Recommendation on a Budget: Column Space Recovery from Partially Observed Entries with Random or Active Sampling

Carolyn Kim
Stanford University

Mohsen Bayati
Stanford University

Abstract

We analyze alternating minimization for column space recovery of a partially observed, approximately low rank matrix with a growing number of columns and a fixed budget of observations per column. We prove that if the budget is greater than the rank of the matrix, column space recovery succeeds – as the number of columns grows, the estimate from alternating minimization converges to the true column space with probability tending to one. From our proof techniques, we naturally formulate an active sampling strategy for choosing entries of a column that is theoretically and empirically (on synthetic and real data) better than the commonly studied uniformly random sampling strategy.

1 Introduction

In many applications of recommendation systems, we have data in the form of an incomplete matrix, where one dimension is growing and the other dimension is fixed. For instance, in recommendation systems, there is a fixed set of potential products (rows of a matrix) to offer customers that arrive over time (columns of a matrix). Three other applications are choosing machine learning models (rows) for each new customer's dataset (columns) (Fusi et al., 2018), choosing which survey questions (rows) to ask to respondents (columns) that arrive sequentially (Zhang et al., 2019), or choosing which lab tests (rows) to order for each new patient (columns) (Huck and Lewandrowski, 2014). In these cases, there is an inherent asymmetry with respect to the dimensions in the budget: we have a budget over each column, not over each row. We could choose any machine learning model and recommend it for each dataset, or choose any survey question and give it to every user, but it is very hard to run every machine learning pipeline on an arbitrary dataset, or to give every survey question to an arbitrary respondent (indeed, in Zhang et al. (2019), users omitting too many answers was the precise motivation for their problem). Similarly, running all lab tests on one patient siginificantly exceeds the time and cost budget per patient.

In these applications, we are often interested in approximately recovering the column space of a matrix, or equivalently, the subspace spanned by the top principal components of a data matrix. This subspace would give insights as to which machine learning models tend to perform better, which questions are most informative to ask in a survey, or which lab tests would be most valuable to order.

In particular, for a matrix that has approximately low rank r, we are interested in the case where we have a fixed number k of entries that are sampled for each new column. We can then pose the following questions – is it possible to recover the column space accurately? And if we learn the column space more accurately, does this lead to better imputation of the matrix?

In this work, we show that for an approximately rank r matrix with N rows and t columns, when we have a budget of $k > r$ observations per column, we can recover the column space with probability tending to one (as t grows) using alternating minimization when samples are randomly selected. Moreover, we establish theoretically and experimentally that an active learning strategy can help learn this subspace faster. We also show experimentally that more accurate column space recovery can lead to more accurate matrix completion.

1.1 Related Works

There are two natural ways to approach column space recovery with random sampling, which leads to two areas of related work: using the empirical covariance matrix, or using matrix completion results.

Proceedings of the 23$^{\text{rd}}$International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

One approach, typically taken in the streaming PCA literature, is to to assume that columns are i.i.d. and use the empirical covariance matrix of the columns to estimate the true covariance (Lounici et al., 2014; Gonen et al., 2016; Mitliagkas et al., 2014). We can then use the column space of this estimated covariance matrix. This approach works, but it loses efficiency due to rescaling: for instance, if every entry is observed with probability p, then because each entry of the empirical covariance matrix is the product of two observed entries of the original matrix, each (off-diagonal) entry of the empirical covariance matrix is observed with probability p^2. Therefore, this approach pays a p^{-2} penalty instead of p^{-1} penalty in terms of missingness. Moreover, while matrix completion approaches can have a $\log(\epsilon^{-1})$ dependence on the desired accuracy ϵ (in the low noise regime) for sample complexity, passing through the empirical covariance matrix naturally results in an ϵ^{-2} penalty (Lounici et al., 2014; Gonen et al., 2016; Mitliagkas et al., 2014). Other works (Eftekhari et al., 2019) in the streaming PCA literature avoids covariance estimation using a least squares approach (similar to us), but do not prove convergence to the true subspace.

Another approach would be to rely on powerful results in matrix completion (See, for instance, Candès and Recht (2009); Candès and Plan (2010); Candès and Tao (2009); Koltchinskii et al. (2011); Recht (2011); Cai et al. (2010); Chatterjee et al. (2015); Jain et al. (2013); Hardt (2014); Keshavan et al. (2010a,b); Ge et al. (2016)). However, there is no straightforward way to do this. For instance, one might think one could first perform matrix completion on the partially observed matrix, and then use its singular value decomposition to recover the column space. However, for an $N \times t$ matrix with $t > N$ whose rank is r, matrix completion results typically require more than $rt \log t$ observations. Exceptions to the superlinear (in t) number of total observations (Krishnamurthy and Singh, 2013, 2014) violate our per-column budget or require a higher per-column budget for higher accuracy (Gamarnik et al., 2017). This means that in order to get the desired guarantees from the matrix completion literature, we need to observe an *increasing* number of entries per column. This is not a natural model for the budgeted learning case (there is no reason to assume that our budget increases with time) and is unnecessary, as we show in our theory. Another way to try to apply these matrix completion results is to split an $N \times t$ matrix into $N \times a$ matrices, with $a < t$, perform matrix completion on these smaller matrices (which now have enough samples), and then combine the resulting column space estimates. This might work if matrix completion were unbiased, but since the estimates tend to be the solution of a regularized problem, they tend to be biased (and bias correction is not simple (Javanmard and Montanari, 2014)).

As for active learning, there have been experimental results on active learning for matrix factorization and completion (Elahi et al., 2016; He and Cai, 2009; Kawale et al., 2015), but they rarely come with theoretical guarantees, and we are not aware of a work that gives guarantees for growing number of degrees of freedom. For instance, Kallus and Udell (2016) also consider a setting where customers are arriving with time, but their algorithm employs non-uniform sampling only for minimization of a bandit-like regret quantity, not for better estimation. As mentioned above, Krishnamurthy and Singh (2013, 2014) prove theoretical results on matrix completion with active sampling, but they violate the budget assumption by sampling some columns in their entirety. Gonen et al. (2016) prove active sampling can help, but they share the drawbacks of using the first (covariance matrix estimation) approach and their error bounds hold only in expectation, not with high probability.

Therefore, matrix completion results do not apply to our setting. However, in this work, we will leverage some of the technical components from that literature. In particular, we show theoretically that alternating minimization will consistently recover the column subspace, both for uniformly random sampling and for active sampling.

1.2 Organization

The paper is organized in the following way: We first state the notation and assumptions (Section 2), followed by our algorithms (Section 3). We then state our theoretical results (Section 4) and present our experimental results (Section 5). We conclude by mentioning ideas of the proof (Section 6) followed by a brief summary (Section 7).

2 Background

Notation For $M \in \mathbb{N}$, we use $[M]$ to denote $\{1, \dots, M\}$ and for $M' \in \mathbb{N}$, $M' \leq M$, we use $[M' : M]$ to denote $\{M', M'+1, \dots, M\}$. For a matrix $Y \in \mathbb{R}^{N \times M}$, given $\Omega \subset [N] \times [M]$, a subset of indices (typically the indices of the observed entries), we define $\mathcal{P}_\Omega(Y) \in \mathbb{R}^{N \times M}$ by setting the entries with indices not in Ω to 0:

$$(\mathcal{P}_\Omega(Y))_{ij} = \begin{cases} Y_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega. \end{cases}$$

For $\Omega \subset [N] \times [M]$, $I \subset [M]$, we denote by Ω_I the set $\{(n,m) \in \Omega \mid m \in I\}$. We take complements of these sets by $\Omega_I^C := \{(n,m) \in [N] \times I \mid (n,m) \notin \Omega_I\}$.

The singular value decomposition (SVD) of Y expresses Y as $U\Sigma V^T$, $U \in \mathbb{R}^{N\times r}, V \in \mathbb{R}^{M\times r}$, where r is the rank of Y, and the columns of U are orthornormal (known as the left singular vectors of Y), the columns of V are orthonormal (the right singular vectors of Y), and Σ is diagonal and contains the singular values. $\|\cdot\|_F$ is the Frobenius norm, given by $\|Y\|_F = \sqrt{\sum_{n=1}^N \sum_{m=1}^M Y_{nm}^2}$. We use $\|\cdot\|$ to denote the operator norm, given by $\|Y\| = \sigma_1(Y)$, where $\sigma_1(Y) \geq \ldots \geq \sigma_r(Y)$ are the singular values of Y. Throughout our paper, t will denote the total number of columns of $Y_t \in \mathbb{R}^{N\times t}$ that are available, whereas $M \leq t$ is the second dimension of an $(N \times M)$ submatrix we are considering at a particular point.

2.1 Assumptions

Our goal is to estimate the column space of an approximately low rank matrix $Y_t \in \mathbb{R}^{N\times t}$ as the number of columns of the matrix grows. This is not possible for arbitrary growing matrices Y_t. As an extreme example, if all the columns after some point are identically zero, then we will no longer be able to learn anything about the column space, which means we need to assume that $\|Y_t\|$ is "not too small". On the other hand, if $\|Y_t\|$ keeps growing too fast, we will only fit on the latest columns, which makes learning impossible, so we need $\|Y_t\|$ to be "not too large".

First, we will assume that Y_t arises from a low rank plus noise model. We will assume that the noise is actually Gaussian because we will use its rotational symmetry in the proofs. It is likely possible to relax this to more general classes of noise matrices, but we leave this for future work.

Assumption 2.1 (Low rank plus Gaussian noise). *$Y_t = \mathring{X} W_t^T + Z_t$, where $\mathring{X} \in \mathbb{R}^{N\times r}, W_t \in \mathbb{R}^{t\times r}$, and where $(Z_t)_{n,m} \overset{iid}{\sim} \mathcal{N}(0, \sigma_z^2)$.*

Next, we need to make assumptions about W_t. Due to limited space, here we present high-level versions of these assumptions. Specifically, the conditions we need are sub-Gaussian tails (Assumption A.3), a bound on how fast (and slow) the singular values of Y_t grow (Assumption A.4), and incoherence of the row space (Assumption A.6). The details of these assumptions can be found in Section A of the Appendix . An example that satisfies all these assumptions is when each column $w_m \in \mathbb{R}^r$ of W_t has entries that are distributed i.i.d. according to $\mathcal{N}(0, B)$ for some rank r covariance matrix B.

3 Algorithms

One way to view the column space of a matrix $Y \in \mathbb{R}^{N\times M}$ is to view it as the span of the top r eigenvectors of YY^T. We have $YY^T = \sum_{m=1}^M B_m$ where $B_m = y_m y_m^T$, and y_m are the columns of Y. If we sampled each entry uniformly at random with probability p, we can get an estimate of each $B_m \in \mathbb{R}^{N\times N}$ in the following way: let y'_m be the columns of $\mathcal{P}_\Omega(Y)$, and consider $B'_m = y'_m(y'_m)^T \in \mathbb{R}^{N\times N}$. For independent Bernoulli(p) sampling, if we form the matrix $D'_m := p^{-2}B'_m + (p^{-1} - p^{-2})\text{diag}(B'_m)$, we have $\mathbb{E}[D'_m] = B'_m$. So if we approximate the eigenvectors of $\sum_{m=1}^M B'_m$, we might expect them to be close to the eigenvectors of YY^T under mild assumptions. This is the approach taken by Gonen et al. (2016) and Mitliagkas et al. (2014). Indeed, under our assumptions, this will properly estimate the column subspace in expectation (Lemma 2 in Gonen et al. (2016)). If we exactly compute the eigendecomposition (which is computationally less efficient but has the best theoretical guarantees), we obtain ScaledPCA (Algorithm 4), essentially the same as POPCA of Gonen et al. (2016)), whose pseudocode is included in the Appendix. [1]

This is a nice and intuitive algorithm, but for matrix completion, it is known that methods based purely on spectral decompositions are outperformed by methods based on optimization on the Frobenius norm of recovery error $\|\mathcal{P}_\Omega(Y - \hat{Y})\|_F^2$ (such as least squares, gradient descent, or message passing) (Keshavan et al., 2012). What is worse for ScaledPCA is that because it estimates the covariance matrix first, it essentially pays a p^{-2} penalty in terms of missingness instead of a p^{-1} penalty.

In this work, we give a proof that alternating minimization (Algorithm 1) can indeed be used to recover the column subspace. Algorithm 1 performs spectral intialization followed by alternating minimization, using some of the samples ($\Omega^{(1)}$) to estimate W and the remaining samples ($\Omega^{(2)}$) to estimate X. Algorithm 1 uses two subroutines, Sample and MedianLS. MedianLS uses SmoothQR (Hardt, 2014), which is a version of QR factorization that adds noise before performing QR, which for completeness, we include in Section C of the Appendix. SmoothQR helps maintains incoherence of the estimate of W in MedianLS , and taking the median of estimates of X leads to a higher probability bound, which are useful for our theory, but not necessary in practice (Hardt, 2014).

We denote by $S \sim \text{Unif}(\mathcal{C}(N, k))$ a subset $S \subset [N]$

[1] Lounici et al. (2014) aims to estimate just the true covariance matrix, not the underlying subspace, under the setting where $t < N$.

that was sampled uniformly at random among subsets of $[N]$ of size k. In our algorithms, we assume we have enough columns to observe (e.g., for Algorithm 1, $t \geq M_{\text{init}} + sC^{\text{med}}M\lceil \log M \rceil$). C^{med} is an absolute constant that is not required as input. C_{inc} is a constant from our incoherence assumption (Assumption A.6). We use ▷ to denote comments.

Algorithm 1 ColumnSpaceEstimate

```
Input: Partially observable $Y_t \in \mathbb{R}^{N \times t}$; $k^{(1)}, k^{(2)} \in \mathbb{N}$,
    such that the total number of samples per column
    is $k^{(1)} + k^{(2)}$; $M_{\text{init}} \in \mathbb{N}$, the number of columns for
    initialization; $M \in \mathbb{N}$, the size of blocks of columns
    for least squares; $s \in \mathbb{N}$, the number of blocks;
    $\epsilon$, the desired accuracy; $a$, a boolean indicator of
    active sampling
Output: $\hat{X} \in \mathbb{R}^{N \times r}$, the column space estimate, $\Omega \subset$
    $[N] \times [t]$, the subset of observed indices
 1: Algorithm ColumnSpaceEstimate($Y_t$, $k^{(1)}$,
    $k^{(2)}$, $M_{\text{init}}$, $M$, $s$, $\epsilon$, $a$)
 2:     ▷ Spectral initialization with uniform random
          sampling
 3:     Initialize: $\Omega \leftarrow \emptyset$
 4:     for $m = 1, \ldots, M_{\text{init}}$ do
 5:         $S \sim \text{Unif}(\mathcal{C}(N, k^{(1)} + k^{(2)}))$
 6:         $\Omega \leftarrow \Omega \cup (S \times \{m\})$
 7:     end for
 8:     $\hat{X} \leftarrow$ScaledPCA $(\mathcal{P}_\Omega(Y_t), k^{(1)} + k^{(2)}, N)$
 9:     ▷ Least squares iteration
10:     $L \leftarrow C^{\text{med}}\lceil \log M \rceil$
11:     for $i = 1, \ldots, s$ do
12:         ▷ The next block of $LM$ columns to use,
              which further gets broken down into $L$
              blocks of size $M$ in MedianLS
13:         $m \leftarrow M^{\text{init}} + (i-1)LM + 1$
14:         $I \leftarrow [m : (m + LM - 1)]$
15:         $\Omega^{(1)}, \Omega^{(2)} \leftarrow$Sample$(\hat{X}, k^{(1)}, k^{(2)}, I, a)$
16:         $\hat{X} \leftarrow$ MedianLS $(\hat{X}, Y_t, \Omega^{(1)}, \Omega^{(2)},$
                                  $M$, $m$, $\epsilon)$
17:         $\Omega \leftarrow \Omega \cup \Omega^{(1)} \cup \Omega^{(2)}$
18:     end for
19:     return $\hat{X}$, $\Omega$
20: end Algorithm
```

Practical Considerations We state our algorithms in a way that is natural to prove theoretical results, which is the main goal of this paper. However, for more practical purposes, the large block size M might at first seem prohibitive to use in MedianLS . We mitigate this in the following way: first, as mentioned above, the SmoothQR step in Line 4 of MedianLS and median step in Line 11 are not necessary in practice. Therefore, given an X^{prev}, we need only to perform two linear least squares regressions (lines 2 and 9). The first regression (line 2), which fits $\tilde{W}$, can be done separately for each column. The second regression, which fits X (line 9), can be performed in an online manner. Two possible options are to perform least squares recursively (which gives exactly the same result as doing a batch linear least squares), or to do gradient descent (which is more practical). Both of these options process one column at a time (instead of processing it as a block as in Lines 15 and 16 in Algorithm 1), and lead to time and space complexity that is linear in the block size M.

Sample: Choose samples for one block of columns

```
Input: current estimate of column space $\hat{X} \in \mathbb{R}^{N \times r}$;
    $k^{(1)}, k^{(2)} \in \mathbb{N}$, such that the total number of sam-
    ples per column is $k^{(1)} + k^{(2)}$; block of columns
    $I \subset [M]$; $a$, a boolean indicator of active sampling
Output: $\Omega^{(1)}, \Omega^{(2)} \subset [N] \times I$, the samples for columns
    indexed by $I$
function Sample$(\hat{X}, k^{(1)}, k^{(2)}, I, a)$
    Initialize: $\Omega^{(1)} \leftarrow \emptyset$, $\Omega^{(2)} \leftarrow \emptyset$
    for $m \in I$ do
        ▷ Choose each slice of $\Omega^{(1)}, \Omega^{(2)}$
        if $a$ then
            ▷ Use Equation (1) for active sampling
            $S^{(1)} \leftarrow \Omega^*(\hat{X}; k^{(1)}) \subset [N]$
        else
            $S^{(1)} \sim \text{Unif}(\mathcal{C}(N, k^{(1)}))$
        end if
        $S^{(2)} \sim \text{Unif}(\mathcal{C}(N, k^{(2)}))$
        ▷ Add the slices to $\Omega^{(1)}, \Omega^{(2)}$
        $\Omega^{(1)} \leftarrow \Omega^{(1)} \cup (S^{(1)} \times \{m\})$
        $\Omega^{(2)} \leftarrow \Omega^{(2)} \cup (S^{(2)} \times \{m\})$
    end for
    return $\Omega^{(1)}, \Omega^{(2)}$
end function
```

Active Sampling Our proof naturally leads to an active sampling strategy that can help subspace recovery, as confirmed in our experiments. Each iteration of fitting a $\tilde{W}$ (Line 3 of MedianLS) is a linear least squares regression, whose estimation error decreases as the minimum singular value of the design matrix increases. Therefore, a good candidate strategy for Sample is to choose the rows of X^{prev} to maximize the minimum singular value of the induced submatrix. More precisely, for $S = \{s_1, \ldots, s_k\} \subset [N]$, we define $\mathcal{Q}_S$ as the operator that projects the $N \times r$ matrix to a $k \times r$ matrix specified by $[\mathcal{Q}_S(X)]_{ij} = X_{s_i,j}$. (The objective in Equation (1) is invariant to the ordering chosen on S.) Given an estimate $\hat{X}$, our active sampling chooses

$$\Omega^*(X; k^{(1)}) = \underset{S \subset [N], |S| = k^{(1)}}{\text{argmax}} \quad \sigma_r(\mathcal{Q}_S(X)), \qquad (1)$$

Median least squares

```
Input: Prior estimate $X^{\text{prev}} \in \mathbb{R}^{N\times r}$; Partially observable $Y \in \mathbb{R}^{N\times t}$ and $\Omega^{(1)}, \Omega^{(2)} \in [N]\times[M]$ such
    that $\Omega^{(1)}(Y)$ and $\Omega^{(2)}(Y)$ are observed; $M \in \mathbb{N}$, the block size to subdivide into for the median
    step; $m \in \mathbb{N}$, the beginning index of block of columns of size $C^{\text{med}}M\lceil\log M\rceil$; $\epsilon$, the desired accuracy
Output: $X \in \mathbb{R}^{N\times r}$, a column space estimate
 1: function MedianLS($X^{\text{prev}}, Y, \Omega^{(1)}, \Omega^{(2)}$, $M$, $m$, $\epsilon$)
 2:     $\tilde{W} \leftarrow \operatorname*{argmin}_{W' \in \mathbb{R}^{M\times r}} \|\mathcal{P}_{\Omega^{(1)}}(Y - X^{\text{prev}}(W')^T)\|_F^2$
 3:     ▷ QR factorization with added noise for incoherence
 4:     $\hat{W} \leftarrow$ SmoothQR $(\tilde{W}, \sigma_r(\mathring{X})\epsilon, C_{\text{inc}}\log M)$
 5:     $L \leftarrow C^{\text{med}}\lceil\log M\rceil$
 6:     for $i = 1, \ldots, L$ do
 7:         ▷ Get the next block of $M$ columns to use for median
 8:         $J_i \leftarrow [(m + (i-1)M) : (m + iM - 1)]$
 9:         $\tilde{X}^{(i)} \leftarrow \operatorname*{argmin}_{X' \in \mathbb{R}^{N\times r}} \|\mathcal{P}_{\Omega^{(2)}_{J_i}}(Y - X'(\hat{W})^T)\|_F^2$
10:     end for
11:     $\tilde{X} \leftarrow$ elementwise median of $\{\tilde{X}^{(1)}, \ldots, \tilde{X}^{(L)}\}$
12:     $X \leftarrow$ Orthonormal basis of column space of $\tilde{X}$
13:     return $X$
14: end function
```

as $S^{(1)} \subset [N]$. We will need other samples of rows of Y to estimate X from this estimated $\hat{W}$, and we choose these samples randomly, so we can get equal informations about every row of X, i.e., $S^{(2)}$ is chosen uniformly at random.

4 Theoretical Results

Budget per column In the following theorems, we will assume that $k^{(1)} \geq r$, and $k^{(2)} \geq 1$. We need $k^{(1)}$ to be at least r because we observe $k^{(1)}$ entries per column for Line 2 of MedianLS . However, $k^{(2)}$ need not be as large because as the number of columns tends to infinity, we will observe at least r entries in each row. Therefore, the total number of required samples is only $r + 1$ per column. But we do not recommend setting $k^{(2)}$ as low as 1 in practice, especially without sample splitting.

Subspace Recovery Metric For our theorem statements, we let $U \in \mathbb{R}^{N\times r}$ be the matrix whose orthonormal columns are the left singular vectors of $\mathring{X}$. In general, when we compute the SVD of $\mathring{X}W_t = U_t\Sigma_t V_t^T$, the resulting U_t might not contain the same singular vectors as U, but they span the same subspace. We use a distance measure on subspaces that does not depend on such representations, namely the largest principal angle between subspaces. This can be defined for two matrices with orthonormal columns $U, X \in \mathbb{R}^{N\times r}$ by $\sin\theta(X, U) = \|(I_N - XX^T)U\|$ (Zhu and Knyazev, 2013). Note $\sin\theta(U, UO) = 0$ for any orthogonal matrix $O \in \mathbb{R}^{r\times r}$.

Initialization The initialization conditions are quite stringent in theory, but in practice, as has been empirically[2] shown in other optimization approaches, only mild initialization can suffice. This is consistent with our own experiments in Section 5.

Proofs of all theorems are deferred to the extended version of this paper (Kim and Bayati, 2019). For ease of notation, we define $q^{(1)} := \frac{k^{(1)}-r+1}{r(N-k^{(1)})+k^{(1)}-r+1}$. Note that $\frac{1}{rN} \leq q^{(1)} \leq \frac{k^{(1)}}{r(N-k^{(1)})}$, and that $\frac{k^{(1}}{q^{(1)}}$ is a decreasing quantity with respect to $k^{(1)}$. In order to simplify our bounds a little, we will additionally assume that $k^{(1)} \leq \frac{N}{2}$, which implies that $q^{(1)} \leq \frac{2k^{(1)}}{rN}$.

4.1 Active Sampling

Noise in observations presents an obstacle to recovering the column space, and if the noise variance is too large compared to the r-th singular value of $\mathring{X}$, then it can drown out this 'signal' in the noise when performing alternating minimization. Therefore, we impose Assumption 4.1 or 4.5 to ensure that we have enough signal.

Assumption 4.1 (Size of Noise for Active Sampling)**.** $\sigma_Z \leq \frac{1}{48}\frac{\sqrt{q^{(1)}}}{\sqrt{k^{(1)}}}\sigma_r(\mathring{X})$.

There are two factors that influence the rate of convergence. One factor is that we only have partial observations. The other factor is that we have noise in our observations. When σ_Z is small compared to the desired accuracy ϵ,

$$\sigma_z\sqrt{k^{(1)}} \leq \epsilon\sigma_1(\mathring{X})\sqrt{r}, \tag{2}$$

the effect of having only partial observations dominates. For instance, this is always true when observations do not contain noise. When σ_Z is large compared to the desired accuracy ϵ,

$$\sigma_z\sqrt{k^{(1)}} \geq \epsilon\sigma_1(\mathring{X})\sqrt{r}, \tag{3}$$

the effect of noise dominates. Therefore, we prove different convergence rates for each regime.

Theorem 4.2 (Noisy observations, active sampling, for small σ_Z/ϵ)**.** *Suppose Assumptions 2.1, 4.1, A.3,*

[2] For much higher sampling complexity and Bernoulli samples, it has been shown theoretically by Ge et al. (2016) and Ge et al. (2017).

A.4, A.6 hold, $N/2 \geq k^{(1)} \geq r, k^{(2)} \geq 1$, $1 \geq \sigma_r(\mathring{X})\epsilon \geq e^{-M}M$, and Equation (2) *holds. Then there exist constants $C^{\text{init}}_{4.2}$, $C^{\text{iter}}_{4.2}$, $C^{\text{prob}}_{4.2}$ such that for all $\epsilon > 0$, if we initialize with M_{init} columns, where*

$$M_{\text{init}} \geq C^{\text{init}}_{4.2} \tfrac{\sigma_1(\mathring{X})^6 N^2 (\log M_{\text{init}})^3 r^2}{\sigma_r(\mathring{X})^6 (k^{(1)}+k^{(2)})^2 q^{(1)}}, \tag{4}$$

and we use s blocks, where

$$\text{s} \geq \log\left(\tfrac{\sigma_r(\mathring{X})\sqrt{q^{(1)}}}{48\sqrt{r}\epsilon}\right), \tag{5}$$

and each block has size M, with

$$M \geq C^{\text{iter}}_{4.2} \tfrac{\sigma_1(\mathring{X})^6 r^3 N (\log M)^2}{\sigma_r(\mathring{X})^6 k^{(2)} q^{(1)}} + \log\left(\tfrac{1}{\epsilon}\right), \tag{6}$$

then ColumnSpaceEstimate*(Y_t, $k^{(1)}, k^{(2)}$, M_{init}, M, s, ϵ, True) returns an $\hat{X}$ such that* $\sin\theta(U, \hat{X}) \leq \epsilon$ *with probability at least* $1 - 2M_{\text{init}}^{-2} - C^{\text{prob}}_{4.2} sM^{-2}$.

Whenever Theorem 4.2 holds, the sample complexity grows only logarithmically with ϵ^{-1}, which is a feature of a matrix completion approach (versus a spectral approach, which always has a dependence of ϵ^{-2}) in the small σ_Z/ϵ regime.

When the σ_Z is large compared to the desired accuracy ϵ, we can get a σ_Z-dependent bound, with a ϵ^{-2} dependence on desired accuracy ϵ. The initialization step for this regime consists of Algorithm 1 instead of a spectral initialization. The full pseudocode for DoubleColumnSpaceEstimate (Algorithm 2) can be found in the Appendix.

Theorem 4.3 (Noisy observations, active sampling, for large $\frac{\sigma_Z}{\epsilon}$)**.** *Suppose Assumptions 2.1, 4.1,A.4, A.6 hold, $N/2 \geq k^{(1)} \geq r, k^{(2)} \geq 1$, $1 \geq \sigma_r(\mathring{X})\epsilon \geq e^{-M}M$. Let ϵ satisfy equation* (3)*. Then there exist constants $C^{\text{init}}_{4.3}, C^{\text{iter}}_{4.3}, C^{\text{prob}}_{4.3}$ such that for every $\epsilon > 0$, if we initialize with M_{init} columns, where*

$$M_{\text{init}} \geq C^{\text{init}}_{4.3} \tfrac{\sigma_1(\mathring{X})^6 N^2 (\log M_{\text{init}})^3 r^2}{\sigma_r(\mathring{X})^6 (k^{(1)}+k^{(2)})^2 q^{(1)}}$$

and perform alternating minimization with $s_1 \geq \log\left(\frac{\sigma_1(\mathring{X})\sigma_r(\mathring{X})\sqrt{q^{(1)}}}{48\sigma_Z\sqrt{k^{(1)}}}\right)$ blocks of size

$$M_1 \geq C^{\text{iter}}_{4.2} \tfrac{\sigma_1(\mathring{X})^6 r^3 N (\log M)^2}{\sigma_r(\mathring{X})^6 k^{(2)} q^{(1)}} + \log\left(\tfrac{1}{\epsilon}\right),$$

followed by alternating minimization with $s_2 = 1$ block of size

$$M_2 \geq C^{\text{iter}}_{4.3} \tfrac{r^2 \sigma_Z^2 \sigma_1(\mathring{X})^4 N k^{(1)} (\log M)^2}{\sigma_r(\mathring{X})^6 k^{(2)} q^{(1)} \epsilon^2} + \log\left(\tfrac{1}{\epsilon}\right),$$

then DoubleColumnSpaceEstimate*(Y_t, $k^{(1)}$, $k^{(2)}$, M_{init}, M_1, M_2, s_1, s_2, ϵ, True) returns an $\hat{X}$ such that* $\sin\theta(U, \hat{X}) \leq \epsilon$ *with probability at least* $1 - 2M_{\text{init}}^{-2} - C^{\text{prob}}_{4.2} s_1 M_1^{-2} - C^{\text{prob}}_{4.3} M_2^{-2}$.

Comparison with ScaledPCA We compare with the theoretical results from using the ScaledPCA approach with Proposition 3 from Lounici et al. (2014), as Gonen et al. (2016) prove bounds for a quantity ($\langle \hat{\Pi} - \Pi, -C \rangle$ in their notation) that is weaker (i.e., $\sin\theta(\hat{X}, U) \leq \epsilon$ implies their quantity is less than ϵ, but not vice versa), and they only prove bounds in expectation. For simplicity, we will omit dependence on the condition number (assume $\sigma_1(\mathring{X}) = \sigma_r(\mathring{X}) = 1$) and assume that $k^{(1)} = k^{(2)} =: k$. When σ_Z/ϵ is small (Equation (2)), Theorem 4.2's logarithmic dependence on ϵ^{-1} is better than the ϵ^{-2} dependence of Lounici et al. (2014), but the dependence on r and k is worse, by r^3k. When σ_Z/ϵ is large (Equation (3), Theorem 4.3), our sample complexity needs $\tilde{O}(rk/N)$ as many samples as Lounici et al. (2014), which can be fairly small.

4.2 Uniformly random sampling

When we use random sampling, there is a chance per column that we might choose a "bad" subset, which is small with respect to N, but does not change with respect to M. Since we need to avoid "bad" subsets for all M columns, in the regime of $M \gg N$, this would give us an unacceptable probability of failure in theory, though in practice, this probably does not occur. Therefore, we assume that the true $\mathring{X}$ has no "bad" subsets and use a longer initialization period to ensure that our $\hat{X}$ also has no "bad" subsets. When $\mathring{X} \in \mathbb{R}^{N\times r}$ has rank r (which is true by Assumption A.4), the assumption about the absence of "bad" subsets is equivalent to the k-isomeric condition by Liu et al. (2017).

Definition 4.4 (k-isomeric (Liu et al., 2017))**.** *A matrix $X \in \mathbb{R}^{N\times r}$ is called k-*isomeric *if and only if any k rows of M can linearly represent all rows in X.*

We define the smallest singular value of any $k^{(1)}$ rows of a matrix $X \in \mathbb{R}^{N\times r}$, which is the opposite of the desired criterion in Equation (1).

$$\sigma_*(X; k^{(1)}) := \min_{S\subset[N],|S|=k} \sigma_r(\mathcal{Q}_S(\mathring{X})) \tag{7}$$

Assuming that U has rank r, if $\mathring{X}$ is $k^{(1)}$-isomeric, $\sigma_*(U; k^{(1)}) > 0$.

We note that every $N \times r$ matrix X with orthogonal columns has $\sigma_*(X; k^{(1)}) \leq \sqrt{p^{(1)}}$ (Kim and Bayati, 2019), and in fact, $\sigma_*(X; k^{(1)})$ could be arbitrarily small. For random sampling, $\sigma_*(U; k^{(1)})$ will play (up to a constant term) the same role as $\sqrt{q^{(1)}}$ in active sampling, for instance, in the bound on the noise variance.

Assumption 4.5 (Size of Noise for Random Sampling)**.** $\sigma_Z \leq \frac{1}{96}\frac{\sigma_*(\mathring{X};k^{(1)})}{\sqrt{k^{(1)}}}\sigma_r(\mathring{X})$.

The difference in sampling complexity in active versus random sampling is the difference between $(\sigma_*(U;k^{(1)}))^2$ and $q^{(1)}$. Theorems 4.2 and 4.3 still hold with exactly the same proof if we replace $q^{(1)}$ with $(\max_{|S|=k^{(1)}} \sigma_r(U_S))^2$. With this replacement, the corresponding bound for the active learning case will always be better than the bound for the noisy case. For instance, because there is, in general, no lower bound for $\sigma_*(U;k^{(1)})$, we cannot give an upper bound on the initialization step of random sampling that holds independent of $\mathring{X}$, which is something we *can* do in the case of active sampling. The full statements and proofs for the theorems for the uniformly random sampling case (Theorems B.1 and B.2) can be found in the Appendix.

5 Experiments

Synthetic data For the simulated data experiments, we use the model from Assumption 2.1 with i.i.d. Gaussian columns. That is, for each simulation, we generate a fixed $\mathring{X} \in \mathbb{R}^{N\times r}$, and we generate the t-th column by $\mathring{X}w_t + z_t$, where $w_t, z_t \in \mathbb{R}^{r\times 1}$ and $w_t \sim \mathcal{N}(0, I_r)$, $z_t \sim \mathcal{N}(0, \sigma_Z^2 I_r)$. Since we do not require $\mathring{X}$ to be incoherent (which would result from light tailed distributions), we use a heavy tailed distribution (specifically the standard Cauchy distribution) to generate each entry of $\mathring{X}$ independently. We set $\sigma_Z^2 = 0.1$, $N = 50$, and $r = 6$.

MIMIC data For our real data experiments, we use the MIMIC II dataset, which contains data for ICU visits at Beth Isreael Deaconess Medical Center in Boston, Massachusetts between 2001 and 2008 (Lee et al., 2011). We focused on patients aged 18-89 (inclusive) who were having their first ICU visit, and who stayed in the ICU for at least 3 days. For these patients (columns), we took 1269 features which mostly include lab test results. Because the data has many missing entries, we restricted the data to those columns and rows that had less than 50% missing entries, which led to 115 covariates (rows) and 14584 patients (columns). Then, for each run, we randomly chose a submatrix of $N = 50$ covariates and $t = 5100$ patients, and we use $r = 6$ as in the simulated data. To evaluate column space recovery , we estimated a "ground truth" $\mathring{X}$ using SVD on our data, with missing values replaced by zeros. However, when evaluating Y_t recovery, we only measure error on the non-missing values (i.e., those that were present in the data, which is a strict superset of those that were observed by the algorithms).

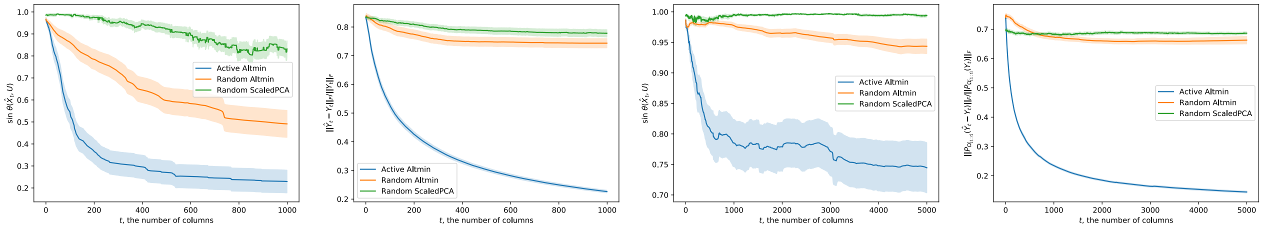
Approximately active greedy sampling We choose a fixed number $k = k^{(1)} + k^{(2)}$ to sample per column. For active sampling, we set $k^{(1)} = k^{(2)} = 6$, and for random sampling, we set $k = 12$, so that both strategies observe the same number of samples per column. Ideally, our active sampling method would choose the subset $S^{(1)}$ of size $k^{(1)}$ that satisfies Equation (1). However, since exhaustive search is computationally infeasible, we use an efficient method that approximates this optimization, namely, Algorithm 1 from Avron and Boutsidis (2013) . This algorithm produces an $S^{(1)}$ such that $\|(\mathcal{Q}_{S^{(1)}}(X^{\text{prev}})^T\mathcal{Q}_{S^{(1)}}(X^{\text{prev}}))^{-1}\mathcal{Q}_{S^{(1)}}(X^{\text{prev}})^T\| \leq 1/\sqrt{\tilde{q}^{(1)}}$, where $\tilde{q}^{(1)} = \frac{k^{(1)}-r+1}{r(N-r+1)}$. $\tilde{q}^{(1)}$ is greater than $q^{(1)}$, but has a similar behavior as $q^{(1)}$ for small $k^{(1)}$. Analogues of Theorems 4.3 and 4.2, with $q^{(1)}$ replaced by $\tilde{q}^{(1)}$, hold when we use this approximation algorithm for active sampling.

Deviation from theoretical assumptions Our recovery methods operate in a more practical setting than our theory requires. For alternating minimization, the initialization uses much fewer columns than our theorems require, we do not do sample splitting, we do not fix the time horizon beforehand, and we update $\hat{X}$ as we partially observe each column. This continual updating means that even if we chose $S^{(1)}$ at time t_0 such that $\mathcal{Q}_{S^{(1)}}(\hat{X}_{t_0})$ was large, when we use it at some timestep $t_1 > t_0$, $\mathcal{Q}_{S^{(1)}}(\hat{X}_{t_1})$ may not be large. We also skip the SmoothQR and Median steps and add L2 regularization with $\lambda = 0.05$ for stabilization.

Matrix recovery In many cases, the reason that we care about recovering subspaces accurately is so that we can recover the original matrix Y_t accurately. Therefore, we also measure matrix recovery. Given an estimate of the column subspace $\hat{X}$, the corresponding estimate $\hat{Y}_t$ is computed by imputing the missing entries by taking the best regularized least-squares fit over the observed entries: $\mathcal{P}_{\Omega^C_{[1:t]}}(\hat{Y}_t) = \mathcal{P}_{\Omega^C_{[1:t]}}(\hat{X}\beta^*)$, where $\beta^* = \operatorname*{argmin}_\beta \|\mathcal{P}_{\Omega_{[1:t]}}(\hat{X}\beta - Y_t)\|_F + 0.05\|\beta\|_F^2$. The algorithms do not have to fit the entries that it has observed, i.e., $\mathcal{P}_{\Omega_{[1:t]}}(\hat{Y}_t) = \mathcal{P}_{\Omega_{[1:t]}}(Y_t)$.

5.1 Results

Figure 1 shows the results of our simulations, averaged over 50 runs. Our active sampling method samples $k^{(1)}$ entries as described above (approximately active greedy sampling) and $k^{(2)}$ samples uniformly at random. We compare three methods: ScaledPCA (green), alternating minimization with uniformly random sampling (orange), and alternating minimization with active sampling (blue). We denote by $\hat{X}_t$ and $\hat{Y}_t$ the estimates of X and Y after observing t (total) columns. We perform the initialization step with 100 columns, and plot the error as additional columns are observed, for 1000 additional columns for the simulated data and 5000 additional columns for the MIMIC II data. We indicate standard error through shad-



(a) Simulated data, X recovery

(b) Simulated data, Y recovery

(c) MIMIC II data, X recovery

(d) MIMIC II data, Y recovery

Figure 1: Error versus number of columns t

ing. In Figures 1a and 1c, the error is the sine of the largest principal angle between two subspaces, as discussed in Section 4, and in Figures 1b and 1d, we use the normalized matrix recovery error, which is given by $\frac{\|\hat{Y}_t - Y_t\|_F}{\|Y_t\|_F}$, for the simulated data. Since we do not know all the entries of the MIMIC II dataset, we use $\frac{\|\mathcal{P}_{\Omega'_{[1:t]}}(\hat{Y}_t - Y_t)\|_F}{\|\mathcal{P}_{\Omega'_{[1:t]}}(Y_t)\|_F}$, where Ω' consists of the entries for which we have ground truth in the dataset (many of which were not observed by the algorithms).

Column space recovery Figures 1a and 1c show that alternating minimization (both random and active sampling) recovers the column space more accurately than ScaledPCA . Furthermore, when using alternating minimization, using active samples results in a lower column space recovery error than using uniformly random samples.

Matrix recovery In Figures 1b and 1d, we can see that when algorithms have more accurate column space estimates, the corresponding matrix estimate $\hat{Y}_t$ also tends to be more accurate. In Figure 1d, for the first few hundred columns, alternating minimization with random sampling has a less accurate matrix estimate $\hat{Y}_t$ than ScaledPCA . However, this is only when alternating minimization with random sampling has a poor column space estimate (though still slightly better than that of ScaledPCA). Moreover, the relative performance of alternating minimization with random sampling improves (both for matrix and column space recovery) as the number of observed columns grows, which is the setting of our theoretical results. Also, note that alternating minimization with active sampling always performs better than ScaledPCA .

6 Ideas of the Proof

Each iteration of alternating minimization involves optimizing $\hat{W} \in \mathbb{R}^{M\times r}$ given a fixed $\hat{X}^{\text{prev}} \in \mathbb{R}^{N\times r}$, and then optimizing $\hat{X}$ given this $\hat{W}$.

Jain et al. (2013) and Hardt (2014) argue that each minimization step is similar to performing a step in in the power method (e.g., finding the top eigenvector of a symmetric matrix A by setting $x_{t+1} = Ax_t/\|Ax_t\|_F$). In their setting, $\tan\theta(\hat{W}, V) \leq \tan\theta(\hat{X}^{\text{prev}}, U)$ and $\tan\theta(\hat{X}, U) \leq \tan\theta(\hat{W}, V)$, leading to successively better estimation, $\tan\theta(\hat{X}, U) \leq \tan\theta(\hat{X}^{\text{prev}}, U)$, with each iteration. (Here, U and W represent the row subspace and column space, respectively, of the de-noised version of Y.)

In our setting, because of the asymmetry between N and M, $\tan\theta(\hat{W}, V) \leq \tan\theta(\hat{X}^{\text{prev}}, U)$ no longer holds. However, it remains true that $\tan\theta(\hat{X}, U) \leq \tan\theta(\hat{W}, V)$. Furthermore, it turns out that by adjusting the block size M appropriately, we can make this decrease be large enough to compensate for the increase from $\tan\theta(\hat{X}^{\text{prev}}, U)$ to $\tan\theta(\hat{W}, V)$. In a way, this is in the spirit of averaging multiple estimates of the column subspace, by first passing through $\hat{W}$, and collecting information from enough columns of $\hat{W}$ to gain a more accurate estimate.

In the small σ_z/ϵ regime, this decrease from $\tan\theta(\hat{X}, U)$ to $\tan\theta(\hat{X}^{\text{prev}}, U)$ is actually multiplicative, leading to exponential convergence in the number of iterations.

7 Conclusion

In this work, we proved that an alternating minimization approach to estimating the column subspace of a partially observed matrix succeeds – as the number of columns grows, we can estimate the column space to any given accuracy with probability tending to 1. We showed theoretically and experimentally that this approach works better than the naive one that performs PCA on the elementwise rescaled empirical covariance matrix. We also showed that using some number $k^{(1)} \geq r$ of actively chosen samples in addition to random samples outperforms random sampling.

References

Haim Avron and Christos Boutsidis. Faster subset selection for matrices and applications. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1464–1499, 2013.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4): 1956–1982, 2010.

Emmanuel J Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6): 925–936, 2010.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *arXiv preprint arXiv:0903.1476*, 2009.

Sourav Chatterjee et al. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.

Armin Eftekhari, Gregory Ongie, Laura Balzano, and Michael B Wakin. Streaming principal component analysis from incomplete data. *Journal of Machine Learning Research*, 20(86):1–62, 2019.

Mehdi Elahi, Francesco Ricci, and Neil Rubens. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20:29–50, 2016.

Nicolo Fusi, Rishit Sheth, and Melih Elibol. Probabilistic matrix factorization for automated machine learning. In *Advances in Neural Information Processing Systems*, pages 3348–3357, 2018.

David Gamarnik, Quan Li, and Hongyi Zhang. Matrix completion from $o(n)$ samples in linear time. *arXiv preprint arXiv:1702.02267*, 2017.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1233–1242. JMLR. org, 2017.

Alon Gonen, Dan Rosenbaum, Yonina C Eldar, and Shai Shalev-Shwartz. Subspace learning with partial information. *The Journal of Machine Learning Research*, 17(1):1821–1841, 2016.

Moritz Hardt. Understanding alternating minimization for matrix completion. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 651–660. IEEE, 2014.

Xiaofei He and Deng Cai. Active subspace learning. In *2009 IEEE 12th International Conference on Computer Vision*, pages 911–916. IEEE, 2009.

Amelia Huck and Kent Lewandrowski. Utilization management in the clinical laboratory: an introduction and overview of the literature. *Clinica Chimica Acta*, 427:111–117, 2014.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

Nathan Kallus and Madeleine Udell. Dynamic assortment personalization in high dimensions. *arXiv preprint arXiv:1610.05604*, 2016.

Jaya Kawale, Hung H Bui, Branislav Kveton, Long Tran-Thanh, and Sanjay Chawla. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in neural information processing systems*, pages 1297–1305, 2015.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE transactions on information theory*, 56 (6):2980–2998, 2010a.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11 (Jul):2057–2078, 2010b.

Raghunandan Hulikal Keshavan et al. *Efficient algorithms for collaborative filtering*. PhD thesis, Stanford University, 2012.

Carolyn Kim and Mohsen Bayati. Recommendation on a budget: Column space recovery from partially observed entries with random or active sampling. *arXiv preprint arXiv:2002.11589*, 2019.

Vladimir Koltchinskii, Karim Lounici, Alexandre B Tsybakov, et al. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.

Akshay Krishnamurthy and Aarti Singh. Low-rank matrix and tensor completion via adaptive sampling. In *Advances in Neural Information Processing Systems*, pages 836–844, 2013.

Akshay Krishnamurthy and Aarti Singh. On the power of adaptivity in matrix completion and approximation. *arXiv preprint arXiv:1407.3619*, 2014.

Joon Lee, Daniel J Scott, Mauricio Villarroel, Gari D Clifford, Mohammed Saeed, and Roger G Mark. Open-access mimic-ii database for intensive care research. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 8315–8318. IEEE, 2011.

Guangcan Liu, Qingshan Liu, and Xiaotong Yuan. A new theory for matrix completion. In *Advances in Neural Information Processing Systems*, pages 785–794. 2017.

Karim Lounici et al. High-dimensional covariance matrix estimation with missing observations. *Bernoulli*, 20(3):1029–1058, 2014.

Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Streaming pca with many missing entries. *Preprint*, 2014.

Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12 (Dec):3413–3430, 2011.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Chelsea Zhang, Sean J Taylor, Curtiss Cobb, and Jasjeet Sekhon. Active matrix factorization for surveys. *arXiv preprint arXiv:1902.07634*, 2019.

Peizhen Zhu and Andrew V Knyazev. Angles between subspaces and their tangents. *Journal of Numerical Mathematics*, 21(4):325–340, 2013.