PROVABLE MEMORY-EFFICIENT ONLINE ROBUST MATRIX COMPLETION

Praneeth Narayanamurthy, Vahid Daneshpajooh and Namrata Vaswani

Department of ECE, Iowa State University, Ames, IA
{pkurpadn, vahidd, namrata}@iastate.edu

ABSTRACT

Robust Matrix Completion (RMC) is the problem of estimating a low-rank matrix in the presence of missing entries and element-wise (sparse) outliers. In this work, we study the RMC problem with the extra assumption that the clean data is generated from either a fixed or a slowly-changing low-dimensional subspace and introduce a provably correct online algorithm for solving it. Our problem can also be interpreted as that of Robust Subspace Tracking with missing data (RST-miss); with "robust" referring to robustness to sparse outliers. Our proposed method, called NORST-miss-robust, and its guarantee both rely on the Recursive Projected Compressive Sensing (ReProCS) framework introduced in our earlier work. We also argue that NORST-miss-robust enjoys near-optimal memory complexity, tracks subspace changes with near-optimal delay, and has time complexity that is order-wise equal to that of vanilla PCA. Detailed experimental comparisons showing the practical advantages of our method are also shown.

***Index Terms*—** Robust Matrix Completion, Online algorithm

1. INTRODUCTION

Robust Matrix Completion (RMC) is the problem of estimating a low-rank matrix by observing a subset of its entries some of which may also be corrupted by sparse outliers. Key applications include recommendation system design, detection of anomalous behavior in dynamic social networks and video analytics. In recommendation systems, only a small subset of data is observed because all users do not label all items, while sparse outliers occur due to users entering a few wrong ratings, which could be due to typographic errors or malicious intent. In video analytics, foreground occlusions are often the source of both missing and corrupted data: if the occlusion is easy to detect by simple procedure such as color-based thresholding, then the occluding pixel can be labeled as "missing" while if this cannot be detected easily, it is labeled as an outlier pixel. Missing data also occurs due to detectable video transmission errors.

In this work, we study RMC with the extra assumption that the true data belongs to a fixed, or slowly changing, low-dimensional subspace. Assuming that the subspace changes slowly enough, this is one (but not the only) way of imposing that the resulting true data matrix is low-rank. We focus on developing an online algorithm that assumes that the observed (missing and corrupted) data comes in one matrix column at a time and we would like to estimate the subspace changes within a short delay. This problem can also be referred to as *robust subspace tracking with missing data (RST-miss)*.

Related Work. Our problem is closely related to two lines of work. In the first line of work the objective is to *recover a low-rank matrix* and in the second line of work the goal is to *track the subspace* in which the true data lies. The first line of work evolved from Matrix Completion (MC) and Robust PCA (RPCA) literature. MC aims to recover a low-rank matrix by observing only a small subset of its entries, while RPCA aims to decompose the observed matrix into a sum of low-rank and sparse matrix. MC was studied in [2–5] while provable RPCA solutions include [6–10]. Also see [11, 12] for comprehensive overviews. RMC is a generalization of both RPCA and MC since it tolerates both missing and corrupted entries. The first provable algorithms for RMC [6, 13] proposed to solve the convex relaxation of the problem and thus, were slow. Recently, two other provable non-convex approaches have been developed [8, 14] - both are projected Gradient Descent based approaches and are significantly faster. These works prove that if the maximum fraction of outliers in any row or column is bounded by $\mathcal{O}(1/r)$, one can provably recover the low-rank matrix with number of observed entries $\mathcal{O}(n \text{ poly}(r, \log n))$ entries as long as the set of observed entries is generated by i.i.d. Bernoulli model[1]. [15] makes an attempt at relaxing the identically distributed part of this assumption.

The second line of work evolves from Subspace Tracking (ST). ST has been extensively studied in both the controls' and the signal processing literature, see [12, 16–18] for comprehensive overviews. Best known existing algorithms for ST and Subspace Tracking with missing entries (ST-miss) include PAST [19, 20], PETRELS [21] and GROUSE [22–25]. Although there has been a lot of work in this space none of them provide provable guarantees in the setting of changing subspaces. The best known algorithms for Robust ST (RST) include GRASTA [26] and ReProCS [27–30] but only ReProCS provides provable guarantees. Analogous to RMC, Robust Subspace Tracking with missing data (RST-miss) is a generalization of RST and ST-miss. The best known algorithms that tackle this problem include GRASTA [26], APSM [31], and ROSETA [32]. There are no theoretical guarantees for GRASTA and ROSETA while APSM comes with only a *partial guarantee*: the result does not tell us what assumptions the algorithm inputs need to satisfy in order to ensure that the algorithm output(s) will be close to the true value(s) of the quantity of interest either at all times, or at least at certain times.

Contributions. In this paper, we develop a simple modification of a recursive projected compressive sensing (ReProCS) based algorithm, ReProCS-NORST (for Nearly Optimal RST via ReProCS) [30] to solve the RMC or RST-miss problem. We call it NORST-miss-robust. We demonstrate that under a few simple assumptions, it provably solves the RMC and RST-miss problems. Our guarantee is the *first complete guarantee* for RST-miss and, in fact, also for any online solution to RMC. Here "complete guarantee" means that, under simple assumptions on only the algorithm inputs (measurements and initialization), we show that, with high probability (w.h.p.), the output subspace estimates are close to the true

A longer version of this document [1] has been submitted to IEEE Transactions on Signal Processing.

[1] A set Ω that is randomly sampled from a universe, $\mathcal{U}$, is said to be *"i.i.d. Bernoulli with parameter ρ"* if each entry of $\mathcal{U}$ has probability ρ of being selected to belong to Ω independent of all others.

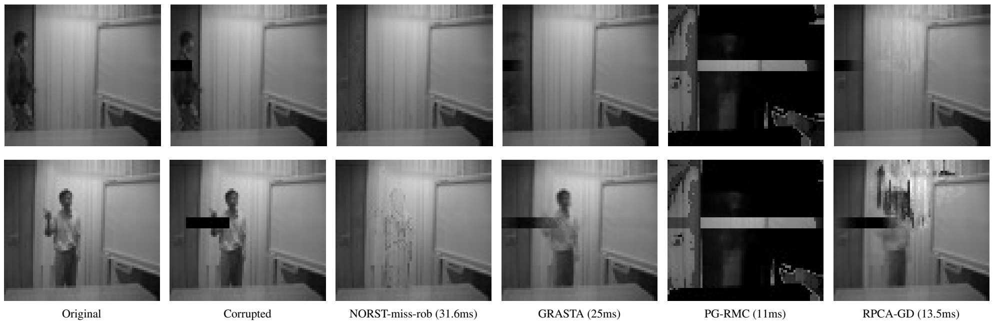

Fig. 1: Background Recovery with foreground layer, and Moving Object missing entries ($\rho = 0.98$). We show the original, observed and recovered frames at $t = 1755 + \{12, 503\}$ for the `Meeting Room` video. NORST-miss-robust exhibits artifacts, but is able to capture most of the background information, whereas, GRASTA, PG-RMC and RPCA-GD fail to obtain meaningful estimates.

data subspaces at all times. Moreover, we can obtain an ϵ-accurate subspace estimate within a short (near-optimal[2]) delay. Our result allows time-varying (piecewise-constant with time) subspaces, and can provably detect and track each changed subspace quickly. Finally, our algorithm has near-optimal memory complexity and has time-complexity equal to a rank-r vanilla SVD upto constant factors. For high-dimensional data sets memory complexity is often as important as time-complexity. Detailed experimental comparisons are provided to back up these claims, e.g., see Fig. 1.

In constrast with past work on RMC, our result does not require a probabilistic model (e.g. i.i.d. Bernoulli) on either the set of missing entries or the set of outlier entries. The tradeoff is that we allow much fewer missing entries and need the subspace changes to be slow enough compared to the minimum nonzero outlier magnitude.

Notation. $[a, b]$ refers to all integers between a and b, inclusive, $[a, b) := [a, b-1]$, and $\boldsymbol{M}_{\mathcal{T}}$ denotes a sub-matrix of $\boldsymbol{M}$ formed by columns indexed by entries in $\mathcal{T}$. $\|\cdot\|$ refers to the l_2 norm of a vector, and the spectral norm of a matrix. A basis matrix is a tall matrix with orthonormal columns. For basis matrices $\hat{\boldsymbol{P}}$, $\boldsymbol{P}$, that are used to denote two r-dimensional subspaces, we use $\mathrm{SE}(\hat{\boldsymbol{P}}, \boldsymbol{P}) = \|(\boldsymbol{I} - \hat{\boldsymbol{P}}\hat{\boldsymbol{P}}')\boldsymbol{P}\|_2$ to quantify the subspace error [33], or the sine of the largest principal angle between $\hat{\boldsymbol{P}}, \boldsymbol{P}$. We let $\hat{\boldsymbol{L}}_{t;\alpha} := [\hat{\boldsymbol{\ell}}_{t-\alpha+1}, \cdots, \hat{\boldsymbol{\ell}}_t]$. r-$SVD[\boldsymbol{M}]$ refers to the matrix of top r left singular vectors of the matrix $\boldsymbol{M}$. We reuse C, c to denote different numerical constants in each use.

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

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

where $\mathcal{P}_{\Omega_t}(\boldsymbol{z}_i) = \boldsymbol{z}_i$ if $i \in \Omega_t$ and 0 otherwise. Here $\boldsymbol{\nu}_t$ is small unstructured noise, Ω_t is the set of observed entries at time t, and $\boldsymbol{\ell}_t$ is the true data vector that lies in a fixed or slowly changing low dimensional subspace of $\mathbb{R}^n$. In other words, $\boldsymbol{\ell}_t = \boldsymbol{P}_{(t)}\boldsymbol{a}_t$ where $\boldsymbol{P}_{(t)}$ is an $n \times r$ basis matrix with $r \ll n$ and $\mathrm{SE}(\boldsymbol{P}_{(t-1)}, \boldsymbol{P}_{(t)}) \ll 1$. Finally, $\boldsymbol{g}_t$'s are sparse outliers and define $\boldsymbol{x}_t := \mathcal{P}_{\Omega_t}(\boldsymbol{g}_t)$. We use $\mathcal{T}_{\text{sparse},t}$ to denote the support of $\boldsymbol{x}_t$. This is the part of the outliers that actually corrupt our measurements, and hence we only work with $\boldsymbol{x}_t$ in the sequel. Defining[3] $\mathcal{T}_{\text{miss},t} = \Omega_t^c$ as the set of missing entries at time t, we have

$$\boldsymbol{y}_t = \mathcal{P}_{\Omega_t}(\boldsymbol{\ell}_t) + \boldsymbol{x}_t + \boldsymbol{\nu}_t = \boldsymbol{\ell}_t - \boldsymbol{I}_{\mathcal{T}_{\text{miss},t}}\boldsymbol{I}_{\mathcal{T}_{\text{miss},t}}{}'\boldsymbol{\ell}_t + \boldsymbol{x}_t + \boldsymbol{\nu}_t$$

The goal is to track $\mathrm{span}(\boldsymbol{P}_{(t)})$ and $\boldsymbol{\ell}_t$ either immediately or within a short delay.

Writing $\boldsymbol{y}_t$ as above allows us to exploit the ReProCS [27, 30] framework. This was developed originally for solving the RST problem which involves tracking $\boldsymbol{\ell}_t$ and $\boldsymbol{P}_{(t)}$ from $\boldsymbol{y}_t := \boldsymbol{\ell}_t + \boldsymbol{z}_t + \boldsymbol{\nu}_t$ where $\boldsymbol{z}_t$ is the sparse (outlier) vector. Observe that (1) can be "reduced" to a RST problem by letting $\boldsymbol{z}_t \equiv \boldsymbol{x}_t - \boldsymbol{I}_{\mathcal{T}_{\text{miss},t}}\boldsymbol{I}'_{\mathcal{T}_{\text{miss},t}}\boldsymbol{\ell}_t$.

Defining the $n \times d$ matrix $\boldsymbol{L} := [\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \ldots \boldsymbol{\ell}_d]$, with $\boldsymbol{Y}$, $\boldsymbol{Z}$, $\boldsymbol{X}$, Ω and $\boldsymbol{V}$ similarly defined, another way (low-rank matrix recovery version) to write the RMC problem using (1) is

$$\boldsymbol{Y} = \mathcal{P}_{\Omega}(\boldsymbol{L}) + \boldsymbol{X} + \boldsymbol{V} = \boldsymbol{L} + \boldsymbol{Z} + \boldsymbol{V}. \tag{2}$$

Identifiability. The above problem definition does not ensure identifiability. If $\boldsymbol{L}$ is sparse, it is impossible to recover it from a subset of its entries. Moreover, even if it is dense, it is impossible to recover $\boldsymbol{L}$ if no entries of a few rows (or columns) are observed or if a few rows (columns) are fully corrupted. Finally, if the subspace changes at every time t, the number of unknowns (nr) is more than the amount of available data at time t (n) making it impossible to recover all of them. One way to ensure subspaces' identifiability is to assume that they are piecewise constant with time, i.e., that

$$\boldsymbol{P}_{(t)} = \boldsymbol{P}_j \text{ for all } t \in [t_j, t_{j+1}), \; j = 1, 2, \ldots, J.$$

with $t_{j+1} - t_j \geq r$. Let $t_0 = 1$ and $t_{J+1} = d$. We refer to the t_j's as the subspace change times.

One way to ensure that $\boldsymbol{L}$ is not sparse is to assume that its left and right singular vectors are dense [2]. This is the well-known incoherence or denseness assumption defined as follows: *An $n \times r_P$ basis matrix $\boldsymbol{P}$ is μ-incoherent if $\max_i \|\boldsymbol{P}^{(i)}\|_2^2 \leq \mu \frac{r_P}{n}$ ($\boldsymbol{P}^{(i)}$ is i-th row of $\boldsymbol{P}$).*

Left singular vectors incoherent is nearly equivalent to imposing the assumption on the $\boldsymbol{P}_j$'s. As explained in [30, Remark 2.4], the following statistical assumption on the $\boldsymbol{a}_t$'s provides a different way of imposing right incoherence: We assume that the $\boldsymbol{a}_t$'s are element-wise bounded, mutually independent and identically distributed (i.i.d.), and have zero mean. We will refer to this as *"statistical right incoherence"*. In fact, as we see in Theorem 2.1, an assumption slightly weaker than i.i.d. suffices.

Motivated by RPCA and MC literature, one way to ensure that the missing entries (outliers) are spread out is to bound the maximum fraction of missing entries in any row and in any column. We use max-miss-frac-row (max-out-frac-row) and max-miss-frac-col (max-out-frac-col) to denote these fractions. Since NORST-miss-robust is an online approach that works on mini-batches of α frames, we actually need to bound the maximum fraction of missing entries (outliers) in any sub-matrix of $\boldsymbol{Y}$ with α consecutive columns. We denote this by max-miss-frac-row$^{\alpha}$ (max-out-frac-row$^{\alpha}$).

[2]"Near-optimal" means the required delay is order-wise within logarithmic factors of the minimum required. Since r data points are needed to define an r-dimensional subspace, the minimum is r in our setting.

[3]Here, ^{c} denotes the complement set w.r.t. $\{1, 2, \ldots, n\}$.

Algorithm 1 NORST-miss-robust. We present a simple version of the algorithm for the case where the subspace change times t_j are known. The actual algorithm that is studied in our guarantee and that is used in our experiments automatically detects subspace changes. This is provided in the long version [1].

1: **Input**: $\boldsymbol{y}_t$, $\mathcal{T}_{\text{miss},t}$, $\hat{\boldsymbol{P}}_{\text{init}} = \hat{\boldsymbol{P}}_{(t_{\text{train}})}$; **Output**: $\hat{\boldsymbol{\ell}}_t$, $\hat{\boldsymbol{P}}_{(t)}$
2: **Param**: $\omega_{supp} \leftarrow x_{\min}/2, \xi \leftarrow x_{\min}/15$ $j \leftarrow 1, k \leftarrow 1$
3: **for** $t > t_{\text{train}}$ **do**
4: $\quad \boldsymbol{\Psi} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_{(t-1)}\hat{\boldsymbol{P}}_{(t-1)}{}'$; $\tilde{\boldsymbol{y}}_t \leftarrow \boldsymbol{\Psi}\boldsymbol{y}_t$;
5: $\quad \hat{\boldsymbol{x}}_{t,cs} \leftarrow \arg\min_{\boldsymbol{x}} \left\|(\boldsymbol{x})_{\mathcal{T}_{\text{miss},t}{}^c}\right\|_1$ s.t $\|\tilde{\boldsymbol{y}}_t - \boldsymbol{\Psi}\boldsymbol{x}\| \leq \xi$.
6: $\quad \hat{\mathcal{T}}_t \leftarrow \mathcal{T}_{\text{miss},t} \cup \leftarrow \{i : \ |\hat{\boldsymbol{x}}_{t,cs}| > \omega_{supp}\}$
7: $\quad \hat{\boldsymbol{\ell}}_t \leftarrow \boldsymbol{y}_t - \boldsymbol{I}_{\hat{\mathcal{T}}_t}(\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t}{}'\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t})^{-1}\boldsymbol{\Psi}_{\hat{\mathcal{T}}_t}{}'\tilde{\boldsymbol{y}}_t$
8: $\quad$ **if** $t = t_j + u\alpha - 1$ for $u = 1, \ 2, \ \cdots$, **then**
9: $\quad\quad \hat{\boldsymbol{P}}_{j,k} \leftarrow r\text{-}SVD[\hat{\boldsymbol{L}}_{t;\alpha}], \hat{\boldsymbol{P}}_{(t)} \leftarrow \hat{\boldsymbol{P}}_{j,k}, k \leftarrow k+1.$
10: $\quad$ **else**
11: $\quad\quad \hat{\boldsymbol{P}}_{(t)} \leftarrow \hat{\boldsymbol{P}}_{(t-1)}$
12: $\quad$ **end if**
13: $\quad$ **if** $t = t_j + K\alpha - 1$ **then**
14: $\quad\quad \hat{\boldsymbol{P}}_j \leftarrow \hat{\boldsymbol{P}}_{(t)}, k \leftarrow 1, j \leftarrow j+1.$
15: $\quad$ **end if**
16: **end for**
17: **Offline**: At $t = t_j + K\alpha$ **for** $t \in [t_{j-1} + K\alpha, t_j + K\alpha - 1]$
$\hat{\boldsymbol{P}}_{(t)}^{\text{offline}} \leftarrow \text{basis}([\hat{\boldsymbol{P}}_{j-1}, \hat{\boldsymbol{P}}_j])$
$\boldsymbol{\Psi} \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_{(t)}^{\text{offline}}\hat{\boldsymbol{P}}_{(t)}^{\text{offline}\prime}$
$\hat{\boldsymbol{\ell}}_t^{\text{offline}} \leftarrow \boldsymbol{y}_t - \boldsymbol{I}_{\mathcal{T}_t}(\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{\Psi}_{\mathcal{T}_t})^{-1}\boldsymbol{\Psi}_{\mathcal{T}_t}{}'\boldsymbol{y}_t$

2. ALGORITHM AND MAIN RESULT

NORST-miss-robust. Algorithm 1 proceeds as follows. At any time t, it iterates between (a) a *Projected Compressive Sensing (CS)* step in order to estimate the outliers and missing entries, followed by (b) *Subspace Update* to update the subspace estimate $\hat{\boldsymbol{P}}_{(t)}$.

Projected CS proceeds as follows. At time t, if the previous subspace estimate, $\hat{\boldsymbol{P}}_{(t-1)}$, is an accurate enough estimate of $\boldsymbol{P}_{(t-1)}$, because of slow subspace change, projecting $\boldsymbol{y}_t$ onto the orthogonal complement of $\hat{\boldsymbol{P}}_{(t-1)}$ will nullify most of $\boldsymbol{\ell}_t$ (Line 4). Let $\boldsymbol{\Psi} := \boldsymbol{I} - \hat{\boldsymbol{P}}_{(t-1)}\hat{\boldsymbol{P}}_{(t-1)}{}'$. This means that $\|\boldsymbol{\Psi}\boldsymbol{\ell}_t\|$ will be small. Estimating $\boldsymbol{z}_t = \boldsymbol{x}_t - \boldsymbol{I}_{\mathcal{T}_{\text{miss},t}}\boldsymbol{I}_{\mathcal{T}_{\text{miss},t}}{}'\boldsymbol{\ell}_t$ from $\tilde{\boldsymbol{y}}_t := \boldsymbol{\Psi}\boldsymbol{y}_t$ is a noisy sparse recovery problem with partial support knowledge because $\boldsymbol{z}_t$ is supported on $\mathcal{T}_{\text{miss},t} \cup \mathcal{T}_{\text{sparse},t}$. We solve this (Line 5) using weighted l_1-minimization, and exploit the result of [34]. The sparse recovery problem works because the denseness assumption on $\boldsymbol{P}_{(t)}$ ensures that $\boldsymbol{\Psi}$ satisfies RIP of level $|\mathcal{T}_{\text{miss},t}| + 2|\mathcal{T}_{\text{sparse},t}|$. A hard thresholding on the above output (Line 6), and the lower bound assumption on outlier magnitudes help ensure that $\mathcal{T}_{\text{sparse},t}$ is accurately recovered. Following this, we perform a least-squares debiasing (Line 7) on $\hat{\mathcal{T}}_t = \mathcal{T}_{\text{miss},t} \cup \hat{\mathcal{T}}_{\text{sparse},t}$ to compute $\hat{\boldsymbol{z}}_t$. Finally, the true data vector is recovered by subtraction: $\hat{\boldsymbol{\ell}}_t = \boldsymbol{y}_t - \boldsymbol{I}_{\hat{\mathcal{T}}_t}{}'\hat{\boldsymbol{z}}_t$.

The $\hat{\boldsymbol{\ell}}_t$'s are used for the Subspace Update step as follows. Let $\hat{t}_j$ be the time at which the j-th subspace change is detected. Let $\hat{t}_0 = t_{\text{train}}$. This step toggles between the "update" and "detect" phases. It starts in the "update" phase which involves K r-SVD steps with the k-th r-SVD step done at $t = \hat{t}_j + k\alpha - 1$. Each such step uses the previous α estimates of $\hat{\boldsymbol{\ell}}_t$, i.e., uses $\hat{\boldsymbol{L}}_{t;\alpha}$. At $t = \hat{t}_j + K\alpha - 1 := \hat{t}_{j,fin}$, the update is complete and the algorithm enters the "detect" phase. To understand the detection strategy, assume that the previous subspace $\boldsymbol{P}_{j-1}$ has been accurately estimated by $t = \hat{t}_{j-1,fin}$ and denote it by $\hat{\boldsymbol{P}}_{j-1}$. Let $\boldsymbol{B}_t := (\boldsymbol{I} - \hat{\boldsymbol{P}}_{j-1}\hat{\boldsymbol{P}}_{j-1}{}')\hat{\boldsymbol{L}}_{t;\alpha}$. At every $t = \hat{t}_{j-1,fin} + u\alpha - 1, u = 1, 2, \ldots,$ we detect change by checking if the maximum singular value of $\boldsymbol{B}_t$ is above a pre-set threshold, $\sqrt{\omega_{evals}\alpha}$, or not. A simple version of the algorithm which assumes t_j's are known is summarized in Algorithm 1.

We initialize the algorithm using AltProj[4] [9] on the first t_{train} frames. Replacing all zeros in $\boldsymbol{Y}_{[1:t_{\text{train}}]}$ by a sufficiently large number allows AltProj obtain a "good enough" initial estimate.

Main Result. We define max-miss-frac-row$^\alpha$ as follows. For a time interval, $\mathcal{J}$, let $\gamma(\mathcal{J}) := \max_i \frac{1}{|\mathcal{J}|}\sum_{t\in\mathcal{J}} \mathbf{1}_{\{i\in\mathcal{T}_{\text{miss},t}\}}$ where $\mathbf{1}_S$ is the indicator function for statement S. Thus, $\sum_{t\in\mathcal{J}} \mathbf{1}_{\{i\in\mathcal{T}_{\text{miss},t}\}}$ counts the number of missing entries in row i of the sub-matrix $\boldsymbol{Y}_\mathcal{J}$. So $\gamma(\mathcal{J})$ is the maximum missing entry fraction in any row of $\boldsymbol{Y}_\mathcal{J}$. Let $\mathcal{J}^\alpha$ denote a time interval of duration α. Then max-miss-frac-row$^\alpha$ $:= \max_{\mathcal{J}^\alpha \subseteq [1,d]} \gamma(\mathcal{J}^\alpha)$. Observe also that max-miss-frac-col $= \max_t |\mathcal{T}_{\text{miss},t}|/n$. Define max-out-frac-row$^\alpha$, max-out-frac-col in an analogous way for outliers. Also define $x_{\min} := \min_t \min_{i\in\mathcal{T}_{\text{sparse},t}} |(\boldsymbol{x}_t)_i|$ to denote the minimum outlier magnitude. Let $\Delta := \max_j \text{SE}(\boldsymbol{P}_{j-1}, \boldsymbol{P}_j)$ denote the maximum subspace change at any t_j.

Theorem 2.1. *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* $\epsilon \leq \min(0.01, 0.03 \min_j \text{SE}(\boldsymbol{P}_{j-1}, \boldsymbol{P}_j)^2/f)$. *Let* $K := C\log(1/\epsilon)$. *If*

1. *left incoherence and statistical right incoherence:* $\boldsymbol{P}_j$*'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* $\mathcal{T}_{\text{miss},t}$*;*
2. *max-miss-frac-col* $+ 2 \cdot$ *max-out-frac-col* $\leq \frac{0.1}{\mu r}$*;*
max-miss-frac-row$^\alpha$ $+$ *max-out-frac-row*$^\alpha$ $\leq b_0 := \frac{0.001}{f^2}$*;*
3. *subspace change:*
   (a) $t_{j+1} - t_j > (K+2)\alpha = Cf^2 r \log n \log(1/\epsilon)$
   (b) $\Delta \leq 0.8$ *and* $15\sqrt{\eta r\lambda^+}(\Delta + 2\epsilon) \leq x_{\min}$
4. $\|\boldsymbol{\nu}_t\|^2 \leq cr\|\mathbb{E}[\boldsymbol{\nu}_t\boldsymbol{\nu}_t{}']\|$, $\|\mathbb{E}[\boldsymbol{\nu}_t\boldsymbol{\nu}_t{}']\| \leq c\epsilon^2\lambda^-$*,* $\boldsymbol{\nu}_t$*'s zero mean, mutually independent, and independent of* $\boldsymbol{\ell}_t$*;*
5. *initialization satisfies* $\text{SE}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}_0) \leq 0.25$
and $15\sqrt{\eta r\lambda^+}\text{SE}(\hat{\boldsymbol{P}}_{\text{init}}, \boldsymbol{P}_0) \leq x_{\min}$*;*

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

1. *subspace change is detected quickly:* $t_j \leq \hat{t}_j \leq t_j + 2\alpha$*,*
2. *the subspace recovery error satisfies*

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

3. *at all times t,* $\|\hat{\boldsymbol{\ell}}_t - \boldsymbol{\ell}_t\| \leq 1.2(\text{SE}(\hat{\boldsymbol{P}}_{(t)}, \boldsymbol{P}_{(t)}) + \epsilon)\|\boldsymbol{\ell}_t\|$.
4. *Line 17 of Algorithm 1 satisfies* $\|\hat{\boldsymbol{L}}^{\text{offline}} - \boldsymbol{L}\|_F^2 \leq \epsilon^2\|\boldsymbol{L}\|_F^2$.

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

The memory complexity is $\mathcal{O}(nr\log n\log(1/\epsilon))$ *and the time complexity is* $\mathcal{O}(ndr\log(1/\epsilon))$*.*

[4] AltProj is a popular RPCA algorithm that works based on the method of alternating projections. The first step of the algorithm is to assign all large values of the measurement to the sparse-outlier matrix.

Table 1: Comparing time, memory, and sample complexity, and assumptions. We treat the condition number and incoherence parameter μ as constants for this table. Here, $f(n) = \Omega(g(n))$ implies that there exist $k > 0$ and $n_0 > 0$ s.t for all $n > n_0$, $|f(n)| \geq k \cdot |g(n)|$

Algorithm	Sample complexity # obs. entries, m	Memory	Time	Observed entries' support	Outliers' support
NNM [6]	$\Omega(nd)$	$\mathcal{O}(nd)$	$\mathcal{O}(n^3/\sqrt{\epsilon})$	i.i.d. Bernoulli (c)	i.i.d. Bernoulli (c)
Projected GD [14]	$\Omega(nr^2 \log n^2 \log^2(1/\epsilon))$	$\mathcal{O}(nd)$	$\mathcal{O}(nr^3 \log n^2 \log^2(1/\epsilon))$	i.i.d. Bernoulli (m/nd)	bounded fraction ($\mathcal{O}(1/r)$ per row and col)
NORST-miss-rob (this work)	$\Omega(nd(1-1/r))$	$\mathcal{O}(nr \log n \log(1/\epsilon))$	$\mathcal{O}(ndr \log(1/\epsilon))$	bounded frac $\mathcal{O}(1/r)$ per row, $\mathcal{O}(1)$ per col	bounded frac. $\mathcal{O}(1/r)$ per row, $\mathcal{O}(1)$ per col
				Extra assumptions: Slow subspace change and lower bound on outlier magnitude	

The proof is similar to that of [30] and provided in the long version of this paper [1, Appendix B]. We can relax the bound on $x_{\min}$ as done in [30, Remark 2.4] to obtain the following.

Remark 2.2. *Assume that the outlier magnitudes are such that the following holds: $\boldsymbol{x}_t$ can be split as $\boldsymbol{x}_t = (\boldsymbol{x}_t)_{small} + (\boldsymbol{x}_t)_{large}$ with the two components having disjoint supports and being such that, $\|(\boldsymbol{x}_t)_{small}\| \leq b_{v,t}$ and the smallest nonzero entry of $(\boldsymbol{x}_t)_{large}$ is greater than $30b_{v,t} = 30 \cdot C \cdot (2\epsilon + \Delta)\sqrt{r\lambda^+}$ for $t \in [t_j, \hat{t}_j + \alpha)$, $30b_{v,t} = 30 \cdot C \cdot 0.3^{k-1}(2\epsilon + \Delta)\sqrt{r\lambda^+}$ for $t \in [\hat{t}_j + (k-1)\alpha, \hat{t}_j + k\alpha - 1]$, $k = 2, 2, \ldots, K$, and $30b_{v,t} = 30 \cdot C \cdot \epsilon\sqrt{r\lambda^+}$ for $t \in \hat{t}_j + K\alpha, t_{j+1})$. Then all conclusions of Theorem 2.1 hold.*

Discussion. Our result proves that NORST-miss-robust tracks time-varying subspaces to ϵ accuracy within a delay that is near-optimal under the following mild assumptions: a lower bound on most outlier magnitudes, accurate initialization of the first subspace, slow subspace change, subspace changes are piecewise constant with time (necessary for identifiability), left and statistical right incoherence assumptions hold, the fraction of missing entries in any column of $\boldsymbol{L}$ is at most $\mathcal{O}(1/r)$ while that in any row (of α-consecutive column sub-matrices of it) is at most $\mathcal{O}(1)$, fraction of outlier entries in any column and in row being $\mathcal{O}(1/r)$ and $\mathcal{O}(1)$ respectively, and the noise $\boldsymbol{\nu}_t$ is small. Observe that it estimates each subspace to ϵ accuracy within a delay of at most $Cr \log n \log(1/\epsilon)$: this is near-optimal because r is the minimum delay required to find an r-dimensional subspace even from perfect data. The memory complexity is near-optimal because nr is the memory required to output the subspace estimate and the complexity is within log factors of nr. Our result does not need probabilistic models on the set of observed entries and is able to automatically detect subspace changes quickly. On the negative side, it needs many more observed entries. We summarize a comparison of the assumptions with state-of-the-art provable algorithms in Table 1.

3. EXPERIMENTAL COMPARISONS

All time comparisons are performed on a Desktop Computer with Intel Xeon E3-1200 CPU, and 8GB RAM. The synthetic data experiment is averaged over 100 independent trials. Details regarding data generation and parameter setting are described in the long version [1]. MATLAB code for NORST-miss-robust is available at `https://github.com/vdaneshpajooh/NORST-rmc`.

Synthetic Data. In the first part of this experiment we generate the data according to (1) and set $\boldsymbol{\nu}_t = 0$. We generate the first subspace basis matrix $\boldsymbol{P}_0 \in \mathbb{R}^{n \times r}$ with $n = 1000$ and $r = 30$. We set $J = 2$, $t_1 = 4000$, $t_2 = 8000$. The basis matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are generated according to the model in [26]. The $\boldsymbol{a}_t \in \mathbb{R}^r$ (for $t = 1, \cdots, d$ and $d = 10000$) are generated independently as $(\boldsymbol{a}_t)_i \overset{\text{i.i.d}}{\sim} \text{unif}[-q_i, q_i]$ 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 of $\boldsymbol{\Lambda}$ is f and we set $f = 100$. We generate the sparse matrix, $\boldsymbol{X}$ using the Bernoulli model with $\rho_{\text{sparse}} = 0.1$ *fraction of sparse outliers*. The non-zero magnitudes of $\boldsymbol{X}$ are generated uniformly at random

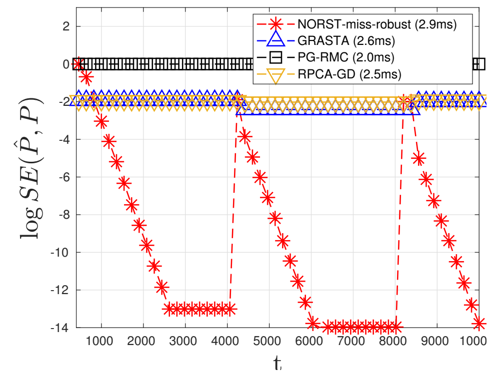

Fig. 2: Subspace error plot versus time. Here, $\rho_{\text{obs}} = 0.9$ under Moving Object model and $\rho_{\text{sparse}} = 0.1$ with Bernoulli model. Time taken per frame in milliseconds are shown in parenthesis.

between $[x_{\min}, x_{\max}]$ with $x_{\min} = 10$ and $x_{\max} = 25$. We generated the support of observed entries using the Moving Object Model [29, Model 6.19] with $\rho_{\text{obs}} = 0.9$ fraction of observed entries.

For initialization of NORST-miss-robust, in the first $t_{\text{train}} = 400$ data samples, we set $(\boldsymbol{y}_t)_i = 10$ for all $i \in \mathcal{T}_{\text{miss},t}$. The other algorithm parameters are $\alpha = 60$, $K = 33$, $\omega_{evals} = 7.8 \times 10^{-4}$, $\xi = x_{\min}/15$, and $\omega_{supp} = \boldsymbol{x}_{\min}/2 = 5$. We compare our algorithm with GRASTA, PG-RMC and RPCA-GD[5]. We use default parameters for GRASTA, PG-RMC and RPCA-GD. Observe that only NORST-miss-robust is able to obtain an accurate estimate since the missing entries are generated from the moving object model. The results are presented in Fig. 2.

Real Data. Here we consider the task of Background Recovery. We evaluate NORST-miss-robust on the `Meeting Room` video. In addition to the foreground (sparse outlier) we generate missing entries from the Moving Object model with $\rho_{\text{obs}} = 0.98$. We initialize using AltProj with tolerance 10^{-2} and 100 iterations. We set $\omega_{supp,t} = 0.9\|\boldsymbol{y}_t\|/\sqrt{n}$ using the approach of [30]. The comparison results are provided in Fig. 1. Notice that GRASTA, PG-RMC and RPCA-GD fail to accurately recover the background. Although NORST-miss-robust exhibits certain artifacts around the edges of the sparse object, it is able to capture most of the information in the background.

4. CONCLUSIONS AND OPEN QUESTIONS

In this work we proposed a provably correct online, fast, and memory-efficient algorithm for Robust Matrix Completion. We also showed that our proposed method, NORST-miss-robust, has competitive experimental performance. Two open questions for future include: (i) can the required number of observed entries be reduced (the limiting bound here is the bound on missing fractions per column); and (ii) can the lower bound on outlier magnitudes be removed?

[5]These codes are downloaded from `https://github.com/andrewssobral/lrslibrary`. We do not evaluate Nuclear Norm Minimization as RPCA-GD and PG-RMC are significantly faster [8, 14].

5. REFERENCES

[1] P. Narayanamurthy, V. Daneshpajooh, and N. Vaswani, "Subspace tracking from missing and outlier corrupted data," *arXiv preprint arXiv:1810.03051 (submitted to IEEE Trans. Sig. Proc.)*, 2018.

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

[3] R. Ge, J. D. Lee, and T. Ma, "Matrix completion has no spurious local minimum," in *Neural Information Processing Systems*, 2016, pp. 2973–2981.

[4] P. Jain and P. Netrapalli, "Fast exact matrix completion with finite samples," in *Conference on Learning Theory*, 2015, pp. 1007–1034.

[5] S. Bhojanapalli and P. Jain, "Universal matrix completion," in *International Conference on Machine Learning*, 2014, pp. 1881–1889.

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

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

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

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

[10] P. Narayanamurthy and N. Vaswani, "A fast and memory-efficient algorithm for robust pca (merop)," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4684–4688.

[11] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *Journal of Special Topics in Signal Processing*, vol. 10, no. 4, pp. 608–622, 2016.

[12] N. Vaswani and P. Narayanamurthy, "Static and dynamic robust pca and matrix completion: A review," *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1359–1379, 2018.

[13] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Trans. Inform. Theory*, vol. 59(7), pp. 4324–4337, 2013.

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

[15] O. Klopp, K. Lounici, and A. B. Tsybakov, "Robust matrix completion," *Probability Theory and Related Fields*, vol. 169, no. 1-2, pp. 523–564, 2017.

[16] P. Comon and G. 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.

[17] L. Balzano, Y. Chi, and Y. M. Lu, "Streaming pca and subspace tracking: The missing data case," *Proceedings of IEEE*, 2018.

[18] A. Gonen, D. Rosenbaum, Y. C. Eldar, and S. Shalev-Shwartz, "Subspace learning with partial information," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1821–1841, 2016.

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

[20] B. Yang, "Asymptotic convergence analysis of the projection approximation subspace tracking algorithms," *Signal Processing*, vol. 50, pp. 123–136, 1996.

[21] 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.

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

[23] L. Balzano and S. Wright, "Local convergence of an algorithm for subspace identification from partial data," *Found. Comput. Math.*, vol. 15, no. 5, 2015.

[24] D. Zhang and L. Balzano, "Global convergence of a grassmannian gradient descent algorithm for subspace estimation," in *AISTATS*, 2016.

[25] G. Ongie, D. Hong, D. Zhang, and L. Balzano, "Enhanced online subspace estimation via adaptive sensing," in *Asilomar*, 2018.

[26] 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.

[27] 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.

[28] C. Qiu and N. Vaswani, "Support predicted modified-cs for recursive robust principal components' pursuit," in *IEEE Intl. Symp. Info. Th. (ISIT)*, 2011.

[29] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust pca or robust subspace tracking," *IEEE Trans. Info. Theory*, to appear, 2018.

[30] P. Narayanamurthy and N. Vaswani, "Nearly optimal robust subspace tracking," in *ICML 2018*, 2018, pp. 3701–3709.

[31] S. Chouvardas, Y. Kopsinis, and S. Theodoridis, "Robust subspace tracking with missing entries: a set–theoretic approach," *IEEE Trans. Sig. Proc.*, vol. 63, no. 19, pp. 5060–5070, 2015.

[32] H. Mansour and X. Jiang, "A robust online subspace estimation and tracking algorithm," in *ICASSP*, 2015, pp. 4065–4069.

[33] K. Ye and L. H. Lim, "Schubert varieties and distances between subspaces of different dimensions," *SIAM Journal on Matrix Analysis and Applications*, vol. 37, no. 3, pp. 1176–1197, 2016.

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