An optimized version of non-negative OMP

Thanh T. NGUYEN[1], Charles SOUSSEN[1], Jérôme IDIER[2], El-Hadi DJERMOUNE[1]

[1]Centre de Recherche en Automatique de Nancy (UMR 7039). Campus Sciences, B.P. 70239, F-54506 Vandœuvre-lès-Nancy

[2]Laboratoire des Sciences du Numérique de Nantes (UMR 6004), 1 rue de la Noë, BP 92101, F-44321 Nantes Cedex 3

{thi-thanh.nguyen,charles.soussen,el-hadi.djermoune}@univ-lorraine.fr;jerome.idier@ls2n.fr

Résumé – Le problème traité est l'approximation parcimonieuse sous contrainte de positivité. Nous proposons une implémentation récursive de l'algorithme Non-Negative Orthogonal Matching Pursuit (NNOMP) basée sur la résolution de sous-problèmes de moindres carrés par l'algorithme des contraintes actives. Nous proposons de plus une amélioration de NNOMP, appelée SNNOMP, basée sur le rétrécissement du support lorsque les coordonnées du vecteur parcimonieux s'annulent. Les algorithmes proposés sont comparés avec les implémentations existantes de NNOMP pour un problème de déconvolution impulsionnelle qui met en jeu un dictionnaire mal conditionné.

Abstract – This article addresses least-squares minimization under sparsity and non-negativity constraints. We propose a recursive implementation of Non-Negative Orthogonal Matching Pursuit (NNOMP) based on the active set method for solving least-squares subproblems. We further propose an improvement of NNOMP, named support-Shrinkage NNOMP (SNNOMP), based on the shrinkage of the support of iterates when some coordinates vanish. SNNOMP is compared with the existing versions of NNOMP for a sparse deconvolution problem.

1 Introduction

Sparse approximation under non-negativity constraints naturally arises in several applications such as image processing, optical spectroscopy and non-negative matrix factorization [1,2]. Many sparse solvers such as Matching Pursuit (MP), Iterative Hard Thresholding, FISTA can be directly extended to the non-negative setting. It is not the case of OMP [3], which is a well-known extension of MP. Its principle is to gradually update the sparse solution support by selecting a new dictionary atom at each iteration. When dealing with non-negative constraints, the orthogonal projection computed at each OMP iteration is replaced by a non-negative least-squares (NNLS) subproblem whose solution is not explicit. Therefore, the usual recursive (fast) implementations of OMP [4] do not apply. The NNOMP algorithm was first proposed by Bruckstein *et al.* [5]. This algorithm was later renamed as CNNOMP by Yaghoobi *et al.* [6]. Within NNOMP, the atom selection and the NNLS subproblem are two separate tasks. The NNLS subproblems in NNOMP are solved independently by calling an NNLS subroutine, which makes NNOMP computationally inefficient. [6] introduced a faster implementation, named Fast NNOMP (FNNOMP), which combines the atom selection and the NNLS subproblem in only one task. By this alternative approach, FNNOMP recursively *approximates* the sought solution by making use of QR factorization without solving NNLS subproblems. Therefore, FNNOMP may not yield the NNOMP output.

We introduce an *exact* and *recursive* implementation of NNOMP based on the active set algorithm for solving the NNLS subproblems [7]. We further propose an improvement of NNOMP called support-Shrinkage NNOMP (SNNOMP). The NNOMP and SNNOMP implementations are recursive in the sense that their current iterate is used as a warm start for initializing the next call to NNLS. It is noticeable that the structure of the active set algorithm [8] for solving NNLS shares a strong similarity with OMP (without constraint) [9]. The active set iterations [8] are based on an atom selection step similar to that in OMP, followed by a support shrinkage. This similarity was not further exploited in [9], whereas it is a keypoint of our contribution.

Ce travail est financé par le projet ANR BECOSE (ANR-15-CE23-0021).

2 Problem statement and prerequisites

Given a data signal $\boldsymbol{y} \in \mathbb{R}^m$ and a dictionary $H \in \mathbb{R}^{m \times n}$ whose columns are normalized, the aim is to find a K-sparse non-negative vector $\boldsymbol{x} \in \mathbb{R}^n$ yielding an accurate approximation $\boldsymbol{y} \approx H\boldsymbol{x}$. This leads to solving the following problem:

$$\min_{\boldsymbol{x} \geq 0, \|\boldsymbol{x}\|_0 \leq K} \quad \|\boldsymbol{y} - H\boldsymbol{x}\|_2^2 \tag{1}$$

where $\|\cdot\|_0$ is the ℓ_0-"norm" counting the number of non-zero elements. Note that the NNOMP scheme in [5] actually aims to minimize $\|\boldsymbol{x}\|_0$ s.t. $\boldsymbol{x} \geq 0$ and $\boldsymbol{y} = H\boldsymbol{x}$ (*i.e.*, in the noise-free case) wherein H has more columns than rows. The structure of NNOMP remains unchanged in the noisy case, where the stopping condition reads $\|\boldsymbol{y} - H\boldsymbol{x}\|_2^2 < \epsilon$ or $\|\boldsymbol{x}\|_0 \leq K$ [6]. Hereafter, we denote by S the support of $\boldsymbol{x}$ ($S = \{i : x_i \neq 0\}$), $\overline{S}$ the complement of S, H_S the subdictionary indexed by S and $\boldsymbol{x}_S$ the solution restricted to S.

A key ingredient to address (1) is to solve the NNLS subproblem $\min_{\boldsymbol{z} \geq 0} \|\boldsymbol{y} - A\boldsymbol{z}\|_2^2$ with $A \leftarrow H_S$. Assuming A to

Algorithm 1: active set algorithm for solving NNLS [7]

input : $\boldsymbol{y}, A, \boldsymbol{z}_0$
output: $\boldsymbol{z}$, nonnegative minimizer of $\|\boldsymbol{y} - A\boldsymbol{z}\|_2^2$

1 $\boldsymbol{z} \leftarrow \boldsymbol{z}_0$;
2 $S \leftarrow \text{supp}(\boldsymbol{z})$;
3 **while** *not stop* **do**
4 $\quad \boldsymbol{p} \leftarrow A_S^\dagger \boldsymbol{y} - \boldsymbol{z}_S$;
5 $\quad \alpha \leftarrow 1$;
6 $\quad$ **if** $S \neq \emptyset$ *and* $\boldsymbol{p} \neq 0$ **then**
7 $\quad\quad j \leftarrow \arg\min\{-z_i/p_i : i \in S, p_i < 0\}$;
8 $\quad\quad \alpha \leftarrow \min\{1, \min\{-z_i/p_i : i \in S, p_i < 0\}\}$;
9 $\quad\quad \boldsymbol{z}_S \leftarrow \boldsymbol{z}_S + \alpha \boldsymbol{p}$;
10 $\quad$ **end**
11 $\quad$ **if** $\alpha = 1$ **then**
12 $\quad\quad \boldsymbol{r} \leftarrow \boldsymbol{y} - A_S \boldsymbol{z}_S$;
13 $\quad\quad \lambda \leftarrow -A_{\overline{S}}^T \boldsymbol{r}$;
14 $\quad\quad$ **if** $\lambda \geq 0$ *or* $\overline{S} = \emptyset$ **then**
15 $\quad\quad\quad$ **stop**
16 $\quad\quad$ **else**
17 $\quad\quad\quad S \leftarrow S \cup \{\ell\}$ with $\ell \leftarrow \arg\min_i \lambda_i$;
18 $\quad\quad$ **end**
19 $\quad$ **else**
20 $\quad\quad S \leftarrow S \setminus \{j\}$;
21 $\quad$ **end**
22 **end**
23 $\boldsymbol{z}_{\overline{S}} \leftarrow 0$;

be full column rank, the NNLS subproblem is strictly convex, and since it is a quadratic problem, it admits a unique solution. One of the widely used NNLS algorithms is the active set method [7, 8]. The active set is defined as the set of indices of the inequality constraints that become equalities at the current point. In the case of NNLS, the active set coincides with $\overline{S} = \{i : x_i = 0\}$. Therefore, we can rewrite active set algorithms in terms of support. The algorithm in [7] is rewritten in Algorithm 1. Observe the analogy with the OMP selection rule [9] in lines 12-13 and 17.

3 Active set implementation of NNOMP

3.1 Principle of NNOMP

NNOMP (Algorithm 2 without line 9) shares the structure of OMP. The latter starts from the empty support; at each iteration, it selects the atom having the highest correlation with current residual; it adds the atom index to the support and then computes the projection of $\boldsymbol{y}$ onto the subspace induced by the current support. Because of the non-negativity constraint, NNOMP exhibits two essential differences with OMP. First, the selection rule (line 5) involves the inner product without absolute value. Second, the orthogonal projection computed at each iteration in OMP is replaced by an NNLS subproblem (line 7) whose solution is not explicit. Hence, an iterative algorithm is required to solve the NNLS subproblem.

Besides, OMP and NNOMP are *forward* greedy algorithms

Algorithm 2: NNOMP (without line 9) and SNNOMP (with line 9) for solving (1)

input : $\boldsymbol{y}, H, K$
output: $\boldsymbol{x}$

1 $\boldsymbol{z} \leftarrow 0$;
2 $S \leftarrow \emptyset$;
3 $\boldsymbol{r} \leftarrow \boldsymbol{y}$;
4 **while** $|S| < K$ *and* $\max H_{\overline{S}}^T \boldsymbol{r} > 0$ **do**
5 $\quad \ell \leftarrow \arg\max_{i \in \overline{S}} \quad \langle \boldsymbol{r}, H_i \rangle$;
6 $\quad S \leftarrow S \cup \{\ell\}$;
7 $\quad \boldsymbol{z} \leftarrow \arg\min_{\boldsymbol{z} \geq 0} \quad \|\boldsymbol{y} - H_S \boldsymbol{z}\|_2^2$;
8 $\quad \boldsymbol{r} \leftarrow \boldsymbol{y} - H_S \boldsymbol{z}$;
9 $\quad \boxed{S \leftarrow S(\text{supp}(\boldsymbol{z}))}$;
10 **end**
11 $\boldsymbol{x}_S \leftarrow \boldsymbol{z}_S$;$\boldsymbol{x}_{\overline{S}} \leftarrow 0$;

which gradually add atoms to the support. When studying NNOMP, we noticed that because of the non-negativity constraint, some coefficients x_i ($i \in S$) are likely to vanish, *i.e.*, $S = \text{supp}(\boldsymbol{x})$ is not always true. NNOMP and FNNOMP choose to keep the related atoms in the current support whereas in the proposed SNNOMP algorithm, we choose to deselect them so that $S = \text{supp}(\boldsymbol{x})$ is always true. The difference between SNNOMP and NNOMP is emphasized in line 9 of Algorithm 2: the current support S is shrinked to the support of the NNLS solution. This is a *backward* move in which atoms are deselected, however the squared error is unchanged (contrary to the forward-*backward* extensions of OMP [10, 11] wherein atom deselections are allowed provided that the increase of squared error is small enough). Therefore, the shrinkage step helps correcting potential wrong selections. At the same time, the squared error keeps decreasing after each iteration. Because a given support cannot be explored twice, the algorithm terminates after a finite number of iterations.

3.2 Implementation

On the one hand, CNNOMP [6] solves NNLS subproblems independently by calling the Matlab function *lsqnonneg*. *lsqnonneg* is a non-recursive implementation of the Lawson & Hanson active set algorithm [8] which starts with the empty support. Therefore, CNNOMP is highly time-consuming when the expected solution support is large. On the other hand, FNNOMP avoids solving NNLS subproblems by recursively updating the QR decomposition [6]. When dealing with highly correlated dictionaries, we found that FNNOMP has limitations since it often returns negative coefficients.

While studying active set implementations of NNLS, we noticed that although [7] and [8] obey the same rules, the former accepts any feasible initial solution (*i.e.*, a warm start to speed up convergence) whereas the latter starts from the zero solution. Using warm starts is a key to an efficient recursive implementation of NNOMP. By using the previous NNLS solution as a warm start for the current NNLS subproblem, it is

expected that the active set algorithm will terminate after a few subiterations only.

In our implementation of NNOMP, the NNLS subproblem is solved at each iteration by calling Algorithm 1 with inputs: $\boldsymbol{y}$, $A = H_S$, $\boldsymbol{z}_0 = [\boldsymbol{z}^T, 0]^T$, where $\boldsymbol{z}$ is the NNLS solution found in the previous iteration. In our experiments, the convergence of NNLS was reached after no more than 2 subiterations. In addition, the computation of the orthogonal projection $A_S^\dagger \boldsymbol{y}$ in line 4 of Algorithm 1 can be accelerated by using either the matrix inversion lemma, the QR decomposition or the Cholesky factorization as in OMP [4]. The SNNOMP implementation was done similarly except that $\boldsymbol{z}_0 = [\hat{\boldsymbol{z}}^T, 0]^T$, where $\hat{\boldsymbol{z}}$ is the NNLS solution at the previous iteration restricted to its support.

4 Comparison in sparse deconvolution

4.1 Data generation

In our simulations, the model $\boldsymbol{y} = H\boldsymbol{x}^* + \boldsymbol{n}$ is considered where $\boldsymbol{x}^*$ and $\boldsymbol{n}$ stand for the ground truth and noise, respectively. The matrix H has normalized columns. The signal-to-noise ratio (SNR) is defined by $\text{SNR} = 10\log_{10}(P_{H\boldsymbol{x}^*}/P_{\boldsymbol{n}})$ where $P_{H\boldsymbol{x}^*} = \|Hx^*\|_2^2/m$ is the average power of the noise-free data and $P_{\boldsymbol{n}}$ is the noise variance.

We consider a convolution problem whose kernel $\boldsymbol{h}$ is a zero-centered Gaussian function of standard deviation σ. The sampling frequency of $\boldsymbol{x}$ is twice that of the data signal $\boldsymbol{y}$, so that the dictionary H has roughly twice more columns than rows. Note that H is not a Toeplitz matrix since the sampling frequency of $\boldsymbol{x}$ differs from that of $\boldsymbol{y}$. The ground truth support S^* of cardinality K^* is randomly generated so that the distance between consecutive support elements cannot be less than some fixed value to ensure that H_{S^*} is (in)coherent enough. The non-negative coefficients $\boldsymbol{x}^*_{S^*}$ are randomly generated using a folded Gaussian distribution[1]. Finally, the data $\boldsymbol{y}$ are obtained by adding Gaussian noise to $H\boldsymbol{x}^*$.

4.2 Process and test result

The FNNOMP implementation downloaded from the author's webpage[2] is compared with our NNOMP and SNNOMP implementations. The CNNOMP implementation [6] is not considered since it is not recursive. The maximum support cardinality is chosen as $K > K^*$ to compensate for the fact that the vector $\boldsymbol{z}$ found at line 7 in Algorithm 2 might have zero amplitudes. In other words, more than K^* iterations are necessary to reach a K^*-sparse iterate. For each algorithm, the "best" k-sparse iterate (that is, the iterate having the lowest residual among all iterates satisflying $\|\boldsymbol{x}\|_0 = k$) is stored for all $k \in \{0, \ldots, K\}$. A single solution is chosen afterwards by using a model order selection method (recall that the model order is $\|\boldsymbol{x}^*\|_0 = K^*$). Following [12], we noticed that Minimum Description Length (MDL) is more efficient for sparse deconvolution than Akaike and cross-validation criteria [13], which often over-estimate the expected order. Specifically, the corrected MDL (MDLc) [14] version dedicated to short data records (when the number of data points m is moderately larger than K) is defined as

$$\min_k \left\{ \log \varepsilon(\boldsymbol{x}_k) + \frac{\log(m)(k+1)}{m-k-2} \right\} \tag{2}$$

where $\boldsymbol{x}_k$ is the k-sparse solution and $\varepsilon(\boldsymbol{x}_k)$ denotes the corresponding squared residual norm. Broadly speaking, as the first term in (2) is a non-increasing function of k and the second term always increases as k increases, (2) tends to return the cardinality from which the squared residual almost stops decreasing, that is, when the residual signal mostly contains the observation noise. We next present our tests for noise-free and noisy cases. In both cases, the dictionary H is the same, $\sigma = 5$, $m = 150$, $n = 270$, the mutual coherence of $H \approx 0.99$, and K is set to $K^* + 20$.

Table 1: [Noise-free case] Average CPU time, Recall and Precision w.r.t. model order.

Generative model order		5	10	15	20	25	30
Time (ms)	NNOMP	10	9	10	12	14	17
	FNNOMP	7	5	6	8	8	10
	SNNOMP	12	18	28	23	24	28
Recall (%)	NNOMP	100	78	35	23	24	30
	FNNOMP	100	63	36	25	22	25
	SNNOMP	100	99	96	60	47	52
Precision (%)	NNOMP	95	47	20	14	16	21
	FNNOMP	95	27	16	13	12	16
	SNNOMP	95	61	48	30	26	32

4.2.1 Noise-free case

The same test is repeated for $K^* = 5, 10, 15, 20, 25$, and 30. For each K^*, we generated 30 trials, and then computed the average value of CPU time, Recall and Precision for each algorithm. We recall that "true positives" (TP) and "false positives" (FP) are the number of good detections and wrong detections, respectively. Recall and Precision are the rate of good detections among the expected support (*i.e.*, TP/K^*) and among all detections (*i.e.*, $TP/(TP + FP)$), respectively. As shown in Table 1, NNOMP takes more time than FNNOMP but often has higher Recall and Precision. SNNOMP takes the longest time due to the shrinkage step but also has the highest Recall and Precision. In the noise-free case, FNNOMP often returned some negative coefficients; the average rate of negative coordinates amounts 0, 41, 32, 29, 25 and 21 %, respectively.

4.2.2 Noisy case

The same process as above is repeated for SNRs ranging in $[0, 40]$ dB. We observed that SNNOMP often performs better than competitors at high SNRs. For SNRs lower than 20 dB, the three algorithms return similar results. FNNOMP may return a few negative coefficients, especially for large SNRs. A

[1] defined as $\boldsymbol{x}^*_{S^*} = |\boldsymbol{t}|$ where $\boldsymbol{t}$ obeys a Gaussian distribution.

[2] `http://www.mehrdadya.com`

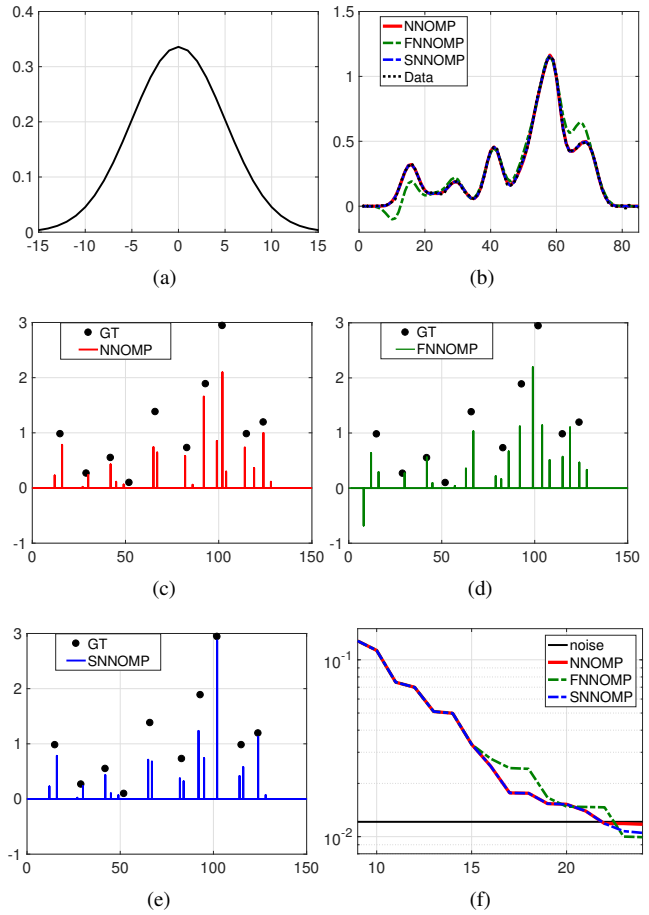

Figure 1: [SNR = 30 dB] (a) Kernel $\boldsymbol{h}$. (b) Data $\boldsymbol{y}$ and approximations $H\boldsymbol{x}$. (c, d, e) Ground truth (GT) and sparse recoveries found using the MDLc rule. (f) Noise level ($\|\boldsymbol{n}\|_2^2$) and squared residuals w.r.t. $\|\boldsymbol{x}\|_0$.

typical situation wherein SNR = 30 dB and $K^* = 10$ is illustrated in Figure 1. In Fig. 1(b), the SNNOMP and NNOMP approximations are both superimposed with the data $\boldsymbol{y}$, contrary to the FNNOMP approximation, which is less accurate. From Figs. 1(c)–(e), it appears that SNNOMP yields more accurate detection of the spike locations and amplitudes (*e.g.*, around index 100) than NNOMP and FNNOMP. The latter returns one negative coefficient. The decrease of the NNOMP and SNNOMP squared error w.r.t. $\|\boldsymbol{x}\|_0$ is faster than that of FNNOMP (Fig. 1(f)).

5 Conclusion

Addressing least-squares minimization with sparsity and non-negativity constraints, we introduced a recursive implementation of NNOMP using an active set algorithm for solving NNLS subproblems. We also proposed to introduce a support shrinkage step to improve the practical performance of NNOMP, leading to the so-called SNNOMP algorithm.

Other greedy algorithms, *e.g.*, Orthogonal Least Squares and its forward-backward extensions could also lend themselves to non-negative versions, with the same requirement of recursively solving many NNLS subproblems. We expect that these more involved algorithms will improve the effectiveness of the NNOMP-like algorithms.

References

[1] T. Virtanen, J. F. Gemmeke, and B. Raj, "Active-set Newton algorithm for overcomplete non-negative representations of audio", *IEEE Trans. Audio, Speech, Language Process.*, vol. 21, no. 11, pp. 2277–2289, Nov. 2013.

[2] R. Bro and S. De Jong, "A fast non-negativity-constrained least squares algorithm", *Journal of Chemometrics*, vol. 11, no. 5, pp. 393–401, 1997.

[3] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition", in *Proc. 27th Asilomar Conf. on Signals, Sys. and Comp.*, Nov. 1993, vol. 1, pp. 40–44.

[4] B. L. Sturm and M. G. Christensen, "Comparison of orthogonal matching pursuit implementations", in *Proc. Eur. Sig. Proc. Conf.*, Bucharest, Romania, Aug. 2012, pp. 220–224.

[5] A. M. Bruckstein, M. Elad, and M. Zibulevsky, "On the uniqueness of nonnegative sparse solutions to underdetermined systems of equation", *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 4813–4820, Nov. 2008.

[6] M. Yaghoobi, D. Wu, and M. E. Davies, "Fast non-negative orthogonal matching pursuit", *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1229–1233, Sept. 2015.

[7] J. Nocedal and S. Wright, *Numerical optimization*, Springer-Verlag, New York, 1999.

[8] C. L. Lawson and R. J. Hanson, *Solving least squares problems*, pp. 149–199, Prentice-Hall, Saddle River, NJ, USA, Society for Industrial and Applied Mathematics (SIAM) edition, 1974.

[9] S. Foucart and D. Koslicki, "Sparse recovery by means of nonnegative least squares", *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 498–502, Apr. 2014.

[10] C. Herzet and A. Drémeau, "Bayesian pursuit algorithms", in *Proc. Eur. Sig. Proc. Conf.*, Aalborg, Denmark, Aug. 2010, pp. 1474–1478.

[11] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations", *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4689–4708, July 2011.

[12] C. Soussen, J. Idier, J. Duan, and D. Brie, "Homotopy based algorithms for ℓ_0-regularized least-squares", *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3301–3316, 2015.

[13] P. Stoica and Y. Selén, "Model-order selection: a review of information criterion rules", *IEEE Sig. Proc. Mag.*, vol. 21, no. 4, pp. 36–47, July 2004.

[14] F. de Ridder, R. Pintelon, J. Schoukens, and D. P. Gillikin, "Modified AIC and MDL model selection criteria for short data records", *IEEE Trans. Instrum. and Meas.*, vol. 54, no. 1, pp. 144–150, Feb. 2005.