Performance Guarantees for Undersampled Recursive Sparse Recovery in Large but Structured Noise

Brian Lois, Namrata Vaswani, and Chenlu Qiu
Iowa State University, Ames IA

***Abstract*—We study the problem of recursively reconstructing a time sequence of sparse vectors S_t from measurements of the form $M_t = AS_t + BL_t$ where A and B are known measurement matrices, and L_t lies in a slowly changing low dimensional subspace. We assume that the signal of interest (S_t) is sparse, and has support which is correlated over time. We introduce a solution which we call Recursive Projected Modified Compressed Sensing (ReProMoCS), which exploits the correlated support change of S_t. We show that, under weaker assumptions than previous work, with high probability, ReProMoCS will exactly recover the support set of S_t and the reconstruction error of S_t is upper bounded by a small time-invariant value. A motivating application where the above problem occurs is in functional MRI imaging of the brain to detect regions that are "activated" in response to stimuli. In this case both measurement matrices are the same (i.e. $A = B$). The active region image constitutes the sparse vector S_t and this region changes slowly over time. The background brain image changes are global but the amount of change is very little and hence it can be well modeled as lying in a slowly changing low dimensional subspace, i.e. this constitutes L_t.**

I. Introduction

We study the problem of recursively reconstructing a time sequence of sparse vectors S_t from under-sampled measurements corrupted by (possibly) large magnitude but low-dimensional noise. By low-dimensional we mean that the L_t's all lie in a low-dimensional subspace, which is allowed to change slowly over time. Specifically we observe

$$M_t := AS_t + BL_t$$

where A and B are known matrices, and L_t is (potentially) large but low-dimensional and non-sparse noise. If we let $\tilde{L}_t := BL_t$ then M_t can also be expressed as $AS_t + \tilde{L}_t$. This problem has applications in fMRI imaging where the "active" region of the brain is the sparse signal of interest, and the background brain image is slowly changing low dimensional background. In this case, $B = A$ is the partial Fourier matrix. Another application is separating the foreground and background of a sequence of single-pixel images. Some experiments are available in [1].

A. Related Work and Contribution

This work is related to [2], [3] which study the problem of decomposing an observed matrix $M = [M_1 \cdots M_t]$ into the sum of a sparse matrix $S = [S_1 \cdots S_t]$ and a low rank matrix $L = [L_1 \cdots L_t]$. These papers can all be viewed as batch solutions to the problem studied here. A limitation of these works is that they either require independence over time of the support of S [2] or stronger restrictions on the sparsity pattern of S [3], [4]. A more detailed discussion is provided in [5]. Both of these are not practical assumptions for image analysis because foreground objects often move in a correlated fashion over time. This can result in some rows of S having many non-zero entries, while other rows have very few or no non-zero entries. These papers also do not consider the case where the sparse vectors are undersampled. Other related works, some of which consider the under-sampled case, include [4], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16].

The problem of recursively reconstructing sparse vectors in large but low-dimensional noise was studied in [17], where the solution called Recursive Projected CS (ReProCS) was presented and analyzed. This work extends these results in two ways. First, we study the case where the sparse and low rank vectors may be under-sampled (multiplied by the matrices A and B). To the best of our knowledge this is the first work that analyzes recursive sparse recovery from undersampled measurements in the presence of large but structured (low-dimensional) noise. Second, we modify the algorithm in [17] by using modified-CS [18] in place of the simple ℓ_1 minimization (basis pursuit denoising) step in ReProCS. This allows us to prove the same performance guarantees under weaker assumptions as long as the support of S_t changes slowly over time.

A key difference of our work compared with most existing work analyzing finite sample PCA, e.g. [19], and references therein, is that in these works the noise/error in the observed data is independent of the true (noise-free) data. However, in our case, because of how $\hat{\tilde{L}}_t$ is computed, the error $e_t = \tilde{L}_t - \hat{\tilde{L}}_t$ is correlated with $\tilde{L}_t$. As a result the tools developed in these earlier works cannot be used for our problem. This is the main reason we need to develop and analyze projection-PCA based approaches for subspace addition.

B. Notation

For a matrix M, M' denotes its transpose and $M^\dagger$ denotes is Moore-Penrose pseudo-inverse. For an Hermitian matrix H, we use $H \overset{\text{EVD}}{=} U\Lambda U'$ to mean the eigenvalue decomposition where U is a unitary matrix, and Λ is diagonal with entries arranged in decreasing order. For an indexing set T and matrix

M, M_T is the matrix formed by retaining the columns of M indexed by T. We use $\operatorname{range}(M)$ to refer to the subspace spanned by the columns of M.

Definition 1.1. *We refer to a matrix P as a* basis matrix *if $P'P = I$.*

Definition 1.2. *For a matrix C we use the notation $Q = \operatorname{basis}(C)$ to mean: Q is a matrix such that the columns of Q form an orthonormal basis for $\operatorname{range}(C)$ i.e. $Q'Q = I$ and $\operatorname{range}(Q) = \operatorname{range}(C)$.*

Definition 1.3. *The s-restricted isometry constant (RIC) [20] δ_s for an $n\times m$ matrix Ψ is the smallest real number satisfying $(1-\delta_s)\|x\|_2^2 \leq \|\Psi x\|_2^2 \leq (1+\delta_s)\|x\|_2^2$ for all s-sparse vectors x.*

II. Problem Formulation and Model Assumptions

The measurement vector at time t, M_t, is an m-dimensional vector which is constituted as $M_t = AS_t + BL_t$. A and B are known $m \times n$ matrices where n can be greater than m. S_t is a sparse vector in $\mathbb{R}^n$ with support size at most s and whose non-zero entries have magnitude at least $S_{\min}$. Furthermore, the number of elements that enter the support and the number of elements that exit the support at a given time are both bounded by s_a. L_t is a dense vector which lies in a slowly changing low-dimensional subspace of $\mathbb{R}^n$. Since all of the L_t's lie in a low dimensional subspace so do the vectors BL_t. Therefore let $\tilde{L}_t = BL_t$, then the observed vector can be expressed as $M_t = AS_t + \tilde{L}_t$.

We assume that we have access to an accurate estimate of the subspace in which the first t_{train} vectors $\tilde{L}_t$ lie. That is we have a basis matrix $\hat{Q}_0$ such that $\|(I - Q_0'Q_0)\hat{Q}_0\|_2$ is small. Here $Q_0 = \text{basis}([\tilde{L}_1, \tilde{L}_2, \ldots, \tilde{L}_{t_{\text{train}}}])$. Our goal is to estimate S_t at every time $t > t_{\text{train}}$.

Notation for S_t. Let $T_t := \{i \in \{1, \ldots, n\} | (S_t)_i \neq 0\}$ be the support of S_t. At each time $t > t_{\text{train}}$ T_t changes as: $T_t = (T_{t-1} \cup T_{\text{add},t}) \setminus T_{\text{del},t}$ where $T_{\text{add},t}$ are the newly added elements of the support, and $T_{\text{del},t}$ are the elements deleted from the support at time t. Define $s := \max_t |T_t|$ and and $s_a := \max_t \max \{|T_{\text{add},t}|, |T_{\text{del},t}|\}$ Let $S_{\min} := \min\limits_t \min\limits_{i \in T_t} |(S_t)_i|$.

A. Model on L_t

1) The L_t's lie in a low-dimensional subspace that changes every so often. Let $P_{(t)}$ be a basis matrix for this subspace at time t. Then $L_t = P_{(t)} a_t$ where $P_{(t)}$ changes every so often. Denote the subspace change times by t_j, $j = 0, 1, 2, \cdots, J$. (There are a maximum of J subspace change times.) For $t_j \leq t \leq t_{j+1}$, $P_{(t)} = P_j$ where P_j is an $m \times r_j$ basis matrix with $r_j \ll m$ and $r_j \ll (t_{j+1} - t_j)$.
2) At the change times, t_j, P_j changes as $P_j = [P_{j-1} \;\; P_{j,\text{new}}]$ where $P_{j,\text{new}}$ is an $m \times c_{j,\text{new}}$ basis matrix with columns orthogonal to the columns of P_{j-1}. Therefore $r_j = r_{j-1} + c_{j,\text{new}}$. We can then write $a_t = \begin{bmatrix} a_{t,*} \\ a_{t,\text{new}} \end{bmatrix}$ conformal with the partition of P_j.
3) There exists a constant c_{mx} such that $0 \leq c_{j,\text{new}} \leq c_{mx}$.
4) The vector of coefficients $a_t = P_{(t)}' L_t$ is a zero mean bounded random variable. That is $\mathbf{E}(a_t) = 0$ and there is a constant γ_* such that $\|a_t\|_\infty \leq \gamma_*$ for all t. The covariance matrix $\Lambda_t := \text{Cov}[a_t] = \mathbf{E}(a_t a_t')$ is diagonal with $\lambda^- := \min_t \lambda_{\min}(\Lambda_t) > 0$ and $\lambda^+ := \max_t \lambda_{\max}(\Lambda_t) < \infty$. Thus the condition number of any Λ_t is bounded by $f := \frac{\lambda^+}{\lambda^-}$.
5) Finally, the a_t's are mutually independent over time.

Definition 2.1. *Define $Q_j = \operatorname{basis}(BP_j)$, and $Q_{j,new} = basis\big((I - Q_{j-1}Q_{j-1}{}')BP_j\big)$.*

Notice that $Q_{j-1}' Q_{j,\text{new}} = 0$.

B. Slow Subspace Change

By slow subspace change we mean all of the following:

1) The delay between consecutive subspace change times $(t_{j+1} - t_j)$ is large enough.
2) The vector of coefficients for the new directions, $a_{t,\text{new}}$ is initially small. That is $\max\limits_{t_j \leq t < t_j + \alpha} \|a_{t,\text{new}}\|_\infty \leq \gamma_{\text{new}}$ with $\gamma_{\text{new}} \ll \gamma_*$ and $\gamma_{\text{new}} \ll S_{\min}$, but can increase gradually. This is modeled as follows. Split the interval $[t_j, t_{j+1} - 1]$ into α length periods. We assume that $\max_j \max_{t \in [t_j + (k-1)\alpha, t_j + k\alpha - 1]} \|a_{t,\text{new}}\|_\infty \leq \min(v^{k-1}\gamma_{\text{new}}, \gamma_*)$ for a $v > 1$ but not too large.
3) The number of newly added directions is small i.e. $c_{j,\text{new}} \leq c_{mx} \ll r_0$.

C. Matrix Coherence Parameter and its Relation with RIC

Definition 2.2. *For an $m \times n$ matrix C and $m \times p$ matrix A define the incoherence coefficient $\kappa_{s,A}(C)$ by*

$$\kappa_{s,A}(C) := \max_{|T| \leq s} \|A_T{}' \operatorname{basis}(C)\|_2$$

and let $\kappa_s(C) := \kappa_{s,I}(C)$.

$\kappa_{s,A}(C)$ can be thought of as a measure of the coherence between $\operatorname{range}(A_T)$ and $\operatorname{range}(C)$ for any set T of size less than or equal to s. When $A = I$, $\kappa_s(C)$ measures the denseness of columns of $\text{basis}(C)$ and hence in [17] it was referred to as the denseness coefficient. Notice that, $\|A_T{}'C\|_2 \leq \kappa_{s,A}(C)\|C\|_2$ for any set T such that $|T| \leq s$.

Lemma 2.3. *For a basis matrix P ($P'P = I$) and matrix A. $\delta_s((I - PP')A) \leq \delta_s(A) + (\kappa_{s,A}(P))^2$.*

III. Recursive Projected Modified CS

The Recursive Projected Modified CS algorithm is summarized in Algorithm 2. First we need the following definition

Definition 3.1. *Define the time interval $\mathcal{I}_{j,k} := [t_j + (k-1)\alpha, t_j + k\alpha - 1]$ for $k = 1, \ldots K$ and $\mathcal{I}_{j,K+1} := [t_j + K\alpha, t_{j+1} - 1]$ where K is the algorithm parameter in Algorithm 1.*

A. The Projection-PCA algorithm

Given a data matrix $\mathcal{D}$, a basis matrix P and an integer r, projection-PCA (proj-PCA) applies PCA on $\mathcal{D}_{\text{proj}} := (I - PP')\mathcal{D}$.

Algorithm 2 Recursive Projected Modified CS (ReProMoCS)

Parameters: algorithm parameters: ξ, ω, α, K, model parameters: t_j, r_0, $c_{j,\text{new}}$ (set as in Theorem 4.1 or as in [17, Sec X-B] when the model is not known)

Input: M_t, *Output:* $\hat{S}_t$, $\hat{\tilde{L}}_t$, $\hat{Q}_{(t)}$

Initialization: Compute $\hat{Q}_0 \leftarrow$ proj-PCA$\Big([\tilde{L}_1, \tilde{L}_2, \cdots, \tilde{L}_{t_{\text{train}}}], [.], r_0\Big)$, and set $\hat{Q}_{(t)} \leftarrow \hat{Q}_0$.

Let $\hat{T}_{t_{\text{train}}+1} = \emptyset$, and let $j \leftarrow 1$, $k \leftarrow 1$.

For $t > t_{\text{train}}$, do the following:

1) Estimate N_t and S_t via Projected Modified CS:
   a) Nullify most of $\tilde{L}_t$: compute $\Phi_{(t)} \leftarrow I - \hat{Q}_{(t-1)}\hat{Q}'_{(t-1)}$, compute $y_t \leftarrow \Phi_{(t)} M_t$
   b) Sparse Recovery: compute $\hat{S}_{t,\text{modcs}}$ as the solution of $\min_x \|x_{\hat{T}_t^c}\|_1 \ s.t. \ \|y_t - \Phi_{(t)} A x\|_2 \leq \xi$
   c) Support Estimate: compute $\hat{N}_t = \{i : |(\hat{S}_{t,\text{modcs}})_i| > \omega\}$, and set $\hat{T}_{t+1} = \hat{N}_t$
   d) LS Estimate of S_t: compute $(\hat{S}_t)_{\hat{N}_t} = ((\Phi_{(t)} A)_{\hat{N}_t})^{\dagger} y_t, \ (\hat{S}_t)_{\hat{N}_t^c} = 0$
2) Estimate $\tilde{L}_t$: $\hat{\tilde{L}}_t = M_t - A\hat{S}_t$.
3) Update $\hat{Q}_{(t)}$ by K Projection PCA steps.
   a) If $t = t_j + k\alpha - 1$,
      i) $\hat{Q}_{j,\text{new},k} \leftarrow$ proj-PCA$\left(\left[\hat{\tilde{L}}_{t_j+(k-1)\alpha}, \ldots, \hat{\tilde{L}}_{t_j+k\alpha-1}\right], \hat{Q}_{j-1}, c_{j,\text{new}}\right)$,
      ii) set $\hat{Q}_{(t)} \leftarrow [\hat{Q}_{j-1} \ \hat{Q}_{j,\text{new},k}]$; increment $k \leftarrow k+1$.

      Else
      i) set $\hat{Q}_{(t)} \leftarrow \hat{Q}_{(t-1)}$.
   b) If $t = t_j + K\alpha - 1$, then set $\hat{Q}_j \leftarrow [\hat{Q}_{j-1} \ \hat{Q}_{j,\text{new},K}]$. Increment $j \leftarrow j+1$. Reset $k \leftarrow 1$.
4) Increment $t \leftarrow t+1$ and go to step 1.

Algorithm 1 projection-PCA: $Q \leftarrow$ proj-PCA$(\mathcal{D}, P, r)$

1) Projection: compute $\mathcal{D}_{\text{proj}} \leftarrow (I - PP')\mathcal{D}$
2) PCA: $\frac{1}{\alpha_{\mathcal{D}}}\mathcal{D}_{\text{proj}}{\mathcal{D}_{\text{proj}}}' \overset{EVD}{=} \begin{bmatrix} Q \ Q_{\perp} \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & \Lambda_{\perp} \end{bmatrix} \begin{bmatrix} Q' \\ {Q_{\perp}}' \end{bmatrix}$ where Q is an $n \times r$ basis matrix and $\alpha_{\mathcal{D}}$ is the number of columns in $\mathcal{D}$.

B. ReProMoCS

The idea behind ReProMoCS (Algorithm 2) is as follows. Assume that the current basis matrix, $\hat{Q}_{(t-1)}$ is an accurate estimate of the current subspace where the $\tilde{L}_t$'s lie. We project the measurement M_t perpendicular to $\operatorname{range}\hat{Q}_{(t-1)}$ in order to nullify most of $\tilde{L}_t$ and obtain $y_t := \Phi_{(t)} M_t$ where $\Phi_{(t)} = I - \hat{Q}_{(t-1)}\hat{Q}'_{(t-1)}$. Since the projection matrix only has rank $n - r^*$ where $r^* = \operatorname{rank}(\hat{Q}_{(t-1)})$, there are only $n - r^*$ "effective" linear measurements. Recovering the n dimensional sparse vector S_t now becomes a sparse recovery problem in small noise. Using the support estimate from the previous time we use the modified-CS algorithm to recover $\hat{S}_{t,\text{modcs}}$ and estimate its support by thresholding on the recovered vector. By computing a least squares (LS) estimate of S_t on the estimated support and setting it to zero everywhere else (step 1d), we can get a more accurate final estimate, $\hat{S}_t$. This $\hat{S}_t$ is used to estimate $\tilde{L}_t$ as $\hat{\tilde{L}}_t = M_t - A\hat{S}_t$ (step 2). The sparse recovery error is then $e_t := S_t - \hat{S}_t$. Since $\hat{\tilde{L}}_t = M_t - A\hat{S}_t$, e_t also satisfies $Ae_t = \hat{\tilde{L}}_t - BL_t$. Thus, a small e_t means that $\tilde{L}_t$ is also recovered accurately. The estimated $\hat{\tilde{L}}_t$'s are used to obtain new estimates of $Q_{j,\text{new}}$ every α frames for a total of $K\alpha$ frames via a modification of the standard PCA procedure, which we call projection PCA (step 3).

IV. Performance Guarantees

This result says that if (a) the algorithm parameters are set appropriately; (b) The RIC of A is small enough; (c) the basis matrices for the previous subspace, the newly added subspace, and the unestimated part of the newly added subspace are sufficiently incoherent with A; (d) the low-dimensional subspace changes slowly enough; (e) the condition number of the covariance matrix of $a_{t,\text{new}}$ is small enough, then with high probability, the algorithm exactly recovers the support of S_t at all times t. A detailed discussion of the assumptions and how they compare with other results is available in [5].

Theorem 4.1. *Consider Algorithm 2. Let $c := c_{mx}$ and $r := r_0 + (J-1)c$. Assume that L_t obeys the model given in Sec. II-A and there are a total of J change times. Assume also that the initial subspace estimate is accurate enough, i.e. $\|(I - \hat{Q}_0\hat{Q}_0')Q_0\| \leq r_0\zeta$, for a ζ that satisfies*

$$\zeta \leq \min\left(\frac{10^{-4}}{r^2}, \frac{1.5 \times 10^{-4}}{r^2 f}, \frac{1}{r^3\gamma_*^2}\right) \ \textit{where} \ f := \frac{\lambda^+}{\lambda^-}$$

If the following conditions hold:

1) *the algorithm parameters are set as $\xi = \xi_0(\zeta)$, $\frac{\varrho}{\sqrt{s_\Delta}}7.50\xi \leq \omega < S_{\min} - \frac{\varrho}{\sqrt{s_\Delta}}7.50\xi$, $K = K(\zeta)$, $\alpha \geq \alpha_{add,A} = 1.05\alpha_{add}(\zeta)$, where*

$\xi_0(\zeta), \varrho, K(\zeta), \alpha_{add}(\zeta)$ are defined in Definition 4.2.

2) *A, P_{j-1}, $P_{j,new}$, $D_{j,new,k} := (I - \hat{Q}_{j-1}\hat{Q}'_{j-1} - \hat{Q}_{j,new,k}\hat{Q}'_{j,new,k})Q_{j,new}$, and $G_{j,new,k} := (I - P_{j,new}{P_{j,new}}')\hat{P}_{j,new,k}$ satisfy $\delta_{s+3s_\Delta}(A) \leq 0.05$, $\kappa_{s+3s_\Delta,A}(Q_{J-1}) \leq 0.22$, $\max_j \kappa_{s+3s_\Delta,A}(Q_{j,new}) \leq 0.14$, $\max_j \max_{0\leq k\leq K} \kappa_{s+3s_\Delta,A}(D_{j,new,k}) \leq 0.15$, $\max_j \max_{0\leq k\leq K} \kappa_{s+3s_\Delta,A}(G_{j,new,k}) \leq 0.14$, $\max_j \max_{0\leq k\leq K} \kappa_{s+3s_\Delta,A}(D_{j,*,k}) \leq 1$ with $\hat{P}_{j,new,0} = [.]$ (empty matrix).*
3) *for a given value of $S_{\min}$, the subspace change is slow enough, i.e. $\min_j(t_{j+1} - t_j) > K\alpha$, $\max_j \max_{t\in\mathcal{I}_{j,k}} \|a_{t,new}\|_\infty \leq \min(1.2^{k-1}\gamma_{new}, \gamma_*)$, $14\rho\xi_0(\zeta) \leq S_{\min}$,*
4) *the condition number of the covariance matrix of $a_{t,new}$ is bounded, i.e. $\frac{\lambda_{\max}[\mathrm{Cov}(a_{t,new})]}{\lambda_{\min}[\mathrm{Cov}(a_{t,new})]} \leq \sqrt{2}$*

then, with probability at least $(1 - n^{-10})$,

1) *at all times, t, $\hat{T}_t = T_t$ and $\|e_t\|_2 = \|\hat{S}_t - S_t\|_2 \leq 0.18\sqrt{c}\gamma_{new} + 1.2\sqrt{\zeta}(\sqrt{r} + 0.06\sqrt{c})$.*
2) *the subspace error $SE_{(t)} := \|(I - \hat{Q}_{(t)}\hat{Q}'_{(t)})Q_{(t)}\|_2$ satisfies the bounds given in [17, Theorem 18].*

The proof can be found in [5].

Definition 4.2. 1) *Define $K(\zeta) := \left\lceil \frac{\log(0.6c\zeta)}{\log 0.6} \right\rceil$*

2) *Define $\xi_0(\zeta) := \sqrt{c}\gamma_{new} + \sqrt{\zeta}(\sqrt{r} + \sqrt{c})$*
3) *Define ϱ to be the smallest real number such that $\|S_t - \hat{S}_{t,\mathrm{modcs}}\|_\infty \leq \frac{\varrho}{\sqrt{s_\Delta}}\|S_t - \hat{S}_{t,\mathrm{modcs}}\|_2$ for all t. Notice that $\varrho \leq \sqrt{s_\Delta}$ because the infinity norm is always less than or equal to the two norm.*
4) *Let $K = K(\zeta)$. Define*
$$\alpha_{add} = \left\lceil \frac{4608(\log 6KJ + 11\log n)}{\zeta^2(\lambda^-)^2} \max\left(\min(1.2^{4K}\gamma_{new}^4, \gamma_*^4), \frac{16}{c^2}, 4(0.186\gamma_{new}^2 + 0.0034\gamma_{new} + 2.3)^2\right)\right\rceil.$$
In words, α_{add} is the smallest value of the number of data points, α, needed for one projection PCA step to ensure that Theorem 4.1 holds w.p. at least $(1 - n^{-10})$.

V. Proof Outline

The proof of Theorem 4.1 essentially follows from two main lemmas. The first Lemma gives an exponentially decaying upper bound on a high probability[1] upper bound for the subspace error under the assumptions of the Theorem. The other main lemma says that if, during the time interval $\mathcal{I}_{j,k-1}$, the algorithm has worked well (recovered the support of S_t exactly and recovered the background subspace with small enough error), then it will also work well in $\mathcal{I}_{j,k}$ w.h.p.. The proof of this lemma requires two lemmas: one for the projected CS step and one for the projection PCA step of the algorithm. The proof the CS lemma is fairly strightforward and uses the ModCS error bound in [18]. The proof of the subspace lemma is longer and uses the $\sin\theta$ theorem [21] and matrix Hoeffding inequalities [22].

[1]We choose α_{add} (the amount of time between projection-PCA steps) so that the conclusions of Theorem 4.1 hold with probability at least $1-n^{-10}$).

References

[1] C. Qiu and N. Vaswani, "Recursive sparse recovery in large but correlated noise," in *48th Allerton Conference on Communication Control and Computing*, 2011.

[2] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of ACM*, vol. 58, no. 3, 2011.

[3] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, 2011.

[4] H. Xu, C. Caramanis, and S. Sanghavi, "Robust pca via outlier pursuit," *IEEE Tran. on Information Theorey*, vol. 58, no. 5, 2012.

[5] B. Lois, N. Vaswani, and C. Qui, "Performance guarantees for undersampled recursive sparse recovery in large but structured noise (long version)." [Online]. Available: http://www.public.iastate.edu/%7Eblois/ReProModCSLong.pdf

[6] M. McCoy and J. Tropp, "Two proposals for robust pca using semidefinite programming," *arXiv:1012.1086v3*, 2010.

[7] M. B. McCoy and J. A. Tropp, "Sharp recovery bounds for convex deconvolution, with applications," *arXiv:1205.1580*.

[8] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Foundations of Computational Mathematics*, no. 6, 2012.

[9] Y. Hu, S. Goud, and M. Jacob, "A fast majorize-minimize algorithm for the recovery of sparse and low-rank matrices," *IEEE Transactions on Image Processing*, vol. 21, no. 2, p. 742=753, Feb 2012.

[10] A. E. Waters, A. C. Sankaranarayanan, and R. G. Baraniuk, "Sparcs: Recovering low-rank and sparse matrices from compressive measurements," in *Proc. of Neural Information Processing Systems(NIPS)*, 2011.

[11] E. Richard, P.-A. Savalle, and N. Vayatis, "Estimation of simultaneously sparse and low rank matrices," *arXiv:1206.6474, appears in Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*.

[12] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with outliers," *arXiv:1011.1518*.

[13] M. Mardani, G. Mateos, and G. B. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *arXiv:1204.6537*.

[14] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *arXiv:1202.4596*.

[15] A. Ganesh, K. Min, J. Wright, and Y. Ma, "Principal component pursuit with reduced linear measurements," *arXiv:1202.6445*.

[16] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 57–81, 2011.

[17] C. Qiu, N. Vaswani, and L. Hogben, "Recursive robust pca or recursive sparse recovery in large but structured noise," in *IEEE Intl. Conf. Acoustics, Speech, Sig. Proc. (ICASSP)*, 2013, longer version in arXiv: 1211.3754 [cs.IT].

[18] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Processing*, September 2010.

[19] B. Nadler, "Finite sample approximation results for principal component analysis: A matrix perturbation approach," *The Annals of Statistics*, vol. 36, no. 6, 2008.

[20] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Info. Th.*, vol. 51(12), pp. 4203 – 4215, Dec. 2005.

[21] C. Davis and W. M. Kahan, "The rotation of eigenvectors by a perturbation. iii," *SIAM Journal on Numerical Analysis*, Mar. 1970.

[22] J. A. Tropp, "User-friendly tail bounds for sums of random matrices," *Foundations of Computational Mathematics*, vol. 12, no. 4, 2012.