

Stochastic CoSaMP: Randomizing Greedy Pursuit for Sparse Signal Recovery

Dipan K. Pal (✉) and Ole J. Mengshoel

{dipanp,ole.mengshoel}@cmu.edu
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15232

Abstract. In this paper, we formulate the K -sparse compressed signal recovery problem with the L_0 norm within a Stochastic Local Search (SLS) framework. Using this randomized framework, we generalize the popular sparse recovery algorithm CoSaMP, creating Stochastic CoSaMP (StoCoSaMP). Interestingly, our deterministic worst case analysis shows that under the Restricted Isometric Property (RIP), even a purely random version of StoCoSaMP is guaranteed to recover a notion of strong components of a sparse signal, thereby leading to support convergence. Empirically, we find that StoCoSaMP outperforms CoSaMP, both in terms of signal recoverability and computational cost, on different problems with up to 1 million dimensions. Further, StoCoSaMP outperforms several other popular recovery algorithms, including StoGradMP and StoIHT, on large real-world gene-expression datasets.

1 Introduction

Sparse Signal Recovery. The fundamental problem of K -sparse signal recovery from compressed samples is to identify the correct support over the measurement matrix atoms or columns. Given an $M \times N$ measurement matrix Φ and a set of M measurements in the form of the measurement vector \mathbf{y} , we want to determine an N -dimensional vector \mathbf{x} , a K -sparse signal. Moreover, we want to identify which K atoms of Φ (*i.e.*, the support) were used to generate the signal. Once the support is known, it is trivial to recover the signal using least squares estimation. Looking at this problem naively, one sees that the problem is to pick the right support among $\binom{N}{K}$ different ones, which is known to be NP-hard [13].

Early sparse signal recovery algorithms for Compressed Sensing were greedy pursuit algorithms, *e.g.*, OMP [22], ROMP [14], and CoSaMP [15] (among others such as GraDes [7], IHT [2], and AMP [6]). A key assumption that many greedy pursuit recovery algorithms, including OMP and CoSaMP, make is that Φ satisfies the Restricted Isometry Property (RIP) [4]. A Φ matrix that satisfies RIP preserves the norm of K -sparse signals under its transformation, *i.e.*, $\Phi\mathbf{x}$.

Definition 1 (RIP). A real-valued matrix Φ satisfies the Restricted Isometry Property (RIP) with constant δ_K if for all K -sparse vectors \mathbf{x} we have

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

OMP, ROMP, and CoSaMP all have a well-defined phase transition boundary beyond which they begin to fail due to RIP breaking down. This is also because of the inherent non-convexity of the search space as the recovery problems become more difficult (higher sparsity K and more columns than rows for Φ). The phase transition diagram characterizes the degree of difficulty of the recovery problem and is parameterized by M , N , and K .

Stochastic Local Search (SLS). Stochastic Local Search (SLS) algorithms seek to obtain approximate solutions for non-convex problems through randomization. SLS has long played a key role in state-of-the-art algorithms for tackling NP-hard and other difficult computational problems [8]. Example problems for which SLS algorithms are competitive include satisfiability of propositional logic formulas [8]; most probable explanation in Bayesian networks [11]; and maximum a posteriori hypothesis in Bayesian networks [18]. SLS algorithms show great variety [8], however they all use pseudo-randomness (often called “noise” in the SLS literature) during initialization, local search, or both. Carefully balancing randomness and greediness typically has a dramatic and positive impact on the run-time of SLS algorithms [20, 10], leading us to strive for similar positive results for sparse signal recovery in this paper.

Related Work. A stochastic approach based on Threshold Accepting (a deterministic form of Simulated Annealing) has been developed [1]. This approach uses an objective function as the product between the L_1 norm and the spectral entropy. A more direct method of randomizing atom selection has also emerged [19], based on Matching Pursuit and OMP. Limitations of this work are experimental validation using low-dimensional problems (up to 128 dimensions) and modest theoretical insights. Another randomized approach to Matching Pursuit uses a non-adaptive random sequence of sub-dictionaries in the decomposition process [12]. More recently, StoGradMP [17] has focused on an approach similar to ours, namely randomizing GradMP [16]. GradMP is a generalized version of CoSaMP. However, StoGradMP is based on stochastic gradient descent, randomly picking one support component at every iteration before projecting the gradient onto a $2K$ dimensional subspace before merging. Under RIP, the algorithm is shown to have exponential convergence in error on average.

Contribution. Our StoCoSaMP method is inspired by SLS and generalizes CoSaMP. Thus, we randomly execute a greedy or a stochastic step at every iteration. During a stochastic step, we randomly choose $2K$ atoms to merge into the support (see Section 2.1). In contrast to StoGradMP [17], which has only been shown to handle problems with up to 1000 dimensions so far, we show StoCoSaMP to be effective even in problems with up to 1 million dimensions. This renders StoCoSaMP immediately available to real-world applications. Further, StoGradMP requires a careful choice of the block size parameter which has a significant impact on performance. StoCoSaMP also has a parameter, namely the probability P_R of a random step. However, as we find in our experiments, StoCoSaMP’s performance is robust for a large range of values for P_R .

In this paper, we make the following contributions:

- We formulate the sparse recovery problem within an SLS framework and propose a novel randomized version of CoSaMP: Stochastic CoSaMP (StoCoSaMP). Key in StoCoSaMP is to randomize support selection: we randomly choose $2K$ elements of the support, augmenting the greedy selection through a signal proxy as in CoSaMP [15].
- In a worst case deterministic analysis of StoCoSaMP, we find that under RIP, random support selection is sufficient for StoCoSaMP to recover β -strong components of the true sparse signal. We also show that even a purely random version of StoCoSaMP converges to the true support.
- In experiments with random Gaussian and Hadamard measurement matrices, we demonstrate that StoCoSaMP is more efficient than and outperforms CoSaMP for problems with up to 1 million dimensions.
- In experiments focused on the problem of classification of large-scale real-world gene expression cancer datasets, we compare StoCoSaMP with StoGradMP, StoIHT, CoSaMP, OMP, ROMP, IHT, and AMP. We find that StoCoSaMP outperforms all of these algorithms in terms of test error.

2 Stochastic Local Search for Sparse Signal Recovery

2.1 Stochastic CoSaMP (StoCoSaMP)

Signal proxy and CoSaMP. A popular greedy recovery algorithm is CoSaMP [15]. We generalize CoSaMP to Stochastic CoSaMP (StoCoSaMP) as presented in Fig. 1. StoCoSaMP takes as input a measurement matrix Φ , a measurement vector \mathbf{y} , a sparsity level K , and the probability of a random step P_R . It outputs a sparse coefficient vector \mathbf{a} .

StoCoSaMP invokes, with probability P_R (line 6 in Fig. 1), a random step (line 7 in Fig. 1). This random step complements the greedy optimization of the most correlated atoms through the signal proxy (lines 9–10 in Figure 1); see Definition 6. The random step randomly chooses $2K$ atoms to merge into the current support set T (line 12 in Fig. 1). Lines 12 to 16 proceed exactly like CoSaMP. CoSaMP maintains a constant sized support (size K) at every iteration (while loop in Fig. 1) and hence can be represented in an SLS framework (see Section 2.2). Theoretically, in Section 2.5, we find that a least squares approximation provides guarantees that do not require a greedy selection such as through the signal proxy. Example stopping criteria (line 3 in Fig. 1) are (i) a maximum number of iterations and (ii) a threshold on the difference between reconstruction errors in subsequent iterations.

We now introduce the framework for defining an SLS algorithm and proceed to model StoCoSaMP in that framework.

2.2 Stochastic Local Search Framework

Definition 2 (General SLS model). An SLS model is a 4-tuple (S, N_b, G, O) , where: S is the set of all states in the search space; $G : S \rightarrow \mathbb{R}$ is the objective

```

1: StoCoSaMP ( $\Phi, \mathbf{y}, K, P_R$ ):
2:  $\mathbf{a}^0 \leftarrow 0, \mathbf{v} \leftarrow \mathbf{y}, i \leftarrow 0, T \leftarrow \{\}$  (Initialization)
3: while stopping criterion not satisfied do
4:    $i \leftarrow i + 1$ 
5:    $r \leftarrow$  sample uniform  $U[0, 1]$ 
6:   if  $r < P_R$  then
7:      $\lambda \leftarrow$  Randomly choose  $2K$  atoms to merge (Novel random step)
8:   else
9:      $\mathbf{u} \leftarrow \Phi^T \mathbf{v}$  (Form signal proxy or estimate, i.e., Traditional greedy step from CoSaMP)
10:     $\lambda \leftarrow \text{supp}(\mathbf{u}_{2K})$  (Identify  $2K$  largest components)
11:   end if
12:    $T \leftarrow \lambda \cup \text{supp}(\mathbf{a}^{k-1})$  (Merge supports)
13:    $\mathbf{b}|_T \leftarrow \Phi_T^\dagger \mathbf{y}$  (Estimate signal using least squares)
14:    $\mathbf{b}|_{T^c} \leftarrow 0$  (Set the complement support to 0)
15:    $\mathbf{a}^i \leftarrow \mathbf{b}_K$  (Next approximation: keep largest  $K$  elements)
16:    $\mathbf{v} \leftarrow \mathbf{y} - \Phi \mathbf{a}^i$ 
17: end while
18: return  $\mathbf{a}^i$ 

```

Fig. 1. StoCoSaMP algorithm. $\text{supp}(\cdot)$ returns the indices of the non-zero atoms. Lines 6 and 7 contain the key randomization step distinguishing StoCoSaMP from CoSaMP.

or evaluation function; O is the set of optimal states, defined as $O = \{s^* | s^* = \arg \max_s G(s)\}$; and N_b denotes the neighborhood relation, *i.e.*, $N_b \subseteq S \times S$.

Sparse signal recovery can be framed as an SLS problem. We consider binary vectors $\mathbf{s} \in \mathbb{B}^N$. If $s_i = 1$, then the i -th atom or column is included in the support estimate while if $s_i = 0$, then it is not. Thus, the cardinality of the search space is $|S| = |\mathbb{B}^N| = 2^N$. Typically, SLS techniques randomly alternate between a greedy step and a random step. The greedy step usually *enumerates* the entire neighborhood search space N_b and chooses the state which produces the lowest error with respect to some objective function. Sometimes, the random step is only invoked when the previous greedy step produced no improvement.

2.3 SLS for Sparse Signal Recovery

We consider a relaxed neighborhood relation definition as in Definition 3 in order to utilize an efficient search method prevalent in sparse signal recovery: the signal proxy [15]. The signal proxy results in a closed form search step of polynomial complexity. CoSaMP and StoCoSaMP on the whole, at every iteration, search for the next best K atoms for approximation. However, the signal proxy step searches for the top $2K$ atoms. Thus, a relaxation in the neighborhood size is required for modelling it in the SLS framework. This also results in StoCoSaMP being modelled as *two* interconnected sub-models in the SLS framework (see Definition 5 and Definition 6).

Definition 3 (Relaxed Neighborhood). For some $\eta, K \in \mathbb{N}$, we define the neighborhood relation $N_b^{\eta K}(\mathbf{s}) = \{\mathbf{s}' \in \mathbb{B}^N \mid \|\mathbf{s} - \mathbf{s}'\|_0 \leq \eta K\}$ with a neighbor threshold of ηK .

For binary vectors, the Manhattan distance between two vectors equals the L_1 distance. Also, η is chosen to model an algorithm. For CoSaMP and StoCoSaMP, $\eta = 2$. This relaxation on the neighborhood size threshold from 2 to $2K$ enables using the signal proxy as the greedy element of the search. We now model a sparse signal recovery algorithm in the SLS framework that is constrained to maintain a fixed sized support of cardinality K .

Definition 4 (SLS model for K -sparse signal recovery). An SLS model for K -sparse signal recovery is a 4-tuple $(S, N_b^{\eta K}, G, O)$, where: S is the set of all states in the search space with each binary vector state \mathbf{s} satisfying $\sum_{i=1}^N s_i = K$. Lastly, $N_b^{\eta K}$ denotes the relaxed neighborhood function, i.e., $N_b^{\eta K} \subseteq S \times S$ as defined in Definition 3.

In the case of well-conditioned signal recovery problems, $|O| = 1$, i.e., there exists a single unique solution to the problem [5], [3]. The framework remains the same even if the solution is not unique.

2.4 StoCoSaMP in the SLS Framework

StoCoSaMP uses a polynomial complexity search step called the signal proxy [22]. We utilize this technique to get around the computational bottleneck of the standard SLS greedy step.

Signal proxy (line 9 in StoCoSaMP): An efficient greedy search.

Greedy sparse signal recovery utilizes the top γK components (for some $\gamma \in \mathbb{N}$) of the signal proxy. The signal proxy is defined as $(\Phi^T \mathbf{v})_{\gamma K}$, where $(\cdot)_{\gamma K}$ chooses the top γK elements, and \mathbf{v} is the current residue. The signal proxy is an efficient way to determine the most likely active components in the residue. This provides an efficient closed form solution to evaluate the greedy step with $\mathbf{s}^* = \arg \max_{\mathbf{s}} H(\mathbf{s})$ where $H(\mathbf{s}) = \|(\Phi^T \mathbf{v}) \odot \mathbf{s}\|_1$, with \odot denoting the Hadamard product. Recall that \mathbf{s} is a binary vector with γK non-zeros. Note that H depends on the residue \mathbf{v} and thus might change with every iteration.

Due to the incorporation of the signal proxy step, a relaxed neighborhood definition was needed. This complicates the SLS model for StoCoSaMP since there are now two different search spaces. One is the overall space of the top K atoms for the current support estimate (line 15 in Fig. 1) and the other is the selection of the top $2K$ atoms through the signal proxy (line 10 in Fig. 1). StoCoSaMP therefore needs a more elaborate SLS model than given in Definition 4. We thus introduce two connected sub-models: Definition 5 models the overall K -sparse support search of StoCoSaMP whereas Definition 6 models the top $2K$ atom selection through the signal proxy.

Definition 5 (Sub-model 1: SLS model for K -sparse signal recovery with StoCoSaMP). This SLS model is a 4-tuple $(S, N_b^{\eta K}, G, O)$ and parameterizes Definition 4 with $S = \{\mathbf{s} \in \mathbb{B}^N \mid \|\mathbf{s}\|_1 = K\}$. Also, $\eta = 2$, therefore N_b^{2K} denotes the relaxed neighborhood function for S as in Definition 3. Definitions for G and O remain the same.

Definition 6 (Sub-model 2: SLS model for support selection in StoCoSaMP). The SLS model is a 4-tuple $(S', N_b^{\eta K}, G', O'_s)$ and parameterizes Definition 4 with $S' = \{\mathbf{s}' \in \mathbb{B}^N \mid \|\mathbf{s}'\|_1 = \gamma K\}$. $G' : S' \times S' \rightarrow \mathbb{R}$ where S belongs to sub-model 1. $G'_s(\mathbf{s}') = \|(\Phi^T(\mathbf{y} - \Phi(\mathbf{a}^i \odot \mathbf{s}))) \odot \mathbf{s}'\|_1$ where \mathbf{a}^i is the current coefficient estimate (line 15 in StoCoSaMP) and $G'_s(\mathbf{s}')$ is parameterized by $\mathbf{s}' \in S'$; $O'_s = \{\mathbf{s}'^* \mid \mathbf{s}'^* = \arg \max_{\mathbf{s}'} G'_s(\mathbf{s}')\}$ denoting the optimal state for $G'_s(\mathbf{s}')$, which is unique, i.e., $|O'_s| = 1$. Lastly, with $\eta = 2\gamma$, $N_b^{2\gamma K}$ denotes the relaxed neighborhood function for S' as in Definition 3.

In StoCoSaMP, the SLS technique is only explicitly used within S' in Sub-model 2 (for support selection) and not in S where the original problem lies. However, as we explain Section 2.5, SLS effects in S' allow StoCoSaMP to escape local minima in S as well. In Sub-model 1, G is not evaluated explicitly by the algorithm. G' in Sub-model 2, however, is efficient to evaluate while being parameterized by $\mathbf{s} \in S = \{\mathbf{s} \in \mathbb{B}^N \mid \|\mathbf{s}\|_1 = K\}$. Greedily optimizing G' for $\mathbf{s}'^* = \arg \max_{\mathbf{s}'} G'_s(\mathbf{s}')$, such that $\|\mathbf{s}'\|_1 = \gamma K$, offers the exponential recovery guarantees that CoSaMP enjoys. These guarantees also arise due to a least squares approximation in subsequent steps.

2.5 Analysis of StoCoSaMP

Since StoCoSaMP randomly picks a random step or a greedy step, a comprehensive analysis of the phenomenon of escaping local minima is difficult. Experimentally, we observe strong performance of StoCoSaMP as reported in Section 3. Analytically, we have some but limited results as reported below.¹

We first analyze the extreme cases $P_R = 0$ and $P_R = 1$, under RIP, before discussing the general case $0 \leq P_R \leq 1$. Note that our analysis assumes RIP only for $P_R = 0$ and $P_R = 1$. We only hypothesize a condition (when RIP breaks down) under which local minima arise in the general case.

Special Case: $P_R = 0$ (Purely Greedy Pursuit)

Lemma 1. When $P_R = 0$, StoCoSaMP is **equivalent** to CoSaMP, i.e., $\text{CoSaMP} = \text{StoCoSaMP}(\Phi, \mathbf{y}, K, 0)$.

StoCoSaMP with $P_R = 0$ enjoys the same exponential recovery guarantees as CoSaMP [15]. Unfortunately, it also inherits the propensity of CoSaMP to get trapped in local optima that may not be global.

¹ Proofs of all results not found here will be included in the full version of this paper.

Special Case: $P_R = 1$ (Purely Random Pursuit) The primary goal here is to show that even if $P_R = 1$, StoCoSaMP will retain strong components per Definition 2.5.

Lemma 2. *When $P_R = 1$, StoCoSaMP is equivalent to a random walk or a purely random pursuit.*

We have Φ as a normalized matrix with the RIP constant δ_K for K -sparse signals. Let $\mathbf{x} \in \mathbb{R}^N$ be a K -sparse vector. The following two definitions model lines 13 in Fig. 1 and the true support and the current support estimate $\text{supp}(\mathbf{b}_K)$ respectively.

Definition 7 (Least Squares Estimate). *For any randomly selected or arbitrary support set T (see lines 6 and 7 of StoCoSaMP) with $|T| = 2K$, the least squares estimate for the signal at any particular iteration \mathbf{b} is defined as $\mathbf{b}_{|T} = \Phi_{|T}^\dagger \mathbf{y}$ together with $\mathbf{b}_{|T^c} = \mathbf{0}$.*

Definition 8 (True Support and Current Support). *We define the true support to be $\lambda^* = \text{supp}(\mathbf{x})$ and the current support to be $\lambda = \text{supp}(\mathbf{b}_K)$ where \mathbf{b}_K is the best K -sparse approximation of \mathbf{b} at the current StoCoSaMP iteration.*

The following notion of strong components will prove useful in our analysis (see Theorem 1 and Theorem 2).

Definition 9 (β -Strong Component w.r.t. Ψ). *We define a true signal component \mathbf{x}_Ω ($|\Omega| = 1$) to be β -strong w.r.t. Ψ , if for a subset $\Psi \subset \lambda^*$ of the indices of the true components, with $|\Psi| \leq K - 1$ and $\Omega \notin \Psi$, we have $\frac{|\mathbf{x}_\Omega|}{\|\mathbf{x}_\Psi\|_2} \geq \beta$.*

Notation: We now define notations for the rest of Section 2.5. We denote, for one iteration of StoCoSaMP, a selected support by λ . For the analysis, we also define $\Omega \in F = \{T \cap \lambda^*\}$ with $|\Omega| = 1$; F represents the true components in the current support estimate. $Z = T \setminus \lambda^*$ represents the rest of the false components in the current support set T , which are not active in the actual signal. Lastly $\Psi = F \setminus \Omega$ for every iteration of the random step (lines 6 and 7 of StoCoSaMP).

The following lemma is useful in proving Theorem 1.

Lemma 3. *We have*

$$\|((\Phi_F^* \Phi_F)^{-1})^\Omega \mathbf{x}\|_2 \lesssim (1 \pm \delta_K^2 \eta) |\mathbf{x}_\Omega| \pm \eta \|\mathbf{x}_\Psi\|_2$$

where $\eta = \left(\frac{\delta_K}{1 - \delta_{K-1} - \delta_K^2}\right)$ and $((\Phi_F^* \Phi_F)^{-1})^\Omega$ corresponds to the Ω^{th} row of $(\Phi_F^* \Phi_F)^{-1}$. Further, we assume $\Phi_\Psi^* \Phi_\Psi$ is full rank and that Φ has normalized columns.

Lemma 3 involves two inequalities which have been combined in one statement. It is useful since it bounds the projection of \mathbf{x} onto the Ω^{th} row of $(\Phi_F^* \Phi_F)^{-1}$. We now present our main result.

Theorem 1. (*Retaining Strong True Support*). For StoCoSaMP with $P_R = 1$, if $\delta_Z \leq \delta_{Z+K} \leq 0.03$, $\delta_K \leq \delta_{Z+K}$ and $\mathbf{x}_{|\Omega}$ is β -strong w.r.t. Ψ with $\beta = 0.1$, then $\Omega \in \lambda$.

Interestingly, under RIP, Theorem 1 shows that even if an algorithm is not greedy and *always* picks a random support, β -strong true components of the signal present in the current support (w.r.t. to the other true components) are guaranteed to be retained. Our proof strategy is to find a condition such that the lower bound for $\|\mathbf{b}_{|\Omega}\|_2$ is greater than the upper bound for $\|\mathbf{b}_{|\Psi}\|_2$. This forces the true component Ω into the top K elements chosen during pruning. We find the lower and upper bounds through RIP.

When the sparse signal is exactly K -sparse, the β -strong constraint might seem restrictive, since it only allows components stronger w.r.t. the rest of the components by a factor of β to be recovered. However, in the case of general signals, where the true sparse vector contains noise, the β -strong constraint is more easily satisfied due the presence of very small noisy components. Thus, the K large components are more likely to be recovered.

Support convergence under β -strong condition: Now let \mathcal{Y}^i be the set of true components in the support estimate at the i^{th} StoCoSaMP iteration, i.e., $\mathcal{Y}^i = \lambda^i \cap \lambda^*$. Thus, at every iteration, we would like $|\mathcal{Y}|$ to increase up until the desired cardinality of the support, (i.e., K). We have the following result on support convergence.

Theorem 2. (*Support Convergence for Purely Random StoCoSaMP*) For StoCoSaMP with $P_R = 1$, i.e., a purely random pursuit, if $\{\delta_K, \delta_Z\} \leq \delta_{Z+K} \leq 0.03$ and $\exists \mathbf{x}_{|\Omega}$ in support T^i at iteration i , such that $\mathbf{x}_{|\Omega}$ is β -strong w.r.t. some Ψ with $\beta = 0.1$ and $\Omega \notin \lambda^i$, then $|\mathcal{Y}^{i+1}| \geq |\mathcal{Y}^i|$.

Theorem 2 shows that even for StoCoSaMP's purely random case ($P_R = 1$), the support estimate does not worsen as the algorithm progresses. There will be no improvement when β -strong components are not present: the support estimate does not change. These results are deterministic since they analyze the worst case and are stronger guarantees than the average case analysis typical of randomized algorithms. For $0 < P_R < 1$, the greedy step has already been shown to have exponential reduction in error [15], thus the results presented here ensure that the algorithm does not diverge while executing a random step.

Theorem 2 is a contrast to earlier results suggesting that greed is important for recoverability [21]. Although a greedy algorithm might have stronger guarantees for recoverability, random support selection allows for practical improvements (analogous to those in the SLS literature) in situations where RIP breaks down, (e.g., past the phase transition boundaries of greedy pursuits). Note that in such a case, Theorem 1 will not hold and StoCoSaMP, like CoSaMP, currently has no theoretical guarantees. The SLS properties of StoCoSaMP then assume a larger role, which is difficult to analyze theoretically.²

² Indeed, in our experiments, we find that for $P_R = 1$, when RIP breaks down, StoCoSaMP does not perform well. A few greedy steps are needed for convergence (see

General Case: $0 \leq P_R \leq 1$ (Randomized Greedy Pursuit) With $0 \leq P_R \leq 1$, under RIP, the individual theoretical guarantees of greedy and random steps still hold. However, when RIP does not hold, the random effects of the SLS model S from Definition 5 become interesting. We now define an active variable for use in the informal discussion about escaping local minima.

Definition 10 (Active Variable). Let $\mathbf{y} = \Phi \mathbf{x}$ where $\mathbf{x} \in \mathbb{R}^N$ is a K -sparse vector. Then the set $A = \{i \mid |x_i| > 0\}$ is called the active set, and the corresponding i -th element of \mathbf{x} is called an active variable.

Assume that a column or atom τ of Φ is approximately linearly dependent on some other set of columns Γ belonging to Φ , i.e., $\phi_{|\tau} \approx \sum_{i \in \Gamma} \alpha_i \phi_i$ for some α_i . Now, if the signal had each element in Γ as its active variable, but not τ , then the signal proxy $\Phi^T \mathbf{v}$ (line 9 in StoCoSaMP) forces CoSaMP (and StoCoSaMP) to pick τ . The atom τ can be said to be “stronger” than the atoms in the set Γ since τ is more likely to be picked by the signal proxy rather than the true atoms in Γ . This is because picking τ explains much more of the signal.

In this situation, when a “stronger” component τ exists w.r.t. a set Γ , the search falls into a *local minimum*. It would be hard to drop τ from the support estimate, as it approximately explains the components Γ in the signal by itself. This is where StoCoSaMP randomness could help. In randomly choosing $2K$ atoms from Φ , it is more likely than in a greedy setting that the algorithm might pick a few atoms that are active and “weak” compared to some other atom. Once CoSaMP (and StoCoSaMP) chooses a variable, it explains away that component. Thus, the random step in StoCoSaMP (for $0 < P_R \leq 1$) helps the search to avoid being trapped in a local minimum.

This effective dodging of local minima acts even when RIP might not hold. However, in the case where RIP does hold for Φ , Theorem 1 shows that greed is not necessary for recovering β -strong components of the signal w.r.t. Ψ . In many of our experiments, such as the real-world gene expression data (Section 3.5), we do not check for the RIP condition, but StoCoSaMP still works well.

3 Experimental Results

3.1 Phase Transition Diagrams

Goal. The goal of this experiment is to investigate whether StoCoSaMP can solve a broader and harder range of problems compared to CoSaMP. Specifically, we seek to reconstruct K -sparse i.i.d. Gaussian signals with no noise added.

Method and Data. The inherent dimensionality of the problem was set to $N = 200$. The measurement matrices were i.i.d. sampled from the standard

Fig. 4(a)). Nonetheless, in experiments in Section 3 we find that StoCoSaMP converges towards the true solution even in large dimensions for high $P_R = 0.9$ but not for $P_R = 1$ (see Fig. 4(a)). In most cases, we find empirically that StoCoSaMP converges on average faster than CoSaMP.

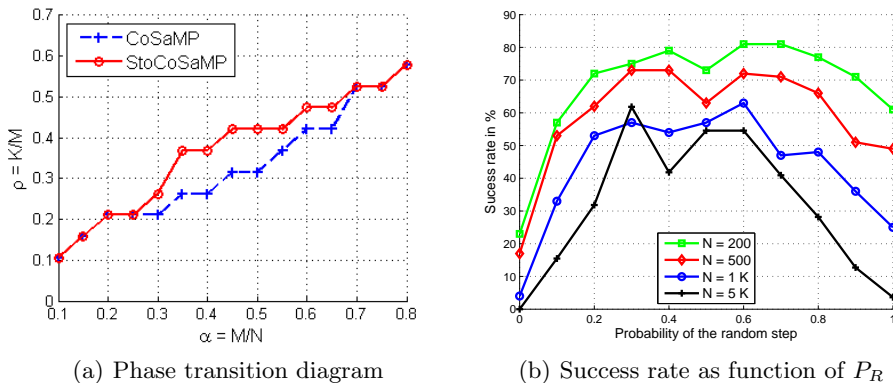


Fig. 2. (a) Phase transition boundary diagrams (showing probability of signal recovery) for CoSaMP and StoCoSaMP (with $P_R = 0.3$). (b) Effect of varying the probability of the random step P_R , along the x -axis, on StoCoSaMP's signal recoverability, along the y -axis, for Gaussian measurement matrices. CoSaMP is $P_R = 0$ in (b).

normal distribution, $\mathcal{N}(0, 1)$. For StoCoSaMP, $P_R = 0.3$.³ As is standard in the literature [9], intrinsic recovery capability of the algorithm was measured in a noiseless setting with a 90% threshold in probability of recovery.

Results. We compare the phase transition diagrams of CoSaMP and StoCoSaMP in Fig. 2(a). The x -axis is $\alpha = M/N$ and the y -axis is $\rho = K/M$, where K is the sparsity and M, N are the dimensions of Φ . If a point is below the transition boundary, problems of that setting are considered solved given a threshold for the probability of recovery (90%). The axes denote gradual change in difficulty, with the most difficult setting being the top left corner and the easiest being the bottom right corner. Fig. 2(a) shows that StoCoSaMP clearly improves on the phase transition region over CoSaMP, especially for $0.35 \leq \alpha \leq 0.5$.

3.2 Effect on Recoverability: Random Gaussian Matrices

Goal. The goal of this experiment is to compare the performance of CoSaMP and StoCoSaMP for a broad range of P_R -values.⁴ This will (i) shed light on the problem of local optima in sparse signal recovery and (ii) provide an experimental counter-point to Theorem 1.

Method and Data. We constructed 100 normalized synthetic signal recovery problems; the $M \times N$ measurement matrix was sampled i.i.d. from $\mathcal{N}(0, 1)$. Then, by varying P_R , we examine the percentage of successful recoveries for StoCoSaMP (recovered SNR > 50 dB). We investigate a challenging point on the

³ For all experiments, we set the maximum number of iterations for both CoSaMP and StoCoSaMP generously to 250.

⁴ For comparative results on real-world data please refer to Sec. 3.5

phase transition, specifically $\alpha = 0.6$ and $\rho = 0.5$ (see Fig. 2(a)). The dimensionality is varied from $N = 200$ to $N = 5000$.

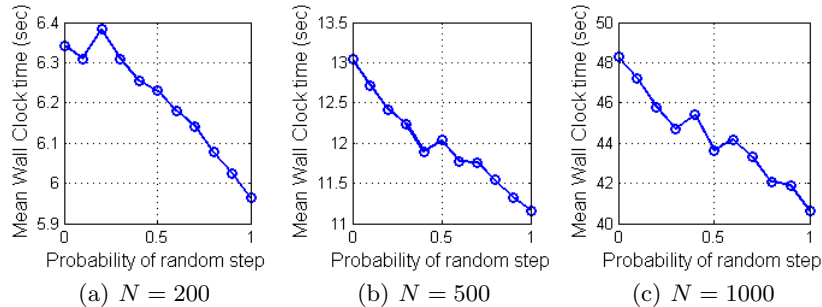


Fig. 3. Effect of varying the random step probability P_R (x -axes) on the mean wall clock run time (y -axes). CoSaMP is $P_R = 0$. The reduction in mean wall clock time is relatively small, but comes in addition to improved accuracy (see Fig. 2(b)).

Results. Fig. 2(b) illustrates the serious handicap of purely greedy pursuits in escaping local optima. Specifically, Fig. 2(b) shows that at $\alpha = 0.6$ and $\rho = 0.5$, CoSaMP ($P_R = 0$) performs poorly.⁵ For $0.1 \leq P_R \leq 0.9$, StoCoSaMP tends to succeed significantly more often. *For these lower dimensions, even a purely random walk ($P_R = 1$) performs better than CoSaMP ($P_R = 0$), owing to SLS properties and Theorem 1 (random Gaussian matrices are known to satisfy RIP).* The result, though perhaps initially surprising, is consistent with previous studies of the role of randomization in hard combinatorial problems. In problems of high difficulty, the expected time to find a global optimum is minimized when search is close to a random walk [10].

Fig. 3(a)-Fig. 3(c) show the mean wall clock run time for the convergence over these 100 problems as P_R in StoCoSaMP was varied. Since the time complexity of a single random step is lower than that of a single greedy step, the overall computational time generally decreases as we increase P_R . Hence, StoCoSaMP can not only outperforms CoSaMP in terms of recoverability, but also in terms of computation time for a significant range of P_R .

3.3 Effect on Recoverability: High-Dimensional Problems

Goal. To handle high-dimensional data such as images or spatio-temporal data, we experiment with Hadamard matrices with up to 1 million dimensions.

Method and Data. We use sets of randomly permuted rows of the Hadamard matrix as the measurement matrix, and set $\alpha = 0.1$ and $\rho = 0.05$, giving us rea-

⁵ We consider a recovery successful if the SNR of the recovered sparse signal to the ground truth is above a certain threshold (50 dB for this experiment).

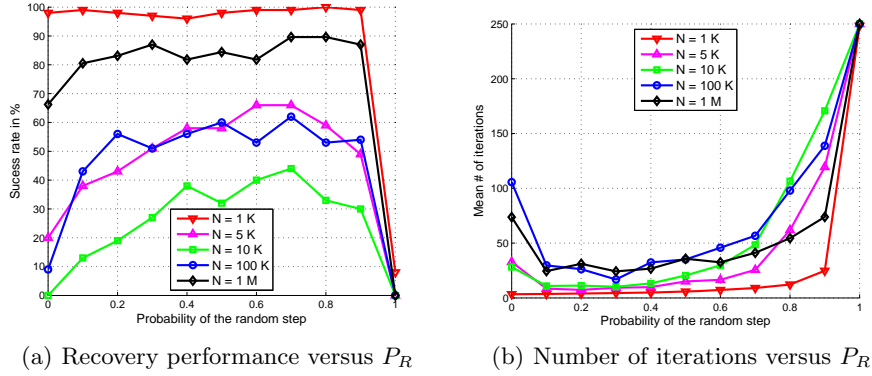


Fig. 4. Effect of varying P_R , along x -axis, on: (a) StoCoSaMP's signal recoverability for randomly permuted Hadamard measurement matrices and (b) the mean number of iterations required for convergence by StoCoSaMP. $P_R = 0$ is CoSaMP.

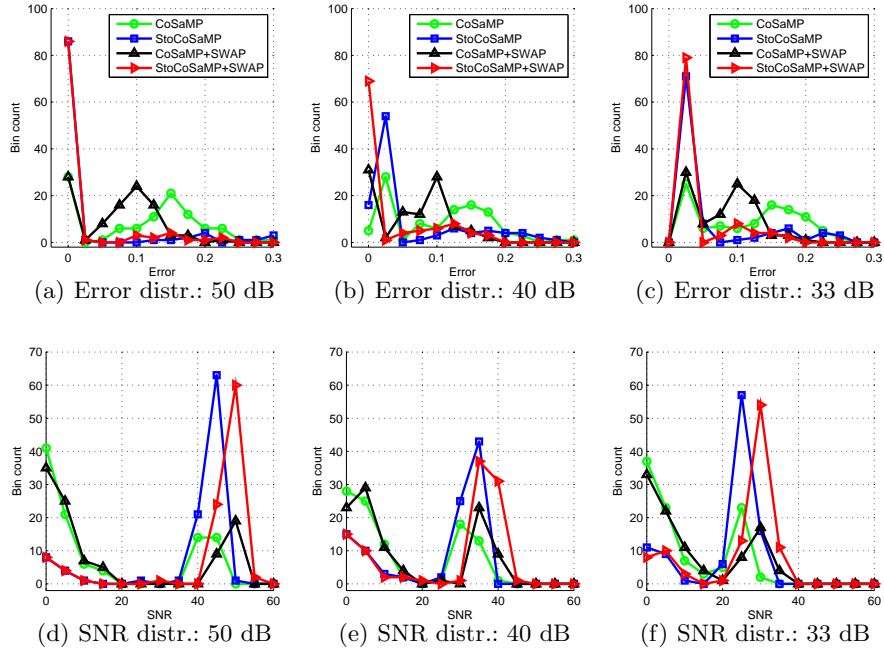


Fig. 5. Histograms of recovery errors (top row) and SNRs (bottom row) for CoSaMP (green), StoCoSaMP (blue), CoSaMP+SWAP (black), and StoCoSaMP+SWAP (red), for different levels of noise (50 dB, 40dB, or 33dB) added to the measurement vector.

Table 1. Mean (μ) and Standard deviations (σ) of recovery errors and SNRs for the comparison of CoSaMP, StoCoSaMP and SWAP initialized by each algorithm (denoted by CoSaMP+SWAP and StoCoSaMP+SWAP), under different noise added to measurement vector (SNR in dB). For a given SNR level, **bold** and *italics* signify the best and the second-best results respectively.

SNR	Algorithm	μ_{Error}	σ_{Error}	μ_{SNR}	σ_{SNR}
50	CoSaMP	0.1135	0.0803	13.934	18.393
	StoCoSaMP	<i>0.0334</i>	0.0784	<i>37.954</i>	14.595
	CoSaMP+SWAP	0.0735	<i>0.0527</i>	15.923	20.543
	StoCoSaMP+SWAP	0.0216	0.0516	42.157	<i>15.942</i>
40	CoSaMP	0.1031	0.0719	13.222	13.749
	StoCoSaMP	<i>0.0600</i>	0.0748	<i>24.157</i>	<i>14.002</i>
	CoSaMP+SWAP	0.0665	0.0478	15.134	15.634
	StoCoSaMP+SWAP	0.0390	<i>0.0508</i>	27.471	15.624
33	CoSaMP	0.1189	0.0678	9.633	<i>10.589</i>
	StoCoSaMP	<i>0.0625</i>	0.0652	<i>20.974</i>	9.718
	CoSaMP+SWAP	0.0781	<i>0.0473</i>	11.331	12.254
	StoCoSaMP+SWAP	0.0401	0.0441	24.179	10.887

sonable values for the sparsity K and the number of measurements M . We simulate 100 different problems, and define a strict SNR threshold for a successful recovery at 120 dB.

Results. Fig. 4(a) reports success rates, while Fig. 4(b) reports the number of iterations. The figures suggest that the advantages, both in terms of success rate and computation time (iterations), of StoCoSaMP over CoSaMP are not restricted to the lower-dimensional case. For *success rate* (see Fig. 4(a)), the advantage of StoCoSaMP over CoSaMP is very clear for the high-dimensional cases of $N = 100K$ (blue line) and $N = 1$ million (black line).⁶ In image processing applications, for example, $N = 1$ million is typical. As seen in Fig. 4(a), at these high dimensions, a purely random pursuit ($P_R = 1$) fails in many cases, whereas $0.1 \leq P_R \leq 0.9$ does much better.

Using $0.1 \leq P_R \leq 0.6$, StoCoSaMP also gives faster convergence than CoSaMP ($P_R = 0$) in terms of number of iterations, see Fig. 4(b). This and the results of the previous experiment in Section 3.2 experimentally validate Theorem 1. From this and the previous experiment, it seems that recoverability is best when StoCoSaMP employs a combination of greedy and random steps. To the practitioner, we suggest to use $0.2 \leq P_R \leq 0.6$, where both high success rates and computational gains are apparent.⁷ The theoretical justification is unclear and would be interesting to explore in future work.

⁶ The variation of success for different dimensions for both StoCoSaMP and CoSaMP is inherent to the performance characteristics of CoSaMP itself.

⁷ This does not hold for the smallest $N = 1K$ problems, which on the other hand are the least interesting from a scalability point of view.

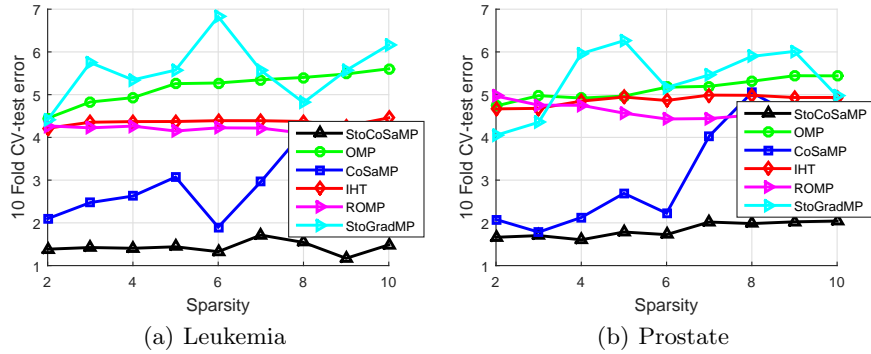


Fig. 6. 10-fold cross-validation testing error on the gene-expression datasets for (a) Leukemia data (72×5147) (b) and Prostate data (102×12533).

3.4 Effect of Noise on Recoverability

Goal. We now focus on noisy measurement signals. We also evaluate the performance of SWAP when it is initialized by CoSaMP and StoCoSaMP. SWAP is a greedy algorithm; it can be used to fine-tune the solutions of other recovery algorithms [2]. Here, we use SWAP to investigate whether a solution obtained by StoCoSaMP on average lies in a better basin than a solution obtained by CoSaMP. Intuitively, SWAP’s explicit greediness forces the solution within each basin towards the local optimum of that particular basin.

Method and Data. We experiment with three noise levels in the measurement vector 50 dB (low noise), 40 dB, 33 dB (high noise). We construct 100 synthetic problems with the measurement matrix and the sparse vectors being sampled i.i.d. from $N(0,1)$. We add varying amounts of white Gaussian noise to our measurement vectors, such that the SNR is at one of the three levels of noise. For all problems, we set $P_R = 0.3$ in StoCoSaMP and $N = 200$.

Results. We report the error and SNR statistics and the corresponding histograms for the four algorithmic combinations in Table 1 and Fig. 5. Table 1 suggests that StoCoSaMP achieves higher quality recoveries (SNR) on average compared to CoSaMP over all problems ($\text{SNR} \geq 33$ dB) in the presence of varying levels of noise. In fact, StoCoSaMP in most cases performs better on average than SWAP initialized with CoSaMP. This is powerful, since StoCoSaMP is also computationally less expensive than either CoSaMP (purely greedy) or SWAP (exponential complexity). The histogram plots in Fig. 5 clearly show that both StoCoSaMP and StoCoSaMP + SWAP tend to achieve much lower errors than their CoSaMP counterparts. This experiment suggests that the advantages of StoCoSaMP extend to situations with significant noise.

3.5 Real-World Data: Classifying Gene Expression

Goal. To evaluate StoCoSaMP’s real-world classification performance, we study two large-scale gene expression cancer datasets. We compare StoCoSaMP, CoSaMP, StoGradMP, StoIHT, OMP, ROMP, IHT, and AMP.

Method and Data. The datasets contain the gene expression levels for leukemia ($M = 72$ and $N = 5147$) and prostate cancer ($M = 102$ and $N = 12533$) with a binary label.⁸ This is a classification problem where unseen gene expression values are to be classified. We learn a K -sparse linear classifier for this experiment. The labels act as the signal to be expressed as a linear combination of the gene expressions. The linear combination is parameterized by a weight vector that is recovered by a sparse signal recovery algorithm. Following a previous study [23], we explore sparsities ranging from $K = 2$ to $K = 10$.

AMP determines the sparsity level internally through soft thresholding. For AMP, unlike for the other algorithms, we do not enforce K -sparsity. We perform 10-fold cross validation for each level of sparsity (20 trials for StoCoSaMP, StoIHT and StoGradMP for each sparsity) with $P_R = 0.5$ for StoCoSaMP. For StoGradMP and StoIHT, we set the block size to $\min(K, M)$ as in previous work [17]. We then pick the classifier (sparse solution) that minimizes the training error as our model for testing for all stochastic algorithms.

Results. Fig. 6(a) and Fig. 6(b) show the experimental results for all algorithms for the two datasets. StoCoSaMP’s results (black line) are a consistent improvement over all other algorithms (including StoGradMP) in all cases. AMP achieved an error of 11.10 (with $K = 62$) on the Leukemia dataset and 31.35 (with $K = 89$) on the Prostate dataset, worse than StoCoSaMP. StoIHT performed worse in all cases and reported an error consistently above 10, and is not plotted in the figure.

4 Conclusion

In this paper, we present a Stochastic CoSaMP (StoCoSaMP) method. Under RIP, even a purely random version of StoCoSaMP ($P_R = 1$) will observe support convergence. This provides an interesting addition to previous results, which have suggested that greed is good for recovery [21]. Our experiments show that StoCoSaMP out-performs CoSaMP on a variety of signal recovery problems, and other algorithms (including CoSaMP, StoGradMP and StoIHT) on a real-world large scale classification task.

Acknowledgements. We thank Aswin C. Sankaranarayanan for his helpful comments and for checking the correctness of the proofs.

References

1. Andrecut, M.: Stochastic recovery of sparse signals from random measurements. *Engineering Letters* 19(1), 1–6 (2011)

⁸ See <http://www.biomed.gov/supp/bi-cancer/projections/info/leukemia.htm> and [/prostate.htm](http://www.biomed.gov/supp/bi-cancer/projections/info/prostate.htm)

2. Blumensath, T., Davies, M.E.: Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis* 27(3), 265 – 274 (2009)
3. Candes, E.: Compressive sampling. In: *International Congress of Mathematicians*. vol. 3, pp. 1433–1452 (2006)
4. Candes, E., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory* 52(2), 489–509 (2006)
5. Candes, E., Tao, T.: Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory* 52(12), 5406–5425 (2006)
6. Donoho, D., Tanner, J.: Precise undersampling theorems. *Proceedings of the IEEE* 98(6), 913–924 (June 2010)
7. Garg, R., Khandekar, R.: Gradient descent with sparsification: An iterative algorithm for sparse recovery with restricted isometry property. pp. 337–344. *ICML '09*, New York, NY, USA (2009)
8. Hoos, H.H., Stützle, T.: *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, San Francisco (2005)
9. Maleki, A., Donoho, D.L.: Optimally tuned iterative reconstruction algorithms for compressed sensing. *IEEE Journal of Selected Topics in Signal Processing* 4(2), 330–341 (2010)
10. Mengshoel, O.J.: Understanding the role of noise in stochastic local search: Analysis and experiments. *Artificial Intelligence* 172(8-9), 955–990 (2008)
11. Mengshoel, O.J., Wilkins, D.C., Roth, D.: Initialization and restart in stochastic local search: Computing a most probable explanation in bayesian networks. *IEEE Transactions on Knowledge and Data Engineering* 23(2), 235–247 (Feb 2011)
12. Moussallam, M., Daudet, L., Richard, G.: Matching pursuits with random sequential subdictionaries. *Signal Processing* 92(10), 2532–2544 (2012)
13. Needell, D.: *Topics in Compressed Sensing*. ArXiv e-prints (May 2009)
14. Needell, D., Vershynin, R.: Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of Selected Topics in Signal Processing* 4(2), 310–316 (2010)
15. Needell, D., Tropp, J.A.: Cosamp: iterative signal recovery from incomplete and inaccurate samples. *Commun. ACM* 53(12), 93–100 (Dec 2010)
16. Nguyen, N., Chin, S., Tran, T.: A unified iterative greedy algorithm for sparsity-constrained optimization (2012)
17. Nguyen, N., Needell, D., Woolf, T.: Linear convergence of stochastic iterative greedy algorithms with sparse constraints. arXiv preprint arXiv:1407.0088 (2014)
18. Park, J.D., Darwiche, A.: Complexity results and approximation strategies for MAP explanations. *Journal of Artificial Intelligence Research* 21, 101–133 (2004)
19. Peel, T., Emiya, V., Ralaivola, L., Anthoine, S.: Matching pursuit with stochastic selection. In: *EUSIPCO*. pp. 879–883. *IEEE* (2012)
20. Selman, B., Kautz, H.A., Cohen, B.: Noise strategies for improving local search. In: *Proc. AAAI-94*. pp. 337–343 (1994)
21. Tropp, J.A.: Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory* 50(10), 2231–2242 (2004)
22. Tropp, J.A., Gilbert, A.C.: Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions of Information Theory* 53, 4655–4666 (2007)
23. Vats, D., Baraniuk, R.: When in doubt, swap: High-dimensional sparse recovery from correlated measurements. In: *NIPS* (2013)