

Recursive Sparse Recovery in Large but Correlated Noise

Chenlu Qiu and Namrata Vaswani

Dept. of Electrical and Computer Engineering, Iowa State University
 Ames, IA, 50010
 {chenlu, namrata}@iastate.edu

Abstract—In this work, we focus on the problem of recursively recovering a time sequence of sparse signals, with time-varying sparsity patterns, from highly undersampled measurements corrupted by very large but correlated noise. It is assumed that the noise is correlated enough to have an approximately low rank covariance matrix that is either constant, or changes slowly, with time. We show how our recently introduced Recursive Projected CS (ReProCS) and modified-ReProCS ideas can be used to solve this problem very effectively. To the best of our knowledge, except for the recent work of dense error correction via ℓ_1 minimization, which can handle another kind of large but “structured” noise (the noise needs to be sparse), none of the other works in sparse recovery have studied the case of any other kind of large noise.

I. INTRODUCTION

In this work we focus on the recursive sparse recovery problem in large noise. To be precise, given

$$y_t := \Psi \Phi s_t + l_t \quad (1)$$

where Ψ is an $m \times n$ measurement matrix with $m \ll n$, Φ is an $n \times n$ orthonormal matrix (sparsity basis matrix), y_t is an $m \times 1$ measurement vector, s_t is an $n \times 1$ sparse vector, and l_t is the noise. The signal, $z_t := \Phi s_t$, is sparse in some transform domain specified by Φ . The signal is usually correlated over time, and as a result the support of s_t , denoted by T_t , often changes in a correlated fashion over time. Our goal is to recursively recover s_t , and hence $z_t = \Phi s_t$, in the presence of large noise, i.e. in the case where

$$\|l_t\|_2 \gg \|\Psi \Phi s_t\|_2.$$

In the most general case, clearly this problem cannot be solved. All sparse recovery and recursive sparse recovery techniques [1], [2], [3], [4], [5], [6], [7], [8] studied so far can only handle small to medium noise. In any case none of them, will work if the noise energy is much larger than the signal energy. In fact, this problem is a difficult one even in the case where $m = n$ and Ψ is an identity matrix [9], [10]. In this work we show that we can indeed solve this problem if the noise is correlated enough so that

- 1) the matrix $[l_{t-\tau}, \dots, l_t]$ has low rank, i.e., $\text{rank}([l_{t-\tau}, \dots, l_t]) \ll m$ for τ large enough and
- 2) the subspace in which the noise lies only changes slowly over time and we have available some initial noise-only

This research was partially supported by NSF grants ECCS-0725849 and CCF-0917015.

training data that helps us get an initial estimate of the noise subspace.

- 3) We can further improve our solutions if the support of s_t also changes slowly over time or changes in some other correlated fashion over time.

If a sequence of vectors l_t satisfy the first two assumptions above, we refer to them as a “low rank” sequence.

The problem defined above is the case where the measurement noise, l_t , is large but correlated. However it can also model the the case where the unknown signal, z_t , that we are imaging, consists of the “signal of interest”, Φs_t plus a background signal, b_t , i.e. $z_t = \Phi s_t + b_t$ and b_t has large magnitude (can be much larger than Φs_t) but is “low rank”. In this case the measurement,

$$y_t := \Psi(\Phi s_t + b_t) = \Psi \Phi s_t + \Psi b_t \quad (2)$$

Notice that (2) can be rewritten in the form (1) by letting $l_t = \Psi b_t$. Clearly if b_t is “low rank”, then so is $l_t = \Psi b_t$.

To make things precise, we can rewrite b_t as

$$b_t = U x_t$$

where U is an *unknown* orthonormal matrix and x_t is an $n \times 1$ sparse vector whose support set, denoted by N_t , changes every d time units, and whose elements are spatially uncorrelated. In other words, N_t is piecewise constant with time. Thus, the columns of the sub-matrix, $\Psi(U)_{N_t}$, span the low dimensional subspace in which the current set of l_t 's lie and $l_t = \Psi U x_t = \Psi(U)_{N_t}(x_t)_{N_t}$. Every d time units, there are k additions to the set N_t , or, equivalently, k directions get added to the subspace of l_t . When a new direction gets added, the magnitude of x_t along it is initially small but gradually increases to a larger stable value. Also, the values of x_t along k existing directions gradually decay down to zero, i.e. the corresponding directions get slowly removed from the subspace of l_t . We provide a generative model for x_t that satisfies these assumptions in the Appendix.

In earlier work [9], [10], [11], we studied a special case of the above problem, the case where both Ψ and Φ are identity matrices. In this work, we show how the Recursive Projected Compressed Sensing (ReProCS) and modified-ReProCS ideas introduced there can be adapted to the above more general problem. We demonstrate their usefulness via exhaustive simulations. To the best of our knowledge, except for the dense error correction work of [12] which can handle another kind

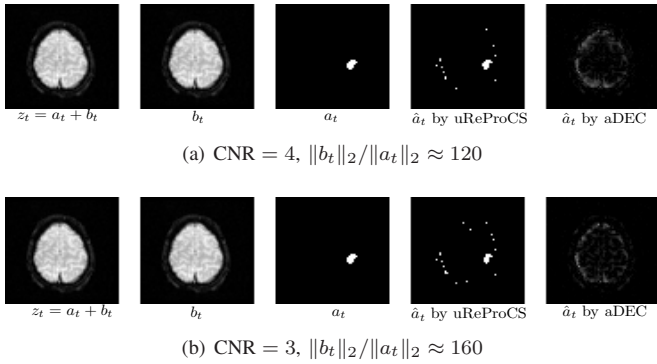


Fig. 1: Comparison of uReProCS and a-DEC for undersampled fMRI based brain active detection. Here, Ψ is a partial Fourier matrix with $m = 2048$ and $n = 4096$ and Φ is an identity matrix. We captured real fMRI brain image, b_t , when no stimulus was provided to the subject. We then add synthetic active pixels, a_t , at a low CNR of 4 and 3. Clearly, a_t is much smaller than that of b_t . Adapted-DEC solves (3) where W is a two level Daubechies-4 DWT matrix.

of large but “structured” noise (the noise is large but needs to be sparse), none of the other works in sparse recovery or recursive sparse recovery have studied the case of any other kind of large noise. However, this is a very important problem. The measurement noise could be large due to a variety of reasons. Also, there are many problems where only a part of the unknown signal, z_t , is the “signal of interest” while the rest is “background” noise, b_t , that we do not care about. The question is can we directly recover the “signal of interest” from highly undersampled measurements of z_t and can we do this even when the “signal of interest” is much smaller than the background, i.e. $\|s_t\|_2 \ll \|b_t\|_2$?

An important application where the latter problem occurs is in undersampled functional magnetic resonance imaging (fMRI) based brain activity detection [13], [14]. Here the goal is to detect which part of the brain is “active” in response to stimuli. Assume that all images are rewritten as 1D vectors. Let a_t be the sparse “active region” image (nonzero in the active region, zero elsewhere) and let b_t be the background brain image. Then the fMRI image of the brain is $z_t = a_t + b_t$. Undersampled fMRI means that we do not acquire all the Fourier coefficients, but only a randomly selected subset of them, i.e. we measure

$$y_t := \Psi z_t = \Psi a_t + \Psi b_t$$

where Ψ is the partial Fourier matrix. We do not care about recovering b_t , our goal is to only recover the active region image a_t . The background sequence, b_t , does not change much over time and hence can be well modeled as being “low rank”. This problem is then the same as (2) with $\Phi = I$ and $s_t = a_t$ or the same as (1) with $\Phi = I$, $s_t = a_t$ and $l_t = \Psi b_t$. As is well known, in MRI, the Fourier samples are acquired one sample at a time. Hence if we can accurately recover a_t from a smaller dimensional y_t then it allows the possibility of faster imaging which is needed in order to image fast-changing brain activity, e.g. image the brain activity in response to a fast-changing sequence of stimuli provided to the subject (this issue

is elaborated on in [14]).

A. Related Work

An important question that comes to mind is the following. In earlier work on undersampled MRI, it has often been assumed that the organ being imaged is wavelet sparse, e.g. see [15], [14]. Let W denote an 2D discrete wavelet transformation (DWT) matrix corresponding to a chosen wavelet and let W^{-1} be its inverse. So the question is, what happens if we let $b_t = W^{-1}x_t$ where x_t is the wavelet transform vector and then try to use the following adaptation of the dense error correction [12] idea (henceforth, we call this *adapted-DEC* or *aDEC*): solve

$$\min_{s,x} \|s\|_1 + \|x\|_1, \text{ subject to } y_t = \Psi s + \Psi W^{-1}x \quad (3)$$

The above is also related to the idea suggested in Pursuit of Justice [16]. We show in Fig. 1 that this does not work. Here we used the two level Daubechies-4 (db-4) wavelet since this is also the one that is used in other works, e.g. [14]. Also, we used $m = n/2$ partial Fourier measurements selected using the variable density undersampling scheme of [15]. Here the image is 64×64 , i.e. $n = 64^2 = 4096$. We think that the main reason for this is that the particular wavelet we chose may be a valid sparsity basis for b_t , but is not the best one. Hence one may require many more measurements to correctly recover both b_t and a_t . A second reason is that a_t may itself be compressible in the chosen wavelet basis. On the other hand, as can be seen, our proposed approach, undersampled ReProCS (details given in Sec. II) does much better. It has some extras, but at least it recovers most of the active pixels correctly.

To explain this simply, consider a toy example. Suppose b_t is exactly sparse with small support in a basis U , i.e. $b_t = Ux_t$ and the support of x_t , N_t is small, $|N_t| \ll n$. We do not know U . However, based on analyzing various images of the same type we believe that b_t is compressible in a given basis \tilde{U} which is close to U but not equal to it. Then if we look at a the $p\%$ energy support (e.g. 99.9%-energy support) of $\tilde{U}^{-1}b_t$ it will be much larger than the $p\%$ energy support of x_t . In fact as we show in Fig. 2 and 3, the 99.9% energy support of $\tilde{U}^{-1}b_t$ is also larger than the 100%-support of x_t , $|N_t|$. This happens even when \tilde{U} is a small rotation of U . On the other hand, undersampled ReProCS uses principal components’ analysis (PCA) to find the smallest dimensional subspace to approximate a given training sequence of b_t ’s and then keeps updating this PC basis on the fly (recursive PCA). For a given tolerance, PCA finds the smallest dimensional subspace to approximate a given data sequence. Thus, assuming that the estimates of b_t are accurate, PCA gives an estimate of the PC matrix, \hat{P}_t , such that the span of its columns, $\text{span}(\hat{P}_t)$, is close to the true subspace $\text{span}(\Psi(U)_{N_t})$. As a result the rank of \hat{P}_t , r , is also approximately equal to the support size of x_t , $|N_t|$.

A limitation of undersampled ReProCS and modified-ReProCS is that it requires a short noise-only training sequence (i.e. a sequence of b_t ’s) to obtain an initial PC matrix estimate. However, in most applications, this is fairly easy to get. For

example, if fMRI, one could begin by not providing any stimulus to the subject for a short initial period. In our experiments' section, we show extensive simulation experiments that compare undersampled ReProCS and undersampled modified-ReProCS with aDEC and modified-aDEC both when they use the true U and when they use a \tilde{U} which is a perturbation of U generated in various ways. Modified-aDEC modifies aDEC using the modified-CS idea [17]. We see from our experiments that even when \tilde{U} is a small rotation of U , the performance of even modified-aDEC is much worse than that of undersampled ReProCS.

In earlier attempts to solve the undersampled fMRI problem, Lu et al [14] have tried to first use modified-CS [17] or modified-CS-residual to recursively recover the entire brain image, $z_t := a_t + b_t$, by assuming that the whole thing is wavelet sparse. They then apply a t-test on the recovered sequence of z_t 's to detect the active region a_t as explained in [13]. The problem here is that a t-test is not able to work at lower contrast to noise ratios (CNR), i.e. when $\|a_t\|$ is much smaller than $\|b_t\|$. This is a key open problem in fMRI. The smallest CNR used in [14] is $\text{CNR} = 4$, where as the example we show in Fig. 1 uses $\text{CNR} = 3$. The experiments we show in this work are fully simulated because we wanted full control over the true sparsity basis of b_t . However, these indicate that we can potentially handle much smaller CNR's in the fMRI problem. This will be studied in later work.

Another possible way to solve our problem may be to use adaptations of various robust PCA or recursive robust PCA approaches, e.g. [18], [19], [20], [21], [22]. Robust PCA and recursive robust PCA would treat $\Psi\Phi s_t$ as the "outlier" while treating the sequence of l_t 's as the "data" for which one needs to estimate/update the PC space. For example, one could try to solve undersampled principal components' pursuit (PCP) [22] as follows. Compute $\min_{S,L} \|S\|_1 + \|L\|_*$ s.t. $[y_1, y_2, \dots, y_t] = \Psi\Phi S + L$ where $\|M\|_1$ and $\|M\|_*$ are the ℓ_1 norm and the nuclear norm respectively of a matrix M . But as we show in [11], even in the full sampled case, i.e. the case where $\Psi = I$ and $\Phi = I$, PCP does not work whenever the s_t 's are correlated temporally or spatially (in our application they are) and/or when its support is large. Other approaches, e.g. [18], [19], [20], [21], do not work because the magnitude of the sparse part is much smaller than that of the low rank part and also because of the large support size. Thus we do not expect these to work in the current case which is even more difficult due to the reduced number of measurements.

B. Notations

For any set $T \subset \{1, \dots, n\}$, T^c denotes its complement, i.e., $T^c := \{i \in [1, \dots, n] : i \notin T\}$, and $|T|$ denotes its cardinality, i.e., the number of elements in T . But $|a|$ where a is a real number denotes the magnitude of a .

For vector v , v_i denotes the i th entry of v and v_T denotes a vector consisting of the entries of v indexed by T . We use $\|v\|_p$ to denote the ℓ_p norm of v . The support of v , $\text{supp}(v)$, is the set of indices at which v has nonzero value, $\text{supp}(v) := \{i : v_i \neq 0\}$. We say that v is k -sparse if $|\text{supp}(v)| \leq k$. Sorting $|v_i|$ in descending order, we define the $p\%$ -energy set

of v as $T_p := \{|v_i| \geq \kappa\}$ where κ is the largest value of $|v_i|$ such that $\|v_{T_p}\|_2^2 \geq p\% \|v\|_2^2$, i.e., v_{T_p} contains the significantly nonzero elements of v .

For an $m \times \tau$ matrix A , A_i denote the i th column of A and A_T denote a matrix composed of the columns of A indexed by T . We use A' to denote its conjugate transpose, and A^\dagger to denote its pseudoinverse. If A is a tall matrix with $m > \tau$, $A^\dagger = (A'A)^{-1}A'$. We use $\text{span}(A)$ to denote the subspace spanned by the columns of A . Let $A \stackrel{\text{SVD}}{=} P\Lambda V'$ be a singular value decomposition (SVD) of A where P is an $m \times m$ unitary matrix, Λ is an $m \times \tau$ diagonal matrix with nonnegative real numbers on the diagonal, and V is an $\tau \times \tau$ unitary matrix. The column vectors of P are called as the left singular vectors of A . The α -truncated SVD that approximates A is $P_{T_\alpha}\Lambda_{T_\alpha}V'_{T_\alpha}$, which zeros out all but the singular values above a threshold α .

II. UNDERSAMPLED RECURSIVE PROJECTED CS (UREPROCS) AND MODIFIED-REPROCS

We first explain the undersampled ReProCS idea and then discuss undersampled modified-ReProCS which is needed when too few measurements are available for undersampled ReProCS. Finally we discuss the implicit assumptions for these methods to work.

A. Undersampled Recursive Projected CS (uReProCS)

Let P_t denote the PC of l_t 's and let \hat{P}_t be an estimate of P_t . To approximately nullify the low rank noise, l_t , we project the undersampled measurements, y_t , into the space perpendicular to \hat{P}_t , i.e. compute

$$\tilde{y}_t := (\hat{P}_{t,\perp})' y_t \quad (4)$$

where $\hat{P}_{t,\perp}$ is one orthogonal complement of \hat{P}_t . The dimension of the projected data vector, \tilde{y}_t , reduces to $m - r$ where $r := \text{rank}(\hat{P}_t)$. Notice that

$$\tilde{y}_t = A_t s_t + \beta_t, \text{ where } A_t = (\hat{P}_{t,\perp})' \Psi \Phi \text{ and } \beta_t := (\hat{P}_{t,\perp})' l_t \quad (5)$$

If $\hat{P}_t \approx P_t$, then $\beta_t \approx 0$. Finding the n -dimensional sparse vector, s_t , from the $m - r$ dimensional projected data vector, \tilde{y}_t , now becomes the traditional noisy sparse reconstruction / compressive sensing (CS) [1], [2], [3] problem with the "projected noise" β_t resulting from the error in estimating P_t . As long as $\|\beta_t\|_2$ is small and A_t does not nullify any nonzero elements of s_t , we can recover s_t by solving

$$\min_s \|s\|_1 \text{ subject to } \|\tilde{y}_t - A_t s\|_2 \leq \epsilon \quad (6)$$

with ϵ chosen proportional to $\|\beta_t\|_2$. In practice, a less computationally and memory intensive way is to solve

$$\min_s \|s\|_1 \text{ subject to } \|(I - \hat{P}_t \hat{P}_t')(\tilde{y}_t - \Psi \Phi s)\|_2 \leq \epsilon \quad (7)$$

This is exactly equivalent to (6) because $\hat{P}_{t,\perp}$ is an orthonormal complement of \hat{P}_t with $\hat{P}_t \hat{P}_t' + \hat{P}_{t,\perp} \hat{P}_{t,\perp}' = I$ and $\|\hat{P}_{t,\perp}' v\|_2 = \|\hat{P}_{t,\perp} \hat{P}_{t,\perp}' v\|_2$.

Let \hat{s}_t be the output of (6). As suggested in [23], \hat{s}_t is biased and we can do support estimation followed by least squares (LS) estimation on the support, i.e., we can compute

$$\hat{T}_t = \{i : (\hat{s}_t)_i \geq \gamma\} \quad (8)$$

$$(\hat{s}_t)_{\hat{T}_t} = ((A_t)_{\hat{T}_t})^\dagger \tilde{y}_t, (\hat{s}_t)_{\hat{T}_t^c} = 0 \quad (9)$$

Using \hat{s}_t , we can then estimate

$$\hat{l}_t = y_t - \Psi\Phi\hat{s}_t \quad (10)$$

which can be used to recursively update \hat{P}_t every-so-often to prevent the projected noise β_t from getting large (recursive PCA). We explain how to do this in Sec. III.

In (6) and (7), we set ϵ adaptively as $\epsilon = \|\hat{\beta}_{t-1}\|_2 = \|(\hat{P}_{t,\perp})'\hat{l}_{t-1}\|_2 = \|(I - \hat{P}_t\hat{P}_t')\hat{l}_{t-1}\|_2$.

We call the above approach undersampled recursive projected compressive sensing (uReProCS). The stepwise algorithm is given Algorithm 1.

B. Modified-uReProCS

The number of measurements available for uReProCS in (5) is $m - r$, where $r = \text{rank}(\hat{P}_t)$. Let T_t denote the support of s_t . For a given $r = \text{rank}(\hat{P}_t)$, if the support size of s_t , $|T_t|$, increases, or if r increases for a given sparsity level, $|T_t|$, the ratio $\frac{m-r}{|T_t|}$ may become too small for (6) to recover s_t accurately. However, if the support of s_t , T_t , changes very slowly, and if the previous support estimate, \hat{T}_{t-1} , is accurate, $T \leftarrow \hat{T}_{t-1}$ is a reliable approximation if T_t . Therefore, we can use modCS [17] to recover s_t , i.e., solve

$$\min_s \|s_{T^c}\|_1 \quad \text{subject to } \|\tilde{y}_t - A_t s\|_2 \leq \epsilon \quad (11)$$

In practice, a less computationally and memory intensive way is to solve

$$\min_s \|s_{T^c}\|_1 \quad \text{subject to } \|(I - \hat{P}_t\hat{P}_t')(\tilde{y}_t - \Psi\Phi s)\|_2 \leq \epsilon \quad (12)$$

Let \hat{s} denote the output of modCS. As explained in [24], a single support threshold for support estimation as in (8) does not work well for modCS. A better approach is to use the Add-LS-Del procedure summarized in (13)-(16), which was first introduced in our older work [25], [26] and simultaneously also in [27], [4].

$$T_{\text{add}} = T \cup \{i \in T^c : |(\hat{s})_i| > \alpha_{\text{add}}\} \quad (13)$$

$$(\hat{s})_{T_{\text{add}}} = ((A_t)_{T_{\text{add}}})^\dagger \tilde{y}_t, (\hat{s})_{T_{\text{add}}^c} = 0 \quad (14)$$

$$\hat{T}_t = T_{\text{add}} \setminus \{i \in T_{\text{add}} : |(\hat{s})_i| < \alpha_{\text{del}}\} \quad (15)$$

$$(\hat{s}_t)_{\hat{T}_t} = ((A_t)_{\hat{T}_t})^\dagger \tilde{y}_t, (\hat{s}_t)_{\hat{T}_t^c} = 0 \quad (16)$$

The addition step threshold, α_{add} , needs to be just large enough to ensure that the matrix used for LS estimation, $A_{T_{\text{add}}}$ is well-conditioned. If α_{add} is chosen properly, the LS estimate on T_{add} will have smaller error than the modCS output. As a result, deletion will be more accurate when done using this estimate. This also means that one can use a larger α_{del} to ensure quicker deletion of extras.

Replacing (6) and (8)-(9) of uReProCS by (11) and (13)-(16), we call it undersampled modified-ReProCS(mod-uReProCS). The stepwise algorithm is given Algorithm 1.

Algorithm 1 uReProCS and mod-uReProCS

At $t = t_0$, suppose a good estimate of PC matrix, \hat{P}_{t_0} is available from training data. For $t > t_0$, do the following:

- 1) Obtain \tilde{y}_t by (4).
 - 2a) For uReProCS,
 - Estimate s_t by solving (7) with $\epsilon = \|(I - \hat{P}_t\hat{P}_t')\hat{l}_{t-1}\|_2$.
 - Support thresholding and least square estimation: do (8) and (9).
 - 2b) For mod-uReProCS,
 - Estimate s_t by solving (12) with $T \leftarrow \hat{T}_{t-1}$ and $\epsilon = \|(I - \hat{P}_t\hat{P}_t')\hat{l}_{t-1}\|_2$.
 - Add-LS-Del procedure: do (13)-(16).
 - 3) Estimate $\hat{l}_t = y_t - \hat{s}_t$.
 - 4) Recursive PCA: Update \hat{P}_t using Algorithm 2 which is based on [21].
 - 5) Increment t by 1 and go to step 1).
-

C. Implicit Requirements**

Suppose that $|T_t| \leq k$ for all t , i.e. all s_t 's are k -sparse. Clearly, a necessary condition for the CS step of uReProCS, i.e. (6), to recover the support of all s_t 's correctly is that $A_t = (\hat{P}_{t,\perp})'\Psi\Phi$ does not nullify any k -sparse vector, and thus does not nullify any nonzero part of any s_t . In the case that the rows of Ψ are orthonormal (i.e. $\Psi\Psi' = I$), this is equivalent to requiring that no k -sparse vector belongs to $\text{span}(\Phi'\Psi'\hat{P}_t)$. Notice that many frequently used measurement matrices satisfy $\Psi\Psi' = I$ either always (e.g. partial Fourier matrix) or with high probability (e.g. random Gaussian matrix). We can use the null space property (NSP) [28] to show that a slightly stronger version of this requirement also serves as a sufficient condition, at least in the noise-free case.

Consider (6). Assume that $\hat{P}_t = P_t$ so that the projected noise is indeed zero, i.e. $\beta_t = 0$. In this case, (6) with $\epsilon = 0$ will exactly recover any k -sparse s_t if the following holds with a $\theta < 1$:

$$\|(\eta)_T\|_1 \leq \theta \|(\eta)_{T^c}\|_1$$

for all sets T with $|T| \leq k$ and for all $\eta \in \text{null}(A_t)$ [28], [7]. Here $\text{null}(A_t) := \{\eta : A_t\eta = 0\}$ refers to the null space of A_t . In words, we need that all k -sparse or k -approximately-sparse vectors (i.e. vectors for which $\|(\eta)_{T^c}\|_1 \leq \|(\eta)_T\|_1$ for some set T with $|T| \leq k$) do not lie in $\text{null}(A_t)$. But $\text{null}(A_t) = \text{span}(\Phi'\Psi'\hat{P}_t)$ if the rows of Ψ are orthonormal. Thus, a sufficient condition for ReProCS to exactly recover s_t in the noise-free case ($\beta_t = 0$ and $\epsilon = 0$ in (6)) is that no k -sparse or k -approximately-sparse vector lies in $\text{span}(\Phi'\Psi'\hat{P}_t)$.

Remark 1: We should note that we can also get a sufficient condition using the restricted isometry property (RIP) [8], [29] and in fact it would hold even in the noisy case, but it is not as illustrative. Let δ_k be the k -RIP constant [8] for the matrix A_t . If the projected noise β_t satisfies $\|\beta_t\|_2 \leq \epsilon$ and if $\delta_{2k} < \sqrt{2} - 1$, the reconstruction error is bounded by a constant times ϵ , i.e. $\|\hat{s}_t - s_t\|_2 \leq C(\delta_{2k})\epsilon$ [29].

Algorithm 2 Recursive PCA

Initialization: Compute $[l_1, \dots, l_{t_0}] \stackrel{SVD}{=} P\Lambda V'$; set $T \leftarrow \{i : (\Lambda)_{i,i} > \alpha_0\}$, $\hat{P}_{t_0} \leftarrow P_T$, $\hat{\Lambda}_{t_0} \leftarrow \Lambda_{T,T}$. Let $D = \emptyset$. At each time $t > t_0$, store \hat{l}_t in D , i.e., $D \leftarrow [D \hat{l}_t]$. Do the following:

- 1) If there are less than τ frames in D ,
 - Let $\hat{P}_t \leftarrow \hat{P}_{t-1}$, $\hat{\Lambda}_t \leftarrow \hat{\Lambda}_{t-1}$.

- 2) If there are τ frames in D

- 2a) Remove decayed directions from \hat{P}_{t-1}

$$\begin{aligned} \hat{\Sigma}_\varepsilon &= \frac{1}{\tau} \hat{P}'_{t-1} D D' \hat{P}_{t-1} \\ \hat{N}_\varepsilon &= \{i : (\hat{\Sigma}_\varepsilon)_{i,i} \geq \alpha\} \\ \hat{P}_{t-1} &\leftarrow (\hat{P}_{t-1})_{\hat{N}_\varepsilon}, \quad \hat{\Lambda}_{t-1} \leftarrow (\hat{\Lambda}_{t-1})_{\hat{N}_\varepsilon, \hat{N}_\varepsilon} \\ \hat{r}_{t-1} &\leftarrow \text{rank}(\hat{P}_{t-1}) \end{aligned}$$

- 2b) Update \hat{P}_t and $\hat{\Lambda}_t$ by incremental SVD [21]

- Compute $C = \hat{P}'_{t-1} D$ and $E = D - \hat{P}_{t-1} C$
- Compute QR decomposition of E : $E \stackrel{QR}{=} JK$
- Compute SVD of $\begin{bmatrix} \hat{\Lambda}_{t-1} C \\ 0 \quad K \end{bmatrix} : \begin{bmatrix} \hat{\Lambda}_{t-1} C \\ 0 \quad K \end{bmatrix} \stackrel{SVD}{=} P_r \Lambda_r V_r'$
- Update \hat{P}_t and $\hat{\Lambda}_t$ as

$$\hat{P}_t \leftarrow [\hat{P}_{t-1} \quad J] P_r, \quad \hat{\Lambda}_t \leftarrow \Lambda_r$$

- 2c) Add new PCs with large variance

$$\begin{aligned} \hat{N}_t &= \{1, \dots, \hat{r}_{t-1}\} \cup \{i > \hat{r}_{t-1} : \frac{(\hat{\Lambda}_t)_{i,i}^2}{\tau} \geq \alpha\} \\ \hat{P}_t &\leftarrow (\hat{P}_t)_{\hat{N}_t}, \quad \hat{\Lambda}_t \leftarrow (\hat{\Lambda}_t)_{\hat{N}_t} \end{aligned}$$

- 2e) Reset $D = \emptyset$.
-

III. RECURSIVE PCA

In the recursive PCA step of uReProCS, we update \hat{P}_t for every τ frames by Algorithm 2, which is complete different from [9, Algorithm 1]. Alternatively, we can also do this whenever $\|(\hat{P}_{t-1, \perp})' \hat{l}_{t-1}\|_2$ exceeds a threshold.

The initial PC matrix, \hat{P}_{t_0} , is estimated by computing an α_0 -truncated SVD approximation of the noise-only training data, $[l_1, \dots, l_{t_0}]$, i.e., compute $[l_1, \dots, l_{t_0}] \stackrel{SVD}{=} P\Lambda V'$; set $T \leftarrow \{i : (\Lambda)_{i,i} > \alpha_0\}$, $\hat{P}_{t_0} \leftarrow P_T$, $\hat{\Lambda}_{t_0} \leftarrow \Lambda_{T,T}$, and $\hat{V}_{t_0} \leftarrow V_{T,T}$. Usually, α_0 is picked empirically according to the distribution of the singular values. If $[l_1, \dots, l_{t_0}]$ is exactly low rank, α_0 can be zero. We use \hat{V}_t to only explain how we update \hat{P}_t , but we do not need to compute and store \hat{V}_t .

The PC update is done as follows. At $t = t_0 + k\tau$, let $D = [\hat{l}_{t-\tau+1}, \dots, \hat{l}_t]$. In step 2a), we first estimate the variance of D along the columns of \hat{P}_{t-1} and remove the PCs along which the variance is below a threshold α . Let σ_{\min}^2 be the smallest nonzero diagonal elements of $\hat{\Lambda}_{t_0}^2/t_0$. We use $\alpha = 0.5\sigma_{\min}^2$. Note that after doing step 2a), the column vectors of \hat{P}_{t-1} contain all the non-decayed PCs.

In step 2b), we rotate \hat{P}_{t-1} and find the new PCs based on the idea of incremental SVD [21]. We first decompose the

new collected data D into two components C and E , where $C = \hat{P}'_{t-1} D$ and $E = D - \hat{P}_{t-1} C$. The parallel component C rotates the existing singular vectors and the orthogonal component E estimates the new PCs [21]. Let $E \stackrel{QR}{=} JK$ be a QR decomposition of E . Notice that \hat{P}_{t-1} and J are orthogonal, i.e., $\hat{P}'_{t-1} J = 0$. The column vectors of J are the basis vectors of the subspace spanned by the new PCs. It is easy to see that [21]

$$[\hat{P}_{t-1} \hat{\Lambda}_{t-1} \hat{V}'_{t-1} \quad D] = [\hat{P}_{t-1} \quad J] \begin{bmatrix} \hat{\Lambda}_{t-1} C \\ 0 \quad K \end{bmatrix} \begin{bmatrix} \hat{V}_{t-1} 0 \\ 0 \quad I \end{bmatrix}'$$

Let $\begin{bmatrix} \hat{\Lambda}_{t-1} C \\ 0 \quad K \end{bmatrix} \stackrel{SVD}{=} P_r \Lambda_r V_r'$. Clearly,

$$[\hat{P}_{t-1} \hat{\Lambda}_{t-1} \hat{V}'_{t-1} \quad D] \stackrel{SVD}{=} ([\hat{P}_{t-1} \quad J] P_r) \Lambda_r \left(\begin{bmatrix} \hat{V}_{t-1} 0 \\ 0 \quad I \end{bmatrix} V_r \right)'$$

In words, P_r rotates the old PCs, \hat{P}_{t-1} , and the new basis vectors, J , to the current PCs, i.e., $\hat{P}_t = [\hat{P}_{t-1} \quad J] P_r$. Also, the singular values along \hat{P}_t are the diagonal elements of $\hat{\Lambda}_t = \Lambda_r$.

Let $\hat{r}_{t-1} = \text{rank}(\hat{P}_{t-1})$ and let $\hat{N}_A = \{\hat{r}_{t-1} + 1, \dots, \text{rank}(\hat{P}_t)\}$. If $\hat{P}_{t-1} \approx P_{t-1}$, the old PCs are already correctly estimated and do not need to be rotated. Under this assumption, $(\hat{P}_t)_{\hat{N}_A}$ contains the new PCs. The variance of D along the columns of $(\hat{P}_t)_{\hat{N}_A}$ is given by the diagonal elements of $\frac{1}{\tau} (\hat{\Lambda}_t)_{\hat{N}_A}^2$. Therefore, to only retain the new PCs with large variance, we threshold on $\frac{1}{\tau} (\hat{\Lambda}_t)_{\hat{N}_A}^2$ in step 2c).

IV. SIMULATION EXPERIMENTS

As explained earlier, in the scenario where the signal energy is much smaller than the noise energy, none of the existing sparse recovery or recursive sparse recovery approaches have any chance of working. In fact even if we have full samples, e.g. if $\Psi = I$, it is impossible to recover s_t from $y_t = s_t + l_t$ via thresholding because the magnitude of any nonzero element of s_t is small compared with that of l_t . We have shown this in [11, Fig 5]. A possible approach to compare with is adapted dense error correction (aDEC) that uses, \tilde{U} as its ‘‘knowledge’’ of U and solves

$$\min_{s,x} \|s\|_1 + \|x\|_1, \quad \text{subject to } y_t = \Psi s + \tilde{\Psi} \tilde{U} x \quad (17)$$

We experiment with two types of situations – (a) $\tilde{U} = U$, in this case U is perfectly known; and (b) \tilde{U} is a perturbation of U generated in various different ways.

We also compare with modified-aDEC (mod-aDEC) which solves

$$\min_{s,x} \|s_{T^c}\|_1 + \|x_{N^c}\|_1, \quad \text{subject to } y_t = \Psi s + \tilde{\Psi} \tilde{U} x \quad (18)$$

where T and N are estimates of T_t and N_t , respectively.

Undersampled ReProCS(uReProCS) and undersampled modified ReProCS(mod-uReProCS) get the initial PC matrix, \hat{P}_{t_0} , by computing a truncated SVD of a noise-only training sequence as explained in Algorithm 2, i.e. of $[l_1, \dots, l_{t_0}]$ for $t_0 = 10^4$. Mod-aDEC uses T_{t_0} and N_{t_0} at $t = t_0 + 1$. Mod-uReProCS used T_{t_0} at $t = t_0 + 1$.

Our data is generated as follows. The sparse signal, s_t , has support set T_t . All its nonzero values are equal to 10, i.e., $(s_t)_i = 10, \forall i \in T_t$ and $(s_t)_i = 0, \forall i \in T_t^c$. We generate T_t as follows. At each time t , there are several square blocks in s_t (when we write s_t as an 2D matrix). Each block can either stay static with probability 0.8 or move independently one step towards top/bottom/left/right with probability 0.05. Hence, T_t is spatially and temporal correlated and it changes slowly over time.

The sparse vector, x_t , (used to generate $b_t = Ux_t$) is simulated using the regression model described in the Appendix with $f = 0.5, f_d = 0.1$, and $\theta = 0.5$. Initially there are 32 indices in N_t along which x_t has variances $10^4, 0.8058 \times 10^4, 0.8058^2 \times 10^4, \dots, 12.4$. At time $t = 5$, two new indices are added to N_t , along which x_t starts with a small nonzero value and their variance slowly stabilizes to 50 and 60. The value of x_t along two old indices begins to decay exponentially to zero. At time $t = 50$, another two new indices are added in N_t along which x_t starts with a small nonzero value and its variance slowly stabilizes to variance 55 and 65. The value of x_t along two old indices begins to decay exponentially to zero.

In all experiments, the background energy $\|b_t\|_2$, is much larger than that of the sparse signal $\|s_t\|_2$, for example for the case of Fig 3(a), at $t = t_0 + 1, \|b_t\|_2/\|s_t\|_2 = 5$.

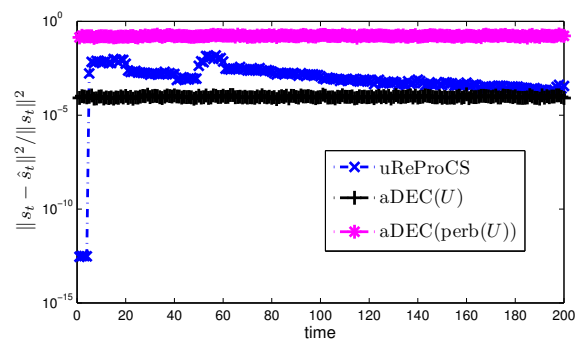
In Sec. 2, U is an orthonormalized random Gaussian matrix (first generate a random Gaussian matrix and then orthonormalize it using the Gram-Schmidt process) and Ψ is a random Gaussian matrix. In Sec. 3, U is an inverse discrete wavelet transformation (DWT) matrix corresponding to a four-level db-4 wavelet and Ψ is the partial Fourier matrix generated by randomly selecting frequencies.

In all experiments, we show the normalized mean square error (NMSE) defined as $\frac{\mathbb{E}\|s_t - \hat{s}_t\|_2^2}{\mathbb{E}\|s_t\|_2^2}$. The expected value is computed using 50 times Monte Carlo averaging.

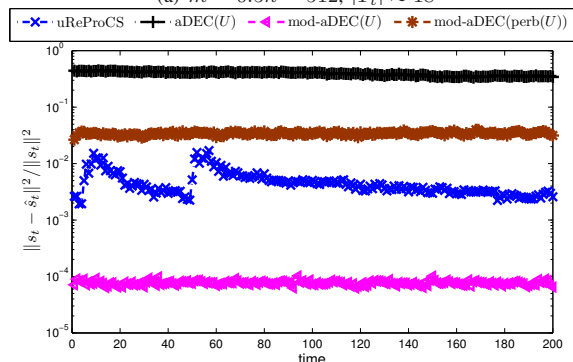
A. Ψ is a random Gaussian matrix

In this experiment, Ψ is an $m \times n$ ($n = 1024$) random Gaussian matrix with entries sampled from $\mathcal{N}(0, 1/n)$. The sparsity basis of l_t, U , is an orthormalized random Gaussian matrix (generated by first generating an $n \times n$ matrix with entries randomly sampled from $\mathcal{N}(0, 1)$, and then orthonormalizing it using the Gram-Schmidt process). We get $\tilde{U} = \text{perb}(U)$ by first adding a random Gaussian matrix with entries sampled from $(0, 0.01/n)$ to U , and then orthonormalizing it using the Gram-Schmidt process. Hence, $\text{perb}(U)$ is a small rotation of U , which would give a compressible representation of l_t , i.e., $\text{perb}(U)^{-1}l_t$ has a few significantly large coefficients while rest of them are small. For example, the 99%-energy set of $\text{perb}(U)^{-1}b_t$ is about 34 and the 99.9%-energy set of $\text{perb}(U)^{-1}b_t$ is about 931. The perturbation we add to U has small variance $0.01/n$. If the variance of the perturbation increases, the performance of aDEC and mod-aDEC using $\tilde{U} = \text{perb}(U)$ degrades dramatically.

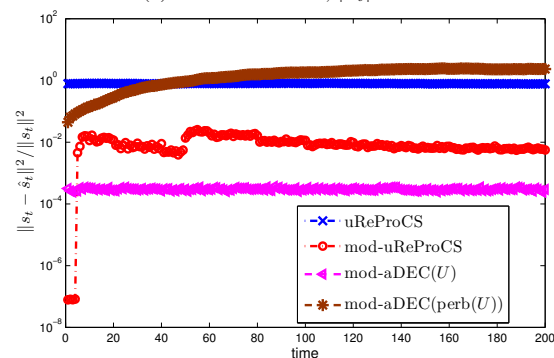
In Fig. 2(a), $m = 512$ and $|T_t| \approx 18$, i.e., there are two 3×3 nonzero blocks in s_t . As can be seen, uReProCS recursively recover s_t with small reconstruction error. Adapted DEC(U)



(a) $m = 0.5n = 512, |T_t| \approx 18$



(b) $m = 0.5n = 512, |T_t| \approx 147$



(c) $m = 341 \approx n/3, |T_t| \approx 147$

Fig. 2: Ψ and U are random Gaussian matrices.

gives the smallest reconstruction error because it use the perfect knowledge about the sparsity basis U , while uReProCS does not.

In Fig. 2(b), $m = 512$ and the support set of s_t increases to $|T_t| \approx 147$, i.e., there are three 7×7 nonzero blocks in s_t . Undersampled ReProCS can still recover s_t . Adapted DEC(U) cannot recover s_t accurately, even though it uses the perfect sparsity basis U . Adapted DEC($\text{perb}(U)$) gives worse result than aDEC(U) (not shown). By utilizing the prior knowledge on the support of s_t and x_t and perfect knowledge of U , mod-aDEC(U) gives the smallest reconstruction error. However the error of mod-aDEC($\text{perb}(U)$) is larger than uReProCS. Mod-uReProCS gave similar result as uReProCS (not shown).

In Fig. 2(c), $m = 341$ and $|T_t| \approx 147$. The ratio, $\frac{m-r}{|T_t|} \approx 2.1$, becomes too small for uReProCS to work. However, mod-uReProCS still gives small reconstruction error since it utilizes the knowledge of the support of s_t . Again, aDEC(U) does

not work (not shown) and mod-aDEC(U) gives the smallest reconstruction error. The error of mod-aDEC(perb(U)) blows up due to the support error of T and N in (18).

B. Ψ is a partial Fourier matrix

We now do experiments to simulate the undersampled fMRI based brain active region detection problem as shown in Fig. 1. Here, Ψ is an $m \times n$ 2D partial Fourier matrix with $m = 512$ and $n = 1024$. We let $U = W_{\text{db4}}^{-1}$, where W_{db4} is a four-level Daubechies-4 2D DWT matrix. The background, b_t , has the sparsest representation in the space specified by W_{db4} with a support size about $|N_t| \approx 32$. However, U may be unknown in practice. Let W_{db3} denote a four-level Daubechies-3 2D DWT matrix which is used as \tilde{U} for aDEC. In this case, the representation of b_t by W_{db3} , $W_{\text{db3}}^{-1}b_t$, has a much larger support set. For example, the 99.9%-energy set of $W_{\text{db3}}^{-1}b_t$ is about 400, which is much larger than 32. We also compare with \tilde{U} being a perturbation of $U = W_{\text{db4}}^{-1}$ generated as in the previous subsection, i.e. perb(W_{db4}^{-1}) is generated by adding i.i.d. Gaussian noise of variance $0.01/n$ to U and then orthonormalizing the resulting matrix.

In Fig. 3(a), $|T_t| \approx 18$. As can be seen, aDEC with perfect sparsity basis knowledge, aDEC(W_{db4}^{-1}), also works only as well as ReProCS which does not use this knowledge. modDEC(W_{db4}^{-1}) gives the smallest reconstruction error because it use the support knowledge on s_t and x_t and uses perfect sparsity basis knowledge. The errors of aDEC(perb(W_{db4}^{-1})) and aDEC(W_{db3}^{-1}) are much larger for the reasons explained earlier.

In Fig. 3(b), $m = 512$ and $|T_t| \approx 147$. Undersampled ReProCS gives large reconstruction error because the ratio, $\frac{m-r}{|T_t|} \approx 2.1$, is too small for uReProCS to work. Mod-uReProCS improves the performance of uReProCS greatly because it use the support knowledge of s_t . Note that aDEC(W_{db4}^{-1}) and aDEC(W_{db3}^{-1}) (not shown) fail completely. The error of mod-aDEC(W_{db4}^{-1}) and mod-aDEC(W_{db3}^{-1}) blow up because of the support errors in T and N .

V. CONCLUSIONS

In this work we studied the problem of recursively recovering a time sequence of sparse signals, with time-varying sparsity patterns, from highly undersampled measurements corrupted by very large but correlated noise. It is assumed that the noise is correlated enough to have an approximately low rank covariance matrix that is either constant, or changes slowly, with time. We showed how our recently introduced Recursive Projected CS (ReProCS) and modified-ReProCS ideas can be adapted to solve this problem very effectively. The only other sparse recovery work that studies another kind of large but “structured” noise is the dense error correction (DEC) work [12] (it requires sparse noise).

We showed exhaustive simulation comparisons with DEC adapted to our problem (aDEC), i.e. aDEC that solves (17) and with modified-aDEC which solves (18). Both aDEC and modified-aDEC need to know the sparsity basis for the background noise. We simulated the perfect knowledge case, i.e. U is perfectly known, and the imperfect knowledge case

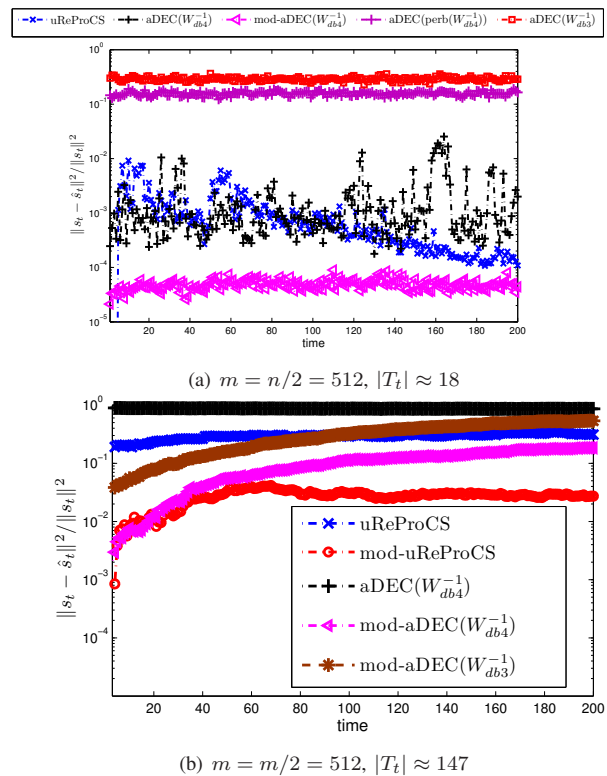


Fig. 3: Ψ is a partial Fourier matrix and $U = W_{\text{db4}}^{-1}$.

where $\tilde{U} \neq U$ is our knowledge of the sparsity basis and \tilde{U} is a small rotation of U . Modified-aDEC is provided the support of both x_t and s_t at the initial time. On the other hand, uReProCS and modified-uReProCS *do not know* U or \tilde{U} . However, both have initial background-only training data available, so it is fair to say that both also have initial support knowledge of x_t . Modified-uReProCS is also provided initial support knowledge of s_t . Thus, modified-aDEC(\tilde{U}) and modified-uReProCS have similar amount of prior knowledge available to them, whereas modified-aDEC(\tilde{U}) has more prior knowledge than uReProCS. As can be seen from our comparisons, modified-aDEC(\tilde{U}) usually is much worse than both uReProCS and modified-uReProCS.

APPENDIX: MODEL ON THE SUPPORT CHANGE OF x_t

We provide here a realistic generative model for x_t and hence the low rank part, $l_t = \Psi U x_t$. The support set of x_t , N_t , is a union of three disjoint sets \mathcal{A}_t , \mathcal{D}_t , and \mathcal{E}_t , i.e., $N_t = \mathcal{A}_t \cup \mathcal{D}_t \cup \mathcal{E}_t$. At $t = 1$, we let $N_t = \mathcal{E}_t$ and $\mathcal{A}_t \cup \mathcal{D}_t = \emptyset$. For $t > 1$, the new addition set $\mathcal{A}_t := N_t \setminus N_{t-1}$ is the set of indices along which $(x_t)_{\mathcal{A}_t}$ starts to be nonzero. \mathcal{A}_t is nonempty once every d frames. The set $\mathcal{D}_t \subset (N_t \cap N_{t-1})$ is the set of indices along which $(x_t)_{\mathcal{D}_t}$ decay exponentially to zero. We assume that \mathcal{D}_t will not get added to N_t at any future time. The set $\mathcal{E}_t := N_t \cap N_{t-1} \setminus \mathcal{D}_t$ is the set of indices along with $(x_t)_{\mathcal{E}_t}$ follows a first order autoregressive (AR-1) model.

Let Σ be a diagonal matrix with nonnegative diagonal elements σ_i^2 s. We model x_t as

$$x_0 = 0, \quad x_t = F_t x_{t-1} + \nu_t, \quad \nu_t \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, Q_t)$$

where ν_t is independent and identically distributed Gaussian noise with zero mean and diagonal covariance matrix Q_t ; F_t and Q_t are two diagonal matrices defined as below

$$F_t = \begin{bmatrix} 0_{\mathcal{A}_t} & 0 & 0 & 0 \\ 0 & (fI)_{\mathcal{E}_t} & 0 & 0 \\ 0 & 0 & (f_d I)_{\mathcal{D}_t} & 0 \\ 0 & 0 & 0 & (0)_{N_t^c} \end{bmatrix}$$

$$Q_t = \begin{bmatrix} \theta(\Sigma)_{\mathcal{A}_t} & 0 & 0 & 0 \\ 0 & (1-f^2)(\Sigma)_{\mathcal{E}_t} & 0 & 0 \\ 0 & 0 & (0)_{\mathcal{D}_t} & 0 \\ 0 & 0 & 0 & (0)_{N_t^c} \end{bmatrix}$$

The three scalars f , f_d , and θ satisfy $0 < f_d < f < 1$ and $0 < \theta < 1$.

From the model on x_t given in (V), we notice the following:

a) At time $t = jd$, $(x_t)_{\mathcal{A}_t}$ starts with

$$(x_t)_{\mathcal{A}_t} \sim \mathcal{N}(0, \theta(\Sigma)_{\mathcal{A}_t}).$$

Small θ ensures that new directions get added at a small value and increase slowly. $(x_t)_{\mathcal{D}_t}$ decays as

$$(x_t)_{\mathcal{D}_t} = f_d(x_t)_{\mathcal{D}_t}$$

$(x_t)_{\mathcal{E}_t}$ follows an AR-1 model with parameter f :

$$(x_t)_{\mathcal{E}_t} = f(x_{t-1})_{\mathcal{E}_t} + (v_t)_{\mathcal{E}_t}$$

b) At time $t > jd$, the variance of $(x_t)_{\mathcal{A}_{jd}}$ gradually increases as

$$(x_t)_i \sim \mathcal{N}(0, (1 - (1 - \theta)f^{2(t-jd)})_{\Sigma_{i,i}}), \quad i \in \mathcal{A}_{jd}$$

Eventually, the variance of $(x_t)_{\mathcal{A}_{jd}}$ converges to $(\Sigma)_{\mathcal{A}_{jd}}$. We assume this becomes approximately true much before $t = (j+1)d$ (the next support change time).

c) At time $t > jd$, the variance of $(x_t)_{\mathcal{D}_{jd}}$ decays exponentially to zero as

$$(x_t)_{\mathcal{D}_{jd}} \sim \mathcal{N}(0, f_d^{2(t-jd)}(\Sigma)_{\mathcal{D}_{jd}})$$

We assume that this has approximately decayed to zero much before $t = (j+1)d$.

For every d frames, at $t = jd$, the values of x_t along the new indices, \mathcal{A}_t , become nonzero and the values of x_t along existing indices, \mathcal{D}_t , start to decay exponentially. The values of x_t along all other existing indices follow an independent AR-1 model. After a short period Δ_d which is much less than d , the variances of x_t along the new indices increase to some stable values and the values of x_t along the decaying indices decayed to zero. Therefore, N_t is piecewise constant from $t = jd + \Delta_d$ to $t = j(d + d)$.

REFERENCES

- [1] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, "Atomic decomposition by basis pursuit", *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.
- [2] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information", *IEEE Trans. Info. Th.*, vol. 52(2), pp. 489–509, February 2006.
- [3] D. Donoho, "Compressed sensing", *IEEE Trans. on Information Theory*, vol. 52(4), pp. 1289–1306, April 2006.
- [4] D. Needell and J.A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples", *Appl. Comp. Harm. Anal.*, vol. 26, pp. 301–321, 2008.
- [5] D. L. Donoho, Y. Tsaig, I. Drori, and J-L. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit", *IEEE Trans. on Information Theory*, (Submitted) 2007.
- [6] Richard G. Baraniuk, Volkan Cevher, Marco F. Duarte, and Chinmay Hegde, "Model-based compressive sensing", *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.
- [7] Y. Wang and W. Yin., "Sparse signal reconstruction via iterative support detection", *SIAM Journal on Imaging Sciences*, pp. 462–491, 2010.
- [8] E. Candes and T. Tao, "Decoding by linear programming", *IEEE Trans. Info. Th.*, vol. 51(12), pp. 4203 – 4215, Dec. 2005.
- [9] C. Qiu and N. Vaswani, "Real-time robust principal components' pursuit", in *Allerton Conference on Communication, Control, and Computing*, 2010.
- [10] C. Qiu and N. Vaswani, "Support-predicted modified-cs for principal components' pursuit", in *IEEE Intl. Symp. on Information Theory (ISIT)*, 2011.
- [11] Chenlu Qiu and Namrata Vaswani, "Reprocs: A missing link between recursive robust pca and recursive sparse recovery in large but correlated noise", *Arxiv*.
- [12] John Wright and Yi Ma, "Dense error correction via l1-minimization", *IEEE Trans. on Info. Th.*, vol. 56, no. 7, pp. 3540–3560, 2010.
- [13] I. C. Atkinson, D. L. J. Farzad Kamalabadi, and Thulborn, "BLIND estimation for localized low contrast-to-noise ratio bold signals", *IEEE Journal of Selected Topics in Signal Processing*, pp. 350–364, 2008.
- [14] Wei Lu, Taoran Li, Ian Atkinson, and Namrata Vaswani, "Modified-cs-residual for recursive reconstruction of highly undersampled functional mri sequences", in *IEEE Intl. Conf. Image Proc. (ICIP)*, 2011.
- [15] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse mri: The application of compressed sensing for rapid mr imaging", *Magnetic Resonance in Medicine*, vol. 58(6), pp. 1182–1195, Dec. 2007.
- [16] J.N. Laska, M.A. Davenport, and R.G. Baraniuk, "Exact signal recovery from sparsely corrupted measurements through the pursuit of justice", in *Asilomar Conf. on Sig. Sys. Comp.*, Nov 2009, pp. 1556 –1560.
- [17] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support", *IEEE Trans. Signal Processing*, Sep 2010.
- [18] D. Skocaj and A. Leonardis, "Weighted and robust incremental method for subspace learning", in *IEEE Intl. Conf. on Computer Vision (ICCV)*, Oct 2003, pp. 1494 –1501.
- [19] Fernando De La Torre and Michael J. Black, "A framework for robust subspace learning", *International Journal of Computer Vision*, vol. 54, pp. 117–142, 2003.
- [20] Yongmin Li, Li qun Xu, Jason Morphet, and Richard Jacobs, "An integrated algorithm of incremental and robust pca", in *IEEE Intl. Conf. Image Proc. (ICIP)*, 2003, pp. 245–248.
- [21] M. Brand, "Incremental singular value decomposition of uncertain data with missing values", in *European Conference on Computer Vision*, 2002, pp. 707–720.
- [22] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?", *submitted to Journal of the ACM, December 2009*.
- [23] E. Candes and T. Tao, "The dantzig selector: statistical estimation when p is much larger than n", *Annals of Statistics*, 2006.
- [24] N. Vaswani, "Stability (over time) of modified-cs and ls-cs for recursive causal sparse reconstruction", *Allerton Conference on Communication, Control, and Computing*, 2010.
- [25] N. Vaswani, "Ls-cs-residual (ls-cs): Compressive sensing on least squares residual", *IEEE Trans. Sig. Proc.*, vol. 58(8), pp. 4108–4120, 2010.
- [26] N. Vaswani, "Kalman filtered compressed sensing", in *IEEE Intl. Conf. Image Proc. (ICIP)*, 2008.
- [27] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction", *IEEE Trans. Info. Th.*, vol. 55, no. 5, pp. 2230–2249, May 2009.
- [28] I. Daubechies, R. DeVore, M. Fornasier, and S. Gunturk, "Iteratively re-weighted least squares minimization: Proof of faster than linear rate for sparse recovery", in *Proceedings of the 42nd IEEE Annual Conference on Information Sciences and Systems (CISS 2008)*, March 2008.
- [29] E. Candes, "The restricted isometry property and its implications for compressed sensing", *Compte Rendus de l'Academie des Sciences, Paris, Serie I*, pp. 589–592, 2008.