



Contents lists available at ScienceDirect

## Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha



Letter to the Editor

## On the number of iterations for convergence of CoSaMP and Subspace Pursuit algorithms

Siddhartha Satpathi, Mrityunjoy Chakraborty\*

Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology,  
Kharagpur, India

## ARTICLE INFO

*Article history:*

Received 23 November 2015  
Received in revised form 10 June 2016  
Accepted 7 October 2016  
Available online xxxx  
Communicated by Ming-Jun Lai

*Keywords:*

Compressive sensing  
CoSaMP  
Subspace pursuit  
Restricted isometry property  
Support set

## ABSTRACT

In compressive sensing, one important parameter that characterizes the various greedy recovery algorithms is the iteration bound which provides the maximum number of iterations by which the algorithm is guaranteed to converge. In this letter, we present a new iteration bound for the compressive sampling matching pursuit (CoSaMP) algorithm by certain mathematical manipulations including formulation of appropriate sufficient conditions that ensure passage of a chosen support through the two selection stages of CoSaMP, “Augment” and “Update”. Subsequently, we extend the treatment to the subspace pursuit (SP) algorithm. The proposed iteration bounds for both CoSaMP and SP algorithms are seen to be improvements over their existing counterparts, revealing that both CoSaMP and SP algorithms converge in fewer iterations than suggested by results available in literature.

© 2016 Elsevier Inc. All rights reserved.

## 1. Introduction

Reconstruction of signals in compressed sensing (CS) [1] involves obtaining the sparsest solution to an underdetermined set of equations given as  $\mathbf{y} = \mathbf{A}\mathbf{x}$ , where  $\mathbf{A}$  is an  $m \times l$  ( $m \ll l$ ) complex valued, sensing matrix and  $\mathbf{y}$  is an  $m \times 1$  complex valued observation vector. It is assumed that the sparsest solution to the above system is  $K$ -sparse, i.e., not more than  $K$  (for some minimum  $K, K > 0$ ) elements of  $\mathbf{x}$  are non-zero and also that the sparsest solution is unique, which can be guaranteed if every  $2K$  columns of  $\mathbf{A}$  are linearly independent [2]. Greedy approaches like orthogonal matching pursuit (OMP) [3], compressive sampling matching pursuit (CoSaMP) [4], subspace pursuit (SP) [5], hard thresholding pursuit (HTP) [6] and others recover the  $K$ -sparse signal by iteratively constructing the support set of the sparse signal (i.e., index of non-zero elements in the sparse vector) by some greedy principles. These greedy pursuits are well known for their low complexity.

\* Corresponding author.

E-mail addresses: [sidd.piku@gmail.com](mailto:sidd.piku@gmail.com) (S. Satpathi), [mrityun@ece.iitkgp.ernet.in](mailto:mrityun@ece.iitkgp.ernet.in) (M. Chakraborty).

Convergence of these iterative procedures in a finite number of steps requires the matrix  $\mathbf{A}$  to satisfy the so-called “Restricted Isometry Property (RIP)” [7] of appropriate order as given below.

**Definition 1.** A matrix  $\mathbf{A} \in \mathbb{C}^{m \times l}$  ( $m < l$ ) is said to satisfy the RIP of order  $K$  if there exists a “Restricted Isometry Constant (RIC)”  $\delta_K \in (0, 1)$  so that

$$(1 - \delta_K) \|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_K) \|\mathbf{x}\|_2^2 \quad (1)$$

for all  $K$ -sparse  $\mathbf{x}$ . The constant  $\delta_K$  is taken as the smallest number from  $(0, 1)$  for which the RIP is satisfied.

Convergence of the CS greedy recovery algorithms is usually established by imposing certain upper bounds on the RIC  $\delta_K$  as a sufficient condition. In the case of CoSaMP, such a bound is given by  $\delta_{4K} < 0.5$  [8], which is a refined (i.e., more relaxed) version of two earlier bounds, namely,  $\delta_{4K} < 0.17157$  [4] and  $\delta_{4K} < 0.38427$  [9]. Similarly, for SP, the bound proposed originally is  $\delta_{3K} < 0.205$  [5], which was improved afterwards to  $\delta_{3K} < 0.325$  [10] and  $\delta_{3K} < 0.4859$  [8].

Apart from the convergence bound on the RIC, there is another important parameter that characterizes a greedy algorithm, namely, the iteration bound, which provides the maximum (finite) number of iterations by which the algorithm is guaranteed to converge. For CoSaMP, a signal independent iteration bound of  $6(K+1)$  was presented in [4], assuming  $\delta < 0.1$ . In this letter, we present a new iteration bound for CoSaMP which refines the above result and is given as a function of  $\delta$  over the entire range for which convergence of CoSaMP is currently guaranteed (i.e.,  $0 < \delta < 0.5$ ). For this, we first develop a sufficient condition for capturing the support of the  $p+q$  largest (in magnitude) elements of  $\mathbf{x}$  within certain number of iterations ( $0 < p+q \leq K$ ), given that the support for the  $p$  largest elements of  $\mathbf{x}$  has already been captured. The derivation takes appropriate steps so that the above sufficient condition is obtained in a form structurally similar to the one proposed earlier for the HTP algorithm [11]. This permits computation of the iteration bound via a procedure suggested in [11]. Subsequently, we extend our approach to the SP algorithm and compute the corresponding iteration bound which is seen to be tighter than existing results on this [5] for more practical ranges of the RIC and is thus an improvement, as it establishes that the SP algorithm in more practical cases converges in fewer iterations than suggested in [5].

## 2. Notations and a brief review of the CoSaMP & the SP algorithms

We denote by  $Z$  the index set  $\{1, 2, \dots, l\}$ . Then, given  $B \subseteq Z$  and  $\mathbf{z} \in \mathbb{C}^l$ , the vector  $\mathbf{z}_B \in \mathbb{C}^l$  is defined as follows:  $[\mathbf{z}_B]_i = [\mathbf{z}]_i$  for  $i \in B$  and  $[\mathbf{z}_B]_i = 0$  otherwise. Similarly, given the matrix  $\mathbf{A} \in \mathbb{C}^{m \times l}$ , the matrix  $\mathbf{A}_B \in \mathbb{C}^{m \times l}$  is defined such that for  $i \in B$ ,  $[\mathbf{A}_B]_i = [\mathbf{A}]_i$  (where  $[\cdot]_i$  denotes the  $i$ -th column of the matrix  $[\cdot]$ ) and  $[\mathbf{A}_B]_i = \mathbf{0}$  otherwise. The notation  $\text{supp}(\cdot)$  denotes the support of the vector  $(\cdot)$ , i.e.,  $\text{supp}(\mathbf{z}) = \{i \in Z \mid z_i \neq 0\}$ . By  $S$  and  $S^k$ , we denote respectively the true support set of  $\mathbf{x}$  and the estimated support set after  $k$  iterations. Elements of the vector  $|\mathbf{x}|$  sorted in descending order form the vector  $\tilde{\mathbf{x}}$  and the  $r$ -th element of  $\tilde{\mathbf{x}}$ ,  $r = 1, 2, \dots, l$  is denoted by  $\tilde{x}_r$ , i.e.,  $\tilde{x}_r = [\tilde{\mathbf{x}}]_r$ . The index of the  $i$ -th largest (in magnitude) element of  $\mathbf{x}$ ,  $i = 1, 2, \dots, |S|$  is denoted by  $\pi(i)$ , implying  $|\mathbf{x}_{\pi(i)}| = \tilde{x}_i$ . Lastly, for a matrix  $\mathbf{A}$ ,  $\mathbf{A}^h$  denotes its Hermitian transposition.

For convenience of presentation, we adopt the convention of using the notation:  $\stackrel{(\cdot)}{=}$  to indicate that the equality “=” follows from Equation  $(\cdot)$  (same for inequalities). Also, unless stated otherwise, the more generalized form of CS will be considered in this paper where  $\mathbf{x}$  is  $K$ -sparse but  $\mathbf{y}$  is contaminated with a noise vector  $\mathbf{e} \in \mathbb{C}^m$ , i.e.,  $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$ .

The CoSaMP and the SP algorithms are given in Table 1 and 2 respectively. Both algorithms iteratively estimate  $\mathbf{x}$ , with  $\mathbf{x}^n$  denoting the estimate at the  $n$ -th iteration. At the “Identification” stage, in both

**Table 1**  
Compressive Sampling Matching Pursuit Algorithm.

<b>Input:</b> measurement $\mathbf{y} \in \mathbb{C}^m$ , sensing matrix $\mathbf{A} \in \mathbb{C}^{m \times l}$ , sparsity $K$ , stopping error $\epsilon$ , initial estimate $\mathbf{x}^0$
<b>For</b> ( $n = 1 ; \ \mathbf{y} - \mathbf{A}\mathbf{x}^{n-1}\ _2 > \epsilon ; n = n + 1$ )
<b>Identification:</b> $\Gamma^n = \text{supp}(H_{2K}(\mathbf{A}^h(\mathbf{y} - \mathbf{A}\mathbf{x}^{n-1})))$
<b>Augment:</b> $U^n = S^{n-1} \cup \Gamma^n ; S^{n-1} = \text{supp}(\mathbf{x}^{n-1})$
<b>Estimate:</b> $\mathbf{u}^n = \arg \min_{\mathbf{z}: \text{supp}(\mathbf{z})=U^n} \ \mathbf{y} - \mathbf{A}\mathbf{z}\ _2$
<b>Update:</b> $\mathbf{x}^n = H_K(\mathbf{u}^n)$
<b>Output:</b> $\hat{\mathbf{x}} = \mathbf{x}^{n-1}$

**Table 2**  
Subspace Pursuit Algorithm.

<b>Input:</b> measurement $\mathbf{y} \in \mathbb{C}^m$ , sensing matrix $\mathbf{A} \in \mathbb{C}^{m \times l}$ , sparsity $K$ , stopping error $\epsilon$ , initial estimate $\mathbf{x}^0$
<b>For</b> ( $n = 1 ; \ \mathbf{y} - \mathbf{A}\mathbf{x}^{n-1}\ _2 > \epsilon ; n = n + 1$ )
<b>Identification:</b> $\Gamma^n = \text{supp}(H_K(\mathbf{A}^h(\mathbf{y} - \mathbf{A}\mathbf{x}^{n-1})))$
<b>Augment:</b> $U^n = S^{n-1} \cup \Gamma^n ; S^{n-1} = \text{supp}(\mathbf{x}^{n-1})$
<b>Estimate:</b> $\mathbf{u}^n = \arg \min_{\mathbf{z}: \text{supp}(\mathbf{z})=U^n} \ \mathbf{y} - \mathbf{A}\mathbf{z}\ _2$
<b>Update:</b> $S^n = \text{supp}(H_K(\mathbf{u}^n))$
$\mathbf{x}^n = \arg \min_{\mathbf{z}: \text{supp}(\mathbf{z})=S^n} \ \mathbf{y} - \mathbf{A}\mathbf{z}\ _2$
<b>Output:</b> $\hat{\mathbf{x}} = \mathbf{x}^{n-1}$

algorithms, the residue vector  $(\mathbf{y} - \mathbf{A}\mathbf{x}^{n-1})$  is first correlated with the columns of  $\mathbf{A}$ . The support  $\Gamma^n$  of the top  $r$  elements ( $r = 2K$  in case of CoSaMP and  $r = K$  in case of SP) in terms of magnitude of correlations is then identified, using a hard thresholding operator  $H_r(\cdot)$  that retains the top  $r$  (in magnitude) elements of the vector  $(\cdot)$  and sets other elements to zero. The vector  $\mathbf{y}$  is then projected orthogonally on the column space of  $\mathbf{A}_{S^{n-1} \cup \Gamma^n}$  where  $S^{n-1} = \text{supp}(\mathbf{x}^{n-1})$ , generating the projection coefficient vector  $\mathbf{u}^n$ . In CoSaMP, the new estimate  $\mathbf{x}^n$  is taken as  $H_K(\mathbf{u}^n)$ , whereas in SP,  $\mathbf{y}$  is further projected on the column space of  $\mathbf{A}_{S^n}$ , where  $S^n = \text{supp}(H_K(\mathbf{u}^n))$ , and  $\mathbf{x}^n$  is taken as the corresponding projection coefficient vector.

**3. Proposed iteration bound analysis for CoSaMP and SP algorithms**

*3.1. Iteration bound for CoSaMP*

The proposed iteration bound computation for CoSaMP depends on the dynamics of decay of  $\|\mathbf{x}_{\overline{U}^n}\|_2$  over  $n$  (under appropriate conditions on the RIC), which is presented in Lemma 1 below and is obtained by introducing suitable modifications in the corresponding analysis in [8], which considers decay of  $\|\mathbf{x} - \mathbf{x}^n\|_2$  (rather than  $\|\mathbf{x}_{\overline{U}^n}\|_2$ ) over  $n$ .

**Lemma 1.** *In CoSaMP algorithm, the metric  $\|\mathbf{x}_{\overline{U}^n}\|_2$  decays over  $n$  with the rate  $\rho_{4K} = \sqrt{\frac{2\delta_{4K}^2(1+2\delta_{4K}^2)}{1-\delta_{4K}^2}}$  as per the following:*

$$\|\mathbf{x}_{\overline{U}^n}\|_2 < \rho_{4K} \|\mathbf{x}_{\overline{U}^{n-1}}\|_2 + (1 - \rho_{4K})\tau \|\mathbf{e}\|_2 \tag{2}$$

where the constant  $\tau$  is defined using  $\rho_{4K}$  as,  $(1 - \rho_{4K})\tau = \frac{\delta_{4K}\sqrt{6(1+\delta_{3K})}}{1-\delta_{4K}} + \sqrt{2(1 + \delta_{4K})}$ .

**Proof.** Given in Appendix A.  $\square$

Using the above lemma, we next derive a sufficient condition for capturing the support of, say, the  $p + q$  largest elements of  $\mathbf{x}$  in  $k$  iterations ( $0 < p + q \leq K$ ), assuming that the support of the  $p$  largest elements of  $\mathbf{x}$  has already been captured (where by “largest”, we mean largest in magnitude). In particular, we strive

to obtain the above sufficient condition in a form that is structurally identical to the one developed for the HTP algorithm in Lemma 3 of [11], so that the procedure to compute the iteration bound as presented in [11] can be applied. This is, however, not easy considering that algorithmically, CoSaMP is substantially different from the HTP algorithm and is in particular characterized by certain steps like ‘‘Augment’’ (i.e., expansion of the support set to size  $3K$ , as given in Table 1) and ‘‘Update’’ (i.e., pruning the support set to size  $K$  from  $3K$ ) not present in HTP. In Theorem 1 below, we show how the above can be achieved by deploying suitable mathematical manipulations, and in particular, by formulating appropriate sufficient conditions that ensure that the support of the  $p+q$  largest elements of  $\mathbf{x}$  gets selected in both the ‘‘Augment’’ step and the subsequent ‘‘Update’’ step.

**Theorem 1.** *Assume that at the  $n$ -th iteration in the CoSaMP algorithm,  $S^n : \text{supp}(\mathbf{x}^n)$  contains the support of the  $p$  ( $p < K$ ) largest (in magnitude) entries of  $\mathbf{x}$ . Then, a sufficient condition for capturing the support of the  $p+q$  largest (in magnitude) entries of  $\mathbf{x}$  in  $k$  additional iterations for some integer  $q$ ,  $K - p \geq q \geq 1$  is given by*

$$\tilde{x}_{p+q} > \rho_{4K}^k \|\tilde{\mathbf{x}}_{\{p+1,p+2,\dots,K\}}\|_2 + \gamma \|e\|_2, \quad (\rho_{4K} < 1) \tag{3}$$

where  $\gamma$  is a function of  $\delta_{3K}$  and  $\delta_{4K}$ .

**Proof.** We need to ensure that the support of the largest (in magnitude)  $p+q$  elements of  $\mathbf{x}$ , i.e.,  $\{\pi(1), \dots, \pi(p+q)\}$  gets selected in the  $(n+k)$ -th iteration. This means,  $\pi(j)$ ,  $j \in \{1, \dots, p+q\}$  should first belong to  $U^{n+k}$ , and then it also should go through the update step in CoSaMP. Now, for  $\pi(j)$ ,  $j \in \{1, \dots, p+q\}$  to belong to  $U^{n+k}$ , it is sufficient to have

$$\tilde{x}_{p+q} > \|(\mathbf{x})_{\overline{U^{n+k}}}\|_2, \tag{4}$$

as this ensures that the top  $p+q$  elements of  $\mathbf{x}$  can not belong to  $(\mathbf{x})_{\overline{U^{n+k}}}$  and thus, their support is captured in  $U^{n+k}$ . In order that the above support passes through the update step in CoSaMP under the satisfaction of (4), it is sufficient to have,

$$\min_{j \in \{\pi(1), \dots, \pi(p+q)\}} |\mathbf{u}^{n+k}_j| > \max_{i \in U^{n+k} \setminus S} |\mathbf{u}^{n+k}_i|. \tag{5}$$

In the following, we first develop a sufficient condition (viz. (9)) which, under the satisfaction of (4), guarantees satisfaction of (5). Condition (3) is then obtained by deriving a sufficient condition for simultaneous satisfaction of (4) and (9).

Note that one can write  $|\mathbf{u}^{n+k}_j| = |\mathbf{x}_j - (\mathbf{x}_j - \mathbf{u}^{n+k}_j)| \geq |\mathbf{x}_j| - |\mathbf{u}^{n+k}_j - \mathbf{x}_j|$ . Using this and some basic properties of inequalities, the LHS of (5) can be written as,

$$\begin{aligned} & \min_{j \in \{\pi(1), \dots, \pi(p+q)\}} |\mathbf{u}^{n+k}_j| \\ & \geq \min_{j \in \{\pi(1), \dots, \pi(p+q)\}} (|\mathbf{x}_j| - |(\mathbf{u}^{n+k} - \mathbf{x})_j|) \\ & \geq \tilde{x}_{p+q} - \max_{j \in \{\pi(1), \dots, \pi(p+q)\}} |(\mathbf{u}^{n+k} - \mathbf{x})_j|, \end{aligned} \tag{6}$$

while the RHS of (5) can be written as  $\max_{i \in U^{n+k} \setminus S} |\mathbf{u}^{n+k}_i| = \max_{i \in U^{n+k} \setminus S} |(\mathbf{u}^{n+k} - \mathbf{x})_i|$ , since  $\mathbf{x}_i = 0$  for  $i \in U^{n+k} \setminus S$ . Combining, in order to have (5) satisfied, it is then sufficient to have,

$$\tilde{x}_{p+q} > \max_{j \in \{\pi(1), \dots, \pi(p+q)\}} |(\mathbf{u}^{n+k} - \mathbf{x})_j| + \max_{i \in U^{n+k} \setminus S} |(\mathbf{u}^{n+k} - \mathbf{x})_i|. \tag{7}$$

Now, under the satisfaction of (4), we have  $\pi(j) \in U^{n+k}$ ,  $j \in \{1, \dots, p+q\}$ . Again,  $\pi(j) \in S$ ,  $j \in \{1, \dots, p+q\}$ . Together, these mean that  $\pi(j) \in U^{n+k} \cap S$ ,  $j \in \{1, \dots, p+q\}$ . One can then write,

$$\max_{j \in \{\pi(1), \dots, \pi(p+q)\}} |(\mathbf{u}^{n+k} - \mathbf{x})_j| \leq \|(\mathbf{u}^{n+k} - \mathbf{x})_{U^{n+k} \cap S}\|.$$

Also,

$$\max_{i \in U^{n+k} \setminus S} |(\mathbf{u}^{n+k} - \mathbf{x})_i| \leq \|(\mathbf{u}^{n+k} - \mathbf{x})_{U^{n+k} \setminus S}\|.$$

Using these and the fact that for two real numbers  $a, b$ ,  $a + b \leq \sqrt{2} \sqrt{a^2 + b^2}$ , the RHS of (7) can be written as,

$$\begin{aligned} & \max_{j \in \{\pi(1), \dots, \pi(p+q)\}} |(\mathbf{u}^{n+k} - \mathbf{x})_j| + \max_{i \in U^{n+k} \setminus S} |(\mathbf{u}^{n+k} - \mathbf{x})_i| \\ & < \sqrt{2} \|(\mathbf{u}^{n+k} - \mathbf{x})_{U^{n+k}}\|_2 \\ & \stackrel{(A.4)}{\leq} \frac{\sqrt{2} \delta_{4K}}{\sqrt{1 - \delta_{4K}^2}} \|(\mathbf{x})_{\overline{U^{n+k}}}\|_2 + \sqrt{2} \tau_1 \|\mathbf{e}\|_2. \end{aligned} \tag{8}$$

From (7) and (8), it then follows that it is sufficient to have,

$$\tilde{x}_{p+q} > \frac{\sqrt{2} \delta_{4K}}{\sqrt{1 - \delta_{4K}^2}} \|(\mathbf{x})_{\overline{U^{n+k}}}\|_2 + \sqrt{2} \tau_1 \|\mathbf{e}\|_2, \tag{9}$$

in order to satisfy (5) under the condition that (4) holds. Now, to satisfy both (4) and (9) simultaneously, a sufficient condition will be  $\tilde{x}_{p+q} > \max(\text{RHS of (4), RHS of (9)})$ . Recalling that convergence of CoSaMP requires  $\delta_{4K} < 0.5$  [8] which implies  $\frac{\sqrt{2} \delta_{4K}}{\sqrt{1 - \delta_{4K}^2}} < 1$ , it will then be enough to have  $\tilde{x}_{p+q} > \|(\mathbf{x})_{\overline{U^{n+k}}}\|_2 + \sqrt{2} \tau_1 \|\mathbf{e}\|_2$  for simultaneous satisfaction of (4) and (9), where,

$$\begin{aligned} & \|(\mathbf{x})_{\overline{U^{n+k}}}\|_2 + \sqrt{2} \tau_1 \|\mathbf{e}\|_2 \\ & < \stackrel{(2)}{\rho_{4K}^k} \|(\mathbf{x})_{\overline{S^n}}\|_2 + (\tau + \sqrt{2} \tau_1) \|\mathbf{e}\|_2 \\ & \leq \rho_{4K}^k \|(\mathbf{x})_{\overline{S^n}}\|_2 + \gamma \|\mathbf{e}\|_2 \\ & \leq \rho_{4K}^k \|\tilde{\mathbf{x}}_{\{p+1, \dots, K\}}\|_2 + \gamma \|\mathbf{e}\|_2, \end{aligned} \tag{10}$$

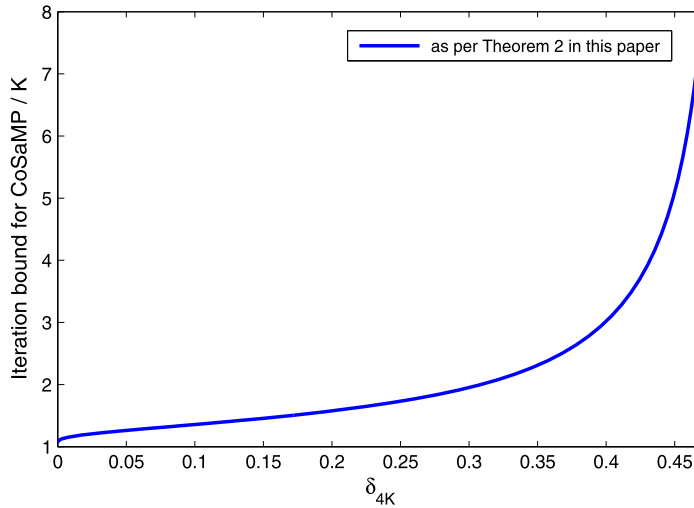
where  $\gamma = \tau + \sqrt{2} \tau_1$  and the last step follows from the assumption that  $S^n$  has captured the  $p$  largest (in magnitude) elements of  $\mathbf{x}$ .

Hence, in order that the  $p+q$  largest (in magnitude) elements of  $\mathbf{x}$  get selected in  $k$  additional iterations, it is sufficient to have,

$$\tilde{x}_{p+q} > \rho_{4K}^k \|\tilde{\mathbf{x}}_{\{p+1, \dots, K\}}\|_2 + \gamma \|\mathbf{e}\|_2. \tag{11}$$

Hence proved.  $\square$

Note that by substituting  $p = 0$  and  $q = K$  in Theorem 1 and taking  $\|\mathbf{e}\|_2$  to be zero, one can obtain the minimum number of iterations required to guarantee perfect recovery in the noiseless case (i.e., iteration bound), which is given by  $k_{min} = \lceil \frac{\log(\|\mathbf{x}\|_2 / \tilde{x}_K)}{\log(1/\rho_{4K})} \rceil$ . The above, however, provides a  $k_{min}$  that is dependent on the signal structure, i.e.,  $\|\mathbf{x}\|_2$  and  $\tilde{x}_K$ . A signal independent iteration bound can, however, be computed



**Fig. 1.** Dependence of the proposed iteration bound for CoSaMP (after normalizing by  $K$ ) on the RIC  $\delta_{4K}$ .

by noticing the similarity between (2) and the corresponding sufficient condition for the HTP algorithm derived in Lemma 3 of [11] (with the only difference being in the expressions for  $\rho_{4K}$  and  $\gamma$ ). To calculate  $k_{min}$ , one then simply has to apply the arguments used in the proof of Theorem 5 of [11] to the above context, which will require successive application of Theorem 1 on certain partitions of the index set  $S$ . The resulting iteration bound is given in Theorem 2 below.

**Theorem 2.** *With measurements  $\mathbf{y} = \mathbf{Ax}$ , the CoSaMP algorithm converges to  $\mathbf{x}$  in  $\lceil cK \rceil$  number of iterations where  $c = \ln(4/\rho_{4K}^2)/\ln(1/\rho_{4K}^2)$ .*

**Proof.** The proof follows directly by applying the arguments used in the proof of Theorem 5 of [11] to (2) and is thus omitted.  $\square$

Note that unlike [4] where an iteration bound for CoSaMP was calculated assuming  $0 < \delta_{4K} < 0.1$ , the proposed bound is defined for  $0 < \delta_{4K} < 0.5$ , i.e., over the entire range for which CoSaMP is guaranteed to converge. In order to have some quantitative idea, we plot the proposed iteration bound (after normalizing by  $K$ ) against  $\delta_{4K}$  in Fig. 1. Clearly, in comparison to [4] which obtained the iteration bound as  $6(K + 1)$  (for  $0 < \delta_{4K} < 0.1$ ), the proposed bound is about four to six times less, which is a significant improvement.

### 3.2. Extension to the subspace pursuit algorithm

Like Lemma 1 for CoSaMP above, there exists a similar decay relation for the SP algorithm as well [8]. However, due to the presence of an additional orthogonal projection step in the SP algorithm (viz., the second operation in the ‘‘Update’’ step), the decay relation is obtained here directly in terms of  $\|\mathbf{x}_{\overline{S}^n}\|_2$ . This makes it possible to formulate a sufficient condition to capture the  $p+q$  largest (in magnitude) elements of  $\mathbf{x}$  within a certain number of iterations in a much simpler way than in CoSaMP and thus the derivation becomes lot simpler.

**Theorem 3.** *With measurements  $\mathbf{y} = \mathbf{Ax}$ , the SP algorithm converges to  $\mathbf{x}$  in  $\lceil cK \rceil$  number of iterations, where  $c = \ln(4/\rho_{3K}^2)/\ln(1/\rho_{3K}^2)$ .*

**Proof.** In the SP algorithm, a decay relation analogous to Lemma 1 for CoSaMP is given by [8]

$$\|\mathbf{x}_{\overline{S}^n}\|_2 < \rho_{3K} \|\mathbf{x}_{\overline{S}^{n-1}}\|_2, \tag{12}$$

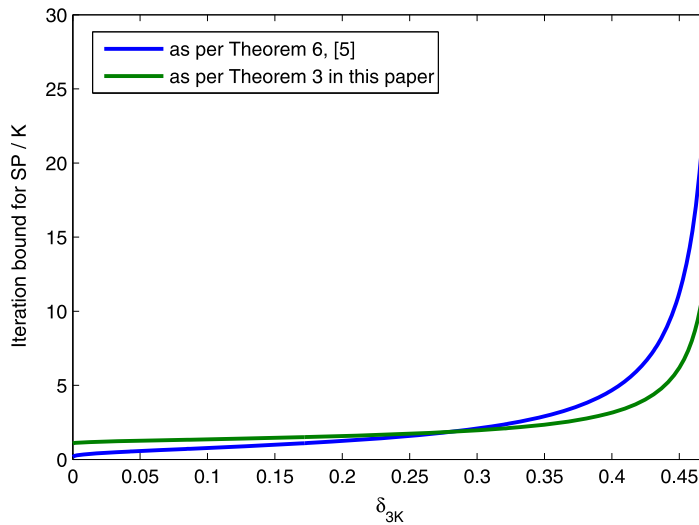


Fig. 2. Comparison of the proposed iteration bound of the SP algorithm vis-a-vis the existing iteration bound as given in [5].

where  $\rho_{3K} = \frac{\sqrt{2\delta_{3K}^2(1+\delta_{3K}^2)}}{1-\delta_{3K}^2}$ . As before, we now develop conditions to ensure that the support of the largest (in magnitude)  $p + q$  elements of  $\mathbf{x}$ , i.e.,  $\{\pi(1), \dots, \pi(p + q)\}$  get selected in  $S^{n+k}$ , assuming that the support  $\{\pi(1), \dots, \pi(p)\}$  has already been selected in  $S^n$ . A sufficient condition to ensure that the support  $\{\pi(1), \dots, \pi(p + q)\}$  is captured in  $S^{n+k}$  will be given by,

$$\tilde{x}_{p+q} > \|\mathbf{x}_{\overline{S^{n+k}}}\|_2. \tag{13}$$

From (12), proceeding recursively backwards, we can write  $\|\mathbf{x}_{\overline{S^{n+k}}}\|_2 < \rho_{3K}^k \|\mathbf{x}_{\overline{S^n}}\|_2$ . Again, from the assumption that the support  $\{\pi(1), \dots, \pi(p)\}$  has already been selected in  $S^n$ , we have,  $\|\mathbf{x}_{\overline{S^n}}\|_2 \leq \|(\tilde{\mathbf{x}})_{\{p+1, \dots, K\}}\|_2$ . From this and (13), a sufficient condition for ensuring that  $\{\pi(1), \dots, \pi(p + q)\} \subseteq S^{n+k}$  given that  $\{\pi(1), \dots, \pi(p)\} \subseteq S^n$  is given by,

$$\tilde{x}_{p+q} > \rho_{3K}^k \|(\tilde{\mathbf{x}})_{\{p+1, \dots, K\}}\|_2. \tag{14}$$

Since (14) has the same form as that of Lemma 3 of [11], one can compute the iteration bound by directly applying the procedure given in Theorem 5 of [11]. The resulting iteration bound is given by  $\lceil cK \rceil$ , where  $c = \frac{\ln(4/\rho_{3K}^2)}{\ln(1/\rho_{3K})}$ . □

For higher and thus more practical values of  $\delta_{3K}$ , the iteration bound proposed in Theorem 3 is an improvement over the existing result  $\lceil \frac{1.5K}{\ln(1/\rho_{3K})} \rceil$  as given in Theorem 6 of [5]. To show this, we plot both the iteration bounds (after normalizing by  $K$ ) against  $\delta_{3K}$  in Fig. 2 over the range  $0 < \delta_{3K} < 0.4859$ , i.e., the range for which the SP algorithm is guaranteed to converge, after adopting  $\rho_{3K}$  from [8]. It is seen from Fig. 2 that while for  $0 < \delta_{3K} < 0.28$ , the proposed iteration bound is slightly higher than that of [5], for  $\delta_{3K} > 0.28$  (which is also a more practical range for  $\delta_{3K}$  for the SP algorithm), the former is significantly lesser than the latter and the difference grows with  $\delta_{3K}$ . This shows that for more practical ranges of  $\delta_{3K}$ , the SP algorithm actually converges in significantly fewer iterations than suggested in [5].

### Appendix A. Proof of Lemma 1

As a consequence of the identification step in CoSaMP, as proved in Lemma 7, [8], we can write,

$$\|(\mathbf{x})_{\overline{U^n}}\|_2 \leq \sqrt{2}\delta_{4K} \|\mathbf{x} - \mathbf{x}^{n-1}\|_2 + \sqrt{2(1 + \delta_{4K})} \|\mathbf{e}\|_2. \tag{A.1}$$

Now, we need to upper bound  $\|\mathbf{x} - \mathbf{x}^{n-1}\|_2$  in terms of  $\|(\mathbf{x})_{\overline{U^{n-1}}}\|_2$ . For notational convenience, we obtain the upper bound of  $\|\mathbf{x} - \mathbf{x}^n\|_2$  in terms of  $\|(\mathbf{x})_{\overline{U^n}}\|_2$  and then replace  $n$  with  $n - 1$  later. For this, the estimation step of CoSaMP is analyzed next. For any  $\mathbf{z} \in \mathbb{C}^{l \times 1}$  with  $\text{supp}(\mathbf{z}) \subseteq U^n$ , the estimation step in CoSaMP ensures that  $\langle \mathbf{y} - \mathbf{A}\mathbf{u}^n, \mathbf{A}\mathbf{z} \rangle = 0$ . Substituting  $\mathbf{y}$  by  $\mathbf{A}\mathbf{x} + \mathbf{e}$ , this leads to  $\langle \mathbf{u}^n - \mathbf{x}, \mathbf{A}^h \mathbf{A}\mathbf{z} \rangle = \langle \mathbf{e}, \mathbf{A}\mathbf{z} \rangle$ . Defining  $V^n = U^n \cup S$ , this can also be written as,

$$\langle \mathbf{u}^n - \mathbf{x}, \mathbf{A}_{V^n}^h \mathbf{A}_{V^n} \mathbf{z} \rangle = \langle \mathbf{e}, \mathbf{A}\mathbf{z} \rangle. \tag{A.2}$$

Taking  $z = (\mathbf{u}^n - \mathbf{x})_{U^n}$ , one then obtains,

$$\begin{aligned} \|(\mathbf{u}^n - \mathbf{x})_{U^n}\|_2^2 &= \langle \mathbf{u}^n - \mathbf{x}, (\mathbf{u}^n - \mathbf{x})_{U^n} \rangle, \\ &\stackrel{(A.2)}{=} \langle \mathbf{u}^n - \mathbf{x}, (I - \mathbf{A}_{V^n}^h \mathbf{A}_{V^n})(\mathbf{u}^n - \mathbf{x})_{U^n} \rangle + \langle \mathbf{e}, \mathbf{A}(\mathbf{u}^n - \mathbf{x})_{U^n} \rangle, \\ &\leq \delta_{4K} \|\mathbf{u}^n - \mathbf{x}\|_2 \|(\mathbf{u}^n - \mathbf{x})_{U^n}\|_2 + \sqrt{1 + \delta_{3K}} \|\mathbf{e}\|_2 \|(\mathbf{u}^n - \mathbf{x})_{U^n}\|_2, \end{aligned} \tag{A.3}$$

where (A.3) follows from consequences of RIP as presented in Lemma 1,2 [8]. With  $A = \|(\mathbf{u}^n - \mathbf{x})_{U^n}\|_2$ ,  $B = \|(\mathbf{u}^n - \mathbf{x})_{\overline{U^n}}\|_2$  and  $\|\mathbf{u}^n - \mathbf{x}\|_2^2 = A^2 + B^2$ , (A.3) can be reduced to  $A \leq \delta_{4K} \sqrt{A^2 + B^2} + \sqrt{1 + \delta_{3K}} \|\mathbf{e}\|_2$ . After solving the above quadratic equation in  $A$  with appropriate inequalities, we get,

$$\|(\mathbf{u}^n - \mathbf{x})_{U^n}\|_2 \leq \frac{\delta_{4K}}{\sqrt{1 - \delta_{4K}^2}} \|(\mathbf{u}^n - \mathbf{x})_{\overline{U^n}}\|_2 + \tau_1 \|\mathbf{e}\|_2, \tag{A.4}$$

where  $\tau_1 = \frac{\sqrt{1 + \delta_{3K}}}{1 - \delta_{4K}}$ . Coming to the update step, as  $S^n$  is the best  $K$  term approximation to  $U^n$ , we can say

$$\|\mathbf{u}^n_{A \setminus B}\|_2 \leq \|\mathbf{u}^n_{B \setminus A}\|_2 = \|(\mathbf{u}^n - \mathbf{x})_{B \setminus A}\|_2, \tag{A.5}$$

where  $A = U^n \setminus S^n$  and  $B = U^n \setminus S$ . We also have,

$$\|\mathbf{u}^n_{A \setminus B}\|_2 = \|(\mathbf{u}^n - \mathbf{x})_{A \setminus B} + (\mathbf{x})_A\|_2 \tag{A.6}$$

$$\geq \|(\mathbf{x})_{U^n \setminus S^n}\|_2 - \|(\mathbf{u}^n - \mathbf{x})_{A \setminus B}\|_2. \tag{A.7}$$

From (A.5) and (A.7), we get,

$$\|(\mathbf{x})_{U^n \setminus S^n}\|_2 \leq \sqrt{2} \|(\mathbf{u}^n - \mathbf{x})_{A \cup B}\|_2 \leq \sqrt{2} \|(\mathbf{u}^n - \mathbf{x})_{U^n}\|_2. \tag{A.8}$$

Finally, we can upper bound  $\|\mathbf{x} - \mathbf{x}^n\|_2$  as,

$$\begin{aligned} \|\mathbf{x} - \mathbf{x}^n\|_2^2 &= \|(\mathbf{x} - \mathbf{x}^n)_{S^n}\|_2^2 + \|(\mathbf{x} - \mathbf{x}^n)_{\overline{S^n}}\|_2^2 \\ &= \|(\mathbf{x} - \mathbf{x}^n)_{S^n}\|_2^2 + \|(\mathbf{x})_{\overline{U^n}}\|_2^2 + \|(\mathbf{x})_{U^n \setminus S^n}\|_2^2 \\ &\stackrel{(A.8)}{\leq} \|(\mathbf{x} - \mathbf{x}^n)_{S^n}\|_2^2 + \|(\mathbf{x})_{\overline{U^n}}\|_2^2 + 2 \|(\mathbf{u}^n - \mathbf{x})_{U^n}\|_2^2 \\ &\leq 3 \|(\mathbf{x} - \mathbf{u}^n)_{U^n}\|_2^2 + \|(\mathbf{x})_{\overline{U^n}}\|_2^2 \\ &\stackrel{(A.4)}{\leq} \left( \frac{\sqrt{3}\delta_{4K}}{\sqrt{1 - \delta_{4K}^2}} \|(\mathbf{x})_{\overline{U^n}}\|_2 + \sqrt{3}\tau_1 \|\mathbf{e}\|_2 \right)^2 + \|(\mathbf{x})_{\overline{U^n}}\|_2^2 \\ &\leq \left( \sqrt{\frac{1 + 2\delta_{4K}^2}{1 - \delta_{4K}^2}} \|(\mathbf{x})_{\overline{U^n}}\|_2 + \sqrt{3}\tau_1 \|\mathbf{e}\|_2 \right)^2. \end{aligned} \tag{A.9}$$

Using (A.9) with (A.1), we then arrive at the result.



**Comment:** Lemma 1 and its proof as given above has some important differences with its counterpart presented in [8] (i.e., Theorem 2 of [8]). In [8], a similar decay relation was presented in terms of  $\|\mathbf{x} - \mathbf{x}^n\|_2$ . For this,  $\|(\mathbf{x} - \mathbf{x}^n)_{S^n}\|_2$  and  $\|(\mathbf{x} - \mathbf{x}^n)_{S^n^c}\|_2$  were separately upper bounded by  $\|\mathbf{x} - \mathbf{x}^{n-1}\|_2$ , and then the results were combined to obtain an upper bound of

$$\|\mathbf{x} - \mathbf{x}^n\|_2 = \sqrt{\|(\mathbf{x} - \mathbf{x}^n)_{S^n}\|_2^2 + \|(\mathbf{x} - \mathbf{x}^n)_{S^n^c}\|_2^2}$$

in terms of  $\|\mathbf{x} - \mathbf{x}^{n-1}\|_2$ . In contrast, in our treatment here, we try to upper bound  $\|(\mathbf{x})_{\overline{U}^n}\|_2$  in terms of  $\|(\mathbf{x})_{\overline{U}^{n-1}}\|_2$ , for which we first obtain an upper bound of  $\|\mathbf{x} - \mathbf{x}^n\|_2$  in terms of  $\|(\mathbf{x})_{\overline{U}^n}\|_2$  (i.e., (A.9)) and subsequently combine it with (A.1) (with  $n$  replaced by  $(n-1)$  in (A.9)). Interestingly, in doing so, we can obtain a decay relation of the metric  $\|\mathbf{x} - \mathbf{x}^n\|_2$  by combining (A.9) with (A.1), given as

$$\|\mathbf{x} - \mathbf{x}^n\|_2 \leq \rho_{4K} \|\mathbf{x} - \mathbf{x}^{n-1}\|_2 + (1 - \rho_{4K})\alpha \|\mathbf{e}\|_2, \quad (\text{A.10})$$

where  $(1 - \rho_{4K})\alpha = \frac{\sqrt{3(1+\delta_{3K})}}{1-\delta_{4K}} + \sqrt{\frac{2(1+\delta_{4K})(1+2\delta_{4K}^2)}{1-\delta_{4K}^2}}$ . The relation (A.10) is almost identical to the decay relation given in Theorem 2 of [8], with the only difference being that the coefficient  $(1 - \rho_{4K})\alpha$  in (A.10) is lesser than the corresponding coefficient in [8] as can be verified trivially. For example, with  $\rho_{4K} = 0.5$  (meaning  $\delta_{4K} \approx 0.3$ ),  $\alpha \approx 9.4$  in (A.10), whereas  $\alpha \approx 13.7$  in Theorem 2 of [8].

## References

- [1] D.L. Donoho, Compressed sensing, *IEEE Trans. Inform. Theory* 52 (4) (Apr. 2006) 1289–1306.
- [2] J.A. Tropp, S.J. Wright, Computational methods for sparse solution of linear inverse problems, *Proc. IEEE* 98 (6) (June 2010) 948–958.
- [3] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Trans. Inform. Theory* 53 (12) (Dec. 2007) 4655–4666.
- [4] D. Needell, J. Tropp, CoSaMP: iterative signal recovery from incomplete and inaccurate samples, *Appl. Comput. Harmon. Anal.* 26 (May 2009) 301–321; Also, ACM Technical Report 2008-01, California Institute of Technology, Pasadena, July 2008.
- [5] W. Dai, O. Milenkovic, Subspace pursuit for compressive sensing signal reconstruction, *IEEE Trans. Inform. Theory* 55 (5) (May 2009) 2230–2249.
- [6] S. Foucart, Hard thresholding pursuit: an algorithm for compressive sensing, *SIAM J. Numer. Anal.* 49 (6) (2011) 2543–2563.
- [7] M. Elad, *Sparse and Redundant Representations*, Springer, 2010.
- [8] C.B. Song, S.T. Xia, X.J. Liu, Improved analyses for SP and CoSaMP algorithms in terms of restricted isometry constants, arXiv:1309.6073, Sept. 2013.
- [9] Simon Foucart, Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants, in: *Approximation Theory XIII: San Antonio 2010*, in: Springer Proceedings in Mathematics, vol. 13, 2012, pp. 65–77.
- [10] Lee Kiryung, Y. Bresler, M. Junge, Oblique pursuits for compressed sensing, *IEEE Trans. Inform. Theory* 59 (9) (Sept. 2013) 6111–6141.
- [11] J.-L. Bouchot, S. Foucart, P. Hitczenko, Hard thresholding pursuit algorithms: number of iterations, preprint, <http://www.math.uga.edu/~foucart/HTPbis.pdf>.