\bar{\mathbf{H}}_{\bar{\mathbb{S}}} := \mathbf{P}\mathbf{H}_{\bar{\mathbb{S}}}$ and $\mathbf{P} := \mathbf{I}_k - \mathbf{H}_{\mathbb{S}}(\mathbf{H}_{\mathbb{S}}^T \mathbf{H}_{\mathbb{S}})^{-1} \mathbf{H}_{\mathbb{S}}^T$, with \mathbf{I}_k the $k \times k$ identity matrix.

3.2. Exploiting the regularization path

We now build a selection rule based on problem \mathcal{P}^λ in (4). Instead of considering a particular solution for a given value of λ , we propose the following two-step strategy.

Homotopy – We first build the solution path, that is, the set of solutions of \mathcal{P}^λ for $\lambda \in [\lambda^{\min}, \lambda^{\max}]$, by the homotopy continuation algorithm, which is acknowledged as a very efficient algorithm for ℓ_1 -norm optimization problems when the solution is sparse [17, 18]. Starting from $\lambda^{\max} := \|\bar{\mathbf{H}}_{\bar{\mathbb{S}}}^T \bar{\mathbf{y}}\|_\infty$ (such that the solution is identically zero for $\lambda \geq \lambda^{\max}$), homotopy iteratively identifies the different *breakpoints*, denoted $\lambda^{(p)}$, $p = 1, \dots, P$, that lead to changes in the support configuration (the set of positive, negative and zero variables). Between two breakpoints, the solution is linear in λ and has an analytical expression. Implementation details can be found, *e.g.*, in [18]. Figure 2 illustrates the iterations performed by the homotopy algorithm.

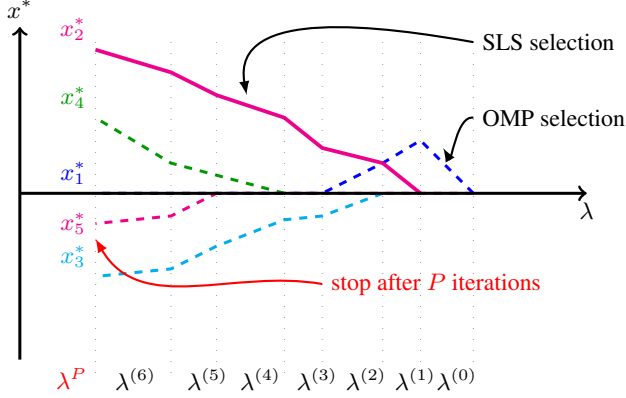


Fig. 2. Selection rule computed over the solution path obtained by the homotopy method, as λ decreases from $\lambda^{(0)} = \lambda^{\max}$ to $\lambda^{(P)}$. Path corresponding to non-zero variables in \mathbf{x}^* , renamed x_1^* to x_5^* . Although x_1 is the first component to be non-zero, x_2 is the most prominent variable on $[\lambda^{(P)}, \lambda^{(0)}]$.

Scoring – Then, we use the score function defined over the set of solutions for $\lambda \in [\lambda^{\min}, \lambda^{\max}]$ by:

$$\mathcal{F}_{\text{SLS}}(j) = \sum_{p=0}^P |x_j^{*(p)}|, \quad j \in \bar{\mathbb{S}}, \quad (5)$$

where $\mathbf{x}^{*(p)}$ denotes the solution of \mathcal{P}^λ at the breakpoint $\lambda = \lambda^{(p)}$. The score function can be updated recursively at each iteration of the homotopy algorithm. This heuristic rule aims at selecting the most prominent component over the regularization path $[\lambda^{\min}, \lambda^{\max}]$.

In order to control the computing time with a simple rule, we limit the number P of iterations of the homotopy algorithm (which defines λ^{\min}), to $P = c(K - k)$, where $K - k$ is the number of components that still need to be included in the greedy search, and c is a parameter whose tuning will be discussed in Section 4. Such an empirical rule follows the idea that a larger solution path is preferred for the first iterations of the greedy procedure, because more competition may exist between atoms, so the solution may be less stable as a function of λ .

The SLS selection rule is illustrated on a simulated regularization path in Figure 2. Remark that in the limiting case $c \rightarrow 0$ ($\lambda \rightarrow \lambda^{\max}$), it corresponds to selecting the first non-zero component in the regularization path, that is, $\arg \max_{j \in \bar{\mathbb{S}}} \bar{\mathbf{h}}_{\bar{\mathbb{S}}_j}^T \bar{\mathbf{y}}$, corresponds to the OMP selection rule. On the toy example of Section 2, the first iteration of the SLS algorithm allows the correct selection of one true atom — see Figure 1 (e). Then, after two iterations, the true support is correctly estimated, as shows Figure 1 (f). Algorithm 2 summarizes the computation of \mathcal{F}_{SLS} via the homotopy algorithm.

At each iteration k of SLS, we call the homotopy algorithm to build the solution path in $[\lambda^{(P)}, \lambda^{(0)}]$. From the computational point of view, each step of the homotopy algorithm essentially involves the rank-one update of a linear system to compute \mathbf{d} . Our implementation makes use of the blockwise inversion technique to solve the linear systems in an incremental way.

4. SIMULATION RESULTS

We evaluate the performance of the SLS algorithm, compared to several well-known sparse estimation algorithms: OMP [10], ℓ_1 -

Algorithm 2: Homotopy algorithm for computing the score function \mathcal{F} . Indexation by I returns non-zero variables.

```

1 Set  $p = 0$ ,  $s^{(0)} = 0$  and  $\mathbf{x}^{*(0)} = 0$ .
2 Initialize  $\lambda^{(0)} = \lambda^{\max}$  and  $I = \{\arg \max_{j \in \bar{\mathbb{S}}} \bar{\mathbf{h}}_{\bar{\mathbb{S}}_j}^T \bar{\mathbf{y}}\}$ .
3 while  $p \leq P$  do
4    $p \leftarrow p + 1$ 
5   % Compute the change of direction %
6    $\mathbf{d}_I^{(p)} = (\bar{\mathbf{H}}_I^T \bar{\mathbf{H}}_I)^{-1} \text{sgn}(\mathbf{x}_I^{(p-1)})$ 
7   % Compute the step length  $\gamma^{(p)}$  %
8    $\gamma^{(p)}$ : smallest decrease in  $\lambda$  to reach a new breakpoint
9   % Update the solution %
10   $\mathbf{x}_I^{(p)} = \mathbf{x}_I^{(p-1)} + \gamma^{(p)} \mathbf{d}_I^{(p)}$ ;
11  % Recursive computation of the score function %
12   $\mathcal{F}_I^{(p)} = \mathcal{F}_I^{(p-1)} + |\mathbf{x}_I^{(p)}|$ ;
13  % Update  $I$  %
14 end
15 Return the score  $\mathcal{F}$ .
```

norm regularization or BP (Basis Pursuit), computed here by the homotopy algorithm [17], OLS [11], SBR¹ [12], Subspace Pursuit² [14], accelerated Iterative Hard Thresholding³ (IHT) [19] and A*OMP⁴ [15]. All algorithms are implemented in Matlab and are tuned in order to obtain solutions with the true sparsity level K .

Algorithms are tested on difficult sparse deconvolution problems, with an up-sampled convolution model in order to achieve high-resolution spike locations [20]. The problem is underdetermined with $M = 1,000$ and $N = 350$. Columns of \mathbf{H} are then highly correlated, with mutual coherence $\max_{i \neq j} |\mathbf{h}_i^T \mathbf{h}_j| = 0.81$. White Gaussian noise ϵ is then added with $\text{SNR}_{\text{dB}} = 10 \log \frac{\|\mathbf{H}\mathbf{x}\|_2^2}{\|\epsilon\|_2^2} = 20$ dB. Results are averaged over 50 random

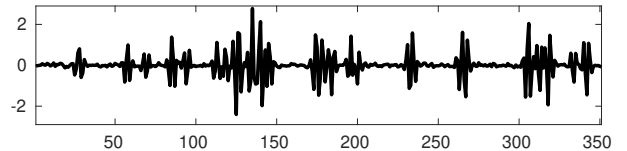


Fig. 3. Typical sparse deconvolution data \mathbf{y} .

realizations of the sparse sequence and of noise. A typical signal is shown in Figure 3.

We respectively denote by \mathbf{x} and \mathbf{x}^{true} the estimated and the true solutions. In order to focus on the capacity of the methods to find the best solution and to detect the correct support, we consider several error measures:

- the quadratic error $E_Q = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2$;
- the support error $E_S = \|\mathbf{b} - \mathbf{b}^{\text{true}}\|_0$, where $b_j = 1$ (respectively, $b_j = 0$) if $x_j \neq 0$ (respectively, if $x_j = 0$);

¹Implementation was taken from the multimedia material linked to [12].
²B. Sturm's implementation taken at http://media.aau.dk/null_space_pursuits/2011/07/08/subspacepursuit.m
³T. Blumensath's implementation AIHT.m taken at <http://www.personal.soton.ac.uk/tblm08/sparsify/sparsify.htm>
⁴N.B. Karahonoğlu's implementation taken at <http://myweb.sabanciuniv.edu/karahanoglu/research>

- the exact recovery rate E_{exact} , which is the average number of instances for which the algorithm correctly locates the support of the true sequence.

We first analyze the efficiency of our SLS algorithm as a function of the control parameter c . Figure 4 shows errors E_Q and E_S by varying c from 0.5 to 5, as a function of the computation time, compared to the previously introduced algorithms, and for sparsity levels K ranging from 10 to 40.

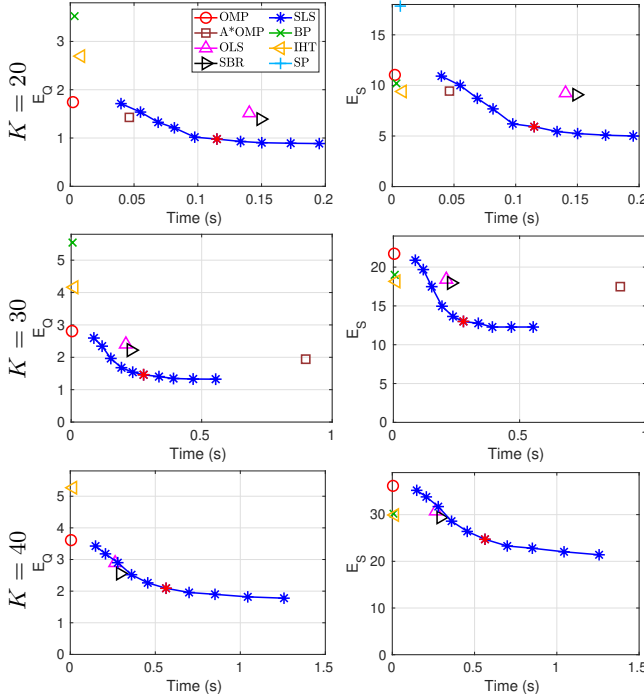


Fig. 4. Performance of SLS (*) as a function of the computation time, by varying parameter c from 0.5 (left) to 5 (right), compared to different sparse approximation methods. The tuning with $c = 3$ is highlighted with a red marker.

For small c , SLS still behaves as a fast and poorly performing algorithm on such difficult problems (recall that for $c = 0$, SLS identifies with OMP). As could be expected, the quality of the solution improves as c increases, at the expense of the increase of the computation time. In particular, for $K \leq 30$, the quality of solutions provided by OLS and SBR is achieved with smaller computation time ($c \sim 1.5$), and much better solutions than OLS and SBR are obtained for their respective computation time ($c \sim 3$). For less sparse solutions ($K \geq 40$), SLS behaves like OLS and SBR on such “quality vs. time diagram”, although better solutions can be obtained with higher computation times. A*OMP competes with SLS for $K = 20$, but takes much more time for greater values of K .

From Figure 4, we remark that the value of $c = 3$ is a good compromise between computation time and solution quality. Thus, we fix it at this value for the next comparisons. Figure 5 presents quantitative results for all algorithms as a function of K . The performance of SLS is clearly the best one in terms of solution quality with respect to the three error criteria. Finally, SLS has a lower computation time than OLS and SBR up to $K = 30$ and slightly higher for $K = 40$. Compared to A*OMP, SLS has a slightly higher computation time up to $K = 20$, but A*OMP becomes computationally too expensive. Note that other fast algorithms (OMP, BP, IHT and SP)

are always much faster than SLS—but always give worse solutions.

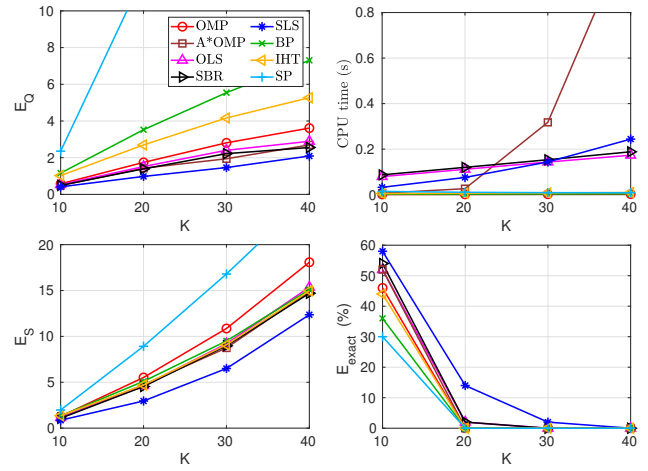


Fig. 5. Estimation errors and computation time as a function of K for SLS with $c = 3$ (*) and other sparse approximation methods.

5. CONCLUSIONS

We proposed a novel sparse approximation greedy algorithm, called SLS for *Single L1 Selection*, whose atom selection rule is based on the ℓ_1 -norm solutions. This selection rule exploits the regularization path of the least squares criterion penalized by the ℓ_1 -norm, with a view to improve the robustness of the selection step. We empirically showed that SLS outperforms other standard greedy methods to solve difficult inverse problems where the dictionary columns are highly correlated such as in deconvolution. Moreover, the computing cost of SLS can be made comparable to that of OLS and SBR, exploiting the fact that only partial ℓ_1 regularization paths are needed.

6. REFERENCES

- [1] H. Taylor, S. Banks, and F. McCoy, “Deconvolution with the ℓ_1 norm,” *Geophysics*, vol. 44, no. 1, pp. 39–52, Jan. 1979.
- [2] J. Mendel, *Optimal seismic deconvolution: An estimation-based approach*, Monograph Series. Academic Press, 1983.
- [3] C. A. Zala, “High-resolution inversion of ultrasonic traces,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 39, no. 4, pp. 458–463, July 1992.
- [4] M. S. O’Brien, A. N. Sinclair, and S. M. Kramer, “Recovery of a sparse spike time series by L1 norm deconvolution,” *IEEE Transactions on Signal Processing*, vol. 42, pp. 3353–3365, 1994.
- [5] Y. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge University Press, 2012.
- [6] A. Miller, *Subset selection in regression*, Chapman and Hall/CRC, 2002.
- [7] J. A. Tropp and S. J. Wright, “Computational methods for sparse solution of linear inverse problems,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, June 2010.
- [8] B. K. Natarajan, “Sparse approximate solutions to linear systems,” *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.

- [9] S. Mallat, "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [10] Y. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44 vol.1.
- [11] S. Chen, S. 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.
- [12] C. Soussen, J. Idier, D. Brie, and J. Duan, "From Bernoulli Gaussian Deconvolution to Sparse Signal Restoration," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4572–4584, 2011.
- [13] I. Selesnick and I. Bayram, "Sparse signal estimation by maximally sparse convex optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1078–1092, Mar. 2014.
- [14] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301 – 321, 2009.
- [15] N. B. Karahanoglu and H. Erdogan, "A* orthogonal matching pursuit: Best-first search for compressed sensing signal recovery," *Digital Signal Processing*, vol. 22, no. 4, pp. 555 – 568, 2012.
- [16] S. Bourguignon, J. Ninin, H. Carfantan, and M. Mongeau, "Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance," *IEEE Transactions on Signal Processing*, vol. 64, no. 6, pp. 1405–1419, March 2016.
- [17] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1996.
- [18] D. L. Donoho and Y. Tsaig, "Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse," *IEEE Transactions on Information Theory*, vol. 54, pp. 4789–4812, 2008.
- [19] T. Blumensath and M. E. Davies, "Iterative Thresholding for Sparse Approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, Dec. 2008.
- [20] E. Carcreff, S. Bourguignon, J. Idier, and L. Simon, "Resolution enhancement of ultrasonic signals by up-sampled sparse deconvolution," in *IEEE International Conference on Acoustic, Speech and Signal Processing*, Vancouver, Canada, May 2013.