

# Double Least Squares Pursuit for Sparse Decomposition

Wanyi Li, Peng Wang, and Hong Qiao

Institute of Automation, Chinese Academy of Sciences  
No. 95 of Zhongguancun East Road, Haidian District, Beijing, China 100190  
{wanyi.li,peng\_wang,hong.qiao}@ia.ac.cn  
<http://www.ia.ac.cn>

**Abstract.** Sparse decomposition has been widely used in numerous applications, such as image processing, pattern recognition, remote sensing and computational biology. Despite plenty of theoretical developments have been proposed, developing, implementing and analyzing novel fast sparse approximation algorithm is still an open problem. In this paper, a new pursuit algorithm Double Least Squares Pursuit (DLSP) is proposed for sparse decomposition. In this algorithm, the support of the solution is obtained by sorting the coefficients which are calculated by the first Least-Squares, and then the non-zero values over this support are detected by the second Least-Squares. The results of numerical experiment demonstrate the effectiveness of the proposed method, which is with less time complexity, more simple form, and gives close or even better performance compared to the classical Orthogonal Matching Pursuit (OMP) method.

**Keywords:** Sparse decomposition, Sparse representation, Sparse approximation algorithm, Double Least-Squares Pursuit.

## 1 Introduction

The sparse decomposition problem (also referred to as sparse approximation) is one of the main problems for sparse representation and compressed sensing. Given a full rank matrix  $\mathbf{A} \in \mathbb{R}^{n \times m}$  with  $n < m$  and a vector  $\mathbf{b} \in \mathbb{R}^n$ , the sparse decomposition problem can be stated as follows [1]:

$$(P_0) : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{b} = \mathbf{A}\mathbf{x} \quad (1)$$

i.e., find a sparsest representation for  $\mathbf{b}$  over  $\mathbf{A}$ , or

$$(P_0^\epsilon) : \min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon \quad (2)$$

i.e., find a sparsest approximation for  $\mathbf{b}$  with error  $\epsilon$ .

$(P_0)$  is the exact case and  $(P_0^\epsilon)$  is the error-tolerant version of  $(P_0)$ , with error tolerance  $\epsilon > 0$ , and  $\|\mathbf{x}\|_0$  represents the number of nonzero entries in vector  $\mathbf{x}$ .

In both  $(P_0)$  and  $(P_0^\epsilon)$ , matrix  $\mathbf{A}$  is often referred to as dictionary while vector  $\mathbf{b}$  is referred to as observation. In practice,  $(P_0^\epsilon)$  is more suitable for real world problems than  $(P_0)$ .

It has been proven that, given an arbitrary redundant dictionary  $\mathbf{A}$  and an observation  $\mathbf{b}$ , to solve the sparse representation  $(P_0)$  and  $(P_0^\epsilon)$  is a NP-hard problem [2]. As a result, researchers turn to find the approximate solutions for this problem.

The methods for solving  $(P_0)$  and  $(P_0^\epsilon)$  mainly include two categories: greedy algorithms [3,4] and convex relaxation techniques [5-8]. Greedy algorithms for approximating the solution of  $l_0 - norm$ , such as orthogonal matching pursuit (OMP) [3] and matching pursuit(MP) [4], make a sequence of locally optimal choices in hope of determining a globally optimal solution. Although these methods are simple and efficient, the solutions are sub-optimal. Convex relaxation techniques replace the combinatorial sparse approximation problem with a related convex program in hope that the solutions coincide, i.e., replacing the highly discontinuous  $l_0 - norm$  by a continuous or even smooth approximation, such as  $l_p$  norms for  $p \in (0, 1]$  or even by smooth function. A lot of algorithms have been proposed for  $l_p - norm$  sparse decomposition [5-8]. Although these methods have made some success in solving many practical problems, the computational cost of the methods still needs to be further treated, and to develop, implement and analyze novel fast sparse approximation algorithms is still an open problem [9].

In this paper, we propose a new pursuit algorithm — Double Least Squares Pursuit (DLSP) motivated by Least-Squares. In the proposed DLSP algorithm, the support of the solution is obtained by sorting the coefficients which are calculated by the first Least-Squares, and then the non-zero values over this support are detected by the second Least-Squares. The results of numerical experiment demonstrate that the proposed method is with less time complexity, more simple form and quite good performance.

The paper is organized as follows. Section 2 introduces the basic Orthogonal Matching Pursuit (OMP) method. Section 3 is devoted to the details of our proposed algorithm — Double Least-Squares Pursuit (DLSP). Section 4 presents the numerical results and comparison experiment results with other methods. Section 5 contains the conclusions.

## 2 Orthogonal Matching Pursuit (OMP)

The OMP [3] is the most classical greedy algorithm for approximating the solution of  $(P_0)$  or  $(P_0^\epsilon)$ . Its basic procedure is described in Fig. 1. OMP selects the column which is most correlated with the current residuals at each step, and the selected column is added into the support set. Then, the residuals are updated by projecting the observation onto the linear subspace which is spanned by the columns that have already been selected, i.e., the support set, and the algorithm

then iterates. The algorithm does not stop until the  $l_2$  - norm of the residual reaches a pre-specified value  $\epsilon$ . Compared with other alternative methods, OMP is with more simple form and faster implementation.

Thresholding-Algorithm is a simplification of the OMP, which selects the  $k$  largest inner product as the desired support, and only the first projection is used. Besides, many improved or extended versions of OMP have been developed [11-13].

**Task:** Approximating the solution of  $(P_0^\epsilon) : \min_{\mathbf{x}} \|\mathbf{x}\|_0$  subject to  $\|\mathbf{b} - \mathbf{Ax}\|_2 \leq \epsilon$   
**Parameters:** Given the matrix  $\mathbf{A}$ , the vector  $\mathbf{b}$ , and the error threshold  $\epsilon$ .  
**Initialization:**  $k=0$ , and

- the initial solution  $\mathbf{x}^0 = 0$ ,
- the initial residual  $\mathbf{r}^0 = \mathbf{b} - \mathbf{Ax}^0 = \mathbf{b}$ ,
- the initial solution support  $S^0 = support(\mathbf{x}^0) = \Phi$ .

**Main Iteration:**  $k=k+1$  and performing the following steps:

- **Greedy selection.** Find atom  $\mathbf{a}_{j_0}, j_0 = arg \max_j \mathbf{a}_j^T \mathbf{r}^{k-1}$
- **update Support:** Update  $S^k = S^{(k-1)} \cup j_0$ .
- **Update Provisional Solution:** Compute  $\mathbf{x}_k$ , the minimizer of  $\min \|\mathbf{b} - \mathbf{Ax}\|_2^2$  subject to  $Support\{\mathbf{x}\} = S^k$ .
- **Update Residual:** Compute  $\mathbf{r}_k = \mathbf{b} - \mathbf{Ax}^k$ .
- **Stopping Rule:** If  $\|\mathbf{r}^k\|_2 \leq \epsilon$ , stop, Otherwise apply another iteration.

**Output:** The proposed solution is  $\mathbf{x}_k$  obtained after  $k$  iterations.

**Fig. 1.** Orthogonal-Matching-Pursuit, a greedy algorithm for approximating the solution of  $(P_0^\epsilon)$

### 3 Double Least Squares Pursuit

#### 3.1 Motivation

Euclidean or  $l_2$  - norm problem is one of the most common norm approximation problems [10], which is also called the least-squares approximation problem, i.e.,

$$\min \|\mathbf{b} - \mathbf{Ax}\|_2^2 \tag{3}$$

and the objective is the sum of squares of the residuals.

The problem has the unique solution,

$$\mathbf{x}_{opt} = \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{b} = \mathbf{A}^+\mathbf{b} \tag{4}$$

From above we have

$$\mathbf{b} \approx \sum_{i=1}^n \mathbf{a}_i x_{opt}^1 + \mathbf{a}_2 x_{opt}^2 + \dots + \mathbf{a}_m x_{opt}^m \tag{5}$$

where  $\mathbf{a}_i$  represents the  $i$ -th column of matrix  $\mathbf{A}$ , and  $x_{opt}^i$  is the  $i$ -th entry of  $x_{opt}$ .

The absolute value of  $x_{opt}^i$  represents the contribution of the  $i$ -th column of  $\mathbf{A}$ , i.e.,  $\mathbf{a}_i$ , for the representation of  $\mathbf{b}$ . So we can select the  $k$  columns of  $\mathbf{A}$  correspond with the  $k$  largest entries of  $x_{opt}$  as the desired support.

### 3.2 Double Least-Squares Pursuit

Fig. 2 presents a formal description of Double Least Squares Pursuit.

Note that the stage “ set  $\mathbf{x}_{opt} = \mathbf{A}^+ \mathbf{b}$  ” in “Initialization ” is our first Least-Squares and the Updating Provisional Solution stage is another Least-Squares, so we called the proposed algorithm as Double Least Squares Pursuit.

After getting the solution of the least-squares approximation problem  $\mathbf{x}_{opt}$ , we select the  $k$  columns of  $\mathbf{A}$  correspond with the  $k$  largest entries of  $\mathbf{x}_{opt}$  as the desired support, and then the non-zero values over this support are detected by the second Least-Squares.

**Task:** Approximating the solution of  $(P_0^\epsilon)$  :  $\min_{\mathbf{x}} \|\mathbf{x}\|_0$  subject to  $\|\mathbf{b} - \mathbf{Ax}\|_2 \leq \epsilon$

**Parameters:** Given the matrix  $\mathbf{A}$ , the vector  $\mathbf{b}$ , and the error threshold  $\epsilon$ .

**Initialization:**  $k=0$ , and

- the initial solution  $\mathbf{x}^0 = 0$ ,
- the initial residual  $\mathbf{r}^0 = \mathbf{b} - \mathbf{Ax}^0 = \mathbf{b}$ ,
- the initial solution support  $S^0 = support(\mathbf{x}^0) = \Phi$ .
- the solution of the least-squares approximation problem,  $\min \|\mathbf{b} - \mathbf{Ax}\|_2^2$   
 $\mathbf{x}_{opt} = \mathbf{A}^+ \mathbf{b}$

**Main Iteration:** Incrementing  $k$  by 1 and performing the following steps:

- **Updating Support.** find the maxima,  $j_0$  of  $x_{opt}^j$  :  $\forall j \notin S_{k-1}, x_{opt}^{j_0} \geq x_{opt}^j$ , and update  $S^k = S^{(k-1)} \cup j_0$
- **Update Provisional Solution:** Compute  $\mathbf{x}_k$ , the minimizer of  $\min \|\mathbf{b} - \mathbf{Ax}\|_2^2$  subject to  $Support\{\mathbf{x}\} = S^k$ .
- **Update Residual:** Compute  $\mathbf{r}_k = \mathbf{b} - \mathbf{Ax}^k$ .
- **Stopping Rule:** If  $\|\mathbf{r}^k\|_2 \leq \epsilon$ , stop, Otherwise apply another iteration.

**Output:** The proposed solution is  $\mathbf{x}_k$  obtained after  $k$  iterations.

Fig. 2. Double Least Squares Pursuit for approximating the solution of  $(P_0^\epsilon)$

In Equation (5), when  $\mathbf{A}$  is an orthogonal matrix, we have

$$\mathbf{x}_{opt} = \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}\mathbf{b} = \mathbf{A}^T\mathbf{Ib} = \mathbf{A}^T\mathbf{b} \tag{6}$$

$\mathbf{x}_{opt}^i$  becomes the inner product of  $\mathbf{a}_i$  and  $\mathbf{b}$ , and Double Least-Squares Pursuit is equivalent to the Thresholding-Algorithm. Both of them find the support of the solution by choosing the  $k$  largest inner products.

In the above algorithm description, the number of the required non-zeros,  $k$ , is assumed to be already obtained. Alternatively, we can increase  $k$  until the error  $\|\mathbf{b} - \mathbf{Ax}^k\|_2$  reaches a pre-specified value  $\epsilon$ .

### 3.3 Time Complexity

For DLSP, due to the pseudo-inverse of matrix  $\mathbf{A}$  can be computed and stored in advance, the searching for the  $k$  elements of the supports amounts to a simple sort of the entries of the vector  $\mathbf{A}^+\mathbf{b}$ . OMP needs to compute the inner products in each iteration. So if the proposed solution has  $k_0$  non-zeros, the OMP method requires  $2k_0mn$  flops, while DLSP and Thresholding-Algorithm methods require only  $k_0mn$  flops. Obviously, DLSP is faster and simpler than OMP.

## 4 Numerical Experiments

We compared the Double Least-Squares Pursuit (DLSP) with OMP and the Thresholding-Algorithm on a simple date set to demonstrate their comparative behavior. Experimental data and measurement method in [1] are adopted in this paper: Random matrix  $\mathbf{A}$  with size  $30 \times 50$  is created with entries drawn from the norm distribution, and the columns of this matrix are normalized to have a unit  $l_2 - norm$ . The sparse vector  $\mathbf{x}$  is with independent and identically-distributed random supports of cardinalities in the range  $[1,10]$ , and its non-zero entries are drawn as random uniform variables in the range  $[-2, 1] \cup [1, 2]$ . Once  $\mathbf{x}$  is generated, we compute  $\mathbf{b} = \mathbf{Ax}$ , and then apply the above mentioned algorithms to seek for  $\mathbf{x}$ . We perform 1,000 such tests per cardinality, and the average results are used. According to the Uniqueness-Spark Theorem[1], in all of our tests the original solution is also the sparsest, as the spark of  $\mathbf{A}$  is 31.

In this experiment, the distance between the solution  $\mathbf{x}_k$  and the ground truth  $\mathbf{x}$  is measured by  $l_2 - error$  and recovery of the support. The  $l_2 - error$  is computed by  $\|\mathbf{x} - \mathbf{x}^k\|^2 / \|\mathbf{x}\|^2$ . The recovery of the support is computed as the distance between the supports of the two solutions, denoting the two supports as  $S'$  and  $S$ , we define the distance by

$$dist(S', S) = \frac{\max\{|S'|, |S|\} - |S' \cap S|}{\max\{|S'|, |S|\}} \tag{7}$$

where,  $|S|$  represents the size of  $S$ .

All these algorithms seek the proper solution until the residual is below a certain threshold (i.e.,  $\|r^k\|_2 \leq 1e-4$ ). The experimental results are summarized in Fig. 3 and Fig. 4.

Fig. 3 and Fig. 4 show that the performance of DLSP is close to OMP and better than Thresholding-Algorithm. DLSPs performance in terms of relative  $l_2$  recovery error is better than OMPs, while in terms of the success rate in detecting the true support is close to OMPs.

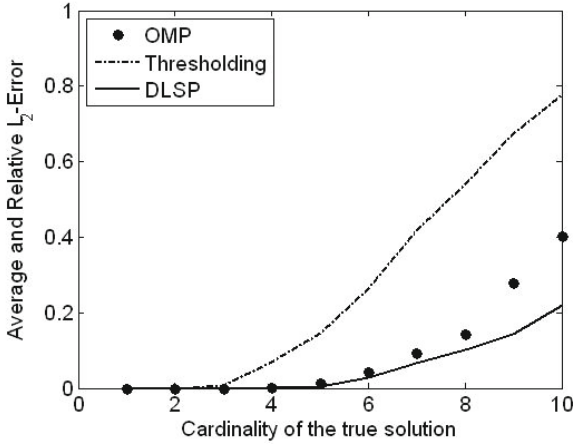


Fig. 3. Algorithms performance in terms of relative  $l_2$  recovery error

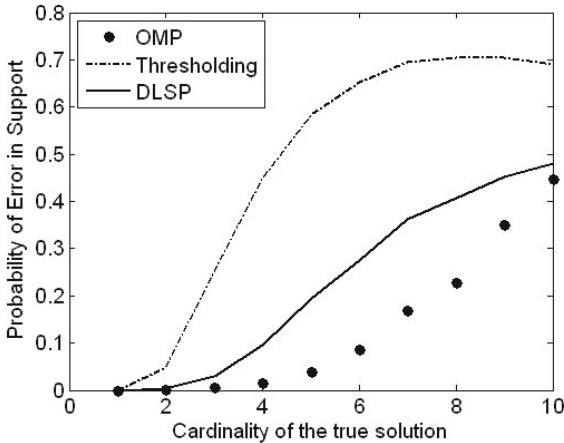


Fig. 4. Algorithms performance in terms of the success rate in detecting the true support

## 5 Conclusion

This paper presents a new pursuit algorithm Double Least-Squares Pursuit (DLSP) for sparse decomposition. This method finds the support of the solution and gets the non-zero values over this support by applying Least-Squares twice. Experimental results demonstrate that the proposed method is with more simple form, less time complexity and quite good performance.

Our future work will focus on deep theoretical analyze of DLSP which will help in understanding the results obtained in this paper and making more theoretical improvement.

**Acknowledgments.** This work was partly supported by the NNSF (National Natural Science Foundation) of China under the grant 61100098 and the Knowledge Innovation Program of the Chinese Academy of Sciences under the grant GY-110502.

## References

1. Elad, M.: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer (2010)
2. Natarajan, B.K.: Sparse approximate solutions to linear systems. *SIAM Journal on Computing* 24, 227–234 (1995)
3. Pati, Y.C., Rezaiifar, R., et al.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: 1993 Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers (1993)
4. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing* 41(12), 3397–3415 (1993)
5. Candes, E., Tao, T.: Decoding by linear programming. *IEEE Trans. Inf. Theory* 51, 4203–4215 (2005)
6. Candes, E., Romberg, J.: l1-magic: Recovery of sparse signals via convex programming. California Institute of Technology (2005), <http://www.acm.caltech.edu/l1magic/>
7. Cands, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics* 59, 1207–1223 (2006)
8. Candes, E., Tao, T.: Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory* 52(12), 5406–5425 (2006)
9. Sparse Representations for Signal Processing and Coding, <http://www.see.ed.ac.uk/~tblumens/Sparse/Sparse.html> (Visited March 22, 2012)
10. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
11. Temlyakov, V.N.: Weak greedy algorithms. *Advances in Computational Mathematics* 5, 173–187 (2000)
12. Cai, T.T., Lie, W.: Orthogonal Matching Pursuit for Sparse Signal Recovery with Noise. *IEEE Transactions on Information Theory* 57(7), 4680–4688 (2011)
13. Tong, Z.: Sparse Recovery with Orthogonal Matching Pursuit under Rip. *IEEE Transactions on Information Theory* 57(9), 6215–6221 (2011)