LOW-RANK MATRIX RECOVERY WITH UNKNOWN CORRESPONDENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

We study a matrix recovery problem with unknown correspondence: given the observation matrix $M_o = [A, \tilde{P}B]$, where $\tilde{P}$ is an unknown permutation matrix, we aim to recover the underlying matrix $M = [A, B]$. Such problem commonly arises in many applications where heterogeneous data are utilized and the correspondence among them are unknown, e.g., due to privacy concerns. We show that it is possible to recover M via solving a nuclear norm minimization problem under a proper low-rank condition on M, with provable non-asymptotic error bound for the recovery of M. We propose an algorithm, M^{3}O (Matrix recovery via Min-Max Optimization) which recasts this combinatorial problem as a continuous minimax optimization problem and solves it by proximal gradient with a Max-Oracle. M^{3}O can also be applied to a more general scenario where we have missing entries in M_o and multiple groups of data with distinct unknown correspondence. Experiments on simulated data, the MovieLens 100K dataset and Yale B database show that M^{3}O achieves state-of-the-art performance over several baselines and can recover the ground-truth correspondence with high accuracy.

1 INTRODUCTION

In the era of big data, one usually needs to utilize data gathered from multiple disparate platforms when accomplishing a specific task. However, the correspondence among the data samples from these different sources are often unknown due to either missing identity information or privacy reasons (Unnikrishnan et al., 2018; Gruteser et al., 2003; Das & Lee, 2018). Examples include the multi-image matching problem studied in (Ji et al., 2014; Zeng et al., 2012; Zhou et al., 2015), the record linkage problem (Chan & Loh, 2001) and the federated recommender system (Yang et al., 2020).

In the simplest scenario, we have two data matrices $A = [a_1, ..., a_n]^\top$, $B = [b_1, ..., b_n]^\top$ with $a_i \in \mathbb{R}^{m_A}$ and $b_i \in \mathbb{R}^{m_B}$, which are from two different platforms (data sources). As discussed above, the correspondence (a_i, b_i) may not be available, and thereby the goal is to recover the underlying correspondence between $a_1, ..., a_n$ and $b_{\tilde{\pi}(1)}, ..., b_{\tilde{\pi}(n)}$, where $\tilde{\pi}(\cdot)$ denotes an unknown permutation. We can translate such problem described above as a matrix recovery problem, i.e., to recover the matrix $M = [A, B]$ from the permuted observation $M_o = [A, \tilde{P}B]$, where $\tilde{P} \in \mathcal{P}_n$ is an unknown permutation matrix and $\mathcal{P}_n$ denotes the set of all $n \times n$ permutation matrices. We term this problem as **M**atrix **R**ecovery with **U**nknown **C**orrespondence (**MRUC**).

Inspired by the classical low-rank model for matrix recovery (Wright & Ma, 2021; Mazumder et al., 2010; Hastie et al., 2015), we especially focus on the scenario where the matrix M features a certain low-rank structure. Such low-rank model has achieved great success in many applications like the recommender system (Schafer et al., 2007; Mazumder et al., 2010) and the image recovery and alignment problem (Zeng et al., 2012; Zhou et al., 2015). By denoting $B_o = \tilde{P}B$, we want to solve the following rank minimization problem for MRUC,

$$\min_{P \in \mathcal{P}_n} \text{rank}([A, PB_o]). \tag{1}$$

Practical applications. It is known that the recommender system often suffers from data sparsity (Zhang et al., 2012) because users typically only provide ratings for very few items. To enlarge the set of observable ratings for each user, we may harness extra data from multiple platforms (Netflix, Amazon, Youtube, etc.). One classical work on this problem is the multi-domain recommender system considered in (Zhang et al., 2012). Unfortunately, their work neglects a crucial issue that

data from these diverse platforms (or domains) are not always well aligned for two primary reasons. The first is that the same user may use different identities, or even leave nothing about their identities, on these platforms. Another reason is that, those platforms are not allowed to share with each other the identity information about their users for preserving privacy. Another application is the visual permutation learning problem (Santa Cruz et al., 2017), where one needs to recover the original image from the *shuffled* pixels. Both of the two applications give rise to a challenging extension of the MRUC problem, where we not only need to recover multiple correspondence across different data sources, but also face the difficulty of dealing with the missing values in data matrix .

Relationship to the multivariate unlabeled sensing problem. Problem (1) is closely related to the **M**ultivariate **U**nlabeled **S**ensing (**MUS**) problem, which has been studied in (Pananjady et al., 2017; Zhang et al., 2019a;b; Zhang & Li, 2020; Slawski et al., 2020b;a). Specifically, the MUS is the multivariate linear *regression* problem with unknown correspondence, i.e., it solves

$$\min_{P\in\mathcal{P}_n, W\in\mathbb{R}^{m_2\times m_1}} \|Y - PXW\|_F^2, \tag{2}$$

where $W \in \mathbb{R}^{m_2\times m_1}$ is the regression coefficient matrix, $Y \in \mathbb{R}^{n\times m_1}$ and $X \in \mathbb{R}^{n\times m_2}$ denotes the output and the permuted input respectively, and $\|\cdot\|_F$ is the matrix Frobenius norm. In fact, a concurrent work (Yao et al., 2021) studies the same rank minimization problem as (1), but their approach is to solve it using the algorithm developed for MUS problem. Despite of the similarity to the MUS problem, we remark that MRUC problem has it own distinct features and, as shown in Section 4, the algorithm for the MUS algorithm can not be directly and effectively applied, especially when there are multiple unknown correspondence and missing entries to be considered.

Related works. To the best of our knowledge, the concurrent and independent (Yao et al., 2021) is the only work that also considers the MRUC problem. Theoretically, (Yao et al., 2021) showed that there exists an non-empty open subset $U \subseteq \mathbb{R}^{n\times(m_1+m_2)}$, such that $\forall M \in U$, solving (1) is bound to recover the original correspondence. However, such results only prove its existence for the subset U and do not provide a concrete characterization. Regarding the algorithm design, (Yao et al., 2021) follows the idea of (Slawski et al., 2020b;a) and treats problem (1) heuristically as a MUS problem. However, there are two main drawbacks in their algorithm that largely limit its practical value. First, in the dense permutation scenario, it ignores the interaction among the shuffled columns and hence can not utilize the prior knowledge on the unknown permutation to have an improved performance; Second, their method can not deal with data with missing values.

Contributions of this work. Our contributions in this work lie in both theoretical and practical aspects. Theoretically, we are the first to rigorously study how the rank of the data matrix is perturbed by the permutation, and show that problem (1) can be used to recover a generic low-rank random matrix almost surely. Besides, we also propose a nuclear norm minimization problem as a surrogate for problem (1). The most important theoretical result in this work is that we provide a non-asymptotic analysis to bound the error of the nuclear norm minimization problem under a mild assumption. Practically, we propose an efficient algorithm M^{3}O that solves the nuclear norm minimization problem, which overcomes the aforementioned two shortcomings in (Yao et al., 2021). Notably, M^{3}O works very well even for an extremely difficult task, where we need to recover multiple unknown correspondence from the data that are densely permuted and contain missing values. We remark that this is so far a challenging problem unexplored in the existing literature.

Outline. For conciseness, we will first study the MRUC problem with single unknown correspondence, and then show that the theoretical results and the algorithm can be readily extended to the more complicated scenarios. We start with building the theoretical results for (1) and its convex relaxation in Section 2. Then, the algorithm is developed in Section 3. The simulation results are presented in Section 4 and the conclusions are drawn in Section 5.

Notations. Given two matrices $X, Y \in \mathbb{R}^{n\times m}$, we denote $\langle X, Y\rangle = \sum_{i=1}^n \sum_{j=1}^m X_{ij}Y_{ij}$ as the matrix inner product. We denote $X(i)$ as the ith row of the matrix X and $X(i,j)$ as the element at the ith row and the jth column. We denote $\mathbf{1}_m \in \mathbb{R}^m$ and $\mathbf{1}_{n\times m} \in \mathbb{R}^{n\times m}$ as the all-one vector and matrix, respectively, and I_n be the $n \times n$ identity matrix. For $\alpha \in \mathbb{R}^m$, $\beta \in \mathbb{R}^n$, we define the operator $\oplus$ as $\alpha \oplus \beta = \alpha\mathbf{1}_n^\top + \mathbf{1}_m\beta^\top \in \mathbb{R}^{m\times n}$. We denote $\|\cdot\|_*$ as the nuclear norm for matrices. For vectors, we denote $\|\cdot\|_0$, $\|\cdot\|_1$ as the zero norm and 1-norm respectively.

2 Matrix Recovery via a Low-rank Model

How is the matrix rank perturbed by the row permutation? To answer this fundamental question, we first introduce the cycle decomposition of a permutation.

Definition 1 (Cycle decomposition of a permutation (Dummit & Foote, 1991)). *Let S be a finite set and $\pi(\cdot)$ be a permutation on S. A cycle $(a_1, ..., a_n)$ is a permutation sending a_j to a_{j+1} for $1 \leq j \leq n-1$ and a_n to a_1. Then a cycle decomposition of $\pi(\cdot)$ is an expression of $\pi(\cdot)$ as a union of several disjoint cycles*[1].

It can be verified that any permutation on a finite set has a unique cycle decomposition (Dummit & Foote, 1991). Therefore, we can define the *cycle number* of a permutation $\pi(\cdot)$ as the number of disjoint cycles with length greater than 1, which is denoted as $C(\pi)$. We also define the non-sparsity of a permutation as the Hamming distance between it and the original sequence, i.e., $H(\pi) = \sum_{s \in S} \mathbb{I}[\pi(s) \neq s]$. It is obvious that $H(\pi) > C(\pi)$ if π is not an identity permutation. As a simple example, we consider the permutation $\pi(\cdot)$ that maps the sequence (1,2,3,4,5,6) to (3,1,2,5,4,6). Now the cycle decomposition for it is $\pi(\cdot) = (132)(45)(6)$, and $C(\pi) = 2$, $H(\pi) = 5$.

In all the following theoretical results, we denote the original matrix as $M = [A, B] \in \mathbb{R}^{n \times m}$ with $A \in \mathbb{R}^{n \times m_A}$, $B \in \mathbb{R}^{n \times m_B}$, and $\text{rank}(M) = r$, $\text{rank}(A) = r_A$, $\text{rank}(B) = r_B$. We denote the corresponding permutation as $\pi_P(\cdot)$ for any permutation matrix $P \in \mathcal{P}_n$. The following proposition says that the perturbation effect of a permutation π on the rank of M becomes stronger, if π permutes more rows and contains less cycles.

Proposition 1. *$\forall P \in \mathcal{P}_n$, we have*

$$\text{rank}([A, PB]) \leq \min\{n, m, r_A + r_B, r + H(\pi_P) - C(\pi_P)\}. \tag{3}$$

We have similar result for the case with multiple permutations, which is summarized in Corollary 1 in Appendix A.1. It turns out that, without any assumption on M, (3) is the tightest upper bound for the rank of a perturbed matrix. Notably, the following proposition says that the upper bound in (3) is attained with probability 1 for a generic low-rank random matrix.

Definition 2. *A probability distribution on $\mathbb{R}$ is called a proper distribution if its density function $p(\cdot)$ is absolutely continuous with respect the Lebesgue measure on $\mathbb{R}$.*

Proposition 2. *If the original matrix M is a random matrix with $M = RE$ where $R \in \mathbb{R}^{n \times r}$ and $E \in \mathbb{R}^{r \times m}$ are two random matrices whose entries are i.i.d and follow a proper distribution on $\mathbb{R}$, and $r \leq \min\{\sqrt{\frac{n}{2}}, m_A, m_B\}$, then $\forall P \in \mathcal{P}_n$, the equality*

$$\text{rank}([A, PB]) = \min\{2r, r + H(\pi_P) - C(\pi_P)\} \tag{4}$$

holds with probability 1.

Convex relaxation for the rank function. Despite the previous theoretical justification for problem (1), it is non-convex and non-smooth. Another crucial issue is that we often have a noisy observation matrix and it is well known that the rank function is extremely sensitive to the additive noise. In this paper, we assume that the observation matrix is corrupted by i.i.d Gaussian additive noise, i.e.,

$$M_o = [A_o, B_o] = [A, \tilde{P}B] + W, \text{ where } W(i, j) \sim \mathcal{N}(0, \sigma^2),$$

where σ^2 reflects the strength of the noise. We first denote the singular values of a matrix $X \in \mathbb{R}^{n \times m}$ as $\sigma_X^1, ..., \sigma_X^k$ where $k = \min\{n, m\}$. Since $\text{rank}(X) = \|[\sigma_X^1, ..., \sigma_X^k]\|_0$, from Proposition 2 we can view the perturbation effect of a permutation to a low-rank matrix as breaking the sparsity of its singular values. This view leads naturally to the well-known 1-norm minimization problem which has been proven robust to additive noise and can yield a sparse solution (Wright & Ma, 2021), i.e.,

$$\min_{P \in \mathcal{P}_n} \|[A_o, PB_o]\|_* = \|[\sigma_{M_o}^1, ..., \sigma_{M_o}^k]\|_1. \tag{5}$$

Since for an arbitrary matrix, the 1-norm of its singular values is equivalent to its nuclear norm, we refer problem (5) as the nuclear norm minimization problem.

Theoretical justification for the nuclear norm. Nuclear norm has a long history used as a convex surrogate for the rank, and it has been theoretically justified for applications like low-rank matrix completion (Candès & Tao, 2010; Wright & Ma, 2021). It is also important to see whether the nuclear norm is still a good surrogate for the rank minimization problem (1). In this work, we establish a sufficient condition on A and B under which problem (5) is provably justified for correspondence recovery. We denote $A = \sum_{i=1}^{r_A} \sigma_A^i u_A^i v_A^{i\top}$, $B = \sum_{i=1}^{r_B} \sigma_B^i u_B^i v_B^{i\top}$ as the singular values decomposition of A and B, where the σ_A^i and σ_B^i are the non-zero singular values.

Firstly, from the definition of nuclear norm, it can be simply verified for any $P \in \mathcal{P}_n$ that

$$-Z/N \leq (\|[A, PB]\|_* - \|M\|_*)/\|M\|_* \leq Z/N, \tag{6}$$

where we denote $N = \max\{\|A\|_*, \|B\|_*\}$ and $Z = \min\{\|A\|_*, \|B\|_*\}$. The inequality (6) indicates that A and B should have comparable magnitude, i.e., $\|A\|_* \approx \|B\|_*$, otherwise the influence of the permutation will be less significant. Therefore, we are interested in the scenario where the singular values of A and B are comparable, which is described as the following Assumption 1.

[1]Two cycles are disjoint if they do not have common elements

Assumption 1. *There exists a constant* $\epsilon_1 \geq 0$ *such that*

$$|\sigma_A^i - \sigma_B^i| \leq \epsilon_1, \ \forall i = 1, .., r, \tag{7}$$

where we denote that $\sigma_A^i = 0$ *if* $i > r_A$*, and* $\sigma_B^i = 0$ *if* $i > r_B$*.*

Similar to the matrix rank, we also need a proper low-rank assumption on the matrix M for the nuclear norm. In this work, we particularly study the scenario that the left singular vectors of A and B are similar, which we formally describe as Assumption 2. We refer Assumption 2 as a proper low-rank assumption, because it indicates that the column space of M can be approximated by the column space of one of its submatrices.

Assumption 2. *There exists a constant* $\epsilon_2 \geq 0$ *such that*

$$\|u_A^i - u_B^i\| \leq \epsilon_2, \forall i = 1, ..., T, \tag{8}$$

where we denote $T = \min\{r_A, r_B\}$*.*

Furthermore, we also need that all the column singular vectors $u_A^1, ..., u_A^T, u_B^1, ..., u_B^T$ are variant under any $P \in \mathcal{P}_n$ with $P \neq I_n$: we define a vector $u \in \mathbb{R}^n$ to be variant under a $P \in \mathcal{P}_n$ if $Pu \neq u$. One simple and weak condition for a vector u to satisfy such property is that u dose not contains duplicated elements, which leads to the following Assumption 3.

Assumption 3. *There exists a constant* $\epsilon_3 \geq 0$ *such that*

$$\min_{u \in U} \min_{i \neq j} |u(i) - u(j)| \geq \epsilon_3 > 0, \tag{9}$$

where $U = \{u_A^1, ..., u_A^T, u_B^1, ..., u_B^T\}$*.*

In summary, the assumptions mentioned above feature a typical low-rank structure in M, and implies that the nuclear norm of M is sensitive to permutation. With the three assumptions, we have the following important theorem, which provides high probability bound for the approximation error of (5).

We denote the solution to (5) as P^*, and let π^* and $\tilde{\pi}$ be the corresponding permutation to the permutation matrices $P^{*\top}$ and $\tilde{P}$, respectively. We define the difference between the two permutations π^* and $\tilde{\pi}$ as the *Hamming* distance

$$d_H(\pi^*, \tilde{\pi}) \stackrel{\text{def.}}{=} \sum_{i=1}^{n} \mathbb{I}(\pi^*(i) \neq \tilde{\pi}(i)).$$

Theorem 1. *Under Assumptions 1, 2 and 3, if additionally* $\epsilon_1 \leq \frac{M}{4r}$*,* $\epsilon_2 \leq \min\{\frac{1}{2\sqrt{2T}}, \frac{\sqrt{2}M}{2N}\}$*, and* $\sigma \leq \frac{M}{16L^2}$*, then the following bound for the Hamming distance*

$$d_H(\pi^*, \tilde{\pi}) \leq \frac{2}{\epsilon_3^2}\left(2 - \left(\frac{\sqrt{2}D}{D + (\sqrt{2}+2)\epsilon_1 r + \sqrt{2}\epsilon_2 N + 2\sqrt{2DL\sigma}} - \sqrt{T}\epsilon_2\right)^2\right) \tag{10}$$

holds with probability at least $1 - 2\exp\{-\frac{D}{8L\sigma}\}$*, where* $L = \max\{n, m\}$*,* $D = \|A\|_* + \|B\|_*$*.*

The proof to all the aforementioned theoretical results are provided in Appendix A.1.

Remark 1. From Theorem 1 we can see that when $\epsilon_3 > 0$, and $\epsilon_1 \to 0$, $\epsilon_2 \to 0$, $\sigma \to 0$, the error $d_H(\pi^*, \tilde{\pi})$ will converge to zero with probability 1. Furthermore, we can also discover that the correspondence can be difficult to recover when:

- The rank of original matrix M is high, which can be seen from (10).
- The magnitude of A and B w.r.t rank or nuclear norm are not comparable, which can be seen from (6) and (7).
- The strength of noise is high, which can be seen from (10) and the probability in Theorem 1.

Notably, the numerical experiments in Section 4.1 corroborate our claims as well.

Remark 2. Additionally, from the proof of Theorem 1 we find that the fundamental reason for the success of (5) is that if M satisfies the previous assumptions, we have

$$\|[A, PB]\|_*/\|M\|_* \approx O\left((1 - H(\pi_P)/2n)^{-\frac{1}{2}}\right). \tag{11}$$

In many applications, we can only observe part of the full data. Therefore, it is also worthwhile to investigate whether (11) still holds when we can only access a small subset of the entries in M_o. Notably, Figure 1 gives the positive answer and shows that the relationship (11) is gracefully degraded when the percentage of observable entries is decreasing. This phenomenon is remarkable since it indicates the original correspondence can be recovered from only part of the full data. The matrices used to generate Figure 1 are the same as those in Section 4.1, and the nuclear norm is computed approximately by first filling the missing entries using Soft-Impute algorithm (Mazumder et al., 2010).

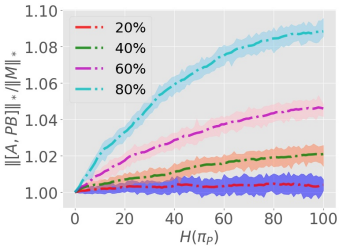


Figure 1: The relationship (11) under different percentages of observable entries.

3 ALGORITHM

In this section, we consider the scenario with missing values, i.e., our observed data is $\mathcal{P}_\Omega(M_o) = \mathcal{P}_\Omega([A_o, B_o])$, where $\mathcal{P}_\Omega$ is an operator that selects entries that are in the set of observable indices Ω. In this scenario, problem (5) can not be directly used since the evaluation of the nuclear norm and optimization of the permutation are coupled together. Inspired by the matrix completion method (Hastie et al., 2015; Mazumder et al., 2010), we propose to solve an alternative form of (5) as follows,

$$\min_{\widehat{M}\in\mathbb{R}^{n\times m}} \min_{P\in\mathcal{P}_n} \left\|\mathcal{P}_\Omega([A_o, PB_o]) - \mathcal{P}_\Omega(\widehat{M})\right\|_F^2 + \lambda\left\|\widehat{M}\right\|_*, \tag{12}$$

where $\lambda > 0$ is the penalty coefficient. We denote that $\widehat{M} = [\widehat{M}_A, \widehat{M}_B]$ and $\widehat{M}_A, \widehat{M}_B$ are the two submatrices with the same dimension as A_o and B_o respectively. We can write (12) equivalently as

$$\min_{\widehat{M}\in\mathbb{R}^{n\times m}} \min_{P\in\mathcal{P}_n} \left\|\mathcal{P}_\Omega(A_o) - \mathcal{P}_\Omega(\widehat{M}_A)\right\|_F^2 + \langle C(\widehat{M}_B), P\rangle + \lambda\left\|\widehat{M}\right\|_*, \tag{13}$$

where $C(\widehat{M}_B) \in \mathbb{R}^{n\times n}$ is the pairing cost matrix with

$$C(\widehat{M}_B)(i,j) = \sum_{(j,j'')\in\Omega}\left(\widehat{M}_B(i,j'') - B_o(j,j'')\right)^2,\ \forall i,j = 1,...,n.$$

Baseline algorithm. A conventional strategy to handle an optimization problem like (13) is the alternating minimization or the block coordinate descent algorithm (Abid et al., 2017). Specifically, it executes the following two updates iteratively until it converges.

$$\widehat{M}^{\text{new}} \leftarrow \operatorname*{arg\,min}_{\widehat{M}\in\mathbb{R}^{n\times m}} \left\|\mathcal{P}_\Omega([A_o, \widehat{P}^{\text{old}}B_o]) - \mathcal{P}_\Omega(\widehat{M})\right\|_F^2 + \lambda\left\|\widehat{M}\right\|_*, \tag{14}$$

$$\widehat{P}^{\text{new}} \leftarrow \operatorname*{arg\,min}_{P\in\mathcal{P}_n} \langle C(\widehat{M}_B^{\text{new}}), P\rangle. \tag{15}$$

The first update step (14) is a convex optimization problem and can be solved by the proximal gradient algorithm (Mazumder et al., 2010). The second update step (15) is actually a discrete optimal transport problem which can be solved by the classical Hungarian algorithm with time complexity $O(n^3)$ (Jonker & Volgenant, 1986). However, as we will see in the Section 4, this algorithm performs poorly, and it is likely to fall into an undesirable local solution quickly in practice. Specifically, the main reason is that the solution of (15) is often not unique and a small change in $\widehat{M}_B$ would lead to large change of $\widehat{P}$. To address this issue, we propose a novel and efficient algorithm M^{3}O algorithm based on the entropic optimal transport (Peyré et al., 2019) and min-max optimization (Jin et al., 2020).

Smoothing the permutation with entropy regularization. For any $a \in \mathbb{R}^n, b \in \mathbb{R}^m$, we define

$$\Pi(a,b) = \{S \in \mathbb{R}^{n\times m} : S\mathbf{1}_m = a, S^\top\mathbf{1}_n = b, S(i,j) \geq 0,\ \forall i,j\},$$

which is also known as the Birkhoff polytope. The famous Birkhoff-von Neumann theorem (Birkhoff, 1946) states that the set of extremal points of $\Pi(\mathbf{1}_n, \mathbf{1}_n)$ is equal to $\mathcal{P}_n$. Inspired by (Xie et al., 2021) and the interior point method for linear programming (Bertsekas, 1997), in order to smooth the optimization process of the baseline algorithm, we relax P from being an exact permutation matrix, i.e., to keep P staying inside the Birkhoff polytope $\Pi(\mathbf{1}_n, \mathbf{1}_n)$. That is, we propose to replace the combinatorial problem (15) with the following continuous optimization problem

$$\min_{P\in\Pi(\mathbf{1}_n,\mathbf{1}_n)} \langle C(\widehat{M}_B), P\rangle + \epsilon H(P), \tag{16}$$

where $H(P) \overset{\text{def.}}{=} \sum_{i,j} P(i,j)(\log(P(i,j)) - 1)$ is the matrix negative entropy and $\epsilon > 0$ is the regularization coefficient. Notably, (16) is also known as the Entropic Optimal Transport (EOT) problem (Peyré et al., 2019), which is a strongly convex optimization problem and can be solved roughly in the $O(n^2)$ complexity per iteration by the Sinkhorn algorithm. Specifically, the Sinkhorn algorithm solves the dual problem of (16),

$$\max_{\alpha,\beta\in\mathbb{R}^n} W_\epsilon(\widehat{M}_B,\alpha,\beta) \overset{\text{def.}}{=} \langle \mathbf{1}_n,\alpha\rangle + \langle \mathbf{1}_n,\beta\rangle - \epsilon\left\langle \mathbf{1}_{n\times n}, \exp\left\{\frac{\alpha\oplus\beta - C(\widehat{M}_B)}{\epsilon}\right\}\right\rangle, \tag{17}$$

which reduces the variables dimension from n^2 to $2n$ and is thus greatly favorable in the high dimension scenario. By substituting the inner minimization problem of (13) with (16), we end up with solving the following unconstrained min-max optimization problem

$$\min_{\widehat{M}}\max_{\alpha,\beta} \left\|A - \widehat{M}_A\right\|_F^2 + W_\epsilon(\widehat{M}_B,\alpha,\beta) + \lambda\left\|\widehat{M}\right\|_*. \tag{18}$$

Follows the idea of (Jin et al., 2020), we consider to adopt a proximal gradient algorithm with a Max-Oracle for (18). Specifically, we employ the Skinhorn algorithm (Peyré et al., 2019) as the Max-Oracle to retrieve an ε-good solution of the inner max problem (17). We summarize our proposed algorithm M^{3}O (**M**atrix recovery via **M**in-**M**ax **O**ptimization) in Algorithm 1, where $\text{prox}_{\lambda\|\cdot\|_*}(\cdot)$ is the proximal operator of nuclear norm and ρ_k is the gradient stepsize. The convergence property of M^{3}O can be obtained by following (Jin et al., 2020), which shows that, with a decaying stepsize, M^{3}O is bound to converge to an ε-good Nash equilibrium within $O(\varepsilon^{-2})$ iterations.

Algorithm 1: M^{3}O

while not converged **do**

  For the tolerance ε, run the Sinkhorn algorithm to find α^*, β^* such that

$$W_\epsilon(\widehat{M}_B^k,\alpha^*,\beta^*) > \max_{\alpha,\beta} W_\epsilon(\widehat{M}_B^k,\alpha,\beta) - \varepsilon;$$

  Perform $\widehat{M}^{\text{k+1}} \leftarrow \text{prox}_{\lambda\|\cdot\|_*}(\widehat{M}^k - \rho_k\nabla_{\widehat{M}}F_\epsilon(\widehat{M}^k,\alpha^*,\beta^*))$, where

$$F_\epsilon(\widehat{M},\alpha,\beta) \overset{\text{def.}}{=} \left\|A - \widehat{M}_A\right\|_F^2 + W_\epsilon(\widehat{M}_B,\alpha,\beta);$$

end

Remark 3. A recent work (Xie et al., 2020) proposes a decaying strategy for the entropy regularization coefficient ϵ in (16) so that the optimal solutions of (15) and (16) do not deviate too much. Inspired by it, in our practice, we take large ϵ in the beginning and gradually shrink it by half until the objective function stops improving for K steps.

Remark 4. A useful trick is that we should not take large stepsize ρ_k in the early iterations because the permutation matrix could still be far away from the optimal one. However, a small stepsize would lead to slow convergence. Heuristically, we propose an adaptive stepsize strategy that performs well in practice. For the solution of (16) $\widehat{P}_k$ at the kth iteration, we compute the two statistics

$$\delta_k = \left\|\widehat{P}_{k-1} - \widehat{P}_k\right\|_F^2/2n \text{ and } c_k = \left\|\max_j \widehat{P}_k(\cdot,j) - \mathbf{1}_n\right\|_1/n.$$

Here δ_k represents how fast the permutation matrix $\widehat{P}_k$ changes over the iterations, while c_k measures how far the current $\widehat{P}_k$ is close to an exact permutation matrix. Both δ_k and c_k reflect the confidence on the current found correspondence. Based on them, we set the stepsize as $\rho_{k+1} = (1-\delta_k)(1-c_k)^\omega$, where $\omega > 0$ is a tunable parameter which is often set to a value between 0.5 to 3. ω actually trades off the convergence speed and final performance. The smaller the ω, the faster the convergence. Therefore, a practical way is to start with a small ω, and gradually increase it until the final performance stops improving.

Remark 5. As discussed in Section 1, in many cases we have to deal with the problem that involves multiple correspondence, i.e., we need to recover the matrix $M = [A, B_1, ..., B_d]$ from the observation data $\mathcal{P}_\Omega(M_o)$, where

$$M_o = [A_o, B_o^1, ..., B_o^d] = [A, \tilde{P}_1B_1, ..., \tilde{P}_dB_d] + W,$$

where $\tilde{P}_l \in \mathcal{P}_n$ and W is a noise matrix. We refer such problem as the ***d*-correspondence** problem. An important observation is that, although the number of possible correspondence increase exponentially as d grows, the complexity of M^{3}O per iteration only linearly increases with d and can be implemented in a fully parallel fashion. Specifically, in this scenario, we solve the problem

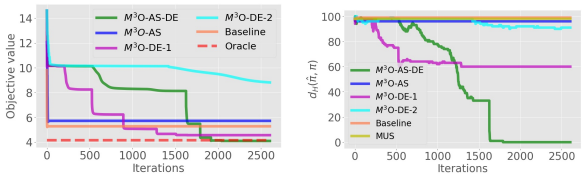
$$\min_{\widehat{M}} \min_{P_1,\ldots,P_d} \left\| \mathcal{P}_\Omega(A_o) - \mathcal{P}_\Omega(\widehat{M}_A) \right\|_F^2 + \sum_{l=1}^{d} \left\{ \langle C(\widehat{M}_{B_l}), P_l \rangle + \epsilon H(P_l) \right\} + \lambda \left\| \widehat{M} \right\|_*, \quad (19)$$
$$\text{s.t. } P_l \in \Pi(\mathbf{1}_n, \mathbf{1}_n),\ l = 1, ..., d,$$

where we denote $\widehat{M} = [\widehat{M}_A, \widehat{M}_{B_1}, ..., \widehat{M}_{B_d}]$. Here $\widehat{M}_A$ and $\widehat{M}_{B_l}$ have the same dimension with A_o and B_o^l, respectively. One can find that the inner problems for solving P_l are actually decoupled for each l, which guarantees an efficient parallel implementation.

Remark 6. Since x problem (12) has a similar form to that considered in (Mazumder et al., 2010). We adopt the same tuning strategy of λ as in (Mazumder et al., 2010), which suggests that we should start with large λ and gradually decrease it.
We relegate more details about M^{3}O to Appendix A.6.

4 Experiments

In this section, we evaluate our proposed M^{3}O on both synthetic and real-world datasets, including the MovieLens 100K and the Extended Yale B dataset. We also provide an ablation study for the decaying entropy regularization strategy and the adaptive stepsize strategy proposed in Remarks 3 and 4. In all the experiments, we employ the Soft-Impute algorithm (Mazumder et al., 2010) as a standard algorithm for matrix completion. Extra experiment details and auxiliary results can be found in Appendix A.9.



(a) Objective value (b) Permutation error

Figure 2: Performance of various algorithms on a simulated 1-correspondence problem.

Algorithms. We denote the following algorithms for comparison in all the experiments:

1. *Oracle*: Running the Soft-Impute algorithm with ground-truth correspondence.
2. *Baseline*: The Baseline algorithm in (14) and (15).
3. *MUS*: Since there is currently no existing algorithm directly applicable to the scenario considered by (19), inspired by (Yao et al., 2021), we modify and extend the algorithm in (Zhang & Li, 2020), which is originally proposed for the MUS problem, to deal with the MRUC problem. The details of the adapted algorithm are provided in Appendix A.8.

4.1 Synthetic Data

We first investigate the property of our proposed M^{3}O algorithm on the synthetic data.

Data generation. We generate the original data matrix in this form $M = RE + \eta W$, where $R \in \mathbb{R}^{n\times r}$, $E \in \mathbb{R}^{r\times m}$, $W \in \mathbb{R}^{n\times m}$ and $\eta > 0$ indicates the strength of the additive noise. The entries of R, E, W are all i.i.d sampled from the $\mathcal{N}(0, 1)$. Then we split the data matrix M by $M = [A, B_1, ..., B_d]$ where we denote $A \in \mathbb{R}^{n\times m_A}$, $B_1 \in \mathbb{R}^{n\times m_1}$, ..., $B_d \in \mathbb{R}^{n\times m_d}$ to represent data from $d + 1$ data sources. The permuted observation matrix M_o is obtained by first generating d permutation matrices $P_1, ..., P_d$ randomly and independently, and then computing $M_o = [A, P_1B_1, ..., P_dB_d]$. Finally, we remove $(1 - |\Omega| \cdot 100\%/(n \cdot m))$ percent of the entries of M_o randomly and uniformly, where $|\Omega|$ indicating the number of observable entries.

Ablation study. We denote the following variants of M^{3}O for the ablation study.

1. *M^{3}O-AS-DE*: M^{3}O with both Adpative Stepsize and Decaying Entropy regularization.
2. *M^{3}O-DE*: M^{3}O with Decaying Entropy regularization only. M^{3}O-DE-1 and M^{3}O-DE-2 adopt constant stepsize $\rho_k = 0.5$ and $\rho_k = 0.01$, respectively.
3. *M^{3}O-AS*: M^{3}O with Adpative Stepsize only. The entropy coefficient ϵ is fixed to 0.0005.

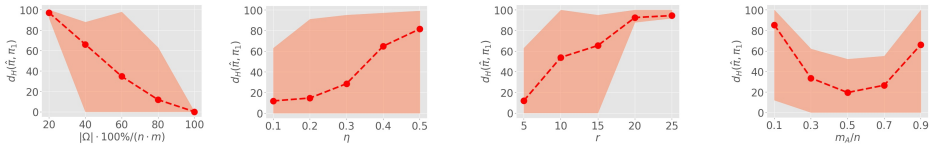
In the following results, we denote π_l as the corresponding permutation to P_l. We initialize $\widehat{M}$ from Gaussian distribution for the M^{3}O algorithm and its variants. We choose initial ϵ as 0.1 and $K = 100$ as the default for the decaying entropy regularization, and set $\omega = 3$ as the default for the adaptive stepsize. We also report the achieved objective values of (19) for the tested algorithms, except for the MUS algorithm since it has a different objective. We denote $\hat{\pi}$ as the recovered permutation.

Results. Figure 2 displays the result under the setting $\eta = 0.1$, $|\Omega| \cdot 100\%/(n \cdot m) = 80\%$, $n = m = 100$, $r = 5$, $d = 1$, $m_A = 60$ and $m_1 = 40$. The algorithm M^{3}O-AS-DE achieves the best result, and can recover the ground-truth correspondence. M^{3}O-AS behaves similarly to Baseline

and MUS. They all converge to a poor local solution quickly. M^{3}O-DE-1 converges quickly and also falls into a poor local solution due to large stepsize, while M^{3}O-DE-2 adopts a small stepsize and hence suffers from slow convergence. Due to the superiority of M^{3}O-AS-DE over the other variants, in the following results, we refer M^{3}O as M^{3}O-AS-DE for short.

Figure 3 examine M^{3}O on a 1-correspondence problem under different regimes w.r.t $|\Omega|$, η, r and m_A/n. Here we use m_A/n to control the difference of the magnitude of the submatrices. As we can see, the results are well aligned with our prediction in Remarks 1 and 2.

Finally, we examine M^{3}O on a few d-correspondence problems. See Table 1 for various results, where we set $r = 5$ and $\varepsilon = 0.1$. Notice that for the 4-correspondence problem in the table, there are $(100!)^4$ possible correspondence. Even for such a difficult problem, M^{3}O is able to recover 61.5% of the ground-truth correspondence with a good initialization.



(a) d_H v.s. $|\Omega|$ (b) d_H v.s. η (c) d_H v.s. r (d) d_H v.s. m_A/n

Figure 3: Performance of M^{3}O on a 1-correspondence problem under different levels of $|\Omega|$, η, r and m_A/n. The default setting is $|\Omega| \cdot 100\%/(n \cdot m) = 80\%$, $\eta = 0.1$, $n = m = 100$, $r = 5$, $m_A = 60$, and $m_1 = 40$. The mean with minimum and maximum are calculated from 10 different random initializations.

Table 1: Performance of M^{3}O for various d-correspondence problems. The normalized permutation error $\sum_{l=1}^{d} d_H(\hat{\pi}_l, \pi_l)/d$ is reported as mean±std (min) over 10 different random initializations.

$(n, m_A, m_1, ..., m_d)$	d	$\lvert\Omega\rvert \cdot 100\%/(n \cdot m)$	$\sum_{l=1}^{d} d_H(\hat{\pi}_l, \pi_l)/d$
(100,40,30,30)	2	40%	33.35 ± 32.85 (0.00)
(100,20,40,40)	2	40%	58.90 ± 27.21 (2.00)
(100,45,25,25,25)	3	50%	61.97 ± 15.41 (37.33)
(100,40,25,25,25,25)	4	60%	59.90 ± 13.64 (38.50)

4.2 Multi-domain recommender system without correspondence

In this section, we study the performance of M^{3}O on a real world dataset MovieLens 100K[2], which is a widely used movie recommendation dataset (Harper & Konstan, 2015). In this application, we mainly focus on the metric Root Mean Squared Error (RMSE), i.e.,

$$\text{RMSE} \overset{\text{def.}}{=} \sqrt{\frac{1}{N}\sum_{i,j}(\widehat{M}_{ij} - M_{ij})^2}.$$

Data. MovieLens 100K contains 100,000 ratings within the scale 1-5. The ratings are given by 943 users on 1,682 movies. Genre information about movies is also provided. We adopt a similar setting with (Zhang et al., 2012). We extract five most popular genres, which are Comedy, Romance, Drama, Action, Thriller respectively, to define the data from 5 different domains (or platforms). In addition to (Zhang et al., 2012), we randomly permute the indexes of the users from these five domains respectively, so that the correspondence among these data become unknown. In this way, the problem belongs to the 4-correspondence problem as discussed before. The ratings are split randomly, with 80% of them as the training data and the other 20% of them as the test data.

Algorithms. We consider the following additional algorithms for comparison.

1. *SIC*: Running the Soft-Impute algorithm independently for the 5 different platforms.
2. *SIR*: Running the Soft-Impute algorithm with Randomly generated correspondence.

Results. As discussed in experiments on the simulated data, the exact recovery of correspondence becomes impossible due to the small amount of observable entries. Therefore, in the following experiment, since exact correspondence is not needed, we fix $\epsilon = 0.05$ for M^{3}O. Table 2 shows the results by averaging the RMSE on the test data over 10 different random seeds.
We can first see that the matrix completion with a wrong correspondence, i.e., SIR, can be harmful to the overall performance since it is even worse than the results of SIC. Notably, although the ground-truth correspondence can not be recovered, each platform can still benefit from M^{3}O since it

[2]https://grouplens.org/datasets/movielens/100k/

improves the performance over SIC. This is mainly because M^{3}O is still able to correspond similar users for inferring missing ratings. On the contrary, since both Baseline and MUS can only establish an exact one-to-one correspondence for each user, they fail to improve SIC significantly. Remarkably, M^{3}O is only inferior to the Oracle method a little, and even achieves lower test RMSE than the Oracle method on the Comedy genre.

Table 2: Test RMSE of various algorithms on MovieLens 100K

Method	Comedy	Romance	Drama	Action	Thriller	Total
SIR	1.0202	1.0158	0.9808	0.9803	0.9811	0.9944
SIC	0.9694	0.9695	0.9317	0.9175	0.9253	0.9418
MUS	0.9659	0.9842	0.9423	0.9305	0.9306	0.9485
Baseline	0.9728	0.9562	0.9379	0.9105	0.9145	0.9395
M^{3}O	**0.9389**	**0.8787**	**0.9139**	**0.8556**	**0.8567**	**0.8948**
Oracle	0.9444	0.7825	0.9058	0.8176	0.8098	0.8667

4.3 Visual permutation recovery

We show that M^{3}O is flexible and can also be used to recover matrix that is not in the form $[A, PB]$. We can see this from the problem formulation in (13), where the cost matrix $C(\cdot)$ can be constructed in other ways as long as it is a function of a permutation. Typically, M^{3}O can be used to solve a challenging face image recovery problem. The original face image with size 180×180 in Figure 4(a) comes from the Extend Yale B database (Georghiades et al., 2001). The corrupted image is visualized in Figure 4(b), where the pixel blocks with size 30×30 in the upper left are shuffled randomly, and 30% of the total pixels are removed. This kind of problem is recently considered in (Santa Cruz et al., 2017), which proposes to recover the corrupted image in a data-driven way using convolutional neural networks. However, we show that it is possible to recover the image without additional data by merely exploiting the underlying low-rank structure of the image itself.

(a) Original (b) Corrupted

(c) Baseline (d) M^{3}O

Figure 4: Performance of M^{3}O on a face recovery problem.

This experiment setting is similar to that in (Yao et al., 2021) but the algorithm in (Yao et al., 2021) can not be applied since it can not work with the missing values. The MUS algorithm is also not applicable since this problem can not be written in the form of linear regression problem. From Figure 4(c) and 4(d) we can find that M^{3}O performs better than the Baseline, and can even recover the original orders of pixel blocks. More results similar to the Figure 4 and experiment details are provided in Appendix A.9.

5 Conclusion

In this paper, we have studied the important MRUC problem where part of the observed submatrix is shuffled. This problem has not been well explored in the existing literature. Theoretically, we are the first to rigorously analyze the role of low-rank model in the MRUC problem, and is also the first to show that minimizing nuclear norm is provably efficient for recovering a typical low-rank matrix. For practical implementations, we propose a highly efficient algorithm, the M^{3}O algorithm, which is shown to consistently achieve the best performance over several baselines in all the tested scenarios.

It is worthwhile to point out that apart from the two applications we have studied in this paper, this problem could arise in more scenarios like the gnome assembly problem (Huang & Madan, 1999), the video pose tracking problem (Ganapathi et al., 2012) and the privacy-aware sensor networks (Gruteser et al., 2003), etc. We believes that our work provides a general framework to deal with unknown correspondence issue in these scenarios.

As we have shown in Figure 3, one major limit of our algorithm is the sensitivity to the initialization. The phenomenon is exacerbated when the additive noise is high or the numbers of observable entries are small. We suggest to try with a few different initialization strategy when applying M^{3}O to a specific task. Finding stable initialization strategy is also an important task for our future works.

REFERENCES

Abubakar Abid, Ada Poon, and James Zou. Linear regression with shuffled labels. *arXiv preprint arXiv:1705.01342*, 2017.

Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334, 1997.

Garrett Birkhoff. Three observations on linear algebra. *Univ. Nac. Tacuman, Rev. Ser. A*, 5:147–151, 1946.

Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Hock-Peng Chan and Wei-Liem Loh. A file linkage problem of degroot and goel revisited. *Statistica Sinica*, pp. 1031–1045, 2001.

Debasmit Das and C. S. George Lee. Sample-to-Sample Correspondence for Unsupervised Domain Adaptation. *Engineering Applications of Artificial Intelligence*, 73:80–91, August 2018. ISSN 09521976. doi: 10.1016/j.engappai.2018.05.001. URL `http://arxiv.org/abs/1805.00355`. arXiv: 1805.00355.

Herbert A David and Haikady N Nagaraja. *Order statistics*. John Wiley & Sons, 2004.

David S Dummit and Richard M Foote. *Abstract algebra*, volume 1999. Prentice Hall Englewood Cliffs, NJ, 1991.

Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. Real-time human pose tracking from range data. In *European conference on computer vision*, pp. 738–751. Springer, 2012.

A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

Marco Gruteser, Graham Schelle, Ashish Jain, Richard Han, and Dirk Grunwald. Privacy-aware location sensor networks. In *HotOS*, volume 3, pp. 163–168, 2003.

Paul R Halmos. *Measure theory*, volume 18. Springer, 2013.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

Trevor Hastie, Rahul Mazumder, Jason D. Lee, and Reza Zadeh. Matrix completion and low-rank SVD via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015. Publisher: JMLR. org.

Xiaoqiu Huang and Anup Madan. Cap3: A dna sequence assembly program. *Genome research*, 9 (9):868–877, 1999.

Pan Ji, Hongdong Li, Mathieu Salzmann, and Yuchao Dai. Robust motion segmentation with unknown correspondence. In *European conference on computer vision*, pp. 204–219. Springer, 2014.

Chi Jin, Praneeth Netrapalli, and Michael Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *International Conference on Machine Learning*, pp. 4880–4889. PMLR, 2020.

Roy Jonker and Ton Volgenant. Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175, 1986.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010. Publisher: JMLR. org.

Ashwin Pananjady, Martin J Wainwright, and Thomas A Courtade. Denoising linear models with permuted data. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 446–450. IEEE, 2017.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. Deeppermnet: Visual permutation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3949–3957, 2017.

J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pp. 291–324. Springer, 2007.

Martin Slawski, Emanuel Ben-David, and Ping Li. Two-stage approach to multivariate linear regression with sparsely mismatched data. *J. Mach. Learn. Res.*, 21(204):1–42, 2020a.

Martin Slawski, Mostafa Rahmani, and Ping Li. A sparse representation-based approach to linear regression with partially shuffled labels. In *Uncertainty in Artificial Intelligence*, pp. 38–48. PMLR, 2020b.

Jayakrishnan Unnikrishnan, Saeid Haghighatshoar, and Martin Vetterli. Unlabeled sensing with random linear measurements. *IEEE Transactions on Information Theory*, 64(5):3237–3253, 2018.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

John Wright and Yi Ma. *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2021.

Yujia Xie, Xiangfeng Wang, Ruijia Wang, and Hongyuan Zha. A fast proximal point method for computing exact wasserstein distance. In Ryan P. Adams and Vibhav Gogate (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 433–453. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/xie20b.html.

Yujia Xie, Yixiu Mao, Simiao Zuo, Hongteng Xu, Xiaojing Ye, Tuo Zhao, and Hongyuan Zha. A hypergradient approach to robust regression without correspondence. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=l35SB-_raSQ.

Liu Yang, Ben Tan, Vincent W Zheng, Kai Chen, and Qiang Yang. Federated recommendation systems. In *Federated Learning*, pp. 225–239. Springer, 2020.

Yunzhen Yao, Liangzu Peng, and Manolis C Tsakiris. Unlabeled principal component analysis. *arXiv preprint arXiv:2101.09446*, 2021.

Zinan Zeng, Tsung-Han Chan, Kui Jia, and Dong Xu. Finding correspondence from multiple images via sparse and low-rank decomposition. In *European Conference on Computer Vision*, pp. 325–339. Springer, 2012.

Hang Zhang and Ping Li. Optimal estimator for unlabeled linear regression. In *International Conference on Machine Learning*, pp. 11153–11162. PMLR, 2020.

Hang Zhang, Martin Slawski, and Ping Li. The benefits of diversity: Permutation recovery in unlabeled sensing from multiple measurement vectors. *arXiv preprint arXiv:1909.02496*, 2019a.

Hang Zhang, Martin Slawski, and Ping Li. Permutation recovery from multiple measurement vectors in unlabeled sensing. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 1857–1861. IEEE, 2019b.

Yu Zhang, Bin Cao, and Dit-Yan Yeung. Multi-domain collaborative filtering. *arXiv preprint arXiv:1203.3535*, 2012.

Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4032–4040, 2015.

A APPENDIX

A.1 PROOF FOR THE THEORETICAL RESULTS

Proof of Proposition 1. We denote that $a_1, .., a_{r_A}$ as the linear bases of the column space of A. We can extend them to the bases of the column space of M as $a_1, .., a_{r_A}, b_1, ..., b_{r-r_A}$. In this way, there must exists a matrix $Q \in \mathbb{R}^{r \times m_B}$ such that

$$B = [a_1, .., a_{r_A}, b_1, ..., b_{r-r_A}]Q.$$

Hence, we have

$$PB = [Pa_1, .., Pa_{r_A}, Pb_1, ..., Pb_{r-r_A}]Q.$$

Similarly, there must exists a matrix $T \in \mathbb{R}^{r_A \times m_A}$ such that

$$A = [a_1, .., a_{r_A}]T.$$

Hence, we obtain that

$$[A, PB] = [a_1, .., a_{r_A}, Pa_1, .., Pa_{r_A}, Pb_1, ..., Pb_{r-r_A}] \begin{bmatrix} T & 0 \\ 0 & Q \end{bmatrix}.$$

Now, we have

$$\begin{aligned} &\text{rank}([A, PB]) \leq \text{rank}([a_1, .., a_{r_A}, Pa_1, .., Pa_{r_A}, Pb_1, ..., Pb_{r-r_A}]) \\ &\leq \text{rank}([a_1, .., a_{r_A}, Pa_1, .., Pa_{r_A}]) + r - r_A \\ &= \text{rank}([a_1, .., a_{r_A}, Pa_1, .., Pa_{r_A}] \begin{bmatrix} I_{r_A} & -I_{r_A} \\ 0 & I_{r_A} \end{bmatrix}) + r - r_A \\ &\leq r_A + r - r_A + \text{rank}([Pa_1 - a_1, .., Pa_{r_A} - a_{r_A}]). \end{aligned} \tag{20}$$

Now we denote the cycles in π_P with length greater than 1 as $C_1, ..., C_{C(\pi_P)}$, and $\zeta_1, ..., \zeta_{n-H(\pi_p)}$ as the indexes that are not in any one of $C_1, ..., C_{C(\pi_P)}$. We construct a matrix $Y \in \mathbb{R}^{(n+C(\pi_P)-H(\pi_p)) \times n}$ as:

$$\begin{aligned} &Y(i,j) = 1 \text{ if } j = \zeta_i \text{ else } Y(i,j) = 0, \text{ for } i = 1, ..., (n - H(\pi_p)); \\ &Y(i,j) = 1 \; \forall j \in C_i, \text{ and } Y(i,j) = 0 \; \forall j \notin C_i, \\ &\quad \text{for } i = (n - H(\pi_p) + 1), ..., (n + C(\pi_P) - H(\pi_p)). \end{aligned}$$

It can be verified that

$$Y(Pa_i - a_i) = 0, \; i = 1, ..., r_A.$$

We denote the null space of Y as Null$(Y) = \{x \in \mathbb{R}^n | Yx = 0\}$. From the construction of Y we can see that dim(Null(Y)) $= H(\pi_P) - C(\pi_P)$. Hence we have

$$\text{rank}([Pa_1 - a_1, .., Pa_{r_A} - a_{r_A}]) \leq H(\pi_P) - C(\pi_P). \tag{21}$$

On the other hand, we have

$$\text{rank}([A, PB]) \leq \text{rank}(A) + \text{rank}(PB) = \text{rank}(A) + \text{rank}(B) = r_A + r_B. \tag{22}$$

Combining (20), (21) and (22) , we can obtain (3). □

Following the proof of Proposition 1, it is easy to show the similar result for the case with multiple permutation, which is summarized as the Corollary 1

Corollary 1. *For the matrix $M = [A, B_1, .., B_d] \in \mathbb{R}^{n \times m}$ with $\text{rank}(M) = r$, $\text{rank}(A) = r_A$, and $\text{rank}(B_i) = r_{B_i}$, $i = 1, ...d$, we have $\forall P_1, ..., P_d \in \mathcal{P}_n$,*

$$\text{rank}([A, P_1B_1, ..., P_dB_d]) \leq \min\{n, m, r_A + \sum_{i=1}^{d} r_{B_i}, r + \sum_{i=1}^{d} H(\pi_{P_i}) - C(\pi_{P_i})\}. \tag{23}$$

Proof of Proposition 2. To prove Proposition 2, we need an important lemma on measure theory from (Halmos, 2013).

Lemma 1. *Let $p(x)$ be a polynomial on $\mathbb{R}^n$. If there exists a $x_0 \in \mathbb{R}^n$ such that $p(x_0) \neq 0$, then the Lebesgue measure of the set $\{x|p(x) = 0\}$ is 0.*

$\forall P \in \mathcal{P}_n$, we define the polynomial on $\mathbb{R}^{n\times r} \otimes \mathbb{R}^{r\times m}$ as

$$p_P^r(R, E) = \sum_{S\in\mathcal{S}_r([A,PB])} \det(S)^2,$$

where det$(\cdot)$ is the determinant of matrix, and $\mathcal{S}_r(X)$ is the set of all $r \times r$ sub-matrices in X. We denote that $r_P = \min\{2r, r + H(\pi_P) - C(\pi_P)$. We can see that $\mathrm{rank}([A, PB]) \geq r_P$ if and if only $p_P^{r_P}(R, E) > 0$. Therefore, from Lemma 1 and Proposition 1 we can conclude that if there exists two matrices $R_0 \in \mathbb{R}^{n\times r}$ and $E_0 \in \mathbb{R}^{r\times m}$ such that $p_P^{r_P}([R_0, E_0]) > 0$, then $\mathrm{rank}([A, PB]) = r_P$ holds with probability 1. In this way, we only need to construct such R_0 and E_0 for every $P \in \mathcal{P}_n$. For simplicity, we denote that $k = H(\pi_p) - C(\pi_P)$. We will discuss how to construct such R_0 and E_0 for the two cases $0 < k \leq n - r$ and $k \geq n - r$, respectively.

(1) If $0 < k \leq n - r$:

We construct the matrix $Y \in \mathbb{R}^{(n+C(\pi_P)-H(\pi_p))\times n}$ the same way with that in the proof of Proposition 1. Firstly, we show that Null(Y) =col$(P - I)$.

col$(P - I)$ $\subseteq$Null(Y): We can verify that $Y(P - I) = 0$.

Null(Y) $\subseteq$col$(P-I)$: This is equivalent to prove that Null$(P-I)$ $\subseteq$col(Y). Now we have $Px = x$, $\forall x \in$Null$(P - I)$. It can be verified that if $Px = x$, then we must have $x(s) = x(q)$ if s and q belong to the same cycle C_i, where C_i is one of the cycles in $C_1, ..., C_{C(\pi_P)}$. By the definition of Y, we can see that $x \in$ col(Y).

Now we know that rank$(P - I)$ =dim(Null(Y)) $= k$. We denote the eigen vectors of $P - I$ with non-zero eigen values as $\phi_1, ..., \phi_k$, and the eigen vectors with zero eigen values as $\phi_{k+1}, ..., \phi_n$. Now we have $(P - I)\phi_i = \lambda_i\phi_i$ for $i = 1, ..., k$ and $(P - I)\phi_i = \lambda_i\phi_i$ for $i = k + 1, ..., n$.

We construct the matrices R_0 and E_0 as

$$R_0 = [\phi_1 + \phi_{k+1}, \phi_{\min\{2,k\}} + \phi_{k+2}, ..., \phi_{\min\{r,k\}} + \phi_{k+r}],$$
$$E_0 = [I_r, \mathbf{0}_{r\times(m_A-r)}, I_r, \mathbf{0}_{r\times(m_B-r)}].$$

Now we have

$$A = [\phi_1 + \phi_{k+1}, \phi_{\min\{2,k\}} + \phi_{k+2}, ..., \phi_{\min\{r,k\}} + \phi_{k+r}, \mathbf{0}_{n\times(m_A-r)}],$$
$$B = [\phi_1 + \phi_{k+1}, \phi_{\min\{2,k\}} + \phi_{k+2}, ..., \phi_{\min\{r,k\}} + \phi_{k+r}, \mathbf{0}_{n\times(m_B-r)}],$$

since $[A, B] = R_0E_0$. Therefore, we have

$$\begin{aligned}\mathrm{rank}([A, PB]) &= \mathrm{rank}([\phi_1 + \phi_{k+1}, ..., \phi_{\min\{r,k\}} + \phi_{k+r}, \lambda_1\phi_1, ..., \lambda_{\min\{r,k\}}\phi_{\min\{r,k\}}]) \\ &= \mathrm{rank}([\phi_{k+1}, ..., \phi_{k+r}, \phi_1, .., \phi_{\min\{r,k\}}]) \\ &= r + \min\{k, r\} = \min\{2r, r + k\}.\end{aligned}$$

Now $\mathrm{rank}([A, PB]) = r_P$ by this construction of R_0 and E_0. Hence $p_P^{r_P}([R_0, E_0]) > 0$.

(2) If $k > n - r$:

We denote that the length of a cycle C as len(C), and denote the cycle with maximum length among the $C_1, ..., C_{C(\pi_P)}$ as C^*. Now we have

$$\mathrm{len}(C^*) \geq \frac{H(\pi_P)}{C(\pi_P)} \geq \frac{n}{n-k} > \frac{n}{r} \geq 2r.$$

To simplify the notations, we assume that the cycle C^* permute the first j numbers, i.e.,

$$C^* = (123...(j-2)(j-1)j),$$

where $j > 2r$. We define the vector u as $u = [1, 2, 3, ..., j-2, j-1, j, 0, ..., 0]^\top \in \mathbb{R}^n$, and denote the corresponding permutation matrix to C^* as $P_* \in \mathcal{P}_n$. We construct the matrices R_0 and E_0 as

$$R_0 = \begin{bmatrix} u & P_*^2 u & \dots & P_*^{2r-2} u \end{bmatrix},$$
$$E_0 = [I_r, \mathbf{0}_{r\times(m_A-r)}, I_r, \mathbf{0}_{r\times(m_B-r)}].$$

Now we have

$$A = [u, P_*^2 u, \dots, P_*^{2r-2} u, \mathbf{0}_{n\times(m_A-r)}],$$
$$B = [u, P_*^2 u, \dots, P_*^{2r-2} u, \mathbf{0}_{n\times(m_B-r)}].$$

Therefore, we have

$$\operatorname{rank}([A, PB]) = \operatorname{rank}([u, P_* u, \dots, P_*^{2r-1} u]) = 2r,$$

because now $[u, P_* u, \dots, P_*^{2r-1} u]$ is a circulant matrix. Now $\operatorname{rank}([A, PB]) = r_P = 2r$ by this construction of R_0 and E_0. Hence $p_P^{r_P}([R_0, E_0]) > 0$. □

Proof of Theorem 1. To prove Theorem 1, we need to derive a series results. We first start with a very important inequality w.r.t nuclear norm.

Proposition 3. *Let P be a permutation matrix, then,*

$$\|A\|_* + \|B\|_* \geq \|[A, PB]\|_* \geq \frac{\|A\|_* + \|B\|_*}{\|[U_A V_A^\top, P U_B V_B^\top]\|} \geq \frac{\|A\|_* + \|B\|_*}{\sqrt{2}}. \tag{24}$$

Based on (24), the general idea is that under the Assumptions 1, 2 and 3, we will have $\|M\|_* \approx \frac{\|A\|_* + \|B\|_*}{\sqrt{2}}$ and $\|[U_A V_A^\top, P U_B V_B^\top]\| \to 1$ as $H(\pi_P)$ increases.

Firstly, we show that under the Assumptions 1, 2, the nuclear norm of the original matrix M will reach the lower bound in (24) approximately, which is summarized as Lemma 2.

Lemma 2. *Under the Assumptions 1, 2, we have*

$$\|M\|_* \leq (\|A\|_* + \|B\|_*)/\sqrt{2} + (\sqrt{2}+1)\epsilon_1 r + \epsilon_2 \max\{\|A\|_*, \|B\|_*\}. \tag{25}$$

Then, we show that under the Assumptions 2, 3, $\|[U_A V_A^\top, P U_B V_B^\top]\| \to 1$ as $H(\pi_P)$ increases, which is summarized as Lemma 3.

Lemma 3. *Under the Assumptions 2, 3, we have*

$$\|[U_A V_A^\top, P U_B V_B^\top]\| \leq \sqrt{2 - H(\pi_P)\epsilon_3^2/2} + \sqrt{T}\epsilon_2. \tag{26}$$

Finally, we need a classical result on the tail bound for the operator norm of Gaussian matrix, whose proof can be found in (Wainwright, 2019).

Lemma 4. *Consider the random matrix $W \in \mathbb{R}^{n\times m}$ whose elements follow $\mathcal{N}(0, \sigma^2)$ i.i.d. For any $\delta > 0$, we have*

$$\|W\| \leq \sqrt{L}(2+\delta)\sigma \tag{27}$$

holds with probability greater than $1 - 2\exp\{\frac{-L\delta^2}{2}\}$, where $L = \max\{n, m\}$.

Based on Lemma 4, we have

$$\|W\|_* \leq L\|W\| \leq \sqrt{MLo\sigma}$$

holds with probability greater than $1 - 2\exp\{-\frac{M}{8Lo\sigma}\}$.

From Proposition 3, Lemma 2 and Lemma 3 we can know that, for any $P \in \mathcal{P}_n$ with $H(\pi_P)$ satisfies that

$$\frac{M}{\sqrt{2 - \frac{H(\pi_p)\epsilon_3}{2}} + \sqrt{T}\epsilon_2} - \|W\|_* > \frac{M}{\sqrt{2}} + (\sqrt{2}+1)\epsilon_1 r + \epsilon_2 N + \|W\|_*,$$

we must have

$$\begin{aligned}\|A_o, PB_o\|_* &\geq \|A, PB\|_* - \|W\|_* \\ &\geq \frac{M}{\sqrt{2 - \frac{H(\pi_p)\epsilon_3^2}{2}} + \sqrt{T}\epsilon_2} - \|W\|_* \\ &> \frac{M}{\sqrt{2}} + (\sqrt{2}+1)\epsilon_1 r + \epsilon_2 N + \|W\|_* \\ &\geq \|A, B\|_* + \|W\|_* \geq \|A_o, B_o\|_*.\end{aligned}$$

Therefore, with probability greater than $1 - 2\exp\{-\frac{M}{8L\sigma}\}$, if $H(\pi_P)$ satisfies that

$$\frac{M}{\sqrt{2 - \frac{H(\pi_p)\epsilon_3^2}{2}} + \sqrt{T}\epsilon_2} > \frac{M}{\sqrt{2}} + (\sqrt{2}+1)\epsilon_1 r + \epsilon_2 N + 2\sqrt{ML\sigma}, \tag{@}$$

we have $\|A_o, PB_o\| > \|A_o, B_o\|_*$. Now we simplify (@) as

$$\begin{aligned}&\frac{M}{\sqrt{2 - \frac{H(\pi_p)\epsilon_3^2}{2}} + \sqrt{T}\epsilon_2} > \frac{M}{\sqrt{2}} + (\sqrt{2}+1)\epsilon_1 r + \epsilon_2 N + 2\sqrt{ML\sigma} \\ \Leftrightarrow &\sqrt{2 - \frac{H(\pi_p)\epsilon_3^2}{2}} < \frac{\sqrt{2}M}{M + (\sqrt{2}+2)\epsilon_1 r + \sqrt{2}\epsilon_2 N + 2\sqrt{2ML\sigma}} - \sqrt{T}\epsilon_2.\end{aligned}$$

It can be verified that

$$\frac{\sqrt{2}M}{M + (\sqrt{2}+2)\epsilon_1 r + \sqrt{2}\epsilon_2 N + 2\sqrt{2ML\sigma}} - \sqrt{T}\epsilon_2 > 0$$

from the condition on ϵ_1, ϵ_2 and σ.

Therefore, we have

$$\begin{aligned}&\sqrt{2 - \frac{H(\pi_p)\epsilon_3^2}{2}} < \frac{\sqrt{2}M}{M + (\sqrt{2}+2)\epsilon_1 r + \sqrt{2}\epsilon_2 N + 2\sqrt{2ML\sigma}} - \sqrt{T}\epsilon_2 \\ \Leftrightarrow\ &H(\pi_P) > \frac{2}{\epsilon_3^2}\left(2 - \left(\frac{\sqrt{2}M}{M + (\sqrt{2}+2)\epsilon_1 r + \sqrt{2}\epsilon_2 N + 2\sqrt{2ML\sigma}} - \sqrt{T}\epsilon_2\right)^2\right).\end{aligned}$$

Since P^* is the optimal solution to (5), we must have

$$\|[A_o, P^*\tilde{P}B_o]\|_* \leq \|[A_o, B_o]\|_*.$$

Besides, $P^*\tilde{P}$ is also a permutation matrix, we denote its corresponding permutation as $\hat{\pi}$. Now we have

$$d_H(\pi_*, \tilde{\pi}) = H(\hat{\pi}) \leq \frac{2}{\epsilon_3^2}\left(2 - \left(\frac{\sqrt{2}M}{M + (\sqrt{2}+2)\epsilon_1 r + \sqrt{2}\epsilon_2 N + 2\sqrt{2ML\sigma}} - \sqrt{T}\epsilon_2\right)^2\right).$$

□

The proof to the auxiliary results used in the proof of Theorem 1 are provided below.

Proof of Proposition 3. Since $\|\cdot\|_*$ is a norm, we have

$$\|[A, PB]\|_* = \|[A, \mathbf{0}] + [\mathbf{0}, PB]\|_* \leq \|A\|_* + \|PB\|_* = \|A\|_* + \|B\|_*.$$

Then since $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$, we have

$$\begin{aligned}\|[A, PB]\|_* &= \sup_{\|Q\|\leq 1} \langle [A, PB], Q\rangle \\ &\geq \langle [A, PB], \frac{[U_A V_A^\top, PU_B V_B^\top]}{\|[U_A V_A^\top, PU_B V_B^\top]\|}\rangle \\ &= \frac{\|A\|_* + \|B\|_*}{\|[U_A V_A^\top, PU_B V_B^\top]\|}.\end{aligned}$$

Finally, we have

$$\begin{aligned}
\|[U_A V_A^\top, PU_B V_B^\top]\| &= \sup_{\substack{x\in\mathbb{R}^m \\ \|x\|\le 1}} \|[U_A V_A^\top, PU_B V_B^\top]x\| \\
&= \sup_{\substack{x_1\in\mathbb{R}^{m_A}, x_2\in\mathbb{R}^{m_B} \\ \|[x_1^\top, x_2^\top]\|\le 1}} \|[U_A V_A^\top x_1, PU_B V_B^\top x_2]\| \\
&\le \sup_{\substack{x_1\in\mathbb{R}^{m_A}, x_2\in\mathbb{R}^{m_B} \\ \|[x_1^\top, x_2^\top]\|\le 1}} \|U_A V_A^\top x_1\| + \|PU_B V_B^\top x_2\| \\
&\le \sup_{\substack{x_1\in\mathbb{R}^{m_A}, x_2\in\mathbb{R}^{m_B} \\ \|[x_1^\top, x_2^\top]\|\le 1}} \|x_1\| + \|x_2\| = \sqrt{2}.
\end{aligned}$$

□

Proof of Lemma 2. If $r_A \ge r_B$, we have

$$\begin{aligned}
\|M\|_* &= \|[U_A\Sigma_A V_A^\top, U_B\Sigma_B V_B^\top]\|_* \\
&= \|[U_A\Sigma_A V_A^\top, [u_A^1, ..., u_A^T, \mathbf{0}, ..., \mathbf{0}]\Sigma_B V_B^\top] + \\
&\qquad [\mathbf{0}, [u_A^1 - u_B^1, ..., u_A^T - u_B^T, u_B^{T+1}, ..., u_B^r]\Sigma_B V_B^\top]\|_* \\
&\le \|[U_A\Sigma_A V_A^\top, [u_A^1, ..., u_A^T, \mathbf{0}, ..., \mathbf{0}]\Sigma_B V_B^\top]\|_* + \\
&\qquad \|[u_A^1 - u_B^1, ..., u_A^T - u_B^T, u_B^{T+1}, ..., u_B^r]\Sigma_B V_B^\top\|_* \\
&\le \|[U_A\Sigma_A V_A^\top, [u_A^1, ..., u_A^T, \mathbf{0}, ..., \mathbf{0}]\Sigma_B V_B^\top]\|_* + \epsilon_2\|B\|_* \\
&= \|[U_A\Sigma_A V_A^\top, U_A\Sigma_B V_B^\top]\|_* + \epsilon_2\|B\|_*. \qquad (*)
\end{aligned}$$

We denote that $trace(\cdot)$ as the trace of matrix. One property of nuclear norm is

$$\|A\|_* = trace(\sqrt{AA^\top}).$$

Then we have

$$\begin{aligned}
\|[U_A\Sigma_A V_A^\top, U_A\Sigma_B V_B^\top]\|_* &= trace(\sqrt{U_A(\Sigma_A^2 + \Sigma_B^2)U_A^\top}) \\
&= \sum_{i=1}^r \sqrt{(\sigma_A^i)^2 + (\sigma_B^i)^2} \\
&\le \sum_{i=1}^r \frac{\sigma_A^i + \sigma_B^i}{\sqrt{2}} + (\sqrt{(\sigma_A^i)^2 + (\sigma_B^i)^2} - \frac{\sigma_A^i + \sigma_B^i}{\sqrt{2}}) \\
&\le \sum_{i=1}^r \frac{\sigma_A^i + \sigma_B^i}{\sqrt{2}} + (\sqrt{(\sigma_A^i)^2 + (\sigma_A^i + \epsilon_1)^2} - \frac{2\sigma_A^i - \epsilon_1}{\sqrt{2}}) \\
&\le \frac{\sqrt{2}\epsilon_1 r}{2} + \frac{\|A\|_* + \|B\|_*}{\sqrt{2}} + \\
&\qquad \sum_{i=1}^r \frac{2\sigma_A^i\epsilon_1 + \epsilon_1^2}{\sqrt{2(\sigma_A^i)^2 + 2\sigma_A^i\epsilon_1 + \epsilon_1^2} + \sqrt{2(\sigma_A^i)^2}} \\
&\le \frac{\sqrt{2}\epsilon_1 r}{2} + \frac{\|A\|_* + \|B\|_*}{\sqrt{2}} + \sum_{i=1}^r \frac{\sqrt{2}\epsilon_1}{2} + \epsilon_1 \\
&= \frac{\|A\|_* + \|B\|_*}{\sqrt{2}} + (\sqrt{2} + 1)\epsilon_1 r. \qquad (**)
\end{aligned}$$

Combining (*) and (**), we have

$$\|[A, B]\|_* \le \frac{\|A\|_* + \|B\|_*}{\sqrt{2}} + (\sqrt{2} + 1)\epsilon_1 r + \epsilon_2\|B\|_*.$$

Similarly, if $r_B \geq r_A$, we have

$$\|[A, B]\|_* \leq \frac{\|A\|_* + \|B\|_*}{\sqrt{2}} + (\sqrt{2}+1)\epsilon_1 r + \epsilon_2 \|A\|_*.$$

Combining them together, we have

$$\|[A, B]\|_* \leq \frac{\|A\|_* + \|B\|_*}{\sqrt{2}} + (\sqrt{2}+1)\epsilon_1 r + \epsilon_2 \max\{\|A\|_*, \|B\|_*\}.$$

□

Proof pf Lemma 3. Firstly, if $r_A \geq r_B$ we have

$$\begin{aligned}\|[U_A V_A^\top, PU_B V_B^\top]\| &= \|[U_A V_A^\top, P[u_A^1, ..., u_A^T, \mathbf{0}, ..., \mathbf{0}]V_B^\top]\| + \\ &\quad \|[0, P[u_B^1 - u_A^1, ..., u_B^T - u_A^T, \mathbf{0}, ..., \mathbf{0}]V_B^\top]\| \\ &\leq \|[U_A V_A^\top, P[u_A^1, ..., u_A^T, \mathbf{0}, ..., \mathbf{0}]V_B^\top]\| + \sqrt{T}\epsilon_2. \qquad (***)\end{aligned}$$

To simplify the notations, we denote that $k = H(\pi_P)$ and assume that π_P permutes the indexes $(1, ..., k)$ into $(\zeta_1, ..., \zeta_k)$. Now we have

$$\langle u_A^i, Pu_A^i \rangle = \sum_{i=1}^{k} u_A^i(i) u_A^i(\zeta_i) + \sum_{i=k+1}^{n} (u_A^i(i))^2,$$

and

$$\begin{aligned}|\sum_{i=1}^{k} u_A^i(i) u_A^i(\zeta_i)| &\leq \sum_{i=1}^{k} |u_A^i(i) u_A^i(\zeta_i)| \\ &= \sum_{i=1}^{k} \frac{(u_A^i(i))^2 + (u_A^i(\zeta_i))^2}{2} - (\frac{(u_A^i(i))^2 + (u_A^i(\zeta_i))^2}{2} - |u_A^i(i) u_A^i(\zeta_i)|) \\ &\leq \sum_{i=1}^{k} (u_A^i(i))^2 - (\frac{(u_A^i(i))^2 + (|u_A^i(i)| - \epsilon_3)^2}{2} - |u_A^i(i)|(|u_A^i(i)| + \epsilon_3)) \\ &= \sum_{i=1}^{k} (u_A^i(i))^2 - (\frac{\epsilon_3^2}{2} + 2|u_A^i(i)|\epsilon_3) \leq \sum_{i=1}^{k} (u_A^i(i))^2 - \frac{\epsilon_3^2}{2}.\end{aligned}$$

Hence we must have

$$|\langle u_A^i, Pu_A^i \rangle| \leq 1 - \frac{k\epsilon_3^2}{2}.$$

Therefore, we have

$$\begin{aligned}\delta(U_A, P) &\overset{\text{def.}}{=} \max_{\substack{x,y \in \mathbb{R}^T, \\ \|x\|=1, \|y\|=1}} \langle [u_A^1, ..., u_A^T]x, [Pu_A^1, ..., Pu_A^T]y \rangle \\ &= \max_{\substack{x,y \in \mathbb{R}^T, \\ \|x\|=1, \|y\|=1}} \sum_{i=1}^{T} x(i) y(i) \langle u_A^i, Pu_A^i \rangle \\ &\leq \max_{\substack{x,y \in \mathbb{R}^T, \\ \|x\|=1, \|y\|=1}} (1 - \frac{k\epsilon_3^2}{2}) \sum_{i=1}^{T} x(i) y(i) \\ &= 1 - \frac{k\epsilon_3^2}{2}.\end{aligned}$$

Now we have,

$$\|[U_A V_A^\top, P[u_A^1, ..., u_A^T, \mathbf{0}, ..., \mathbf{0}]V_B^\top]\| = \sup_{\substack{x\in\mathbb{R}^n,\\ \|x\|=1}} \|[U_A V_A^\top, P[u_A^1, ..., u_A^T, \mathbf{0}, ..., \mathbf{0}]V_B^\top]x\|$$

$$\leq \sup_{\substack{x_1\in\mathbb{R}^{m_A}, x_2\in\mathbb{R}^{m_B}\\ \|[x_1^\top, x_2^\top]\|\leq 1}} \sqrt{1 + \langle U_A V_A^\top x_1, P[u_A^1, ..., u_A^T, \mathbf{0}, ..., \mathbf{0}]V_B^\top x_2\rangle}$$

$$\leq \sup_{\substack{x_1\in\mathbb{R}^{m_A}, x_2\in\mathbb{R}^{m_B}\\ \|[x_1^\top, x_2^\top]\|\leq 1}} \sqrt{1 + \delta(U_A, P)\|x_1\|\|x_2\|} \leq \sqrt{2 - \frac{k\epsilon_3^2}{2}}.. \tag{****}$$

Combining (***) and (****), we have

$$\|[U_A V_A^\top, PU_B V_B^\top]\| \leq \sqrt{2 - \frac{k\epsilon_3^2}{2}} + \sqrt{T}\epsilon_2.$$

The proof is similar for the case $r_B \geq r_A$. □

A.2 Asymptotic behavior of Theorem 1.

In this section, we will discuss about the asymptotic behavior ($n \to \infty$) of the error bound in Theorem 1.

We start with a simple observation: Without $\epsilon_1 \to 0, \epsilon_2 \to 0, \sigma \to 0$, the original matrix will be impossible to recover by minimizing nuclear norm for sufficient large n. This is also reflected in the error bound of Theorem 1, where the right hand side of (10) could become trivial, i.e., larger than n, when n is sufficiently large.

We provide a simple example to validate this observation. Suppose that the original matrix is $M = [u, u] + W$, where the elements of W follow $\mathcal{N}(0, \sigma^2)$ and $u \in \mathbb{R}^n$ is a random vector whose elements are i.i.d. following the uniform distribution on $[0, 1]$. From the result in (David & Nagaraja, 2004), p. 135, we know that

$$\mathbb{E}[\max_{i\neq j} |u(i) - u(j)|] \approx O(n^{-1}\log(n)).$$

Therefore, we can construct a permutation matrix $P \in \mathcal{P}_n$ with $H(\pi_P) = n$, such that the following inequality holds with high probability,

$$|\|[u, Pu]\|_* - \|[u, u]\|_*| \leq \|Pu - u\|_2 = O(n^{-\frac{1}{2}}\log(n)).$$

On the other hand, from Lemma 4 we can know that $\|W\|_* \approx O(\sigma n)$ with high probability. Now if we need that $\|[u, Pu] + W\|_* > \|[u, u] + W\|_*$, we at least require that $\sigma = o(n^{-\frac{3}{2}}\log(n))$. Otherwise, it will be impossible to distinguish the matrices $[u, Pu] + W$ and $[u, u] + W$ through the value of nuclear norm.

Finally, for this simple example, we have $\epsilon_1 = \epsilon_2 = 0$. Besides, from (David & Nagaraja, 2004), we can also know that ϵ_3 is at most $O(n^{-\frac{3}{2}})$ with high probability. With a simple calculattion, we can find that the error bound in Theorem 1 is at least $O(n^{\frac{5}{2}}\sigma^{\frac{1}{2}})$. Therefore, in this example, we at least require that $\sigma = o(n^{-5})$ to guarantee a constant error bound for arbitrary n.

A.3 Dual Problem of (16)

To simplify the notation, we denote the primal problem as

$$\underset{P\in\Pi(\mathbf{1}_n, \mathbf{1}_n)}{\text{minimize}} \langle C, P\rangle + \epsilon H(P).$$

We define two dual variables $\alpha, \beta \in \mathbb{R}^n$. The Lagrangian function is

$$L(P, \alpha, \beta) = \langle C, P\rangle + \epsilon\langle \log P - \mathbf{1}_{n\times n}, P\rangle + \langle \mathbf{1}_n - P\mathbf{1}_n, \alpha\rangle + \left\langle \mathbf{1}_n - P^T\mathbf{1}_n, \beta\right\rangle. \tag{28}$$

Now we minimize the Lagrangian function w.r.t P (We note that $H(P)$ implicitly imposes that $P \in \mathbb{R}_+^{n\times n}$). From the first-order necessary condition of unconstrainted optimization, we have

$$\begin{aligned} & C-\alpha \oplus \beta + \epsilon \log(P) = 0, \\ & \quad \Downarrow \\ & P = \exp\left\{ \frac{\alpha \oplus \beta - C}{\epsilon} \right\}. \end{aligned} \tag{29}$$

Substituting it into the Lagrangian function (28) we have the dual objective

$$q(\alpha, \beta) = \min_P L(P, \alpha, \beta) = \langle \mathbf{1}_n, \alpha \rangle + \langle \mathbf{1}_n, \beta \rangle - \epsilon \left\langle \mathbf{1}_{n\times n}, \exp\left\{ \frac{\alpha \oplus \beta - C}{\epsilon} \right\} \right\rangle.$$

Therefore the dual problem is

$$\max_{\alpha,\beta\in\mathbb{R}^n} \langle \mathbf{1}_n, \alpha \rangle + \langle \mathbf{1}_n, \beta \rangle - \epsilon \left\langle \mathbf{1}_{n\times n}, \exp\left\{ \frac{\alpha \oplus \beta - C}{\epsilon} \right\} \right\rangle. \tag{30}$$

We can recover the primal solution P from the dual solution α, β via (29).

A.4 A Stable Implementation for Skinhorn Algorithm

The Skinhorn algorithm (Peyré et al., 2019) are often used to solve the dual problem (30), and the standard form of it reads

$$p^{(t+1)} \leftarrow \frac{\mathbf{1}_n}{Kq^{(t)}} \text{ and } q^{(t+1)} \leftarrow \frac{\mathbf{1}_n}{K^\top p^{(t+1)}},$$

where $K = \exp\left\{ \frac{\alpha\oplus\beta-C}{\epsilon} \right\}$, and $p = \exp(\frac{\alpha}{\epsilon})$, $q = \exp(\frac{\beta}{\epsilon})$. If we adopt a small ϵ, the elements of K can overflow to infinity or zero, which causes a numerical issue. We can remedy this by using a different implementation from (Peyré et al., 2019).

$$\begin{aligned} \alpha^{(t+1)} &\leftarrow \text{Min}_\epsilon^{\text{row}}(C - \alpha^{(t)} \oplus \beta^{(t)}) + \alpha^{(t)}, \\ \beta^{(t+1)} &\leftarrow \text{Min}_\epsilon^{\text{col}}(C - \alpha^{(t+1)} \oplus \beta^{(t)}) + \beta^{(t)}, \end{aligned}$$

where for any $A \in \mathbb{R}^{n\times m}$, we define the operator $\text{Min}_\epsilon^{\text{row}}$ and $\text{Min}_\epsilon^{\text{col}}$ as

$$\begin{aligned} \text{Min}_\varepsilon^{\text{row}}(\mathbf{A}) &\overset{\text{def.}}{=} \left(\text{min}_\varepsilon \mathbf{A}(i,\cdot)\right)_i \in \mathbb{R}^n, \\ \text{Min}_\varepsilon^{\text{col}}(\mathbf{A}) &\overset{\text{def.}}{=} \left(\text{min}_\varepsilon \mathbf{A}(\cdot,j)\right)_j \in \mathbb{R}^m, \end{aligned}$$

and for any vector $z = [z_1, ..., z_n]^\top \in \mathbb{R}^n$, we denote

$$\text{min}_\epsilon z \overset{\text{def.}}{=} \min_i z_i - \epsilon \log \sum_j e^{-(z_j - \min_i z_i)/\epsilon}$$

as the ϵ-soft minimum for the elements of z.

A.5 Relationship between M^{3}O and the Soft-Impute Algorithm

Soft-Impute algorithm (Mazumder et al., 2010) is a classical algorithm for matrix completion. Specifically, it tries to solve the nuclear norm regularized problem

$$\underset{\widehat{M}}{\text{minimize}} \ \frac{1}{2} \left\| \mathcal{P}_\Omega(X) - \mathcal{P}_\Omega(\widehat{M}) \right\|_F^2 + \lambda \left\| \widehat{M} \right\|_*. \tag{31}$$

Soft-Impute is a simple iterative algorithm with the following two steps:

$$\widehat{X} \leftarrow \mathcal{P}_\Omega(X) + \mathcal{P}_\Omega^\perp(\widehat{M}), \tag{32}$$

$$\widehat{M} \leftarrow \text{prox}_{\lambda\|\cdot\|_*}(\widehat{X}) = U\mathcal{S}_\lambda(D)V^\top, \tag{33}$$

where $\widehat{X} = UDV^\top$ denotes the singular value decomposition of $\widehat{X}$, and $\mathcal{P}_\Omega^\perp$ is the operator that selects entries whose indexes are not belonging to Ω. Here $\mathcal{S}_\lambda$ is the soft-thresholding operator that operates element-wise on the diagonal matrix D, i.e., replacing D_{ii} with $(D_{ii} - \lambda)_+$.

Consider the partial observation extension. For the M^{3}O algorithm, if an exact permutation matrix is obtained, i.e., $\widehat{P} = \exp\left\{\frac{\alpha^* \oplus \beta^* - C(\widehat{M}_B)}{\epsilon}\right\} \in \mathcal{P}_n$, it is easy to verify that the the gradient in Algorithm 1 has the following form,

$$\nabla_{\widehat{M}} F_\epsilon(\widehat{M}, \alpha^*, \beta^*) = 2(\mathcal{P}_\Omega(\widehat{M}) - \mathcal{P}_\Omega([A, \widehat{P}\tilde{B}])).$$

In this way, if we adopts $\rho_k = 0.5$, the proximal gradient update becomes

$$\widehat{M}^{\text{k+1}} \leftarrow \text{prox}_{\lambda\|\cdot\|_*}(\mathcal{P}_\Omega([A, \widehat{P}\tilde{B}]) + \mathcal{P}_\Omega^\perp(\widehat{M}^k)).$$

In practice, $\widehat{P}$ often becomes very close to an exact permutation matrix and the stepsize often reaches the upper bound 0.5, when the algorithm is close to convergence. In this scenario, our algorithm becomes equivalent to the Soft-Impute algorithm. Therefore, we adopt the Soft-Impute algorithm as a baseline method for matrix completion without correspondence issue.

A.6 M^{3}O-AS-DE FOR THE D-CORRESPONDENCE PROBLEM

In this section, we summarize our proposed algorithm M^{3}O-AS-DE for the general d-correspondence problem (19) in Algorithm 2. To determinate the stop of the Max-Oracle, we find that the criterion

$$\frac{1}{\sqrt{n}} \left\| \mathbf{1}_n^\top \widehat{P} - \mathbf{1}_n^\top \right\|_2 \leq \varepsilon$$

works well in practice, which serves as a good indicator for the ε-good optimality.

A.7 THE BASELINE ALGORITHM

We also extend the Baseline algorithm to a similar d-correspondence problem as (19). Specifically, the extended Baseline algorithm tries to solve the unsmoothed problem

$$\min_{\widehat{M}} \min_{P_1,\ldots,P_d} \left\| \mathcal{P}_\Omega(A_o) - \mathcal{P}_\Omega(\widehat{M}_A) \right\|_F^2 + \sum_{l=1}^{d} \langle C(\widehat{M}_{B_l}), P_l \rangle + \lambda \left\| \widehat{M} \right\|_*, \tag{34}$$
$$\text{s.t. } P_l \in \mathcal{P}_n, \text{ for } l = 1, ..., d.$$

We summarize the algorithm in Algorithm 3.

A.8 THE MUS ALGORITHM

In this section, we provide details for the MUS algorithm discussed in the Section 4. Firstly, inspired by (Yao et al., 2021), we first transform the MRUC problem, i.e, to recover $[A, B]$ from $[A, \tilde{P}B]$, into a MUS problem as follows,

$$\min_{P \in \mathcal{P}_n, W \in \mathbb{R}^{m_B \times m_A}} \|A - P\tilde{P}BW\|_F^2. \tag{35}$$

Then, for the scenario without multiple correspondence and missing values, we adopt the algorithm in (Zhang & Li, 2020) to solve (35).

To extend it into the d-correspondence problem considered by (19), we adopt tow simple procedures. Specifically, to deal with the missing value, we first fill in the missing entries of each submatrices using the Soft-Impute algorithm. As for the multiple correspondence issue, we simply run the MUS algorithm in multiple times. For example, if we want solve the d-correspondence problem, we typically apply the MUS algorithm to the following series of problems in turn,

$$\min_{P \in \mathcal{P}_n, W \in \mathbb{R}^{m_B \times m_A}} \|A_o - PB_o^l W\|_F^2, \ l = 1, ..., d.$$

Algorithm 2: M^{3}O-AS-DE

Input: stepsize parameter ω, number of correspondence d, number of iterations N, number of tolerance steps K, initial entropy coefficient ϵ, tolerance ε, observation matrix $M_o = [A_o, B_o^1, ..., B_o^d]$, initial matrix $\widehat{M} = [\widehat{M}_A, \widehat{M}_{B_1}, ..., \widehat{M}_{B_d}]$, nuclear norm coefficient λ, the set of observable indexes Ω.

Initialize $\widehat{P}_{\text{new}}^l = \mathbf{0}_{n\times n}$ for $l = 1, ..., d$.
for $k = 1 : N$ **do**
  for $l = 1 : d$ **do in parallel**
    $\widehat{P}_{\text{old}}^l = \widehat{P}_{\text{new}}^l$.
    $\hat{\alpha}^l = \hat{\beta}^l = \mathbf{1}_n$.
    Compute the partial pairwise cost matrix $C(\widehat{M}_{B_l})$.
    while $\frac{1}{\sqrt{n}} \left\| \mathbf{1}_n^\top \widehat{P} - \mathbf{1}_n^\top \right\|_2 > \epsilon$ **do**
      $\hat{\alpha}^l \leftarrow \text{Min}_\epsilon^{\text{row}}(C(\widehat{M}_{B_l}) - \hat{\alpha}^l \oplus \hat{\beta}^l) + \hat{\alpha}^l$,
      $\hat{\beta}^l \leftarrow \text{Min}_\epsilon^{\text{col}}(C(\widehat{M}_{B_l}) - \hat{\alpha}^l \oplus \hat{\beta}^l) + \hat{\beta}^l$,
      $\widehat{P}_{\text{new}}^l \leftarrow \exp\left\{ \frac{\hat{\alpha}^l \oplus \hat{\beta}^l - C(\widehat{M}_{B_l})}{\epsilon} \right\}$.
    end
    Compute the stepsize ρ_l as discussed in Section 3.
    $\widehat{M}_{B_l} \leftarrow \widehat{M}_{B_l} - \rho_l \nabla_{\widehat{M}} F_\epsilon^l(\widehat{M}_{B_l}, \alpha^l, \beta^l)$, where

$$F_\epsilon^l(\widehat{M}_{B_l}, \alpha, \beta) \overset{\text{def.}}{=} \langle \mathbf{1}_n, \alpha \rangle + \langle \mathbf{1}_n, \beta \rangle - \epsilon \left\langle \mathbf{1}_{n\times n}, \exp\left\{ \frac{\alpha \oplus \beta - C_\Omega(\widehat{M}_{B_l})}{\epsilon} \right\} \right\rangle.$$

  end
  $\widehat{M}_A \leftarrow \mathcal{P}_\Omega(A) + \mathcal{P}_{\bar{\Omega}}^\perp(\widehat{M}_A)$.
  $\widehat{M} \leftarrow \text{prox}_{\lambda\|\cdot\|_*}([\widehat{M}_A, \hat{M}_{B_1}, ..., \widehat{M}_{B_d}]))$.
  if *the objective value is not improved over K steps* **then**
    $\epsilon \leftarrow \epsilon/2$.
  end
end

Algorithm 3: Baseline

Input: number of iterations N, number of Proximal Gradient iterations N_p, tolerance ε, observation matrix $M_o = [A_o, B_o^1, ..., B_o^d]$, initial matrix $\widehat{M} = [\widehat{M}_A, \widehat{M}_{B_1}, ..., \widehat{M}_{B_d}]$, nuclear norm coefficient λ, partial observation operator $\mathcal{P}_\Omega$.

for $k = 1 : N$ **do**
  for $l = 1 : d$ **do in parallel**
    Solving the inner problem of (34) for $\hat{P}^l$ up to tolerance ε via Hungarian algorithm.
  end
  $X \leftarrow [A_o, \hat{P}^1 B_o^1, ..., \hat{P}^d B_o^d]$.
  for $i = 1 : N_p$ **do**
    $\hat{X} \leftarrow \mathcal{P}_\Omega(X) + \mathcal{P}_{\bar{\Omega}}^\perp(\hat{M})$,
    $\hat{M} \leftarrow \text{prox}_{\lambda\|\cdot\|_*}(\hat{X})$.
  end
end

A.9 DETAILS FOR THE EXPERIMENTS

We use Matlab 2020b for the numerical experiments. The computer environment consists of Intel i9-10920x for CPU and 32GB RAM.

A.9.1 HYPERPARAMETERS SETTING

Simulated data. We adopt fixed nuclear norm coefficient λ in the experiments on simulated data. Specifically, for each setting, we choose the best λ out of three candidate values that are 0.4, 0.5 and 0.6. Since adopting large ω will preserve the final performance and only degrade the convergence speed, we take $\omega = 3$ for all the experiments. For the tolerance of Sinkhorn algorithm, we take $\varepsilon = 0.01$ for all the experiments.

MovieLens 100K. For all the algorithms, we adopt a sequence of values for λ. Specifically, we start the algorithm with $\lambda = 300$, and once the algorithm stops improving the objective function for 10 steps, we shrink the value as $\lambda \leftarrow \lambda - 10$ until λ becomes lower than 10. We take $\omega = 0.5$ for all the experiments and also set the tolerance of Sinkhorn algorithm as $\varepsilon = 0.01$.

A.9.2 PHASE TRANSITION WITH DIFFERENT INITIALIZATIONS.

In this section, we conduct a simple experiment to explore the sensitivity of M^{3}O w.r.t initialization by varying the distance between initialization and the ground-truth matrix. We could expect that the variance of the performance of M^{3}O should decrease as the distance decreases.

We generate different initializations in the following way: We first generate two matrices M and W independently following the way described in Section 4.1, and we employ M as the ground-truth matrix. Then, we generate the initialization for M^{3}O as

$$\hat{M} = \Lambda M + (1 - \Lambda)W,$$

where $\Lambda \in (0, 1)$ is a coefficient designed for controlling the distance between initialization and the ground-truth matrix.

Figure 5 shows a phase transition phenomenon for M^{3}O algorithm w.r.t to the coefficient Λ, which is well aligned with our expectation.

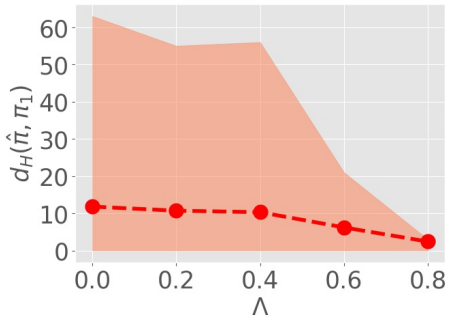


Figure 5: A phase transition phenomenon for M^{3}O algorithm w.r.t to the distance between initialization and the ground-truth matrix. The experiment is conducted on a 1-correspondence problem, with $|\Omega| \cdot 100\%/(n \cdot m) = 80\%$, $\eta = 0.1$, $n = m = 100$, $r = 5$, $m_A = 60$, and $m_1 = 40$. The mean with minimum and maximum are calculated from 10 different random initializations.

A.9.3 NUMBERS OF SKINHORN ITERATION

Typically, the numbers of Skinhorn iteration required to retrieve an ε-good solution mainly depends on the entropy coefficient ϵ. This also implies that the decaying entropy regularization strategy can also accelerate the convergence process. Figure 6 shows the relationship between the numbers of Skinhorn iteration and entropy coefficient ϵ under the same simulated data setting with Figure 2. The dash lines and intervals reflect mean, min, maximum aggregated from 20 independent trials. For a practical implementation, we restrict the maximum numbers of Skinhorn iteration to 10000 on the numerical experiments.

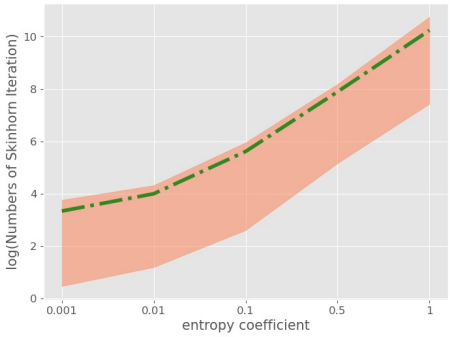


Figure 6: The required numbers of Skinhorn iteration v.s. entropy coefficient ϵ

A.9.4 PROBLEM FORMULATION FOR THE FACE RECOVERY PROBLEM

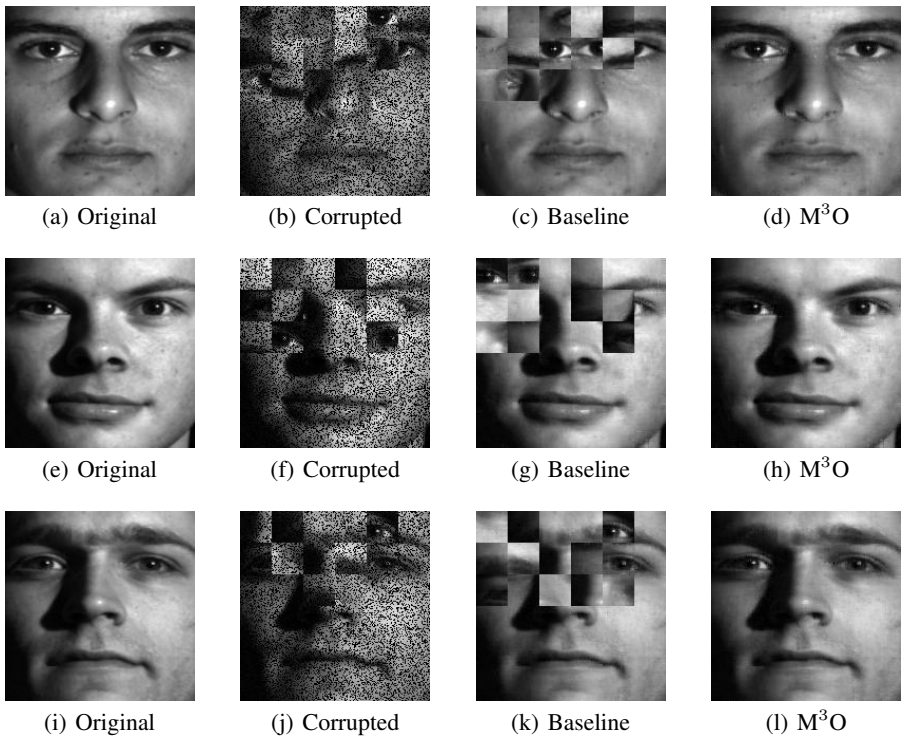
In the face recovery experiment, the cost matrix C is constructed as

$$C(i,j) = \|P_\Omega(B(i) - \widehat{M}(j))\|_F^2,$$

where $B(1), ..., B(13) \in \mathbb{R}^{30\times 30}$ are the shuffled pixel blocks from the upper left of the corrupted image shown in Figure 4(b), and $\widehat{M}(1), ..., \widehat{M}(13) \in \mathbb{R}^{30\times 30}$ are the corresponding recovered pixel blocks from the upper left of the current recovered image.

We choose fixed stepsize $\rho_k = 0.1$, and choose the initial entropy coefficient as $\epsilon = 100$. To obtain the initial matrix $\widehat{M}$, we first complete each pixel blocks independently using the Soft-Impute algorithm. We denote the filled matrix as M_1, and carry out the singular decomposition of it as $M1 = \sum_i \sigma_i u_i v_i^\top$. Then we set the initial matrix as $\widehat{M} = \sigma_1 u_1 v_1^\top$.

More results similar to Figure 4 are shown in Figure 7.



(a) Original (b) Corrupted (c) Baseline (d) M^{3}O

(e) Original (f) Corrupted (g) Baseline (h) M^{3}O

(i) Original (j) Corrupted (k) Baseline (l) M^{3}O

Figure 7: Performance of M^{3}O on more face images from Yale B database.