ELEMENT-WISE MATRIX SPARSIFICATION AND RECONSTRUCTION

By

Abhisek Kundu

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

Approved by the
Examining Committee:

_____
Dr. Petros Drineas, Thesis Adviser

_____
Dr. Malik Magdon-Ismail, Member

_____
Dr. Mark Goldberg, Member

_____
Dr. John E. Mitchell, Member

Rensselaer Polytechnic Institute
Troy, New York

July 2015
(For Graduation August 2015)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I did the work presented in this document as a graduate student in the Computer Science Department at Rensselaer Polytechnic Institute during the period Aug 2012 - June 2015 supervised by Dr. Petros Drineas and Dr. Malik Magdon-Ismail.

During these years I was benefited from the discussion related to the topics of this document with Srinivas Nambirajan (graduate student in Mathematics at RPI) and Saurabh Paul (graduate student in Computer Science at RPI).

ABSTRACT

The work presented here is focused mainly on sampling elements from a matrix. We study three topics of current research in Theoretical Computer Science, Machine Learning, and Compressed Sensing involving element-wise sampling: (1) Element-wise Matrix Sparsification, (2) Low-rank Matrix Completion, and (3) CUR Decomposition using Element-wise Sampling. Below we give a high-level description of the topics while leaving the details in subsequent chapters.

(1) **Fast Low-rank Approximation**: Given a matrix we want to sample elements from it based on some probability distribution defined on its elements, such that, we can (approximately) reconstruct the matrix, in some matrix norm, based only on these sampled elements. We want to sample a small number of elements in order to achieve a certain degree of reconstruction accuracy. We propose a generalization of two existing popular sampling methods, and show that our method requires strictly smaller sample size than existing methods. Further, we show that the computation time of the PCA of such sparsified data is significantly faster than that of the full data, while the quality of the PCA of the sparsified data is nearly as good as the true PCA.

(2) **Low-rank Matrix Completion**: We use the nuclear norm minimization framework to reconstruct a low-rank matrix by observing only few of its elements. We seek to reduce the number of elements to be observed in order to reconstruct the matrix exactly. For this, we investigate a novel form of distribution on the elements of a matrix. We show theoretical analysis and experimental results to highlight some of the properties of this distribution in the context of low-rank matrix completion. Our proposed method outperforms the best-known method.

(3) **CUR-Decomposition using Element-wise Sampling**: Here we consider another reconstruction method, namely the CUR-Decomposition, by sampling elements from a matrix. Existing CUR algorithms use all the elements of a matrix in order to achieve certain

level of reconstruction accuracy. In this work, we discuss some of the reconstruction algorithms that need only a handful of elements of a matrix in order to reconstruct it with some provable guarantee.

CHAPTER 1
Recovering PCA from Hybrid-(ℓ_1, ℓ_2) Sparse Sampling of Data Elements

ABSTRACT: This chapter addresses how well we can recover a data matrix when only given a few of its elements. We present a randomized algorithm that element-wise sparsifies the data, retaining only a few of its elements. Our new algorithm independently samples the data using sampling probabilities that depend on both the squares (ℓ_2 sampling) and absolute values (ℓ_1 sampling) of the entries. We prove that the hybrid algorithm recovers a near-PCA reconstruction of the data from a sublinear sample-size: hybrid-(ℓ_1, ℓ_2) inherits the ℓ_2-ability to sample the important elements, as well as, the regularization properties of ℓ_1 sampling, and gives strictly better performance than either ℓ_1 or ℓ_2 on their own. We also give a one-pass version of our algorithm and show experiments to corroborate the theory.

1.1 Introduction

We address the problem of recovering a near-PCA reconstruction of the data from just a few of its entries – element-wise matrix sparsification ([1], [2]) (say, we have a small sample of data points and those data points have missing features). This is a situation that one is confronted with all too often in machine learning. For example, with user-recommendation data, one does not have all the ratings of any given user. Or in a privacy preserving setting, a client may not want to give us all entries in the data matrix. In such a setting, our goal is to show that if the samples that you do get are chosen carefully, the top-k PCA features of the data can be recovered within some provable error bounds.

More formally, the data matrix is $\mathbf{A} \in \mathbb{R}^{m \times n}$ (m data points in n dimensions). Often, real data matrices have low effective rank, so let $\mathbf{A}_k$ be the best rank-k approxima-

---

This chapter has been submitted to: A. Kundu, P. Drineas, and M. Magdon-Ismail, "Recovering PCA from hybrid-(ℓ_1, ℓ_2) sparse sampling of data elements," 2015. [Online]. Available: http://arxiv.org/pdf/1503.00547v1.pdf (Date Last Accessed 06/14/2015).

tion to $\mathbf{A}$ with $\|\mathbf{A} - \mathbf{A}_k\|_2$ being small, where $\|\mathbf{X}\|_2$ is the spectral norm of matrix $\mathbf{X}$. $\mathbf{A}_k$ is obtained by projecting $\mathbf{A}$ onto the subspace spanned by its top-k principal components. In order to approximate this top-k principal subspace, we adopt the following strategy. Select a small number, s, of elements from $\mathbf{A}$ and produce a sparse sketch $\tilde{\mathbf{A}}$; use the sparse sketch $\tilde{\mathbf{A}}$ to approximate the top-k singular subspace. In Section 1.4, we give the details of the algorithm and the theoretical guarantees on how well we recover the top-k principal subspace. The key quantity that one must control to recover a close approximation to PCA is how well the sparse sketch approximates the data *in the operator norm*. That is, if $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ is small then we can recover PCA effectively.

Problem: sparse sampling of data elements

Given $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\epsilon > 0$, sample a small number of elements s to obtain a sparse sketch $\tilde{\mathbf{A}}$ for which

$$\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 \leq \epsilon \quad \text{and} \quad \|\tilde{\mathbf{A}}\|_0 \leq s. \tag{1.1}$$

Our main result addresses the problem above. In a nutshell, with only partially observed data that have been carefully selected, one can recover an approximation to the top-k principal subspace. An additional benefit is that computing our approximation to the top-k subspace using iterated multiplication can benefit computationally from sparsity. To construct $\tilde{\mathbf{A}}$, we use a general randomized approach which independently samples (and rescales) s elements from $\mathbf{A}$ using probability p_{ij} to sample element $\mathbf{A}_{ij}$. We analyze in detail the case $p_{ij} \propto \alpha|\mathbf{A}_{ij}| + (1-\alpha)|\mathbf{A}_{ij}|^2$ to get a bound on $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$. We now make our discussion precise, starting with our notation.

1.1.1 Notation

We use bold uppercase (e.g., $\mathbf{X}$) for matrices and bold lowercase (e.g., $\mathbf{x}$) for column vectors. The i-th row of $\mathbf{X}$ is $\mathbf{X}_{(i)}$, and the i-th column of $\mathbf{X}$ is $\mathbf{X}^{(i)}$. Let $[n]$ denote the set $\{1, 2, ..., n\}$. $\mathbb{E}(X)$ is the expectation of a random variable X; for a matrix, $\mathbb{E}(\mathbf{X})$ denotes the element-wise expectation. For a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, the Frobenius norm $\|\mathbf{X}\|_F$ is $\|\mathbf{X}\|_F^2 = \sum_{i,j=1}^{m,n} \mathbf{X}_{ij}^2$, and the spectral (operator) norm $\|\mathbf{X}\|_2$ is $\|\mathbf{X}\|_2 = \max_{\|\mathbf{y}\|_2=1} \|\mathbf{X}\mathbf{y}\|_2$. We also have the ℓ_1 and ℓ_0 norms: $\|\mathbf{X}\|_1 = \sum_{i,j=1}^{m,n} |\mathbf{X}_{ij}|$ and $\|\mathbf{X}\|_0$ (the number of non-zero entries in $\mathbf{X}$). The k-th largest singular value of $\mathbf{X}$ is

$\sigma_k(\mathbf{X})$. For symmetric matrices $\mathbf{X},\ \mathbf{Y}$, $\mathbf{Y} \succeq \mathbf{X}$ if and only if $\mathbf{Y} - \mathbf{X}$ is positive semi-definite. $\mathbf{I}_n$ is the $n \times n$ identity and $\ln x$ is the natural logarithm of x. We use $\mathbf{e}_i$ to denote standard basis vectors whose dimensions will be clear from the context.

Two popular sampling schemes are ℓ_1 ($p_{ij} = |\mathbf{A}_{ij}| / \|\mathbf{A}\|_1$ [1], [3]) and ℓ_2 ($p_{ij} = \mathbf{A}_{ij}^2 / \|\mathbf{A}\|_F^2$ [1], [4]). We construct $\tilde{\mathbf{A}}$ as follows: $\tilde{\mathbf{A}}_{ij} = 0$ if the (i,j)-th entry is not sampled; sampled elements $\mathbf{A}_{ij}$ are rescaled to $\tilde{\mathbf{A}}_{ij} = \mathbf{A}_{ij}/p_{ij}$ which makes the sketch $\tilde{\mathbf{A}}$ an unbiased estimator of $\mathbf{A}$, so $\mathbb{E}[\tilde{\mathbf{A}}] = \mathbf{A}$. The sketch is *sparse* if the number of sampled elements is sublinear, $s = o(mn)$. Sampling according to element magnitudes is natural in many applications, for example in a recommendation system users tend to rate a product they either like (high positive) or dislike (high negative).

Our main sparsification algorithm (Algorithm 1) receives as input a matrix $\mathbf{A}$ and an accuracy parameter $\epsilon > 0$, and samples s elements from $\mathbf{A}$ in s independent, identically distributed trials with replacement, according to a hybrid-(ℓ_1, ℓ_2) probability distribution specified in equation (1.2). The algorithm returns $\tilde{\mathbf{A}} \in \mathbb{R}^{m \times n}$, a sparse and unbiased estimator of $\mathbf{A}$, as a solution to (1.1).

1.1.2 Prior work

[1], [2] pioneered the idea of ℓ_2 sampling for element-wise sparsification. However, ℓ_2 sampling on its own is not enough for provably accurate bounds for $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$. As a matter of fact [1], [2] observed that "small" entries need to be sampled with probabilities that depend on their absolute values only, thus also introducing the notion of ℓ_1 sampling. The underlying reason for the need of ℓ_1 sampling is the fact that if a small element is sampled and rescaled using ℓ_2 sampling, this would result in a huge entry in $\tilde{\mathbf{A}}$ (because of the rescaling). As a result, the variance of ℓ_2 sampling is quite high, resulting in poor theoretical and experimental behavior. ℓ_1 sampling of small entries rectifies this issue by reducing the variance of the overall approach.

[5] proposed a sparsification algorithm that deterministically keeps large entries, i.e., entries of $\mathbf{A}$ such that $|\mathbf{A}_{ij}| \geq \epsilon/\sqrt{n}$ and randomly rounds the remaining entries using ℓ_1 sampling. Formally, entries of $\mathbf{A}$ that are smaller than $\epsilon\sqrt{n}$ are set to $\operatorname{sign}(\mathbf{A}_{ij})\, \epsilon/\sqrt{n}$ with probability $p_{ij} = \sqrt{n}\, |\mathbf{A}_{ij}| / \epsilon$ and to zero otherwise. They used an ϵ-net argument to show that $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ was bounded with high probability. [4] bypassed the need for ℓ_1

sampling by zeroing-out the small entries of $\mathbf{A}$ (e.g., all entries such that $|\mathbf{A}_{ij}| < \epsilon/2n$ for a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$) and then use ℓ_2 sampling on the remaining entries in order to sparsify the matrix. This simple modification improves [2] and [5], and comes with an elegant proof using the matrix-Bernstein inequality of [6]. Note that all these approaches need truncation of small entries. Recently, [3] showed that ℓ_1 sampling in isolation could be done without any truncation, and argued that (under certain assumptions) ℓ_1 sampling would be better than ℓ_2 sampling, even using the truncation. Their proof is also based on the matrix-valued Bernstein inequality of [6].

1.1.3 Our Contributions

We introduce an intuitive hybrid approach to element-wise matrix sparsification, by combining ℓ_1 and ℓ_2 sampling. We propose to use sampling probabilities of the form

$$p_{ij} = \alpha \cdot \frac{|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_1} + (1-\alpha)\frac{\mathbf{A}_{ij}^2}{\|\mathbf{A}\|_F^2}, \qquad \alpha \in (0,1] \tag{1.2}$$

for all i, j [1]. We essentially retain the good properties of ℓ_2 sampling that bias us towards data elements in the presence of small noise, while *regularizing* smaller entries using ℓ_1 sampling. The proof of the quality-of-approximation result of Algorithm 1 (i.e. Theorem 1) uses the matrix-Bernstein Lemma 1. We summarize the main contributions below:

• We give a parameterized sampling distribution in the variable $\alpha \in (0,1]$ that controls the balance between ℓ_2 sampling and ℓ_1 regularization. This greater flexibility allows us to achieve greater accuracy.

• We derive the optimal hybrid-(ℓ_1, ℓ_2) distribution, using Lemma 1 for arbitrary $\mathbf{A}$, by computing the optimal parameter α^* which produces the desired accuracy with smallest sample size according to our theoretical bound.

Our result generalizes the existing results because setting $\alpha = 1$ in our bounds reproduces the result of [3] who claim that ℓ_1 sampling is almost always better than ℓ_2 sampling. Our results show that $\alpha^* < 1$ which means that the hybrid approach is best.

• We give a provable algorithm (Algorithm 2) to implement hybrid-(ℓ_1, ℓ_2) sam-

[1] combining ℓ_1 and ℓ_2 probabilities to avoid zeroing out step of ℓ_2 sampling has recently been observed by [7].

pling without knowing α *a priori*, i.e., we need not 'fix' the distribution using some predetermined value of α at the beginning of the sampling process. We can set α at a later stage, yet we can realize hybrid-(ℓ_1, ℓ_2) sampling. We use Algorithm 2 to propose a pass-efficient element-wise sampling model using only one pass over the elements of the data $\mathbf{A}$, using $O(s)$ memory. Moreover, Algorithm 3 gives us a heuristic to estimate α^* in one-pass over the data using $O(s)$ memory.

• Finally, we propose the Algorithm 4 which provably recovers PCA by constructing a sparse unbiased estimator of (centered) data using our optimal hybrid-(ℓ_1, ℓ_2) sampling.

Experimental results suggest that our optimal hybrid distribution (using α^*) requires strictly smaller sample size than ℓ_1 and ℓ_2 sampling (with or without truncation) to solve (1.1). Also, we achieve significant speed up of PCA on sparsified synthetic and real data while maintaining high quality approximation.

1.1.3.1 A Motivating Example for Hybrid-(ℓ_1, ℓ_2) Sampling

The main motivation for introducing the idea of hybrid-(ℓ_1, ℓ_2) sampling on elements of $\mathbf{A}$ comes from achieving a tighter bound on s using a simple and intuitive probability distribution on elements of $\mathbf{A}$. For this, we observe certain good properties of both ℓ_1 and ℓ_2 sampling for sparsification of noisy data (in practice, we experience data that are noisy, and it is perhaps impossible to separate "true" data from noise). We illustrate the behavior of ℓ_1 and ℓ_2 sampling on noisy data using the following synthetic example. We construct a 500×500 binary data $\mathbf{D}$ (Figure 1.1), and then perturb it by a random Gaussian matrix $\mathbf{N}$ whose elements $\mathbf{N}_{ij}$ follow Gaussian distribution with mean zero and standard deviation 0.1. We denote this perturbed data matrix by $\mathbf{A}_{0.1}$. First,

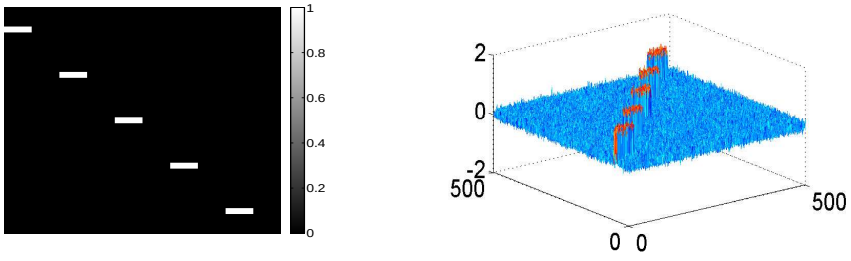
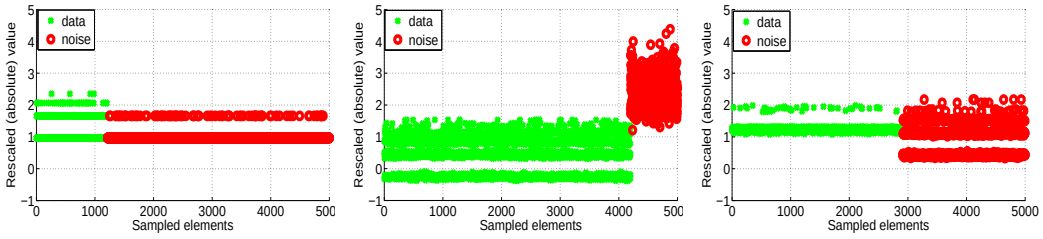


Figure 1.1: (*left*) Synthetic noiseless 500×500 binary data D; (*right*) mesh view of noisy data $\mathrm{A}_{0.1}$.

we note that ℓ_1 and ℓ_2 sampling work *identically* on binary data $\mathbf{D}$. However, Figure 1.2 depicts the change in behavior of ℓ_1 and ℓ_2 sampling sparsifying $\mathbf{A}_{0.1}$. Data elements and noise in $\mathbf{A}_{0.1}$ are the elements with non-zero and zero values in $\mathbf{D}$, respectively. We sample $s = 5000$ indices in i.i.d. trials according to ℓ_1 and ℓ_2 probabilities separately to produce sparse sketch $\tilde{\mathbf{A}}$. Figure 1.2 shows that elements of $\tilde{\mathbf{A}}$, produced by ℓ_1 sampling, have controlled variance but most of them are noise. On the other hand, ℓ_2 sampling is biased towards data elements, although small number of sampled noisy elements create large variance due to rescaling. Our hybrid-(ℓ_1, ℓ_2) sampling benefits from this bias of ℓ_2 towards data elements, as well as, regularization properties of ℓ_1.



(a) ℓ_1 **(b)** ℓ_2 **(c) Hybrid-**(ℓ_1, ℓ_2)

Figure 1.2: Elements of sparse sketch $\tilde{\mathbf{A}}$ produced from $\mathbf{A}_{0.1}$ via (a) ℓ_1 sampling, (b) ℓ_2 sampling, and (c) hybrid-(ℓ_1, ℓ_2) sampling with $\alpha = 0.7$. The y-axis plots the rescaled absolute values (in $\ln$ scale) of $\tilde{\mathbf{A}}$ corresponding to the sampled indices. ℓ_1 sampling produces elements with controlled variance but it mostly samples noise, whereas ℓ_2 samples a lot of data although producing large variance of rescaled elements. Hybrid-(ℓ_1, ℓ_2) sampling uses ℓ_1 as a regularizer while sampling a fairly large number of data that helps to preserve the structure of original data.

We parameterize our distribution using the variable $\alpha \in (0, 1]$ that controls the balance between ℓ_2 sampling and ℓ_1 regularization. We derive an expression to compute α^*, the optimal α, corresponding to the smallest sample size that we need in order to achieve a given accuracy ϵ in (1.1). Setting $\alpha = 1$, we reproduce the result of [3]. However, α^* may be smaller than 1, and the bound on sample size s, using α^*, is guaranteed to be tighter than that of [3].

1.2 Main Result

We present the quality-of-approximation result of our main algorithm (Algorithm 1). We define the sampling operator $\mathcal{S}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ in (1.3) that extracts elements from a given matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$. Let Ω be a multi-set of sampled indices (i_t, j_t), for $t = 1, ..., s$. Then,

$$\mathcal{S}_\Omega(\mathbf{A}) = \frac{1}{s}\sum_{t=1}^{s} \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t}\mathbf{e}_{j_t}^T, \quad (i_t, j_t) \in \Omega \tag{1.3}$$

Algorithm 1 randomly samples (in i.i.d. trials) s elements of a given matrix $\mathbf{A}$, according to a probability distribution $\{p_{ij}\}_{i,j=1}^{m,n}$ over the elements of $\mathbf{A}$. Let the p_{ij}'s be as in eqn. (1.2). Then, we can prove the following theorem.

Theorem 1 *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and let $\epsilon > 0$ be an accuracy parameter. Let $\mathcal{S}_\Omega$ be the sampling operator defined in (1.3), and assume that the multi-set Ω is generated using sampling probabilities $\{p_{ij}\}_{i,j=1}^{m,n}$ as in (1.2). Then, with probability at least $1-\delta$,*

$$\|\mathcal{S}_\Omega(\mathbf{A}) - \mathbf{A}\|_2 \le \epsilon \|\mathbf{A}\|_2, \tag{1.4}$$

if

$$s \ge \frac{2}{\epsilon^2 \|\mathbf{A}\|_2^2}\left(\rho^2(\alpha) + \gamma(\alpha)\epsilon\|\mathbf{A}\|_2/3\right)\ln\left(\frac{m+n}{\delta}\right) \tag{1.5}$$

where,

$$\xi_{ij} = \|\mathbf{A}\|_F^2 / \left(\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha)\right), \text{ for } \mathbf{A}_{ij} \neq 0,$$

$$\rho^2(\alpha) = \max\left\{\max_i \sum_{j=1}^{n}\xi_{ij}, \max_j \sum_{i=1}^{m}\xi_{ij}\right\} - \sigma_{min}^2(\mathbf{A}),$$

$$\gamma(\alpha) = \max_{\substack{i,j:\\ \mathbf{A}_{ij}\neq 0}}\left\{\frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\frac{\|\mathbf{A}\|_1\cdot|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2}}\right\} + \|\mathbf{A}\|_2,$$

$\sigma_{min}(\mathbf{A})$ is the smallest singular value of $\mathbf{A}$. Moreover, we can find α^* (optimal α corresponding to the smallest s) and s^* (the smallest s), by solving the following optimization

Algorithm 1 Element-wise Matrix Sparsification

1: **Input:** $\mathbf{A} \in \mathbb{R}^{m \times n}$, accuracy parameter $\epsilon > 0$.
2: **Set** s as in eq. (1.7).
3: **For** $t = 1 \dots s$ (i.i.d. trials with replacement) **randomly sample** pairs of indices $(i_t, j_t) \in [m] \times [n]$ with $\mathbb{P}\left[(i_t, j_t) = (i, j)\right] = p_{ij}$, where p_{ij} are as in (1.2), using α as in (1.6).
4: **Output**(sparse): $\mathcal{S}_\Omega\left(\mathbf{A}\right) = \frac{1}{s} \sum_{t=1}^{s} \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t} \mathbf{e}_{j_t}^T$.

problem in (1.6):

$$\alpha^* = \min_{\alpha \in (0,1]} f(\alpha), \quad f(\alpha) = \rho^2(\alpha) + \gamma(\alpha)\epsilon \left\|\mathbf{A}\right\|_2 / 3, \tag{1.6}$$

$$s^* = \frac{2}{\epsilon^2 \left\|\mathbf{A}\right\|_2^2} \left(\rho^2(\alpha^*) + \gamma(\alpha^*) \frac{\epsilon \left\|\mathbf{A}\right\|_2}{3}\right) \ln\left(\frac{m+n}{\delta}\right) \tag{1.7}$$

The functional form in (1.5) comes from the Matrix-Bernstein inequality in Lemma 1, with ρ^2 and γ being functions of $\mathbf{A}$ and α. This gives us a flexibility to optimize the sample size with respect to α in (1.5), which is how we get the optimal α^*. For a given matrix $\mathbf{A}$, we can easily compute $\rho^2(\alpha)$ and $\gamma(\alpha)$ for various values of α. Given an accuracy ϵ and failure probability δ, we can compute α^* corresponding to the tightest bound on s. Note that, for $\alpha = 1$ we reproduce the results of [3] (which was expressed using various matrix metrics). However, α^* may be smaller than 1, and is guaranteed to produce tighter s comparing to extreme choices of α (e.g. $\alpha = 1$ for ℓ_1 sampling). We illustrate this by the plot in Figure 1.3. We give a proof of Theorem 1 in Section 1.2.1.

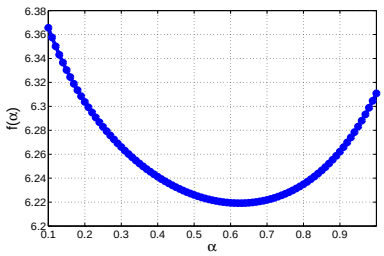


Figure 1.3: Plot of $f(\alpha)$ in eqn (1.6) for data $\mathbf{A}_{0.1}$. We use $\epsilon = 0.05$ and $\delta = 0.1$. x-axis plots α and y-axis is in $\log_{10}$ scale. For this data, $\alpha^* \approx 0.62$.

1.2.1 Proof of Theorem 1

In this section we provide a proof of Theorem 1 following the proof outline of [4], [3]. We use the following non-commutative matrix-valued Bernstein bound of [6] as our main tool to prove Theorem 1. Using our notation we rephrase the matrix Bernstein bound.

Lemma 1 *[Theorem 3.2 of [6]] Let* $\mathbf{M}_1, \mathbf{M}_2, ..., \mathbf{M}_s$ *be independent, zero-mean random matrices in* $\mathbb{R}^{m\times n}$*. Suppose*

$$\max_{t\in[s]} \left\{ \left\|\mathbb{E}(\mathbf{M}_t\mathbf{M}_t^T)\right\|_2, \left\|\mathbb{E}(\mathbf{M}_t^T\mathbf{M}_t)\right\|_2 \right\} \leq \rho^2$$

and $\|\mathbf{M}_t\|_2 \leq \gamma$ *for all* $t \in [s]$*. Then, for any* $\epsilon > 0$*,*

$$\left\| \frac{1}{s}\sum_{t=1}^{s} \mathbf{M}_t \right\|_2 \leq \epsilon$$

holds, subject to a failure probability at most

$$(m+n)\exp\left(\frac{-s\epsilon^2/2}{\rho^2+\gamma\epsilon/3}\right).$$

For all $t \in [s]$ we define the matrix $\mathbf{M}_t \in \mathbb{R}^{m\times n}$ as follows:

$$\mathbf{M}_t = \frac{\mathbf{A}_{i_tj_t}}{p_{i_tj_t}}\mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}.$$

It now follows that

$$\frac{1}{s}\sum_{t=1}^{s}\mathbf{M}_t = \frac{1}{s}\sum_{t=1}^{s}\left[\frac{\mathbf{A}_{i_tj_t}}{p_{i_tj_t}}\mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}\right] = S_\Omega(\mathbf{A}) - \mathbf{A}.$$

We can bound $\|\mathbf{M}_t\|_2$ for all $t \in [s]$. We define the following quantity:

$$\lambda = \frac{\|\mathbf{A}\|_1 \cdot |\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2}, \text{ for } \mathbf{A}_{ij} \neq 0 \tag{1.8}$$

Lemma 2 *Using our notation, and using probabilities of the form (1.2), for all* $t \in [s]$,

$$\|\mathbf{M}_t\|_2 \leq \max_{\substack{i,j: \\ \mathbf{A}_{ij}\neq 0}} \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\lambda} + \|\mathbf{A}\|_2 .$$

Proof: Using probabilities of the form (1.2), and because $\mathbf{A}_{ij} = 0$ is never sampled,

$$\|\mathbf{M}_t\|_2 = \left\| \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t} \mathbf{e}_{j_t}^T - \mathbf{A} \right\|_2 \leq \max_{\substack{i,j: \\ \mathbf{A}_{ij}\neq 0}} \left\{ \left(\frac{\alpha}{\|\mathbf{A}\|_1} + \frac{(1-\alpha)\cdot|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2} \right)^{-1} \right\} + \|\mathbf{A}\|_2$$

Using (1.8), we obtain the bound.

◇

Next we bound the spectral norm of the expectation of $\mathbf{M}_t\mathbf{M}_t^T$.

Lemma 3 *Using our notation, and using probabilities of the form (1.2), for all* $t \in [s]$,

$$\left\|\mathbb{E}(\mathbf{M}_t\mathbf{M}_t^T)\right\|_2 \leq \|\mathbf{A}\|_F^2 \beta_1 - \sigma_{min}^2(\mathbf{A}),$$

where,

$$\beta_1 = \max_i \sum_{j=1}^{n} \left(\frac{\alpha \cdot \|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1}, \text{ for } \mathbf{A}_{ij} \neq 0.$$

Proof: Recall that $\mathbf{A} = \sum_{i,j=1}^{m,n} \mathbf{A}_{ij}\mathbf{e}_i\mathbf{e}_j^T$ and $\mathbf{M}_t = \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}$ to derive

$$\begin{aligned} \mathbb{E}[\mathbf{M}_t\mathbf{M}_t^T] &= \sum_{i,j=1}^{m,n} p_{ij} \left(\frac{\mathbf{A}_{ij}}{p_{ij}} \mathbf{e}_i\mathbf{e}_j^T - \mathbf{A} \right) \left(\frac{\mathbf{A}_{ij}}{p_{ij}} \mathbf{e}_j\mathbf{e}_i^T - \mathbf{A}^T \right) \\ &= \sum_{i,j=1}^{m,n} \left(\frac{\mathbf{A}_{ij}^2}{p_{ij}} \mathbf{e}_i\mathbf{e}_i^T \right) - \mathbf{A}\mathbf{A}^T. \end{aligned}$$

Sampling according to probabilities of eqn. (1.2), and because $\mathbf{A}_{ij} = 0$ is never sampled, we get, for $\mathbf{A}_{ij} \neq 0$,

$$\begin{aligned} \sum_{i,j=1}^{m,n} \frac{\mathbf{A}_{ij}^2}{p_{ij}} &= \|\mathbf{A}\|_F^2 \sum_{i,j=1}^{m,n} \left(\frac{\alpha \cdot \|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1}, \\ &\leq \|\mathbf{A}\|_F^2 \sum_{i=1}^{m} \max_i \sum_{j=1}^{n} \left(\frac{\alpha \cdot \|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1}. \end{aligned}$$

Thus,

$$\mathbb{E}[\mathbf{M}_t\mathbf{M}_t^T] \preceq \|\mathbf{A}\|_F^2 \beta_1 \sum_{i=1}^{m} \mathbf{e}_i\mathbf{e}_i^T - \mathbf{A}\mathbf{A}^T = \|\mathbf{A}\|_F^2 \beta_1 \mathbf{I}_m - \mathbf{A}\mathbf{A}^T.$$

Note that, $\|\mathbf{A}\|_F^2 \beta_1 \mathbf{I}_m$ is a diagonal matrix with all entries non-negative, and $\mathbf{A}\mathbf{A}^T$ is a postive semi-definite matrix. Therefore,

$$\left\|\mathbb{E}[\mathbf{M}_t\mathbf{M}_t^T]\right\|_2 \leq \|\mathbf{A}\|_F^2 \beta_1 - \sigma_{min}^2(\mathbf{A}).$$

$\diamond$

Similarly, we can obtain

$$\left\|\mathbb{E}[\mathbf{M}_t^T\mathbf{M}_t]\right\|_2 \leq \|\mathbf{A}\|_F^2 \beta_2 - \sigma_{min}^2(\mathbf{A}),$$

where,

$$\beta_2 = \max_j \sum_{i=1}^{m} \left(\frac{\alpha \cdot \|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1}, \text{ for } \mathbf{A}_{ij} \neq 0.$$

We can now apply Theorem 1 with

$$\rho^2(\alpha) = \|\mathbf{A}\|_F^2 \max\{\beta_1, \beta_2\} - \sigma_{min}^2(\mathbf{A})$$

and

$$\gamma(\alpha) = \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\lambda} + \|\mathbf{A}\|_2$$

to conclude that $\|\mathcal{S}_\Omega(\mathbf{A}) - \mathbf{A}\|_2 \leq \varepsilon$ holds subject to a failure probability at most

$$(m+n) \exp\left((-s\varepsilon^2/2) / \left(\rho^2(\alpha) + \gamma(\alpha)\varepsilon/3 \right) \right).$$

Bounding the failure probability by δ, and setting $\varepsilon = \epsilon \cdot \|\mathbf{A}\|_2$, we complete the proof.

Algorithm 2 One-pass hybrid-(ℓ_1, ℓ_2) sampling

1: **Input:** $\mathbf{A}_{ij}$ for all $(i,j) \in [m] \times [n]$, arbitrarily ordered, and sample size s.
2: Apply SELECT algorithm in parallel with $O(s)$ memory using ℓ_1 probabilities to sample s independent indices (i_{t_1}, j_{t_1}) and corresponding elements $\mathbf{A}_{i_{t_1} j_{t_1}}$ to form random multiset S_1 of triples $(i_{t_1}, j_{t_1}, \mathbf{A}_{i_{t_1} j_{t_1}})$, for $t_1 = 1, ..., s$.
3: Run step 2 in parallel to form another independent multiset S_3 of triples $(i_{t_3}, j_{t_3}, \mathbf{A}_{i_{t_3} j_{t_3}})$, for $t_3 = 1, ..., s$. (This step is only for Algorithm 3)
4: Apply SELECT algorithm in parallel with $O(s)$ memory using ℓ_2 probabilities to sample s independent indices (i_{t_2}, j_{t_2}) and corresponding elements $\mathbf{A}_{i_{t_2} j_{t_2}}$ to form random multiset S_2 of triples $(i_{t_2}, j_{t_2}, \mathbf{A}_{i_{t_2} j_{t_2}})$, for $t_2 = 1, ..., s$.
5: Run step 4 in parallel to form another independent multiset S_4 of triples $(i_{t_4}, j_{t_4}, \mathbf{A}_{i_{t_4} j_{t_4}})$, for $t_4 = 1, ..., s$. (This step is only for Algorithm 3)
6: Compute and store $\|\mathbf{A}\|_F^2$ and $\|\mathbf{A}\|_1$ in parallel.
7: Set the value of $\alpha \in (0, 1]$ (using Algorithm 3).
8: Create empty multiset of triples S.
9: $\mathbf{X} \leftarrow \mathbf{0}_{m \times n}$.
10: **For** $t = 1 \ldots s$
11: Generate a uniform random number $x \in [0, 1]$.
12: if $x \geq \alpha$, $S(t) \leftarrow S_1(t)$; otherwise, $S(t) \leftarrow S_2(t)$.
13: $(i_t, j_t) \leftarrow S(t, 1:2)$.
14: $p \leftarrow \alpha \cdot \frac{|S(t,3)|}{\|\mathbf{A}\|_1} + (1-\alpha) \cdot \frac{|S(t,3)|^2}{\|\mathbf{A}\|_F^2}$
15: $\mathbf{X} \leftarrow \mathbf{X} + \frac{S(t,3)}{p \cdot s} e_{i_t} e_{j_t}^T$.
16: **End**
17: **Output:** random multiset S, and sparse matrix $\mathbf{X}$.

1.3 One-pass Hybrid-(ℓ_1, ℓ_2) Sampling

Here we discuss the implementation of (ℓ_1, ℓ_2)-hybrid sampling in one pass over the input matrix $\mathbf{A}$ using $O(s)$ memory, that is, a streaming model. We know that both ℓ_1 and ℓ_2 sampling can be done in one pass using $O(s)$ memory (see Algorithm SELECT p. 137 of [8]). In our hybrid sampling, we want parameter α to depend on data elements, i.e., we do not want to 'fix' it prior to the arrival of data stream. Here we give an algorithm (Algorithm 2) to implement a one-pass version of the hybrid sampling *without knowing* α *a priori*.

We note that steps 2-5 of Algorithm 2 access the elements of $\mathbf{A}$ only once, in parallel, to form independent multisets S_1, S_2, S_3, and S_4. Step 6 computes $\|\mathbf{A}\|_F^2$ and $\|\mathbf{A}\|_1$ in parallel in one pass over $\mathbf{A}$. Subsequent steps do not need to access $\mathbf{A}$ anymore. Interestingly, we set α in step 7 when the data stream is gone. Steps 10-16 sample s elements

from S_1 and S_2 based on the α in step 7, and produce sparse matrix $\mathbf{X}$ based on the sampled entries in random multiset S. Theorem 2 shows that Algorithm 2 indeed samples elements from $\mathbf{A}$ according to the hybrid-(ℓ_1, ℓ_2) probabilities in eqn (1.2).

Theorem 2 *Using the notations in Algorithm 2, for* $\alpha \in (0, 1]$, $t = 1, ..., s$,

$$P\left[S(t) = (i, j, \mathbf{A}_{ij})\right] = \alpha \cdot p_1 + (1 - \alpha) \cdot p_2,$$

where $\quad p_1 = \frac{|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_1} \quad$ *and* $\quad p_2 = \frac{\mathbf{A}_{ij}^2}{\|\mathbf{A}\|_F^2}$.

Proof: Here we use the notations in Theorem 2. Note that t-th elements of S_1 and S_2 are sampled independently with ℓ_1 and ℓ_2 probabilities, respectively. We consider the following disjoint events:

$$\begin{aligned}
&\mathcal{E}_1 : S_1(t) = (i, j, \mathbf{A}_{ij}) \wedge S_2(t) \neq (i, j, \mathbf{A}_{ij}) \\
&\mathcal{E}_2 : S_1(t) \neq (i, j, \mathbf{A}_{ij}) \wedge S_2(t) = (i, j, \mathbf{A}_{ij}) \\
&\mathcal{E}_3 : S_1(t) = (i, j, \mathbf{A}_{ij}) \wedge S_2(t) = (i, j, \mathbf{A}_{ij}) \\
&\mathcal{E}_4 : S_1(t) \neq (i, j, \mathbf{A}_{ij}) \wedge S_2(t) \neq (i, j, \mathbf{A}_{ij})
\end{aligned}$$

Let us denote the events $x_1 : x \geq \alpha$ and $x_2 : x < \alpha$. Clearly, $P\left[x_1\right] = \alpha, P\left[x_2\right] = 1 - \alpha$. Since the elements $S_1(t)$ and $S_2(t)$ are sampled independently, we have

$$\begin{aligned}
P\left[\mathcal{E}_1\right] &= P\left[S_1(t) = (i, j, \mathbf{A}_{ij})\right] P\left[S_2(t) \neq (i, j, \mathbf{A}_{ij})\right] = p_1(1 - p_2) \\
P\left[\mathcal{E}_2\right] &= (1 - p_1)p_2 \\
P\left[\mathcal{E}_3\right] &= p_1 p_2 \\
P\left[\mathcal{E}_4\right] &= (1 - p_1)(1 - p_2)
\end{aligned}$$

We note that α may be dependent on the elements of S_3 and S_4 (in Algorithm 3), but is independent of elements of S_1 and S_2. Therefore, events x_1 and x_2 are independent of the

events $\mathcal{E}_j$, $j = 1, 2, 3, 4$. Thus,

$$\begin{aligned} & P\left[S(t) = (i, j, \mathbf{A}_{ij})\right] \\ = \; & P\left[(\mathcal{E}_1 \wedge x_1) \vee (\mathcal{E}_2 \wedge x_2) \vee \mathcal{E}_3\right] \\ = \; & P\left[\mathcal{E}_1 \wedge x_1\right] + P\left[\mathcal{E}_2 \wedge x_2\right] + P\left[\mathcal{E}_3\right] \\ = \; & P\left[\mathcal{E}_1\right] P\left[x_1\right] + P\left[\mathcal{E}_2\right] P\left[x_2\right] + P\left[\mathcal{E}_3\right] \\ = \; & p_1(1-p_2)\alpha + (1-p_1)p_2(1-\alpha) + p_1 p_2 \\ = \; & \alpha \cdot p_1 + (1-\alpha) \cdot p_2 \end{aligned}$$

$\diamond$

Note that, Theorem 2 holds for any arbitrary $\alpha \in (0, 1]$ in line 7 of Algorithm 2, i.e., Algorithm 3 is not essential for correctness of Theorem 2. We only need α to be independent of elements of S_1 and S_2. However, we use Algorithm 3 to get an iterative estimate of α^* (Section 1.3.1) in one pass over $\mathbf{A}$. In this case, we need additional independent multisets S_3 and S_4 to 'learn' the parameter α^*. Algorithm 2 (without Algorithm 3) requires a memory twice as large required by ℓ_1 or ℓ_2 sampling. Using Algorithm 3 this requirement is four times as large. However, in both the cases the asymptotic memory requirement remains the same $O(s)$.

1.3.1 Iterative Estimate of α^*

We obtain independent random multiset of triples S_3 and S_4, each containing s elements from $\mathbf{A}$ in one pass, in Algorithm 2. We can create a sparse random matrix $\mathbf{X}$, as shown in step 11 in Algorithm 3, that is an unbiased estimator of $\mathbf{A}$. We use this $\mathbf{X}$ as a proxy for $\mathbf{A}$ to estimate the quantities we need in order to solve the optimization problem in (1.9).

$$\tilde{\alpha} : \min_{\alpha \in (0,1]} \left\{ \left(\tilde{\rho}^2(\alpha) + \tilde{\gamma}(\alpha) \epsilon \left\| \mathbf{X} \right\|_2 / 3 \right) \right\} \tag{1.9}$$

Algorithm 3 Iterative estimate of α^*

Input: Multiset of triples S_3 and S_4 with s elements each, number of iteration τ, accuracy ϵ, $\|\mathbf{A}\|_F^2$, and $\|\mathbf{A}\|_1$.
Create empty multiset of triples S.
$\alpha_0 = 0.5$
For $k = 1 \dots \tau$
  $\mathbf{X} \leftarrow \mathbf{0}_{m\times n}$.
  For $t = 1 \dots s$
    Generate a uniform random number $x \in [0, 1]$.
    If $x \geq \alpha_{k-1}$, $S(t) \leftarrow S_3(t)$; else, $S(t) \leftarrow S_4(t)$.
    $(i_t, j_t) \leftarrow S(t, 1:2)$.
    $p \leftarrow \alpha_{k-1} \cdot \frac{|S(t,3)|}{\|\mathbf{A}\|_1} + (1-\alpha_{k-1}) \cdot \frac{|S(t,3)|^2}{\|\mathbf{A}\|_F^2}$
    $\mathbf{X} \leftarrow \mathbf{X} + \frac{S(t,3)}{p\cdot s} e_{i_t} e_{j_t}^T$.
  End
  $\alpha_k \leftarrow \tilde{\alpha}$ in (1.9) using $\mathbf{X}$.
End
Output: α_τ.

where, for all $(i, j) \in S(:, 1:2)$

$$\tilde{\xi}_{ij} = \|\mathbf{X}\|_F^2 / \left(\frac{\alpha \cdot \|\mathbf{X}\|_F^2}{|\mathbf{X}_{ij}| \cdot \|\mathbf{X}\|_1} + (1-\alpha) \right),$$

$$\tilde{\rho}^2(\alpha) = \max \left\{ \max_i \sum_{j=1}^{n} \tilde{\xi}_{ij}, \max_j \sum_{i=1}^{m} \tilde{\xi}_{ij} \right\},$$

$$\tilde{\gamma}(\alpha) = \max_{ij} \left\{ \frac{\|\mathbf{X}\|_1}{\alpha + (1-\alpha)\frac{\|\mathbf{X}\|_1 \cdot |\mathbf{X}_{ij}|}{\|\mathbf{X}\|_F^2}} \right\} + \|\mathbf{X}\|_F .$$

We note that $\|\mathbf{X}\|_0 \leq s$. We can compute the quantities $\tilde{\rho}(\alpha)$ and $\tilde{\gamma}(\alpha)$, for a fixed α, using $O(s)$ memory. We consider $\varepsilon = \epsilon \cdot \|\mathbf{X}\|_2$ to be the given accuracy.

1.4 Fast Approximation of PCA

Here, we discuss a provable algorithm (Algorithm 4) to speed up computation of PCA applying element-wise sampling. We sparsify a given centered data $\mathbf{A}$ to produce a sparse unbiased estimator $\tilde{\mathbf{A}}$ by sampling s elements in i.i.d. trials according to our hybrid-(ℓ_1, ℓ_2) distribution in (1.2). Computation of rank-truncated SVD on sparse data

Algorithm 4 Fast Approximation of PCA

1: **Input:** Centered data $\mathbf{A} \in \mathbb{R}^{m \times n}$, sparsity parameter $s > 0$, and rank parameter k.
2: Produce sparse unbiased estimator $\tilde{\mathbf{A}}$ from $\mathbf{A}$, in s i.i.d. trials using Algorithm 1.
3: Perform rank truncated SVD on sparse matrix $\tilde{\mathbf{A}}$, i.e., $[\tilde{\mathbf{U}}_k, \tilde{\mathbf{D}}_k, \tilde{\mathbf{V}}_k]$ = SVD($\tilde{\mathbf{A}}$, k).
4: **Output:** $\tilde{\mathbf{V}}_k$ (columns of $\tilde{\mathbf{V}}_k$ are the ordered PCA's).

is fast, and we consider the right singular vectors of $\tilde{\mathbf{A}}$ as the approximate principal components of $\mathbf{A}$. Naturally, more samples produce better approximation. However, this reduces sparsity, and consequently we lose the speed advantage. Theorem 3 shows the quality of approximation of principal components produced by Algorithm 4.

Theorem 3 *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a given matrix, and $\tilde{\mathbf{A}}$ be a sparse sketch produced by Algorithm 1. Let $\tilde{\mathbf{V}}_k$ be the PCA's of $\tilde{\mathbf{A}}$ computed in step 3 of Algorithm 4. Then*

$$\begin{aligned}
\left\|\mathbf{A} - \mathbf{A}\tilde{\mathbf{V}}_k\tilde{\mathbf{V}}_k^T\right\|_F^2 &\leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + \frac{4\|\mathbf{A}_k\|_F^2}{\sigma_k(\mathbf{A})}\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_2 \\
\left\|\mathbf{A}_k - \tilde{\mathbf{A}}_k\right\|_F &\leq \sqrt{8k}\cdot\left(\|\mathbf{A} - \mathbf{A}_k\|_2 + \left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_2\right) \\
\left\|\mathbf{A} - \tilde{\mathbf{A}}_k\right\|_F &\leq \|\mathbf{A} - \mathbf{A}_k\|_F + \sqrt{8k}\cdot\left(\|\mathbf{A} - \mathbf{A}_k\|_2 + \left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_2\right)
\end{aligned}$$

The first inequality of Theorem 3 bounds the approximation of projected data onto the space spanned by top k approximate PCA's. The second and third inequalities measure the quality of $\tilde{\mathbf{A}}_k$ as a surrogate for $\mathbf{A}_k$ and the quality of projection of sparsified data onto approximate PCA's, respectively.

Proofs of first two inequalities of Theorem 3 follow from Theorem 5 and Theorem 8 of [1], respectively. The last inequality follows from the triangle inequality. The last two inequalities above are particularly useful in cases where $\mathbf{A}$ is inherently low-rank and we choose an appropriate k for approximation, for which $\|\mathbf{A} - \mathbf{A}_k\|_2$ is small.

1.5 Experiments

In this section we perform various element-wise sampling experiments on synthetic and real data to show how well the sparse sketches preserve the structure of the original data, in spectral norm. Also, we show results on the quality of the PCA's derived from sparse sketches.

1.5.1 Algorithms for Sparse Sketches

We use Algorithm 1 as a prototypical algorithm to produce sparse sketches from a given matrix via various sampling methods. Note that, we can plug-in any element-wise probability distribution in Algorithm 1 to produce (unbiased) sparse matrices. We construct sparse sketches via our optimal hybrid-(ℓ_1, ℓ_2) sampling, along with other sampling methods related to extreme choices of α, such as, ℓ_1 sampling for $\alpha = 1$. Also, we use *element-wise leverage scores* ([9]) for sparsification of *low-rank* data. Element-wise leverage scores are used in the context of *low-rank matrix completion* by [9]. Let $\mathbf{A}$ be a $m \times n$ matrix of rank ρ, and its SVD if given by $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Then, we define μ_i (row leverage scores), ν_j (column leverage scores), and element-wise leverage scores p_{lev} as follows:

$$\mu_i = \left\|\mathbf{U}_{(i)}\right\|_2^2, \quad \nu_j = \left\|\mathbf{V}_{(j)}\right\|_2^2, \quad p_{lev} = \frac{\mu_i + \nu_j}{(m+n)\rho}, \quad i \in [m], j \in [n]$$

Note that p_{lev} is a probability distribution on the elements of $\mathbf{A}$. Leverage scores become uniform if the matrix $\mathbf{A}$ is full rank. We use p_{lev} in Algorithm 1 to produce sparse sketch $\tilde{\mathbf{A}}$ of a low-rank data $\mathbf{A}$.

1.5.1.1 Experimental Design for Sparse Sketches

We compute the theoretical optimal mixing parameter α^* by solving eqn (1.6) [2] for various datasets. We compare this α^* with the theoretical condition derived by [3] (for cases when ℓ_1 sampling outperforms ℓ_2 sampling). We verify the accuracy of α^* by measuring the quality of the sparse sketches $\tilde{\mathbf{A}}$, $\mathcal{E} = \|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \|\mathbf{A}\|_2$ for distributions corresponding to various α, for a given sample size s [3]. Let $\mathcal{E}_h$, $\mathcal{E}_1$, and $\mathcal{E}_{lev}$ denote the quality of sparse sketches produced via optimal hybrid sampling, ℓ_1 sampling, and element-wise leverage scores p_{lev}, respectively. We compare $\mathcal{E}_h$, $\mathcal{E}_1$, and $\mathcal{E}_{lev}$ for various sample sizes for real and synthetic datasets.

[2] we find α^* from the plot of $f(\alpha)$ for $\alpha \in [0.1, 1]$.

[3] here we replace ϵ by s as an input to Algorithm 1.

1.5.2 Algorithms for Fast PCA

We compare three algorithms for computing PCA of the centered data. Let the actual PCA of the original data be $\mathcal{A}$. We use Algorithm 4 to compute approximate PCA via our optimal hybrid-(ℓ_1, ℓ_2) sampling. Let us denote this approximate PCA by $\mathcal{H}$. Also, we compute PCA of a Gaussian random projection of the original data to compare the quality of $\mathcal{H}$. Let $\mathbf{A}_G = \mathbf{GA} \in \mathbb{R}^{r \times n}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the original data, and $\mathbf{G}$ is a $r \times m$ standard Gaussian matrix. Let the PCA of this random projection $\mathbf{A}_G$ be $\mathcal{G}$. Also, let T_a, T_h, and T_G be the computation time (in milliseconds) for $\mathcal{A}$, $\mathcal{H}$, and $\mathcal{G}$, respectively.

1.5.2.1 Experimental Design for Fast PCA

We compare the visual quality of $\mathcal{A}$, $\mathcal{H}$, and $\mathcal{G}$ for image datasets. Also, we compare the computation time T_a, T_h, and T_G for these datasets.

1.5.3 Description of Data

In this section we describe the synthetic and real datasets we use in our experiments.

1.5.3.1 Synthetic Data

We construct a binary 500×500 image data $\mathbf{D}$ (see Figure 1.1). We add random noise to perturb the elements of the 'pure' data $\mathbf{D}$. Specifically, we construct a 500×500 noise matrix $\mathbf{N}$ whose elements $\mathbf{N}_{ij}$ are drawn i.i.d from Gaussian with mean zero and standard deviation σ (we use $\sigma = 0.10$). We note the following ratios:

$$\text{Noise-to-signal energy ratio} = \|\mathbf{N}\|_F / \|\mathbf{D}\|_F,$$

$$\text{Spectral ratio} = \|\mathbf{N}\|_2 / \sigma_k(\mathbf{D}),$$

where $\sigma_k(\mathbf{D})$ is the k-th largest singular value of $\mathbf{D}$. For $\sigma = 0.10$, average Noise-to-signal energy ratio is 0.88, average Spectral ratio is 0.17, and average maximum absolute value of noise turn out to be 0.50. We denote this noisy data by $\mathbf{A}_{0.1}$.

1.5.3.2 TechTC Datasets

These datasets ([10]) are bag-of-words features for document-term data describing two topics (ids). We choose four such datasets: TechTC1 with ids 10567 and 11346, TechTC2 with ids 10567 and 12121, TechTC3 with ids 11498 and 14517, TechTC4 with ids 11346 and 22294. Rows represent documents and columns are the words. We preprocessed the data by removing all the words of length four or smaller, and then normalized the rows by dividing each row by its Frobenius norm. The following table lists the dimension of the TechTC datasets.

Table 1.1: Dimension of TechTC datasets.

Dimension ($m \times n$)	m	n
TechTC1	139	15170
TechTC2	138	11859
TechTC3	125	15485
TechTC4	125	14392

1.5.3.3 Handwritten Digit Data

A dataset ([11]) of three handwritten digits: six (664 samples), nine (644 samples), and one (1005 samples). Pixels are treated as features, and pixel values are normalized in [-1,1]. Each 16×16 digit image is first represented by a column vector by appending the pixels column-wise. Then, we use the transpose of this column vector to form a row in the data matrix. The number of rows $m = 2313$, and columns $n = 256$.

1.5.3.4 Stock Data

We use a stock market dataset (S&P) containing prices of 1218 stocks collected between 1983 and 2011. This temporal dataset has 7056 snapshots of stock prices. Thus, we have $m = 1218$ and $n = 7056$.

We provide summary statistics for all the datasets in Table 1.2. In order to compare our results with [3] we review the matrix metrics that they use. Let the numeric density of matrix $\mathbf{X}$ be $\text{nd}(\mathbf{X}) = \|\mathbf{X}\|_1^2 / \|\mathbf{X}\|_F^2$. Clearly, $\text{nd}(\mathbf{X}) \leq \|\mathbf{X}\|_0$, with equality holding

for zero-one matrices. The row density skew of $\mathbf{X}$ is defined as

$$\mathrm{rs}_0(\mathbf{X}) = \frac{\max_i \left\|\mathbf{X}_{(i)}\right\|_0}{\|\mathbf{X}\|_0 / m},$$

i.e., the ratio between number of non-zeros in the densest row and the average number of non-zeros per row. The numeric row density skew,

$$\mathrm{rs}_1(\mathbf{X}) = \frac{\max_i \left\|\mathbf{X}_{(i)}\right\|_1}{\|\mathbf{X}\|_1 / m},$$

is a smooth analog of $\mathrm{rs}_0(\mathbf{X})$. [3] assumed that $m \leq n$ without loss of generality, and for simplicity, $\max_i \left\|\mathbf{X}_{(i)}\right\|_\xi \geq \max_i \left\|\mathbf{X}^{(i)}\right\|_\xi$, for all $\xi \in \{0, 1, 2\}$. We notice that, although the Digit dataset does not satisfy the above conditions, its transpose does. We can work on the transposed dataset without loss of generality, and hence we take note of rs_0 and rs_1 of the transposed Digit data.

Table 1.2: Summary statistics for the data sets.

	$\|\mathbf{X}\|_0$	nd	rs_0	rs_1
$\mathbf{A}_{0.10}$	2.5e+5	9.2e+4	1	1.95
TechTC1	37831	12204	5.14	2.18
TechTC2	29334	9299	3.60	2.10
TechTC3	47304	14201	7.23	2.31
TechTC4	35018	10252	4.99	2.25
Digit	5.9e+5	5.1e+5	1	1.3
Stock	5.5e+6	6.5e+3	1.56	1.1e+03

1.5.4 Results

We report all the results based on an average of five independent trials. We observe a small variance of the results.

1.5.4.1 Quality of Sparse Sketch

We first note that three sampling methods ℓ_1, ℓ_2, and hybrid-(ℓ_1, ℓ_2), perform identically on noiseless data $\mathbf{D}$. We report the total probability of sampling noisy elements in

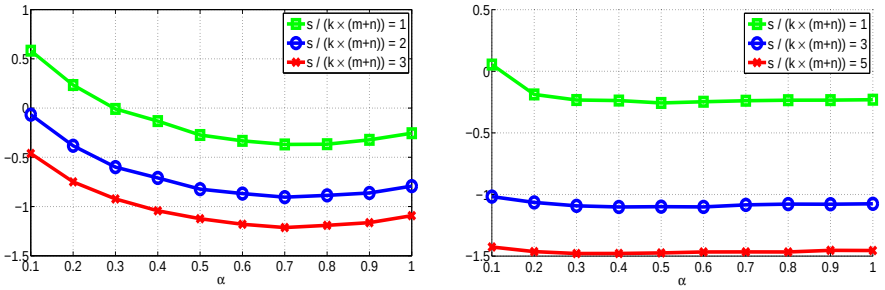
$\mathbf{A} = \mathbf{D} + \mathbf{N}$ (elements which are zeros in $\mathbf{D}$). ℓ_1 sampling shows the highest susceptibility to noise, whereas, small-valued noisy elements are suppressed in ℓ_2. Hybrid-(ℓ_1, ℓ_2) sampling, with $\alpha < 1$, samples mostly from true data elements, and thus captures the low-rank structure of the data better than ℓ_1. The optimal mixing parameter α^* maintains the right balance between ℓ_2 sampling and ℓ_1 regularization and gives the smallest sample size to achieve a desired accuracy. Table 1.3 summarizes α^* for various data sets. [3] argued that, as long as $\text{rs}_0(\mathbf{X}) \geq \text{rs}_1(\mathbf{X})$, ℓ_1 sampling is better than ℓ_2 (even with truncation). Our results on α^* in Table 1.3 confirm this condition. Moreover, our method can derive the right blend of ℓ_1 and ℓ_2 sampling even when the above condition fails. In this sense, we generalize the results of [3].

Table 1.3: α^* for various data sets ($\epsilon = 0.05$ is the desired relative-error accuracy). The last column compares α^* with the condition established by [3]. Whenever $\text{rs}_0 \geq \text{rs}_1$, [3] show that ℓ_1 sampling is always better than ℓ_2 sampling, and we find $\alpha^* = 1$ (ℓ_1 sampling). However, when $\text{rs}_0 < \text{rs}_1$, $\alpha^* < 1$ and our hybrid sampling is strictly better than both ℓ_1 and ℓ_2 sampling.

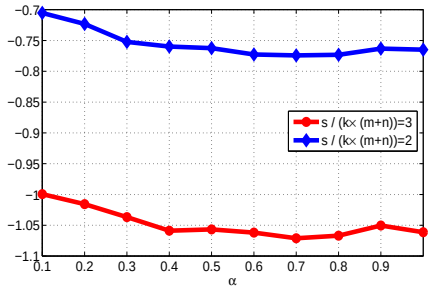
	ϵ	$\text{rs}_0 \geq \text{rs}_1$
$\mathbf{A}_{0.1}$	0.63	no
TechTC1	1	yes
TechTC2	1	yes
TechTC3	1	yes
TechTC4	1	yes
Digit	0.20	no
Stock	0.74	no

Figure 1.4 plots $\mathcal{E} = \|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \|\mathbf{A}\|_2$ for various values of α and sample size s for various datasets. It clearly shows our optimal hybrid sampling is superior to ℓ_1 or ℓ_2 sampling.

We also compare the quality of sparse sketches produced via our hybrid sampling with that of ℓ_2 sampling with truncation. We use two predetermined truncation parameters, $\varepsilon = 0.1$ and $\varepsilon = 0.01$, for ℓ_2 sampling. First, ℓ_2 sampling without truncation turns out to be the worst for all datasets. For real datasets, hybrid sampling with α^* outperforms ℓ_2 with $\varepsilon = 0.01$ and $\varepsilon = 0.1$. For $\mathbf{A}_{0.1}$, ℓ_2 with $\varepsilon = 0.01$ appears to produce sparse sketch $\tilde{\mathbf{A}}$ that is as bad as ℓ_2 without truncation. However, ℓ_2 with $\varepsilon = 0.1$ shows better performance than hybrid sampling (for $\mathbf{A}_{0.1}$ only), because this choice of ε turns out to be



(a) $\mathbf{A}_{0.1}$ **(b) Digit**



(c) Stock

Figure 1.4: Approximation quality of sparse sketch $\tilde{\mathbf{A}}$: hybrid-(ℓ_1, ℓ_2) sampling, for various α and different sample size s, are shown. x-axis is α, and y-axis plots $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2/\|\mathbf{A}\|_2$ (in $\log_2$ scale such that larger negative values indicate better quality). Each figure corresponds to a dataset: (a) $\mathbf{A}_{0.1}$, (b) Digit, and (c) Stock. We set $k = 5$ for synthetic data, $k = 3$ for Digit data, and $k = 1$ for Stock data. Choice of k is close to the stable rank of the data.

an appropriate threshold to zero-out most of the noisy elements. We must point out that, in this example, we control the noise for $\mathbf{A}_{0.1}$, and we know what a good threshold may look like. However, in reality we have no control over the noise. Therefore, choosing the right threshold for ℓ_2, without any prior knowledge, is an improbable task.

We compare the quality of Algorithm 3 producing an iterative estimate of α^* in a very restricted set up, i.e., one pass over the elements of data using $O(s)$ memory. Table 1.4 lists $\tilde{\alpha}$, the estimated α^*, for some of the datasets, for two choices of s using 10 iterations. We compare these values with the plots in Figure 1.4 where the results are generated without any restriction of size of memory or number of pass over the elements of the datasets.

Finally, we compare our hybrid-(ℓ_1, ℓ_2) sampling with *element-wise leverage score* sampling (similar to [9]) to produce quality sparse sketches from low-rank matrices. For

Table 1.4: Values of $\tilde{\alpha}$ (estimated α^* using Algorithm 3) for various data sets using one pass over the elements of data and $O(s)$ memory. We use $\epsilon = 0.05$, $\delta = 0.1$.

	$\frac{s}{k\cdot(m+n)} = 2$	$\frac{s}{k\cdot(m+n)} = 3$
$\mathbf{A}_{0.1}, k = 5$	0.55	0.5
Digit, $k = 3$	0.69	0.89
Stock, $k = 1$	1	1

this, we construct a 500×500 low-rank *power-law* matrix, similar to [9], as follows: $\mathbf{A}_{pow} = \mathbf{DXY}^T\mathbf{D}$, where, matrices $\mathbf{X}$ and $\mathbf{Y}$ are 500×5 i.i.d. Gaussian $\mathcal{N}(0, 1)$ and $\mathbf{D}$ is a diagonal matrix with power-law decay, $\mathbf{D}_{ii} = i^{-\gamma}$, $1 \leq i \leq 500$. The parameter γ controls the 'incoherence' of the matrix, i.e., larger values of γ makes the data more 'spiky'. Table 1.5 lists the quality of sparse sketches produced via the two sampling methods.

Table 1.5: Sparsification quality $\|\mathbf{A}_{pow} - \tilde{\mathbf{A}}_{pow}\|_2/\|\mathbf{A}_{pow}\|_2$ for low-rank 'power-law' matrix $\mathbf{A}_{pow}$ ($k = 5$). We compare the quality of hybrid-(ℓ_1, ℓ_2) sampling and leverage score sampling for two sample sizes. We note (average) α^* of hybrid-(ℓ_1, ℓ_2) distribution for data $\mathbf{A}_{pow}$ using $\epsilon = 0.05, \delta = 0.1$. For $\gamma = 0.5, 0.8, 1.0$, we have $\alpha^* = 0.11, 0.72, 0.8$, respectively.

	$\frac{s}{k(m+n)}$	hybrid-(ℓ_1, ℓ_2)	p_{lev}
$\gamma = 0.5$	3	42%	58%
	5	31%	43%
$\gamma = 0.8$	3	15%	43%
	5	12%	40%
$\gamma = 1.0$	3	8%	42%
	5	6%	39%

We note that, with increasing γ leverage scores get more aligned with the structure of the data, resulting in gradually improving approximation quality, for the same sample size. Larger γ produces more variance in data elements. ℓ_2 component of our hybrid distribution bias us towards the larger data elements, while ℓ_1 works as a regularizer to maintain the variance of the sampled (and rescaled) elements. With increasing γ we need more regularization to counter the problem of rescaling. Interestingly, our optimal parameter α^* adapts itself with this changing structure of data, e.g. for $\gamma = 0.5, 0.8, 1.0$,

we have $\alpha^* = 0.11, 0.72, 0.8$, respectively. This shows the benefit of our parameterized hybrid distribution to achieve a superior approximation quality. Figure 1.5 shows the structure of the data $\mathbf{A}_{pow}$ for $\gamma = 1.0$ along with the optimal hybrid-(ℓ_1, ℓ_2) distribution and leverage score distribution p_{lev}. The figure suggests our optimal hybrid distribution is better aligned with the structure of the data, requiring smaller sample size to achieve a desired sparsification accuracy.

We also compare the performance of the two sampling methods, optimal hybrid and leverage scores, on rank-truncated Digit data. It turns out that projection of Digit data onto top three principal components preserve the separation of digit categories. Therefore, we rank-truncate Digit data via SVD using rank three. Table 1.6 shows the superior quality of sparse sketches produced via optimal hybrid-(ℓ_1, ℓ_2) sampling for this rank-truncated digit data.

Table 1.6: Sparsification quality $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2/\|\mathbf{A}\|_2$ for rank-truncated Digit matrix ($k = 3$). We compare the optimal hybrid-(ℓ_1, ℓ_2) sampling and leverage score sampling for two sample sizes.

	Hybrid-(ℓ_1, ℓ_2)	p_{lev}
$\frac{s}{k(m+n)} = 3$	44%	61%
$\frac{s}{k(m+n)} = 5$	34%	47%

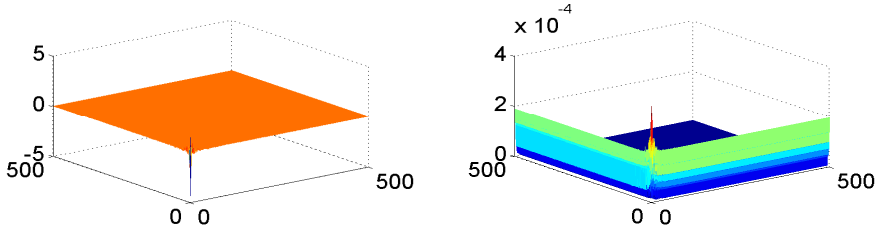
Finally, Table 1.7 shows the superiority of optimal hybrid-(ℓ_1, ℓ_2) sampling for rank-truncated (rank 5) $\mathbf{A}_{0.1}$ matrix for matrix sparsification.

Table 1.7: Sparsification quality $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2/\|\mathbf{A}\|_2$ for rank-truncated $\mathbf{A}_{0.1}$ matrix ($k = 5$). We compare the optimal hybrid-(ℓ_1, ℓ_2) sampling and leverage score sampling using two sample sizes.

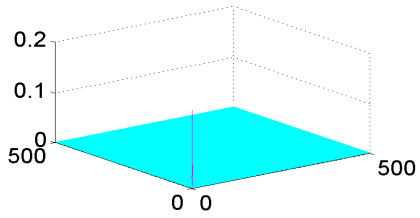
	Hybrid-(ℓ_1, ℓ_2)	p_{lev}
$\frac{s}{k(m+n)} = 3$	25%	80%
$\frac{s}{k(m+n)} = 5$	21%	62%

1.5.4.2 Quality of Fast PCA

We investigate the quality of fast PCA approximation (Algorithm 4) for Digit data and $\mathbf{A}_{0.1}$. We set $r = 30 \cdot k$ for the random projection matrix $\mathbf{A}_G$ to achieve a comparable



(a) Low-rank data $\mathbf{A}_{pow}$ **(b) Element-wise leverage scores for $\mathbf{A}_{pow}$**



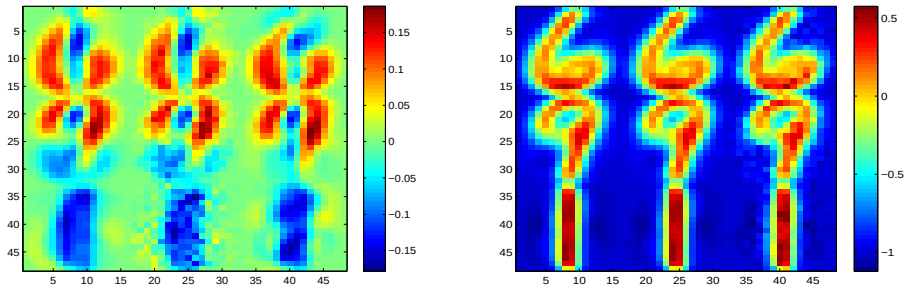
(c) Optimal hybrid distribution for $\mathbf{A}_{pow}$

Figure 1.5: Comparing optimal hybrid-(ℓ_1, ℓ_2) distribution with leverage scores p_{lev} for data $\mathbf{A}_{pow}$ for $\gamma = 1.0$. (a) Structure of $\mathbf{A}_{pow}$, (b) distribution p_{lev}, (c) optimal hybrid-(ℓ_1, ℓ_2) distribution. Our optimal hybrid distribution is more aligned with the structure of the data, requiring much smaller sample size to achieve a given accuracy of sparsification. This is supported by Table 1.5.

runtime of $\mathcal{G}$ with $\mathcal{H}$. Figure 1.6a shows the PCA (exact and approximate) for Digit data. Also, we consider visualization of the projected data onto top three principal components (exact and approximate) in Figure 1.6b. In Figure 1.6b, we form an average digit for each digit category by taking the average of pixel intensities in the projected data over all the digit samples in each category. Similarly, Figure 1.7 shows the visual results for data $\mathbf{A}_{0.1}$ (we set $k = 5$). Finally, Table 1.8 lists the gain in computation time for Algorithm 4 due to sparsification.

Table 1.8: Computational gain of Algorithm 4 comparing to exact PCA. We report the computation time of MATLAB function 'svds(A,k)' for actual data (T_a), sparsified data (T_h), and random projection data $\mathbf{A}_G$ (T_G). We use only 7% and 6% of all the elements of Digit data and $\mathbf{A}_{0.1}$, respectively, to construct respective sparse sketches.

	Sparsified Digit	Sparsified $\mathbf{A}_{0.1}$
Sparsity	93%	94%
$T_h/T_a/T_G$	30/151/36	18/73/36



(a) PCA's **(b) Projected data onto the PCA's**

Figure 1.6: Approximation quality of fast PCA (Algorithm 4) on Digit data. (a) Visualization of principal components as 16×16 image. Principal components are ordered from the top row to the bottom. First column of PCA's are exact $\mathcal{A}$. Second column of PCA's are $\mathcal{H}$ computed on sparsified data using $\sim 7\%$ of all the elements via optimal hybrid sampling. Third column of PCA's are $\mathcal{G}$ computed on A_G. Visually, $\mathcal{H}$ is closer to $\mathcal{A}$. (b) Visualization of projected data onto top three PCA's. First column shows the average digits of projected actual data onto the exact PCA's $\mathcal{A}$. Second column is the average digits of projected actual data onto approximate PCA's (of sampled data) $\mathcal{H}$. We observe a similar quality of average digits of projected actual data onto approximate PCA's $\mathcal{G}$ of A_G. Third column shows the average digits for projected sparsified data onto approximate PCA's $\mathcal{H}$.

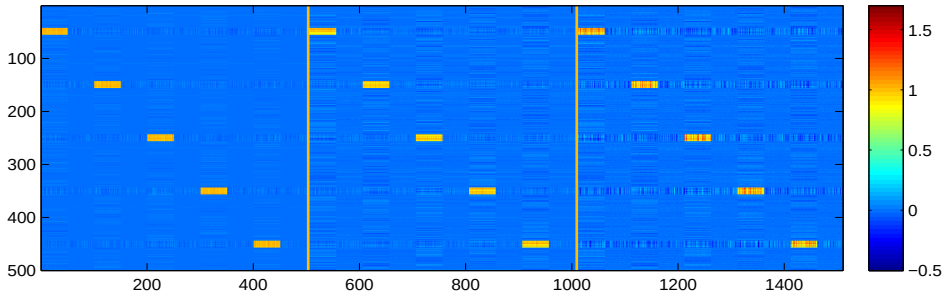


Figure 1.7: Approximation quality of fast PCA (Algorithm 4) for data $\mathrm{A}_{0.1}$. Visualization of projected data onto top five PCA's. Left image shows the projected actual data onto the exact PCAs $\mathcal{A}$. Middle image is the projection of actual data onto approximate PCA's (of sampled data) $\mathcal{H}$. We observe a similar quality of PCA's $\mathcal{G}$ for A_G. Right image shows the projected sparsified data onto approximate PCA's $\mathcal{H}$. We use only 6% of all the elements to produce the sparse sketches via optimal hybrid sampling.

1.5.5 Conclusion

Overall, the experimental results demonstrate the quality of the algorithms presented here, indicating the superiority of our approach to other extreme choices of element-wise sampling methods, such as, ℓ_1 and ℓ_2 sampling. Also, we demonstrate the theoretical and practical usefulness of hybrid-(ℓ_1, ℓ_2) sampling for fundamental data analysis tasks such as fast computation of PCA. Finally, our method outperforms element-wise leverage scores for the sparsification of various *low-rank* synthetic and real data matrices.

CHAPTER 2
Identifying *Influential Entries* in a Matrix

ABSTRACT: We consider the problem of exact recovery of any $m \times n$ matrix of rank ϱ from a small number of observed entries via the nuclear norm minimization in (2.1). Such low-rank matrices have degrees of freedom $(m+n)\varrho - \varrho^2$. We show that arbitrary low-rank matrices can be recovered exactly from $\Theta\left(((m+n)\varrho - \varrho^2)\log^2(m+n)\right)$ randomly sampled entries, thus matching the lower bound on the required number of entries (in degrees of freedom), with an additional factor of $O(\log^2(m+n))$. For this, we introduce a novel probability distribution on the elements of a matrix, namely, *influential entries*, in (2.3). We show that influential entries are capable of taking into account the coherence, as well as, the input sparsity of a low-rank matrix, and are superior to the sum of leverage scores of [9]) to reduce the sample size for exact matrix completion via (2.1), both in theory and practice. Further, we use the influential entries to propose an adaptive multi-phase sampling scheme for general low-rank matrix completion from scratch.

2.1 Introduction

We consider an incomplete matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ with information about only a small number of its elements. The *matrix completion* problem ([12]) is to predict those missing entries as accurately as possible based on the observed ones. Such partially-observed data appear in many application domains, e.g., in a user-recommendation system (a.k.a *collaborative filtering*) we have incomplete user ratings for various products and the goal is to make predictions about a user's preferences for all the products (e.g., the *Netflix problem*), or the incomplete data could represent some partial distance matrix in a sensor network, or missing pixels in digital images because of occlusion or tracking failures in a video surveillance system ([13]).

Mathematically, we have information about the entries $\mathbf{M}_{ij}$, $(i,j) \in \Omega$, where $\Omega \subset [m] \times [n]$ is a sampled subset of all entries, and $[n]$ denotes the list $\{1, ..., n\}$. The problem is to recover the unknown matrix $\mathbf{M}$ in a computationally tractable way from as few observed entries as possible. However, without further assumption on $\mathbf{M}$ it

is impossible to predict the unobserved elements from a limited number of known entries (what if all the entries are independent?). One popular assumption is that $\mathbf{M}$ has low-rank, say rank ϱ. Such matrices have degrees of freedom $(m+n)\varrho - \varrho^2$, that is, only this many independent elements control all the other elements. If $s = |\Omega| < (m+n)\varrho - \varrho^2$, there can be infinitely many matrices of rank at most ϱ with exactly the same entries in Ω; therefore, exact recovery of unobserved entries is impossible. So, we need at least $(m+n)\varrho - \varrho^2$ many observed entries for exact matrix completion for general rank-ϱ matrices.

The matrix $\mathbf{M}$, with the observed entries, can be interpreted as an element in mn-dimensional linear space, with $O((m+n)\varrho - \varrho^2)$ known coordinates. The set of matrices compatible with the observed entries forms a large affine space. Then, exact matrix completion problem is to specify an efficient algorithm which uniquely picks $\mathbf{M}$ from this high-dimensional affine space ([14]). A natural optimization problem for finding the low-rank $\mathbf{M}$ would be to find a matrix with minimum rank that is consistent with the observed entries. However, minimizing rank over an affine space is known to be NP-hard ([15]). [12] proposed to solve (2.1) (surrogate for rank minimization, [16]) to recover the low-rank matrix $\mathbf{M}$.

$$\min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \|\mathbf{X}\|_* \quad \text{subject to} \quad \mathbf{X}_{ij} = \mathbf{M}_{ij} \quad (i,j) \in \Omega, \tag{2.1}$$

where the nuclear norm $\|\mathbf{X}\|_*$ of a matrix $\mathbf{X}$ is defined as the sum of its singular values, $\|\mathbf{X}\|_* = \sum_i \sigma_i(\mathbf{X})$. (2.1) is a convex optimization problem that is efficiently solvable via semi-definite programming. Exact matrix completion thus becomes proving that the nuclear norm restricted to the affine space has a strict and global minima at $\mathbf{M}$. That is, if $\mathbf{M} + \mathbf{Z} \neq \mathbf{M}$ is a matrix in the affine space in (2.1), we need to show $\|\mathbf{M}+\mathbf{Z}\|_* > \|\mathbf{M}\|_*$. [12], [14], [6], [17] developed the sufficient conditions and main probabilistic tools in order to recover $\mathbf{M}$ as a unique solution to (2.1).

The question we try to answer here is: which elements of $\mathbf{M}$ should we observe in (2.1) (how to construct the sample set Ω) to reduce the sample size? We typically define some probabilities on the entries of $\mathbf{M}$ and sample according to them. Most of the existing work focused on the case when Ω in (2.1) is constructed by observing the entries of $\mathbf{M}$ uniformly randomly ([12], [14], [6], [17]). However, this data-oblivious sampling scheme has a cost. If $\mathbf{M}$ has only few non-zero entries, it cannot be recovered using uniform

sampling of its entries, unless we observe almost all the entries, because by observing only zeros it is impossible to predict non-zeros of a matrix. This suggests that $\mathbf{M}$ cannot be in the null-space of the sampling operator (to be defined later) extracting the values of a subset of the entries. Matrices similar to the above example can be characterized by the structure of their singular vectors. The singular vectors are (closely) 'aligned' with the standard basis vectors having high inner product with them. Therefore, the components of singular vectors should be sufficiently spread to reduce the number of observations needed to recover a low-rank matrix. Such restrictions on the row and column spaces of a low-rank matrix are called the *incoherence* assumptions (to be defined later). [14], [6] showed that such restricted class of $n \times n$ matrices of rank ϱ can be recovered exactly, with high probability, by observing $\Theta(n\varrho \log^2 n)$ entries sampled uniformly.

Very recently, [9] proposed non-uniform probabilities proportional to the sum of row and column leverage scores of $\mathbf{M}$ to observe its entries (*leveraged sampling*). They got rid of those 'incoherence' assumptions, and showed that any $n \times n$ matrix of rank ϱ can be recovered exactly, with high probability, from $\Theta(n\varrho \log^2 n)$ observed elements via leveraged sampling.

Similar to [9], we also incorporate the row and column leverage scores of the reconstructing matrix $\mathbf{M}$ into our probability of observing an entry. However, we use an additional *cross-leverage* term related to the inner product of i-th row of $\mathbf{U}$ and j-th row of $\mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular matrices of $\mathbf{M}$, respectively. We argue that this novel form of probabilities, called *influential entries* in (2.3), can represent the structure of a matrix better than the sum of leverage scores; therefore, by observing elements according to them we can reduce the sample complexity significantly for exact completion. Theorem 4 shows that observing entries according to the influential entries, we can recover any $m \times n$ matrix of rank-ϱ exactly, with high probability, from $\Theta(((m+n)\varrho - \varrho^2)\log^2(m+n))$ observed entries, via (2.1). This bound on the sample size is optimal (up to $\log^2(m+n)$ factor) in the degrees of freedom of a rank-ϱ matrix.

Also, we propose an algorithm (Algorithm 5) to design a recommendation system from scratch without knowing the influential entries. This adaptive multi-phase sampling algorithm is a more refined version of the two-phase sampling algorithm proposed by [9]. Experimental results show that the estimated influential entries perform better than

the estimated leverage scores; and the multi-phase adaptive approach is superior to the two-phase sampling.

Notations and preliminaries: We briefly describe the main notations used in this work. Natural number $\{1, ..., n\}$ are denoted by $[n]$. Natural logarithm of x is denoted by $\log(x)$. Matrices are bold uppercase, vectors are bold lowercase, and scalars are not bold. We denote the (i, j)-th entry of a matrix $\mathbf{X}$ by $\mathbf{X}_{ij}$. $\mathbf{e}_i$ denotes the i-th standard basis vector in $\mathbb{R}^d$, with i-th component 1 and other entries zero. The dimension of $\mathbf{e}_i$ will be clear from the context. $\mathbf{X}^T$ and $\mathbf{x}^T$ denote the transpose of matrix $\mathbf{X}$ and vector $\mathbf{x}$, respectively. $\text{Tr}(\mathbf{X})$ denotes the trace of a square matrix $\mathbf{X}$.

Spectral norm of $\mathbf{X}$ is denoted by $\|\mathbf{X}\|_2$. The inner product between two matrices is $\langle \mathbf{X}, \mathbf{Y} \rangle = \text{Tr}(\mathbf{X}^T\mathbf{Y})$. Frobenius norm $\mathbf{X}$ is denoted by $\|\mathbf{X}\|_F$, and $\|\mathbf{X}\|_F = \sqrt{\langle \mathbf{X}, \mathbf{X} \rangle}$. The nuclear norm of $\mathbf{X}$ is denoted by $\|\mathbf{X}\|_*$. The maximum entry of $\mathbf{X}$ is denoted by $\|\mathbf{X}\|_\infty = \max_{i,j} |\mathbf{X}_{ij}|$. For vectors Euclidean ℓ_2 norm is denoted by $\|\mathbf{x}\|_2$.

Linear operators acting on matrices are denoted by calligraphic letters. The spectral norm (largest singular value) of such operator $\mathcal{A}$ will be denoted by $\|\mathcal{A}\|_{op} = \sup_{\mathbf{X}} \frac{\|\mathcal{A}(\mathbf{X})\|_F}{\|\mathbf{X}\|_F}$. Also, we denote $f(n) = \Theta(g(n))$ when $\alpha_1 \cdot g(n) \leq f(n) \leq \alpha_2 \cdot g(n)$, for positive universal constants α_1, α_2.

2.2 Main Results

Our focus is to define probabilities on the entries of $\mathbf{M}$ (i.e., to construct the sample set Ω in (2.1)) to reduce the sample size, such that, $\mathbf{M}$ is the unique optimal solution to (2.1). In this work our sampling follows the Bernoulli model ([17], [9]), where each entry (i, j) is observed independently with some probability p_{ij}. Before we state our main result we first revisit the normalized leverage scores ([12], [6], [9]).

Definition 1 *Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ be of rank ϱ with SVD $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular matrices, respectively, and $\mathbf{\Sigma}$ is the diagonal matrix of singular values. Normalized leverage scores for i-th row (denoted by μ_i) and j-th column (denoted by ν_j) are defined as follows:*

$$\mu_i = (m/\varrho)\|\mathbf{U}^T\mathbf{e}_i\|_2^2, \quad i = 1, ..., m, \qquad \nu_j = (n/\varrho)\|\mathbf{V}^T\mathbf{e}_j\|_2^2, \quad j = 1, ..., n \tag{2.2}$$

Normalized leverage scores [4] are non-negative, and they depend on the structure of row and column spaces of the matrix. Also, $\sum_i \frac{\mu_i \varrho}{m} = \sum_j \frac{\nu_j \varrho}{n} = \varrho$, because $\mathbf{U}$ and $\mathbf{V}$ have orthonormal columns.

Let, $\max\{x, y, z\}$ return the maximum of the scalars x, y, and z. We state our main result.

Theorem 4 *Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ of rank ϱ with SVD $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Suppose, we have a subset of entries $\Omega \subset [m] \times [n]$, where each entry (i, j) is observed independently with probability p_{ij} in (2.3)*

$$p_{ij} \;=\; \max\left\{\min\{c_1 \log^2(m+n) \cdot L_{ij},\ 1\}, \min\{c_1 \log^2(m+n) \cdot C_{ij},\ 1\}, (mn)^{-5}\right\}, \tag{2.3}$$

for some universal constant $c_1 > 0$, where,

$$L_{ij} = \frac{\mu_i \varrho}{m} + \frac{\nu_j \varrho}{n} - \frac{\mu_i \varrho}{m} \cdot \frac{\nu_j \varrho}{n}, \quad C_{ij} = \frac{(m+n-\varrho)\varrho}{\sum_{i,j} |(\mathbf{U}\mathbf{V}^T)_{ij}|} \cdot |(\mathbf{U}\mathbf{V}^T)_{ij}|.$$

Then, $\mathbf{M}$ is the unique solution to (2.1) with probability at least $1 - 33 \log(m+n)(m+n)^{3-c}$, for sufficiently large $c > 3$. Moreover, if the number of observed entries, according to (2.3), is $|\Omega| = \Theta\left((m+n)\varrho - \varrho^2)\log^2(m+n)\right)$, then, $\mathbf{M}$ is the unique solution to (2.1) with probability at least $1 - 66 \log(m+n)(m+n)^{3-c}$, for sufficiently large $c > 3$.

The fundamental difference between the probabilities in (2.3) and existing non-uniform distributions, e.g., sum of leverage scores ([9], [20]) is the presence of the *cross-leverage* term C_{ij}. That is, p_{ij} in (2.3) is biased towards the elements with high leverage scores, as well as, high cross-leverage term. We take a closer look at the significance of the terms in (2.3).

We know row or column leverage scores ([21]) indicate the contribution of a row or column forming the low-rank subspace ([18], [19]). When i-th row or j-th column has high leverage score all the elements of i-th row or j-th column get a high probability value p_{ij} through the sum of leverage scores L_{ij} (elements being part of something important

[4] $\|\mathbf{U}^T \mathbf{e}_i\|_2^2$ and $\|\mathbf{V}^T \mathbf{e}_j\|_2^2$ are called leverage scores for i-th row and j-th column, respectively, by [18], [19] for low-rank matrix approximation.

get attention). However, L_{ij} may not give us a refined view of the structure of the data (all the elements of an important row may not be equally informative). For example, let $\mathbf{M}$ be a rank-1 matrix with $\mathbf{M}_{1,1} = 1$ and all other entries are 0. L_{ij} would suggest us to observe elements along the first row and the first column of $\mathbf{M}$, whereas it is perhaps meaningful to mainly focus on the element $\mathbf{M}_{1,1}$ to recover $\mathbf{M}$. This 'limitation' of L_{ij} could be overcome by C_{ij} as it is capable of revealing the 'local' structure of the data. First, note that $(\mathbf{U}\mathbf{V}^T)_{ij} = (\mathbf{U}^T\mathbf{e}_i)^T(\mathbf{V}^T\mathbf{e}_j)$, i.e., it is an inner product of the i-th row of $\mathbf{U}$ and j-th row of $\mathbf{V}$. We can prove that for all i, j,

$$|(\mathbf{U}\mathbf{V}^T)_{ij}| \leq \|\mathbf{U}^T\mathbf{e}_i\|_2 \cdot \|\mathbf{V}^T\mathbf{e}_j\|_2 = \sqrt{\frac{\mu_i\varrho}{m} \cdot \frac{\nu_j\varrho}{n}} \leq \frac{\mu_i\varrho}{m} + \frac{\nu_j\varrho}{n} - \frac{\mu_i\varrho}{m} \cdot \frac{\nu_j\varrho}{n} = L_{ij}, \quad (2.4)$$

where we use Cauchy-Schwarz inequality in the first inequality, and the last inequality is shown in Lemma 7 of [20]. When L_{ij} is high $|(\mathbf{U}\mathbf{V}^T)_{ij}|$ could be as large as L_{ij}. However, $|(\mathbf{U}\mathbf{V}^T)_{ij}|$, being an inner product, could also be low, say zero, even when L_{ij} is high. This happens when, for example, i-th row of $\mathbf{U}$ and j-th row of $\mathbf{V}$ have high leverage scores but they are mutually (nearly) orthogonal. $\mathbf{M}_{ij}$ with high L_{ij} but low $|(\mathbf{U}\mathbf{V}^T)_{ij}|$ is an interesting case because it suggests this (i, j)-th element may not be structurally important and could be just a noisy one or simply zero despite belonging to an important row or column. On the other hand, $\mathbf{M}_{ij}$ with high L_{ij} and high $|(\mathbf{U}\mathbf{V}^T)_{ij}|$ is the most influential one, and we want it in Ω. Revisiting the example where $\mathbf{M}$ is rank-1 with $\mathbf{M}_{1,1} = 1$, $(\mathbf{U}\mathbf{V}^T)_{1,1}$ would be 1 and all other elements of $\mathbf{U}\mathbf{V}^T$ would be zero. This way, $|(\mathbf{U}\mathbf{V}^T)_{i,j}|$ is a better representative of the local structure of the data. This is a simple practical motivation why we want to incorporate $|(\mathbf{U}\mathbf{V}^T)_{ij}|$ in our element-wise sampling probabilities.

We want to observe elements with high $|(\mathbf{U}\mathbf{V}^T)_{ij}|$. However, $|(\mathbf{U}\mathbf{V}^T)_{ij}|$ is powerless when used along with L_{ij} in p_{ij}, because it is overshadowed by L_{ij} (due to (2.4)). This is why we scale up $|(\mathbf{U}\mathbf{V}^T)_{ij}|$ to form C_{ij}, and we use this C_{ij} along with L_{ij} in (2.3). Note that, C_{ij} can be larger than L_{ij} for some i, j, and these are our desired elements. An algebraic justification for the particular choice of scaling in (2.3) is that observing elements with probabilities proportional to C_{ij} gives us $O((m + n - \varrho)\varrho)$ (order of degrees of freedom) number of elements, in expectation. Thus, p_{ij} in (2.3) suggests: all the ele-

ments belonging to important rows and columns get a score proportional to L_{ij}, and the elements with important local structural information get an *additional* score proportional to C_{ij} (note that p_{ij} is maximum of L_{ij} and C_{ij}). This way the *influential entries* of $\mathbf{M}$, represented by p_{ij} in (2.3), are designed to be more *aligned* with the structure of the data. Observing elements using p_{ij} in (2.3) has the potential to reduce the sample size of Ω to recover a matrix exactly via (2.1). We make this claim more formal in the subsequent discussion starting with the main proof strategy.

The standard proof strategy to show that $\mathbf{M}$ is unique solution to (2.1) is to construct a *dual certificate* $\mathbf{Y}$ that obeys certain sub-gradient properties of nuclear norm. We define: $\mathcal{P}_T(\mathbf{X}) = \mathbf{U}\mathbf{U}^T\mathbf{X} + \mathbf{X}\mathbf{V}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{X}\mathbf{V}\mathbf{V}^T$, $\mathcal{P}_{T^\perp}(\mathbf{X}) = \mathbf{X} - \mathcal{P}_T(\mathbf{X})$. Let, $\mathcal{R}_\Omega(\mathbf{X})$ be a matrix with $(\mathcal{R}_\Omega(\mathbf{X}))_{ij} = \mathbf{X}_{ij}/p_{ij}$, if $(i,j) \in \Omega$, and zero otherwise. Similarly, $\mathcal{P}_\Omega(\mathbf{X})$ is a matrix with $(\mathcal{P}_\Omega(\mathbf{X}))_{ij} = \mathbf{X}_{ij}$, if $(i,j) \in \Omega$, and zero otherwise. Then, $\mathbf{M}$ is unique solution to (2.1) if

1. $\|\mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T - \mathcal{P}_T\|_{op} \leq 1/2$. 2. There exists $\mathbf{Y}$, s.t., $\mathcal{P}_\Omega(\mathbf{Y}) = \mathbf{Y}$, and (a) $\|\mathcal{P}_T(\mathbf{Y}) - \mathbf{U}\mathbf{V}^T\|_F \leq \sqrt{\varrho(m+n)^{-15}}$, (b) $\|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2 \leq 1/2$.

We focus on the construction of $\mathbf{Y}$ as this is the main step that controls the best-known sample size in the literature. Intuitively, $\mathbf{Y} \approx \mathbf{U}\mathbf{V}^T$ satisfies the conditions 2(a) and 2(b) above. For this, a first step would be to set $\mathbf{Y} = \mathcal{R}_\Omega(\mathbf{U}\mathbf{V}^T)$ (i.e., $\mathbf{Y}$ is an element-wise sparse, unbiased sketch [5] of $\mathbf{U}\mathbf{V}^T$), and bound the error in operator norm. However, this is insufficient to achieve very high accuracy, such as, in condition 2(a). This motivates [14] to invent the *golfing scheme* to construct $\mathbf{Y}$ adaptively as follows (see [9]): $\mathbf{W}_0 = 0$, $\mathbf{W}_k = \mathbf{W}_{k-1} + \mathcal{R}_{\Omega_k}\mathcal{P}_T(\mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{W}_{k-1}))$, for $k = 1, ..., k_0$, and $\mathbf{Y} = \mathbf{W}_{k_0}$, where Ω_k is an i.i.d random set of indices with $P[(i,j) \in \Omega_k] = q_{ij} = 1 - (1 - p_{ij})^{1/k_0}$, and $\mathcal{R}_{\Omega_k}$ is defined similar to $\mathcal{R}_\Omega$ (but q_{ij} replaces p_{ij}). Note that, $q_{ij} \geq p_{ij}/k_0$ because of overlapping of Ω_k's. Let, $\Delta_0 = \mathbf{U}\mathbf{V}^T$, and $\Delta_k = \mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{W}_k)$, for each $k = 1, ..., k_0$. Then, $\Delta_k = (\mathcal{P}_T - \mathcal{P}_T\mathcal{R}_{\Omega_k}\mathcal{P}_T)\Delta_{k-1}$. Now, $\|\mathcal{P}_T(\mathbf{Y}) - \mathbf{U}\mathbf{V}^T\|_F = \|\Delta_{k_0}\|_F$, and we apply condition 1 to derive the bound in 2(a) by setting $k_0 = 11\log(m+n)$. This k_0 appears as a factor in the sample size (contributing an extra log factor).

Now, we take a closer look at the golfing scheme. $\mathbf{W}_1 = \mathcal{R}_{\Omega_1}(\mathbf{U}\mathbf{V}^T)$ is a sparse sketch of $\mathbf{U}\mathbf{V}^T$; similarly, $\mathbf{W}_2$ is $\mathbf{W}_1$ plus a sparse sketch of the first residual $\mathbf{U}\mathbf{V}^T -$

[5] see [1], [4], [3], [7], [27] for element-wise matrix sparsification.

$\mathcal{P}_T(\mathbf{W}_1)$, and $\mathbf{Y}$ is the sum of k_0 such sparse sketches. Intuitively, if we have a good sparse sketch $\mathbf{W}_1$ ($\approx \mathbf{U}\mathbf{V}^T$) then the residuals would be close to zero in subsequent steps, and the recursion may converge faster. This may potentially eliminate the need for the extra log factor in a practical situation. Furthermore, we derive

$$\|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2 = \|\mathcal{P}_{T^\perp} \sum_{k=1}^{k_0} \left(\mathcal{R}_{\Omega_k}\mathcal{P}_T - \mathcal{P}_T\right)(\Delta_{k-1})\|_2 \leq \sum_{k=1}^{k_0} \|(\mathcal{R}_{\Omega_k} - I)(\Delta_{k-1})\|_2. \quad (2.5)$$

Quality of the above bound affects the constants involved in the sample size (see [9], [20] for details). Note that, the first term in the summation of (2.5) is $\|\mathcal{R}_{\Omega_1}(\mathbf{U}\mathbf{V}^T) - \mathbf{U}\mathbf{V}^T\|_2$ (for $k = 1$) which quantifies how good the sparse sketch $\mathbf{W}_1$ is. This establishes a formal connection between the quality of $\mathbf{W}_1$ and the sample size.

Above, we reiterate the importance of a high-quality sparsification of $\mathbf{U}\mathbf{V}^T$ to potentially reduce the sample size (both in theory and practice). [9] bounded each term in the summation in (2.5) using their Lemma 10 (involving probabilities proportional to the sum of leverage scores), and eventually upper-bounded the summation using a couple of closed form solutions of infinite sums (disregarding k_0). We also adopt similar strategy, however, with the following exception. We bound the first term in the summation in (2.5) using an improved result (Lemma 4) for the sparsification of $\mathbf{U}\mathbf{V}^T$ involving the C_{ij} in (2.3); other terms are bounded using L_{ij} (similar to [9]).

Lemma 4 *Let $\mathbf{M} \in \mathbb{R}^{m\times n}$ be a rank-ϱ matrix with SVD $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Let Ω be a sampled set of indices of $\mathbf{M}$, such that, each index $(i, j) \in \Omega$ is observed independently with some probability q_{ij}. Then, the following holds with probability at least $1 - (m + n)^{1-c}$ for any universal constant $c > 1$,*

$$\|\mathcal{R}_\Omega(\mathbf{U}\mathbf{V}^T) - \mathbf{U}\mathbf{V}^T\|_2 \leq 2\sqrt{\frac{c}{c_0}} + \frac{c}{c_0} \cdot \frac{\sum_{i,j=1}^{m,n} |(\mathbf{U}\mathbf{V}^T)_{ij}|}{(m + n - \varrho)\varrho}$$

if, for some universal constant $c_0 > 0$ and L_{ij} and C_{ij} as in Theorem 4,

$$q_{ij} \geq \max\left\{\min\{c_0 \cdot \log(m + n) \cdot L_{ij},\ 1\}, \min\{c_0 \cdot \log(m + n) \cdot C_{ij},\ 1\}\right\}.$$

This c_0 affects the sample size because $c_1 \propto c_0 \cdot k_0$, for c_1 in (2.3). Bounding the sum in (2.5) by a fixed constant, say $1/2$, using Lemma 4 (improved $\mathbf{W}_1$ in the golfing scheme) requires strictly smaller c_0 rather than bounding $\|\mathcal{R}_\Omega(\mathbf{U}\mathbf{V}^T) - \mathbf{U}\mathbf{V}^T\|_2$ using sum of leverage scores in [9], if

$$\text{Influential Ratio} := \frac{\sum_{i,j=1}^{m,n} |(\mathbf{U}\mathbf{V}^T)_{ij}|}{(m+n-\varrho)\varrho} < \frac{1}{2}. \tag{2.6}$$

Note that, $\sum_{i,j=1}^{m,n} |(\mathbf{U}\mathbf{V}^T)_{ij}| \leq \sqrt{mn\varrho}$ from Cauchy-Schwartz inequality. Thus, we want $(m+n-\varrho)\varrho > 2\sqrt{mn\varrho}$, which holds any for $\varrho \geq 2$ and $m, n \geq 4$. The bound in (2.6) suggests that the sample size, using our influential entries, may be greatly benefited from the (near)[6] sparsity of $\mathbf{U}\mathbf{V}^T$. Revisiting the example where $\mathbf{M}$ is rank-1 with $\mathbf{M}_{1,1} = 1$ and other entries are zeros, the Influential Ratio in (2.6) is $\approx 1/(m+n)$. This suggests that we need much smaller sample size for exact recovery of (extremely coherent) $\mathbf{M}$ sampling according to the influential entries rather than the sum of leverage scores. Intuitively, the input sparsity of $\mathbf{M}$ helps us to recover the matrix exactly with smaller sample size. This motivates us to investigate a relation between the (near) sparsity of input matrix $\mathbf{M}$ and the sample size.

2.2.1 Relation with Input Sparsity:

Using the SVD of $\mathbf{M}$ we can write $\mathbf{M}_{ij} = (\mathbf{U}^T\mathbf{e}_i)^T\mathbf{\Sigma}(\mathbf{V}^T\mathbf{e}_j)$, and $(\mathbf{U}\mathbf{V}^T)_{ij} = (\mathbf{U}^T\mathbf{e}_i)^T(\mathbf{V}^T\mathbf{e}_j)$. We want to establish the following: $\mathbf{M}_{ij} \approx 0$ implies $(\mathbf{U}\mathbf{V}^T)_{ij} \approx 0$, that is, $\sum_{l=1}^{\varrho} \sigma_l \mathbf{U}_{il}\mathbf{V}_{jl} \approx 0$ implies $\sum_{l=1}^{\varrho} \mathbf{U}_{il}\mathbf{V}_{jl} \approx 0$, for arbitrary singular values σ_l. This happens when $\mathbf{U}_{il}\mathbf{V}_{jl} \approx 0$ for all l. For this, rows of $\mathbf{M}$ must be either very similar to each other or mutually orthogonal such a way that product of the corresponding elements must be nearly zero (Figures 2.1 and 2.4). An example of such matrices is a feature-sample data (features are columns characterizing some clusters) where rows (samples) show exclusive affinity towards their respective clusters, e.g., a row, belonging to cluster 1 which is characterized by say first five columns, has high values for first five components and other components are nearly zero. Such matrices are ubiquitous in machine learning: 1) In a document-term matrix, the key words describing a particular

[6] elements with magnitude ≈ 0

topic have high frequency of occurrence in a document belonging to that topic cluster, and other key words related to a different topic have very low frequency in such a document, 2) Sample images of digits '6', '9', and '1' show high intensity values (nearly) exclusively to certain pixels characterizing them, 3) Adjacency matrix of graph with non-overlapping community structure of nodes show above-mentioned pattern (Figures 2.1 and 2.4). Such matrices can be recovered from a significantly smaller number of elements sampled according to the influential entries rather than the sum of leverage scores, *even when the matrix is completely incoherent* (Figure 2.1).

Few Motivating Examples: We create a 500×500 binary block diagonal matrix with five identical 100×100 blocks of 1's (Figure 2.1). We denote this rank-5 data by $\mathbf{M}_5$ (e.g., adjacency matrix of a graph with five identical cliques). Influential entries of $\mathbf{M}_5$ represent the structure better (Figure 2.2) than the sum of (*uniform*) leverage scores of $\mathbf{M}_5$ (Figure 2.3). Similar, result holds for 500×500 binary rank-10 data $\mathbf{M}_{10}$ in Figure 2.4.

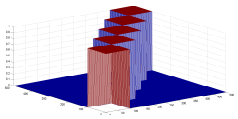


Figure 2.1: Structure of $\mathbf{M}_5$.

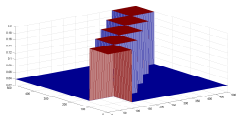


Figure 2.2: Influential entries.

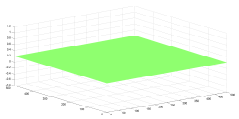


Figure 2.3: Sum of leverage scores.

Consequently, sampling elements according to the influential entries helps us to reduce the sample size for exact matrix completion (see Section 2.3).

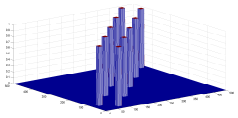


Figure 2.4: Structure of $\mathbf{M}_{10}$.

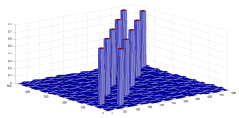


Figure 2.5: Influential entries.

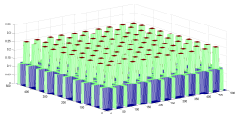


Figure 2.6: Sum of leverage scores.

2.2.2 Matrix Completion using Adaptive Multi-Phase Sampling

We have so far seen that any arbitrary $m \times n$ matrix $\mathbf{M}$ of rank ϱ can be recovered exactly using $\Theta((m+n)\varrho - \varrho^2)\log^2(m+n))$ observed entries according to the influential entries of $\mathbf{M}$ in (2.3). However, in reality, we are ignorant of the terms $\{\mu_i\}$, $\{\nu_j\}$, and

$\{(\mathbf{U}\mathbf{V}^T)_{ij}\}$, even when we are free to choose entries. We propose an adaptive multi-phase sampling scheme (Algorithm 5) which assumes no prior knowledge about the leverage scores and the cross-leverage term.

Informally, we first observe a fraction of our budget s of elements of $\mathbf{M}$ uniformly randomly (without replacement). We perform rank-ϱ SVD on this partially-observed data (other entries are zeros), and get the first estimate of the influential entries. Next, we observe remaining budget of elements in τ iterative steps according to the (adaptively refined) estimates of the influential entries.

Algorithm 5 Adaptive Multi-Phase Sampling for Exact Matrix Completion

1: **Input:** Rank ϱ, sample budget s, parameter $\beta \in [0, 1]$, and positive integer $\tau \leq (1-\beta)s$.
2: Sample elements uniformly randomly without replacement to form set Ω, such that, $|\Omega| = \beta s$.
3: $t = 1$.
4: **do** Perform rank-ϱ SVD to $\mathcal{P}_\Omega(\mathbf{M})$, $\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T$, to compute $\{\tilde{\mu}_i\}$ and $\{\tilde{\nu}_j\}$, and $\{(\tilde{\mathbf{U}}\tilde{\mathbf{V}}^T)_{ij}\}$.
5: Compute ${\tilde{p}_{ij}}^{(t)}$ as in (2.7).
6: Sample without replacement a set Ω_t of $(1-\beta)s/\tau$ new elements according to ${\tilde{p}_{ij}}^{(t)}$.
7: $\Omega \leftarrow \Omega \cup \Omega_t$; and $t = t + 1$.
8: **while** $t \leq \tau$
9: **Output:** (completed matrix) $\tilde{\mathbf{M}} = \min_{\mathbf{X}} \|\mathbf{X}\|_*$, s.t. $\mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{M})$.

Let Ω be a set of sampled entries, and $\mathcal{P}_\Omega(\mathbf{M})$ be a sampling operator that maps elements of $\mathbf{M}$ not in Ω to 0. Let, the total budget of samples be s, and $\beta \in [0, 1]$ be a parameter. First, we utilize β fraction of the budget to construct an initial set Ω of observed entries sampled uniformly (without replacement) in order to estimate the leverage scores and the cross leverage term of $\mathbf{M}$. We perform rank-ϱ SVD of $\mathcal{P}_\Omega(\mathbf{M})$, $\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T$, where $\tilde{\mathbf{U}} \in \mathbb{R}^{m\times\varrho}$, $\tilde{\mathbf{V}} \in \mathbb{R}^{n\times\varrho}$, and $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{\varrho\times\varrho}$, to compute $\tilde{\mu}_i = (m/\varrho)\|\tilde{\mathbf{U}}^T\mathbf{e}_i\|_2^2$, $\tilde{\nu}_j = (n/\varrho)\|\tilde{\mathbf{V}}^T\mathbf{e}_j\|_2^2$, and $\tilde{C}_{ij} = (m+n-\varrho)\varrho \cdot |(\tilde{\mathbf{U}}\tilde{\mathbf{V}}^T)_{ij}|/(\sum_{i,j}|(\tilde{\mathbf{U}}\tilde{\mathbf{V}}^T)_{ij}|)$, and use them as estimates for μ_i, ν_j, and C_{ij}, respectively. Next, we sample a set Ω_t of $(1-\beta)s/\tau$ elements of $\mathbf{M}$ (without replacement) according to the t-th estimate of influential entries in (2.7).

$$ {\tilde{p}_{ij}}^{(t)} \propto \max\{\tilde{L}_{ij}, \tilde{C}_{ij}\}\log^2(m+n) \tag{2.7} $$

where, $\tilde{L}_{ij} = \frac{\tilde{\mu}_i \varrho}{m} + \frac{\tilde{\nu}_j \varrho}{n} - \frac{\tilde{\mu}_i \varrho}{m} \cdot \frac{\tilde{\nu}_j \varrho}{n}$. We adaptively refine $\tilde{p}_{ij}{}^{(t)}$ via SVD of $\mathcal{P}_{\Omega \cup \Omega_t}(\mathbf{M})$ in $t = 1, ..., \tau$ steps, and sample the next set of $(1-\beta)s/\tau$ elements according to these refined probabilities. Finally, we perform matrix completion using this Ω constructed above, with $|\Omega| = s$, in (2.1).

Our adaptive multi-phase sampling Algorithm 5 is similar to the two-phase Algorithm 1 of [9] for $\tau = 1$. However, we divide the remaining budget $(1-\beta)s$ into τ batches, and we refine the estimates of probabilities using all the previously observed entries. This refined approach seems to be a more realistic scenario to design a recommendation system from scratch. Note that we need to compute the SVD of the partially observed data (sparse) τ times in Algorithm 5. However, SVD on sparse data is fast, therefore running time of repeated (sparse) SVD in Algorithm 5 is not a concern (use a small $\tau > 1$). Experiments show that we require much smaller s for exact completion by sampling elements via estimated influential entries rather than estimated leverage scores, and the performance of the estimated influential entries improves setting $\tau = 2$ rather than $\tau = 1$.

2.3 Experiments

We compare exact matrix completion, via (2.1), using influential entries and sum of leverage scores. We use 'TFOCS' v1.2 by Stephen Becker, Emmanuel Candes, and Michael Grant to solve (2.1).

Experimental Design: Let $\mathbf{M}$ be the rank-ϱ data matrix. We construct the sample set Ω_{inf} by observing (i,j)-th entry of $\mathbf{M}$ according to the influential entries in (2.8):

$$p_{ij}^{[inf]} \propto \max\left\{\min\{c_f \cdot L_{ij}, 1\}, \min\{c_f \cdot C_{ij}, 1\}\right\} \quad \forall i,j \tag{2.8}$$

where c_f is a universal constant, and L_{ij} and C_{ij} as in (2.3). Similarly, we construct the sample set Ω_{lev} by observing (i,j)-th entry of $\mathbf{M}$ according to the leverage score probabilities in (2.9):

$$p_{ij}^{[lev]} \propto \min\left\{c_l \cdot (\mu_i \varrho/m + \nu_j \varrho/n), 1\right\}, \quad \forall i,j \tag{2.9}$$

where c_l is a universal constant. We use Ω_{inf} and Ω_{lev} in the optimization problem (2.1),

separately, to recover $\mathbf{M}$. Let $\mathbf{X}^*$ be the unique solution to (2.1) using a sample set Ω. We say $\mathbf{X}^*$ recovers $\mathbf{M}$ exactly if $\|\mathbf{M} - \mathbf{X}^*\|_F < \varepsilon$, where ε is a tiny fraction. We set $\varepsilon = 0.001$. We perform 10 independent trials (sampling and recovery), and declare success if $\mathbf{M}$ is recovered exactly at least 9 times. Let s_f and s_l be the average sample size for successful recovery of $\mathbf{M}$ using Ω_{inf} and Ω_{lev}, respectively. The sample size gain is $(s_l - s_f)/s_l$ for sampling according to the influential entries.

We evaluate the performance of Algorithm 5 using the estimates $\{\tilde{\mu}_i\}$, $\{\tilde{\nu}_j\}$, $\{\tilde{L}_{ij}\}$, and $\{\tilde{C}_{ij}\}$ (in Algorithm 5) of their respective quantities in (2.8) and (2.9).

2.3.1 Datasets

• **Synthetic:** $\mathbf{M}_5$ of rank 5 in Figure 2.1. This matrix is incoherent with uniform leverage scores.

• **MovieLens:** This collaborative filtering dataset (movielens.umn.edu) contains ratings between 1 and 5 by 943 users on 1682 movies. Each user has rated at least 20 movies. This dataset is numerically not low-rank. We perform rank truncation to create a low-rank matrix to apply the theory in (2.1). For this, we choose two values for rank: $\varrho = 10$ and $\varrho = 20$, observing the singular value spectrum. This dataset is reasonably coherent for such choices of rank.

2.3.2 Results

Results for known probabilities: Table 2.1 shows the gain in sample size for exact matrix completion when we observe elements via probabilities in (2.8), as opposed to the probabilities in (2.9). Also, we note the Influential Ratio in (2.6) (a small value $<$ 0.5 implies better gain).

Table 2.1: Gain in sample size $(s_l - s_f)/s_l$ for exact completion via the influential entries in (2.8).

Dataset		c_l/c_f	$s_l/(m \cdot n)$	$s_f/(m \cdot n)$	$(s_l - s_f)/s_l$	$\frac{\sum_{i,j} \lvert(\mathbf{UV}^T)_{ij}\rvert}{(m+n-\varrho)\varrho}$
$\mathbf{M}_5$	$\varrho = 5$	8/2	15.98%	7.17%	55.13%	.101
MovieLens	$\varrho = 10$	11/6	18.18%	12.61%	30.64%	.074
	$\varrho = 20$	7/3	22.92%	13.54%	40.92%	.054

Results for estimated probabilities in Algorithm 5: We show that even sampling 90% elements of the budget uniformly, and only 10% elements according to the estimated influential entries, we need much smaller budget of samples than estimated sum of leverage scores for exact matrix completion for both synthetic and real data (Table 2.2 and Figure 2.10). Figure 2.8 shows how we can sample important elements of a matrix using the estimated influential entries. Furthermore, $\tau > 1$ in Algorithm 5 improves the performance of influential entries for both the datasets, showing the superiority of Algorithm 5 over Algorithm 1 of [9] to design a recommendation system from scratch.

Table 2.2: [Data M_5] Percentage of successful completion using estimated probabilities in Algorithm 5 for various sample size s, and τ. Even drawing 90% ($\beta = 0.9$) of the samples uniformly and using estimated probabilities in Algorithm 5, influential entries outperform leverage scores (uniform in this case). Performance of estimated influential entries improves for $\tau > 1$.

		$\frac{s}{(m+n-\varrho)\varrho} = 8$	$\frac{s}{(m+n-\varrho)\varrho} = 9$
Estimated Influential Entry	$\tau = 1$	80%	100% (success)
	$\tau = 2$	90% (success)	100% (success)
Estimated Leverage Scores	$\tau = 1$	70%	100% (success)
	$\tau = 2$	70%	100% (success)

Overall, these results support the accuracy of the theoretical analysis on the gain in sample size for exact recovery of a low-rank matrix via (2.1) when the elements are sampled according to the influential entries rather than the sum of leverage scores.

2.4 Proof of Theorem 4

The main proof strategy was outlined by [12], [6], [14]: it is sufficient to construct a *dual certificate* $\mathbf{Y}$ obeying specific sub-gradient inequalities in order to show that $\mathbf{M}$ is the unique optimal solution to (2.1) (see Section 2.5 for more detail). We give a proof of Theorem 4 closely following the proof strategy of [6], [9]. Before stating the optimality conditions we need additional notations.

Recall, $\mathbf{U}$ and $\mathbf{V}$ are the left and right singular matrices of $\mathbf{M}$, respectively. Let $\mathbf{u}_k$ (respectively $\mathbf{v}_k$) denote the k-th column of $\mathbf{U}$ (respectively $\mathbf{V}$). Let T be a linear space spanned by elements of the form $\mathbf{u}_k\mathbf{y}^T$ and $\mathbf{x}\mathbf{v}_k^T$, $1 \leq k \leq \varrho$, for arbitrary $\mathbf{x}, \mathbf{y}$,

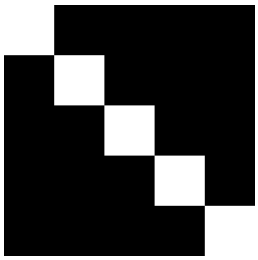


Figure 2.7: Image of binary rank-5 data M_5 where white pixels are 1 and black pixels are 0.

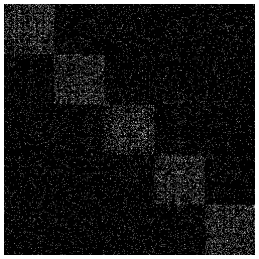


Figure 2.8: Sampled $(1-\beta)s$ indices (white pixels) using estimated influential entries in Algorithm 5 with $\tau = 2$ ($\beta = 0.7$).

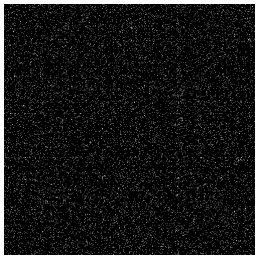


Figure 2.9: Sampled $(1-\beta)s$ indices (white pixels) using estimated leverage scores in Algorithm 5 with $\tau = 2$ ($\beta = 0.7$).

and $T^{\perp}$ be its orthogonal complement, i.e., $T^{\perp}$ is spanned by the family $(\mathbf{x}\mathbf{y}^T)$, where $\mathbf{x}$ (respectively $\mathbf{y}$) is any vector orthogonal to the space spanned by the left singular vectors (respectively right singular vectors). Then, orthogonal projection onto T is given by the linear operator $\mathcal{P}_T : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$, defined as

$$\mathcal{P}_T(\mathbf{X}) = \mathbf{U}\mathbf{U}^T\mathbf{X} + \mathbf{X}\mathbf{V}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{X}\mathbf{V}\mathbf{V}^T.$$

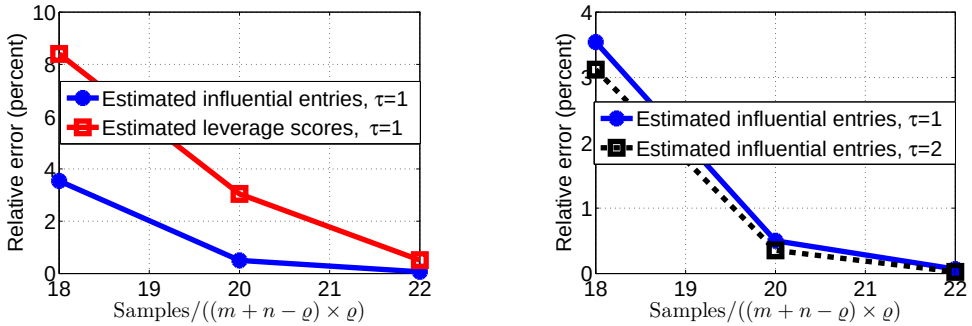


Figure 2.10: [MovieLens, $\varrho = 10$] Plot of relative error for completed matrix using estimated probabilities in Algorithm 5. (Left) Estimated influential entries significantly outperform estimated leverage scores. (Right) Performance of estimated influential entries improves for $\tau > 1$.

Similarly, orthogonal projection onto $T^{\perp}$ is

$$\mathcal{P}_{T^{\perp}}(\mathbf{X}) = \mathbf{X} - \mathcal{P}_T(\mathbf{X}) = \mathbf{U}_{\perp}\mathbf{U}_{\perp}^T\mathbf{X}\mathbf{V}_{\perp}\mathbf{V}_{\perp}^T.$$

Note that any $m \times n$ matrix $\mathbf{X}$ can be expressed as a sum of rank-one matrices as follows:

$$\mathbf{X} = \sum_{i,j=1}^{m,n} \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathbf{X} \right\rangle \mathbf{e}_i\mathbf{e}_j^T. \tag{2.10}$$

We define the sampling operator $\mathcal{R}_{\Omega} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ as,

$$\mathcal{R}_{\Omega}(\mathbf{X}) = \sum_{i,j=1}^{m,n} \frac{1}{p_{ij}}\delta_{ij} \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathbf{X} \right\rangle \mathbf{e}_i\mathbf{e}_j^T \tag{2.11}$$

where, $\delta_{ij} = \mathbb{I}((i,j) \in \Omega)$, $\mathbb{I}(\cdot)$ being the indicator function. That is, $\mathcal{R}_{\Omega}$ extracts the terms, corresponding to the indices $(i,j) \in \Omega$, from (2.10) to form a partial sum in (2.11). Let $\mathcal{P}_{\Omega}(\mathbf{X})$ be the matrix with $(\mathcal{P}_{\Omega}(\mathbf{X}))_{ij} = \mathbf{X}_{ij}$ if $(i,j) \in \Omega$, and zero otherwise.

2.4.1 Optimality Conditions

Following the proof road map of [6], [9], we restate the sufficient conditions for $\mathbf{M}$ to be the unique optimal solution to (2.1) (Section 2.5 contains a proof of sufficiency).

Proposition 1 *The rank-ϱ matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ with SVD $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the unique*

optimal solution to (2.1) if the following conditions hold:

1. $\|\mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T - \mathcal{P}_T\|_{op} \leq 1/2$.
2. *There exists a dual certificate* $\mathbf{Y}$ *which satisfies* $\mathcal{P}_\Omega(\mathbf{Y}) = \mathbf{Y}$*, and*
   (a) $\left\|\mathcal{P}_T(\mathbf{Y}) - \mathbf{U}\mathbf{V}^T\right\|_F \leq \sqrt{\varrho(m+n)^{-15}}$,
   (b) $\|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2 \leq 1/2$.

Condition 1 of Proposition 1 suggests $\mathcal{R}_\Omega$ should be nearly the identity operator on the subspace T. Next we discuss the construction of a dual certificate $\mathbf{Y}$.

2.4.1.1 Constructing the Dual Certificate

We follow the so-called golfing scheme [14], [17], [9] to construct a matrix $\mathbf{Y}$ (the dual certificate) that satisfies Condition 2 in Proposition 1. Recall, we assume that the set of observed elements Ω follows the Bernoulli model with parameter p_{ij}, i.e., each index (i,j) is observed independently with $P\left[(i,j) \in \Omega\right] = p_{ij}$ (p_{ij} in eqn (2.3)). We denote this by $\Omega \sim Bernoulli(p_{ij})$. Further, we assume that Ω is generated from $\Omega = \cup_{k=1}^{k_0}\Omega_k$, where for each k, $\Omega_k \sim Bernoulli(q_{ij})$, and we set $q_{ij} = 1 - (1-p_{ij})^{1/k_0}$. Clearly, this implies $P\left[(i,j) \in \Omega\right] = p_{ij}$ which is the original Bernoulli model for Ω. Note that, $q_{ij} \geq p_{ij}/k_0$ because of overlapping of Ω_k's. We set $k_0 = 11 \cdot \log(m+n)$. Then,

$$q_{ij} \geq \min\left\{c_0 \cdot \log(m+n) \cdot \left(\frac{\mu_i\varrho}{m} + \frac{\nu_j\varrho}{n} - \frac{\mu_i\varrho}{m}\cdot\frac{\nu_j\varrho}{n}\right), 1\right\}, \quad (2.12)$$

$$\text{and,} \quad q_{ij} \geq \min\left\{c_0 \cdot \log(m+n) \cdot \frac{(m+n-\varrho)\varrho}{\sum_{i,j}|(\mathbf{U}\mathbf{V}^T)_{ij}|} \cdot |(\mathbf{U}\mathbf{V}^T)_{ij}|, 1\right\}, \quad (2.13)$$

where $c_0 = c_1/11$. Starting with $\mathbf{W}_0 = 0$ and for each $k = 1, ..., k_0$, we recursively define

$$\mathbf{W}_k = \mathbf{W}_{k-1} + \mathcal{R}_{\Omega_k}\mathcal{P}_T(\mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{W}_{k-1})) \quad (2.14)$$

where the sampling operator $\mathcal{R}_{\Omega_k} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is defined as

$$\mathcal{R}_{\Omega_k}(\mathbf{X}) = \sum_{i,j}\frac{1}{q_{ij}}\mathbb{I}((i,j)\in\Omega_k)\left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathbf{X}\right\rangle \mathbf{e}_i\mathbf{e}_j^T.$$

We set $\mathbf{Y} = \mathbf{W}_{k_0}$. This $\mathbf{Y}$ is supported on Ω, i.e., $\mathcal{P}_\Omega(\mathbf{Y}) = \mathbf{Y}$.
Let the sample set $\tilde{\Omega}$ be such that

$$\tilde{\Omega} \in \{\Omega_k : \Omega = \cup_{k=1}^{k_0} \Omega_k, \Omega_k \sim Bernoulli(q_{ij})\}. \tag{2.15}$$

Since $\Omega_k \sim Bernoulli(q_{ij})$ implies $\Omega \sim Bernoulli(p_{ij})$, for each $k = 1, ..., k_0$, we prove (in Lemma 5) Condition 1 of Proposition 1 using sample set $\tilde{\Omega}$ in (2.15).

Lemma 5 *Let $\tilde{\Omega}$ be a sample set in (2.15). Then, for any universal constant $c > 1$, we have*

$$\left\|\mathcal{P}_T \mathcal{R}_{\tilde{\Omega}} \mathcal{P}_T - \mathcal{P}_T\right\|_{op} \leq 1/2 \tag{2.16}$$

holding with probability at least

$$1 - (m+n)^{1-c}.$$

Before we validate Condition 2 in Proposition 1 using the $\mathbf{Y}$ constructed above, we claim the following results to hold with high probability. First, we borrow the following definitions of weighted infinity norms for a matrix $\mathbf{Z} \in \mathbb{R}^{m \times n}$ from [9].

$$\begin{aligned} \|\mathbf{Z}\|_{\mu(\infty,2)} &:= \max\left\{ \max_i \sqrt{\frac{m}{\mu_i \varrho}} \left\|\mathbf{Z}_{i,*}\right\|_2, \max_j \sqrt{\frac{n}{\nu_j \varrho}} \left\|\mathbf{Z}_{*,j}\right\|_2 \right\} \\ \|\mathbf{Z}\|_{\mu(\infty)} &:= \max_{i,j} |\mathbf{Z}_{ij}| \sqrt{\frac{m}{\mu_i \varrho}} \sqrt{\frac{n}{\nu_j \varrho}} \end{aligned}$$

where $\mathbf{Z}_{i,*}$ and $\mathbf{Z}_{*,j}$ denote the i-th row and j-th column of $\mathbf{Z}$, respectively.

Lemma 6 bounds the spectral norm of the matrix $(\mathcal{R}_{\tilde{\Omega}} - \mathcal{I})(\mathbf{Z})$ using the sample set $\tilde{\Omega}$.

Lemma 6 *Let $\mathbf{Z} \in \mathbb{R}^{m \times n}$ be a fixed matrix. Let $\tilde{\Omega}$ be a sample set in (2.15). Then, for any universal constant $c > 1$, we have*

$$\left\|\left(\mathcal{R}_{\tilde{\Omega}} - \mathcal{I}\right)\mathbf{Z}\right\|_2 \leq 2\sqrt{\frac{c}{c_0}} \|\mathbf{Z}\|_{\mu(\infty,2)} + \frac{c}{c_0} \|\mathbf{Z}\|_{\mu(\infty)}$$

holding with probability at least

$$1 - (m+n)^{1-c}.$$

Next two results control the $\mu(\infty, 2)$ and $\mu(\infty)$ norms of the projection of a matrix after random sampling.

Lemma 7 *Let $\mathbf{Z} \in \mathbb{R}^{m\times n}$ be a fixed matrix. Let $\tilde{\Omega}$ be a sample set in (2.15). Then, for any universal constant $c > 2$, we have*

$$\|(\mathcal{P}_T\mathcal{R}_{\tilde{\Omega}} - \mathcal{P}_T)\mathbf{Z}\|_{\mu(\infty,2)} \leq \frac{1}{2}\left(\|\mathbf{Z}\|_{\mu(\infty,2)} + \|\mathbf{Z}\|_{\mu(\infty)}\right)$$

holding with probability at least

$$1 - (m+n)^{2-c}.$$

Lemma 8 *Let $\mathbf{Z} \in \mathbb{R}^{m\times n}$ be a fixed matrix. Let $\tilde{\Omega}$ be a sample set in (2.15). Then, for any universal constant $c > 3$, we have*

$$\|(\mathcal{P}_T\mathcal{R}_{\tilde{\Omega}} - \mathcal{P}_T)\mathbf{Z}\|_{\mu(\infty)} \leq \frac{1}{2}\|\mathbf{Z}\|_{\mu(\infty)}$$

holding with probability at least

$$1 - (m+n)^{3-c}.$$

We now validate Condition 2 in Proposition 1 using the $\mathbf{Y}$ constructed above.

Bounding $\left\|\mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{Y})\right\|_F$

We set $\Delta_k = \mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{W}_k)$, for $k = 1, ..., k_0$. Then, from definition of $\mathbf{W}_k$ we have

$$\Delta_k = (\mathcal{P}_T - \mathcal{P}_T\mathcal{R}_{\Omega_k}\mathcal{P}_T)\Delta_{k-1}.$$

We used $\mathcal{P}_T(\mathbf{U}\mathbf{V}^T) = \mathbf{U}\mathbf{V}^T$ and $\mathcal{P}_T\mathcal{P}_T(\mathbf{X}) = \mathcal{P}_T(\mathbf{X})$. Using the independence of Δ_{k-1} and Ω_k,

$$\|\Delta_k\|_F = \|(\mathcal{P}_T - \mathcal{P}_T\mathcal{R}_{\Omega_k}\mathcal{P}_T)\Delta_{k-1}\|_F \leq \|\mathcal{P}_T - \mathcal{P}_T\mathcal{R}_{\Omega_k}\mathcal{P}_T\|_{op} \|\Delta_{k-1}\|_F .$$

We can bound this by recursively applying Lemma 5 with Ω_k, for all k. Thus,

$$\left\|\mathcal{P}_T(\mathbf{Y}) - \mathbf{U}\mathbf{V}^T\right\|_F = \|\Delta_{k_0}\|_F = \left(\frac{1}{2}\right)^{k_0} \left\|\mathbf{U}\mathbf{V}^T\right\|_F \leq \sqrt{\frac{\varrho}{(m+n)^{15}}}$$

The above result fails with probability at most $(m+n)^{1-c}$ for each k; thus, total probability of failure is at most $11(m+n)^{1-c}\log(m+n)$.

Bounding $\|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2$

By definition, $\mathbf{Y}$ can be written as

$$\mathbf{Y} = \sum_{k=1}^{k_0} \mathcal{R}_{\Omega_k}\mathcal{P}_T\left(\mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{W}_{k-1})\right) = \sum_{k=1}^{k_0} \mathcal{R}_{\Omega_k}\mathcal{P}_T(\Delta_{k-1})$$

It follows that,

$$\|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2 = \left\|\mathcal{P}_{T^\perp}\sum_{k=1}^{k_0}\left(\mathcal{R}_{\Omega_k}\mathcal{P}_T - \mathcal{P}_T\right)(\Delta_{k-1})\right\|_2 \leq \sum_{k=1}^{k_0} \|(\mathcal{R}_{\Omega_k} - I)(\Delta_{k-1})\|_2$$

We use

$$\mathcal{P}_T(\Delta_k) = \mathcal{P}_T(\mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{W}_k)) = \mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{W}_k) = \Delta_k, \text{ for all } k.$$

In the above summand, for $k = 1$ we have $\|\mathcal{R}_{\Omega_1}(\mathbf{U}\mathbf{V}^T) - \mathbf{U}\mathbf{V}^T\|_2$. We bound the terms from $k = 2, ..., k_0$ of the above summand using Lemma 6, with corresponding Ω_k, to

obtain

$$\begin{aligned}\|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2 &\leq \|\mathcal{R}_{\Omega_1}(\mathbf{U}\mathbf{V}^T) - \mathbf{U}\mathbf{V}^T\|_2 + 2\sqrt{\frac{c}{c_0}}\sum_{k=2}^{k_0}\|\Delta_{k-1}\|_{\mu(\infty,2)} \\ &\quad + \frac{c}{c_0}\sum_{k=2}^{k_0}\|\Delta_{k-1}\|_{\mu(\infty)} \end{aligned} \tag{2.17}$$

We can derive the following, applying Lemma 8 k times, with Ω_k,

$$\begin{aligned}\|\Delta_k\|_{\mu(\infty)} = \|(\mathcal{P}_T - \mathcal{P}_T\mathcal{R}_{\Omega_k})\Delta_{k-1}\|_{\mu(\infty)} &\leq \left(\frac{1}{2}\right)^i \|\Delta_{k-i}\|_{\mu(\infty)} \\ &\leq \left(\frac{1}{2}\right)^k \left\|\mathbf{U}\mathbf{V}^T\right\|_{\mu(\infty)} \end{aligned} \tag{2.18}$$

holding with failure probability at most $k\cdot(m+n)^{3-c}$, for all k. Similarly, applying Lemma 7 and Lemma 8 recursively, with Ω_k, we can derive,

$$\begin{aligned}\|\Delta_k\|_{\mu(\infty,2)} &= \|(\mathcal{P}_T - \mathcal{P}_T\mathcal{R}_{\Omega_k}\mathcal{P}_T)\Delta_{k-1}\|_{\mu(\infty,2)} \leq \frac{1}{2}\|\Delta_{k-1}\|_{\mu(\infty)} + \frac{1}{2}\|\Delta_{k-1}\|_{\mu(\infty,2)} \\ \text{(step } j) &\leq \sum_{i=1}^{j}\left(\frac{1}{2}\right)^i \|\Delta_{k-i}\|_{\mu(\infty)} + \left(\frac{1}{2}\right)^j \|\Delta_{k-j}\|_{\mu(\infty,2)} \\ &\leq \sum_{i=1}^{j}\left(\frac{1}{2}\right)^i \left(\frac{1}{2}\right)^{k-i} \left\|\mathbf{U}\mathbf{V}^T\right\|_{\mu(\infty)} + \left(\frac{1}{2}\right)^j \|\Delta_{k-j}\|_{\mu(\infty,2)} \\ &\leq j\left(\frac{1}{2}\right)^k \left\|\mathbf{U}\mathbf{V}^T\right\|_{\mu(\infty)} + \left(\frac{1}{2}\right)^j \|\Delta_{k-j}\|_{\mu(\infty,2)} \\ \text{(step } k) &\leq k\left(\frac{1}{2}\right)^k \left\|\mathbf{U}\mathbf{V}^T\right\|_{\mu(\infty)} + \left(\frac{1}{2}\right)^k \left\|\mathbf{U}\mathbf{V}^T\right\|_{\mu(\infty,2)} \end{aligned} \tag{2.19}$$

holding with failure probability to be at most $k\cdot(m+n)^{2-c}$, for all k. Using Lemma 4, (2.18), and (2.19), it follows from (2.17),

$$\begin{aligned}&\|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2 \\ \leq\ & 2\sqrt{\frac{c}{c_0}} + \frac{c}{c_0}\cdot\frac{\sum_{i,j}|(\mathbf{U}\mathbf{V}^T)_{ij}|}{(m+n-\varrho)\varrho} + 2\sqrt{\frac{c}{c_0}}\sum_{k=2}^{k_0}(k-1)\left(\frac{1}{2}\right)^{k-1}\left\|\mathbf{U}\mathbf{V}^T\right\|_{\mu(\infty)} \\ &+2\sqrt{\frac{c}{c_0}}\sum_{k=2}^{k_0}\left(\frac{1}{2}\right)^{k-1}\left\|\mathbf{U}\mathbf{V}^T\right\|_{\mu(\infty,2)} + \frac{c}{c_0}\sum_{k=2}^{k_0}\left(\frac{1}{2}\right)^{k-1}\left\|\mathbf{U}\mathbf{V}^T\right\|_{\mu(\infty,2)}\end{aligned}$$

We note that, for all (i,j),

$$\left|(\mathbf{U}\mathbf{V}^T)_{ij}\right| = \left|\mathbf{e}_i^T\mathbf{U}\mathbf{V}^T\mathbf{e}_j\right| \leq \sqrt{\frac{\mu_i\varrho}{m}}\sqrt{\frac{\nu_j\varrho}{n}} \leq 1,$$

$$\left\|(\mathbf{U}\mathbf{V}^T)_{i,*}\right\|_2 = \left\|\mathbf{e}_i^T\mathbf{U}\mathbf{V}^T\right\|_2 = \sqrt{\frac{\mu_i\varrho}{m}}, \qquad \left\|(\mathbf{U}\mathbf{V}^T)_{*,j}\right\|_2 = \left\|\mathbf{U}\mathbf{V}^T\mathbf{e}_j\right\|_2 = \sqrt{\frac{\nu_j\varrho}{n}}$$

Thus,

$$\left\|\mathbf{U}\mathbf{V}^T\right\|_{\mu(\infty,2)} = \max\left\{\max_i \sqrt{\frac{m}{\mu_i\varrho}}\left\|(\mathbf{U}\mathbf{V}^T)_{i,*}\right\|_2, \max_j \sqrt{\frac{n}{\nu_j\varrho}}\left\|(\mathbf{U}\mathbf{V}^T)_{*,j}\right\|_2\right\} = 1$$

Therefore,

$$\begin{aligned}
&\left\|\mathcal{P}_{T^\perp}(\mathbf{Y})\right\|_2 \\
\leq\ & 2\sqrt{\frac{c}{c_0}} + \frac{c}{c_0}\cdot\frac{\sum_{i,j}\left|(\mathbf{U}\mathbf{V}^T)_{ij}\right|}{(m+n-\varrho)\varrho} \qquad (2.20)\\
& +2\sqrt{\frac{c}{c_0}}\sum_{k=2}^{k_0}\left((k-1)\left(\frac{1}{2}\right)^{k-1} + \left(\frac{1}{2}\right)^{k-1}\right) + \frac{c}{c_0}\sum_{k=2}^{k_0}\left(\frac{1}{2}\right)^{k-1} \\
=\ & 2\sqrt{\frac{c}{c_0}}\sum_{k=1}^{k_0} k\left(\frac{1}{2}\right)^{k-1} + \frac{c}{c_0}\sum_{k=1}^{k_0}\left(\frac{1}{2}\right)^{k-1} - \frac{c}{c_0}\cdot\left(1 - \frac{\sum_{i,j}\left|(\mathbf{U}\mathbf{V}^T)_{ij}\right|}{(m+n-\varrho)\varrho}\right) \\
<\ & 2\sqrt{\frac{c}{c_0}}\sum_{k=1}^{\infty} k\left(\frac{1}{2}\right)^{k-1} + \frac{c}{c_0}\sum_{k=1}^{\infty}\left(\frac{1}{2}\right)^{k-1} - \frac{c}{c_0}\cdot\left(1 - \frac{\sum_{i,j}\left|(\mathbf{U}\mathbf{V}^T)_{ij}\right|}{(m+n-\varrho)\varrho}\right) \\
=\ & 8\sqrt{\frac{c}{c_0}} + \frac{2c}{c_0} - \frac{c}{c_0}\cdot\left(1 - \frac{\sum_{i,j}\left|(\mathbf{U}\mathbf{V}^T)_{ij}\right|}{(m+n-\varrho)\varrho}\right) \leq \frac{1}{2}, \qquad (2.21)
\end{aligned}$$

for sufficiently large c_0.The constant c_0 appears in the sample size, and the bound on c_0 above gets tighter as long as $\sum_{i,j}\left|(\mathbf{U}\mathbf{V}^T)_{ij}\right| < \frac{1}{2}(m+n-\varrho)\varrho$. We have from Cauchy-Schwartz inequality $\sum_{i,j}\left|(\mathbf{U}\mathbf{V}^T)_{ij}\right| \leq \sqrt{mn\varrho}$. This suggests as long as $2\sqrt{mn} < (m+n-\varrho)\sqrt{\varrho}$, we expect to see a reduction in sample size using the influential entries. The above condition holds for $\varrho \geq 2$ and $m, n \geq 4$.

We sample each (i,j)-th entry independently with probability p_{ij} to form the set of observations Ω. That is, total number of sampled entries, denoted by s, is a random

variable. Expected number of observed entries required to solve (2.1) is

$$\mathbb{E}(s) = \sum_{i,j} p_{ij} = O(((m+n)\varrho - \varrho^2)\log^2(m+n)).$$

Summing up the failure probabilities of condition 2a and 2b in Proposition 1, the total failure probability never exceeds $33 \cdot \log(m+n)(m+n)^{3-c}$, for sufficiently large $c > 3$.

Finally, we can apply Hoeffding's inequality to show that s is sharply concentrated around its expectation, i.e., $s = \Theta(((m+n)\varrho - \varrho^2)\log^2(m+n))$ with probability at least $1 - 66\,\log(m+n)(m+n)^{3-c}$, for sufficiently large $c > 3$.

This completes the proof of Theorem 4.

2.5 Proof of Optimality Conditions in Proposition 1

Let $\mathbf{M}$ be the low-rank target matrix with rank-ϱ SVD $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. We want to show that any perturbation $\mathbf{Z}$ to $\mathbf{M}$, such that, $\mathbf{M} + \mathbf{Z}$ is a solution to (2.1), strictly increases the nuclear norm, unless $\mathbf{Z} = 0$. Now, $\mathbf{M} + \mathbf{Z}$ is feasible only if $\mathcal{P}_\Omega(\mathbf{M} + \mathbf{Z}) = \mathcal{P}_\Omega(\mathbf{M})$, which implies $\mathcal{R}_\Omega(\mathbf{Z}) = 0$, e.g., $\mathbf{Z}$ is in the null space of $\mathcal{R}_\Omega$ operator. We can choose $\mathbf{U}_\perp$ and $\mathbf{V}_\perp$ such that $[\mathbf{U}, \mathbf{U}_\perp]$ and $[\mathbf{V}, \mathbf{V}_\perp]$ are unitary matrices for which $\left\langle \mathbf{U}_\perp \mathbf{V}_\perp^T, \mathcal{P}_{T^\perp}(\mathbf{Z}) \right\rangle = \|\mathcal{P}_{T^\perp}(\mathbf{Z})\|_*$. Then it follows from standard inequality of trace norm, for some $\mathbf{Y}$ in the range of $\mathcal{R}_\Omega$,

$$\begin{aligned}
& \|\mathbf{M} + \mathbf{Z}\|_* \\
\geq\ & \left\langle \mathbf{U}\mathbf{V}^T + \mathbf{U}_\perp \mathbf{V}_\perp^T, \mathbf{M} + \mathbf{Z} \right\rangle \\
=\ & \|\mathbf{M}\|_* + \left\langle \mathbf{U}\mathbf{V}^T + \mathbf{U}_\perp \mathbf{V}_\perp^T, \mathbf{Z} \right\rangle \\
=\ & \|\mathbf{M}\|_* + \left\langle \mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{Y}), \mathcal{P}_T(\mathbf{Z}) \right\rangle + \left\langle \mathbf{U}_\perp \mathbf{V}_\perp^T - \mathcal{P}_{T^\perp}(\mathbf{Y}), \mathcal{P}_{T^\perp}(\mathbf{Z}) \right\rangle \\
\overset{(a)}{\geq}\ & \|\mathbf{M}\|_* - \|\mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{Y})\|_F \cdot \|\mathcal{P}_T(\mathbf{Z})\|_F + \|\mathcal{P}_{T^\perp}(\mathbf{Z})\|_* - \langle \mathcal{P}_{T^\perp}(\mathbf{Y}), \mathcal{P}_{T^\perp}(\mathbf{Z}) \rangle \\
\geq\ & \|\mathbf{M}\|_* - \|\mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{Y})\|_F \cdot \|\mathcal{P}_T(\mathbf{Z})\|_F + (1 - \|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2)\ \|\mathcal{P}_{T^\perp}(\mathbf{Z})\|_* \\
\overset{(b)}{\geq}\ & \|\mathbf{M}\|_* + \left(1 - \|\mathcal{P}_{T^\perp}(\mathbf{Y})\|_2 - \frac{\|\mathbf{U}\mathbf{V}^T - \mathcal{P}_T(\mathbf{Y})\|_F \left(\max_{i,j} \frac{1}{\sqrt{p_{ij}}} \right)}{\left(1 - \|\mathcal{P}_T \mathcal{R}_\Omega \mathcal{P}_T - \mathcal{P}_T\|_{op} \right)^{\frac{1}{2}}} \right) \|\mathcal{P}_{T^\perp}(\mathbf{Z})\|_* \\
>\ & \|\mathbf{M}\|_*
\end{aligned}$$

Above, (a) follows from Von-Neumann trace inequality, and (b) follows from Lemma 9. Using $\max_{i,j} \frac{1}{\sqrt{p_{ij}}} \leq (mn)^{5/2} \leq (m+n)^5$, and the conditions in Proposition 1, we derive the final inequality. Note that, Condition 1 in Proposition 1 implies $\mathcal{R}_\Omega$ is the identity operator on the elements of subspace T, therefore $\mathcal{P}_{T^\perp}(\mathbf{Z}) = 0$ implies $\mathbf{Z} = 0$.

The following lemma is similar to Lemma 13 of [9].

Lemma 9 *For any* $\mathbf{Z} \in \mathbb{R}^{m\times n}$*, s.t.,* $\mathcal{P}_\Omega(\mathbf{Z}) = 0$*,*

$$\|\mathcal{P}_T(\mathbf{Z})\|_F \leq \left(1 - \|\mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T - \mathcal{P}_T\|_{op}\right)^{-\frac{1}{2}} \left(\max_{i,j} \frac{1}{\sqrt{p_{ij}}}\right) \|\mathcal{P}_{T^\perp}\mathbf{Z}\|_*$$

Proof: Let us define the operator $\mathcal{R}_\Omega^{1/2} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ as

$$\mathcal{R}_\Omega^{1/2}(\mathbf{Z}) := \sum_{i,j} \frac{1}{\sqrt{p_{ij}}} \delta_{ij} \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathbf{Z}\right\rangle \mathbf{e}_i\mathbf{e}_j^T$$

Note that $\mathcal{R}_\Omega^{1/2}$ is self-adjoint, and $\mathcal{R}_\Omega^{1/2}\mathcal{R}_\Omega^{1/2} = \mathcal{R}_\Omega$. Therefore, we have

$$\begin{aligned}
\left\|\mathcal{R}_\Omega^{1/2}\mathcal{P}_T(\mathbf{Z})\right\|_F^2 &= \langle \mathcal{R}_\Omega\mathcal{P}_T(\mathbf{Z}), \mathcal{P}_T(\mathbf{Z})\rangle \\
&= \langle \mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T(\mathbf{Z}), \mathcal{P}_T(\mathbf{Z})\rangle \\
&= \langle \mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T(\mathbf{Z}) - \mathcal{P}_T(\mathbf{Z}), \mathcal{P}_T(\mathbf{Z})\rangle + \langle \mathcal{P}_T(\mathbf{Z}), \mathcal{P}_T(\mathbf{Z})\rangle \\
&\geq (1 - \|\mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T - \mathcal{P}_T\|_{op}) \cdot \|\mathcal{P}_T(\mathbf{Z})\|_F^2 \qquad (2.22)
\end{aligned}$$

Also, we have $\left\|\mathcal{R}_\Omega^{1/2}(\mathbf{Z})\right\|_F = 0$ for any $\mathbf{Z}$ s.t. $\mathcal{P}_\Omega(\mathbf{Z}) = 0$. It follows,

$$\begin{aligned}
0 = \left\|\mathcal{R}_\Omega^{1/2}(\mathbf{Z})\right\|_F &\geq \left\|\mathcal{R}_\Omega^{1/2}\mathcal{P}_T(\mathbf{Z})\right\|_F - \left\|\mathcal{R}_\Omega^{1/2}\mathcal{P}_{T^\perp}(\mathbf{Z})\right\|_F \\
\left\|\mathcal{R}_\Omega^{1/2}\mathcal{P}_T(\mathbf{Z})\right\|_F &\leq \left\|\mathcal{R}_\Omega^{1/2}\mathcal{P}_{T^\perp}(\mathbf{Z})\right\|_F \leq \left(\max_{i,j} \frac{1}{\sqrt{p_{ij}}}\right) \|\mathcal{P}_{T^\perp}(\mathbf{Z})\|_F, \qquad (2.23)
\end{aligned}$$

where we use

$$\left\|\mathcal{R}_\Omega^{1/2}\mathcal{P}_{T^\perp}(\mathbf{Z})\right\|_F \leq \max_{i,j} \frac{1}{\sqrt{p_{ij}}} \left\|\sum_{i,j} \delta_{ij} \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathcal{P}_{T^\perp}(\mathbf{Z})\right\rangle \mathbf{e}_i\mathbf{e}_j^T\right\|_F \leq \max_{i,j} \frac{1}{\sqrt{p_{ij}}} \|\mathcal{P}_{T^\perp}(\mathbf{Z})\|_F$$

Combining (2.22) and (2.23), and using $\|\mathbf{X}\|_F \leq \|\mathbf{X}\|_*$,

$$\begin{aligned}\sqrt{(1 - \|\mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T - \mathcal{P}_T\|_{op})} \cdot \|\mathcal{P}_T(\mathbf{Z})\|_F &\leq \left(\max_{i,j} \frac{1}{\sqrt{p_{ij}}}\right) \|\mathcal{P}_{T^\perp}(\mathbf{Z})\|_F \\ &\leq \left(\max_{i,j} \frac{1}{\sqrt{p_{ij}}}\right) \|\mathcal{P}_{T^\perp}(\mathbf{Z})\|_*\end{aligned}$$

The result follows.

◇

2.6 Proof of Technical Lemmas

Here we prove Lemmas 5 through 8 using the matrix Bernstein inequality of Lemma 11 as the main tool. Also, we frequently use the fact in (2.25) and the result in Lemma 10. Note that $\mathcal{P}_T$ is self-adjoint linear operator. Thus we can write the following for any $\mathbf{X} \in \mathbb{R}^{m\times n}$:

$$\mathcal{P}_T(\mathbf{X}) = \sum_{i,j} \left\langle \mathcal{P}_T(\mathbf{X}), \mathbf{e}_i\mathbf{e}_j^T \right\rangle \mathbf{e}_i\mathbf{e}_j^T = \sum_{i,j} \left\langle \mathbf{X}, \mathcal{P}_T(\mathbf{e}_i\mathbf{e}_j^T) \right\rangle \mathbf{e}_i\mathbf{e}_j^T \tag{2.24}$$

We can derive, for all i and j,

$$\left\|\mathcal{P}_T\left(\mathbf{e}_i\mathbf{e}_j^T\right)\right\|_F^2 = \left\langle \mathcal{P}_T\left(\mathbf{e}_i\mathbf{e}_j^T\right), \mathbf{e}_i\mathbf{e}_j^T \right\rangle = \frac{\mu_i\varrho}{m} + \frac{\nu_j\varrho}{n} - \frac{\mu_i\varrho}{m} \cdot \frac{\nu_j\varrho}{n} \tag{2.25}$$

Also, we know for all i, j,

$$0 \leq \frac{\mu_i\varrho}{m} \leq \sqrt{\frac{\mu_i\varrho}{m}} \leq 1, \quad 0 \leq \frac{\nu_j\varrho}{n} \leq \sqrt{\frac{\nu_j\varrho}{n}} \leq 1. \tag{2.26}$$

Lemma 10 *Using our notations, for all i, j,*

$$\frac{\mu_i\varrho}{m} + \frac{\nu_j\varrho}{n} - \frac{\mu_i\varrho}{m} \cdot \frac{\nu_j\varrho}{n} \geq \sqrt{\frac{\mu_i\varrho}{m}} \cdot \sqrt{\frac{\nu_j\varrho}{n}}$$

Proof: Let, $x = \frac{\mu_i \varrho}{m}$ and $y = \frac{\nu_j \varrho}{n}$. Then,

$$\begin{aligned}(x+y-xy)^2 &= xy + (x^2 - x^2y) + (y^2 - xy^2) + x^2y^2 + xy - x^2y - xy^2 \\ &= xy + x^2(1-y) + y^2(1-x) + xy(1-x)(1-y) \\ &\geq xy \quad \text{using } (2.26)\end{aligned}$$

Also, $x + y - xy \geq 0$. Thus, $x + y - xy \geq \sqrt{xy}$.

$\diamond$

Lemma 11 *([22], [Theorem 16] of [9])*
Let $\mathbf{X}_1$, ..., $\mathbf{X}_N \in \mathbb{R}^{m \times n}$ *be independent, zero-mean random matrices. Suppose*

$$\max\left\{ \left\| \sum_{t=1}^{N} \mathbb{E}\left[\mathbf{X}_t \mathbf{X}_t^T\right] \right\|_2, \left\| \sum_{t=1}^{N} \mathbb{E}\left[\mathbf{X}_t^T \mathbf{X}_t\right] \right\|_2 \right\} \leq \sigma^2$$

and $\|\mathbf{X}_t\|_2 \leq \gamma$ *almost surely for all* t. *Then for any* $c > 0$, *we have*

$$\left\| \sum_{t=1}^{N} \mathbf{X}_t \right\|_2 \leq 2\sqrt{c\sigma^2 \log(m+n)} + c\gamma \log(m+n)$$

with probability at least $1 - (m+n)^{-(c-1)}$.

We consider sampling probabilities $\{q_{ij}\}$ of the form (2.12) to prove Lemmas 5 through 8.

Notation Overloading: For simplicity, we reuse some of the notations in Section 2.6.1 through 2.6.4. Specifically, we replace $\tilde{\Omega}$ by Ω to denote a sample set in (2.15), and, $\delta_{ij} = \mathbb{I}((i,j) \in \tilde{\Omega})$.

2.6.1 Proof of Lemma 5

For any matrix $\mathbf{Z} \in \mathbb{R}^{m \times n}$, we can write

$$\left(\mathcal{P}_T \mathcal{R}_\Omega \mathcal{P}_T - \mathcal{P}_T\right)(\mathbf{Z}) = \sum_{i,j} \left(\frac{1}{q_{ij}} \delta_{ij} - 1 \right) \left\langle \mathcal{P}_T\left(\mathbf{e}_i \mathbf{e}_j^T\right), \mathbf{Z} \right\rangle \mathcal{P}_T\left(\mathbf{e}_i \mathbf{e}_j^T\right) = \sum_{i,j} \mathcal{S}_{ij}(\mathbf{Z}).$$

Using $\mathbb{E}\left[\delta_{ij}\right] = q_{ij}$, we have $\mathbb{E}[\mathcal{S}_{ij}(\mathbf{Z})] = 0$ for any $\mathbf{Z}$. Thus, we conclude that $\mathbb{E}[\mathcal{S}_{ij}] = 0$. Also, $\mathcal{S}_{ij}$'s are independent of each other. Using probabilities in (2.12) ($\mathcal{S}_{ij}$'s vanish when

$q_{ij} = 1$, for all $\mathbf{Z}$ and (i,j)), and (2.25), we derive

$$\|\mathcal{S}_{ij}(\mathbf{Z})\|_F \leq \frac{1}{q_{ij}} \left\|\mathcal{P}_T\left(\mathbf{e}_i\mathbf{e}_j^T\right)\right\|_F^2 \|\mathbf{Z}\|_F \leq \frac{\|\mathbf{Z}\|_F}{c_0 \cdot \log(m+n)}.$$

From definition of operator norm, $\|\mathcal{S}_{ij}\|_{op} \leq \frac{1}{c_0 \cdot \log(m+n)}$. Also, we derive

$$\begin{aligned}
\mathbb{E}\left[\mathcal{S}_{ij}^2(\mathbf{Z})\right] &= \mathbb{E}\left[\left(\frac{1}{q_{ij}}\delta_{ij} - 1\right)^2\right] \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathcal{P}_T(\mathbf{Z})\right\rangle \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathcal{P}_T(\mathbf{e}_i\mathbf{e}_j^T)\right\rangle \mathcal{P}_T\left(\mathbf{e}_i\mathbf{e}_j^T\right) \\
&= \frac{1-q_{ij}}{q_{ij}} \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathcal{P}_T(\mathbf{Z})\right\rangle \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathcal{P}_T(\mathbf{e}_i\mathbf{e}_j^T)\right\rangle \mathcal{P}_T\left(\mathbf{e}_i\mathbf{e}_j^T\right)
\end{aligned}$$

$$\begin{aligned}
&\left\|\sum_{i,j} \mathbb{E}\left[\mathcal{S}_{ij}^2(\mathbf{Z})\right]\right\|_F \\
\leq\ & \left(\max_{i,j} \frac{1-q_{ij}}{q_{ij}} \left\|\mathcal{P}_T(\mathbf{e}_i\mathbf{e}_j^T)\right\|_F^2\right) \left\|\sum_{i,j} \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathcal{P}_T(\mathbf{Z})\right\rangle \mathcal{P}_T\left(\mathbf{e}_i\mathbf{e}_j^T\right)\right\|_F \\
=\ & \left(\max_{i,j} \frac{1-q_{ij}}{q_{ij}} \left\|\mathcal{P}_T(\mathbf{e}_i\mathbf{e}_j^T)\right\|_F^2\right) \left\|\mathcal{P}_T\left(\sum_{i,j} \left\langle \mathbf{e}_i\mathbf{e}_j^T, \mathcal{P}_T(\mathbf{Z})\right\rangle \left(\mathbf{e}_i\mathbf{e}_j^T\right)\right)\right\|_F \\
=\ & \left(\max_{i,j} \frac{1-q_{ij}}{q_{ij}} \left\|\mathcal{P}_T(\mathbf{e}_i\mathbf{e}_j^T)\right\|_F^2\right) \|\mathcal{P}_T(\mathbf{Z})\|_F
\end{aligned}$$

$$\left\|\sum_{i,j} \mathbb{E}\left[\mathcal{S}_{ij}^2\right]\right\|_{op} \leq \max_{i,j} \frac{1-q_{ij}}{q_{ij}} \left\|\mathcal{P}_T(\mathbf{e}_i\mathbf{e}_j^T)\right\|_F^2 \leq \frac{1}{c_0 \cdot \log(m+n)}$$

We apply Matrix Bernstein inequality in Lemma 11 using

$$\sigma^2 = \frac{1}{c_0 \cdot \log(m+n)}, \quad \gamma = \frac{1}{c_0 \cdot \log(m+n)},$$

to obtain, for any $c > 1$, $c_0 \geq 20c$,

$$\|\mathcal{P}_T\mathcal{R}_\Omega\mathcal{P}_T - \mathcal{P}_T\|_{op} \leq 1/2$$

holding with probability at least

$$1 - (m+n)^{(1-c)}.$$

2.6.2 Proof of Lemma 6

We can write the matrix $(\mathcal{R}_\Omega - \mathcal{I})\,\mathbf{Z}$ as sum of independent matrices:

$$(\mathcal{R}_\Omega - \mathcal{I})\,\mathbf{Z} = \sum_{i,j} \left(\frac{1}{q_{ij}} \delta_{ij} - 1 \right) \mathbf{Z}_{ij} \mathbf{e}_i \mathbf{e}_j^T = \sum_{i,j} \mathbf{S}_{ij}.$$

We note that, $\mathbb{E}[\mathbf{S}_{ij}] = 0$, and $\mathbf{S}_{ij}$'s are zero matrix when $q_{ij} = 1$, for all (i,j). We have $\|\mathbf{S}_{ij}\|_2 \leq \frac{|\mathbf{Z}_{ij}|}{q_{ij}}$. Moreover,

$$\sum_{i,j} \mathbb{E}\left[\mathbf{S}_{ij}\mathbf{S}_{ij}^T\right] = \sum_{i,j} \mathbf{Z}_{ij}^2 \mathbf{e}_i \mathbf{e}_i^T \mathbb{E}\left[\left(\frac{1}{q_{ij}} \delta_{ij} - 1 \right)^2 \right] = \sum_i \left(\sum_j \mathbf{Z}_{ij}^2 \frac{1 - q_{ij}}{q_{ij}} \right) \mathbf{e}_i \mathbf{e}_i^T$$

Thus,

$$\left\| \sum_{i,j} \mathbb{E}\left[\mathbf{S}_{ij}\mathbf{S}_{ij}^T\right] \right\|_2 \leq \max_i \sum_{j=1}^n \frac{1 - q_{ij}}{q_{ij}} \mathbf{Z}_{ij}^2$$

Similarly,

$$\left\| \sum_{i,j} \mathbb{E}\left[\mathbf{S}_{ij}^T\mathbf{S}_{ij}\right] \right\|_2 \leq \max_j \sum_{i=1}^m \frac{1 - q_{ij}}{q_{ij}} \mathbf{Z}_{ij}^2$$

Clearly, when $q_{ij} = 1$ the above quantities are zero. Using q_{ij} in (2.12), and Lemma 10, we have

$$\|\mathbf{S}_{ij}\|_2 \leq \frac{1}{c_0 \cdot \log(m+n)} |\mathbf{Z}_{ij}| \sqrt{\frac{m}{\mu_i \varrho}} \sqrt{\frac{n}{\nu_j \varrho}} \leq \frac{\|\mathbf{Z}\|_{\mu(\infty)}}{c_0 \cdot \log(m+n)}.$$

Using q_{ij} in (2.12), and noting that $\left(\frac{\mu_i \varrho}{m} + \frac{\nu_j \varrho}{n} - \frac{\mu_i \varrho}{m} \cdot \frac{\nu_j \varrho}{n}\right) \geq \frac{\mu_i \varrho}{m}$, we have

$$\sum_{j=1}^n \frac{1 - q_{ij}}{q_{ij}} \mathbf{Z}_{ij}^2 \;\; \leq \;\; \frac{1}{c_0 \cdot \log(m+n)} \cdot \frac{m}{\mu_i \varrho} \sum_{j=1}^n \mathbf{Z}_{ij}^2 \leq \frac{\|\mathbf{Z}\|_{\mu(\infty,2)}^2}{c_0 \cdot \log(m+n)}.$$

Similarly,

$$\sum_{i=1}^{m} \frac{1-q_{ij}}{q_{ij}} \mathbf{Z}_{ij}^2 \leq \frac{1}{c_0 \cdot \log(m+n)} \cdot \frac{n}{\nu_j \varrho} \sum_{i=1}^{m} \mathbf{Z}_{ij}^2 \leq \frac{\|\mathbf{Z}\|_{\mu(\infty,2)}^2}{c_0 \cdot \log(m+n)}.$$

The lemma follows from Matrix Bernstein inequality in Lemma 11, with

$$\gamma \log(m+n) \leq \frac{1}{c_0} \|\mathbf{Z}\|_{\mu(\infty)}, \quad \sigma^2 \log(m+n) \leq \frac{1}{c_0} \|\mathbf{Z}\|_{\mu(\infty,2)}^2.$$

2.6.3 Proof of Lemma 7

Let,

$$\mathbf{X} = (\mathcal{P}_T \mathcal{R}_\Omega - \mathcal{P}_T)\mathbf{Z} = \sum_{i,j} \left(\frac{\delta_{ij}}{q_{ij}} - 1 \right) \mathbf{Z}_{ij} \mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T)$$

Weighted b-th column of $\mathbf{X}$ can be written as sum of independent, zero-mean column vectors.

$$\sqrt{\frac{n}{\nu_b \varrho}} \mathbf{X}_{*,b} = \sum_{i,j} \left(\frac{\delta_{ij}}{q_{ij}} - 1 \right) \mathbf{Z}_{ij} \left(\mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T) \mathbf{e}_b \right) \sqrt{\frac{n}{\nu_b \varrho}} = \sum_{i,j} \mathbf{s}_{ij}$$

Clearly, $\mathbb{E}[\mathbf{s}_{ij}] = 0$. We need bounds on $\|\mathbf{s}_{ij}\|_2$ and $\left\| \sum_{i,j} \mathbb{E}\left[\mathbf{s}_{ij}^T \mathbf{s}_{ij}\right] \right\|_2$ to apply Matrix Bernstein inequality. First, we need to bound $\|\mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T)\mathbf{e}_b\|_2$.

$$\begin{aligned} \|\mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T)\mathbf{e}_b\|_2 &= \|\mathbf{U}\mathbf{U}^T(\mathbf{e}_i \mathbf{e}_j^T)\mathbf{e}_b + (\mathbf{e}_i \mathbf{e}_j^T)\mathbf{V}\mathbf{V}^T\mathbf{e}_b - \mathbf{U}\mathbf{U}^T(\mathbf{e}_i \mathbf{e}_j^T)\mathbf{V}\mathbf{V}^T\mathbf{e}_b\|_2 \\ &= \begin{cases} \|\mathbf{U}\mathbf{U}^T\mathbf{e}_i + \left(\mathbf{I} - \mathbf{U}\mathbf{U}^T\right)\mathbf{e}_i \|\mathbf{V}^T\mathbf{e}_b\|_2^2\|_2 \leq \sqrt{\frac{\mu_i \varrho}{m}} + \frac{\nu_b \varrho}{n} & j = b, \\ \|\left(\mathbf{I} - \mathbf{U}\mathbf{U}^T\right)\mathbf{e}_i \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T\mathbf{e}_b\|_2 \leq \left|\mathbf{e}_j^T \mathbf{V}\mathbf{V}^T\mathbf{e}_b\right| & j \neq b, \end{cases} \end{aligned} \tag{2.27}$$

Above we use triangle inequality and definition of μ_i and ν_b. Note that, $\mathbf{s}_{ij}$ is a zero vector when $q_{ij} = 1$, for all (i, j). Otherwise, for $q_{ij} \neq 1$, we consider two cases. Using bounds in (2.27), we have for $j = b$,

$$\|\mathbf{s}_{ij}\|_2 \leq \frac{1}{q_{ib}} |\mathbf{Z}_{ib}| \sqrt{\frac{n}{\nu_b \varrho}} \left(\sqrt{\frac{\mu_i \varrho}{m}} + \frac{\nu_b \varrho}{n} \right)$$

Using q_{ij} in (2.12), $q_{ib} \geq c_0 \log(m+n)\sqrt{\frac{\mu_i \varrho}{m}}\sqrt{\frac{\nu_b \varrho}{n}}$ and $q_{ib} \geq c_0 \log(m+n) \cdot \frac{\mu_i \varrho}{m}$. Combining these two inequalities, we have

$$\|\mathbf{s}_{ij}\|_2 \log(m+n) \leq \frac{2}{c_0} |\mathbf{Z}_{ib}| \sqrt{\frac{m}{\mu_i \varrho}} \cdot \sqrt{\frac{n}{\nu_b \varrho}} \frac{\left(\sqrt{\frac{\mu_i \varrho}{m}} + \frac{\nu_b \varrho}{n}\right)}{\left(\sqrt{\frac{\mu_i \varrho}{m}} + \sqrt{\frac{\nu_b \varrho}{n}}\right)} \leq \frac{2}{c_0} \|\mathbf{Z}\|_{\mu(\infty)}$$

For $j \neq b$, using $q_{ib} \geq c_0 \log(m+n)\sqrt{\frac{\mu_i \varrho}{m}}\sqrt{\frac{\nu_b \varrho}{n}}$ (Lemma 10) and $\left|\mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b\right| \leq \sqrt{\frac{\nu_j \varrho}{n} \cdot \frac{\nu_b \varrho}{n}}$,

$$\|\mathbf{s}_{ij}\|_2 \leq \frac{1}{q_{ij}} |\mathbf{Z}_{ij}| \sqrt{\frac{n}{\nu_b \varrho}} \cdot \sqrt{\frac{\nu_j \varrho}{n}} \cdot \sqrt{\frac{\nu_b \varrho}{n}} \leq \frac{2}{c_0 \log(m+n)} \|\mathbf{Z}\|_{\mu(\infty)}$$

Therefore, for all (i,j), we have $\|\mathbf{s}_{ij}\|_2 \leq \frac{2}{c_0 \log(m+n)} \|\mathbf{Z}\|_{\mu(\infty)}$.
On the other hand,

$$\left| \sum_{i,j} \mathbb{E}\left[\mathbf{s}_{ij}^T \mathbf{s}_{ij}\right] \right| = \left(\sum_{j=b,i} + \sum_{j \neq b,i} \right) \frac{1 - q_{ij}}{q_{ij}} \mathbf{Z}_{ij}^2 \left\| \mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T)\mathbf{e}_b \right\|_2^2 \cdot \frac{n}{\nu_b \varrho}$$

The above quantity is zero for $q_{ij} = 1$. Otherwise, for $q_{ij} \neq 1$, we consider two cases. For $j = b$, using (2.27) we have, $\left\| \mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T)\mathbf{e}_b \right\|_2^2 \leq \left(\sqrt{\frac{\mu_i \varrho}{m}} + \sqrt{\frac{\nu_b \varrho}{n}}\right)^2 \leq 2\left(\frac{\mu_i \varrho}{m} + \frac{\nu_b \varrho}{n}\right)$. Using q_{ij} in (2.12), we have,

$$\sum_{j=b,i} \leq 2 \sum_i \frac{1 - q_{ib}}{q_{ib}} \mathbf{Z}_{ib}^2 \left(\frac{\mu_i \varrho}{m} + \frac{\nu_b \varrho}{n} \right) \cdot \frac{n}{\nu_b \varrho} \leq \frac{4}{c_0 \log(m+n)} \|\mathbf{Z}\|_{\mu(\infty,2)}^2 ,$$

where we use the following bound in the second inequality. For all (i,j), $q_{ij} \neq 0$,

$$\frac{\frac{\mu_i \varrho}{m} + \frac{\nu_j \varrho}{n}}{\frac{\mu_i \varrho}{m} + \frac{\nu_j \varrho}{n} - \frac{\mu_i \varrho}{m} \cdot \frac{\nu_j \varrho}{n}} = 1 + \frac{\frac{\mu_i \varrho}{m} \cdot \frac{\nu_j \varrho}{n}}{\frac{\mu_i \varrho}{m} + \frac{\nu_j \varrho}{n} - \frac{\mu_i \varrho}{m} \cdot \frac{\nu_j \varrho}{n}} \leq 1 + \frac{\frac{\mu_i \varrho}{m} \cdot \frac{\nu_j \varrho}{n}}{\max\{\frac{\mu_i \varrho}{m}, \frac{\mu_j \varrho}{n}\}} \leq 2.$$

For $j \neq b$, using $q_{ij} \geq c_0 \log(m+n) \cdot \frac{\mu_j \varrho}{n}$ and (2.27),

$$\begin{aligned}
\sum_{j \neq b, i} &\leq \sum_{j \neq b, i} \frac{1-q_{ij}}{q_{ij}} \mathbf{Z}_{ij}^2 \left| \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right|^2 \cdot \frac{n}{\nu_b \varrho} \\
&= \frac{n}{\nu_b \varrho} \sum_{j \neq b} \left| \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right|^2 \sum_i \frac{1-q_{ij}}{q_{ij}} \mathbf{Z}_{ij}^2 \\
&\leq \frac{n}{\nu_b \varrho} \sum_{j \neq b} \left| \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right|^2 \left(\frac{1}{c_0 \log(m+n)} \cdot \frac{n}{\nu_j \varrho} \sum_i \mathbf{Z}_{ij}^2 \right) \\
&\leq \left(\frac{\|\mathbf{Z}\|_{\mu(\infty,2)}^2}{c_0 \log(m+n)} \right) \frac{n}{\nu_b \varrho} \sum_{j \neq b} \left| \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right|^2 \\
&\leq \frac{\|\mathbf{Z}\|_{\mu(\infty,2)}^2}{c_0 \log(m+n)},
\end{aligned}$$

where the last inequality follows from, $\sum_{j \neq b} \left| \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right|^2 \leq \left\| \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right\|_2^2 \leq \frac{\nu_b \varrho}{n}$.
Combining the two summations,

$$\left\| \sum_{i,j} \mathbb{E}\left[\mathbf{s}_{ij}^T \mathbf{s}_{ij} \right] \right\|_2 \leq \frac{5}{c_0 \log(m+n)} \|\mathbf{Z}\|_{\mu(\infty,2)}^2$$

We can bound $\left\| \mathbb{E}\left[\sum_{i,j} \mathbf{s}_{ij} \mathbf{s}_{ij}^T \right] \right\|_2$ in a similar way.
We apply Matrix Bernstein inequality in Lemma 11, with

$$\gamma = \frac{2}{c_0 \log(m+n)} \|\mathbf{Z}\|_{\mu(\infty)}, \quad \sigma^2 = \frac{5}{c_0 \log(m+n)} \|\mathbf{Z}\|_{\mu(\infty,2)}^2,$$

to obtain

$$\left\| \sum_{i,j} \mathbf{s}_{ij} \right\|_2 \leq \sqrt{\frac{20c}{c_0}} \|\mathbf{Z}\|_{\mu(\infty,2)} + \frac{2c}{c_0} \|\mathbf{Z}\|_{\mu(\infty)}.$$

We set $c_0 \geq 80c$ to derive

$$\left\| \sqrt{\frac{n}{\nu_b \varrho}} \mathbf{X}_{*,b} \right\|_2 = \left\| \sum_{i,j} \mathbf{s}_{ij} \right\|_2 \leq \frac{1}{2} \left(\|\mathbf{Z}\|_{\mu(\infty,2)} + \|\mathbf{Z}\|_{\mu(\infty)} \right).$$

Similarly, we can bound $\left\| \sqrt{\frac{m}{\mu_a \varrho}} \mathbf{X}_{a,*} \right\|_2$ by the same quantity. We take a union bound over all rows a and all columns b (i.e., total $(m+n)$ events) to obtain, for any $c > 2$,

$$\|(\mathcal{P}_T \mathcal{R}_\Omega - \mathcal{P}_T)(\mathbf{Z})\|_{\mu(\infty,2)} \leq \frac{1}{2} \left(\|\mathbf{Z}\|_{\mu(\infty,2)} + \|\mathbf{Z}\|_{\mu(\infty)} \right)$$

holding with probability at least $1 - (m+n)^{2-c}$.

2.6.4 Proof of Lemma 8

Let, $\mathbf{X} = (\mathcal{P}_T \mathcal{R}_\Omega - \mathcal{P}_T)\mathbf{Z} = \sum_{i,j} \left(\frac{\delta_{ij}}{q_{ij}} - 1 \right) \mathbf{Z}_{ij} \left(\mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T) \right)$. We write rescaled (a,b)-th element of $\mathbf{X}$ as

$$[\mathbf{X}]_{ab} \sqrt{\frac{m}{\mu_a \varrho}} \sqrt{\frac{n}{\nu_b \varrho}} = \sum_{i,j} \left(\frac{\delta_{ij}}{q_{ij}} - 1 \right) \mathbf{Z}_{ij} \left(\mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T) \right)_{ab} \sqrt{\frac{m}{\mu_a \varrho}} \sqrt{\frac{n}{\nu_b \varrho}} = \sum_{i,j} s_{ij}$$

This is a sum of independent, zero-mean random variables. we seek to bound $|s_{ij}|$ and $\left| \sum_{i,j} \mathbb{E}\left[s_{ij}^2 \right] \right|$. First, we need to bound $\left| \left\langle \mathbf{e}_a \mathbf{e}_b^T, \mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T) \right\rangle \right|$.

$$\begin{aligned}
& \left| \left\langle \mathbf{e}_a \mathbf{e}_b^T, \mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T) \right\rangle \right| \\
= \; & \left| \mathbf{e}_a^T \mathbf{U}\mathbf{U}^T (\mathbf{e}_i \mathbf{e}_j^T) \mathbf{e}_b + \mathbf{e}_a^T (\mathbf{e}_i \mathbf{e}_j^T) \mathbf{V}\mathbf{V}^T \mathbf{e}_b - \mathbf{e}_a^T \mathbf{U}\mathbf{U}^T (\mathbf{e}_i \mathbf{e}_j^T) \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right| \\
= \; & \begin{cases}
\left\| \mathcal{P}_T(\mathbf{e}_a \mathbf{e}_b^T) \right\|_F^2 = \frac{\mu_a \varrho}{m} + \frac{\nu_b \varrho}{n} - \frac{\mu_a \varrho}{m} \cdot \frac{\nu_b \varrho}{n} & i = a, j = b, \\
\left| \mathbf{e}_a^T (\mathbf{I} - \mathbf{U}\mathbf{U}^T) \mathbf{e}_a \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right| \leq \left| \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right| & i = a, j \neq b, \\
\left| \mathbf{e}_a^T \mathbf{U}\mathbf{U}^T \mathbf{e}_i \mathbf{e}_b^T (\mathbf{I} - \mathbf{V}\mathbf{V}^T) \mathbf{e}_b \right| \leq \left| \mathbf{e}_a^T \mathbf{U}\mathbf{U}^T \mathbf{e}_i \right| & i \neq a, j = b, \\
\left| \mathbf{e}_a^T \mathbf{U}\mathbf{U}^T \mathbf{e}_i \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right| \leq \left| \mathbf{e}_a^T \mathbf{U}\mathbf{U}^T \mathbf{e}_i \right| \left| \mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b \right| & i \neq a, j \neq b
\end{cases}
\end{aligned} \tag{2.28}$$

where we use $\left\| \mathbf{I} - \mathbf{U}\mathbf{U}^T \right\|_2 \leq 1$ and $\left\| \mathbf{I} - \mathbf{V}\mathbf{V}^T \right\|_2 \leq 1$.
Note that, $s_{ij} = 0$ when $q_{ij} = 1$. Otherwise, for $q_{ij} \neq 1$,

$$|s_{ij}| \leq \frac{1}{q_{ij}} |\mathbf{Z}_{ij}| \left| \left\langle \mathbf{e}_a \mathbf{e}_b^T, \mathcal{P}_T(\mathbf{e}_i \mathbf{e}_j^T) \right\rangle \right| \sqrt{\frac{m}{\mu_a \varrho}} \sqrt{\frac{n}{\nu_b \varrho}}$$

We consider four cases.

For $i = a, j = b$, using $q_{ab} \geq c_0 \log(m+n) \left(\frac{\mu_a \varrho}{m} + \frac{\nu_b \varrho}{n} - \frac{\mu_a \varrho}{m} \cdot \frac{\nu_b \varrho}{n} \right)$

$$\begin{aligned} |s_{ij}| &\leq \frac{1}{q_{ab}} |\mathbf{Z}_{ab}| \left\| \mathcal{P}_T(\mathbf{e}_a \mathbf{e}_b^T) \right\|_F^2 \sqrt{\frac{m}{\mu_a \varrho}} \sqrt{\frac{n}{\nu_b \varrho}} \\ &\leq \frac{|\mathbf{Z}_{ab}|}{c_0 \log(m+n)} \sqrt{\frac{m}{\mu_a \varrho}} \sqrt{\frac{n}{\nu_b \varrho}} \leq \frac{\|\mathbf{Z}\|_{\mu(\infty)}}{c_0 \log(m+n)} \end{aligned}$$

For $i = a, j \neq b$, using $q_{aj} \geq c_0 \log(m+n) \left(\frac{\mu_a \varrho}{m} + \frac{\nu_j \varrho}{n} - \frac{\mu_a \varrho}{m} \cdot \frac{\nu_j \varrho}{n} \right) \geq c_0 \log(m+n) \frac{\nu_j \varrho}{n}$,

$$|s_{ij}| \leq \frac{|\mathbf{Z}_{aj}|}{q_{aj}} \left| \mathbf{e}_j^T \mathbf{V} \mathbf{V}^T \mathbf{e}_b \right| \sqrt{\frac{m}{\mu_a \varrho}} \sqrt{\frac{n}{\nu_b \varrho}} \leq \frac{|\mathbf{Z}_{aj}|}{c_0 \log(m+n)} \sqrt{\frac{n}{\nu_j \varrho}} \sqrt{\frac{m}{\mu_a \varrho}} \leq \frac{\|\mathbf{Z}\|_{\mu(\infty)}}{c_0 \log(m+n)}$$

Similarly, for $i \neq a, j = b$, using $q_{ib} \geq c_0 \log(m+n) \frac{\mu_i \varrho}{m}$

$$|s_{ij}| \leq \frac{\|\mathbf{Z}\|_{\mu(\infty)}}{c_0 \log(m+n)}.$$

For $i \neq a, j \neq b$, using $q_{ij} \geq c_0 \log(m+n) \sqrt{\frac{\mu_i \varrho}{m} \cdot \frac{\nu_j \varrho}{n}}$

$$\begin{aligned} |s_{ij}| &\leq \frac{1}{q_{ij}} |\mathbf{Z}_{ij}| \left| \mathbf{e}_a^T \mathbf{U} \mathbf{U}^T \mathbf{e}_i \right| \left| \mathbf{e}_j^T \mathbf{V} \mathbf{V}^T \mathbf{e}_b \right| \sqrt{\frac{m}{\mu_a \varrho}} \sqrt{\frac{n}{\nu_b \varrho}} \\ &\leq \frac{1}{q_{ij}} |\mathbf{Z}_{ij}| \sqrt{\frac{\mu_i \varrho}{m}} \sqrt{\frac{\mu_a \varrho}{m}} \cdot \sqrt{\frac{\nu_b \varrho}{n}} \sqrt{\frac{\nu_j \varrho}{n}} \cdot \sqrt{\frac{m}{\mu_a \varrho}} \sqrt{\frac{n}{\nu_b \varrho}} \\ &\leq \frac{1}{c_0 \log(m+n)} |\mathbf{Z}_{ij}| \sqrt{\frac{m}{\mu_i \varrho}} \sqrt{\frac{n}{\nu_j \varrho}} \leq \frac{1}{c_0 \log(m+n)} \|\mathbf{Z}\|_{\mu(\infty)}. \end{aligned}$$

Above we use $\sqrt{\frac{\mu_i \varrho}{m}} \leq 1$, $\sqrt{\frac{\nu_j \varrho}{n}} \leq 1$, for all i, j. We conclude, for all (i, j),

$$|s_{ij}| \leq \frac{1}{c_0 \log(m+n)} \|\mathbf{Z}\|_{\mu(\infty)}.$$

On the other hand,

$$\begin{aligned}\left|\sum_{i,j}\mathbb{E}\left[s_{ij}^2\right]\right| &= \sum_{i,j}\mathbb{E}\left[\left(\frac{\delta_{ij}}{q_{ij}}-1\right)^2\right]\mathbf{Z}_{ij}^2\left\langle \mathbf{e}_a\mathbf{e}_b^T, \mathcal{P}_T(\mathbf{e}_i\mathbf{e}_j^T)\right\rangle^2\frac{m}{\mu_a\varrho}\cdot\frac{n}{\nu_b\varrho}\\ &= \sum_{i,j}\frac{1-q_{ij}}{q_{ij}}\mathbf{Z}_{ij}^2\left\langle \mathbf{e}_a\mathbf{e}_b^T, \mathcal{P}_T(\mathbf{e}_i\mathbf{e}_j^T)\right\rangle^2\frac{m}{\mu_a\varrho}\cdot\frac{n}{\nu_b\varrho}\\ &= \sum_{i=a,j=b}+\sum_{i=a,j\neq b}+\sum_{i\neq a,j=b}+\sum_{i\neq a,j\neq b}\end{aligned}$$

The above quantity is zero for $q_{ij}=1$. We bound the above considering four cases for $q_{ij}\neq 1$.

For $i=a, j=b$, using $q_{ab}\geq c_0\log(m+n)\left(\frac{\mu_a\varrho}{m}+\frac{\nu_b\varrho}{n}-\frac{\mu_a\varrho}{m}\cdot\frac{\nu_b\varrho}{n}\right)$,

$$\sum_{i=a,j=b}\leq\frac{\mathbf{Z}_{ab}^2}{q_{ab}}\left(\frac{\mu_a\varrho}{m}+\frac{\nu_b\varrho}{n}-\frac{\mu_a\varrho}{m}\cdot\frac{\nu_b\varrho}{n}\right)^2\frac{m}{\mu_a\varrho}\cdot\frac{n}{\nu_b\varrho}\leq\frac{\|\mathbf{Z}\|_{\mu(\infty)}^2}{c_0\log(m+n)}$$

Above we use $\left(\frac{\mu_i\varrho}{m}+\frac{\nu_j\varrho}{n}-\frac{\mu_i\varrho}{m}\cdot\frac{\nu_j\varrho}{n}\right)\leq 1$, for all i and j.

For $i=a, j\neq b$, using $q_{aj}\geq c_0\log(m+n)\frac{\nu_j\varrho}{n}$,

$$\begin{aligned}\sum_{i=a,j\neq b} &\leq \sum_{j\neq b}\frac{1}{q_{aj}}\mathbf{Z}_{aj}^2\left|\mathbf{e}_j^T\mathbf{V}\mathbf{V}^T\mathbf{e}_b\right|^2\frac{m}{\mu_a\varrho}\frac{n}{\nu_b\varrho}\\ &\leq \frac{1}{c_0\log(m+n)}\sum_{j\neq b}\mathbf{Z}_{aj}^2\left(\frac{n}{\nu_j\varrho}\frac{m}{\mu_a\varrho}\right)\left|\mathbf{e}_j^T\mathbf{V}\mathbf{V}^T\mathbf{e}_b\right|^2\frac{n}{\nu_b\varrho}\\ &\leq \frac{1}{c_0\log(m+n)}\|\mathbf{Z}\|_{\mu(\infty)}^2.\end{aligned}$$

Above we use,

$$\sum_{j\neq b}\left|\mathbf{e}_j^T\mathbf{V}\mathbf{V}^T\mathbf{e}_b\right|^2\leq\left\|\mathbf{V}\mathbf{V}^T\mathbf{e}_b\right\|_2^2\leq\frac{\nu_b\varrho}{n}.$$

Similarly, we can derive identical bound for $\sum_{i\neq a, j=b}$.
We use $q_{ij} \geq c_0 \log(m+n)\sqrt{\frac{\mu_i \varrho}{m} \cdot \frac{\nu_j \varrho}{n}} \geq c_0 \log(m+n) \frac{\mu_i \varrho}{m} \cdot \frac{\nu_j \varrho}{n}$ to bound

$$\begin{aligned}
\sum_{i\neq a, j\neq b} &\leq \sum_{i\neq a, j\neq b} \frac{1}{q_{ij}} \mathbf{Z}_{ij}^2 \left|\mathbf{e}_a^T \mathbf{U}\mathbf{U}^T \mathbf{e}_i\right|^2 \left|\mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b\right|^2 \frac{m}{\mu_a \varrho} \frac{n}{\nu_b \varrho} \\
&\leq \frac{\|\mathbf{Z}\|_{\mu(\infty)}^2}{c_0 \log(m+n)} \sum_{i\neq a, j\neq b} \left|\mathbf{e}_a^T \mathbf{U}\mathbf{U}^T \mathbf{e}_i\right|^2 \left|\mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b\right|^2 \frac{m}{\mu_a \varrho} \frac{n}{\nu_b \varrho} \\
&= \frac{\|\mathbf{Z}\|_{\mu(\infty)}^2}{c_0 \log(m+n)} \sum_{i\neq a} \left|\mathbf{e}_a^T \mathbf{U}\mathbf{U}^T \mathbf{e}_i\right|^2 \frac{m}{\mu_a \varrho} \sum_{j\neq b} \left|\mathbf{e}_j^T \mathbf{V}\mathbf{V}^T \mathbf{e}_b\right|^2 \frac{n}{\nu_b \varrho} \\
&\leq \frac{\|\mathbf{Z}\|_{\mu(\infty)}^2}{c_0 \log(m+n)}
\end{aligned}$$

Combining the summations, we derive

$$\left|\sum_{i,j} \mathbb{E}\left[s_{ij}^2\right]\right| \leq \frac{4\|\mathbf{Z}\|_{\mu(\infty)}^2}{c_0 \log(m+n)}.$$

We now apply Bernstein inequality in Lemma 11 to obtain, for any $c > 3$, $c_0 \geq 68c$

$$\|(\mathcal{P}_T \mathcal{R}_\Omega - \mathcal{P}_T)(\mathbf{Z})\|_{\mu(\infty)} \leq \frac{1}{2}\|\mathbf{Z}\|_{\mu(\infty)}$$

We take union bound over all (a, b) (i.e., total $mn \leq (m+n)^2$ events) to conclude that the above result holds with probability at least

$$1 - (m+n)^{(3-c)}.$$

2.7 Conclusion

It is possible to recover any arbitrary low-rank data matrix exactly via the optimization problem in (2.1) using the relaxed leverage score sampling proposed in this work. This notion of relaxation in leverage scores requires a strictly smaller sample size comparing to the best-known result of [9]. Experimental results on real data sets corroborate

the theoretical analysis.

It would be an interesting problem to reduce the bound on the sample size by a logarithmic factor to $\Theta(((m+n)\varrho - \varrho^2)\log(m+n))$. This is a theoretical lower bound established by [13], and further reduction is not possible.

CHAPTER 3
CUR Decomposition with Element Sampling

3.1 Introduction

Many large $m \times n$ data $\mathbf{A}$ can be represented as m objects, each of which is described by n features. In many cases, a useful step is to find a compressed representation of $\mathbf{A}$ that might be easier to analyze and interpret. That is, we want to find a low-rank approximation to $\mathbf{A}$ (say of rank k, where $k \ll \min\{m, n\}$). SVD is the most common algorithm for finding the best rank-k approximation to $\mathbf{A}$ when the quality of approximation is measured with respect to any unitarily invariant matrix norm. The basis vectors for this best rank-k subspace are given by the singular vectors corresponding to the top k singular values. However, these basis vectors are represented by a complicated linear combination of original rows and columns, and often time very hard to interpret in terms of the underlying data and the process generating the data. Also, for certain applications computing the SVD is prohibitive due to very large dimensions of the data. Thus, it is desirable to find a high-quality low-rank approximation that is expressed in terms of small number of original rows and/or columns (rather than linear combinations of them). This is one of the main motivations for row/column subset sampling. In a seminal paper [23] show that we can sample a small number of rows and/or columns from a given matrix $\mathbf{A}$ to produce an approximation that is close to $\mathbf{A}_k$, the best rank-k approximation to $\mathbf{A}$. Their result was additive in terms of $\|\mathbf{A} - \mathbf{A}_k\|_F$, the smallest error corresponding to the optimal rank-k approximation. Later, [18] introduce a novel matrix factorization $\mathbf{A} \approx \mathbf{CUR}$, where $\mathbf{C}$ and $\mathbf{R}$ contain a small number of sampled columns and rows from original matrix $\mathbf{A}$, respectively, and $\mathbf{U}$ is a generalized inverse of their intersection that takes the product $\mathbf{CUR}$ close to $\mathbf{A}$. Their result is the first one to produce a relative-error approximation to $\|\mathbf{A} - \mathbf{A}_k\|_F$.

CUR matrix decomposition is a randomized algorithm that efficiently computes the low-rank approximation of a given matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ by sampling some of its rows and columns according to some distribution depending on the Euclidean norms of the top sin-

gular vectors ("subspace sampling", see [18], [19]). More specifically, for a given matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and a target rank k, CUR algorithm samples $c = \mathcal{O}(k^2\log(1/\delta)/\epsilon^2)$ columns of $\mathbf{A}$ to construct a matrix $\mathbf{C} \in \mathbb{R}^{m\times c}$, and then samples $r = \mathcal{O}(c^2\log(1/\delta)/\epsilon^2)$ rows of $\mathbf{A}$ to construct a matrix $\mathbf{R} \in \mathbb{R}^{r\times n}$, such that with probability at least $1-\delta$,

$$\|\mathbf{A} - \mathbf{CUR}\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F, \tag{3.1}$$

where $\mathbf{U} \in \mathbb{R}^{c\times r}$ such that $\mathbf{U} = \mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger$, where $\mathbf{X}^\dagger$ is the Moore-Penrose pseudo-inverse of matrix $\mathbf{X}$ (here and onwards).

A key step of CUR algorithms is to compute $\mathbf{U}$ by solving a least square problem involving $\mathbf{C}$, $\mathbf{R}$, and the entire matrix $\mathbf{A}$. This is a limitation for existing CUR methods as in many cases we may not have access to the whole matrix $\mathbf{A}$, e.g., a partially observed matrix. The main question we seek to answer is: can we recover a matrix $\mathbf{A}$ from a small number of observed entries such that the recovered matrix $\tilde{\mathbf{A}}$ is close to $\mathbf{A}_k$ in some matrix norm? This element-wise CUR method clearly would have few benefits over the popular nuclear norm minimization for reconstructing a matrix. First, instead of solving a trace minimization problem using semi-definite programming we may need to solve only a standard least square problem. Second, the reconstruction method for CUR decomposition is just basic multiplication of three matrices, as opposed to, solving a semi-definite program in nuclear norm minimization.

3.2 Main Results

We now present our main algorithms (Algorithm 7 and Algorithm 6) and the related Theorem 5 and Lemma 12, which are our main quality-of-approximation results for Algorithm 7 and Algorithm 6, respectively. First, we briefly describe the main notations we used in this work.

3.2.1 Notation

We use bold capital letters (e.g., $\mathbf{X}$) to denote matrices and bold lowercase letters (e.g., $\mathbf{x}$) to denote column vectors. Let $[n]$ denote the set $\{1, 2, ..., n\}$. We use $\mathbb{E}(X)$

to denote the expectation of a random variable X; when $\mathbf{X}$ is a random matrix, $\mathbb{E}(\mathbf{X})$ denotes the element-wise expectation of each entry of $\mathbf{X}$. For a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, the Frobenius norm $\|\mathbf{X}\|_F$ is defined as $\|\mathbf{X}\|_F^2 = \sum_{i,j=1}^{m,n} \mathbf{X}_{ij}^2$, the spectral norm $\|\mathbf{X}\|_2$ is defined as $\|\mathbf{X}\|_2 = \max_{\|\mathbf{y}\|_2=1} \|\mathbf{X}\mathbf{y}\|_2$, and the nuclear norm $\|\mathbf{X}\|_*$ is defined as $\|\mathbf{X}\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i(\mathbf{X})$, where $\sigma_i(\mathbf{X})$ is the i-th largest singular value of $\mathbf{X}$. For a matrix $\mathbf{A}$, we denote the best rank-k approximation to it as $\mathbf{A}_k$ using singular value decomposition in (3.2).

$$\mathbf{A} = \mathbf{A}_k + \mathbf{A}_{\varrho-k}, \tag{3.2}$$

where $\mathbf{A}_k$ is the component of $\mathbf{A}$ spanned by top k singular vectors, and $\mathbf{A}_{\varrho-k}$ is the remaining orthogonal components of $\mathbf{A}$. We define stable-rank or soft-rank of matrix $\mathbf{X}$, denoted by $\mathbf{sr}(\mathbf{X})$, as $\mathbf{sr}(\mathbf{X}) = \|\mathbf{X}\|_F^2 / \|\mathbf{X}\|_2^2$. $\mathbf{I}_n$ denotes the $n \times n$ identity matrix and $\ln x$ denotes the natural logarithm of x. Finally, we use $\mathbf{e}_i$ to denote standard basis vectors whose dimensionalities will be clear from the context.

Here we present the main technical contribution of our work in section 3.2.2. Subsequent sections contain results on some variants of it. Given a target rank k and an accuracy parameter $\epsilon > 0$, we show how to sample s elements from a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$, such that the reconstruction error is close to the optimal $\|\mathbf{A} - \mathbf{A}_k\|_F$. Specifically, we exploit the CUR based low-rank approximation framework that samples a small number of rows $\mathbf{R}$ and columns $\mathbf{C}$ of $\mathbf{A}$ to approximate the span of the best rank-k subspace of $\mathbf{A}$. Algorithm 6 samples elements from $\mathbf{C}$ and $\mathbf{R}$, and also samples elements from $\mathbf{A}$ in order to solve a regression problem to obtain a good approximation to the optimal $\mathbf{U}$. Algorithm 7 samples elements from $\mathbf{C}$ and $\mathbf{R}$, and retain all elements of $\mathbf{U}$ to reconstruct $\mathbf{A}$.

3.2.2 Reconstruction with Subspace Information and Samples from A

Here we present a reconstruction scheme for $\mathbf{A}$ when we know information about a subspace that approximates the best rank-k subspace of $\mathbf{A}$ well. Let $\mathbf{A} \in \mathbb{R}^{m\times n}$, and let $\mathbf{C} \in \mathbb{R}^{m\times c}$ and $\mathbf{R} \in \mathbb{R}^{r\times n}$, such that,

$$\begin{aligned} \left\|\mathbf{A} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\right\|_F &\leq (1+\gamma_1)\left\|\mathbf{A} - \mathbf{A}_k\right\|_F \\ \left\|\mathbf{A} - \mathbf{A}\mathbf{R}^\dagger\mathbf{R}\right\|_F &\leq (1+\gamma_2)\left\|\mathbf{A} - \mathbf{A}_k\right\|_F \end{aligned} \tag{3.3}$$

for suitably chosen values of c and r. Note that $\mathbf{C}$ and $\mathbf{R}$ could be any set of vectors that approximates reasonably well the row and column space of $\mathbf{A}$. They need not be consisting of columns and rows of $\mathbf{A}$, and they need not be orthogonal (they could).
Our objective is to find a matrix $\mathbf{U}$ such that

$$\min_{\mathbf{U}\in\mathbb{R}^{c\times r}} \|\mathbf{A}-\mathbf{CUR}\|_F^2 \tag{3.4}$$

This is easy to solve analytically when we have access to the full matrix $\mathbf{A}$, and the optimal solution to (3.4) is $\mathbf{U}^* = \mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger$ with error $\left\|\mathbf{A}-\mathbf{C}\mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\right\|_F^2$. However, we may not have access to the full matrix $\mathbf{A}$ and we show how to get a very accurate estimate of $\mathbf{U}^*$ by observing only a few elements of $\mathbf{A}$.
Assume that we form the following sample set of indices $\{(i_t, j_t)\}_{t=1}^w$ of $\mathbf{A}$ using probabilities $\{p_{ij}\}$ to be discussed later:

$$\{\mathbf{A}_{i_t j_t}\}_{t=1}^w$$

for a value of w to be discussed later. We will solve the following system of equations:

$$\min_{\mathbf{X}\in\mathbb{R}^{c\times r}} \sum_{t=1}^{w}\left(\mathbf{A}_{i_t j_t} - \mathbf{C}_{(i_t)}\mathbf{X}\mathbf{R}^{(j_t)}\right)^2 \tag{3.5}$$

where, $\mathbf{C}_{(i_t)}$ is the i_t-th row of $\mathbf{C}$, and $\mathbf{R}^{(j_t)}$ is the j_t-th column of $\mathbf{R}$.
Note that we have w equations and the unknowns are the entries of the matrix $\mathbf{X}$, which is $\mathbb{R}^{c\times r}$ and thus has $c\cdot r$ unknowns. (3.5) can be written as,

$$\min_{\mathbf{X}\in\mathbb{R}^{c\times r}} \sum_{t=1}^{w}\left(\mathbf{A}_{i_t j_t} - \sum_{k,l}\mathbf{C}_{i_t k}\mathbf{R}_{l j_t}\mathbf{X}_{kl}\right)^2 \tag{3.6}$$

We can write this in a matrix form as follows:

$$\begin{bmatrix} \mathbf{A}_{i_1 j_1} \\ \mathbf{A}_{i_2 j_2} \\ \vdots \\ \vdots \\ \mathbf{A}_{i_w j_w} \end{bmatrix} - \overbrace{\begin{bmatrix} \overbrace{\mathbf{C}_{i_1 1}\mathbf{R}_{1 j_1}\mathbf{C}_{i_1 2}\mathbf{R}_{1 j_1}\ldots\mathbf{C}_{i_1 c}\mathbf{R}_{1 j_1}}^{c}\,\mathbf{C}_{i_1 1}\mathbf{R}_{2 j_1}\ldots\mathbf{C}_{i_1 c}\mathbf{R}_{r j_1} \\ \vdots \\ \vdots \end{bmatrix}}^{c \cdot r} \begin{bmatrix} \mathbf{X}_{11} \\ \vdots \\ \mathbf{X}_{c1} \\ \vdots \\ \mathbf{X}_{cr} \end{bmatrix} \; c \cdot r$$

$$= \begin{bmatrix} \mathbf{A}_{i_1 j_1} \\ \mathbf{A}_{i_2 j_2} \\ \vdots \\ \mathbf{A}_{i_w j_w} \end{bmatrix} - \Omega \begin{bmatrix} \mathbf{X}_{11} \\ \vdots \\ \mathbf{X}_{cr} \end{bmatrix} \tag{3.7}$$

This is the least-squares problem that we will solve. We start by considering the full problem where $w = m \cdot n$, e.g., all available entries of $\mathbf{A}$ are sampled. We know this problem has the solution

$$\mathbf{X} = \mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger} \tag{3.8}$$

$$\text{and the error is} \qquad \left\|\mathbf{A} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{A}\mathbf{R}^{\dagger}\mathbf{R}\right\|_F \tag{3.9}$$

Note that the full problem has dimensionalities:

$$m \cdot n \quad \begin{bmatrix} \mathbf{A}_{i_1 j_1} \\ \mathbf{A}_{i_2 j_2} \\ \vdots \\ \mathbf{A}_{i_w j_w} \end{bmatrix} - \Omega_{m \cdot n \times c \cdot r} \begin{bmatrix} \mathbf{X}_{11} \\ \vdots \\ \mathbf{X}_{cr} \end{bmatrix} \; c \cdot r$$

namely, it is very overconstrained. This is precisely why we can actually use the result of [24] and solve it approximately by sub-sampling w constraints out of the $m \cdot n$ constraints. Specifically, Algorithm 2 of [24] guarantees that if we sample the constraints with respect to the row leverage scores of the matrix Ω, and compute $\hat{\mathbf{X}}$, the optimal solution to this sampled problem, then $\hat{\mathbf{X}}$ is an approximation to the optimal solution $\mathbf{X}_{opt}$ of (3.6). We

get, for $w = \tilde{O}(c^2r^2/\epsilon^2)$, with probability at least $1-\delta$,

$$\left\|\mathbf{A} - \mathbf{C}\hat{\mathbf{X}}\mathbf{R}\right\|_F \leq (1+\epsilon)\left\|\mathbf{A} - \mathbf{C}\mathbf{X}_{opt}\mathbf{R}\right\|_F \leq (1+\epsilon)\left\|\mathbf{A} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\right\|_F$$

It follows from [19] and (3.3),

$$\left\|\mathbf{A} - \mathbf{C}\hat{\mathbf{X}}\mathbf{R}\right\|_F \leq (1+\varepsilon)(2+\gamma_1+\gamma_2)\left\|\mathbf{A} - \mathbf{A}_k\right\|_F.$$

Note that, when $\mathbf{A}$ is low-rank, say rank k, we have $\left\|\mathbf{A} - \mathbf{C}\hat{\mathbf{X}}\mathbf{R}\right\|_F = 0$, i.e., exact reconstruction. The above reconstruction is just multiplication of three matrices, where $\hat{\mathbf{X}}$ is tiny, and this is considerably simpler comparing to a semi-definite program use to solve a convex nuclear norm minimization.

3.2.3 Reconstruction using Samples from C, R, and A

Here we consider a variant of the above method where we cannot use all the elements of $\mathbf{C}$ and $\mathbf{R}$. We use sparsification method of Algorithm 1 to sample elements from $\mathbf{C}$ and $\mathbf{R}$, and then apply similar regression method as above.

Sparsified matrix $\tilde{\mathbf{C}}$ can be interpreted as noisy $\mathbf{C}$ where the zero-mean random noise matrix $\mathbf{C} - \tilde{\mathbf{C}}$ is added to $\mathbf{C}$. Similarly, $\mathbf{R} - \tilde{\mathbf{R}}$ is the additive noise producing $\tilde{\mathbf{R}}$ from $\mathbf{R}$. Here we consider a regression step to de-noise the reconstructed matrix $\tilde{\mathbf{A}}$ by considering additional elements from $\mathbf{A}$ sampled according to a particular probability distribution. Given $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$, our objective is to find a matrix $\mathbf{U}$ such that

$$\min_{\mathbf{U}\in\mathbb{R}^{c\times r}} \left\|\mathbf{A} - \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}}\right\|_F^2 \tag{3.10}$$

This is easy to solve analytically when we have access to the full matrix $\mathbf{A}$, and the optimal solution to (3.10) is $\mathbf{U}^* = \tilde{\mathbf{C}}^\dagger\mathbf{A}\tilde{\mathbf{R}}^\dagger$. However, we may not have access to the full matrix $\mathbf{A}$, and following the earlier method we can get a good estimate of $\mathbf{U}^*$ by observing only a few elements of $\mathbf{A}$ according to the row leverage scores of the matrix $\mathbf{\Phi}$ in (3.12).

We have the following least-squares problem for (3.10) as follows,

$$\min_{\mathbf{X}\in\mathbb{R}^{c\times r}} \sum_{t=1}^{w} \left(\mathbf{A}_{i_t j_t} - \sum_{k,l=1}^{c,r} \tilde{\mathbf{C}}_{i_t k} \tilde{\mathbf{R}}_{l j_t} \mathbf{X}_{kl} \right)^2 \tag{3.11}$$

Let us write this in matrix form.

$$\begin{aligned} & \begin{bmatrix} \mathbf{A}_{i_1 j_1} \\ \mathbf{A}_{i_2 j_2} \\ \vdots \\ \vdots \\ \mathbf{A}_{i_w j_w} \end{bmatrix} - \overbrace{\begin{bmatrix} \overbrace{\tilde{\mathbf{C}}_{i_1 1}\tilde{\mathbf{R}}_{1 j_1}\tilde{\mathbf{C}}_{i_1 2}\tilde{\mathbf{R}}_{1 j_1}\ldots\tilde{\mathbf{C}}_{i_1 c}\tilde{\mathbf{R}}_{1 j_1}}^{c} \tilde{\mathbf{C}}_{i_1 1}\tilde{\mathbf{R}}_{2 j_1}\ldots\tilde{\mathbf{C}}_{i_1 c}\tilde{\mathbf{R}}_{r j_1} \\ \vdots \\ \vdots \end{bmatrix}}^{c\cdot r} \begin{bmatrix} \mathbf{X}_{11} \\ \vdots \\ \mathbf{X}_{c1} \\ \vdots \\ \mathbf{X}_{cr} \end{bmatrix} \\ = & \begin{bmatrix} \mathbf{A}_{i_1 j_1} \\ \mathbf{A}_{i_2 j_2} \\ \vdots \\ \mathbf{A}_{i_w j_w} \end{bmatrix} - \mathbf{\Phi} \begin{bmatrix} \mathbf{X}_{11} \\ \vdots \\ \mathbf{X}_{cr} \end{bmatrix} \end{aligned} \tag{3.12}$$

Considering the full problem where $w = m \cdot n$, e.g., all available entries of $\mathbf{A}$ are sampled, we know this problem has the solution

$$\begin{aligned} & \mathbf{X}_{opt} = \tilde{\mathbf{C}}^{\dagger}\mathbf{A}\tilde{\mathbf{R}}^{\dagger} \\ \text{and the error is} \quad & \left\| \mathbf{A} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^{\dagger}\mathbf{A}\tilde{\mathbf{R}}^{\dagger}\tilde{\mathbf{R}} \right\|_F \end{aligned} \tag{3.13}$$

Algorithm 2 of [24] guarantees that if we sample the constraints with respect to the row leverage scores of the matrix $\mathbf{\Phi}$, and compute $\tilde{\mathbf{X}}$ (the optimal solution to this sampled problem), then $\tilde{\mathbf{X}}$ is an approximation to the optimal solution $\mathbf{X}_{opt}$ of (3.13). We get, for $w = \tilde{O}(c^2 r^2/\epsilon^2)$, with probability at least $1-\delta$,

$$\left\| \mathbf{A} - \tilde{\mathbf{C}}\tilde{\mathbf{X}}\tilde{\mathbf{R}} \right\|_F \leq (1+\epsilon) \left\| \mathbf{A} - \tilde{\mathbf{C}}\mathbf{X}_{opt}\tilde{\mathbf{R}} \right\|_F = (1+\epsilon) \left\| \mathbf{A} - \tilde{\mathbf{C}}\tilde{\mathbf{C}}^{\dagger}\mathbf{A}\tilde{\mathbf{R}}^{\dagger}\tilde{\mathbf{R}} \right\|_F$$

Lemma 12 *Let $\tilde{\mathbf{A}} = \tilde{\mathbf{C}}\tilde{\mathbf{U}}\tilde{\mathbf{R}}$ (Algorithm 6). If sample size $s = \tilde{\mathcal{O}}((m+n)k/\epsilon^6 + k^4/\epsilon^{16})$,*

Algorithm 6 CUR-based Element-wise Sampling with Regression

1: **Input:** $\mathbf{A} \in \mathbb{R}^{m\times n}$, rank parameter k, and accuracy parameter $\epsilon > 0$.
2: Find $\mathbf{C} \in \mathbb{R}^{m\times c}$, and $\mathbf{R} \in \mathbb{R}^{r\times n}$ satisfying (3.3).
3: Element-wise sparsify $\mathbf{C}$ and $\mathbf{R}$ to obtain $\tilde{\mathbf{C}} = \mathcal{S}_\Omega(\mathbf{C})$ and $\tilde{\mathbf{R}} = \mathcal{S}_\Omega(\mathbf{R})$, using probabilities of the form (1.2) in Algorithm 1.
4: Construct matrix $\mathbf{\Phi}$, and sample set $\mathcal{S}_\Omega(\mathbf{A})$ using probabilities proportional to the row leverage scores of $\mathbf{\Phi}$ in (3.12), using Algorithm 1.
5: Solve

$$\min_{\mathbf{X}\in\mathbb{R}^{c\times r}} \left\|\mathcal{S}_\Omega(\mathbf{A}) - \tilde{\mathbf{C}}\mathbf{X}\tilde{\mathbf{R}}\right\|_F \tag{3.14}$$

Let $\tilde{\mathbf{U}}$ be the optimal solution to (3.14).
6: **Output:** Sampled pairs of indices (and corresponding rescaled elements) of $\tilde{\mathbf{C}}$, $\tilde{\mathbf{R}}$, and $\tilde{\mathbf{A}}$, and all the elements of matrix $\tilde{\mathbf{U}}$.
7: **Reconstruction:**

$$\tilde{\mathbf{A}} = \tilde{\mathbf{C}}\tilde{\mathbf{U}}\tilde{\mathbf{R}}$$

then, for $\delta \in (0,1)$, the following holds with probability at least $1-3\delta$,

$$\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_F \leq (1+\epsilon)^2 \left\|\mathbf{A} - \mathbf{A}_k\right\|_F + \epsilon(1+\epsilon)\cdot(\alpha_1+\alpha_2)$$

where, $\alpha_1 = \left\|\mathbf{C}\right\|_2 \left\|\mathbf{U}\mathbf{R}\right\|_F$ and $\alpha_2 = 2\left\|\mathbf{R}\right\|_2 \left\|\mathbf{U}\right\|_2 \sum_{i,j=1}^{m,n} |\mathbf{C}_{ij}|$.

Proof:

$$\left\|\mathbf{A} - \tilde{\mathbf{C}}\tilde{\mathbf{U}}\tilde{\mathbf{R}}\right\|_F \leq (1+\epsilon)\left\|\mathbf{A} - \tilde{\mathbf{C}}\mathbf{X}_{opt}\tilde{\mathbf{R}}\right\|_F \leq (1+\epsilon)\left\|\mathbf{A} - \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}}\right\|_F$$

The result follows from Theorem 5.

◇

The bound on s can be improved to $s = \tilde{\mathcal{O}}((m+n)k/\epsilon^6 + k^2/\epsilon^8)$ using the result of [25].

3.2.4 Reconstruction using Samples from only C and R

Here we assume that we have access to some elements of $\mathbf{C}$ and $\mathbf{R}$, and all the elements of $\mathbf{U}$.

Algorithm 7 CUR-based Element-wise Sampling

1: **Input:** $\mathbf{C} \in \mathbb{R}^{m\times c}$, $\mathbf{U} \in \mathbb{R}^{c\times r}$, and $\mathbf{R} \in \mathbb{R}^{r\times n}$, such that $\mathbf{A} \approx \mathbf{CUR}$.
2: Element-wise sparsify $\mathbf{C}$ and $\mathbf{R}$ to obtain $\tilde{\mathbf{C}} = \mathcal{S}_\Omega(\mathbf{C})$ and $\tilde{\mathbf{R}} = \mathcal{S}_\Omega(\mathbf{R})$, using probabilities of the form (1.2) in Algorithm 1.
3: **Output:** Sampled pairs of indices (and corresponding rescaled elements) of $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$, and all the elements of matrix $\mathbf{U}$.
4: **Reconstruction:**

$$\tilde{\mathbf{A}} = \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}}$$

Theorem 5 *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a given matrix, integer k be a rank parameter, and $\epsilon > 0$ be an accuracy parameter. Let $\mathbf{A} = \mathbf{CUR}$ (in Algorithm 7). Let $\tilde{\mathbf{A}}$ be the reconstructed matrix from the samples of $\mathbf{C}$ and $\mathbf{R}$ (in Algorithm 7), such that $\tilde{\mathbf{A}} = \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}}$. If sample size $s = \tilde{\mathcal{O}}((m+n)k/\epsilon^6)$, then, for $\delta \in (0,1)$, the following holds with probability at least $1-2\delta$,*

$$\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_F \;\leq\; (1+\epsilon)\left\|\mathbf{A} - \mathbf{A}_k\right\|_F + \epsilon\cdot(\alpha_1+\alpha_2)$$

where, $\alpha_1 = \left\|\mathbf{C}\right\|_2 \left\|\mathbf{UR}\right\|_F$ and $\alpha_2 = 2\left\|\mathbf{R}\right\|_2 \left\|\mathbf{U}\right\|_2 \sum_{i,j=1}^{m,n} |\mathbf{C}_{ij}|$.

Further, computation complexity of Algorithm 7 is dominated by computation of rank-k SVD of $\mathbf{A}$, $\mathcal{O}(mnk)$. Finally, reconstruction of $\tilde{\mathbf{A}}$ takes time $\tilde{\mathcal{O}}(mk^2/\epsilon^6 + mnk/\epsilon^4)$.

Note that, reconstruction of $\tilde{\mathbf{A}}$ is only the product of three matrices $\tilde{\mathbf{C}}$, $\mathbf{U}$, and $\tilde{\mathbf{R}}$, where $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$ are sparse matrices, and $\mathbf{U}$ is tiny. Therefore, we can expect the computation of $\tilde{\mathbf{A}}$ to be very fast, although we use the upper bound for general matrix multiplication in the analysis of computational complexity. Also, we use CUR decomposition as a 'black box', and express the bounds in Theorem 5 in terms of matrices $\mathbf{C}$, $\mathbf{U}$, and $\mathbf{R}$. The bounds can be improved by choosing a better CUR algorithm. For instance, using the CUR algorithm described in [19], we get $s = \tilde{\mathcal{O}}((m+n)k/\epsilon^4)$ and reconstruction time of $\tilde{\mathbf{A}}$ as $\tilde{\mathcal{O}}(mk^2/\epsilon^4 + mnk/\epsilon^2)$ to produce the final bound $\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_F \leq (2+\epsilon)\left\|\mathbf{A} - \mathbf{A}_k\right\|_F + \epsilon\cdot(\alpha_1+\alpha_2)$.

Corollary 1 *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a given matrix, and let $\tilde{\mathbf{A}} = \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}}$ (in Algorithm 7). If sample size $s = \tilde{\mathcal{O}}((m+n)k/\epsilon^6)$, then, for $\delta \in (0,1)$, each of the following results holds*

with probability at least $1-2\delta$*,*

$$\left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_2 \;\leq\; (1+\epsilon)\left\|\mathbf{A}-\mathbf{A}_k\right\|_F + \epsilon\cdot(\alpha_1+\alpha_2) \tag{3.15}$$

$$\left\|\mathbf{A}_k-\tilde{\mathbf{A}}\right\|_F \;\leq\; (2+\epsilon)\left\|\mathbf{A}-\mathbf{A}_k\right\|_F + \epsilon\cdot(\alpha_1+\alpha_2) \tag{3.16}$$

$$\left\|\mathbf{A}_k-\tilde{\mathbf{A}}\right\|_2 \;\leq\; \left\|\mathbf{A}-\mathbf{A}_k\right\|_2 + (1+\epsilon)\left\|\mathbf{A}-\mathbf{A}_k\right\|_F + \epsilon\cdot(\alpha_1+\alpha_2) \tag{3.17}$$

$$\left\|\mathbf{A}_k-(\tilde{\mathbf{A}})_k\right\|_F \;\leq\; \sqrt{2k}\,(\left\|\mathbf{A}-\mathbf{A}_k\right\|_2) + \sqrt{8k}\,((1+\epsilon)\left\|\mathbf{A}-\mathbf{A}_k\right\|_F + \epsilon\cdot(\alpha_1+\alpha_2)) \tag{3.18}$$

where, $\alpha_1 = \left\|\mathbf{C}\right\|_2 \left\|\mathbf{U}\mathbf{R}\right\|_F$ *and* $\alpha_2 = 2\left\|\mathbf{R}\right\|_2 \left\|\mathbf{U}\right\|_2 \sum_{i,j=1}^{m,n} |\mathbf{C}_{ij}|$.

Proof:
(3.15) follows trivially from Theorem 5 as $\left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_2 \leq \left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_F$.
$\left\|\mathbf{A}_k-\tilde{\mathbf{A}}\right\|_F \leq \left\|\mathbf{A}_k-\mathbf{A}\right\|_F + \left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_F$. (3.16) follows from Theorem 5.
$\left\|\mathbf{A}_k-\tilde{\mathbf{A}}\right\|_2 \leq \left\|\mathbf{A}-\mathbf{A}_k\right\|_2 + \left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_2$. (3.17) follows from (3.15).
Note that, $\text{rank}(\mathbf{A}_k-(\tilde{\mathbf{A}})_k) \leq \text{rank}(\mathbf{A}_k) + \text{rank}(\tilde{\mathbf{A}})_k) = 2k$. Thus,

$$\begin{aligned}
\left\|\mathbf{A}_k-(\tilde{\mathbf{A}})_k\right\|_F &\leq \sqrt{2k}\left\|\mathbf{A}_k-(\tilde{\mathbf{A}})_k\right\|_2 \\
&\leq \sqrt{2k}\left(\left\|\mathbf{A}-\mathbf{A}_k\right\|_2 + \left\|\mathbf{A}-(\tilde{\mathbf{A}})_k\right\|_2\right) \\
&\leq \sqrt{2k}\left(\left\|\mathbf{A}-\mathbf{A}_k\right\|_2 + \left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_2 + \left\|\tilde{\mathbf{A}}-(\tilde{\mathbf{A}})_k\right\|_2\right) \\
&\leq \sqrt{2k}\left(\left\|\mathbf{A}-\mathbf{A}_k\right\|_2 + 2\left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_2\right) \\
&= \sqrt{2k}\left\|\mathbf{A}-\mathbf{A}_k\right\|_2 + \sqrt{8k}\left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_2
\end{aligned}$$

Thus, (3.18) follows from (3.15).

$\diamond$

Algorithm 8 CUR Sparsification Algorithm

1: **Input:** $\mathbf{A} \in \mathbb{R}^{m \times n}$, rank parameter k, and accuracy parameter $\epsilon > 0$.
2: Decompose $\mathbf{A}$ to get $\mathbf{C} \in \mathbb{R}^{m \times c}$, $\mathbf{U} \in \mathbb{R}^{c \times r}$, and $\mathbf{R} \in \mathbb{R}^{r \times n}$, such that $\mathbf{A} \approx \mathbf{CUR}$.
3: Element-wise sparsify $\mathbf{C}$ to get $\tilde{\mathbf{C}}$ using Algorithm 1.
4: Element-wise sparsify $\mathbf{R}$ to get $\tilde{\mathbf{R}}$ using Algorithm 1.
5: Solve

$$\min_{\mathbf{X} \in \mathbb{R}^{c \times r}} \left\| \mathbf{CUR} - \tilde{\mathbf{C}}\mathbf{X}\tilde{\mathbf{R}} \right\|_F \tag{3.19}$$

Let $\tilde{\mathbf{U}}$ be the optimal solution to (3.19).
6: **Output:** Sampled pairs of indices (and corrresponding elements) of $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{R}}$, and the entire matrix $\tilde{\mathbf{U}}$.
7: **Reconstruction:**

$$\tilde{\mathbf{A}} = \tilde{\mathbf{C}}\tilde{\mathbf{U}}\tilde{\mathbf{R}}$$

3.2.5 CUR based Matrix Sparsification

We can extend the above idea in the domain of matrix sparsification problem based on CUR decomposition. Algorithm 8 proposes a sparsification scheme depending on CUR decomposition. Here we assume that $\mathbf{A}$ is given in terms of matrices $\mathbf{C}$, $\mathbf{U}$, and $\mathbf{R}$. Theorem (6) shows the quality of approximation result for Algorithm 8.

Theorem 6 *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a given matrix, integer k be a rank parameter, and $\epsilon > 0$ be an accuracy parameter. Let $\mathbf{A} = \mathbf{CUR}$ (in Algorithm 8). Let $\tilde{\mathbf{A}}$ be the sparsified matrix (in Algorithm 8), such that $\tilde{\mathbf{A}} = \tilde{\mathbf{C}}\tilde{\mathbf{U}}\tilde{\mathbf{R}}$. If sample size $s = \tilde{\mathcal{O}}((m+n)k/\epsilon^6)$, then, for $\delta \in (0,1)$, the following holds with probability at least $1 - 2\delta$,*

$$\left\| \mathbf{A} - \tilde{\mathbf{A}} \right\|_F \;\leq\; (1+\epsilon) \left\| \mathbf{A} - \mathbf{A}_k \right\|_F + \epsilon \cdot (\alpha_1 + \alpha_2)$$

where, $\alpha_1 = \left\| \mathbf{C} \right\|_2 \left\| \mathbf{UR} \right\|_F$ and $\alpha_2 = 2 \left\| \mathbf{R} \right\|_2 \left\| \mathbf{U} \right\|_2 \sum_{i,j=1}^{m,n} |\mathbf{C}_{ij}|$.

Proof:

$$\left\| \mathbf{A} - \tilde{\mathbf{A}} \right\|_F = \left\| \mathbf{A} - \tilde{\mathbf{C}}\tilde{\mathbf{U}}\tilde{\mathbf{R}} \right\|_F \leq \left\| \mathbf{A} - \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}} \right\|_F$$

We can follow the proof of Theorem 5 to derive the results.

$\diamond$

3.3 Tools

In this section we describe the main tools and results we need to give a proof of Theorem 5. Our framework requires the CUR decomposition as a 'black box'. There are quite a few variants of CUR decomposition available in the literature [18], [19], [26]. We rephrase the classic work of [18] below.

Lemma 13 *[Theorem 2 of [18]] Given* $\mathbf{A} \in \mathbb{R}^{m\times n}$*, a target rank* $k \ll \min\{m,n\}$*, and an accuracy parameter* $\epsilon > 0$*, we can apply the subspace sampling algorithm (Algorithm 2 in [18]) to select (in expectation)* $c = \mathcal{O}(k\epsilon^{-2}\ln k \ln(1/\delta))$ *columns and* $r = \mathcal{O}(c\epsilon^{-2}\ln c \ln(1/\delta))$ *rows. Then,*

$$\|\mathbf{A} - \mathbf{CUR}\|_F \leq (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F$$

holds with probability at least $1-\delta$*. Note that the sampling is performed without replacement. The running time is dominated by computation of truncated SVD of* $\mathbf{A}$*, i.e.,* $\mathcal{O}(mnk)$*.*

The following element-wise sparsification result is due to [7]. This gives us a very simple proof to show that element-wise sampling can produce a sparse sketch of a given matrix that is almost as good, with high probability, if we sample sufficiently large elements. We rephrase Theorem 1 and Corollary 1 of [7].

Lemma 14 *[Theorem 1 of [7]] Let* $\mathbf{X} \in \mathbb{R}^{m\times n}$ *and let* $\epsilon > 0$ *be an accuracy parameter. Let* $\mathcal{S}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ *be the sampling operator defined in the element-wise sampling algorithm (Algorithm 1), and assume that the sampling probabilities* $\{p_{ij}\}_{i,j=1}^{m,n}$ *satisfy*

$$p_{ij} \geq \frac{\beta}{2}\left(\frac{\mathbf{X}_{ij}^2}{\|\mathbf{X}\|_F^2} + \frac{|\mathbf{X}_{ij}|}{\sum_{i,j=1}^{m,n}|\mathbf{X}_{ij}|}\right) \tag{3.20}$$

for all i,j *and some* $\beta \in (0,1]$*. If* $\boldsymbol{sr}(\mathbf{X}) \geq \epsilon^2$*, and* $s \geq \frac{6\max\{m,n\}\ln((m+n)/\delta)}{\beta\epsilon^2}\boldsymbol{sr}(\mathbf{X})$*, then, with probability at least* $1-\delta$*,*

$$\|\mathbf{X} - \mathcal{S}_\Omega(\mathbf{X})\|_2 \leq \epsilon\|\mathbf{X}\|_2.$$

Noting that $\mathbf{sr}(\mathbf{X}) \leq \text{rank}(\mathbf{X})$, the above result suggests that when $\mathbf{X}$ is potentially very low-rank (say almost constant), we need only linear number (up to logarithm factor) of samples to get a sparse sketch of $\mathbf{X}$ with a relative error guarantee in spectral norm.
Note that we can replace Lemma 14 by Theorem 1 presented in this document to derive a tighter bound on number of samples required in order to achieve a given accuracy. We mention Lemma 14 due to simplicity of results.
Finally, we need the following theorem in order to bound $\left\|\tilde{\mathbf{C}}\right\|_F$.

Lemma 15 *Let $\mathbf{X} \in \mathbb{R}^{m\times n}$ and let $\mathcal{S}_\Omega : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ be the sampling operator defined in the element-wise sampling algorithm (Algorithm 1), and assume that the sampling probabilities $\{p_{ij}\}_{i,j=1}^{m,n}$ satisfy (3.20). Then,*

$$\|\mathcal{S}_\Omega(\mathbf{X})\|_F \leq \frac{2}{\beta}\sum_{i,j=1}^{m,n} |\mathbf{X}_{ij}|$$

Proof: For any matrix $\mathbf{X}$, we can write, $\mathbf{X} = \mathbf{X}_{ij}\mathbf{e}_i\mathbf{e}_j^T$. Then, for s samples,

$$\mathcal{S}_\Omega(\mathbf{X}) = \frac{1}{s}\sum_{t=1}^{s} \frac{\mathbf{X}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t}\mathbf{e}_{j_t}^T$$

Then,

$$\|\mathcal{S}_\Omega(\mathbf{X})\|_F \leq \frac{1}{s}\sum_{t=1}^{s} \left\|\frac{\mathbf{X}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t}\mathbf{e}_{j_t}^T\right\|_F \leq \frac{1}{s}\sum_{t=1}^{s} \max_t \frac{|\mathbf{X}_{i_t j_t}|}{p_{i_t j_t}} \leq \frac{2}{\beta}\sum_{i,j=1}^{m,n} |\mathbf{X}_{ij}|$$

using the following from (3.20)

$$p_{i_t j_t} \geq \frac{\beta}{2}\frac{|\mathbf{X}_{i_t j_t}|}{\sum_{i,j=1}^{m,n} |\mathbf{X}_{ij}|}$$

$\diamond$

3.4 Proof of Theorem 5

In this section we give a proof of Theorem 5 which is our main result on the quality of approximation for Algorithm 4. Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a given matrix, and $\tilde{\mathbf{A}}$ be the

reconstructed matrix according to Algorithm 4, where $\tilde{\mathbf{A}} = \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}}$. Then, we have

$$\left\|\mathbf{A} - \tilde{\mathbf{A}}\right\|_F \leq \underbrace{\|\mathbf{A} - \mathbf{CUR}\|_F}_{\text{error(CUR)}} + \underbrace{\left\|\mathbf{CUR} - \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}}\right\|_F}_{\text{error(Sampling)}} \tag{3.21}$$

We have from Lemma 13,

$$\text{error(CUR)} \leq (1+\epsilon)\left\|\mathbf{A} - \mathbf{A}_k\right\|_F$$

and

$$\begin{aligned}
\text{error(Sampling)} &= \left\|\mathbf{CUR} - \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}}\right\|_F \\
&\leq \left\|\mathbf{CUR} - \tilde{\mathbf{C}}\mathbf{UR}\right\|_F + \left\|\tilde{\mathbf{C}}\mathbf{UR} - \tilde{\mathbf{C}}\mathbf{U}\tilde{\mathbf{R}}\right\|_F \\
&= \left\|(\mathbf{C} - \tilde{\mathbf{C}})\mathbf{UR}\right\|_F + \left\|\tilde{\mathbf{C}}\mathbf{U}(\mathbf{R} - \tilde{\mathbf{R}})\right\|_F \\
&\leq \left\|\mathbf{C} - \tilde{\mathbf{C}}\right\|_2 \left\|\mathbf{UR}\right\|_F + \left\|\tilde{\mathbf{C}}\mathbf{U}\right\|_F \left\|\mathbf{R} - \tilde{\mathbf{R}}\right\|_2 \\
&\leq \epsilon \cdot \left\|\mathbf{C}\right\|_2 \left\|\mathbf{UR}\right\|_F + \epsilon \cdot \left\|\mathbf{R}\right\|_2 \left\|\tilde{\mathbf{C}}\mathbf{U}\right\|_F \\
&\leq \epsilon \cdot \left\|\mathbf{C}\right\|_2 \left\|\mathbf{UR}\right\|_F + \epsilon \cdot \left\|\mathbf{R}\right\|_2 \left\|\tilde{\mathbf{C}}\right\|_F \left\|\mathbf{U}\right\|_2 \\
&\leq \epsilon \cdot \left\|\mathbf{C}\right\|_2 \left\|\mathbf{UR}\right\|_F + \frac{2\epsilon}{\beta}\left\|\mathbf{R}\right\|_2 \left\|\mathbf{U}\right\|_2 \sum_{i,j=1}^{m,n} |\mathbf{C}_{ij}|
\end{aligned}$$

Setting $\beta = 1$, we get the final expression. Above we use triangle inequality, and $\|\mathbf{XY}\|_F \leq \|\mathbf{X}\|_F \|\mathbf{Y}\|_2$ and $\|\mathbf{XY}\|_F \leq \|\mathbf{X}\|_2 \|\mathbf{Y}\|_F$. Lemma 14 bounds $\left\|\mathbf{C} - \tilde{\mathbf{C}}\right\|_2$ and $\left\|\mathbf{R} - \tilde{\mathbf{R}}\right\|_2$, and Lemma 15 bounds $\left\|\tilde{\mathbf{C}}\right\|_F$.

Bound on sample size:

Choosing $c = \mathcal{O}(k\epsilon^{-2}\ln k \ln(1/\delta))$ and $r = \mathcal{O}(c\epsilon^{-2}\ln c \ln(1/\delta))$ as in Lemma 13, we can guarantee, with failure probability at most δ,

$$\left\|\mathbf{A} - \mathbf{CUR}\right\|_F \leq (1+\epsilon)\left\|\mathbf{A} - \mathbf{A}_k\right\|_F$$

Simplifying the expressions for c and r, we get $c = \tilde{\mathcal{O}}(k/\epsilon^2)$ and $r = \tilde{\mathcal{O}}(k/\epsilon^4)$. We need a sample size of $s_r = \frac{6n\ln((r+n)/\delta)}{\beta\epsilon^2}\mathbf{sr}(\mathbf{R})$ to sparsify $\mathbf{R}$ to get $\tilde{\mathbf{R}}$ such that $\left\|\mathbf{R} - \tilde{\mathbf{R}}\right\|_2 \leq \epsilon \left\|\mathbf{R}\right\|_2$ holds with failure probability at most δ (Lemma 14). Note that $\mathbf{sr(R)} \leq \text{rank}(\mathbf{R}) \leq r$. We simplify the expression to get $s_r = \tilde{\mathcal{O}}(nk/\epsilon^6)$. Similarly, we need a sample size $s_c = \tilde{\mathcal{O}}(mk/\epsilon^4)$ to get $\left\|\mathbf{C} - \tilde{\mathbf{C}}\right\|_2 \leq \epsilon \left\|\mathbf{C}\right\|_2$, holding with failure probability at most δ. Finally, we need to store all the elements of $\mathbf{U}$, i.e., $s_u = \tilde{\mathcal{O}}(k^2/\epsilon^6)$. Thus, overall we need a sample size $s = \tilde{\mathcal{O}}((m+n)k/\epsilon^6)$. Also, we get 2δ as the upper bound for failure probability by applying the union bound.

Computation complexity:

It takes $\mathcal{O}(mnk)$ for SVD($\mathbf{A}, k$) to perform CUR decomposition. Assuming sampling an element from a matrix takes constant time, element-wise sampling from $\mathbf{C}$ can take time linear in its number of elements, i.e., $\mathcal{O}(mc)$. Similarly, complexity for $\mathbf{R}$ is $\mathcal{O}(nr)$. Thus, overall runtime is dominated by $\mathcal{O}(mnk)$.

Reconstruction of the approximate matrix $\tilde{\mathbf{A}}$ is basically multiplication of three matrices $\tilde{\mathbf{C}}$, $\mathbf{U}$, and $\tilde{\mathbf{R}}$. This takes time $\tilde{\mathcal{O}}(mk^2/\epsilon^6 + mnk/\epsilon^4)$ using basic matrix multiplication.

This completes the proof of Theorem 5.

REFERENCES

[1] D. Achlioptas and F. McSherry, "Fast computation of low rank matrix approximations," in *Proc. Symp. Theory Computing*, 2001, pp. 611–618.

[2] D. Achlioptas and F. McSherry, "Fast computation of low-rank matrix approximations," *J. ACM*, vol. 54, no. 2, pp. 9, Apr. 2007.

[3] D. Achlioptas, Z. Karnin, and E. Liberty, "Matrix entry-wise sampling: simple is best," 2013. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.297.576rep=rep1type=pdf (Date Last Accessed 06/14/2015).

[4] P. Drineas and A. Zouzias, "A note on element-wise matrix sparsification via a matrix-valued Bernstein inequality," *Inform. Process. Lett.*, vol. 111, no. 8, pp. 385–389, Mar. 2011.

[5] S. Arora, E. Hazan, and S. Kale, "A fast random sampling algorithm for sparsifying matrices," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techn.*, 2006, pp. 272–279.

[6] B. Recht, "A simpler approach to matrix completion," *J. Mach. Learning Res.*, vol. 12, pp. 3413–3430, Feb. 2011.

[7] A. Kundu and P. Drineas, "A note on randomized element-wise matrix sparsification," 2014. [Online]. Available: http://arxiv.org/pdf/1404.0320v1.pdf (Date Last Accessed 06/14/2015).

[8] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast Monte Carlo algorithms for matrices I: approximating matrix multiplication," *SIAM J. Computing*, vol. 36, no. 1, pp. 132–157, 2006.

[9] Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward, "Coherent matrix completion," in *Proc. Int. Conf. Mach. Learning*, 2014, pp. 674–682.

[10] E. Gabrilovich and S. Markovitch, "Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5," in *Proc. Int. Conf. Mach. Learning*, 2004, pp. 41.

[11] J. J. Hull, "A database for handwritten text recognition research," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[12] E. J. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Computational Mathematics*, vol. 9, no. 6, pp. 717–772, Dec. 2009.

[13] E. J. Candes and T. Tao, "The power of convex relaxation: near-optimal matrix completion," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.

[14] D. Gross, "Recovering low-rank matrices from few coefficients in any basis," *IEEE Trans. Inform. Theory*, vol. 57, no. 3, pp. 1548–1566, Mar. 2011.

[15] B. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Computing*, vol. 24, no. 2, pp. 227–234, 1995.

[16] M. Fazel, "Matrix rank minimization with applications," PhD dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, 2002.

[17] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11, May 2011.

[18] P. Drineas, M. Mahoney, and S. Muthukrishnan, "Relative-error CUR matrix decompositions," *SIAM J. Matrix Anal. and Appl.*, vol. 30, no. 2, pp. 844–881, Aug. 2008.

[19] M. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," *Proc. Nat. Academy Sci.*, vol. 106, no. 3, pp. 697–702, Jan. 2009.

[20] A. Kundu, "Relaxed leverage sampling for low-rank matrix completion," 2015. [Online]. Available: http://arxiv.org/pdf/1503.06379v2.pdf (Date Last Accessed 06/14/2015).

[21] S. Chatterjee and A. Hadi, "Influential observations, high leverage points, and outliers in linear regression," *Statistical Sci.*, vol. 1, no. 3, pp. 379–393, Aug. 1986.

[22] J. Tropp, "User-friendly tail bounds for sums of random matrices," *Found. Computational Mathematics*, vol. 12, no. 4, pp. 389–434, Aug. 2012.

[23] A. Frieze, R. Kannan, and S. Vempala, "Fast Monte-Carlo algorithms for finding low-rank approximations," in *Proc. Annu. IEEE Symp. Found. Comput. Sci.*, 1998, pp. 370–378.

[24] P. Drineas, M. Mahoney, and S. Muthukrishnan, "Sampling algorithms for l_2 regression and applications," in *ACM-SIAM Symp. Discrete Algorithms*, 2006, pp. 1127–1136.

[25] C. Boutsidis, P. Drineas, and M. Magdon-Ismail, "Near-optimal coresets for least-squares regression," 2013. [Online]. Available: http://arxiv.org/pdf/1202.3505v2.pdf (Date Last Accessed 06/14/2015).

[26] S. Wang and Z. Zhang, "Improving CUR matrix decomposition and the Nystrom approximation via adaptive sampling," *J. Mach. Learning Res.*, vol. 14, no. 1, pp. 2549–2589, Jan. 2013.

[27] A. Kundu, P. Drineas, and M. Magdon-Ismail, "Recovering PCA from hybrid-(ℓ_1, ℓ_2) sparse sampling of data elements," 2015. [Online]. Available: http://arxiv.org/pdf/1503.00547v1.pdf (Date Last Accessed 06/14/2015).