Low-Rank Matrix Recovery with Unknown Correspondence

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 data mishandling or privacy concern. We show that, in some applications, it is possible to recover M via solving a nuclear norm minimization problem. Moreover, under a proper low-rank condition on M, we derive a 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 or noisy, due to either missing identity information or privacy reasons [Unnikrishnan et al., 2018, Gruteser et al., 2003, Das and Lee, 2018]. Examples include the record linkage problem [Chan and Loh, 2001], the federated recommender system [Yang et al., 2020] and the vertical federated learning [Nock et al., 2021]. Consider 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 and 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,

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

Applications. The major application of MRUC problem is related to Vertical Federated Learning (VFL) [Kairouz et al., 2021], which aims at learning from feature partitioned data. This work specifically considers Recommender System (RS) in the context of VFL. One classical work on this problem is the multi-domain recommender system considered in [Zhang et al., 2012]. Unfortunately, they neglect 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 correspondence information could be noisy due to mishandle in data processing. The other is that those platforms may not be allowed to share the true linkage information for preserving privacy. As the first step to address these issues, in this work, we study RS in an

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

extreme setting of VFL, i.e., no correspondence information is provided. 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. Though less practical, this problem is still interesting to know under what structure in data one can guarantee a successful recovery. 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.

Unlabeled Sensing. One similarly motivated problem is the **U**nlabeled **S**ensing (**US**) problem considered by [Unnikrishnan et al., 2018, Pananjady et al., 2017, Tsakiris et al., 2020, Peng and Tsakiris, 2020, Tsakiris and Peng, 2019, Slawski et al., 2021, Xie et al., 2021]. Especially, as discussed in Appendix J, the MRUC problem is closely related to the **M**ultivariate **U**nlabeled **S**ensing (**MUS**) problem, which has been studied in [Zhang et al., 2019a,b, Zhang and 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. When $m_1 = 1$, the MUS problem reduces to an US problem. Despite of the similarity to the MUS problem, we remark that MRUC problem has its own distinct features and, as shown in Section 4, the algorithm for the MUS problem 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 work [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] first learn a robust subspace following the idea of [Slawski et al., 2020b,a], and then solves problem (1) heuristically as multiple independent US problems using algorithms from [Tsakiris et al., 2020, Peng and Tsakiris, 2020]. However, there are two main drawbacks in their algorithm that largely limit its practical value. First, as discussed in Appendix J and Remark 8, it ignores the interaction among the shuffled columns and hence can not recover the permutation correctly. Second, their method can not deal with data with missing values. Another recent paper [Nock et al., 2021] also shares a similar concern with ours on how correspondence information can affect VFL, though in a different context.

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 propose a nuclear norm minimization problem as a surrogate for problem (1), and is also the first to study the property of nuclear norm under permutation. Practically, we propose an efficient algorithm $\mathrm{M}^3\mathrm{O}$ that solves the nuclear norm minimization problem, which overcomes the aforementioned two shortcomings in [Yao et al., 2021]. Notably, $\mathrm{M}^3\mathrm{O}$ 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. Based on these findings, we also reach a novel and important observation for VFL: *Even without any data linkage information, it is still possible for each participant/platform to benefit from VFL.*

Outline. We start with building the theoretical understanding for the problem (1) and its convex relaxation in Section 2. Then, based on the theoretical intuition obtained from Section 2, we develop an efficient algorithm in Section 3 for most complicated scenario. 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

In this section, we study the role of low-rank model for recovering row permutation.

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

Definition 2.1 (Cycle decomposition of a permutation [Dummit and Foote, 1991])**.** Let $\mathcal{S}$ be a finite set, $\pi(\cdot)$ be a permutation on $\mathcal{S}$. A cycle $(a_1, ..., a_n)$ is a permutation sending a_j to a_{j+1} for $1 \le j \le n-1$ 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].

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

It can be verified that any permutation on a finite set has a unique cycle decomposition [Dummit and 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 $\mathcal{C}(\pi)$. 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 $H(\pi) > \mathcal{C}(\pi)$ 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 $\mathcal{C}(\pi) = 2$, $H(\pi) = 5$.

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 $r = \mathrm{rank}(M)$, $r_A = \mathrm{rank}(A)$, $r_B = \mathrm{rank}(B)$. 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 could become stronger, if π permutes more rows and contains less cycles.

Proposition 2.2. *For all $P \in \mathcal{P}_n$, we have*

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

Similar result for the case with multiple permutations is summarized in Corollary A.1 in Appendix A. It turns out that, without any further assumption on M, (3) is sharp and cannot be improved. Notably, the upper bound in (3) is attained with probability 1 for a generic low-rank random matrix.

Definition 2.3. 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 2.4. *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 below holds with probability 1.*

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

Discussion on Proposition 2.4. It is worthwhile to mention that our Proposition 2.4 strengthens the Theorem 1 in [Yao et al., 2021] to some extent. Specifically, [Yao et al., 2021] shows that, with probability 1, the rank of the perturbed matrix will never be lower than that of the original matrix. Compared to them, our result precisely predicts how much the rank will increase after row perturbation. Besides, Proposition 2.4 is especially favorable from the optimization perspective, as now the rank is a monotone function w.r.t the degree of perturbation.

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.,

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

where σ^2 denotes the variance of the noise. We 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 $\mathrm{rank}(X) = \|[\sigma_X^1, ..., \sigma_X^k]\|_0$, from Proposition 2.4 we can view the perturbation effect of a permutation to a low-rank matrix as breaking the sparsity of its singular values, which leads naturally to the nuclear norm minimization problem that has been shown to be robust to additive noise and favor low-rank solution [Wright and Ma, 2021], i.e.,

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

Theoretical justification for the nuclear norm. Nuclear norm has a long history being used as a convex surrogate for the rank, and it has been theoretically justified for applications like low-rank matrix completion [Candès and Tao, 2010, Wright and 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. To derive the worst-case error bound of nuclear norm minimization, we propose the following assumption on M.

Assumption 2.5. There exists a constant $\epsilon_1 \geq 0, \epsilon_2 \geq 0, \epsilon_3 \geq 0$ such that

$$|\sigma_A^i - \sigma_B^i| \leq \epsilon_1,\ \forall i = 1, .., r, \quad (6)$$

$$\|u_A^i - u_B^i\| \leq \epsilon_2, \forall i = 1, ..., T, \quad (7)$$

$$\min_{u\in U} \min_{i\neq j} |u(i) - u(j)| \geq \epsilon_3 > 0, \quad (8)$$

where we denote $\sigma_A^i = 0$ if $i > r_A$, and similarly for σ_B^i, $T = \min\{r_A, r_B\}$ and $U = \{u_A^1, ..., u_A^T, u_B^1, ..., u_B^T\}$.

Here we provide some intuition behind these assumptions. Firstly, from the definition of nuclear norm, it can be simply verified for any $P \in \mathcal{P}_n$ that

$$-Z/N \leq (\|[A, PB]\|_* - \|M\|_*)/\|M\|_* \leq Z/N, \quad (9)$$

where $N = \max\{\|A\|_*, \|B\|_*\}$ and $Z = \min\{\|A\|_*, \|B\|_*\}$. The inequality (9) indicates that A and B should have comparable magnitude, i.e., $\|A\|_* \approx \|B\|_*$, otherwise the influence of the permutation will be less significant. With this observation, as depicted by (6), we assume that the singular values of A and B

are comparable. As for (7), we propose it with an aim to capture the intuition that if A and B are data from the same group of users, the distance (in SVD sense) between A and B should be close, i.e., the matrix $[A, B]$ should be "low-rank". We would like to interpret the constants ϵ_2 as a continuous measure for the low-rankness of a matrix, because it indicates that the column space of M can be approximated by the column space of one of its submatrices. Lastly, it is easy to verify that if there is a $P \in \mathcal{P}_n$ such that $u_B^i = P u_B^i$ for all i, then $[A, PB] = [A, B]$. Therefore, we propose (8) to avoid this case.

Remark 1. Though these assumptions could be refined, we remark that they are almost sharp. In Appendix B, we construct a few concrete counterexamples which do not satisfy these assumptions and are impossible to be recovered within meaningful accuracy by nuclear norm minimization problem.

With these assumptions, we derive the following result, 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

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

Proposition 2.6. *Under Assumptions 2.5, if additionally* $\epsilon_1 \leq \frac{D}{4r}$, $\epsilon_2 \leq \min\{\frac{1}{2\sqrt{2T}}, \frac{\sqrt{2}D}{2N}\}$, *and* $\sigma \leq \frac{D}{16L^2}$, *then the following bound*

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

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.

Remark 2. From Proposition 2.6 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. We can also discover that the correspondence can be difficult to recover when: The rank of original matrix M is high; The magnitude of A and B w.r.t rank or nuclear norm are not comparable; The strength of noise is high. Notably, the numerical experiments in Section 4.1 corroborate these findings as well. Due to page limit, we refer detailed discussion and analysis on Proposition 2.6 to Appendix C.

Remark 3. In many applications, we can only observe part of the full data. Therefore, it is worthwhile to investigate

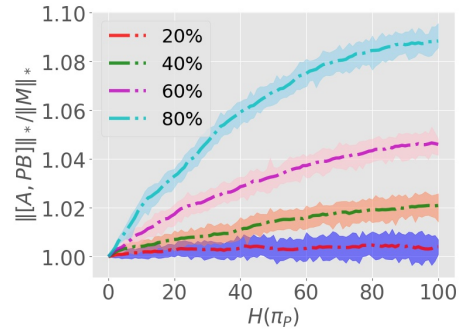


Figure 1: The relationship under different percentages of observable entries.

whether nuclear norm minimization could work when we can only access a small subset of the entries in M_o. Notably, Figure 1 empirically gives the positive answer and shows that the "*monotone relationship of nuclear norm w.r.t numbers of permuted rows*" 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].

3 ALGORITHM

In this section, we develop an algorithm for MRUC based on the intuition obtained from Section 2. Moreover, we require that the algorithm can deal with 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,

$$\min_{\widehat{M} \in \mathbb{R}^{n\times m}} \min_{P \in \mathcal{P}_n} \left\|\mathcal{P}_\Omega([A_o, PB_o]) - \mathcal{P}_\Omega(\widehat{M})\right\|_F^2 + \lambda \left\|\widehat{M}\right\|_*, \quad (11)$$

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 (11) equivalently as

$$\min_{\widehat{M} \in \mathbb{R}^{n\times m}} \min_{P \in \mathcal{P}_n} \left\|\mathcal{P}_\Omega(A_o) - \mathcal{P}_\Omega(\widehat{M}_A)\right\|_F^2 + \langle C(\widehat{M}_B), P\rangle + \lambda \left\|\widehat{M}\right\|_*, \quad (12)$$

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

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

Baseline algorithm. A conventional strategy to handle an optimization problem like (12) 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.

$$\widehat{M}^{\text{new}} \leftarrow \underset{\widehat{M}\in\mathbb{R}^{n\times m}}{\arg\min} \left\|\mathcal{P}_\Omega([A_o, \widehat{P}^{\text{old}}B_o]) - \mathcal{P}_\Omega(\widehat{M})\right\|_F^2 + \lambda\left\|\widehat{M}\right\|_*, \quad (13)$$

$$\widehat{P}^{\text{new}} \leftarrow \underset{P\in\mathcal{P}_n}{\arg\min} \langle C(\widehat{M}_B^{\text{new}}), P\rangle. \quad (14)$$

The first update step (13) is a convex optimization problem and can be solved by the proximal gradient algorithm [Mazumder et al., 2010]. The second update step (14) is actually a discrete optimal transport problem which can be solved by the classical Hungarian algorithm with time complexity $O(n^3)$ [Jonker and 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 (14) 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 (14) with the following continuous optimization problem

$$\min_{P\in\Pi(\mathbf{1}_n,\mathbf{1}_n)} \langle C(\widehat{M}_B), P\rangle + \epsilon\mathcal{H}(P), \quad (15)$$

where $\mathcal{H}(P) \overset{\text{def.}}{=} \sum_{i,j} P(i,j)(\log(P(i,j))-1)$ is the matrix negative entropy and $\epsilon > 0$ is the regularization coefficient.

Algorithm 1 M^{3}O (Simplified)

Input: tolerance ε, observation M_o, initialization $\widehat{M}$.
repeat
Run the Sinkhorn algorithm to find α^*, β^* such that

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

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

until converged

Notably, (15) 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 (15),

$$\max_{\alpha,\beta\in\mathbb{R}^n} W_\epsilon(\widehat{M}_B, \alpha, \beta) \overset{\text{def.}}{=} \langle\mathbf{1}_n, \alpha\rangle + \langle\mathbf{1}_n, \beta\rangle - \epsilon\left\langle\mathbf{1}_{n\times n}, \exp\left\{\frac{\alpha\oplus\beta - C(\widehat{M}_B)}{\epsilon}\right\}\right\rangle, \quad (16)$$

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 (12) with (15), we end up with solving the following unconstrained min-max optimization problem

$$\min_{\widehat{M}} \max_{\alpha,\beta} \left\|A - \widehat{M}_A\right\|_F^2 + W_\epsilon(\widehat{M}_B, \alpha, \beta) + \lambda\left\|\widehat{M}\right\|_*. \quad (17)$$

Follows the idea of [Jin et al., 2020], we consider to adopt a proximal gradient algorithm with a Max-Oracle for (17). Specifically, we employ the Sinkhorn algorithm [Peyré et al., 2019] as the Max-Oracle to retrieve an ε-good solution of the inner max problem (16). 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, ρ_k is the gradient stepsize and

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

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.

Remark 4. A recent work [Xie et al., 2020] proposes a decaying strategy for the entropy regularization coefficient ϵ in (15) so that the optimal solutions of (14) and (15) do not deviate too much. Inspired by it, in our practice, we take large ϵ in the beginning and gradually shrink it by half whenever the objective value stops improving for K steps.

Remark 5. 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 (15) $\widehat{P}_k$ at the kth iteration, we compute the two statistics

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

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 6. 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

$$M_o = [A_o, B_o^1, ..., B_o^d] = [A, \tilde{P}_1 B_1, ..., \tilde{P}_d B_d] + W,$$

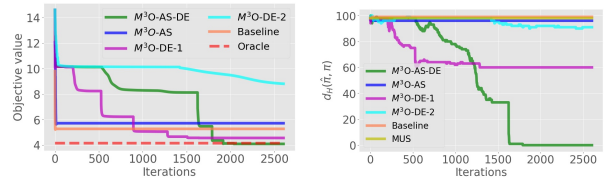
where $\tilde{P}_l \in \mathcal{P}_n$ and W is a noise matrix. We refer such problem as the ***d*-correspondence** problem. An important observation is that, although the number of possible correspondence increase exponentially as d grows, the complexity of M^{3}O per iteration only linearly increases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

$$\min_{\widehat{M}} \min_{P_1,...,P_d} \left\|\mathcal{P}_\Omega(A_o) - \mathcal{P}_\Omega(\widehat{M}_A)\right\|_F^2 + \sum_{l=1}^{d} \left\{ \langle C(\widehat{M}_{B_l}), P_l \rangle + \epsilon \mathcal{H}(P_l) \right\} + \lambda \left\|\widehat{M}\right\|_*, \quad (18)$$
$$\text{s.t. } P_l \in \Pi(\mathbf{1}_n, \mathbf{1}_n), \; l = 1, ..., d,$$

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 7. Since problem (11) 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 G.



(a) Objective value (b) Permutation error

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

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 4 and 5. 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 K.

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 (13) and (14).

3. MUS: Since there is currently no existing algorithm directly applicable to the scenario considered by (18), we modify and extend the algorithm in [Zhang and 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 I.

Remark 8. As discussed in [Pananjady et al., 2017], leveraging the prior knowledge that multiple columns are shuffled by the same permutation is generally helpful for permutation recovery. This is why we only adopt the MUS algorithm in [Zhang and Li, 2020] instead of those US algorithms considered by [Yao et al., 2021] for comparison. For a more serious and experimental discussion, we refer readers to Appendix J.

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 $M_o = [A, P_1 B_1, ..., P_d B_d]$. Finally, we remove $(1 - |\Omega| \cdot 100\%/(n \cdot m))$ percent of the entries of M_o randomly and uniformly, where $|\Omega|$ indicates 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.

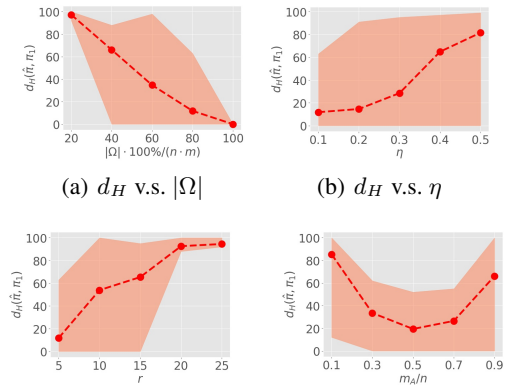
In the following results, we denote π_l as the corresponding permutation to P_l. We initialize $\widehat{M}$ from Gaussian distribution for the M^{3}O algorithm and its variants. We choose initial ϵ as 0.1 and $K = 100$ as the default for the decaying entropy regularization, and set $\omega = 3$ as the default for the adaptive stepsize. We also report the achieved objective values of (18) 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.

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	$\frac{\lvert\Omega\rvert \cdot 100\%}{nm}$	$\frac{1}{d}\sum_{l=1}^{d} d_H(\hat{\pi}_l, \pi_l)$
(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)

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 2 and 3. We also find that the performance of M^{3}O tends to have high variance. This is mainly because M^{3}O is sensitive to random initialization, and more details on this phenomenon are in Appendix K.2. In practice, we recommend to run M^{3}O a few times with different random initializations.



(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.

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.

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 and Konstan, 2015]. In this application, we mainly focus on the metric Root Mean Squared Error (RMSE), i.e.,

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

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

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

extract five most popular genres, which are Comedy (C), Romance (R), Drama (D), Action (A), Thriller (T) 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.

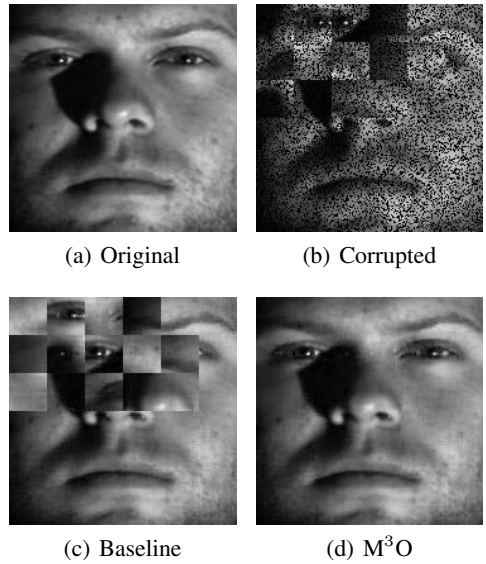
Results. As discussed in experiments on the simulated data, the exact recovery of correspondence becomes impossible due to the small amount of observable entries. Therefore, in the following experiment, since exact correspondence is not needed, we fix $\epsilon = 0.05$ for M^{3}O. Table 2 shows the results 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 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	C	R	D	A	T	Total
SIR	1.020	1.016	0.981	0.980	0.981	0.994
SIC	0.969	0.970	0.932	0.918	0.925	0.942
MUS	0.966	0.984	0.942	0.931	0.931	0.949
Baseline	0.973	0.956	0.938	0.911	0.915	0.940
M^{3}O	**0.9399**	**0.879**	**0.914**	**0.856**	**0.857**	**0.895**
Oracle	0.944	0.783	0.906	0.818	0.810	0.867

4.3 VISUAL PERMUTATION RECOVERY

We also show that M^{3}O is flexible and can also be applied to a visual jigsaw puzzle. 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 extra data by merely exploiting the underlying low-rank structure of the image itself. A typical result is shown in Figure 4. The experiment details and more results are provided in Appendix K.



(a) Original (b) Corrupted

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

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

5 CONCLUSION

In this paper, we study the important MRUC problem where part of the observed submatrix is shuffled. Such problem underlies the record linkage problem in VFL [Nock et al., 2021]. 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 also provide an almost sharp sufficient condition under which minimizing nuclear norm is provably efficient for recovering permutation. For practical implementations, we propose an efficient algorithm, the M^{3}O algorithm, which consistently achieves the best performance over several baselines in all the tested scenarios. For future works, it is important to extend the theoretical results to the scenario with missing values, and hopefully derive a theorem that can rigorously quantify the remarkable phenomenon exhibited in Figure 1.

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*, pages 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.

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, pages 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.

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

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

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

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.

Richard Nock, Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Jakub Nabaglo, Giorgio Patrini, Guillaume Smith, and Brian Thorne. The impact of record linkage on learning from feature partitioned data. In *International Conference on Machine Learning*, pages 8216–8226. PMLR, 2021.

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

Liangzu Peng and Manolis C Tsakiris. Linear regression without correspondences via concave minimization. *IEEE Signal Processing Letters*, 27:1580–1584, 2020.

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*, pages 3949–3957, 2017.

J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 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*, pages 38–48. PMLR, 2020b.

Martin Slawski, Guoqing Diao, and Emanuel Ben-David. A pseudo-likelihood approach to linear regression with partially shuffled data. *Journal of Computational and Graphical Statistics*, pages 1–13, 2021.

Manolis Tsakiris and Liangzu Peng. Homomorphic sensing. In *International Conference on Machine Learning*, pages 6335–6344. PMLR, 2019.

Manolis C Tsakiris, Liangzu Peng, Aldo Conca, Laurent Kneip, Yuanming Shi, and Hayoung Choi. An algebraic-geometric approach for linear regression without correspondences. *IEEE Transactions on Information Theory*, 66(8):5130–5144, 2020.

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, editors, *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pages 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*, pages 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*, pages 325–339. Springer, 2012.

Hang Zhang and Ping Li. Optimal estimator for unlabeled linear regression. In *International Conference on Machine Learning*, pages 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)*, pages 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*, pages 4032–4040, 2015.