Subspace Tracking From Few Entries

Praneeth Narayanamurthy and Namrata Vaswani
{pkurpadn, namrata}@iastate.edu
Department of Electrical and Computer Engineering,
Iowa State University, Ames, IA

***Abstract*—In this work we study the problem of subspace tracking from few observed entries. In particular we assume that we are given a data stream, which originally lies in a low dimensional subspace of the ambient dimension. However, due to errors in transmission, we only observe the data at certain location at each given time. We show through theory and extensive empirical evaluation that we are able to still recover the underlying subspace and show that the delay to estimate the changed subspace is nearly optimal (upto constant factors and logarithmic terms). To the best of our knowledge this is the first guarantee that allows arbitrary distribution on the set of missing entries.**

***Index Terms*—Subspace Tracking, Robust Subspace Tracking, Matrix Completion, Robust Matrix Completion, Online Algorithms**

I. Introduction

It is well known that many real world datasets exhibit a low-dimensional structure. These low dimensional structures arise from a plethora of different possibilities, but the most important of which being that the data are generated from a small number of latent factors. This manifests in several forms, the most popular embeddings being sparsity and low-rank structures. In datasets that are functions of time, i.e., time-series data, it is natural to assume that the latent factors demonstrate statistical variation over time, i.e., the latent factors are non-stationary. In such a scenario, a natural question to ask is whether we can identify and track this low dimensional structure over time from "corrupted" data samples. In particular, is it possible to recover the low dimensional structure when, (a) the data samples are corrupted with arbitrarily large outliers? (b) lot of data is missing? and (c) when the data is corrupted with arbitrary outliers and also a part of the data is missing? And if it is indeed possible to recover the structure, what are some fast, sample efficient, and principled approaches to achieve the goal? In this work, we seek to answer the above questions and propose provable algorithms that achieve this goal.

Subspace Tracking is the problem of tracking a low dimensional subspace that is slowly changing over time. The problem of subspace tracking has been extensively studied in Automatic Control and Signal processing literature. However, most of these works consider the statistically stationary setting, i.e., one in which the underlying subspace is fixed. In this work, we extend this study to include the following (i) We allow for the subspace to change, albeit slowly, and (ii) we assume that due to transmission errors, we are only provided partial observations. This is an ill-posed problem since there are more unknowns than the number of measurements, and thus we need to impose additional constraints to provably solve the problem.

In particular, we assume that the subspace is constant for a $\mathcal{O}(nr \log n \log(1/\varepsilon))$ frames, which is nearly optimal since we need nr measurements to even store a r-dimensional subspace. Further, we show that the subspace change is detected within $Cf^2 r \log n$ frames, which is also nearly optimal. Further, we assume that the subspace change follows the model

A. Contributions

In this work we extend a recent line of work called ReProCS [1], [2] and solve two related problems (i) Subspace Tracking with missing data; and (ii) Robust Subspace Tracking with missing data. In particular, we build upon a recent pre-print [2] in which we proposed Although these problems (or similar variants) have been studied before, there are no complete guarantees for the case when either the data subspace is allowed to change with time, or when the set of missing entries are drawn from an arbitrary distribution. To the best of our knowledge, this is also the first guarantee for the Robust Subspace Tracking with missing data problem.

II. Problem Statement, Algorithm, and Main Result

At each time t, we observe a data vector $\boldsymbol{y}_t \in \mathbb{R}^n$ that satisfies

$$\boldsymbol{y}_t = \mathcal{P}_{\Omega_t}(\boldsymbol{\ell}_t) + \boldsymbol{\nu}_t, \text{ for } t = 1, 2, \dots, d \tag{1}$$

where $\mathcal{P}_{\Omega_t}(\cdot)$ selects elements that are indexed by the set Ω_t, $\boldsymbol{\nu}_t$ is small unstructured noise, $\boldsymbol{\ell}_t$ is the true data vector that lies in a fixed or slowly changing low-dimensional subspace of $\mathbb{R}^n$, i.e., $\boldsymbol{\ell}_t = \boldsymbol{P}_{(t)} \boldsymbol{a}_t$ where $\boldsymbol{P}_{(t)}$ is an $n \times r$ basis matrix with $r \ll n$ and with $\|(\boldsymbol{I} - \boldsymbol{P}_{(t-1)} \boldsymbol{P}_{(t-1)}{}') \boldsymbol{P}_{(t)}\|$ small compared to $\|\boldsymbol{P}_{(t)}\| = 1$. We use $\mathcal{T}_t$ to denote the support set of $\boldsymbol{x}_t$. The goal is to track $\text{span}(\boldsymbol{P}_{(t)})$ and $\boldsymbol{\ell}_t$ either immediately or within a short delay. The initial subspace estimate, $\hat{\boldsymbol{P}}_0$, can be computed by generating it from the Random Orthogonal Model [3]. As explained in [3], a basis matrix generated from this model is already μ-incoherent. We have the following corollary. Denoting the set of missing entries at time t as $\mathcal{T}_t$, (1) can also be written as

$$\boldsymbol{y}_t := \boldsymbol{\ell}_t - \boldsymbol{I}_{\mathcal{T}_t} \boldsymbol{I}_{\mathcal{T}_t}{}' \boldsymbol{\ell}_t, \text{ for } t = 1, 2, \dots, d. \tag{2}$$

This method of expressing the problem allows us to tap into existing literature in the area of Robust Subspace Tracking(RST). Subspace Tracking with missing data (`ST-missing`) can be considered to be a special case of RST, wherein, the sparse, outliers are replaced with $\boldsymbol{x}_t \equiv \boldsymbol{I}_{\mathcal{T}_t}\boldsymbol{I}_{\mathcal{T}_t}{}'\boldsymbol{\ell}_t$. Notice that using the standard set difference notation, and defining $[n] := \{1, \cdots, n\}$, the two sets can be related as $\mathcal{T}_t = [n] \backslash \Omega_t = \Omega_t^c$.

A. Main Result

Before stating the result, we precisely define max-miss-frac-col and max-miss-frac-row$^\alpha$. Since NORST is an online approach that performs outlier support recovery one data vector at a time, it needs different bounds on both. Let max-miss-frac-col $:= \max_t |\mathcal{T}_t|/n$. We define max-miss-frac-row$^\alpha$ as the maximum fraction of outliers (nonzeros) per row of any sub-matrix of $\boldsymbol{X}$ with α consecutive columns. To understand this precisely, for a time interval, $\mathcal{J}$, define $\gamma(\mathcal{J}) := \max_{i=1,2,\ldots,n} \frac{1}{|\mathcal{J}|}\sum_{t\in\mathcal{J}} \mathbf{1}_{\{i\in\mathcal{T}_t\}}$ where $\mathbf{1}_S$ is the indicator function for statement S. Thus, $\sum_{t\in\mathcal{J}} \mathbf{1}_{\{i\in\mathcal{T}_t\}}$ counts the number of outliers (nonzeros) in row i of $\boldsymbol{X}_{\mathcal{J}}$, and so $\gamma(\mathcal{J})$ is the maximum outlier fraction in any row of the sub-matrix $\boldsymbol{X}_{\mathcal{J}}$ of $\boldsymbol{X}$. Let $\mathcal{J}^\alpha$ denote a time interval of duration α. Then max-miss-frac-row$^\alpha := \max_{\mathcal{J}^\alpha\subseteq[1,d]} \gamma(\mathcal{J}^\alpha)$.

We use $\hat{t}_j$ to denote the time instant at which the j-th subspace change time is detected by Algorithm 1.

Theorem 2.1 (ST-miss)**.** *Consider Algorithm 1. Let* $\alpha := Cf^2 r\log n$, $\boldsymbol{\Lambda} := \mathbb{E}[\boldsymbol{a}_1\boldsymbol{a}_1{}']$, $\lambda^+ := \lambda_{\max}(\boldsymbol{\Lambda})$, $\lambda^- := \lambda_{\min}(\boldsymbol{\Lambda})$, $f := \lambda^+/\lambda^-$. *Pick an* $\varepsilon \le \min(0.01, 0.03\min_j \mathrm{SE}(\boldsymbol{P}_{j-1}, \boldsymbol{P}_j)^2/f)$. *Let* $K := C\log(1/\varepsilon)$. *If*

1) $\boldsymbol{P}_j$*'s are* μ*-incoherent; and* $\boldsymbol{a}_t$*'s are zero mean, mutually independent over time* t*, have identical covariance matrices, i.e.* $\mathbb{E}[\boldsymbol{a}_t\boldsymbol{a}_t{}'] = \boldsymbol{\Lambda}$*, are element-wise uncorrelated (*$\boldsymbol{\Lambda}$ *is diagonal), are element-wise bounded (for a numerical constant* η*,* $(\boldsymbol{a}_t)_i^2 \le \eta\lambda_i(\boldsymbol{\Lambda})$*), and are independent of all outlier supports* $\mathcal{T}_t$*;*
2) $\|\boldsymbol{\nu}_t\|^2 \le cr\|\mathbb{E}[\boldsymbol{\nu}_t\boldsymbol{\nu}_t{}']\|$, $\|\mathbb{E}[\boldsymbol{\nu}_t\boldsymbol{\nu}_t{}']\| \le c\varepsilon^2\lambda^-$, $\boldsymbol{\nu}_t$*'s are zero mean, mutually independent, and independent of* $\boldsymbol{\ell}_t$*;*
3) *max-miss-frac-col* $\le c_1/\mu r$*, max-miss-frac-row*$^\alpha \le b_0 := \frac{c_2}{f^2}$*;*
4) *subspace change: let* $\Delta := \max_j \mathrm{SE}(\boldsymbol{P}_{j-1}, \boldsymbol{P}_j)$*, assume that*
   a) $t_{j+1} - t_j > (K+2)\alpha$*, and*
   b) $\Delta \le 0.8$*;*

then, with probability (w.p.) at least $1 - 10dn^{-10}$*,*

$$\mathrm{SE}(\hat{\boldsymbol{P}}_{(t)}, \boldsymbol{P}_{(t)}) \le \begin{cases} (\varepsilon+\Delta) & \text{if } t \in \mathcal{J}_1, \\ (0.3)^{k-1}(\varepsilon+\Delta) & \text{if } t \in \mathcal{J}_k, \\ \varepsilon & \text{if } t \in J_K. \end{cases}$$

where, $\mathcal{J}_1 = [t_j, \hat{t}_j + \alpha)$, $\mathcal{J}_k = [\hat{t}_j + (k-1)\alpha, \hat{t}_j + k\alpha)$ *and* $\mathcal{J}_K = [\hat{t}_j + K\alpha + \alpha, t_{j+1})$.

Algorithm 1 NORST-random.

1: **Input**: $\boldsymbol{y}_t$, $\mathcal{T}_t$ **Output**: $\hat{\boldsymbol{\ell}}_t$, $\hat{\boldsymbol{P}}_{(t)}$
2: **Parameters:** r, $K \leftarrow C\log(1/\varepsilon)$, $\alpha \leftarrow Cf^2 r\log n$, $\omega_{evals} \leftarrow 2\varepsilon^2\lambda^+$.
3: $\hat{\boldsymbol{P}}_0 \leftarrow \boldsymbol{0}_{n\times r}$, $\hat{\boldsymbol{P}}_{(t_{\text{train}})} \leftarrow \hat{\boldsymbol{P}}_0$
4: $j \leftarrow 1$, $k \leftarrow 1$
5: phase $\leftarrow$ update; $\hat{t}_0 \leftarrow t_{\text{train}}$;
6: **for** $t > t_{\text{train}}$ **do**
7: $\quad\boldsymbol{\Psi} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_{(t-1)}\hat{\boldsymbol{P}}_{(t-1)}{}'$; $\tilde{\boldsymbol{y}}_t \leftarrow \boldsymbol{\Psi}\boldsymbol{y}_t$;
8: $\quad\hat{\boldsymbol{x}}_t \leftarrow \boldsymbol{I}_{\mathcal{T}_t}(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t})^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\tilde{\boldsymbol{y}}_t$
9: $\quad\hat{\boldsymbol{\ell}}_t \leftarrow \boldsymbol{y}_t - \hat{\boldsymbol{x}}_t$.
10: $\quad$**if** phase $=$ detect and $t = \hat{t}_{j-1,fin} + u\alpha$ **then**
11: $\quad\quad\boldsymbol{\Phi} \leftarrow (\boldsymbol{I} - \hat{\boldsymbol{P}}_{j-1}\hat{\boldsymbol{P}}_{j-1}{}')$.
12: $\quad\quad\boldsymbol{B} \leftarrow \boldsymbol{\Phi}\hat{\boldsymbol{L}}_{t,\alpha}$
13: $\quad\quad$**if** $\lambda_{\max}(\boldsymbol{B}\boldsymbol{B}') \ge \alpha\omega_{evals}$ **then**
14: $\quad\quad\quad$phase $\leftarrow$ update, $\hat{t}_j \leftarrow t$,
15: $\quad\quad$**end if**
16: $\quad$**end if**
17: $\quad$**if** phase $=$ update **then**
18: $\quad\quad$**if** $t = \hat{t}_j + u\alpha - 1$ for $u = 1,\ 2,\ \cdots$, **then**
19: $\quad\quad\quad\hat{\boldsymbol{P}}_{j,k} \leftarrow SVD_r[\hat{\boldsymbol{L}}_{t;\alpha}]$, $\hat{\boldsymbol{P}}_{(t)} \leftarrow \hat{\boldsymbol{P}}_{j,k}$, $k \leftarrow k+1$.
20: $\quad\quad$**else**
21: $\quad\quad\quad\hat{\boldsymbol{P}}_{(t)} \leftarrow \hat{\boldsymbol{P}}_{(t-1)}$
22: $\quad\quad$**end if**
23: $\quad\quad$**if** $t = \hat{t}_j + K\alpha - 1$ **then**
24: $\quad\quad\quad\hat{t}_{j,fin} \leftarrow t$, $\hat{\boldsymbol{P}}_j \leftarrow \hat{\boldsymbol{P}}_{(t)}$
25: $\quad\quad\quad k \leftarrow 1$, $j \leftarrow j+1$, phase $\leftarrow$ detect.
26: $\quad\quad$**end if**
27: $\quad$**end if**
28: **end for**
29: **Offline NORST**: At $t = \hat{t}_j + K\alpha$
30: **for** $t \in [\hat{t}_{j-1} + K\alpha, \hat{t}_j + K\alpha - 1]$ **do**
31: $\quad\hat{\boldsymbol{P}}_{(t)}^{\text{offline}} \leftarrow [\hat{\boldsymbol{P}}_{j-1}, (\boldsymbol{I} - \hat{\boldsymbol{P}}_{j-1}\hat{\boldsymbol{P}}_{j-1}{}')\hat{\boldsymbol{P}}_j]$
32: $\quad\boldsymbol{\Psi} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_{(t)}^{\text{offline}}\hat{\boldsymbol{P}}_{(t)}^{\text{offline}\prime}$
33: $\quad\hat{\boldsymbol{x}}_t^{\text{offline}} \leftarrow \boldsymbol{I}_{\mathcal{T}_t}(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t})^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{y}_t$
34: $\quad\hat{\boldsymbol{\ell}}_t^{\text{offline}} \leftarrow \boldsymbol{y}_t - \hat{\boldsymbol{x}}_t^{\text{offline}}$.
35: **end for**

Corollary 2.2. *Under Theorem 2.1 assumptions, the following also hold:*

1) $\|\hat{\boldsymbol{\ell}}_t - \boldsymbol{\ell}_t\| \le 1.2(\mathrm{SE}(\hat{\boldsymbol{P}}_{(t)}, \boldsymbol{P}_{(t)}) + \varepsilon)\|\boldsymbol{\ell}_t\|$ *with* $\mathrm{SE}(\hat{\boldsymbol{P}}_{(t)}, \boldsymbol{P}_{(t)})$ *bounded as above,*
2) $t_j \le \hat{t}_j \le t_j + 2\alpha$,

B. The NORST-random Algorithm

The algorithm is a minor modification of the one presented in [2]. The algorithm works as follows. Firstly, we show that generating the initialization, $\hat{\boldsymbol{P}}_0$ using the Random Orthogonal model, it follows that with high probability, $\hat{\boldsymbol{P}}_0$ is μ-incoherent for some constant μ. Following the idea of [1] incoherence of $\hat{\boldsymbol{P}}_0$, the triangle inequality shows that $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_0\hat{\boldsymbol{P}}_0{}'$ satisfies the $2s$ Restricted Isometry Property. With this, we obtain a bound on $\left\|(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t})^{-1}\right\|$ which is further used in the subspace estimation step (lines $10-28$ of

Algorithm 2 NORST-random for Dynamic Robust Matrix Completion.

1: **Input**: $\boldsymbol{y}_t$, $\mathcal{T}_t$ **Output**: $\hat{\boldsymbol{\ell}}_t$, $\hat{\boldsymbol{P}}_{(t)}$
2: **Parameters:** r, $K \leftarrow C\log(1/\varepsilon)$, $\alpha \leftarrow Cf^2 r\log n$, $\omega_{evals} \leftarrow 2\varepsilon^2\lambda^+$.
3: $\hat{\boldsymbol{P}}_0 \leftarrow n \times r$ basis matrix from NO-RMC [5].
4: $\hat{\boldsymbol{P}}_{(t_{\text{train}})} \leftarrow \hat{\boldsymbol{P}}_0$; $j \leftarrow 1$, $k \leftarrow 1$
5: phase $\leftarrow$ update; $\hat{t}_0 \leftarrow t_{\text{train}}$;
6: **for** $t > t_{\text{train}}$ **do**
7: $\quad\boldsymbol{\Psi} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_{(t-1)}\hat{\boldsymbol{P}}_{(t-1)}{}'$; $\tilde{\boldsymbol{y}}_t \leftarrow \boldsymbol{\Psi}\boldsymbol{y}_t$;
8: $\quad\mathcal{T}_t^1 \leftarrow (\mathcal{T}_t^{\text{miss}})^{\mathsf{C}}$
9: $\quad\hat{\boldsymbol{x}}_{t,cs} \leftarrow \arg\min_{\boldsymbol{x}} \left\|(\boldsymbol{x})_{(\mathcal{T}_t^1)}\right\|_1$ s.t $\|\tilde{\boldsymbol{y}}_t - \boldsymbol{\Psi}\boldsymbol{x}\| \leq \xi$.
10: $\quad\hat{\mathcal{T}}_t^{\text{sparse}} \leftarrow \{i : |\hat{\boldsymbol{x}}_{t,cs}| > \omega_{supp}\}$
11: $\quad\hat{\mathcal{T}}_t \leftarrow \hat{\mathcal{T}}_t^{\text{sparse}} \cup \mathcal{T}_t^{\text{miss}}$
12: $\quad\hat{\boldsymbol{x}}_t \leftarrow \boldsymbol{I}_{\hat{\mathcal{T}}_t}(\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t}{}'\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t})^{-1}\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t}{}'\tilde{\boldsymbol{y}}_t$
13: $\quad\hat{\boldsymbol{\ell}}_t \leftarrow \boldsymbol{y}_t - \hat{\boldsymbol{x}}_t$.
14: $\quad$Lines $10-27$ of Algorithm 1
15: **end for**

Algorithm 1). The Least Squares step (line 8 of Algorithm 1) also ensures that the "noise" is sparse and data dependent. In this setting, we can easily apply the result of [4] to obtain finite sample guarantees for the overall algorithm.

III. Dynamic Robust Matrix Completion

As a direct corollary of Theorem 2.1, we are also able to provide the following guarantee for the dynamic counterpart of the Robust Matrix Completion problem. The robust matrix completion problem is a strict generalization of both the Matrix Completion and the Robust PCA problem. Formally, we observe the n-dimensional data vectors that satisfy

$$\boldsymbol{y}_t = \mathcal{P}_{\Omega_t}(\boldsymbol{\ell}_t + \boldsymbol{x}_t) + \boldsymbol{\nu}_t \tag{3}$$

where $\boldsymbol{\ell}_t$'s span a slowly changing low dimensional subspace, $\boldsymbol{x}_t$'s are sparse-vectors, and $\boldsymbol{\nu}_t$'s are the small uncorrelated noise. The projection $\mathcal{P}_{\Omega_t}(\cdot)$ is the same as before. However, we cannot use the same algorithm to solve this problem since we do not know the support of the outliers, $\boldsymbol{x}_t$'s. Thus, we need to include a sparse recovery step as done in [2, Algorithm 2]. In particular, since this is a more general problem than that studied in [2], we need to make the following changes to the algorithm and the guarantee. The modified algorithm is provided as Algorithm 2.

Firstly, we define $\mathcal{T}_t^{\text{miss}} = \Omega_t^{\mathsf{C}}$ and we use $\mathcal{T}_t^{\text{sparse}}$ to denote the support of $\boldsymbol{x}_t$. Now, re-defining $\mathcal{T}_t = \mathcal{T}_t^{\text{miss}} \cup \mathcal{T}_t^{\text{sparse}}$, the goal is to recover $\mathcal{T}_t$ as done in [2, Algorithm 2]. A key idea that we use here is that since we know $\mathcal{T}_t^{\text{miss}}$ the problem of estimating $\mathcal{T}_t$ is that of support recovery with partial knowledge. In particular, we can now tap into the result of [6]. Another difference is that we need a good estimate of the initial subspace, which can no longer be set directly as a matrix of zeros. This is explained in detail in [2], but to intuitively understand this, we need to bound the $2s$-restricted isometry constant of $\boldsymbol{\Psi} = \boldsymbol{I} - \hat{\boldsymbol{P}}_{\text{init}}\hat{\boldsymbol{P}}_{\text{init}}{}'$ by a small number that is strictly smaller than one.

To satisfy the initialization condition, we make the following assumption. For the initial $Cr\log n\log(1/\varepsilon)$ samples, we assume that one of the following hold. (a) Either we have outlier-free data available so that we can use the approach of NORST-random for initialization, or (b) There are no missing entries so that any of the batch RPCA techniques such as AltProj [7] can be applied to get a coarse initialization, followed by improving the estimate using NORST [2]. We also cannot directly obtain a coarse estimate using a batch RMC technique such as NO-RMC [5] since this requires $\mathcal{O}(n)$ samples.

??pn – need to edit this to include only the changes required rather than re-stating the whole result.

Corollary 3.3 (Robust-ST-miss)**.** *Consider Algorithm 2. Let* $\alpha := Cf^2 r\log n$, $\boldsymbol{\Lambda} := \mathbb{E}[\boldsymbol{a}_1\boldsymbol{a}_1{}']$, $\lambda^+ := \lambda_{\max}(\boldsymbol{\Lambda})$, $\lambda^- := \lambda_{\min}(\boldsymbol{\Lambda})$, $f := \lambda^+/\lambda^-$ *and let* $x_{\min} := \min_t \min_{i\in\mathcal{T}_t}(\boldsymbol{x}_t)_i$ *denote the minimum outlier magnitude. Pick an* $\varepsilon \leq \min(0.01, 0.03\min_j \text{SE}(\boldsymbol{P}_{j-1}, \boldsymbol{P}_j)^2/f)$. *If*

1) $\boldsymbol{P}_j$*'s are* μ*-incoherent; and* $\boldsymbol{a}_t$*'s are zero mean, mutually independent over time* t*, have identical covariance matrices, i.e.* $\mathbb{E}[\boldsymbol{a}_t\boldsymbol{a}_t{}'] = \boldsymbol{\Lambda}$*, are element-wise uncorrelated (*$\boldsymbol{\Lambda}$ *is diagonal), are element-wise bounded (for a numerical constant* η*,* $(\boldsymbol{a}_t)_i^2 \leq \eta\lambda_i(\boldsymbol{\Lambda})$*), and are independent of all outlier supports* $\mathcal{T}_t$*;*
2) $\|\boldsymbol{\nu}_t\|^2 \leq cr\|\mathbb{E}[\boldsymbol{\nu}_t\boldsymbol{\nu}_t{}']\|$, $\|\mathbb{E}[\boldsymbol{\nu}_t\boldsymbol{\nu}_t{}']\| \leq c\varepsilon^2\lambda^-$, $\boldsymbol{\nu}_t$*'s are zero mean, mutually independent, and independent of* $\boldsymbol{x}_t, \boldsymbol{\ell}_t$*;*
3) *miss-entry-frac-col* $\leq c_1/\mu r$*, miss-entry-frac-row*$^\alpha \leq b_0 := \frac{c_2}{f^2}$*;*
4) *subspace change: let* $\Delta := \max_j \text{SE}(\boldsymbol{P}_{j-1}, \boldsymbol{P}_j)$*, assume that*
   a) $t_{j+1} - t_j > (K+2)\alpha$*, and*
   b) $\Delta \leq 0.8$ *and* $C_1\sqrt{r\lambda^+}(\Delta + 2\varepsilon) \leq x_{\min}$
5) *initialization satisfies* $\text{SE}(\hat{\boldsymbol{P}}_0, \boldsymbol{P}_0) \leq 0.25$, $C_1\sqrt{r\lambda^+}\text{SE}(\hat{\boldsymbol{P}}_0, \boldsymbol{P}_0) \leq x_{\min}$*;*

then, with probability (w.p.) at least $1 - 10dn^{-10}$*,*

$$\text{SE}(\hat{\boldsymbol{P}}_{(t)}, \boldsymbol{P}_{(t)}) \leq \begin{cases} (\varepsilon + \Delta) & \text{if } t \in \mathcal{J}_1, \\ (0.3)^{k-1}(\varepsilon + \Delta) & \text{if } t \in \mathcal{J}_k, \\ \varepsilon & \text{if } t \in J_K. \end{cases}$$

where, $\mathcal{J}_1 = [t_j, \hat{t}_j + \alpha)$, $\mathcal{J}_k = [\hat{t}_j + (k-1)\alpha, \hat{t}_j + k\alpha)$ *and* $\mathcal{J}_K = [\hat{t}_j + K\alpha + \alpha, t_{j+1})$.

Proof. The proof follows from using the same ideas that are used to prove Theorem 2.1 and replacing the guarantee of the sparse recovery step of [8] with [9] □

IV. Discussion and Prior Art

There are, broadly speaking, two primary lines of work that relates to the problem of Subspace Tracking with missing data (ST-missing). The first one is that which evolved through classical literature on Subspace Tracking and Subspace Identification and the more modern approach that stems from Matrix Completion. The problem of subspace

tracking and identification has been studied extensively in the literature, see for e.g., [10], [11] provide comprehensive overview. It finds applications in Seismic Event detection [12], identifying moving objects in videos [13], detecting anomalies in brain networks [14] to name a few. However, due to uncertainty in source/channel coding schemes, communication over networks, hardware problems associated with acquisition sensors, frequently, the observation are "incomplete" in that a part of the data is missing.

This problem of subspace tracking and identification in missing data has also received considerable interest in recent years, in [15], [16], [17], to name a few. Despite significant theoretical and algorithmic advances in the methods to analyze this problem, the best existing theoretical result due to [18] only provides convergence to a single subspace. In this work we seek to extend this and provide finite sample guarantees for the problem where the "true data subspace" is allowed to change over time. In particular, we must mention that a short version of the main theoretical result (Theorem 2.1) appeared in [2]. In this work, we provide a modification to the algorithm and a guarantee to solve the dynamic Robust Matrix Completion of Roubst Subspace Tracking with missing entries problem (see Corollary 3.3). We also validate our algorithm through extensive numerical evaluation.

The problem of subspace tracking with missing data is closely related to *Matrix Completion*, and, in fact, the problems are exactly equivalent if the data subspace is assumed to be fixed. Both these problems have been studied extensively in recent years. Missing data increases the complexity of the problem and now we need to resolve the identifiability issue that arises. To understand this intuitively, if the low rank matrix (obtained by stacking the data stream samples as columns) is also sparse, there is no hope to recover the elements at the points where the data is not observed. Thus to alleviate this, the Matrix Completion line of work imposes an additional constraint of "incoherence" to the subspace.

A seminal paper by Candès and Recht [19] showed that casting the matrix completion as a convex problem, and minimizing the *nuclear norm* provided exact recovery of the low rank matrix. However, since the matrix is ill-posed, it assumed that (i) the matrix is incoherent; and (ii) the set of observed entries is drawn from a uniform random probability distribution. The first assumption is essential to analysis as explained earlier, however, imposing a distribution on the set of observed entries is neither practically valid, nor is essential to analysis. Although, there is a vast body of provable literature which analyse MC, most of these approaches also assume that the set of observed entries is generated uniformly at random. Some very recent papers that do not require a uniform random model, include [20] which uses leverage score based sampling, [21] which requires that the set of observed entries follows power-law distributed graphs, and finally, [22] assumes that set of observed entries are generated from the edges of a bipartite graph with a large spectral gap. However, to the best of our knowledge, Theorem 2.1 is the first provable guarantee for the ST-missing problem without imposing *any* assumptions on the set of observed entries, barring that the fraction of missing entries per row and per column is upper bounded. This is a more realistic assumption.

To the best of our knowledge, Theorem 2.1 is the *first complete, non-asymptotic guarantee* for ST-missing; and the first result that allows changing subspaces. All existing guarantees only provide asymptotic results [15], [16], or consider a fixed subspace [18]. Moreover, from a dynamic matrix completion viewpoint, Theorem I also giving a matrix completion solution without assuming that the set of observed entries is generated from a uniform or a Bernoulli model. Of course the tradeoff is that it needs many more observed entries.

Finally, an easy extension to the RST-missing problem is provided in Corollary 3.3. This problem in the static setting, i.e., the Robust Matrix Completion has received significant interest in the recent years. This problem is a strict generalization of the Robust PCA [23], [24] and the Matrix Completion [19], [3] problems. Some works that present a nice treatment are [5], [25], [26], [27], [28], [29]. In the dynamic setting, i.e., if the data subspace is allowed to change over time, there is a line of work that extends the GROUSE and GRASTA framework in [30]. However, this is a partial guarantee since it imposes unrealistic assumptions on the intermediate algorithm estimates, and only provides an asymptotic guarantee. To the best of our knowledge, Corollary 3.3 is the first result to consider the case of changing subspace and provide a complete, finite-sample guarantee for the RST-missing problem.

V. Empirical Evaluation

In this section we present the results for extensive numerical experiments on synthetic and real data to validate our theoretical claims. All time comparisons are performed on a Desktop Computer with Intel® Xeon E3-1240 8-core CPU @ 3.50GHz and 32 GB RAM and all synthetic data experiments are averaged over 50 independent trials. The codes are available at https://github.com/praneethmurthy/NORST-random. For simplicity, in this section we refer to NORST-random as NORST.

A. Synthetic Data

Experiment 1. In the first experiment, we compare NORST-random with Matrix Completion and the ST-missing algorithms. We compare the results of NORST with Matrix Completion algorithms, Subspace Tracking with missing data methods proposed in literature. For our first experiment, we generate the changing subspaces using $\boldsymbol{P}_j = e^{\gamma_j \boldsymbol{B}_j} \boldsymbol{P}_{j-1}$ as done in [30] where γ_j controls the subspace change and $\boldsymbol{B}_j$'s are skew-symmetric matrices. In the first experiment we used the following parameters. $n = 1000$, $d = 11000$, $J = 2$, $t_1 = 5000$, $t_2 = 8000$, $r = 30$, $\gamma_1 = 0.005$, $\gamma_2 = 0.004$ and the matrices $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are generated as $\boldsymbol{B}_1 = (\tilde{\boldsymbol{B}}_1 - \tilde{\boldsymbol{B}}_1{}')$ and $\boldsymbol{B}_2 = (\tilde{\boldsymbol{B}}_2 - \tilde{\boldsymbol{B}}_2{}')$ where

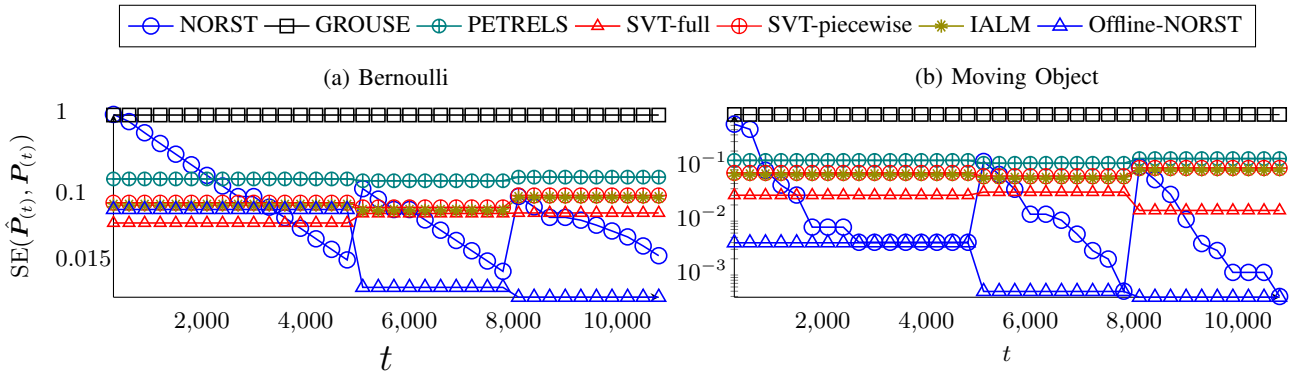

Fig. 1: Left plot illustrates the Subspace error for missing entry supports generated using Bernoulli Model with the fraction of observed entries $\rho = 0.3$ and right plot illustrates the subspace error when the set of missing entries are generated according to the Moving Object model with fraction of observed entries 0.75. The values are plotted every $k\alpha - 1$ time-frames. All results are averaged over 50 independent runs.

TABLE I: Comparison of $\|\hat{\boldsymbol{L}} - \boldsymbol{L}\|_F / \|\boldsymbol{L}\|_F$ for Matrix Completion and Subspace Tracking methods. The time taken per data sample (frame) for the Moving Object model is given in parentheses. The offline (batch) methods are performed once on the complete dataset.

Outlier Model	GROUSE (5.0 ms)	PETRELS (27.1 ms)	**NORST (7.1 ms)**	SVT-full (46.2ms)	SVT-piecewise (43.6ms)	IALM-full (7.7 ms)	**Offline NORST (14 ms)**
Bernoulli ($\rho = 0.3$)	1.0294	0.1067	**0.2148**	0.0856	0.0563	0.0764	**0.0036**
Moving Object ($\rho = 0.75$)	0.7021	0.0815	**0.1105**	0.0705	0.0172	0.0988	**0.0188**

the entries of $\tilde{\boldsymbol{B}}_1, \tilde{\boldsymbol{B}}_2$ are generated independently from a standard normal distribution. This gives us the basis matrices $\boldsymbol{P}_{(t)}$ for all t and the choice of γ_1 and γ_2 ensure that $\mathrm{SE}(\boldsymbol{P}_1, \boldsymbol{P}_0) \approx \mathrm{SE}(\boldsymbol{P}_2, \boldsymbol{P}_1) \approx 0.01$. To obtain the low-rank matrix $\boldsymbol{L}$ from this we generate the coefficients $\boldsymbol{a}_t \in \mathbb{R}^r$ as independent zero-mean, bounded random variables. They are $(\boldsymbol{a}_t)_i \overset{i.i.d}{\sim} unif[-q_i, q_i]$ where $q_i = \sqrt{f} - \sqrt{f}(i-1)/2r$ for $i = 1, 2, \cdots, r-1$ and $q_r = 1$. thus the condition number is f and we selected $f = 10$. For the missing entries supports, we considered two models according to which the supports are generated. First we use Model G.24 [31] which simulates a moving object pacing in the video. For we used $s/n = 0.25$ and $b_0 = 0.25$. Thus, the fraction of observed entries is 0.75 but the missing entry supports Ω_t are highly correlated over time. Secondly, we used the Bernoulli model to simulate sampling uniformly at random, i.e., each entry of the matrix, is independently selected with probability ρ or not selected with probability $1-\rho$. We generate the sparse supports using the Bernoulli model using $\rho = 0.3$. Thus, in expectation, the fraction of observed entries is 0.3.

We initialized the NORST algorithm using the Random Orthogonal Model. The other algorithm parameters are set as mentioned in the theorem, i.e., $K = \lceil \log(c/\varepsilon) \rceil = 8$, $\alpha = Cr \log n = 300$, $\omega_{evals} = 2\varepsilon^2 \lambda^+ = 7.5 \times 10^{-4}$. For the least-squares step we use the Conjugate Gradient Least Squares instead of the well-known "backslash" operator in `MATLAB` since this is a well conditioned problem. For this we set the tolerance as 10^{-10} and the number of iterations as 10. We have not done any code optimization such as use of `MEX` files for various sub-routines to speed up our algorithm.

For the ST-missing algorithms, we implement the algorithms without modifications. In particular, since the PETRELS [16] algorithm is based on GROUSE [32], the common parameter, max_cycles is set as 1. For GROUSE, we use the step size as 0.1 as specified in the code and the maximum rank estimate is provided as r. For PETRELS, we use $\lambda = 0.98$ as suggested in the code from authors' website. For the Nuclear Norm Minimization (NNM) implementation, we use two solvers, (i) Singular Value Thresholding (`SVT`) [33] and (ii) Inexact Augmented Lagrangian Multiplier (`IALM`) [34]. For SVT, we use the default parameters, namely number of iterations as 500, tolerance as 10^{-4}, $\delta = 2$, $\tau = 5\sqrt{nd}$ (this value is changed to t_1, $t_2 - t_1$ and $d - t_2$ for the piecewise implementations). For IALM also we use the default parameters, namely number of iterations 100 and tolerance 10^{-4}. For the piecewise implementation of SVT, we estimate $\hat{\boldsymbol{L}}_{(t_{j-1}, t_j]}$ for each subspace change interval using only the relevant frames followed by $\hat{\boldsymbol{P}}_{j-1} = \mathrm{basis}(\hat{\boldsymbol{L}}_{(t_{j-1}, t_j]})$.

Experiment 2. In this experiment we consider the problem of *Static Matrix Completion*, i.e., the setting of the previous experiment but when the subspace does not change. Thus, this can be thought of as the setting of Matrix Completion. We use exactly the two previous settings as before with the exception that, now, $d = 3000$ and $J = 1$, i.e., there are no subspace changes. In this case, we observe that NORST and Offline-NORST are generally outperformed by the other algorithms.

REFERENCES

[1] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust pca or recursive sparse recovery in large but structured noise," *IEEE Trans. Info. Th.*, pp. 5007–5039, August 2014.

[2] P. Narayanamurthy and N. Vaswani, "Nearly optimal robust subspace tracking," *arXiv:1712.06061 [cs.IT] (Submitted to IEEE Trans. Info Theory)*, 2017.

[3] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. of Comput. Math*, , no. 9, pp. 717–772, 2008.

[4] N. Vaswani and P. Narayanamurthy, "Finite sample guarantees for pca in non-isotropic and data-dependent noise," *arXiv:1709.06255*, 2017.

[5] Y. Cherapanamjeri, K. Gupta, and P. Jain, "Nearly-optimal robust matrix completion," *ICML*, 2016.

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

[7] P. Netrapalli, U N Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust pca," in *NIPS*, 2014.

[8] E. Candes, "The restricted isometry property and its implications for compressed sensing," *C. R. Math. Acad. Sci. Paris Serie I*, 2008.

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

[10] Pierre Comon and Gene H Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proceedings of the IEEE*, vol. 78, no. 8, pp. 1327–1343, 1990.

[11] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning and tracking," *accepted with minor revisions for IEEE Signal Proc. Magazine*, 2018.

[12] G. S. Wagner and T. J. Owens, "Signal detection using multi-channel seismic data," *Bulletin of the Seismological Society of America*, vol. 86, no. 1A, pp. 221–231, 1996.

[13] J.P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 159–179, 1998.

[14] A. Ozdemir, E. M. Bernat, and S. Aviyente, "Recursive tensor subspace tracking for dynamic brain network analysis," *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

[15] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Sig. Proc.*, pp. 95–107, 1995.

[16] Y. Chi, Y. C. Eldar, and R. Calderbank, "Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations," *IEEE Trans. Sig. Proc.*, December 2013.

[17] L. Balzano, B. Recht, and R. Nowak, "Online Identification and Tracking of Subspaces from Highly Incomplete Information," in *Allerton Conf. Comm., Control, Comput.*, 2010.

[18] Dejiao Zhang and Laura Balzano, "Global convergence of a grassmannian gradient descent algorithm for subspace estimation," in *AISTATS*, 2016.

[19] M. Fazel, "Matrix rank minimization with applications," *PhD thesis, Stanford Univ*, 2002.

[20] Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward, "Coherent matrix completion," in *International Conference on Machine Learning*, 2014, pp. 674–682.

[21] Raghu Meka, Prateek Jain, and Inderjit S Dhillon, "Matrix completion from power-law distributed samples," in *Advances in neural information processing systems*, 2009, pp. 1258–1266.

[22] Srinadh Bhojanapalli and Prateek Jain, "Universal matrix completion," in *International Conference on Machine Learning*, 2014, pp. 1881–1889.

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

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

[25] Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi, "Robust matrix completion with corrupted columns," in *ICML*, 2011.

[26] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust pca via gradient descent," in *NIPS*, 2016.

[27] Olga Klopp, Karim Lounici, and Alexandre B Tsybakov, "Robust matrix completion," *Probability Theory and Related Fields*, vol. 169, no. 1-2, pp. 523–564, 2017.

[28] Daniel Hsu, Sham M Kakade, and Tong Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7221–7234, 2011.

[29] Feiping Nie, Hua Wang, Xiao Cai, Heng Huang, and Chris Ding, "Robust matrix completion via joint schatten p-norm and lp-norm minimization," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 566–574.

[30] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *IEEE Conf. on Comp. Vis. Pat. Rec. (CVPR)*, 2012.

[31] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust pca or robust subspace tracking," *arXiv:1705.08948 (submitted to IEEE Trans. Info. Theory)*, 2017.

[32] Laura Balzano, Robert Nowak, and Benjamin Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 704–711.

[33] J. Cai, E. J Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[34] Zhouchen Lin, Minming Chen, and Yi Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.