

Matrix Completion with Heterogeneous Observation Cost Using Sparsity-Number of Column-Space

Ilqar Ramazanli

Abstract—The matrix completion problem has been studied broadly under many underlying conditions. In many real-life scenarios, we could expect elements from distinct columns or distinct positions to have a different cost. In this paper, we explore this generalization under adaptive conditions. We approach the problem under two different cost models. The first one is that entries from different columns have different observation costs, but, within the same column, each entry has a uniform cost. The second one is any two entry has different observation cost, despite being the same or different columns. We provide complexity analysis of our algorithms and provide tightness guarantees.

Keywords—Matrix completion, adaptive learning, heterogeneous cost, Matroid optimization.

I. INTRODUCTION

MATRIX completion and compressed sensing has gained attention of machine learning and signal processing communities recently. This attention particularly increased since modern data analysis gained research attention and low-rank structures has started to be studied in detail. Announcement of Netflix Prize problem has especially triggered these studies and many researchers started studying underlying structure of real-life datasets and their singular value decomposition. One of the earliest work in this field is due to [7] which discusses that the problem could be solved as convex relaxation of the simplest problem that could be attained from observed data:

$$\begin{aligned} & \text{minimize} && \text{rank}(X) \\ & \text{subject to} && X_{ij} = M_{i,j} \text{ for } (i, j) \in \Omega \end{aligned}$$

which is discussed in [10] that solving this problem is exponential in terms of the size of the matrix both theoretically and practically. Therefore, the following relaxation of the problem has been suggested in [7]:

$$\begin{aligned} & \text{minimize} && \|X\|_* \\ & \text{subject to} && X_{ij} = M_{i,j} \text{ for } (i, j) \in \Omega \end{aligned}$$

where $\|\cdot\|_*$ stands for nuclear norm minimization. Authors has proved that given $|\Omega| > Cn^{1.2}r \log n$ and if the matrix is not coherent then the problem above has a unique solution given $n = \max(n_1, n_2)$. In follow-up works, the problem has been studied in more details and more tight bounds has been suggested and proved [8], [13], [29].

Ilqar Ramazanli is with Meta, New York, NY 10003 USA (phone 412-436-6794; e-mail: iramazan@alumni.cmu.edu).

Recently it has been shown in many problems that adaptive sensing gives stronger performance guarantees [23], [30] compared to traditional methods. Active learning has been comprehensively studied in many survey papers [33], [38]. For optimization problems, adaptivity has been studied in [3], [21], [28], [35], for curriculum learning it is studied in [15], [18], [36] and the idea has been applied to many other problems.

In terms of compressed sensing and exact matrix completion, some of the earliest works are due [11], [12], [19]. Later many active matrix completion algorithms were proposed by researchers [5], [22], [32]. In the streamline of works [19], [20] authors suggest a single phase algorithm which is further extended by [2], [25]–[27].

Matrix completion problem has been studied both with noise and noiseless setting. In noisy setting, generally we try to estimate the underlying low-rank structure, using the idea of to threshold singular values. Some of the example works are [1], [6], [17], [31], [34]. In noiseless setting, we generally call the problem exact low-rank matrix completion and it also has been in the center of attention of many researchers [4], [9], [16], [37].

So far all the discussed matrix completion algorithms described here work assuming that observation complexity of each entry is the same across entries. An interesting and intuitive extension of this problem is to assume different entries of the matrix having different cost. It is in a way intuitive, that in an example problem of rating movies, watching longer movies requires more time cost than watching shorter ones. Also assuming that different people having different time availability, even the cost of the same length movie could be perceived to be different across various people.

Firstly, we will assume that underlying $m \times n$ sized rank- r matrix \mathbf{M} has different cost for each column. Specifically, there is a set of costs $\{c_1, c_2, \dots, c_n\}$ which any entry \mathbf{M}_{ij} has the cost c_j no matter which index we pick for i . Secondly, we assume that there is a set of cost entries $\{c_{11}, \dots, c_{mn}\}$ that each entry i, j of the matrix \mathbf{M}_{ij} has the cost of c_{ij} . We give different optimality condition based on the different cost model.

A. Preliminaries

We briefly introduce notations and definitions used in this paper. We use similar notations as it has been used before in the literature to keep the consistency. \mathbf{M} stands for the target matrix that we want to recover which has size of $m \times n$ and rank of r . $\|x\|$ denote the L_2 norm of a vector $x \in \mathbb{R}^n$. x_Ω denote the sub-vector of x where coordinates are from Ω . For

any $\mathbf{R} \subset [m]$, $\mathbf{M}_{\mathbf{R}}$ stands for an $|\mathbf{R}| \times n$ sized submatrix of \mathbf{M} that rows are restricted by \mathbf{R} . We define $\mathbf{M}_{\mathbf{C}}$ in a similar way for restriction with respect to columns. Intuitively, $\mathbf{M}_{\mathbf{R}:\mathbf{C}}$ defined for $|\mathbf{R}| \times |\mathbf{C}|$ sized submatrix of \mathbf{M} with rows restricted to \mathbf{R} and columns restricted to \mathbf{C} . Moreover, for the special case \mathbf{M}_{i} stands for i -th row and $\mathbf{M}_{:j}$ stands for the j 'th column. Similarly, $\mathbf{M}_{i:\mathbf{C}}$ will represent the restriction of the row i by \mathbf{C} and $\mathbf{M}_{\mathbf{R}:j}$ represents restriction of the column j by \mathbf{R} . As mentioned in [24] the sparsity number is defined by first defining the nonsparsity number:

$$\psi(\mathbf{M}) = \min\{\|x\|_0 \mid x = \mathbf{M}z \text{ and } z \notin \text{null}(\mathbf{M})\}$$

$$\psi(\mathbb{U}) = \min\{\|x\|_0 \mid x \in \mathbb{U} \text{ and } x \neq 0\}$$

then using this number we define sparsity-number:

$$\bar{\psi}(x) = m - \psi(x)$$

$$\bar{\psi}(\mathbf{M}) = m - \psi(\mathbf{M})$$

$$\bar{\psi}(\mathbb{U}) = m - \psi(\mathbb{U})$$

II. MAIN RESULTS

The algorithm **ERCS** is just modification of the algorithm [19] or [25] in the input phase. We show observing just $d = \bar{\psi}(\mathbb{U}) + 1$ many entries in each column is enough to decide whether partially observed column is contained in the subspace or not.

Under the condition that the column space \mathbb{U} of the underlying matrix satisfies $r - 1 = \bar{\psi}(\mathbb{U})$, the observation complexity of the algorithm **ERCS** becomes $m \times r + (n - r) \times r = (m + n - r)r$ which is the degree of freedom of the set of $m \times n$ sized rank- r matrices. Therefore, under this condition **ERCS** is absolutely optimal

Algorithm 1 ERCS Exact Recovery with Column Sparsity

Input: $d = \bar{\psi}(\mathbb{U}) + 1$

Initialize: $k = 0, \hat{\mathbf{U}}^0 = \emptyset$

- 1: Draw uniformly random entries $\Omega \subset [m]$ of size d
- 2: Observe entire \mathbf{M}_{Ω} :
- 3: **for** i from 1 to n **do**
- 4: **if** $\|\mathbf{M}_{\Omega:i} - \mathcal{P}_{\mathbb{U}_{\Omega}}\mathbf{M}_{\Omega:i}\| > 0$
- 5: Fully observe $\mathbf{M}_{:i}$
- 6: $\hat{\mathbf{U}}^{k+1} \leftarrow \hat{\mathbf{U}}^k \cup \mathbf{M}_{:i}$
- 7: Orthogonalize $\hat{\mathbf{U}}^{k+1}$
- 8: $k = k + 1$
- 9: **otherwise:** $\hat{\mathbf{M}}_{:i} = \hat{\mathbf{U}}^k \hat{\mathbf{U}}_{\Omega}^{k+} \hat{\mathbf{M}}_{\Omega:i}$
- 10: **end for**

Output: Underlying matrix $\hat{\mathbf{M}}$

Theorem 1: Let \mathbb{U} represent the column space of the $m \times n$ sized matrix \mathbf{M} of rank r . Then, **ERCS** exactly recovers \mathbf{M} by

$$m \times r + (n - r)(\bar{\psi}(\mathbb{U}) + 1)$$

observations.

Proof: We start by showing **ERCS** recovers \mathbf{M} exactly and later we focus on observation count. To prove correctness of exact recovery we use mathematical induction as follows:

Hypothesis: after i -th iteration **ERCS** already correctly recovered first i columns.

Base case: $i = 1$ is trivial as if at least one of the observed entries is nonzero we completely observe the column, which guarantees correctness. On the other hand, if it happens all of $\bar{\psi}(\mathbb{U}) + 1$ entries are zero, then the first column is indeed completely zero because the definition of the *space sparsity number* implies there can be at most $\psi(\mathbb{U})$ many zero coordinates in a nonzero vector in the column space.

Hypothesis proof: Let us assume after step $i - 1$, **ERCS** recovered first $i - 1$ columns correctly and we want to show that the algorithm exactly recovers i -th column too. We use the lemma 1 and 6 from [27] to verify this fact:

Lemma 1: Let \mathbb{U} be a subspace of \mathbb{R}^m and x^1, x^2, \dots, x^n be any set of vectors from \mathbb{U} . Then the linear dependence of $x_{\Omega}^1, x_{\Omega}^2, \dots, x_{\Omega}^n$ implies linear dependence of x^1, x^2, \dots, x^n , for any $\Omega \subset [m]$ such that $|\Omega| > \psi(\mathbb{U})$.

Lemma 2: Let \mathbb{U} be a subspace of \mathbb{R}^m and x^1, x^2, \dots, x^n be any set of vectors from \mathbb{U} . Then the linear dependence of x^1, x^2, \dots, x^n implies linear dependence of $x_{\Omega}^1, x_{\Omega}^2, \dots, x_{\Omega}^n$, for any $\Omega \subset [m]$.

From the design of the algorithm $\mathbf{M}_{\Omega:i}$ is already observed. Then, if in the line 4, **ERCS** decides the column is linearly independent with previous columns, as in the next line we completely observe the column there is no chance that the algorithm can do mistake under this case. Therefore, the only remaining case is, if in the line 4 the algorithm decides the column i is linearly dependent.

From the statement of lemma 1, if a set of vectors from a subspace \mathbb{U} are linearly dependent on a given subset of coordinates, then they are indeed linearly dependent. We conclude that the algorithm's decision is correct and by just back projection method, the algorithm recovers remaining entries of the partially observed column. Therefore, column i also recovered correctly and we are done with the proof of induction hypothesis.

Our next goal is to show the observation complexity is $m \times r + (n - r)(\bar{\psi}(\mathbb{U}) + 1)$. From the lemma 1, we conclude that whenever current column is indeed linearly independent with previous columns, the **ERCS** also decides it is linearly independent. Moreover, from the lemma 10, we conclude that if the current column is linearly dependent with previous columns, then in this case **ERCS** decides it is linearly dependent. As there are r many linearly independent columns in the underlying matrix \mathbf{M} , the algorithm decides independence exactly r times and in each of them it does complete observations. However, in remaining $n - r$ columns, number of observations is exactly $\bar{\psi}(\mathbb{U}) + 1$. As a conclusion, number of total observations is: $rm + (n - r)(\bar{\psi}(\mathbb{U}) + 1)$ ■

Corollary 2: **ERCS** still performs correctly under the case in each column, number of observed entries is strictly lower bounded by $\bar{\psi}(\mathbb{U})$: $d \geq \bar{\psi}(\mathbb{U}) + 1$. Moreover, the number of observations will be $m \times r + (n - r)d$

Proof: The proof is exactly proceeds as proof of the theorem. The key point is to notice, lemma 1 and lemma 2

are still satisfying.

a) *Optimality of ERCS*:: We notice that many adaptive algorithms such as [19] and ERCS have two stages of observations.

- Select subset of rows and observe them completely.
- Detect linearly independent columns and observe them completely.

The discussion for tightness of the lemma 1 implies that ERCS is optimal deterministic two stage observation, low-rank exact recovery algorithm. Moreover, in the corollary we discussed for any $d \geq \bar{\psi}(\mathbb{U})$ the ERCS algorithm would still perform correctly. Therefore, having constant factor approximation of the *space sparsity number* of the column space would lead asymptotically optimal algorithm:

$$\begin{aligned} \hat{d} \leq Kd &\implies \hat{d} + 1 \leq K(d + 1) \\ &\implies (n - r)(\hat{d} + 1) \leq (n - r)K(d + 1) \end{aligned}$$

adding mr to both side leads to

$$\begin{aligned} rm + (n - r)(\hat{d} + 1) &\leq rm + (n - r)K(d + 1) \\ &\leq Krm + K(n - r)(d + 1) \\ &= K(rm + (n - r)(d + 1)) \end{aligned}$$

Notice that $rm + (n - r)(\hat{d} + 1)$ is the observation complexity we have and $(rm + (n - r)(d + 1))$ is optimal two stage as we discussed here. All together, the inequality implies constant approximation to *space sparsity number* gives as constant approximation to optimal solution.

Here, we discuss matrix completion problem with relatively different setting. Previously, we focused on the case, where each of the entry of underlying matrix \mathbf{M} has the same observation cost. In this section we discuss the completion problem where entries of the matrix has non uniform cost to observe. We tackle with two type of non uniformity here:

- Entries has uniform cost across the same column, but different columns has different costs.
- Each entry of the matrix has different cost.

A. Uniform Cost Across Columns

Problem: For any fixed j , the cost of observing $\mathbf{M}_{i,j}$ is equal to χ_j for any $1 \leq i \leq m$, and $\chi_1, \chi_2, \dots, \chi_n$ are arbitrary positive numbers and we target to recover the matrix \mathbf{M} as cheap as possible.

Solution: We propose a slight modification of the ERCS to solve optimally among the two staged methods as we discussed before. Lets remind that in the algorithm we show that selecting any $d = \bar{\psi}(\mathbb{U}) + 1$ many rows is enough to guarantee exact recovery deterministically. In the next stage, we iteratively go through columns one by one starting with the first column, and if we detect a column is linearly independent with previous ones, we completely observe it. If not, we recover it using the pre-determined subspace. To adapt the solution for this problem, we just need to change the order of

the columns we start to check. Basically, instead of starting with the first column, we should start with the cheapest one. If we decide its not contained in the current subspace, we completely observe all entries and if it is contained then we just recover with the current subspace. Then, we move to second cheapest column and so on so forth with the increasing order of cost.

Correctness: We can see that the proof of the correctness of ERCS is independent of the order of the columns. Therefore, selecting columns with increasing order of the cost would not change the correctness of the algorithm.

Optimality: The set of two stage algorithm can be parameterized by two numbers. First one is - d - the number of rows fully observed and the second is the subset of indices of columns to observe fully. We analyse the optimal algorithm for three cases of values of d :

1. $d \leq \bar{\psi}(\mathbb{U})$. It is obvious that optimal algorithm cannot have $d \leq \bar{\psi}(\mathbb{U})$, because from the discussion for tightness of the lemma 1 and optimality of ERCS, there are matrices that selection of $d = \bar{\psi}(\mathbb{U})$ rows is not enough to guarantee the existence of r linearly independent rows.

2. $d = \bar{\psi}(\mathbb{U}) + 1$ It is a well known fact that the set of column bases are matroids and Greedy algorithms gives the optimal solution for matroids [14]. Note that the algorithm discussed is efficient way of giving greedy solution.

3. $d > \bar{\psi}(\mathbb{U}) + 1$. Let us assume that the optimal algorithm takes $\tilde{d} > d$ rows in the first phases and columns: $i_1, i_2, \dots, i_{\tilde{r}}$. We first note that, $\tilde{r} = r$, it is because if $\tilde{r} < r$ then selected columns are not enough to learn the column space and if $\tilde{r} > r$ we can pick subset of these columns that is basis for column space and selecting this basis has less cost which contradicts to optimality. Therefore, $\tilde{r} = r$ for optimal case. Moreover, we can use the same subset selection argument to pick $\bar{\psi}(\mathbb{U}) + 1$ sized subset of rows then select the same set of columns and it will be cheaper. Therefore, for optimality we should select exactly $d = \bar{\psi}(\mathbb{U}) + 1$ rows.

B. Exact Recovery with Full Heterogeneity

Problem: For any i, j , the cost of observing $\mathbf{M}_{i,j}$ is equal to χ_{ij} and $\chi_{11}, \chi_{12}, \dots, \chi_{mn}$ are arbitrary positive numbers and similar to the previous problem we target to recover the matrix \mathbf{M} as cheap as possible.

Solution: We describe the solution in the following algorithm:

a) *Correctness*:: We can see that the correctness of ERHC is due to the correctness of ERCS as selecting cheapest $\bar{\psi}(\mathbb{U}) + 1$ is special case of selecting any $\bar{\psi}(\mathbb{U}) + 1$ many columns and iteration order over the columns doesn't matter similarly for this case too.

b) *Optimality*:: Unlike to the previous case, greedy algorithm does not give us the cheapest combination of columns and rows. Following example provides a matrix and entry costs that shows that greedy algorithm is not optimal.

Algorithm 2 ERHC: Exact recovery with heterogeneous cost

Input: $d = \bar{\psi}(\mathbb{U}) + 1$ here \mathbb{U} is the column space of the underlying matrix

Initialize: $\widehat{\mathbf{M}}$ set to $m \times n$ sized null matrix, $\widehat{\mathbf{U}}^0 = \emptyset$, $k = 0$

```

1: for  $i$  from 1 to  $m$ , do
2:    $\chi^i = \sum_{j=1}^n \chi_{ij}$ 
3: end for
4: Sort  $\chi^i$ s with increasing order and select first  $d$  and denote their index set by  $-R$ 
5: Observe entire  $\mathbf{M}_{R:}$ 
6: for  $i$  from 1 to  $m$ , do
7:    $\bar{\chi}^i = \sum_{j \in [n] \setminus R} \chi_{ij}$ 
8: end for
9: Sort  $\bar{\chi}^i$ s with increasing order and lets denote  $\{i_1, i_2, \dots, i_n\}$  as  $\bar{\chi}^{i_1} \leq \bar{\chi}^{i_2} \leq \dots \leq \bar{\chi}^{i_n}$ 
10: for  $h$  from 1 to  $m$ , do
11:   If  $\|\mathbf{M}_{R:i_h} - \mathcal{P}_{\widehat{\mathbf{U}}_R^k} \mathbf{M}_{R:i_h}\|^2 > 0$ 
12:     Fully observe  $\mathbf{M}_{:i_h}$  add it to the basis  $\widehat{\mathbf{U}}^k$ 
13:     Orthogonalize  $\widehat{\mathbf{U}}^k$ 
14:      $k = k + 1$ 
15:   Otherwise:  $\widehat{\mathbf{M}}_{:i_h} = \widehat{\mathbf{U}}^k \widehat{\mathbf{U}}_R^{k+} \widehat{\mathbf{M}}_{R:i_h}$ 
16: end for
17: return  $\widehat{\mathbf{M}}$ 

```

Output: Underlying matrix $\widehat{\mathbf{M}}$

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 4 & 5 \\ 1 & 4 & 5 & 6 \end{bmatrix} \quad \chi = \begin{bmatrix} 1 & 1 & 4 & 1 \\ 1 & 5 & 3 & 4 \\ 4 & 3 & 4 & 4 \\ 1 & 4 & 4 & 8 \end{bmatrix}$$

The greedy algorithm for this case observes rows $R = \{1, 2\}$ and columns $C = \{1, 2\}$ which has overall cost of:

$$(1 + 1 + 4 + 1) + (1 + 5 + 3 + 4) + (4 + 1) + (3 + 4) = 32$$

However, observing $R = \{1, 3\}$ and columns $C = \{1, 3\}$ would give us overall cost of:

$$(1 + 1 + 4 + 1) + (4 + 3 + 4 + 4) + (1 + 1) + (3 + 4) = 31$$

which is cheaper than greedy algorithm. However, with the same cost matrix, there are other matrices that shares the same column space as \mathbf{M} (therefore the same column space sparsity number) but greedy algorithm is still optimal. For the same cost matrix with a slightly modified underlying matrix, we can give an example:

$$\widehat{\mathbf{M}} = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 2 & 2 & 3 \\ 1 & 3 & 2 & 4 \\ 1 & 4 & 2 & 5 \end{bmatrix} \quad \chi = \begin{bmatrix} 1 & 1 & 4 & 1 \\ 1 & 5 & 3 & 4 \\ 4 & 3 & 4 & 4 \\ 1 & 4 & 4 & 8 \end{bmatrix}$$

This gives us the conclusion, with just information of the observation cost matrix and column space sparsity number,

we cannot pick theoretical optimal set of rows and columns that is guaranteed carrying all of information of the underlying matrix.

c) *2-Optimality*:: Even though greedy algorithm cannot return the optimal set of rows and columns, here we show that the overall cost of the cost of the algorithm is at most twice expensive than optimal.

We denote the row set and column set parameter of optimal 2-stage algorithm \tilde{R} and \tilde{C} and cost of it by σ_{OPT} . Then, we can decompose optimal solution into its parts as following:

$$\sigma_{OPT} = \chi(\mathbf{M}_{\tilde{R}:}) + \chi(\mathbf{M}_{:\tilde{C}}) - \chi(\mathbf{M}_{\tilde{R}:\tilde{C}})$$

Trivially, both of the following inequalities satisfied

$$\begin{aligned} \chi(\mathbf{M}_{\tilde{R}:\tilde{C}}) &\leq \chi(\mathbf{M}_{:\tilde{C}}) \\ \chi(\mathbf{M}_{\tilde{R}:\tilde{C}}) &\leq \chi(\mathbf{M}_{\tilde{R}:}) \end{aligned}$$

for which these inequalities implies that

$$\sigma_{OPT} \geq \max(\chi(\mathbf{M}_{:\tilde{C}}), \chi(\mathbf{M}_{\tilde{R}:}))$$

Now lets decompose cost of greedy algorithm to its pieces:

$$\sigma_G = \chi(\mathbf{M}_{R:}) + \chi(\mathbf{M}_{:C}) - \chi(\mathbf{M}_{R:C})$$

Note that the greedy algorithm does not necessarily selects cheapest basis columns, however selected columns minimizes the overall cost after rows selected. Therefore, we conclude that if we denote the set of cheapest columns by C^B , then the following inequality satisfied:

$$\begin{aligned} \sigma_G &= \chi(\mathbf{M}_{R:}) + \chi(\mathbf{M}_{:C}) - \chi(\mathbf{M}_{R:C}) \\ &\leq \chi(\mathbf{M}_{R:}) + \chi(\mathbf{M}_{:C^B}) - \chi(\mathbf{M}_{R:C^C}) \\ &\leq 2 \max(\chi(\mathbf{M}_{R:}), \chi(\mathbf{M}_{:C^B})) \end{aligned}$$

As we discussed before in order to have guarantee that we will be able to have full information to detect linearly independent columns we need to observe at least $\psi(\mathbb{U}) + 1$ many rows. Moreover, as greedy algorithm observe exactly $\bar{\psi}(\mathbb{U}) + 1$ many rows by choosing cheapest columns we are guaranteed to have:

$$\chi(\mathbf{M}_{R:}) \leq \chi(\mathbf{M}_{\tilde{R}:})$$

Similarly as C^B represents the set of cheapest columns, we have:

$$\chi(\mathbf{M}_{:C^B}) \leq \chi(\mathbf{M}_{:\tilde{C}})$$

which together implies

$$\max(\chi(\mathbf{M}_{R^*}), \chi(\mathbf{M}_{C^*})) \leq \max(\chi(\mathbf{M}_{\tilde{R}^*}), \chi(\mathbf{M}_{\tilde{C}^*})).$$

Putting all inequalities together we conclude:

$$\begin{aligned} \sigma_G &\leq 2 \max(\chi(\mathbf{M}_{R^*}), \chi(\mathbf{M}_{C^*})) \\ &\leq 2 \max(\chi(\mathbf{M}_{\tilde{R}^*}), \chi(\mathbf{M}_{\tilde{C}^*})) \\ &\leq 2\sigma_{OPT} \end{aligned}$$

Therefore, we conclude that greedy algorithm gives us 2-optimal algorithm.

d) *Tightness*:: In the following example, we see that greedy algorithm cannot guarantee better than 2-optimality:

$$\chi = \begin{bmatrix} \frac{\epsilon}{100} & \frac{\epsilon}{100} & \frac{\epsilon}{100} & \frac{\epsilon}{100} & 10 - \epsilon & 10 - \epsilon \\ \frac{\epsilon}{100} & \frac{\epsilon}{100} & \frac{\epsilon}{100} & \frac{\epsilon}{100} & 10 - \epsilon & 10 - \epsilon \\ 10 & 10 & \frac{\epsilon}{100} & \frac{\epsilon}{100} & \frac{\epsilon}{100} & \frac{\epsilon}{100} \\ 10 & 10 & \frac{\epsilon}{100} & \frac{\epsilon}{100} & \frac{\epsilon}{100} & \frac{\epsilon}{100} \\ \frac{\epsilon}{100} & \frac{\epsilon}{100} & 10 & 10 & 10 - \epsilon & 10 - \epsilon \\ \frac{\epsilon}{100} & \frac{\epsilon}{100} & 10 & 10 & 10 - \epsilon & 10 - \epsilon \end{bmatrix}$$

It is clear that optimal choice is $C = \{1, 2\}$ and $R = \{3, 4\}$ which gives the cost of:

$$\sigma_{OPT} = 10 + 10 + 10 + 10 + 16 \times \frac{\epsilon}{100} = 40 + \frac{\epsilon}{6.25}$$

However, greedy algorithm will pick $R = \{1, 2\}$ in the first stage which has overall cost of

$$\begin{aligned} &(10 - \epsilon) + (10 - \epsilon) + (10 - \epsilon) + (10 - \epsilon) \\ &+ 8 \times \frac{\epsilon}{100} \\ &= 40 - 4\epsilon + \frac{\epsilon}{12.5} \end{aligned}$$

Then in the next stage it choose columns $C = \{5, 6\}$ which also has cost of

$$\begin{aligned} &(10 - \epsilon) + (10 - \epsilon) + (10 - \epsilon) + (10 - \epsilon) \\ &+ 4 \times \frac{\epsilon}{100} \\ &= 40 - 4\epsilon + \frac{\epsilon}{25} \end{aligned}$$

All together cumulative cost is

$$(40 - 4\epsilon) + \frac{\epsilon}{12.5} + (40 - 4\epsilon) + \frac{\epsilon}{25} = 80 - 8\epsilon + \frac{3}{25}\epsilon.$$

To find the fraction of this cost to optimal cost we get

$$\frac{\sigma_G}{\sigma_{OPT}} = \frac{80 - 8\epsilon + \frac{2\epsilon}{25}}{40 + \frac{\epsilon}{6.25}} \approx 2 - \frac{\epsilon}{5}.$$

Therefore for any number smaller than 2, we can choose an ϵ which ratio of the cost of greedy algorithm to optimal set is larger than that number. This implies that, 2-optimality of the algorithm **ERHC** is tight.

III. CONCLUSION

In this paper we have discussed heterogeneous matrix problem. Matrix completion has been mainly studied in uniform cost model, which in many real-life applications this seems to be unrealistic. Different features might have different costs to compute them, and even each feature might have different cost to different users. In that light, we think it's necessary to study matrix completion in this setting. Interesting extension of this work would be in the space of low-rank estimation. In real-life examples, data are mainly coming as low-rank structure with additional small noise. Studying this problem with heterogeneous model, finding cheapest solution to the estimation problem would cover many real-life scenarios.

ACKNOWLEDGMENT

We would like to thank professor Barnabas Poczos for his insightful discussion in the space of matrix completion and heterogeneity cost model. His ideas inspired author of this paper to solve low-rank matrix completion under this scenario.

REFERENCES

- [1] Candes, Emmanuel J., and Yaniv Plan. *Matrix completion with noise*. Proceedings of the IEEE 98.6 (2010): 925-936.
- [2] Balcan, Maria-Florina F., and Hongyang Zhang. *Noise-tolerant life-long matrix completion via adaptive sampling*. Advances in Neural Information Processing Systems 29 (2016).
- [3] Bécigneul, Gary, and Octavian-Eugen Ganea. *Riemannian adaptive optimization methods*. arXiv preprint arXiv:1810.00760 (2018).
- [4] Bertsimas, Dimitris, and Michael Lingzhi Li. *Fast exact matrix completion: A unified optimization framework for matrix completion*. Journal of Machine Learning Research 21.231 (2020): 1-43.
- [5] Bhargava, Aniruddha, Ravi Ganti, and Rob Nowak. *Active positive semidefinite matrix completion: Algorithms, theory and applications*. Artificial Intelligence and Statistics. PMLR, 2017.
- [6] Cai, Jian-Feng, Emmanuel J. Candès, and Zuowei Shen. *A singular value thresholding algorithm for matrix completion*. SIAM Journal on optimization 20.4 (2010): 1956-1982.
- [7] Candès, Emmanuel J., and Benjamin Recht. *Exact matrix completion via convex optimization*. Foundations of Computational mathematics 9.6 (2009): 717-772.

- [8] Candès, Emmanuel J., and Terence Tao. *The power of convex relaxation: Near-optimal matrix completion*. IEEE Transactions on Information Theory 56.5 (2010): 2053-2080.
- [9] Candès, Emmanuel J., and Benjamin Recht. *Exact low-rank matrix completion via convex optimization*. 2008 46th Annual Allerton Conference on Communication, Control, and Computing. IEEE, 2008.
- [10] Chistov, Alexander L., and D. Yu Grigor'Ev. *Complexity of quantifier elimination in the theory of algebraically closed fields*. International Symposium on Mathematical Foundations of Computer Science. Springer, Berlin, Heidelberg, 1984.
- [11] Deshpande, Amit, et al. *Matrix approximation and projective clustering via volume sampling*. Theory of Computing 2.1 (2006): 225-247.
- [12] Frieze, Alan, Ravi Kannan, and Santosh Vempala. *Fast Monte-Carlo algorithms for finding low-rank approximations*. Journal of the ACM (JACM) 51.6 (2004): 1025-1041.
- [13] Gross, David. *Recovering low-rank matrices from few coefficients in any basis*. IEEE Transactions on Information Theory 57.3 (2011): 1548-1566.
- [14] Helman, Paul, Bernard ME Moret, and Henry D. Shapiro. *An exact characterization of greedy structures*. SIAM Journal on Discrete Mathematics 6.2 (1993): 274-283.
- [15] Huang, Yuge, et al. *Curricularface: adaptive curriculum learning loss for deep face recognition*. proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
- [16] Jain, Prateek, and Praneeth Netrapalli. *Fast exact matrix completion with finite samples*. Conference on Learning Theory. PMLR, 2015.
- [17] Keshavan, Raghunandan, Andrea Montanari, and Sewoong Oh. *Matrix completion from noisy entries*. Advances in neural information processing systems 22 (2009).
- [18] Kong, Yajing, et al. *Adaptive Curriculum Learning*. Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
- [19] Krishnamurthy, Akshay, and Aarti Singh. *Low-rank matrix and tensor completion via adaptive sampling*. Advances in neural information processing systems 26 (2013).
- [20] Krishnamurthy, Akshay, and Aarti Singh. *On the power of adaptivity in matrix completion and approximation*. arXiv preprint arXiv:1407.3619 (2014).
- [21] Ma, Jerry, and Denis Yarats. *On the adequacy of untuned warmup for adaptive optimization*. arXiv preprint arXiv:1910.04209 7 (2019).
- [22] Misir, Mustafa. *Active matrix completion for algorithm selection*. International Conference on Machine Learning, Optimization, and Data Science. Springer, Cham, 2019.
- [23] Paramythis, Alexandros, and Susanne Loidl-Reisinger. *Adaptive learning environments and e-learning standards*. Second european conference on e-learning. Vol. 1. No. 2003. 2003.
- [24] Ramazanli, Ilqar. *Adaptive noisy matrix completion*. arXiv preprint arXiv:2203.08340 (2022).
- [25] Ramazanli, Ilqar. *Lifelong matrix completion with sparsity-number*. arXiv preprint arXiv:2203.07637 (2022).
- [26] Ramazanli, Ilqar. *Adaptive noisy matrix completion*. arXiv preprint arXiv:2203.08340 (2022).
- [27] Ramazanli, Ilqar, and Barnabas Poczos. *Optimal exact matrix completion under new parametrization*. arXiv preprint arXiv:2002.02431 (2020).
- [28] Ramazanli, Ilqar, et al. *Adaptive sampling distributed stochastic variance reduced gradient for heterogeneous distributed datasets*. arXiv preprint arXiv:2002.08528 (2020).
- [29] Recht, Benjamin. *A simpler approach to matrix completion*. Journal of Machine Learning Research 12.12 (2011).
- [30] Riedmiller, Martin, and Heinrich Braun. *Rprop—a fast adaptive learning algorithm*. Proc. of ISICIS VII), Universitat. 1992.
- [31] Rohde, Angelika, and Alexandre B. Tsybakov. *Estimation of high-dimensional low-rank matrices*. The Annals of Statistics 39.2 (2011): 887-930.
- [32] Ruchansky, Natali, Mark Crovella, and Evimaria Terzi. *Matrix completion with queries*. Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 2015.
- [33] Settles, Burr. *Active learning literature survey*. (2009).
- [34] Xie, Kun, et al. *Low cost and high accuracy data gathering in WSNs with matrix completion*. IEEE Transactions on Mobile Computing 17.7 (2017): 1595-1608.
- [35] Zeiler, Matthew D. *Adadelta: an adaptive learning rate method*. arXiv preprint arXiv:1212.5701 (2012).
- [36] Zhao, R., et al. *An Adaptive Curriculum Learning Framework for Unbiased Glaucoma Diagnosis*. Proceedings of the Computer Vision—ECCV (2020).
- [37] Zhang, Hui, et al. *Strongly convex programming for exact matrix completion and robust principal component analysis*. arXiv preprint arXiv:1112.3946 (2011).
- [38] Arnold, Matthew, et al. *A survey of adaptive optimization in virtual machines*. Proceedings of the IEEE 93.2 (2005): 449-466.