Column Subset Selection with Missing Data via Active Sampling

Yining Wang
Machine Learning Department
Carnegie Mellon University

Aarti Singh
Machine Learning Department
Carnegie Mellon University

Abstract

Column subset selection of massive data matrices has found numerous applications in real-world data systems. In this paper, we propose and analyze two sampling based algorithms for column subset selection without access to the complete input matrix. To our knowledge, these are the first algorithms for column subset selection with missing data that are provably correct. The proposed methods work for row/column coherent matrices by employing the idea of adaptive sampling. Furthermore, when the input matrix has a noisy low-rank structure, one algorithm enjoys a relative error bound.

1 INTRODUCTION

Given a matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$, the *column subset selection* problem aims to find s exact columns in $\mathbf{M}$ that capture as much of $\mathbf{M}$ as possible. More specifically, we want to select s columns of $\mathbf{M}$ to form a "compressed" matrix $\mathbf{C} \in \mathbb{R}^{n_1 \times s}$ to minimize the norm of the following residue

$$\min_{\mathbf{X} \in \mathbb{R}^{s \times n_2}} \|\mathbf{M} - \mathbf{C}\mathbf{X}\|_\xi = \|\mathbf{M} - \mathcal{P}_C(\mathbf{M})\|_\xi, \quad (1)$$

where $\mathcal{P}_C(\mathbf{M}) = \mathbf{C}\mathbf{C}^\dagger\mathbf{M}$ [1] is the projection of $\mathbf{M}$ onto the selected column subspace and $\xi = 2$ or F denotes the spectral or Frobenious norm. To evaluate the performance of column subset selection, one compares the residue norm (reconstruction error) defined in Eq. (1) with $\|\mathbf{M} - \mathbf{M}_k\|_\xi$, where $\mathbf{M}_k$ is the best rank-k approximation of $\mathbf{M}$. [2] Two forms of error guarantee are common: additive error guarantee in Eq. (2) and relative error guarantee in Eq. (3), with $0 < \varepsilon < 1$ and $c > 1$ (ideally $c = 1 + \epsilon$).

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi \leq \|\mathbf{M} - \mathbf{M}_k\|_\xi + \varepsilon\|\mathbf{M}\|_\xi; \quad (2)$$
$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi \leq c\|\mathbf{M} - \mathbf{M}_k\|_\xi. \quad (3)$$

In general, relative error bound is much more appreciated because $\|\mathbf{M}\|_\xi$ is usually large in practice. In addition, when $\mathbf{M}$ is an exact low-rank matrix Eq. (3) implies perfect reconstruction, while the error in Eq. (2) remains non-zero.

The column subset selection problem can be considered as a form of *unsupervised feature selection*, which arises frequently in the analysis of large datasets. For example, column subset selection has been applied to various tasks such as summarizing population genetics, testing electronic circuits, recommendation systems, etc. Interested readers can refer to [5, 2] for further motivations.

Many methods have been proposed for the column subset selection problem [7, 16, 15, 9, 14]. Most of these methods can be roughly categorized into two classes. One class of algorithms are based on *rank-revealing QR* (RRQR) decomposition [7, 16] and it has been shown that RRQR is nearly optimal for solving the column subset selection problem (see e.g., Table 1 in [5]). On the other hand, sampling based methods [15, 9, 14] try to select columns by sampling from certain distributions over all columns of an input matrix. These algorithms are much faster than RRQR and achieves comparable performance if the sampling distribution is carefully selected [9, 14].

Although the column subset selection problem with access to the full input matrix has been extensively studied, often in practice it is hard or even impossible to obtain the complete data. For example, for the genetic variation detection problem it could be expensive and time-consuming to obtain full DNA sequences of an entire population. The presence of missing data poses new challenges for column subset selection, as many well-established algorithms seem incapable of handling missing data in an elegant way. Several heuristic al-

[1] $\mathbf{C}^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{C}$.

[2] In general, the number of selected columns s is larger than or equal to the target rank k.

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38. Copyright 2015 by the authors.

gorithms have been proposed recently, including the Block OMP algorithm [2] and the group Lasso formulation proposed in [4]. Nevertheless, no theoretical guarantee or error bounds have been derived for these methods.

One key challenge posed by the absence of data is the difficulty of computing certain column sampling distributions when the input matrix has coherent rows/columns. For instance, it is very difficult to compute statistical leverage scores (which is essential to the subspace sampling algorithm [14]) using partially obesrved data because closeness of two subspaces (e.g., $\|\mathbf{U}-\widetilde{\mathbf{U}}\|_\xi \leq \epsilon$, $\xi = 2$ or F) does not imply closeness of their incoherence levels (i.e., $\mu(\mathcal{U}) \not\approx \mu(\widetilde{\mathcal{U}})$). Though Chen et al. [8] proposed an algorithm estimating statistical leverage scores without access to the complete input matrix, their method only works for exact low-rank matrices, and it is not clear the method will work in the approximately low-rank setting (at least any tiny amount of deterministic noise will break the algorithm). On the other hand, when both the row and the column space of an input matrix are incoherent, column subset selection becomes trivial because uniform sampling of columns is sufficient to achieve good performance.

In this paper, we propose two column subset selection algorithms based on the idea of *active sampling* of the input matrix. In our algorithms, observed matrix entries are chosen sequentially and in a feedback-driven manner. Note that the algorithms make very few measurements of the input matrix, which differs from previous feedback-driven resampling methods in the theoretical computer science literature (e.g., [24]). The active sampling scheme has been shown to outperform all passive schemes in several settings [17, 20, 1], and furthermore it works for matrices with coherent rows/columns under which passive learning provably fails [20].

The contribution of this paper is two-fold. To the best of our knowledge, the proposed methods are the first column subset selection algorithms that enjoy theoretical guarantee of reconstruction error with missing data. Furthermore, when the input matrix is a noisy version of an underlying column-incoherent low-rank matrix, our proposed algorithm achieves a relative error bound.

Finally, we note that the reconstruction error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi$ and the approximation error $\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_\xi$ are not necessarily the same with missing data, because there is no simple procedure to compute $\mathbf{C}\mathbf{C}^\dagger\mathbf{M}$ without access to the complete input matrix $\mathbf{M}$. In this paper we primarily focus on the reconstruction error, but we prove upper bounds for both errors. We also focus on the Frobenious norm $\|\cdot\|_F$ for the error term just like previous work on sampling based matrix approximation methods.

2 PRELIMINARIES

2.1 Notations

For any matrix $\mathbf{M}$ we use $\mathbf{M}^{(i)}$ to denote the i-th column of $\mathbf{M}$. Similarly, $\mathbf{M}_{(i)}$ denotes the i-th row of $\mathbf{M}$. All norms $\|\cdot\|$ are two norms unless otherwise specified.

We assume the input matrix is of size $n_1 \times n_2$, $n = \max(n_1, n_2)$. We further assume that $n_1 \leq n_2$. We use $\boldsymbol{x}_i = \mathbf{M}^{(i)} \in \mathbb{R}^{n_1}$ to denote the i-th column of $\mathbf{M}$. Furthermore, for any column vector $\boldsymbol{x}_i \in \mathbb{R}^{n_1}$ and index subset $\Omega \subseteq [n_1]$, define the subsampled vector $\boldsymbol{x}_{i,\Omega}$ and the scaled subsampled vector $\mathcal{R}_\Omega(\boldsymbol{x}_i)$ as

$$\boldsymbol{x}_{i,\Omega} = \mathbf{1}_\Omega \circ \boldsymbol{x}_i, \quad \mathcal{R}_\Omega(\boldsymbol{x}_i) = \frac{n_1}{|\Omega|}\mathbf{1}_\Omega \circ \boldsymbol{x}_i, \tag{4}$$

where $\mathbf{1}_\Omega \in \{0,1\}^{n_1}$ is the indicator vector of Ω and $\circ$ is the Hadarmard product (entrywise product). We also generalize the definition in Eq. (4) to matrices by applying the same operator on each column.

2.2 Subspace and vector incoherence

Matrix incoherence plays a vital role in various matrix completion and approximation tasks [22, 20, 6, 18]. For any matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$ of rank k, singular value decomposition yields $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n_1 \times k}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times k}$ have orthonormal columns. Let $\mathcal{U} = \text{span}(\mathbf{U})$ and $\mathcal{V} = \text{span}(\mathbf{V})$ be the column and row space of $\mathbf{M}$. The *column space incoherence* is defined as

$$\mu(\mathcal{U}) := \frac{n_1}{k}\max_{i=1}^{n_1}\|\mathbf{U}^\top \boldsymbol{e}_i\|_2^2 = \frac{n_1}{k}\max_{i=1}^{n_1}\|\mathbf{U}_{(i)}\|_2^2. \tag{5}$$

Note that $\mu(\mathcal{U})$ is always between 1 and n_1/k. Similarly, the *row space incoherence* is defined as

$$\mu(\mathcal{V}) := \frac{n_2}{k}\max_{i=1}^{n_2}\|\mathbf{V}^\top \boldsymbol{e}_i\|_2^2 = \frac{n_2}{k}\max_{i=1}^{n_2}\|\mathbf{V}_{(i)}\|_2^2. \tag{6}$$

In this paper we also make use of incoherence level of vectors, which previously appears in [3, 19, 20]. For a column vector $\boldsymbol{x} \in \mathbb{R}^{n_1}$, its incoherence is defined as

$$\mu(\boldsymbol{x}) := \frac{n_1\|\boldsymbol{x}\|_\infty^2}{\|\boldsymbol{x}\|_2^2}. \tag{7}$$

It is an easy observation that if $\boldsymbol{x}$ lies in the subspace $\mathcal{U}$ then $\mu(\boldsymbol{x}) \leq k\mu(\mathcal{U})$. In this paper we adopt incoherence assumptions on the column space $\mathcal{U}$, which subsequently yields incoherent column vectors $\boldsymbol{x}_i$. No incoherence assumption on the row space $\mathcal{V}$ or row vectors $\mathbf{M}_{(i)}$ is made.

2.3 Norm and volume sampling

Norm sampling for column subset selection was proposed in [15] and has found applications in a number of matrix computation tasks, e.g., approximate matrix multiplication [11] and low-rank or compressed matrix approximation [12, 13]. The idea is to sample each column with probability proportional to its squared ℓ_2 norm, i.e., $\Pr[i \in C] \propto \|\mathbf{M}^{(i)}\|_2^2$ for $i \in \{1, 2, \cdots, n_2\}$. These types of algorithms usually come with an additive error bound on their approximation performance.

For volume sampling [9], a subset of columns C is picked with probability proportional to the volume of the simplex spanned by columns in C. That is, $\Pr[C] \propto \mathrm{vol}(\Delta(C))$ where $\Delta(C)$ is the simplex spanned by $\{\mathbf{M}^{(C(1))}, \cdots, \mathbf{M}^{(C(k))}\}$. Unlike norm sampling, volume sampling achieves a relative error bound for column subset selection. Although exact volume sampling could be hard, it was shown that an iterative norm sampling procedure serves as a nice approximation [10].

3 COLUMN SUBSET SELECTION VIA ACTIVE SAMPLING

In this section we propose two column subset selection algorithms that only observe a small portion of an input matrix. Both algorithms employ the idea of active sampling to handle matrices with coherent rows. While Algorithm 1 achieves an additive approximation error guarantee for any matrix, Algorithm 2 achieves a relative-error approximation guarantee when the input matrix has certain structure.

3.1 Active norm sampling

We first present an active norm sampling algorithm (Algorithm 1) for column subset selection under the missing data setting. The algorithm is inspired by the norm sampling work for column subset selection by Frieze et al. [15] and the low-rank matrix approximation work by Krishnamurthy and Singh [20].

The first step of Algorithm 1 is to estimate the ℓ_2 norm for each column by uniform subsampling. Afterwards, s columns of $\mathbf{M}$ are selected independently with probability proportional to their ℓ_2 norms. Finally, the algorithm constructs a sparse approximation of the input matrix by sampling each matrix entry with probability proportional to the square of the corresponding column's norm and then a $\mathbf{CX}$ approximation is obtained.

When the input matrix $\mathbf{M}$ has incoherent columns, the reconstruction error as well as $\mathbf{CX}$ approximation error can be bounded as in Theorem 1.

Theorem 1. *Suppose* $\max_{i=1}^{n_2} \mu(\boldsymbol{x}_i) \le \mu_1$ *for some positive constant* μ_1. *Let* $\mathbf{C}$ *and* $\mathbf{X}$ *be the output of Algorithm 1. Denote* $\mathbf{M}_k$ *the best rank-k approximation of* $\mathbf{M}$. *Fix* $\delta = \delta_1 + \delta_2 + \delta_3 > 0$. *With probability at least* $1 - \delta$, *we have*

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F \le \|\mathbf{M} - \mathbf{M}_k\|_F + \varepsilon\|\mathbf{M}\|_F \qquad (8)$$

provided that $s = \Omega(k\varepsilon^{-2}/\delta_2)$, $m_1 = \Omega(\mu_1 \log(n/\delta_1))$. *Furthermore, if* $m_2 = \Omega(\mu_1 s \log^2(n/\delta_3)/(\delta_2\varepsilon^2))$ *then with probability* $\ge 1 - \delta$ *we have the following bound on approximation error:*

$$\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F \le \|\mathbf{M} - \mathbf{M}_k\|_F + 2\varepsilon\|\mathbf{M}\|_F. \qquad (9)$$

As a remark, Theorem 1 shows that one can achieve ε additive approximation error using Algorithm 1 with expected sample complexity (omitting dependency on δ)

$$\Omega\left(\mu_1 n_2 \log(n) + \frac{kn_1}{\varepsilon^2} + \frac{k\mu_1 n_2 \log^2(n)}{\varepsilon^4}\right) = \Omega(k\mu_1\varepsilon^{-4} n \log^2 n).$$

Note that if we only care about ε reconstruction error then the sample complexity only depends on ε^{-2} instead of ε^{-4}.

3.2 Active approximate volume sampling

In this section we present Algorithm 2, another active sampling algorithm based on approximate volume sampling [10]. Although Algorithm 2 is more complicated than Algorithm 1, it achieves a relative error bound on inputs that are noisy perturbation of some underlying low-rank matrix.

Algorithm 2 employs the idea of *iterative norm sampling*. That is, after selecting l columns from $\mathbf{M}$, the next column is sampled according to column norms of a *projected* matrix $\mathcal{P}_{\mathcal{C}^\perp}(\mathbf{M})$, where $\mathcal{C}$ is the subspace spanned by currently selected columns. It can be shown that iterative norm sampling serves as an approximation of *volume sampling*, which leads to relative error bounds [9, 10].

Theorem 2 provides a relative-error analysis of Algorithm 2 when the input matrix $\mathbf{M}$ is the sum of a low rank matrix $\mathbf{A}$ and a noise matrix $\mathbf{R}$. Such assumptions bound the incoherence level of projected columns, which is required for estimating prjected column norms [19, 20]. Note that a statistical noise model is necessary for our analysis because the projection of a deterministic incoherent noise vector may no longer be incoherent.

Theorem 2. *Fix* $\delta > 0$. *Suppose* $\mathbf{M} = \mathbf{A} + \mathbf{R}$, *where* $\mathbf{A}$ *is a rank-k deterministic matrix with incoherent column space (i.e.,* $\mu(\mathcal{U}(\mathbf{A})) \le \mu_0$*) and* $\mathbf{R}$ *is a random*

Algorithm 1 Active norm sampling for column subset selection with missing data

Input: size of column subset s, expected number of samples per column m_1 and m_2.
Norm estimation: For each column i, sample each index in $\Omega_{1,i} \subseteq [n_1]$ i.i.d. from Bernoulli(m_1/n_1). observe $\boldsymbol{x}_{i,\Omega_{1,i}}$ and compute $\hat{c}_i = \frac{n_1}{m_1}\|\boldsymbol{x}_{i,\Omega_{1,i}}\|_2^2$. Define $\hat{f} = \sum_i \hat{c}_i$.
Column subset selection: Set $\mathbf{C} = \mathbf{0} \in \mathbb{R}^{n_1 \times s}$.
- For $t \in [s]$: sample $i_t \in [n_2]$ such that $\Pr[i_t = j] = \hat{c}_j/\hat{f}$. Observe $\mathbf{M}^{(i_t)}$ in full and set $\mathbf{C}^{(t)} = \mathbf{M}^{(i_t)}$.

Matrix approximation: Set $\widehat{\mathbf{M}} = \mathbf{0} \in \mathbb{R}^{n_1 \times n_2}$.
- For each column $\boldsymbol{x}_i$, sample each index in $\Omega_{2,i} \subseteq [n_1]$ i.i.d. from Bernoulli($m_{2,i}/n_1$), where $m_{2,i} = m_2 n_2 \hat{c}_i/\hat{f}$; observe $\boldsymbol{x}_{i,\Omega_{2,i}}$.
- Update: $\widehat{\mathbf{M}} = \widehat{\mathbf{M}} + (\mathcal{R}_{\Omega_{2,i}}(\boldsymbol{x}_i))\boldsymbol{e}_i^\top$.

Output: selected columns $\mathbf{C}$ and coefficient matrix $\mathbf{X} = \mathbf{C}^\dagger\widehat{\mathbf{M}}$.

Algorithm 2 Active approximate volume sampling for column subset selection with missing data

Input: target rank $k \ll n_1$, expected number of samples per column m.
Entrywise sampling: For each column i, sample each index i in an index set $\Omega_i \subseteq [n_1]$ i.i.d. from Bernoulli(m/n_1). Observe $\boldsymbol{x}_{i,\Omega_i}$.
Column subset selection: Set $C = \emptyset$, $\mathcal{U} = \emptyset$. Suppose $\mathbf{U}$ is an orthonormal basis of $\mathcal{U}$.
for $t \in \{1, 2, \cdots, k\}$ **do**
    For $i \in \{1, \cdots, n_2\}$, compute $\hat{c}_i^{(t)} = \frac{n_1}{m}\|\boldsymbol{x}_{i,\Omega_i} - \mathbf{U}_{\Omega_i}(\mathbf{U}_{\Omega_i}^\top\mathbf{U}_{\Omega_i})^{-1}\mathbf{U}_{\Omega_i}^\top\boldsymbol{x}_{i,\Omega_i}\|_2^2$. Set $\hat{f}^{(t)} = \sum_{i=1}^{n_2}\hat{c}_i^{(t)}$.
    Select a column i_t at random, with probability $\Pr[i_t = j] = \hat{p}_j^{(t)} = \hat{c}_j^{(t)}/\hat{f}^{(t)}$.
    Observe $\mathbf{M}^{(i_t)}$ in full and update: $C \leftarrow C \cup \{i_t\}$, $\mathcal{U} \leftarrow \mathrm{span}(\mathcal{U}, \{\mathbf{M}^{(i_t)}\})$.
end for
Matrix approximation: Compute $\widehat{\mathbf{M}} = \sum_{i=1}^{n_2}\mathbf{U}(\mathbf{U}_{\Omega_i}^\top\mathbf{U}_{\Omega_i})^{-1}\mathbf{U}_{\Omega_i}^\top\boldsymbol{x}_{i,\Omega_i}\boldsymbol{e}_i^\top$.
Output: Selected columns $\mathbf{C} = (\mathbf{M}^{(C(1))}, \cdots, \mathbf{M}^{(C(k))})$ and $\mathbf{X} = \mathbf{C}^\dagger\widehat{\mathbf{M}}$.

matrix with i.i.d. zero-mean Gaussian distributed entries. Suppose $k = O(n_1/\log(n_2/\delta))$. Let $\mathbf{C}$ and $\mathbf{X}$ be the output of Algorithm 2 run with parameter

$$m = \Omega(k^2\mu_0\log^2(n/\delta)).$$

Then with probability $\geq 1-\delta$ the following holds:

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F^2 \leq \frac{2.5^k(k+1)!}{\delta}\|\mathbf{R}\|_F^2; \tag{10}$$

furthermore,

$$\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F^2 \leq \frac{2.5^{k+1}(k+1)!}{\delta}\|\mathbf{R}\|_F^2. \tag{11}$$

Compared to Theorem 1, the error bound in Theorem 2 is relative to the noise level $\|\mathbf{R}\|_F^2$. As a consequence, when the noise goes to zero the reconstruction error of Algorithm 2 will go to zero too, while the bound on Algorithm 1 still has certain amount of reconstruction error even under the exact low rank case. In fact, when the noise is eliminated Algorithm 2 is similar in spirit to adaptive matrix/tensor completion algorithms presented in [19, 20].

4 PROOFS

4.1 Proof sketch of Theorem 1

The proof of Theorem 1 can be divided into two steps. First, in Lemma 1 we show that (approximate) column sampling yields an additive error bound for column subset selection. Its proof is very similar to the one presented in [15] and we defer it to Appendix A. Second, we cite a lemma from [20] to show that with high probability the first pass in Algorithm 1 gives accurate estimates of column norms of the input matrix $\mathbf{M}$.

Lemma 1. *Provided that $(1-\alpha)\|\boldsymbol{x}_i\|_2^2 \leq \hat{c}_i \leq (1+\alpha)\|\boldsymbol{x}_i\|_2^2$ for $i = 1, 2, \cdots, n_2$, with probability $\geq 1-\delta$ we have*

$$\|\mathbf{M} - \mathcal{P}_C(\mathbf{M})\|_F \leq \|\mathbf{M} - \mathbf{M}_k\|_F + \sqrt{\frac{(1+\alpha)k}{(1-\alpha)\delta s}}\|\mathbf{M}\|_F, \tag{12}$$

where $\mathbf{M}_k$ is the best rank-k approximation of $\mathbf{M}$.

Lemma 2 ([20], Lemma 10)**.** *Fix $\delta \in (0,1)$. Assume $\mu(\boldsymbol{x}_i) \leq \mu_1$ holds for $i = 1, 2, \cdots, n_2$. For some fixed $i \in \{1, \cdots, n_2\}$ with probability $\geq 1-2\delta$ we have*

$$(1-\alpha)\|\boldsymbol{x}_i\|_2^2 \leq \hat{c}_i \leq (1+\alpha)\|\boldsymbol{x}_i\|_2^2 \tag{13}$$

with $\alpha = \sqrt{\frac{2\mu_1}{m_1}\log(1/\delta)} + \frac{2\mu_1}{3m_1}\log(1/\delta)$. Furthermore, if $m_1 = \Omega(\mu_1\log(n_2/\delta))$ with carefully chosen constants then Eq. (13) holds uniformly for all columns with $\alpha = 0.5$.

Combining Lemma 1 and Lemma 2 and setting $s = \Omega(k\varepsilon^{-2}/\delta)$ for some target accuracy threshold ε we

have that with probability $1-3\delta$ the reconstruction error bound Eq. (8) holds.

In order to bound the approximation error $\|\mathbf{M}-\mathbf{CX}\|_F^2$, we cite another lemma from [20] that analyzes the performance of the second pass of Algorithm 1.

Lemma 3 ([20], Lemma 9)**.** *Provided that* $(1-\alpha)\|\boldsymbol{x}_i\|_2^2 \le \hat{c}_i \le (1+\alpha)\|\boldsymbol{x}_i\|_2^2$ *for* $i=1,2,\cdots,n_2$*, with probability* $\ge 1-\delta$ *we have*

$$\|\mathbf{M}-\widehat{\mathbf{M}}\|_2 \le \|\mathbf{M}\|_F\sqrt{\frac{1+\alpha}{1-\alpha}}\left(\frac{4}{3}\sqrt{\frac{n_1\mu_1}{m_2 n_2}}\log\left(\frac{n_1+n_2}{\delta}\right)+\sqrt{\frac{4}{m_2}\max\left(\frac{n_1}{n_2},\mu_1\right)\log\left(\frac{n_1+n_2}{\delta}\right)}\right). \quad (14)$$

The complete proof of Theorem 1 is deferred to Appendix A.

4.2 Proof sketch of Theorem 2

We take four steps to prove Theorem 2. At the first step, we show that when the input matrix has a low rank plus noise structure then with high probability *for all small subsets of columns* the spanned subspace has incoherent column space (assuming the low-rank matrix has incoherent column space) and furthermore, the projection of the other columns onto the orthogonal complement of the spanned subspace are incoherent, too. Given the incoherence condition we can easily prove a norm estimation result similar to Lemma 2, which is the second step. For the third step, we note that the approximate iterative norm sampling procedure is an approximation of *volume sampling*, a column sampling scheme that is known to have a relative error bound. Finally, the coefficient matrix $\mathbf{X}$ is reconstructed using a method similar to the matrix completion algorithm in [20].

STEP 1: We first prove that when the input matrix $\mathbf{M}$ is a noisy low-rank matrix with incoherent column space, with high probability *a fixed* column subset also has incoherent column space. This is intuitive because the Gaussian perturbation matrix is highly incoherent with overwhelming probability. A more rigorous statement is shown in Lemma 4.

Lemma 4. *Suppose* $\mathbf{A}$ *has incoherent column space, i.e.,* $\mu(\mathcal{U}(\mathbf{A})) \le \mu_0$*. Fix* $C \subseteq [n_2]$ *to be any subset of column indices that has* s *elements and* $\delta > 0$*. Let* $\mathbf{C} = [\mathbf{M}^{(C(1))},\cdots,\mathbf{M}^{(C(s))}] \in \mathbb{R}^{n_1\times s}$ *be the compressed matrix and* $\mathcal{U}(C) = span(\mathbf{C})$ *denote the subspace spanned by the selected columns. Suppose* $s \le k$*,* $k \le n_1/4 - k$ *and* $\log(4n_2/\delta) \le n_1/64$*. Then with probability* $\ge 1-\delta$ *over the random drawn of* $\mathbf{R}$ *we have*

$$\mu(\mathcal{U}(C)) = \frac{n_1}{s}\max_{1\le i\le n_1}\|\mathcal{P}_{\mathcal{U}(C)}\boldsymbol{e}_i\|_2^2 = O\left(\frac{k\mu_0 + s + \sqrt{s\log(n_1/\delta)}+\log(n_1/\delta)}{s}\right); \quad (15)$$

furthermore, with probability $\ge 1-\delta$ *the following holds:*

$$\mu(\mathcal{P}_{\mathcal{U}(C)^\perp}(\mathbf{M}^{(i)})) = O(k\mu_0+\log(n_1n_2/\delta)), \quad \forall i \notin C. \quad (16)$$

The proof of Lemma 4 is based on the fact that Gaussian noise is highly incoherent, and that the randomness imposed on each column of the input matrix is independent. The complete proof can be found in Appendx B.

Given Lemma 4, Corollary 1 holds by taking a uniform bound over all $\sum_{s=1}^k \binom{n_2}{s} = O(k(n_2)^k)$ column subsets that contain no more than k elements. The $2k\log(4n_2/\delta) \le n_1/64$ condition is only used to ensure that the desired failure probability δ is not exponentially small. Typically, in practice the intrinsic dimension k is much smaller than the ambient dimension n_1.

Corollary 1. *Fix* $\delta > 0$*. Suppose* $k \le n_1/8$ *and* $2k\log(4n_2/\delta) \le n_1/64$*. With probability* $\ge 1-\delta$ *the following holds: for any subset* $C \subseteq [n_2]$ *with at most* k *elements, the spanned subspace* $\mathcal{U}(C)$ *satisfies*

$$\mu(\mathcal{U}(C)) \le O(k|C|^{-1}\mu_0\log(n/\delta)); \quad (17)$$

furthermore,

$$\mu(\mathcal{P}_{\mathcal{U}(C)^\perp}(\mathbf{M}^{(i)})) = O(k\mu_0\log(n/\delta)), \quad \forall i \notin C. \quad (18)$$

STEP 2: In this step, we prove that the norm estimation scheme in Algorithm 2 works when the incoherence conditions in Eq. (17) and (18) are satisfied. More specifically, we have the following lemma bounding the norm estimation error:

Lemma 5. *Fix* $i \in \{1,\cdots,n_2\}$*,* $t \in \{1,\cdots,k\}$ *and* $\delta,\delta' > 0$*. Suppose Eq. (17) and (18) hold with probability* $\ge 1-\delta$*. Let* $\mathcal{S}_t$ *be the subspace spanned by selected columns at the* t*-th round and let* $\hat{c}_i^{(t)}$ *denote the estimated squared norm of the* i*th column. If* m *satisfies*

$$m = \Omega(k\mu_0\log(n/\delta)\log(k/\delta')), \quad (19)$$

then with probability $\ge 1-\delta-4\delta'$ *we have*

$$\frac{1}{2}\|[\mathbf{E}_t]_{(i)}\|_2^2 \le \hat{c}_i^{(t)} \le \frac{5}{4}\|[\mathbf{E}_t]_{(i)}\|_2^2. \quad (20)$$

Here $\mathbf{E}_t = \mathcal{P}_{\mathcal{S}_t^\perp}(\mathbf{M})$ *denotes the projected matrix at the* t*-th round.*

Lemma 5 is similar with previous results on subspace detection [3] and matrix approximation [20]. Namely, one can accurately estimate the ℓ_2 norm of a vector provided that the vector is incoherent. The proof of Lemma 5 is deferred to Appendix B.

Similar to the first step, by taking a union bound over all possible subsets of picked columns and $n_2 - k$ unpicked columns we can prove a stronger version of Lemma 5, as shown in Corollary 2.

Corollary 2. *Fix $\delta, \delta' > 0$. Suppose Eq. (17) and (18) hold with probability $\geq 1 - \delta$. If*

$$m = \Omega(k^2 \mu_0 \log(n/\delta) \log(n/\delta')) \tag{21}$$

then with probability $\geq 1-\delta-4\delta'$ the following property holds for any selected column subset by Algorithm 2:

$$\frac{2}{5} \frac{\|[\mathbf{E}_t]_{(i)}\|_2^2}{\|\mathbf{E}_t\|_F^2} \leq \hat{p}_i^{(t)} \leq \frac{5}{2} \frac{\|[\mathbf{E}_t]_{(i)}\|_2^2}{\|\mathbf{E}_t\|_F^2}, \forall i \in [n_2], t \in [k], \tag{22}$$

where $\hat{p}_i^{(t)} = \hat{c}_i^{(t)} / \hat{f}^{(t)}$ is the sampling probability of the ith column at round t.

STEP 3: To begin with, we define *volume sampling* distributions:

Definition 1 (volume sampling, [9])**.** *A distribution p over column subsets of size k is a volume sampling distribution if*

$$p(C) = \frac{vol(\Delta(C))^2}{\sum_{T:|T|=k} vol(\Delta(T))^2}, \quad \forall |C| = k. \tag{23}$$

Volume sampling has been shown to achieve a relative error bound for column subset selection, which is made precise by Theorem 3 cited from [10, 9].

Theorem 3 ([10], Theorem 4)**.** *Fix a matrix $\mathbf{M}$ and let $\mathbf{M}_k$ denote the best rank-k approximation of $\mathbf{M}$. If the sampling distribution p is a volume sampling distribution defined in Eq. (23) then*

$$\mathbb{E}_C \left[\|\mathbf{M} - \mathcal{P}_{\mathcal{V}(C)}(\mathbf{M})\|_F^2 \right] \leq (k+1) \|\mathbf{M} - \mathbf{M}_k\|_F^2; \tag{24}$$

furthermore, applying Markov's inequality one can show that with probability $\geq 1 - \delta$

$$\|\mathbf{M} - \mathcal{P}_{\mathcal{V}(C)}(\mathbf{M})\|_F^2 \leq \frac{k+1}{\delta} \|\mathbf{M} - \mathbf{M}_k\|_F^2. \tag{25}$$

In general, volume sampling is intractable. However, in [10] it was shown that iterative norm sampling serves as an approximate of volume sampling and achieves a relative error bound as well. In Lemma 6 we present an extension of this result. Namely, *approximate* iterative column norm sampling is an approximate of volume sampling, too. Its proof is very similar to the one presented in [10] and we defer it to Appendix B.

Lemma 6. *Let p be the volume sampling distribution defined in Eq. (23). Suppose the sampling distribution of a k-round sampling strategy $\hat{p}$ satisfies Eq. (22). Then we have*

$$\hat{p}_C \leq 2.5^k k! p_C, \quad \forall |C| = k. \tag{26}$$

STEP 4: We can now prove the error bound for reconstruction error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger \mathbf{M}\|_F$ of Algorithm 2 by combining Corollary 1, 2, Lemma 6 and Theorem 3. In particular, Corollary 1 and 2 guarantees that Algorithm 2 estimates column norms accurately with high probability; then one can apply Lemma 6 to show that the sampling distribution employed in the algorithm is actually an approximate volume sampling distribution, which is known to achieve relative error bounds (by Theorem 3).

To reconstruct the coefficient matrix $\mathbf{X}$ and to further bound the approximation error $\|\mathbf{M} - \mathbf{C}\mathbf{X}\|_F$, we apply the $\mathbf{U}(\mathbf{U}_\Omega^\top \mathbf{U}_\Omega)^{-1} \mathbf{U}_\Omega$ operator on every column to build a low-rank approximation $\widehat{\mathbf{M}}$. It was shown in [19, 3] that this operator recovers all components in the underlying subspace $\mathcal{U}$ with high probability, and hence achieves a relative error bound for low-rank matrix approximation. More specifically, we have Lemma 7, which is proved in Appendix B.

Lemma 7. *Let $C \subseteq [n_2]$, $|C| = k$ be the indices of columns selected in the column subset selection phase of Algorithm 2. Suppose Eq. (17) and (18) are satisfied with probability $\geq 1 - \delta$. If m satisfies*

$$m = \Omega(k \mu_0 \log(n/\delta) \log(n/\delta'')), \tag{27}$$

then with probability $\geq 1 - \delta - \delta''$ we have

$$\|\mathbf{M} - \widehat{\mathbf{M}}\|_F^2 \leq 2.5 \|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger \mathbf{M}\|_F^2. \tag{28}$$

Note that all columns of $\widehat{\mathbf{M}}$ are in the subspace $\mathcal{U}(C)$. Therefore, $\mathbf{C}\mathbf{X} = \mathbf{C}\mathbf{C}^\dagger \widehat{\mathbf{M}} = \widehat{\mathbf{M}}$. The proof of Eq. (11) is then straightforward.

5 SIMULATIONS

In this section we use simulations to demonstrate the effectiveness of our proposed algorithms. All input matrices are 50×50. To obtain exact low rank inputs, we first generate a random Gaussian matrix and use its top k SVD as the input. For noisy perturbed low rank inputs $\mathbf{M} = \mathbf{A} + \mathbf{R}$, the noise-to-signal ratio (NSR). is measured by $\|\mathbf{R}\|_F / \|\mathbf{A}\|_F$, where $\mathbf{A}$ is an exact low rank matrix and $\mathbf{R}$ is a Gaussian white noise matrix.

We first compare the reconstruction error of Algorithm 1 and 2 for noisy low-rank inputsunder different settings of column norms, missing rates and NSR. Under incoherent column settings column norms are distributed fairly uniformly while under coherent column

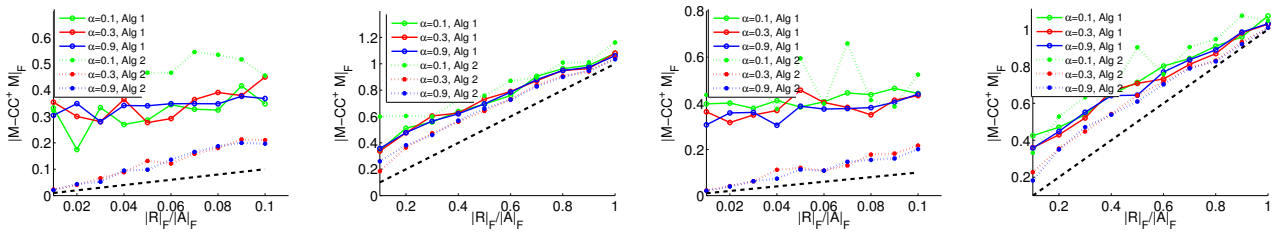

Figure 1: Reconstruction error of Algorithm 1 and 2 for noisy low-rank matrices with incoherent (left two) and coherent columns (right two) under various settings of missing rate $(1-\alpha)$ and NSR ratio (the black line).

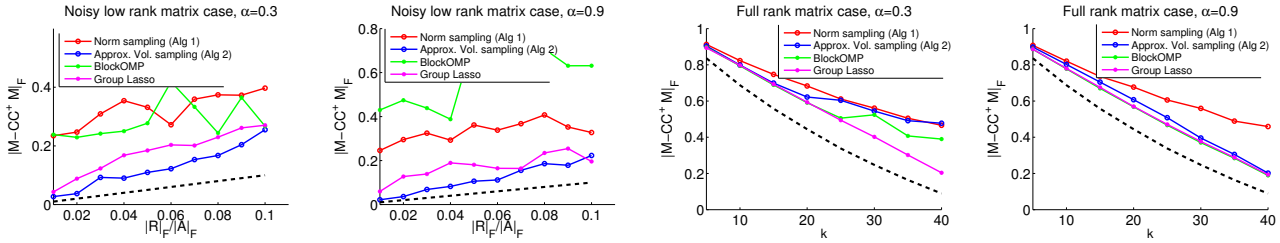

Figure 3: Reconstruction error of sampling and heuristic based algorithms. The black dash line indicates $\|\mathbf{R}\|_F/\|\mathbf{A}\|_F$ in the noisy low-rank matrix case and $\|\mathbf{M}-\mathbf{M}_k\|_F$ in the full-rank matrix case.

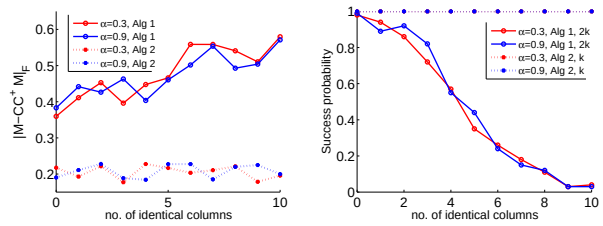

Figure 2: Reconstruction error for noisy low-rank matrices (left) and matrix completion success probability for exact low-rank matrices (right) when multiple columns are identical.

settings the column norms are distributed in a log-normal way. α indicates the entrywise sampling probability (that is, $\alpha n_1 n_2$ is the expected number of observed entries and $1-\alpha$ is the missing rate). After obtaining a subset of columns $\mathbf{C}$, we evaluate its performance by the Frobenius-norm reconstruction error $\|\mathbf{M}-\mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F$.

From Figure 1 it can be seen that when the missing rate is not too high ($1-\alpha \le 0.7$) the active approximate volume sampling algorithm always outperforms the norm sampling one. This phenomenon is more evident when the noise-to-signal ratio $\|\mathbf{R}\|_F/\|\mathbf{A}\|_F$ is small, because as an algorithm with relative error bounds, the reconstruction error of Algorithm 2 goes down with $\|\mathbf{R}\|_F/\|\mathbf{A}\|_F$. This is not the case for Algorithm 1, which only has additive error guarantees. On the other hand, when the magnitude of the noise $\mathbf{R}$ is comparable to $\mathbf{A}$ both algorithms are equally effective.

The disadvantage of Algorithm 1 is more apparent in Figure 2, where the input matrix contains multiple identical columns with large norm. Under such settings, it is highly likely that Algorithm 1 will select identical columns many times, which leads to performance deterioration. Figure 2 shows that as the number of identical columns increases, the reconstruction error of Algorithm 1 gets larger and the success probability of matrix completion (with exact low-rank inputs) decreases rapidly. In contrast, the performance of Algorithm 2 remains the same regardless of repeated columns.

In Figure 3 we compare our active sampling based algorithms with several heuristic based methods, for example, Block OMP [2] and Group Lasso [4]. The observation is that the active approximate volume sampling algorithm (Algorithm 2) outperforms both Block OMP and group Lasso for noisy low-rank inputs, and its performance is comparable with group Lasso for full-rank deterministic matrices when the missing rate is not too high. On the other hand, both of the proposed sampling based algorithms are quite efficient, only scanning through the input and computing SVD on small matrices of size $O(k)$. In contrast, for the group Lasso method one needs to compute a solution path of a Lasso problem with $n_1 n_2$ variables. Though computationally expensive, under high missing rates block OMP and group Lasso outperform our proposed methods and it would be interesting to study their

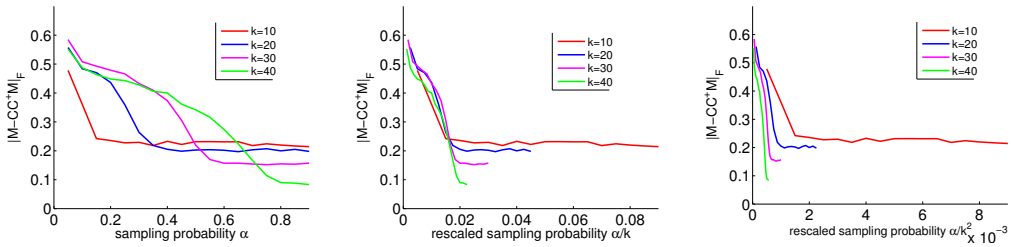

Figure 4: Reconstruction error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F$ for the active approximate volume sampling algorithm as a function of α (left), α/k (middle) and α/k^2 (right). Error curves plotted under 4 different rank (k) settings.

theoretical properties.

We also try to verify the sample complexity dependence on the intrinsic matrix rank k for Algorithm 2. To do this, we run the algorithm under various settings of intrinsic dimension k and the sampling probability α (which is basically proportional to the expected number of per-column samples m). We then plot the reconstruction error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F$ against α, α/k and α/k^2 in Figure 4.

Theorem 2 states that the dependence of m on k should be $m = \widetilde{O}(k^2)$, ignoring logarithmic factors. However, in Figure 4 one can observe that when the reconstruction error is plotted against α/k the different curves coincide. This suggests that the actual dependence of m on k should be close to linear instead of quadratic. It is an interesting question whether we can get rid of the use of union bounds over all n_2-choose-k column subsets in the proof of Theorem 2 in order to get a near linear dependence over k.

6 DISCUSSION

6.1 Sample complexity, column subset size and reconstruction error

We first remark on the connection of sample complexity (i.e., number of observed matrix entries), size of column subsets and reconstruction error for column subset selection. In many matrix completion/approximation tasks increasing the sample complexity usually leads to increased approximation accuracy (e.g., [20]). However, for column subset selection when the target column subset size is fixed the sample complexity acts more like a threshold: if not enough number of matrix entries are observed then the algorithm fails, but otherwise the reconstruction error does not differ much. In fact, the guarantee in Eq. (8), for example, is exactly the same as in [15] under the fully observed setting, i.e., $m_1 = n_1$.

Figure 1 is an excellent illustration of this phenomenon. When $\alpha = 0.1$ the reconstruction error of Algorithm 2 is very high, which means the algorithm does not have enough samples. However, for $\alpha = 0.3$ and $\alpha = 0.9$ the performance of Algorithm 2 is very similar. Such phase transition is also present in low-rank matrix completion; e.g., see Figure 2 in [20].

6.2 Error analysis of Algorithm 2

In Theorem 2 we derive a relative error bound with a sample complexity analysis for the adaptive approxiamte volume sampling algorithm. However, the results are not completely satisfactory. First, the sample complexity dependency on the target matrix rank k is quadratic, which we conjecture is too loose based on simulations shown in Figure 4. We believe it is possible to improve over the quadratic dependency by avoiding the n-choose-k union bound argument in our proof. In addition, the relative error in Eq. (11) is exponential with respect to the target rank k, which makes the analysis inapplicable for high-rank cases. However, in simulations we observe no significant error increase when the rank of the input matrix is high (e.g., see Figure 3 and 4). It is an interesting question whether this exponential dependency can be improved.

6.3 Relative error bound for general inputs

Theorem 2 shows that Algorithm 2 has relative error guarantee when the input matrix is a low-rank matrix perturbed by Gaussian noise. In fact, we believe that Algorithm 2 works for low-rank inputs with sub-Gaussian noise, too. However, getting a relative error algorithm for more general inputs (e.g., low-rank matrices with deterministic noise or even full-rank matrices) remains an open problem with missing data.

Acknowledgements

We would like to thank Akshay Krishnamurthy for helpful discussions on the proof of Theorem 2. This research is supported in part by grants NSF-1252412 and AFOSR-FA9550-14-1-0285.

References

[1] M. F. Balcan and P. M. Long. Active and passive learning of linear separator under log-concave distributions. In *COLT*, 2013.

[2] L. Balzano, R. Nowak, and W. Bajwa. Column subset selection with missing data. In *NIPS Workshop on Low-Rank Methods for Large-Scale Machine Learning*, 2010.

[3] L. Balzano, B. Recht, and R. Nowak. High-dimensional matched subspace detection when data are missing. In *ISIT*, 2010.

[4] J. Bien, Y. Xu, and M. Mahoney. CUR from a sparse optimization viewpoint. In *NIPS*, 2010.

[5] C. Boutsidis, M. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *SODA*, 2009.

[6] E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[7] T. F. Chan. Rank revealing QR factorizations. *Linear Algebra and Its Applications*, 88:67–82, 1987.

[8] Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward. Completing any low-rank matrix, provably. *arXiv:1306.2979*, 2013.

[9] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2:225–247, 2006.

[10] A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 292–303. 2006.

[11] P. Drineas, R. Kannan, and M. Mahoney. Fast monte carlo algorithms for matrices I: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[12] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[13] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006.

[14] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

[15] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

[16] M. Gu and S. C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996.

[17] J. Haupt, R. M. Castro, and R. Nowak. Distilled sensing: Adaptive sampling for sparse detection and estimation. *IEEE Transactions on Information Theory*, 57(9):6222–6235, 2011.

[18] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[19] A. Krishnamurthy and A. Singh. Low-rank matrix and tensor completion via adaptive sampling. In *NIPS*, 2013.

[20] A. Krishnamurthy and A. Singh. On the power of adaptivity in matrix completion and approximation. *arXiv:1407.3619*, 2014.

[21] B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

[22] B. Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 12:3413–3430, 2011.

[23] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv:1011.3027*, 2010.

[24] S. Wang and Z. Zhang. Improving CUR matrix decomposition and the nyström approximation via adaptive sampling. *The Journal of Machine Learning Research*, 14(1):2729–2769, 2013.