NASA Public Access

Author manuscript

2015 IEEE Signal Process Signal Process Educ Workshop SP SPE (2015). Author manuscript; available in PMC 2017 November 20.

Published in final edited form as:

2015 IEEE Signal Process Signal Process Educ Workshop SP SPE (2015). 2015 August ; 2015: 220–225.

doi:10.1109/DSP-SPE.2015.7369556

ON THE BLOCK-SPARSITY OF MULTIPLE-MEASUREMENT VECTORS

Mohammad Shekaramiz, Todd K. Moon, and Jacob H. Gunther

Electrical and Computer Engineering Department and Information Dynamics Laboratory Utah State University

Abstract

Based on the compressive sensing (CS) theory, it is possible to recover signals, which are either compressible or sparse under some suitable basis, via a small number of non-adaptive linear measurements. In this paper, we investigate recovering of block-sparse signals via multiple measurement vectors (MMVs) in the presence of noise. In this case, we consider one of the existing algorithms which provides a satisfactory estimate in terms of minimum mean-squared error but a non-sparse solution. Here, the algorithm is first modified to result in sparse solutions. Then, further modification is performed to account for the unknown block sparsity structure in the solution, as well. The performance of the proposed algorithm is demonstrated by experimental simulations and comparisons with some other algorithms for the sparse recovery problem.

Index Terms

Multiple measurement vectors (MMVs); Block-sparsity; Support recovery

1. INTRODUCTION

The core idea behind compressive sensing (CS) is the possibility of measuring and representing a sparse or compressible signal from a set of non-adaptive linear measurements [1, 2]. In CS, the high-dimensional signal $x \in R^n$ is measured and modeled via the linear equation $y = \Phi x$, where $y \in R^m$ is the measurement vector (with $m \ll n$) and $\Phi \in R^{m \times n}$ is a wide sensing matrix. In this context, it is further assumed that x is sparse (having few non-zero elements) under some proper basis Ψ i.e., $x = \Psi x_s$, where x_s denotes a sparse vector. Combining the two above equations, we obtain $y = Ax_s$, where $A = \Phi\Psi$ [3]. Here, in order to reconstruct x , the underlying basis Ψ has to be known. The case where the sparsity basis Ψ is unknown has been called blind CS (BCS) in the literature [4], which is not our focus in this paper.

Since the sensing matrix A is wide, the model is under-determined and CS instead looks for the sparsest solution $\hat{x}_s$ such that $y = Ax_s$. Finding such sparsest possible representation is mathematically formulated as follows [5]

$$\hat{x}_s = \underset{x_s}{\operatorname{argmin}} \|x_s\|_0 \text{ subject to } y = Ax_s. \quad (1)$$

Solving the above problem is in general NP-hard. One of the most common approaches to tackle this problem is to replace $\|x_s\|_0$ in (1) with $\|x_s\|_1$. Another method is based on the prior knowledge of the sparsity level of the solution and relaxing the equality constraint in (1) as follows

$$\hat{x}_s = \underset{x_s}{\operatorname{argmin}} \quad \|y - Ax_s\|_2^2 \quad \text{subject to} \quad \|x_s\|_0 \leq k, \quad (2)$$

where k denotes the sparsity level of the solution i.e., the maximum possible non-zero elements of the solution. The uniqueness of the solution is usually guaranteed via imposing conditions on the sensing matrix A such as spark, mutual coherence, or restricted isometry property (RIP).

The set of linear measurements of a sparse signal in the presence of noise is modeled as $y = Ax_s + e$, where e denotes the noise. This model is often called a sparse single-measurement vector (SMV) in the literature. Finding the best sparse solution $\hat{x}_s$ for the SMV problem is practically achieved using greedy algorithms (such as matching pursuit (MP) and orthogonal matching pursuit (OMP)) [5], or relaxed-to-be-convex methods (such as basis pursuit de-noising (BPDN) and in-crowd algorithm) [6, 7], or sparse Bayesian learning (SBL) approaches [8, 9].

OMP is one of the famous greedy and sub-optimal algorithms to solve the SMV problem. At each iteration, OMP seeks the column of the sensing matrix A which has the maximum usefulness among all the others. The usefulness u_j is related to the normalized version of the correlation between the obtained residual at the last iteration and the j -th column of the matrix A . The algorithm usually stops when either the residual energy reaches a predefined threshold [5] or reaches the sparsity level in a case where it is known in advance [10].

Another practical algorithm for finding the sparse solution of the SMV is called basis-pursuit de-noising (BPDN) and is described as follows [6]

$$\min_x \quad \frac{1}{2} \|y - Ax\|_2^2 + \lambda \|x\|_1. \quad (3)$$

Here, the regularizer weight λ provides the emphasis on the sparsity of the solution in the l_1 sense compared to the energy of the error (residual).

Based on (3), Gill et al. proposed the in-crowd algorithm which is a fast version of BPDN, especially when dealing with high-dimensional data. Unlike most greedy algorithms that choose one non-zero location for the solution vector $\hat{x}_s$ at each iteration, the in-crowd algorithm seeks for L components and picks them under some criteria [7].

By randomizing the OMP algorithm, Elad and Yavneh demonstrated that collecting a group of sparse representations for the SMV problem results in less mean-squared error (MSE) than the sparsest solution alone [11]. As was mentioned before, the regular OMP decides on

the supports based on the maximum usefulness at each iteration. But though assigning low probability, RandOMP also gives a chance for those indices with small usefulness. By running the algorithm several times (with such a randomizing feature), one may expect to obtain different sparse representations for the same SMV problem. Then, they showed that averaging over all the solutions, though not being sparse anymore, provides less error in the MSE sense than the other greedy algorithms. Below, we present RandOMP from [11].

RandOMP Algorithm:

Main initialization:

$\mathbf{x}_{\text{sum}} = \mathbf{0}$ and Iter = 0 ($\mathbf{x}_{\text{sum}}$ denotes the sum of all solutions.)

Main iteration:

Increment Iter by one and perform the followings

Sub-initialization:

$k=0$, $\mathbf{x}^{(0)} = \mathbf{0}$, $\mathbf{r}^{(0)} = \mathbf{y}$, and $\mathcal{S}^{(0)} = \{\emptyset\}$ ($\mathbf{r}^{(k)}$ and $\mathcal{S}^{(k)}$ denote residual and support set of solution at k -th iteration, respectively.)

Sub-iteration:

Increment k by one and perform the followings:

• Compute usefulness:

$$u_j = (\mathbf{a}_j^T \mathbf{r}^{(k-1)} / \|\mathbf{a}_j\|)^2, \forall j \in \mathcal{S}^{(k-1),c}$$

• Update support set:

Draw j_0 at random with the probability proportional to $\exp \left\{ \frac{c^2}{2\sigma^2} u_j \right\}$ and update support by $\mathcal{S}^{(k)} = \mathcal{S}^{(k-1)} \cup \{j_0\}$

• **Update solution:** Compute $\mathbf{x}^{(k)}$ which is the minimizer of $\|A\mathbf{x}^{(k)} - \mathbf{y}\|_2^2$ subject to support $\{\mathbf{x}^{(k)}\} = \mathcal{S}^{(k)}$

• **Update residual:** Compute $\mathbf{r}^{(k)} = \mathbf{y} - A\mathbf{x}^{(k)}$

• **Inner loop breaking rule:** If $\|\mathbf{r}^{(k)}\|_2 < T$, break the inner loop, set $\mathbf{x}_{\text{sum}} = \mathbf{x}_{\text{sum}} + \mathbf{x}^{(k)}$, and go to the main-iteration step. T is a pre-specified threshold. Otherwise, go to the sub-iteration step for another iteration.

Stopping rule:

If Iter reaches pre-specified value, compute $\mathbf{x}_{\text{RandOMP}} = \mathbf{x}_{\text{sum}} / \text{Iter}$.

In the above algorithm, $c^2 = \frac{\sigma_x^2}{\sigma_x^2 + \sigma^2}$, and σ_x^2 and σ^2 are the noise variance of the non-zero entries of the representation of the original signal and the measurement noise variance, respectively. The term $\mathcal{S}^{(k-1),c}$ denotes the complement of the accepted support set at the $k-1$ th iteration.

Finding the sparsest solution $\mathbf{X}$ for the case where $\mathbf{Y}$ and $\mathbf{X}$ in $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E}$ are matrices is called a multiple measurement vector (MMV) problem. The MMV problem occurs in many applications such as neuromagnetic imaging [12], the reconstruction stage of Xampling (Compressed-Sensing of Analog Signals) [10, 13], and direction of arrival (DOA) estimation problem [14]. In this model it is usually assumed that all the columns of the solution matrix

X have the same unknown non-zero locations. In other words, the columns of X share joint sparsity.

In the case of MMV, there are two main approaches to find the sparsest representation. The first approach is modified versions of SMV greedy based algorithms such as MMV basic matching pursuit (M-BMP), MMV orthogonal matching pursuit (M-OMP), and MMV order recursive matching pursuit (M-ORMP) [12]. The second approach is based on hierarchical Bayesian modeling which are more flexible for incorporating the prior knowledge about the structure of the solution than the greedy algorithms [15–17].

In some practical applications, the non-zero entries of the sparse signals appear in clusters over each column of the matrix X in $Y = AX$ [10,13,15,18]. This feature has been referred to as block-sparsity in the literature. One example is magnetoencephalography (MEG), which investigates the locations where most brain activities are produced. The brain activities exhibit contiguity, meaning that they occur in localized regions [18]. Therefore, the measured signal at each snapshot can be modeled as a block-sparse SMV problem. When taking successive and almost simultaneous snapshots from the phenomena, one expects the block-sparsity structure to be preserved. Hence it is possible to model the phenomena with the block-sparse MMV problem where the block partitions are unknown a priori.

In [15], a new hierarchical Bayesian approach was proposed to deal with the block-sparse MMV problem. In this case, they incorporated a prior which encourages both contiguity and sparsity in the solution. This prior is based on the parameter referred to as Σ , which is a measure of contiguity [19]. Though this hierarchical model works well, it suffers from not being a fast algorithm.

In this paper, we modify the RandOMP algorithm to obtain a sparse solution for the MMV problem. In order to obtain better performance for the block-sparse MMVs, we then incorporate Σ to the algorithm. Since the proposed algorithm combines the ideas behind the greedy OMP algorithm with our recent work in SBL, it is faster than what was proposed in [15].

This paper is organized as follows. In section 2, the new proposed algorithm for the block-sparse MMV is described. The simulation results and comparisons will be illustrated in section 3. Finally, section 4 presents conclusions.

2. NEW ALGORITHM FOR BLOCK-SPARSE MMVS

As was discussed in section 2, the randomized OMP (RandOMP) algorithm assigns probabilities to all the indices of x to have a chance of being selected as the support of the solution in the SMV problem [11]. These probabilities are assigned based on the measure of usefulness of the corresponding column of A in reducing the residual $\|Ax - y\|_2$. In this case, at each iteration RandOMP finds a different k sparse solution and finally it fuses all of the obtained sparse solutions to represent the original signal. Here, we perform modifications to this algorithm in order to solve X in $Y = AX + E$ for the class of block-sparse MMVs. In the regular MMV, the solution matrix X has the same sparsity profile over its columns. For this

purpose, we define the support learning vector s to be a binary vector which operates on all the columns of the solution matrix X . The reason for defining the support by the vector s rather than a matrix form is due to the fact that all the columns of the solution matrix X share joint-sparsity. In other words the sparse solution is defined as $X_s = s \circ X$, where “ $\circ$ ” denotes Hadamard product. Having an entry of s equal to 1 shows that the corresponding row of the solution matrix is non-zero. Once the support vector s is found, the sparse solution matrix X can be easily obtained. We refer to our modified version of RandOMP for solving regular MMVs as sRandOMP, where the letter ‘s’ stands for sparse.

In order to improve the performance, we further modify the sRandOMP algorithm to account for block-sparsity MMVs, which means that each column of X has unknown groups of adjacent non-zeros. For this purpose, we incorporate an additional term, Σ , to the acceptance probability of each entry of s . Drawing from [15, 19], the contiguity measure of the support vector s is defined as follows. At each iteration, we first compute the absolute sum of the differences in s (the “sigma-delta”) via

$$\sum \Delta(s) = \sum_{i=2}^P |s_i - s_{i-1}|. \quad (4)$$

Based on (4), there exist few transitions for the case where the supports of the solution have block-sparsity structure compared to the non-block-wise case. For example, a constant vector (all ones or all zeros) has a Σ of 0. The prior for the support learning vector s was then made to depend on the term $\exp \{-\alpha(\Sigma(s))\}$ for some $\alpha > 0$. The parameter α specifies the significance of the Σ . Large values of α causes the supports of s to be more contiguous [15]. Here, we model the behavior of the function $\exp \{-\alpha(\Sigma(s))\}$ via the below Gamma distribution

$$(\sum \Delta) | a, \alpha \sim \Gamma(a, \alpha), \quad (5)$$

where a and α are the shape and rate parameters of Gamma distribution, respectively. We also assume that $a = 1$ and a result

$$(\sum \Delta) | a, \alpha \propto \exp \{-\alpha \sum \Delta\}. \quad (6)$$

The conditional joint probability density function on Y and (Σ) can be written as

$$P(Y, \sum \Delta) | \alpha, D, X, s \sim P(Y | D, X, s, \alpha, \sum \Delta) P(\sum \Delta | a, \alpha). \quad (7)$$

The prior on the hyper-parameter α in (5) is assumed to have Gamma distribution, $\alpha/k, \theta \sim \Gamma(k, \theta)$, and we experimentally set $k = 0.3n$ and $\theta = 1$, where n denotes the length of the

support learning vector $\mathbf{s}$. Notice that with this setting as a prior, we assume very large value for α before having any observation of data. In other words, we assume that the support vector $\mathbf{s}$ is very contiguous and has very few transitions. Once we incorporate the measurements, the posterior density on α will be updated. This posterior distribution can be described as

$$P(\alpha | -) \sim \text{Gamma}(k, \theta + \sum \Delta), \quad (8)$$

where $P(\alpha | -)$ denotes the conditional posterior density on α given the related parameters. It can be seen from (8) that when there is no transition in $\mathbf{s}$, $(\sum \Delta) = 0$ and therefore the posterior density on α is equivalent to its prior. In contrast, as $\mathbf{s}$ becomes less contiguous,

$(\sum \Delta)$ increases and the average estimation on α decreases via $E[\alpha | -] = \frac{k}{\theta + (\sum \Delta)}$. We refer to our algorithm as SDsRandOMP, where “SD” denotes sigma delta and accounts for the block-sparsity of the solution. Table (1) shows the notation used in our algorithm.

Below, we describe the steps of the SDsRandOMP algorithm for solving the MMV problem.

Outer Loop of SDsRandOMP Algorithm:

Define: $T_e = C \sigma_{n_s}^2$, $c_2 = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_n^2}$ and set $\mathbf{s}_{\text{tot}} = \mathbf{0}$
Draw a sample for α from $\Gamma(0.3n, 1)$
For $c=1:C$
 $[\mathbf{s}_c, \alpha] = \text{SDsRandOMP}(\mathbf{y}_c, A, \text{Iter}, T_e(\text{or } k_{sp}), c_2, \alpha)$
 $\mathbf{s}_{\text{tot}} = \mathbf{s}_{\text{tot}} + \mathbf{s}_c$
End For
 $\hat{\mathbf{s}} = \lambda \frac{\mathbf{s}_{\text{tot}}}{C * \text{Iter}}$

Here, λ is a tuning parameter that is set based on the noise variance. The obtained experimental curve for this relationship will be demonstrated in the next section. Based on the outer loop, we essentially treat the MMV problem as a set of SMVs having the same sensing matrix A , and let the tuning parameter λ decide on the final estimation of the support learning vector $\mathbf{s}$.

The main algorithm steps are represented below.

$[\mathbf{s}_c, \alpha] = \text{SDsRandOMP}(\mathbf{y}_c, A, \text{Iter}, T_e(\text{or } k_{sp}), c_2, \alpha)$

For Counter=1, 2, ..., Iter
 $\mathbf{I} = \mathbf{0}$, $\mathbf{x}^{(0)} = \mathbf{0}$, $\mathbf{r}^{(0)} = \mathbf{y}$, and $S^{(0)} = \{\emptyset\}$

While $\{I < K_{sp} \text{ or } \|\mathbf{r}^{(I)}\|_2 > T_e\}$

If $I < K_{sp}$ increment I by one

End If

• **Compute usefulness:**

$$u_j = (\mathbf{a}_j^T \mathbf{r}^{(k-1)} / \|\mathbf{a}_j\|)^2, \forall j \in S^{(I-1), c}$$

• **Compute Sigma-Delta:**

 Set $s_j = 1$ and compute $(\Sigma)_{j,j} \forall j \in S^{(I-1), c}$

• **Update support:**

 Draw j_0 at random with probability proportional to

$$\exp\left\{\frac{c^2}{2\sigma^2} u_j - \alpha \left(\sum \Delta\right)_{j,j}\right\},$$

 and update the support $S^{(k)} = S^{(k-1)} \cup \{j_0\}$

• **Update contiguity parameter:**

 Draw α from $\Gamma(k, \theta + (\Sigma)_{j_0})$

• **Update solution:**

 Compute $\mathbf{x}_c^{(I)}$, the minimizer of $\|\mathbf{A}\mathbf{x}_c^{(I)} - \mathbf{y}_c\|_2^2$ subject to support $\{\mathbf{x}_c^{(I)}\} = S^{(I)}$

• **Update residual:**

 Compute $\mathbf{r}^{(I)} = \mathbf{y}_c - \mathbf{A}\mathbf{x}_c^{(I)}$

End While

$s_c = s_c + s$

End For

3. SIMULATION RESULTS

In this section, we experimentally show the advantage of SDsRandOMP algorithm over sRandOMP, in-crowd algorithm, and basis pursuit de-noising (BPDN). In this case, our multiple-measurement vectors (MMVs) is a set of linear equations where the supports of the solution are binary and randomly drawn from a Bernoulli distribution in such a way to have a block-sparsity structure. The matrix $\mathbf{X}$ is drawn i.i.d. from Gaussian distribution with zero mean and variance $\sigma_x^2 = 1$. The true solution is constructed from $\mathbf{X}_s = \mathbf{s} \mathbf{O} \mathbf{X}$. The sensing matrix $\mathbf{A}$ is 50×195 and the entries of the noise vector are drawn i.i.d. from $e_{ij} \sim N(0, \sigma_n^2)$. In order to show the performance, we do simulations for different values of noise variance. Finally, the measurement matrix $\mathbf{Y}$ is 50×5 and is computed from $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E}$.

Performance of the algorithms is evaluated using the probability of correct detection of a support location (probability of detection) P_D and the probability of detecting a support location where there is none (probability of false alarm) P_{FA} , and are defined as

$$P_D = \frac{\# \text{Correct detections}}{\text{Possible Correct detections}},$$

$$P_{FA} = \frac{\# \text{False detections}}{(\# \text{Possible detections}) - (\# \text{Correct detections})}.$$

In all of our simulations for sRandOMP and SDsRandOMP, the total number of iterations is set to 5000. The first simulation is the comparison of sRandOMP with BPDN. Here, we used SPGL1 as a solver for the BPDN [20]. In Fig.1, we illustrate the experimental receiver-operating-curves (ROC) comparisons for different SNRs. The legend denotes the applied algorithm and the number in parenthesis is the noise standard deviation.

Shown in Fig.1, one can see that for larger noise variances, the sRandOMP algorithm does not do a better job compared to BPDN, and for low noise variance, the performance is approximately equal. Notice that sRandOMP does not have the learning parameter for the block sparsity structure.

In Fig.2, we demonstrate the performance of the SDsRandOMP which incorporates the Sigma-Delta parameter to account for block-sparsity and to compare it with BPDN.

From Fig.1 and Fig.2, it is clear that the SDsRandOMP which has an emphasizing parameter on block-sparsity provides better results than sRandOMP. Moreover, we observe in Fig.2 that for high SNRs our algorithm has a higher performance than BPDN. However, as the noise variance becomes greater than 0.25, BPDN overcomes SDsRandOMP.

In Fig.3 we illustrate the performance of our algorithm compared to the in-crowd algorithm. Here, we observe that our algorithm outperforms in-crowd at high SNRs, though at higher noise variance the in-crowd has better performance.

As was described in section 2, the final estimate of the support s depends on the tuning parameter λ . In Fig.4, we illustrate the behavior of λ as a function of noise variance for different SNRs. This can be used as an experimental tuning for λ when using SDsRandOMP.

We now apply the SDsRandOMP to two specific case scenarios using SDsRandOMP with setting the noise standard deviation to $\sigma_n = 0.1$. The first example is a clumpy case where non-zero rows of the true solution (non-zeros of s) are $\{17 : 24, 171 : 180\}$ with $\{17 : 24\}$ denoting the 17th through 24th entries of s . In the second example, our multiple-measurement vectors (MMVs) is a set of linear equations that comes from the continuous-to-finite (CTF) stage of the Xampling problem [10,13]. The sensing matrix A for such a problem is in general a complex-valued matrix, but to simplify the model for the simulation purposes, we take only the real part. The true supports for the second example are less clumpy with the non-zeros occurring at $\{33, 34, 47, 48, 92, 93, 103, 104, 148, 149, 162, 163\}$. In Fig.5, we show the histogram of α as it is being updated.

Below, we also illustrate the recovered supports for our two case scenarios as the support learning vector s is being updated.

4. CONCLUSION

Multiple-measurement vectors (MMVs) with unknown block-sparsity structure in their solutions were considered. A new algorithm was proposed to deal with such categories of sparse recovery. The algorithm has a merit of learning the block-sparsity via defining Sigma-Delta as a measure of contiguity in the supports of the solution. Regarding the running time of the algorithm, our algorithm provides a trade-off between the speed of greedy-based OMP and the exactness of the SBL approach in [15]. Based on the demonstrated simulations, we showed that the proposed algorithm has a better performance in high SNRs compared to basis-pursuit de-noising, modified RandOMP, and in-crowd algorithm.

Acknowledgments

This work is supported by the Grant NASA NNX13 AD 39 A.

References

1. Candes EJ, Romberg J, Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans Info Th.* Feb.2006 52:489–509.
2. Donoho D. Compressed sensing. *IEEE Trans Info Th.* Apr.2006 52:1289–1306.
3. Candes EJ, Wakin MB. An introduction to compressive sampling. *IEEE Sig Proc Mag.* Mar.2008 25:21–30.
4. Gleichman S, Eldar YC. Blind compressive sensing. *IEEE Trans Info Theory.* Oct.2011 57:6958–6975.
5. Elad, M. *Sparse and Redundant Representation: From Theory to Applications in Signal and Image Processing.* Springer; 2010.
6. Chen S, Donoho D. Basis pursuit. 28th Asilomar Conf on Sig, Syst, and Comp. Oct.1994 :41–44.
7. Gill PR, Wang A, Molnar A. The in-crowd algorithm for fast basis pursuit denoising. *IEEE Trans Sig Proc.* Oct.2011 59:4595–4605.
8. Zhang Z, Rao B. Recovery of block sparse signals using the framework of block sparse Bayesian learning. *IEEE Int Conf on Acoust, Speech and Sig Proc (ICASSP).* 2012:3345–3348.
9. Wipf D, Rao B. Sparse Bayesian learning for basis pursuit selection. *IEEE Trans Sig proc.* Aug. 2004 52:2153–2164.
10. Mishali M, Eldar YC. Reduce and boost: Recovering arbitrary sets of jointly sparse vectors. *IEEE Trans Sig Proc.* Oct.2008 56:4692–4702.
11. Elad M, Yavneh I. A plurality of sparse representations is better than the sparsest one alone. *IEEE Trans Info Theory.* Oct.2009 55:4701–4714.
12. Cotter SF, Rao BD, Engan K, Delgado KK. Sparse solutions to linear inverse problem with multiple measurement vectors. *IEEE Trans Sig Proc.* Jul.2005 53:247–248.
13. Mishali M, Eldar YC. Xampling: Signal acquisition and processing in union of subspaces. *IEEE Trans Sig Proc.* Mar.2011 59:4719–4734.
14. Luo J, Zhang XP, Wang Z. Direction-of-arrival estimation using sparse variable projection optimization. *Proc Int Symposium on Circs and Syst (ISCAS).* May.2012 :3106–3109.
15. Shekaramiz M, Moon T, Gunther J. Hierarchical Bayesian approach for jointly-sparse solution of multiple-measurement vectors. 48th Asilomar Conf on Sig, Syst and Compt. Nov.2014 :1962–1966.
16. Baron D, Sarvotham S, Baraniuk RG. Bayesian compressive sensing via belief propagation. *IEEE Trans Sig Proc.* Jan.2010 58:269–280.
17. Wipf DP, Rao BD. An empirical Bayesian strategy for solving the simultaneous sparse approximation problem. *IEEE Trans Sig Proc.* Jul.2007 55:3704–3716.

18. Kwon HE, Rao B. On the benefits of the block-sparsity structure in sparse signal recovery. IEEE Int Conf on Acoust, Speech and Sig Proc (ICASSP). 2012:3685–3688.
19. Grant C, Moon T, Gunther J, Stites M, Williams GP. Detection of amorphously shaped objects using spatial information detection enhancement (side). IEEE J Sel Topics Appl Earth Observ Remote Sens. 2012; 5(2):478–487.
20. Berg, E., Friedlander, M. SPGL1. <https://www.math.ucdavis.edu/~mpf/spgl1/>

Figure 1.
Experimental ROC comparison between BPDN and sRandOMP.

Figure 2.
Experimental ROC comparison with BPDN.

Figure 3.
Experimental ROC comparison with in-crowd.

Figure 4.
Behavior of λ as SNR varies.

Figure 5.
Histogram of α for more- and less-clumpy cases.

Figure 6.
More clumpy case.

Figure 7.
Less clumpy case.

Table 1

Notation for SDsRandOMP Algorithm

T_e	Threshold error for stopping condition
C	Number of columns of X and Y
σ_x^2	Variance of each entry of X
σ_n^2	Noise variance
K_{sp}	Sparsity level
Iter	Number of iterations
y_c	c th column of matrix Y
λ	Tuning parameter, $\lambda \in (0, 1]$
$r^{(l)}$	Residual at the l th iteration
$s^{(l)}$	Support of the solution at the l th iteration
s^c	Complement of the support set for s