Journal of Mathematical Research with Applications
Nov., 2018, Vol. 38, No. 6, pp. 643–658
DOI:10.3770/j.issn:2095-2651.2018.06.010
Http://jmre.dlut.edu.cn

Piecewise Sparse Recovery via Piecewise Greedy Method

Yijun ZHONG, Chongjun LI*
School of Mathematical Sciences, Dalian University of Technology, Liaoning 116024*, P. R. China*

Abstract In some applications, there are signals with piecewise structure to be recovered. In this paper, we propose a piecewise OMP (P_OMP) method which aims to preserve the piecewise sparse structure (or the small-scaled entries) of piecewise signals. Besides the merits of OMP, the P_OMP, which is a generalization of the combination of CoSaMP and OMMP (Orthogonal Multi-matching Pursuit) on piecewise sparse recovery, possesses the advantages of comparable approximation error decay as CoSaMP with more relaxed sufficient condition and better recovery success rate. Moreover, the P_OMP algorithm recovers the piecewise sparse signal according to its piecewise structure, which results in better details preservation. Numerical experiments indicate that compared with CoSaMP, OMP, OMMP and BP methods, the P_OMP algorithm is more effective and robust for piecewise sparse recovery.

Keywords piecewise sparse; OMP; greedy algorithms

MR(2010) Subject Classification 90C25; 94A12

1. Introduction

In this paper, we consider recovering a sparse signal $\mathbf{x}^* \in \mathbf{R}^n$ from its noisy linear measurements

$$\mathbf{b} = A\mathbf{x}^* + \mathbf{e}, \tag{1.1}$$

where $\mathbf{b} \in \mathbf{R}^m$ is a measurement vector, $A \in \mathbf{R}^{m\times n}$ is a measurement matrix, and $\mathbf{e} \in \mathbf{N}(0, \sigma^2\mathbf{I}_n)$ is Gaussian noise. The sparse vector $\mathbf{x}^*$ has $s \le m < n$ nonzero entries.

The key of recovering a signal from its noisy measurement (1.1) is to find the support of the signal, i.e., find the set $\mathbf{S}$ satisfying $\mathrm{supp}(\mathbf{x}^*) = \mathbf{S}$, it is named as "exact support recovery". In some applications, the signal is indeed "piecewise sparse". To be general, we recover a sparse signal $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$ which is piecewise sparse structured by a partition of support set $\mathbf{S} = (S_i)_{i=1}^N$. Denote the corresponding partition of $\mathbf{D} = \{1, \ldots, n\}$ as $\mathbf{D} = (D_i)_{i=1}^N$. It is clear that $S_i \subseteq D_i$. Then we recover N sub-signals $\mathbf{x}_i$ ($\mathbf{x}_i \in \mathbf{R}^{n_i}$ is s_i-sparse vector on set D_i, where $s_i = |S_i|$) for $i = 1, \ldots, N$, respectively and simultaneously. We call this type of signal as "piecewise sparse" vector, denoted by $(s_1, \ldots, s_N)$-sparse vector. According to the piecewise structure of the signal $\mathbf{x}$, the measurement matrix A is also structured as $A = [A_1, \ldots, A_N]$

Received June 7, 2018; Accepted September 14, 2018
Supported by the National Natural Science Foundation of China (Grant Nos. 11471066; 11572081; 11871137) and the Fundamental Research Funds for the Central Universities (Grant No. QYWKC2018007).
* Corresponding author
E-mail address: chongjun@dlut.edu.cn (Chongjun LI)

where $A_i \in \mathbf{R}^{m\times n_i}$. Then the linear measurements (1.1) can be rewritten as

$$\mathbf{b} = \sum_{i=1}^{N} A_i \mathbf{x}_i^* + \epsilon.$$

Remark 1.1 It is necessary to claim that the piecewise sparse vector is quite different from block sparse vector mentioned in [1–4]. A block s-sparse vector $x = (x[1], \dots, x[N])^T$ is assumed to have at most s blocks with nonzero entries while each block $x[l]$ ($l = 1, \dots, N$) is not necessary sparse. A piecewise sparse vector $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^T$ is partitioned into N blocks and it is assumed that every $\mathbf{x}_i \in \mathbf{R}^{n_i}$ containing nonzero entries is sparse.

Denote $s = s_1 + \cdots + s_N$, the piecewise sparse vector $\mathbf{x}$ is also a global s-sparse vector. One approach for solving (1.1) is greedy method, which approximates the l^0 minimizing solution. One of the most popular greedy method is the orthogonal matching pursuit (OMP) as proposed in [5–7]. It iteratively adds components to the support of the approximation x^k whose correlation to the current residual is maximal. Many other greedy methods for sparse recovery have been proposed, for example, iterative hard thresholding (IHT) [8], stagewise OMP (StOMP) [9], regularized OMP (ROMP) [10, 11], compressive sampling matching pursuit (CoSaMP) [12], Orthogonal Multi-matching Pursuit (OMMP) (or named KOMP, MOMP, OSGA and GOMP) [13–17], subspace pursuit (SP) [18], iterative thresholding with inversion (ITI) [19], hard thresholding pursuit (HTP) [20] and many others. Another approach is convex relaxation which solves a convex program whose minimizer is known to approximate the target signal. Many algorithms have been proposed to complete the optimization, including interior-point methods [21, 22], projected gradient methods [23], and iterative thresholding [24]. Besides, combinatorial algorithms which acquire highly structured samples of the signal that support rapid reconstruction via group testing are also widely used. For example, Fourier sampling [25, 26], chaining pursuit [27], and HHS pursuit [28], as well as some algorithms in [29, 30].

Each type of algorithm described above has its native shortcomings. Many of the combinatorial algorithms are extremely fast–sublinear in the length of the target signal but they require a large number of somewhat unusual samples that may not be easy to acquire [12]. The convex relaxation tends to be computationally expensive when dealing with the case of large number of measurements. Compared with convex relaxation, greedy pursuits are better in their running time but worse in recovery accuracy. In particular, for recovering $(s_1, \dots, s_N)$-piecewise sparse signal in the noisy case, besides recovering the nonzero entries of signal, the convex relaxation methods may fail on some parts, such as recovering redundant entries. If one treats the $(s_1, \dots, s_N)$-piecewise sparse signal as a global s-sparse signal recovered by greedy methods, they may find "details" missing, i.e., recover the false entries instead of the exact smallest magnitudes. The major algorithmic challenge in noisy piecewise sparse recovery is to locate every element exactly and preserve the smallest magnitudes. Since greedy method requires the global sparsity level s as part of its input, they rarely recover redundant entries. Thus the greedy methods tend to perform better than the convex relaxation in terms of exact sparsity recovery.

Among these methods, one greedy method called compressive sampling matching pursuit

(CoSaMP) which is inspired by the work on ROMP, is attracting extensive attention since it incorporates ideas from the combinatorial algorithms to guarantee speed and to provide rigorous error bounds [12]. Unlike the simplest greedy algorithms, CoSaMP identifies many components during each iteration, which allows the algorithm to run faster for many types of signals. However, CoSaMP is at heart a greedy algorithm which may also fail on exact support recovery (select false small entries) for recovering piecewise sparse signals. In order to preserve the details of the signal and overcome the disadvantages of CoSaMP. We propose a piecewise_OMP (P_OMP) method to recover piecewise sparse signal, i.e., apply the main ideas of CoSaMP and OMMP to $\mathbf{x}_i$ for $i = 1, \ldots, N$. OMMP, which is a generalization of OMP, selects multiple atoms per iteration, enjoys the merit of less iterations compared to the OMP [13]. OMMP was also studied in [14-17], named as KOMP, MOMP, OSGA and GOMP. The CoSaMP first locates the largest $2s$ entries by proxy, then sorts the components of the vector solved by the least-square by magnitude and selects the first s entries [12]. Different from the CoSaMP, the P_OMP algorithm locates the largest $(s_1 + \cdots + s_N)$ entries by piecewise proxy which is inspired by the idea from OMMP, and selects the first s_i entries for $i = 1, \ldots, N$ by piecewise pruning at the last step. The piecewise proxy and pruning ensure that the P_OMP algorithm finds the exact smallest magnitudes.

Organization. The rest of the paper is organized as follows. In Section 2 we give some preliminaries for the analysis of P_OMP. We provide the P_OMP algorithm and state the major theorems in more detail about P_OMP in Section 3. Finally, Section 4 shows the numerical performance of P_OMP in comparison with CoSaMP, OMP and BP methods.

2. Preliminaries

It is shown in [31] that when the sampling matrix satisfies the restricted isometry inequalities, it has several other properties that one may require in the proof of the CoSaMP algorithm. In this section we present these properties together with the CoSaMP algorithm and the OMMP algorithm.

Definition 2.1 ([31]) *RIC: The r-th restricted isometry constant of a matrix A is the least number δ_r for which*

$$(1-\delta_r)\|\mathbf{x}\|_2^2 \le \|A\mathbf{x}\|_2^2 \le (1+\delta_r)\|\mathbf{x}\|_2^2 \quad \textit{whenever} \quad \|\mathbf{x}\|_0 \le r.$$

Here $\|\cdot\|_2$ represents the l^2 vector norm.

Proposition 2.2 ([12]) *Suppose A has RIC δ_r. Let T be a set of r indices or fewer. A_T is the column sub–matrix of A whose columns are listed in the set T. Then*

$$\|A_T^T\mathbf{x}\|_2 \le \sqrt{1+\delta_r}\|\mathbf{x}\|_2, \qquad \|A_T^\dagger\mathbf{x}\|_2 \le \frac{1}{\sqrt{1-\delta_r}}\|\mathbf{x}\|_2,$$
$$\|A_T^TA_T\mathbf{x}\|_2 \lesseqqgtr (1\pm\delta_r)\|\mathbf{x}\|_2,$$
$$\|(A_T^TA_T)^{-1}\mathbf{x}\|_2 \lesseqqgtr \frac{1}{1\pm\delta_r}\|\mathbf{x}\|_2$$

where the last two statements contain an upper and lower bound, depending on the sign chosen.

Proposition 2.3 ([12]) *Suppose A has RIC δ_r. Let T_1 and T_2 be disjoint sets of indices whose combined cardinality does not exceed r. Then $\|A_{T_1}^T A_{T_2}\| \le \delta_r$.*

Corollary 2.4 ([12]) *Suppose A has RIC δ_r. Let T be a set of indices, and let $\mathbf{x}$ be a vector. Provided that $r \ge |T \cup \operatorname{supp}(\mathbf{x})|$, $\|A_T^T A \cdot \mathbf{x}|_{T^c}\|_2 \le \delta_r \|\mathbf{x}|_{T^c}\|_2$.*

Corollary 2.5 ([12]) *Let c and r be positive integers. Then $\delta_{cr} \le c \cdot \delta_{2r}$.*

Assumption 2.6 ([12]) *The standing assumptions for CoSaMP are:*

- *The sparsity level s is fixed.*
- *The $m \times n$ matrix A has RIC $\delta_{4s} \le 0.1$.*
- *The vector $\mathbf{b} = A\mathbf{x} + \mathbf{e}$.*

Algorithm 1: CoSaMP Recovery Algorithm ([12])
Input: Matrix A, noisy vector $\mathbf{b}$, sparsity level s
Output: An s-sparse approximation $\mathbf{x}$ of the target signal
Initialization: $\mathbf{x}^0 \leftarrow 0$, $\mathbf{res} \leftarrow \mathbf{b}$, $k \leftarrow 0$
Repeat
$k \leftarrow k + 1$
$\mathbf{y} = A^T \mathbf{res}$ (Form signal proxy)
$\Omega \leftarrow \operatorname{supp}(\mathbf{y}_{2s})$ (Identify $2s$ large components)
$\Lambda \leftarrow \Omega \bigcup \operatorname{supp}(\mathbf{x}^{k-1})$ (Merge supports)
$\tilde{\mathbf{x}}|_\Lambda \leftarrow A_\Lambda^\dagger \mathbf{b}$, $\tilde{\mathbf{x}}|_{\Lambda^c} \leftarrow 0$ (Signal estimation by least-squares)
$\mathbf{x}^k \leftarrow \tilde{\mathbf{x}}_s$ (Select the first s large entries)
$\mathbf{res} \leftarrow \mathbf{b} - A\mathbf{x}^k$ (Update current residual)
Until Stopping criterion *true*

Theorem 2.7 ([12]) *Assume the Assumption 2.6 hold. Then for each $k \ge 0$, the signal approximation $\mathbf{x}^k$ is s-sparse, and*

$$\|\mathbf{x}^* - \mathbf{x}^{k+1}\|_2 \le 0.5\|\mathbf{x}^* - \mathbf{x}^k\|_2 + 7.5\|\mathbf{e}\|_2. \tag{2.1}$$

In particular,

$$\|\mathbf{x}^* - \mathbf{x}^k\|_2 \le 2^{-k}\|\mathbf{x}^*\|_2 + 15\|\mathbf{e}\|_2.$$

The theorem in [12] states that each iteration of the CoSaMP algorithm reduces the approximation error by a constant factor, while adding a small multiple of the noise.

The orthogonal multi-matching pursuit (OMMP) in [13] is a natural extension of the orthogonal matching pursuit (OMP). The main difference between OMP and OMMP (M) is that OMMP (M) selects M atoms per iteration, while OMP only adds one atom to the optimal atom set. Results in [14–17] show that when RIP constant $\delta = \mathrm{O}(\sqrt{M/s})$, OMMP (M) can recover the s-sparse signal in s iterations. In particular, results in [13] show that when the measurement matrix A satisfies $(9s, \frac{1}{10})$-RIP, there exists an absolute constant $M_0 \leq 8$ so that OMMP (M_0) can recover s-sparse signal within s iterations.

Algorithm 2: OMMP Recovery Algorithm ([13])
Input: Matrix A, noisy vector $\mathbf{b}$,
candidate number M for each step,
stopping iteration index H,
initial feature set $\Lambda_0 \in \{1, \ldots, n\}$
Output: An s-sparse approximation $\mathbf{x}$ of the target signal
Initialization: $\mathbf{x}^0 = \arg \min_{\mathbf{z}:\mathrm{supp}\mathbf{z} \subset \Lambda_0} \|\mathbf{b} - A\mathbf{z}\|_2$,
$\mathbf{res} \leftarrow \mathbf{b} - A\mathbf{x}^0, k = 0$
While $k < H$ **do**
$\mathbf{y} = A^T \mathbf{res}$ (match)
$\Omega \leftarrow \mathrm{supp}(\mathbf{y}_M)$ (Identify M largest components)
$\Lambda \leftarrow \Omega \bigcup \mathrm{supp}(\mathbf{x}^{k-1})$ (Merge supports)
$\tilde{\mathbf{x}}|_\Lambda \leftarrow A_\Lambda^\dagger \mathbf{b}$, $\tilde{\mathbf{x}}|_{\Lambda^c} \leftarrow 0$ (Signal estimation by least-squares)
$\mathbf{res} \leftarrow \mathbf{b} - A\mathbf{x}^k$ (Update current residual)
$k = k + 1$
end while

Notation. For convenience, let $\mathbf{S} = \mathrm{supp}(x^*)$ and $\mathbf{S}^c$ be its complement, i.e., $\mathbf{S}^c = \{i : x_i^* = 0\}$. $A_\mathbf{S}$ denotes the submatrix of A formed by the columns of A in $\mathbf{S}$, which are assumed to be linearly independent. Similarly define $A_{\mathbf{S}^c}$ so that $[A_\mathbf{S} \ \ A_{\mathbf{S}^c}] = A$. Denote $s = |\mathbf{S}| = |\mathrm{supp}(x^*)|$, $s_i = |S_i|$. Define $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{y}$. The pseudoinverse of a tall, full-rank matrix B is defined as $B^\dagger = (B^T B)^{-1} B^T$.

3. Analysis of P_OMP iteration

Before proceeding the analysis of P_OMP iteration, we give the hypotheses for P_OMP.

- The sparsity level s and piecewise sparsity level s_i for $i = 1, \ldots, N$ are fixed.
- The matrix A has RIC constant $\delta_{3s} \leq 0.1$.
- The signal $\mathbf{x}^*$ is s-sparse and sub-signal $\mathbf{x}_i^*$ is x_i-sparse.

It is obvious that the major difference between P_OMP and CoSaMP lies in the Identification and Pruning steps. It is noticed that we locate s entries of a vector in step Identification similar to the first step of OMMP. However, we locate the largest s_i entries of $\mathbf{x}_i$ $(i = 1, \ldots, N)$ and form all the components together as the $s = (s_1 + \cdots + s_N)$ entries, which is different from both CoSaMP and OMMP. Instead of selecting s-largest components of least-square solution, we sort the components of the piecewise vector $\tilde{\mathbf{x}} = (\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N)^T$ solved by least-square by magnitudes of each piece. The piecewise pruning process make progress in preserving small-scaled entries in each piece from false chosen cause of noise perturbation.

It was shown in [12] that compared with s iteration in OMP, the iteration cost of CoSaMP is at most s. The iteration cost of P_OMP is at most $s_{max} = \max\{s_1, \ldots, s_N\} \le s$.

Algorithm 3: P_OMP Recovery Algorithm
Input: Matrix A, noisy vector $\mathbf{b}$,
sparsity level s, piecewise sparsity level s_i
for $i = 1, \ldots, N$
Output: An $(s_1, \ldots, s_N)$-piecewise sparse approximation
$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$ of the target signal
Initialization: $\mathbf{x}^0 \leftarrow 0$, $\mathbf{res} \leftarrow \mathbf{b}$, $k \leftarrow 0$
Repeat
$k \leftarrow k + 1$
For $i = 1, \ldots, N$ **in parallel**
$\mathbf{y}_i = A_i^T \mathbf{res}$ (Form piecewise signal proxy)
$\Omega_i \leftarrow \mathrm{supp}(\mathbf{y}_{i(s_i)})$ (Identify s_i large components of every piece)
$\Lambda \leftarrow \bigcup \Omega_i \bigcup \mathrm{supp}(\mathbf{x}^{k-1})$ (Merge supports)
$\tilde{\mathbf{x}}|_\Lambda \leftarrow A_\Lambda^\dagger \mathbf{b}$, $\tilde{\mathbf{x}}|_{\Lambda^c} \leftarrow 0$ (Signal estimation by least-squares)
$\mathbf{x}^k \leftarrow (\tilde{\mathbf{x}}_{1(s_1)}, \ldots, \tilde{\mathbf{x}}_{N(s_N)})^T$ (Select the first x_i for each
piece and form together)
$\mathbf{res} \leftarrow \mathbf{b} - A\mathbf{x}^k$ (Update surrent residual)
Until Stopping criterion *true*

Theorem 3.1 *Assume the above hypotheses hold. Then for each $k \ge 0$, the signal approximation $\mathbf{x}^k$ recovered by P_OMP is $(s_1, \ldots, s_N)$-piecewise sparse, and*

$$\|\mathbf{x}^* - \mathbf{x}^{k+1}\|_2 \le 0.75\|\mathbf{x}^* - \mathbf{x}^k\|_2 + 7.5\|\mathbf{e}\|_2. \tag{3.1}$$

It is obvious that (3.1) is comparable to the results of (2.1). We proceed with our proof of the theorem by estimating each step in P_OMP similar to the proof in [12].

Residual. For an iteration $k \ge 1$, we use $\mathbf{res}$ to represent the residual vector

$$\mathbf{res} = \mathbf{b} - A\mathbf{x}^{k-1}$$

which in iteration $k-1$ (or at the beginning of the iteration). Define $\mathbf{r} = \mathbf{x}^* - \mathbf{x}^{k-1}$ as the part of the signal we have not yet recovered. It is obvious the vector $\mathbf{r}$ is $2s$-sparse since both $\mathbf{x}^*$ and $\mathbf{x}^{k-1}$ are always s-sparse. Then the residual vector $\mathbf{res}$ can be rewritten as

$$\mathbf{res} = A(\mathbf{x}^* - \mathbf{x}^{k-1}) + \mathbf{e} = A\mathbf{r} + \mathbf{e}.$$

Piecewise Identification. We apply the identification of OMMP to each piece, respectively and simultaneously to obtain the piecewise identification of P_OMP.

Lemma 3.2 (Piecewise Identification) *The set $\Omega = \bigcup \Omega_i = \bigcup \mathrm{supp}(\mathbf{y}_{i(s_i)})$ contains at most s*

indices, and

$$\|\mathbf{r}|_{\Omega^c}\|_2 \le \frac{2\delta_{3s}+\delta_{2s}}{1-\delta_{2s}}\|\mathbf{r}\|_2 + \frac{2\sqrt{1+\delta_{2s}}+\sqrt{1+\delta_{2s}}}{1-\delta_{2s}}\|\mathbf{e}\|_2. \tag{3.2}$$

The (3.2) is a relatively large bound on $\|\mathbf{r}|_{\Omega^c}\|_2$ derived from the following inequality:

$$\|\mathbf{r}|_{\Omega^c}\|_2 \le \frac{2\sum_{i=1}^{N}\delta_{s_i+2s}+N\delta_{2s}}{N(1-\delta_{2s})}\|\mathbf{r}\|_2 + \frac{2\sum_{i=1}^{N}\sqrt{1+\delta_{s_i}}+N\sqrt{1+\delta_{2s}}}{N(1-\delta_{2s})}\|\mathbf{e}\|_2, \tag{3.3}$$

where (3.2) is a relaxed bound of (3.3).

Proof Define a total proxy $\mathbf{y}=(\mathbf{y}_1,\dots,\mathbf{y}_N)^T$ with piecewise proxy $\mathbf{y}_i=A_i^T\mathbf{res}$. The piecewise identification set $\Omega=\bigcup\Omega_i$ contains s components, where Ω_i is the index set in accordance with selecting the largest s_i entries of $\mathbf{y}_i$. Similar to the proof of [12, Lemma 4.2], we use the notation $R=\operatorname{supp}(\mathbf{r})$ which contains at most $2s$ elements, notation $R_i=\operatorname{supp}(\mathbf{x}_i^*-\mathbf{x}_i^{k-1})$ which contains at most $2s_i$ elements, and the notation $\hat{\Omega}_i$ which is the index set corresponding to selecting the largest $2s_i$ entries of $\mathbf{y}_i$. Thus

$$\begin{aligned}\|\mathbf{y}|_R\|_2 &= \Big(\sum_{i=1}^{N}\|\mathbf{y}_i|_{R_i}\|_2^2\Big)^{1/2} \le \Big(\sum_{i=1}^{N}\|\mathbf{y}_i|_{\hat{\Omega}_i}\|_2^2\Big)^{1/2}\\ &\le \Big(\sum_{i=1}^{N}2\|\mathbf{y}_i|_{\Omega_i}\|_2^2\Big)^{1/2} = \sqrt{2}\|\mathbf{y}|_\Omega\|_2.\end{aligned}$$

By squaring this inequality and canceling the terms in $R\bigcap\Omega$, we obtain that

$$\|\mathbf{y}|_{R\setminus\Omega}\|_2 \le \sqrt{2}\|\mathbf{y}|_{\Omega\setminus R}\|_2.$$

Next we use the RIC to provide bounds on both sides of the above inequality.

We start with the most strict bound (3.3) which is equipped with δ_{s_i} for $i=1,\dots,N$. It is observed that the set $R\setminus\Omega$ contains at most $2s$ elements and the set $\Omega\setminus R$ contains at most s elements. Therefore, we apply the Proposition 2.2 and Corollary 2.4 to obtain

$$\begin{aligned}\|\mathbf{y}|_{\Omega\setminus R}\|_2 &= (\|\mathbf{y}_1|_{\Omega\setminus R}\|_2^2+\cdots+\|\mathbf{y}_N|_{\Omega\setminus R}\|_2^2)^{1/2}\\ &\le \|\mathbf{y}_1|_{\Omega\setminus R}\|_2+\cdots+\|\mathbf{y}_N|_{\Omega\setminus R}\|_2\\ &= \|A_{1(\Omega\setminus R)}^T(A\mathbf{r}+\mathbf{e})\|_2+\cdots+\|A_{N(\Omega\setminus R)}^T(A\mathbf{r}+\mathbf{e})\|_2\\ &\le \sum_{i=1}^{N}(\|A_{i(\Omega\setminus R)}^T A\mathbf{r}\|_2+\|A_{i(\Omega\setminus R)}^T\mathbf{e}\|_2)\\ &\le \sum_{i=1}^{N}(\delta_{s_i+2s}\|\mathbf{r}\|_2+\sqrt{1+\delta_{s_i}}\|\mathbf{e}\|_2)\end{aligned}$$

by using $A_{i(\Omega\setminus R)}=A_{\Omega_i\setminus R}$.

Likewise, the set $R\setminus\Omega$ contains at most $2s$ elements, and we obtain the following inequality using Proposition 2.2 and Corollary 2.4

$$\begin{aligned}
\|\mathbf{y}|_{R\setminus\Omega}\|_2 &= (\|\mathbf{y}_1|_{R\setminus\Omega}\|_2^2+\cdots+\|\mathbf{y}_N|_{R\setminus\Omega}\|_2^2)^{1/2} \ge \frac{\sqrt{2}}{2}\sum_{i=1}^N \|\mathbf{y}_i|_{R\setminus\Omega}\|_2\\
&= \frac{\sqrt{2}}{2}\sum_{i=1}^N (\|A_{i(R\setminus\Omega)}^T A\cdot \mathbf{r}|_{R\setminus\Omega} + A_{i(R\setminus\Omega)}^T A\cdot\mathbf{r}|_{\Omega} + A_{i(R\setminus\Omega)}\mathbf{e}\|_2)\\
&\ge \frac{\sqrt{2}}{2}\sum_{i=1}^N (\|A_{i(R\setminus\Omega)}^T A\cdot \mathbf{r}|_{R\setminus\Omega}\|_2 - \|A_{i(R\setminus\Omega)}^T A\cdot\mathbf{r}|_{\Omega}\|_2 - \|A_{i(R\setminus\Omega)}\mathbf{e}\|_2)\\
&\ge \frac{\sqrt{2}}{2}\sum_{i=1}^N ((1-\delta_{2s})\|\mathbf{r}|_{R\setminus\Omega}\|_2 - \delta_{2s}\|\mathbf{r}\|_2 - \sqrt{1+\delta_{2s}}\|\mathbf{e}\|_2)\\
&= \frac{\sqrt{2}}{2}N((1-\delta_{2s})\|\mathbf{r}|_{R\setminus\Omega}\|_2 - \delta_{2s}\|\mathbf{r}\|_2 - \sqrt{1+\delta_{2s}}\|\mathbf{e}\|_2).
\end{aligned}$$

Since $\mathbf{r}$ is supported on R, then $\mathbf{r}|_{R\setminus\Omega} = \mathbf{r}|_{\Omega^c}$. Thus, combining the above three inequalities, we have (3.3). Applying $s_i = s$ for $i = 1,\ldots,N$ we obtain (3.2). □

Similar to [12, Lemmas 4.3 and 4.4], we obtain the same conclusion for the P_OMP algorithm in the following steps.

Support Merger. In the P_OMP algorithm we merge the support of current signal approximation $\mathbf{x}^{k-1}$ with the newly identified set $\Omega = \bigcup \Omega_i$, thus the set $\Lambda = \Omega \bigcup \operatorname{supp}(\mathbf{x}^{k-1})$ contains at most $2s$ indices and $\|\mathbf{x}^*|_{\Lambda^c}\|_2 \le \|\mathbf{r}|_{\Lambda^c}\|_2$.

Piecewise Estimation. In this step, we solve a least-squares problem similar to all greedy methods in the set Λ.

Lemma 3.3 *Denote $\hat{\mathbf{x}}$ as $\hat{\mathbf{x}}|_\Lambda = (A_\Lambda^T A_\Lambda)^{-1}A_\Lambda^T b$, $\hat{\mathbf{x}}|_{\Lambda^c} = \mathbf{0}$. Then*

$$\|\mathbf{x}^* - \hat{\mathbf{x}}\|_2 \le (1+\frac{\delta_{3s}}{1-\delta_{2s}})\|\mathbf{x}^*|_{\Lambda^c}\|_2 + \frac{1}{\sqrt{1-\delta_{2s}}}\|\mathbf{e}\|_2.$$

Proof Note that $\|\mathbf{x}^* - \hat{\mathbf{x}}\|_2 \le \|\mathbf{x}^*|_{\Lambda^c}\|_2 + \|\mathbf{x}^*|_\Lambda - \hat{\mathbf{x}}|_\Lambda\|_2$. By computation:

$$\begin{aligned}
\|\mathbf{x}^*|_\Lambda - \hat{\mathbf{x}}|_\Lambda\|_2 &= \|\mathbf{x}^*|_\Lambda - A_\Lambda^\dagger(A\cdot\mathbf{x}^*|_\Lambda + A\cdot\mathbf{x}^*|_{\Lambda^c} + \mathbf{e})\|_2\\
&= \|A_\Lambda^\dagger(A\cdot\mathbf{x}^*|_{\Lambda^c}+\mathbf{e})\|_2\\
&\le \|(A_\Lambda^T A_\Lambda)^{-1}A_\Lambda^T A\cdot\mathbf{x}^*|_{\Lambda^c}\|_2 + \|A_\Lambda^\dagger \mathbf{e}\|_2.
\end{aligned}$$

The set Λ contains at most $2s$ entries, thus by using Proposition 2.2 and Corollary 2.4 we obtain that

$$\begin{aligned}
\|\mathbf{x}^*|_\Lambda - \hat{\mathbf{x}}|_\Lambda\|_2 &\le \frac{1}{1-\delta_{2s}}\|A_\Lambda^T A\cdot\mathbf{x}^*|_{\Lambda^c}\|_2 + \frac{1}{\sqrt{1-\delta_{2s}}}\|\mathbf{e}\|_2\\
&\le \frac{\delta_{3s}}{1-\delta_{2s}}\|\mathbf{x}^*|_{\Lambda^c}\|_2 + \frac{1}{\sqrt{1-\delta_{2s}}}\|\mathbf{e}\|_2.
\end{aligned}$$

Then we obtain the inequality

$$\|\mathbf{x}^* - \hat{\mathbf{x}}\|_2 \le (1+\frac{\delta_{3s}}{1-\delta_{2s}})\|\mathbf{x}^*|_{\Lambda^c}\|_2 + \frac{1}{\sqrt{1-\delta_{2s}}}\|\mathbf{e}\|_2. \quad \Box$$

Piecewise Pruning. The final step of P_OMP algorithm is to prune the current estimation $\hat{\mathbf{x}} = (\hat{\mathbf{x}}_1,\ldots,\hat{\mathbf{x}}_N)^T$ to its largest piecewise s terms, i.e., prune each $\hat{\mathbf{x}}_i$ to its largest s_i terms for

$i = 1, \ldots, N$. The pruned approximation $\hat{\mathbf{x}}_s = (\hat{\mathbf{x}}_{s_1}, \ldots, \hat{\mathbf{x}}_{s_N})^T$, where $\hat{\mathbf{x}}_{s_i}$ is the vector with pruning $\hat{\mathbf{x}}_i$ to its largest s_i terms, satisfies

$$\|\mathbf{x}^* - \hat{\mathbf{x}}_s\|_2 \le 2\|\mathbf{x}^* - \hat{\mathbf{x}}\|_2.$$

whose proof can be found in [12].

Proof of Theorem 3.1 We now complete the proof of Theorem 3.1. After the five procedures described above of P_OMP, the algorithm reachs a new approximation $\mathbf{x}^k = \hat{\mathbf{x}}_s = (\hat{\mathbf{x}}_{s_1}, \ldots, \hat{\mathbf{x}}_{s_N})^T$, which is $(s_1, \ldots, s_N)$-piecewise sparse. Using the lemmas we have provided:

$$\begin{aligned}
\|\mathbf{x}^* - \mathbf{x}^k\|_2 &= \|\mathbf{x}^* - \hat{\mathbf{x}}_s\|_2 \le 2\|\mathbf{x}^* - \hat{\mathbf{x}}\|_2 \\
&\le 2 \cdot ((1 + \frac{\delta_{3s}}{1 - \delta_{2s}})\|\mathbf{x}^*|_{\Lambda^c}\|_2 + \frac{1}{\sqrt{1 - \delta_{2s}}}\|\mathbf{e}\|_2) \\
&\le 2 \cdot ((1 + \frac{\delta_{3s}}{1 - \delta_{2s}})\|\mathbf{r}^*|_{\Lambda^c}\|_2 + \frac{1}{\sqrt{1 - \delta_{2s}}}\|\mathbf{e}\|_2) \\
&\le 2(1 + \frac{\delta_{3s}}{1 - \delta_{2s}})\frac{2\delta_{3s} + \delta_{2s}}{1 - \delta_{2s}}\|\mathbf{r}\|_2 + 2\frac{1}{\sqrt{1 - \delta_{2s}}}\frac{2\sqrt{1 + \delta_{2s}} + \sqrt{1 + \delta_{2}s}}{1 - \delta_{2s}}\|\mathbf{e}\|_2 \\
&< 0.75\|\mathbf{r}\|_2 + 7.5\|\mathbf{e}\|_2
\end{aligned}$$

by invoking the assumption that $\delta_s \le \delta_{2s} \le \delta_{3s} \le 0.1$. Then we complete the proof. □

4. Numerical experiments

Since the P_OMP algorithm is at heart a piecewise greedy algorithm. In this section, we show the numerical performance of the P_OMP algorithm for recovering piecewise sparse signals, compared with the CoSaMP, OMP, OMMP and BP algorithms for recovering the same signal as global sparse vector.

It is worth noting that P_OMP is proposed aiming at recovering piecewise sparse signals, especially signals with both large and very small scaled elements. For example, the signal $\mathbf{x}$ is $(|S_1| + |S_2|)$-sparse, where both large entries and very small entries lie in S_1 and S_2 (or S_1 contains large elements and small components lie in S_2). In this part, the $m \times n$ matrix A is the partial discrete cosine transform (DCT) matrix with m rows chosen randomly from the $n \times n$ DCT matrix. n is the dimension of the signal and s is the total number of nonzero entries of $\mathbf{x}^*$. Similar to the notation of "distinct weights" in [32], we suppose that the support set $\mathbf{S}$ is partitioned by $\mathbf{S} = (S_i)_{i=1}^N$ with $|S_i| = \rho_i|S|$, i.e., $s_i = \rho_i s$. Thus the structure of the piecewise sparse signal is denoted by $(\rho_1, \ldots, \rho_N)$. In order to demonstrate the effectiveness of support recovery, we use the "distance" between two support sets which is measured by [33]:

$$D(S1, S2) = \frac{\max\{|S1|, |S2|\} - |S1 \cap S2|}{\max\{|S1|, |S2|\}}. \tag{4.1}$$

which is a useful tool for estimating exact support recovery. A exact support recovery means the distance between $\tilde{\mathbf{S}}$ recovered by algorithm and true support set $\mathbf{S}$ is 0.

In our simulations, the following algorithms are considered.

(1) P_OMP algorithm ($N = 2$).

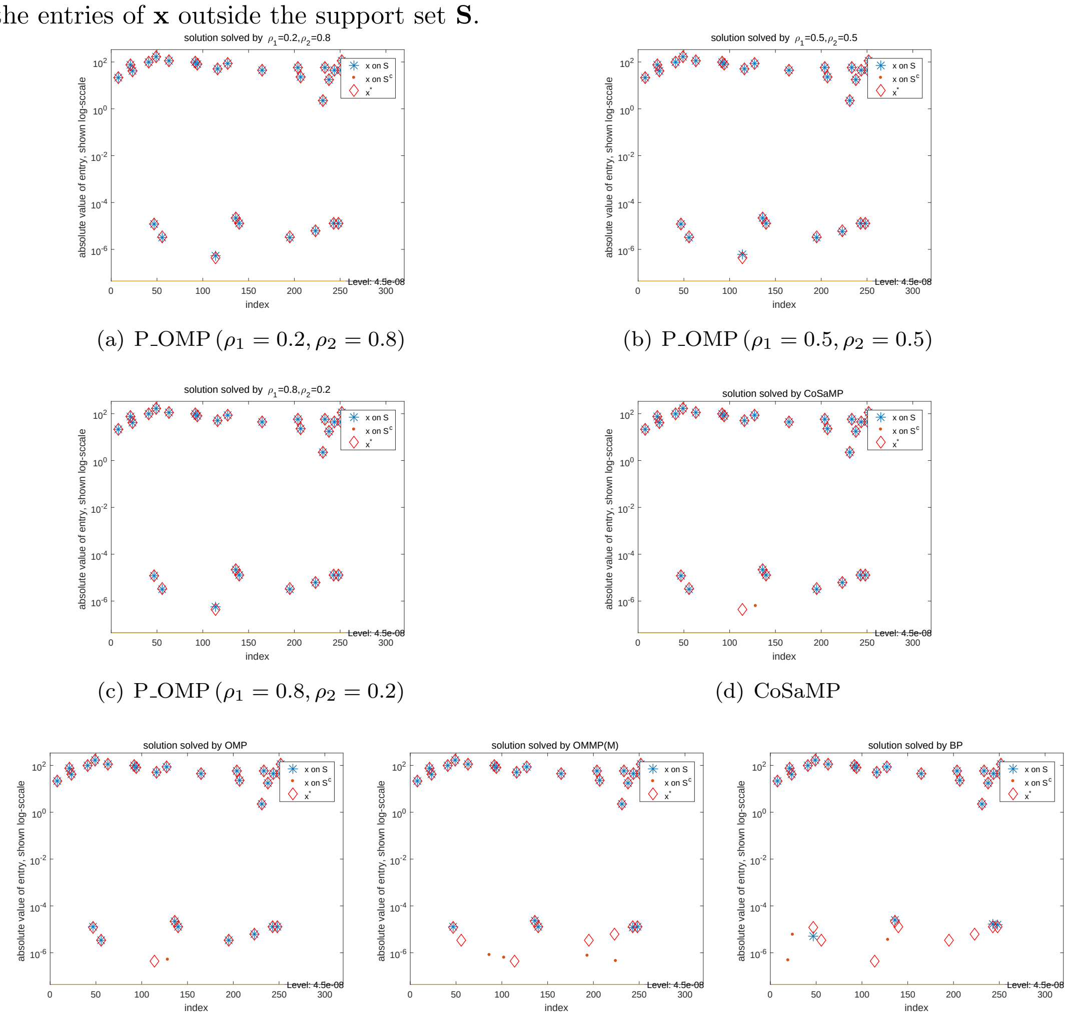
(2) CoSaMP algorithm (http://www.cmc.edu/pages/faculty/DNeedell/index.html).
(3) OMP algorithm.
(4) OMMP ($M = \sqrt{s}$) algorithm, which is also equivalent to GOMP algorithm.
(5) LP technique for solving l^1-minimization (http://cvxr.com/cvx/).

Example 4.1 Random data test 1 (Recoverability on randomly generated problems). For our first set of experiments, we test these algorithms for solving several problems including a set of small-scale components, including "pathological" problems where the magnitudes of the nonzero entries of the exact solutions lie in a large range, i.e., the largest magnitudes are significantly larger than the smallest magnitudes. The measurement noise $\mathbf{e}$ is i.i.d. Gaussian with mean zero and standard deviation $0.1\mathbf{x}^*_{\min}$. The information of the dataset and recovery numerical results are displayed in Table 1 and Figures 1–3. We set $N = 2$ and vary the size of ρ_1 and ρ_2 while maintaining that $\rho_1 + \rho_2 = 1$.

Problem Structure (m, n, s)	Algorithm	$D(\tilde{\mathbf{S}}, \mathbf{S})$	Running Time	Num. of False
Problem 1 $(128, 256, 28)$	P_OMP ($\rho_1 = 0.2, \rho_2 = 0.8$)	0	0.035	0
	P_OMP ($\rho_1 = 0.5, \rho_2 = 0.5$)	0	0.021	0
	P_OMP ($\rho_1 = 0.8, \rho_2 = 0.2$)	0	0.033	0
	CoSaMP	0.0357	0.043	1
	OMP	0.0357	0.011	1
	OMMP ($M = \sqrt{s}$)	0.1428	0.001	4
	BP	0.1786	0.102	3
Problem 2 $(128, 512, 28)$	P_OMP ($\rho_1 = 0.2, \rho_2 = 0.8$)	0	0.032	0
	P_OMP ($\rho_1 = 0.5, \rho_2 = 0.5$)	0	0.021	0
	P_OMP ($\rho_1 = 0.8, \rho_2 = 0.2$)	0	0.032	0
	CoSaMP	0.0536	0.013	15
	OMP	0	0.004	0
	OMMP ($M = \sqrt{s}$)	0	0.009	0
	BP	0.7811	0.102	100
Problem 3 $(128, 512, 28)$	P_OMP ($\rho_1 = 0.2, \rho_2 = 0.8$)	0	0.033	0
	P_OMP ($\rho_1 = 0.5, \rho_2 = 0.5$)	0	0.020	0
	P_OMP ($\rho_1 = 0.8, \rho_2 = 0.2$)	0	0.034	0
	CoSaMP	0.0357	0.806	1
	OMP	0.0357	0.006	1
	OMMP ($M = \sqrt{s}$)	0.0357	0.013	1
	BP	0.7811	0.039	100

Table 1 Comparison of results by the P_OMP, CoSaMP, OMP, OMMP and BP methods for Example 4.1 (the size of the dataset is described as (m, n, s) where the measurement matrix $A \in \mathbf{R}^{m\times n}$ and piecewise sparse signal $\mathbf{x}^*$ has totally s nonzero entries

We show the magnitude and location of true signal $\mathbf{x}^*$ and approximated piecewise sparse signal $\mathbf{x}$ recovered by algorithms in Figures 1–3 with x-axis denoting the index of the signal from 1 to n, and y-axis representing the absolute value of nonzero entries in log–scale of $\mathbf{x}^*$ and $\mathbf{x}$. In the plots of Figurers 1–3, the red diamonds represent the true vector $\mathbf{x}^*$, the blue stars are the nonzero entries of vector $\mathbf{x}$ on the true support $\mathbf{S}$ solved by algorithms while the red dots show the entries of $\mathbf{x}$ outside the support set $\mathbf{S}$.



(a) P_OMP ($\rho_1 = 0.2, \rho_2 = 0.8$)

(b) P_OMP ($\rho_1 = 0.5, \rho_2 = 0.5$)

(c) P_OMP ($\rho_1 = 0.8, \rho_2 = 0.2$)

(d) CoSaMP

(e) OMP

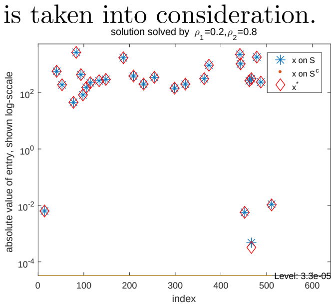
(f) OMMP ($M = \sqrt{s}$)

(g) BP

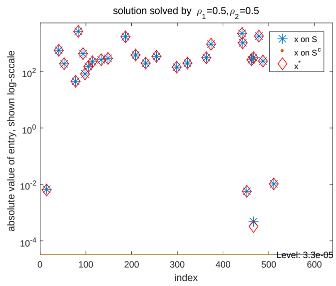
Figure 1 Recovery results of P_OMP, CoSaMP, OMP, OMMP and BP methods on Problem 1

From Table 1, the superiority of P_OMP is obvious on exact support recovery. It is observed from Figurers 1–3 that signal with same global sparsity and different piecewise structure can be exactly support recovered by P_OMP while CoSaMP, OMP, OMMP and BP may always lose the smallest magnitude and recover false entries.

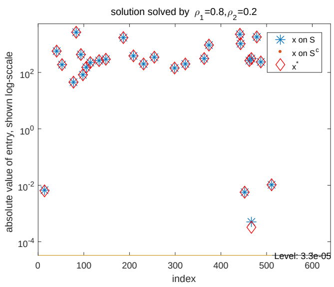
Remark 4.2 It is shown in Example 4.1 that P_OMP may produce different results for different piecewise structure. It seems that P_OMP with ($\rho_1 = \rho_2 = 0.5$) runs faster than other structures. In our future work, selection on piecewise structure for P_OMP in order to obtain best recovery results is taken into consideration.



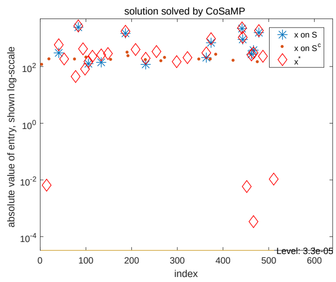
(h) P_OMP ($\rho_1 = 0.2, \rho_2 = 0.8$)



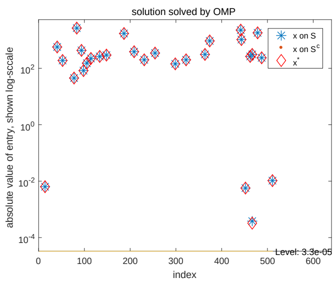
(i) P_OMP ($\rho_1 = 0.5, \rho_2 = 0.5$)



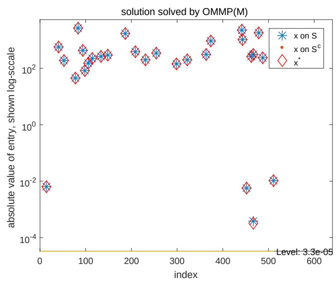
(j) P_OMP ($\rho_1 = 0.8, \rho_2 = 0.2$)



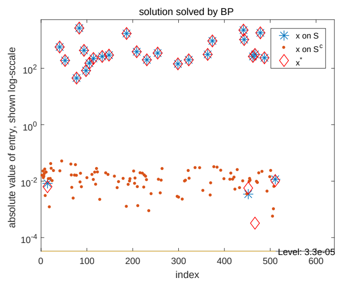
(k) CoSaMP



(l) OMP



(m) OMMP ($M = \sqrt{s}$)

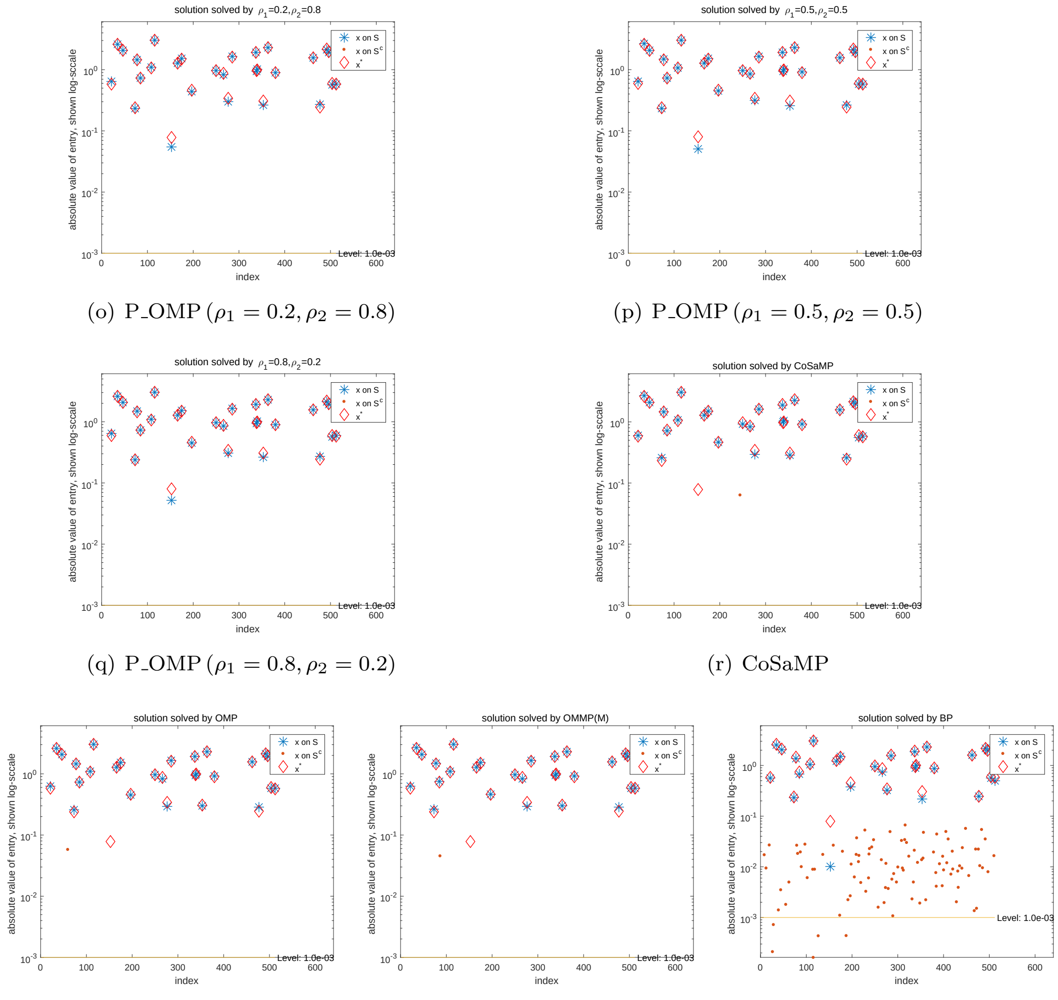


(n) BP

Figure 2 Recovery results of P_OMP, CoSaMP, OMP, OMMP and BP methods on Problem 2

Example 4.3 Random data Test 2 (Success Frequency). In this example, we test these algorithms by several problems with the same settings as difficult problem settings in [34]. We generate these problems in the same way as "CalTechTest" problems in [34] and run 500 times for each setting in order to obtain a frequency result of the comparison. Figure 4 shows the plot of these problems. We generate random piecewise sparse signals with global sparsity from 10 to 70 in order to test the robustness of our algorithm. We display the success rate in Figure 5.

We define the "succeed" when the distance between the support set and the true support set is 0, i.e., exact support recovery. The plots in Figures 5–7 show that the P_OMP algorithm always performs better in terms of success recovery rate than other algorithms.

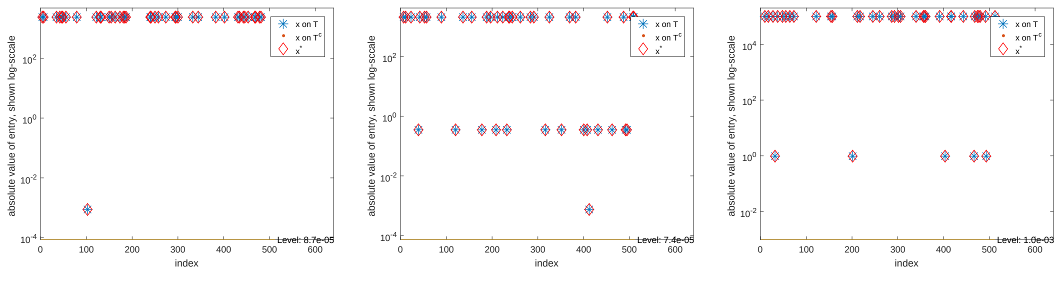


(o) P_OMP ($\rho_1 = 0.2, \rho_2 = 0.8$) (p) P_OMP ($\rho_1 = 0.5, \rho_2 = 0.5$)

(q) P_OMP ($\rho_1 = 0.8, \rho_2 = 0.2$) (r) CoSaMP

(s) OMP (t) OMMP ($M = \sqrt{s}$) (u) BP

Figure 3 Recovery results of P_OMP, CoSaMP, OMP, OMMP and BP methods on Problem 3



(v) $\mathbf{x}^*$ in Problem 4 (w) $\mathbf{x}^*$ in Problem 5 (x) $\mathbf{x}^*$ in Problem 6

Figure 4 Difficult problems in Example 4.3

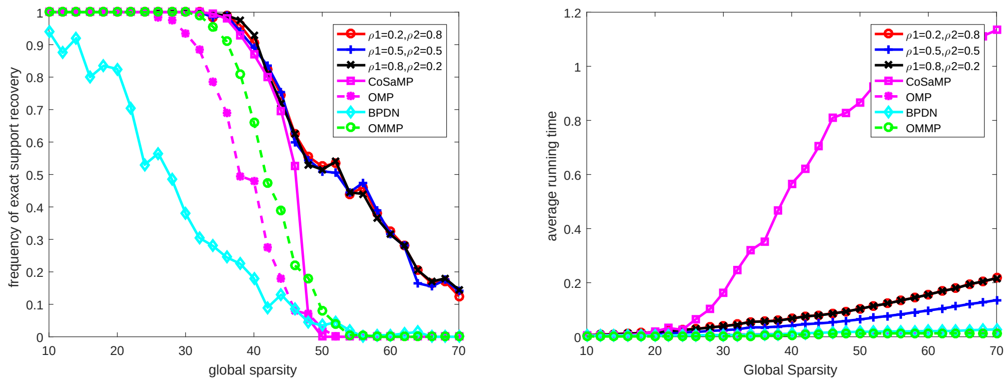


Figure 5 Comparison of success recovery rate (left plot) and average running time (right plot) for Problem 4 in Example 4.3

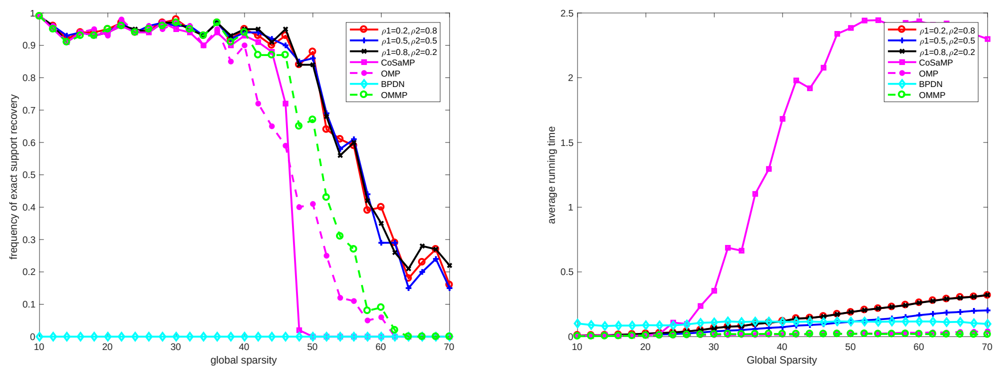


Figure 6 Comparison of success recovery rate (left plot) and average running time (right plot) for Problem 5 in Example 4.3

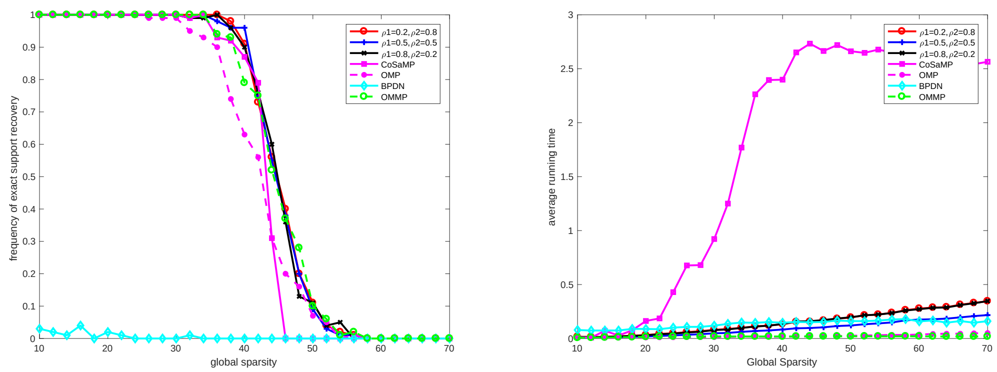


Figure 7 Comparison of success recovery rate (left plot) and average running time (right plot) for Problem 6 in Example 4.3

5. Conclusion and discussion

In this paper, we study the recovery of piecewise sparse signal by Piecewise OMP. The P_OMP algorithm recovers a $(|S_1|, \ldots, |S_N|)$ sparse signal by parallel selecting s_i largest entries of each piece and obtaining a least-square solution of $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$. Equipped with piecewise pruning $\mathbf{x}_i$ to its largest s_i terms, P_OMP which enjoys the merits of less running time than CoSaMP and exact support recovery better than CoSaMP, OMP, OMMP and BP methods, is more suitable for piecewise sparse recovery. Moreover, the P_OMP has comparable approximation error decay rate to CoSaMP with more relaxed sufficient condition $\delta_{3s} \leq 0.1$. Finally, extensions are to be discussed on the partition method on support set $\mathbf{S}$, i.e., what is the best choice of $(s_1, \ldots, s_N)$ (or piecewise structure $(\rho_1, \ldots, \rho_N)$) for P_OMP if given a piecewise sparse signal with global sparsity s and piecewise structure $(s_1, \ldots, s_N)$ is unknown. Our future study is related with this discussion.

References

[1] L. PEOTTA, P. VANDERGHEYNST. *Matching pursuit with block incoherent dictionaries.* IEEE Trans. Signal Process, 2007, **55**(9): 4549–4557.

[2] Y. C. ELDAR, M, MISHALI. *Block sparsity and sampling over a union of subspaces.* in International Conference on Digital Signal Processing. IEEE, 2009, 1–8.

[3] Y. C. ELDAR, P. KUPPINGER, H. BOLCSKEI. *Block-sparse signals: uncertainty relations and efficient recovery.* IEEE Trans. Signal Process, 2010, **58**(6): 3042–3054.

[4] E. ELHAMIFAR, R. VIDAL. *Block-sparse recovery via convex optimization.* IEEE Trans. Signal Process, 2011, **60**(8): 4094–4107.

[5] S. G. MALLAT, Zhifeng ZHANG. *Matching pursuits with time-frequency dictionaries*, IEEE Trans. Signal Process, 1993, **12**: 3397–3415.

[6] Y. C. PATI, R. REZAIIFAR, P. S. KRISHNAPRASAD. *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition.* Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers, 1993, 40–44.

[7] J. A. TROPP, A. C. GILBERT. *Signal recovery from random measurements via orthogonal matching pursuit.* IEEE Trans. Inform. Theory, 2007, **53**(12): 4655–4666.

[8] T. BLUMENSATH, M. E. DAVIES. *Iterative hard thresholding for compressed sensing.* Appl. Comput. Harmon. Anal., 2009, **27**(3): 265–274.

[9] D. L. DONOHO, Y. TSAIG, I. DRORI, et al. *Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit.* IEEE Trans. Inform. Theory, 2012, **58**(2): 1094–1121.

[10] D. NEEDELL, R. VERSHYNIN. *Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit.* IEEE Journal of Selected Topics in Signal Processing, 2010, **4**(2): 310–316.

[11] D. NEEDELL, R. VERSHYNIN. *Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit.* Found. Comput. Math., 2009, **9**(3): 317–334.

[12] D. NEEDELL, J. A. TROPP. *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples.* Applied & Computational Harmonic Analysis, 2008, **26**(3): 301–321.

[13] Zhiqiang XU. *The performance of orthogonal multi-matching pursuit under RIP.* J. Comp. Math., 2015, **33**: 495–516.

[14] Shisheng HUANG, Jubo ZHU. *Recovery of sparse signals using OMP and its variants: convergence analysis based on RIP.* Inverse Problems, 2011, **27**(3): 1–14.

[15] R. MALEH. *Improved RIP Analysis of Orthogonal Matching Pursuit.* Computer Science, 2011.

[16] Entao LIU, V. N. TEMLYAKOV. *The orthogonal super greedy algorithm and applications in compressed sensing.* IEEE Trans. Inform. Theory, 2012, **58**(4): 2040–2047.

[17] Jian WANG, S. KWON, B. SHIM. *Generalized orthogonal matching pursuit.* IEEE Trans. Signal Process, 2012, **60**(12): 6202–6216.

[18] Wei DAI, O. MILENKOVIC. *Subspace pursuit for compressive sensing signal reconstruction*. IEEE Trans. Inform. Theory, 2009, **55**(5): 2230–2249.

[19] A. MALEKI. *Coherence analysis of iterative thresholding algorithms*. Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing, IEEE Press, 2009, 236–243.

[20] S. FOUCART. *Hard thresholding pursuit: an algorithm for compressive sensing*. SIAM J. Numer. Anal., 2011, **49**(6): 2543–2563.

[21] E. CANDES, J. ROMBERG, T. TAO. *Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information*. IEEE Trans. Inform. Theory, 2006, **52**(2): 489–509.

[22] S. J. KIM, K. KOH, M. LUSTIG, et al. *A method for l^1-regularized least squares*. IEEE J. Select. Top. Signal Process, 2007, **1**(4): 606–617.

[23] M. A. T. FIGUEIREDO, R. D. NOWAK, S. J. WRIGHT. *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*. IEEE J. Select. Top. Signal Process: Special Issue on Convex Optimization Methods for Signal Processing, 2007, **1**(4): 586–598.

[24] I. DAUBECHIES, M. DEFRISE, C. D. MOL. *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*. Comm. Pure Appl. Math., 2004, **57**(11): 1413–1457.

[25] A. C. GILBERT, S. GUHA, P. INDYK, et al. *Near-Optimal Sparse Fourier Representations Via Sampling*. ACM, New York, 2002.

[26] A. C. GILBERT, S. MUTHUKRISHNAN, M. J. STRAUSS. *Improved time bounds for near-optimal sparse Fourier representation via sampling*. In Proceedings of SPIE Wavelets XI, San Diego, (CA), 2005.

[27] A. C. GILBERT, M. J. STRAUSS, J. TROPP, et al. *Algorithmic linear dimension reduction in the l^1 norm for sparse vectors*. In Proceedings of the 44th Deanna Needell (dneedell@stanford abs/cs/0608079(2006)).

[28] A. C. GILBERT, M. J. STRAUSS, J. TROPP, et al. *One sketch for all: Fast algorithms for compressed sensing*. In Proc. 39th ACM Symp. Theory of Computing, San Diego, June 2007.

[29] G. CORMODE, S. MUTHUKRISHNAN. *Combinatorial algorithms for compressed sensing*. International Conference on Structural Information and Communication Complexity. Springer-Verlag, 2006, 280–294.

[30] M. IWEN. *A Deterministic Sub-Linear Time Sparse Fourier Algorithm Via Non-Adaptive Compressed Sensing Methods*. ACM, New York, 2008.

[31] E. J. CANDES, T. TAO. *Near optimal signal recovery from random projections: Universal encoding strategies*. IEEE Trans. Inform. Theory, 2006, **52**(12): 5406–5425.

[32] D. NEEDELL, R. SAAB, T. WOOLF. *Weighted ℓ_1-minimization for sparse recovery under arbitrary prior information*. Inf. Inference, 2017, **6**(3): 284–309.

[33] M. ELAD. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[34] Zaiwen WEN, Wotao YIN, D. GOLDFARB, et al. *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation*. SIAM J. Sci. Comput., 2010, **32**(4): 1832–1857.