2022 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-2159-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISIT50566.2022.9834608

Nonparametric Matrix Estimation with One-Sided Covariates

Christina Lee Yu
Operations Research and Information Engineering
Cornell University
Ithaca, NY, USA
cleeyu@cornell.edu

***Abstract*—Consider the task of matrix estimation, in which we desire to estimate a ground truth matrix given sparse and noisy observations. Each entry is observed independently with probability p, and additionally perturbed with additive observation noise. Assume the (u, i)-th entry of the ground truth matrix can be described by $f(\alpha_u, \beta_i)$ for some Holder smooth function f. We consider the setting where the row covariates α are unobserved yet the column covariates β are observed. We provide an algorithm and accompanying analysis which shows that our algorithm improves upon naively estimating each row separately when the number of rows is not too small. Furthermore when the matrix is moderately proportioned, our algorithm achieves the minimax optimal nonparametric rate of an oracle algorithm that knows the row covariates. In simulated experiments we show our algorithm outperforms other baselines in low data regimes.**

***Index Terms*—nonparametric regression, matrix estimation, side information**

I. Introduction

Matrix completion, or matrix estimation, refers to the task of estimating a ground truth matrix $F \in \mathbb{R}^{n \times m}$ from a sparse and noisy dataset X. Matrix estimation is a fundamental building block of standard data analysis pipelines, as most datasets in reality have measurement noise, mistakes, and missing data. The statistical properties of matrix estimation has been well studied in the context of low-rank models under anonymity, i.e. supposing that the only available data is the matrix X itself, and no further attributes of the rows and columns are known.

In this paper, we additionally consider the impact of having access to one-sided covariate information, which is often available in real-world applications. Previous works in matrix estimation that consider access to side information assume that the side information reveals the row or column subspace of the ground truth matrix. In simple terms, this imposes that the data behaves linearly with respect to the revealed side information. In practice this is unrealistic, and there is often considerable effort put into heuristic feature engineering to generate a large set of functions of the covariates, with the hope that the generated features contain the desired subspace. In contrast, in this work we consider models of side information in which the subspace is nonlinearly related to the covariates.

The results presented in this paper quantify the statistical gain for matrix estimation due to having access to one-side covariate information, under a nonparametric setting where the primary assumption is that the ground truth matrix is smooth with respect to the covariates. In particular, we assume that the ground truth matrix F can be described by a latent function f such that $F_{ij} = f(\alpha_i, \beta_j)$, where $\alpha_i \in [0,1]^{d_1}$ are *unknown latent* features of the rows, and $\beta_j \in [0,1]^{d_2}$ are the *known* column features. For each of the n rows and m columns, the corresponding features, or covariates, are assumed to be sampled independently uniformly on the unit hypercubes. This nonparametric setting is a more expressive model class than low rank models. The dataset consists of the known column features $\{\beta_j\}_{j \in [m]}$ along with a sparsely observed matrix X where each entry $(i,j) \in [n] \times m$ is sampled independently with probability p. For each observed entry (i,j), $X_{ij} = F_{ij} + \epsilon_{ij}$ where $\epsilon_{ij} \sim N(0, \sigma^2)$. The goal is to produce an estimate $\hat{F}$ for the ground truth matrix F given the dataset of observations and column features. We measure performance by the mean squared error (MSE) defined as

$$\text{MSE} = \mathbb{E}\Big[\tfrac{1}{mn} \textstyle\sum_{i \in [n]} \sum_{j \in [m]} (\hat{F}_{ij} - F_{ij})^2\Big].$$

A. Contributions

Our results show that there are three data regimes depending on the number of rows relative to the number of datapoints per row. When the matrix is short and fat, in particular $n = O((mp)^{d_1/(2\lambda+d_2)})$, then the optimal estimator is to simply estimate each row separately. In this regime, the bias introduced by incorporating data from another row is larger than the accuracy to which a single row can already be estimated.

In a moderate regime when $n = \omega((mp)^{d_1/(2\lambda+d_2)})$ and $n = O\left((mp)^{\min((2\lambda+d_1)/d_2, (2d_1+d_2)/(4\lambda+d2))}\right)$, we propose an algorithm that matches the minimax-optimal nonparametric rate of the oracle algorithm which is given access to the row covariates. This implies that we in fact lose very little by only knowing the column covariates and not the row covariates as our algorithm performs as well as if it knew the row covariates.

When the matrix is tall and narrow, in particular $n = \omega\left((mp)^{\min((2\lambda+d_1)/d_2, (2d_1+d_2)/(4\lambda+d2))}\right)$, we show that our proposed algorithm still outperforms naive regression on each row separately, as the distance between rows can be estimated more efficiently than estimating the full function for each row separately. However, our algorithm does not match the oracle algorithm that has knowledge of the row covariates, as the relatively few observations per row limits the accuracy to which the relationship between rows can be estimated.

We provide simulations that show our algorithm outperforms other existing baselines, even compared to low rank matrix estimation algorithms on low rank data itself. Our empirical results highlight that in sparse data regimes, even when the true model is actually low rank, our algorithm which utilizes nonlinear side information at a cost of considering the larger class of nonparametric models outperforms low rank matrix completion algorithms.

B. Related Literature

The related work spans a wide literature across matrix estimation and nonparametric regression, both old and well-established fields of study. In addition, there is a host of empirical work in recommendation systems that proposes heuristics for matrix completion with side information, e.g. using neural netowrks. As the contribution of this paper is in theoretically quantifying statistical complexity, we focus our comparison on algorithms that have accompanying theoretical guarantees, giving attention to existing lower and upper bounds from nonparametric regression and matrix estimation.

Classical sparse matrix estimation assumes that there are no observed covariates, and the only data available is a sparse noisy observation of a ground truth matrix. This problem has been widely studied for the setting when the ground truth matrix is low rank and incoherent, i.e. the latent factors exhibit regularity, and the observations are sampled uniformly across the matrix. Algorithms include nuclear norm minimization [1], [2], singular value thresholding [3], [4], gradient descent [5], [6], alternating least squares [7], and nearest neighbor [8], [9]. For any rank r matrix, low rank matrix completion algorithms produce estimates that converge in MSE as long as the number of observed entries scales as $\Omega(r \max(n,m) \log \min(n,m))$, linear in the maximum dimension of the matrix. This was shown to be tight up to polylog factors in [1], [5].

In the low rank setting, there have been a sequence of works that consider additional side information in the form of covariate matrices or similarity based graphs. In inductive matrix completion, the primary assumption is that the covariate matrices reveal the row and column subspace of the ground truth matrix [10]–[20]. This implies that the ground truth matrix F can be factored according to YMZ^T, where $Y \in \mathbb{R}^{n \times r_1}$ and $Z \in \mathbb{R}^{m \times r_2}$ are the given covariate matrices. As the unknown parameters reduces to only $r_1 \times r_2$, the given side information significantly reduces the sample complexity of matrix estimation from linear in $\max(n,m)$ to logarithmic in the dimension. Assuming that the observed covariates directly reveal the subspace is a strong condition that is often not satisfied in practice, as it assumes the data is linearly related to the observed covariates.

An alternate form of side information has been considered in the form of graph or clustering based side information. The key idea of graph regularized matrix completion is to impose a regularizer that encourages the estimate to be smooth with respect to an underlying graph [21]–[27]. The majority of these works are primarily empirical with limited statistical guarantees, with limited results where the sample complexity still scales linearly in $\max(n,m)$.

There has been a limited number of works that also consider a nonparametric model class as we assume in this paper. Under a Lipschitz model, [28], [29] require a significantly costlier sample complexity of $\min(n,m)\max(n,m)^{1/2}$. The models most relevant to our setting is from the graphon estimation literature, which specifically focuses on the case with binary observations and a symmetric matrix [30]–[33]. When $n = m$, $\alpha_i = \beta_i \in [0,1]^d$, and f is a symmetric $2d$-dimensional (λ, L)-Holder function, the singular value thresholding estimator achieves MSE $= O((pm)^{-2\lambda/(2\lambda+d)})$. This matches the mimimax-optimal nonparametric rate for estimating a d-dimensional function given pn observations, which would be the setting of estimating the latent function for a single row given the column covariates using only the datapoints within the row. Under the nonparametric setting, there has not been any existing works that incorporate side information with matrix estimation.

As we assume a nonparametric model, the crux of our algorithm will build upon nonparametric kernel regression. An excellent presentation of results and techniques in nonparametric estimation can be found in [34], of which we summarize a few key results below. Let the function class $\mathcal{F}(\lambda, L)$ denote all d-dimensional (λ, L)-Holder functions with $\lambda \in (0,1]$ such that for all $x, x' \in [0,1]^d$ and $f \in \mathcal{F}(\lambda, L)$,

$$|f(x) - f(x')| \leq L\|x - x'\|_\infty^\lambda. \tag{1}$$

Let N denote the total number of observed datapoints. The minimax optimal mean squared error rate for the class of (λ, L)-Holder functions is $\Omega(N^{-2\lambda/(2\lambda+d)})$. This rates is achieved by locally polynomial estimators and thus it is tight. This literature however does not consider the value of sharing data amongst different regression tasks, as considered in our setting. If we performed regression on each row's data separately, then the minimax error rate would be $(pm)^{-2\lambda/(2\lambda+d)}$, as the number of datapoints in a given row is $N = pm$.

Our proposed algorithm and analysis will rely upon the task of estimating the L_2 distance between two functions f and f' given observations from both. In particular we take inspiration from the results in [35] which show that the minimax optimal rate for estimating the norm $\|f\|_2$ with N observations is

$$\mathbb{E}[(\|\hat{f}\|_2 - \|f\|_2)^2] \asymp \max(N^{-4\lambda/(4\lambda+d)}, N^{-1/2}),$$

which is faster than the minimax rate of estimating the full f.

II. Model

Consider a dataset consisting of a sparse data matrix $X \in \mathbb{R}^{n \times m}$ and observed column covariates $\{\beta_i\}_{i \in [n]}$. The goal is to estimate a ground truth matrix F given the data matrix and observed covariates. We make the following assumptions on the data generating model.

Assumption 1 (Row and column covariates)**.** *Each row $u \in [n]$ is associated with a latent covariate $\alpha_u \in [0,1]^{d_1}$, and each column $i \in [m]$ is associated with an observed covariate*

$\beta_i \in [0,1]^{d_2}$. *These covariates are sampled independently uniformly on the specified unit hypercubes,* $\alpha_u \sim U([0,1]^{d_1})$ *and* $\beta_i \sim U([0,1]^{d_2})$. *The column covariates* $\{\beta_i\}_{i\in[m]}$ *are observed, but the row covariates* $\{\alpha_u\}_{u\in[n]}$ *are not known.*

Assumption 2 (Gaussian observation noise)**.** *Each observed datapoint* X_{ui} *is a noisy signal of a ground truth function* f*, perturbed with additive Gaussian noise,*

$$X_{ui} = f(\alpha_u, \beta_i) + \epsilon_{ui},$$

where $\epsilon_{ui} \sim N(0, \sigma^2)$ *are independent mean-zero Gaussian noise terms. For* $F \in \mathbb{R}^{n\times m}$ *denoting the ground truth matrix, it follows that* $\mathbb{E}[X_{ui}] = F_{ui} = f(\alpha_u, \beta_i)$.

Assumption 3 (Smoothness of latent function)**.** *The latent function* f *is an* (λ, L)*-Holder function with* $\lambda \in (0, 1]$. *For* $\lambda = 1$, f *is L-Lipschitz.*

Assumption 4 (Uniform Bernoulli sampling)**.** *Each entry is observed independently with probability* p. *For* $\mathcal{E}$ *denoting the set of observed indices, each index pair* $(u, i) \in \mathcal{E}$ *with probability* p. *We overload notation and also let* $\mathcal{E}_{ui}$ *denote the indicator function* $\mathbb{I}((u, i) \in \mathcal{E})$.

We assume a nonparametric model, where the ground truth matrix is described by a latent function f. This is in contrast to the majority of the literature which assumes a low rank model. Any Lipschitz function can be approximated by an approximately low rank matrix [4], [33], [36]. Any rank r matrix can be described with the inner product function computed over r dimensional latent feature spaces, which also exhibits Lipschitzness. When it comes to side information however, our model is more realistic as we allow for nonlinear relationships between the side information and the observed matrix data. In particular, the attempts to incorporate side information to low rank models require the side information to reveal the latent subspace, which is a significantly stronger assumption than mildly assuming smoothness as in our model.

We consider an asymmetric side information setting, in which the row covariates are not known whereas the column covariates are know. This asymmetry arises in applications where one side could be anonymized due to privacy concerns, e.g. a customer-product interaction dataset with anonymized customer information, or when one side corresponds to time, date, or geographical location, which can be linked to publically available covariates.

III. Algorithm

Assume that we have two freshly sampled sets of observations, $\mathcal{E}'$ used for learning the row distances, and $\mathcal{E}''$ used for generating the final prediction. The independence of the two datasets facilitates easier analysis by decoupling the different steps of the algorithm; empirically we reuse the same dataset for each part of the algorithm, which still performs well.

If we knew the row latent variables α in addition to the column latent variables β, then we can simply use any nonparametric regression estimator such as kernel regression to match the minimax optimal rates. The idea of kernel regression is simple; estimate the value of the target function at (α, β) using a weighted average of the datapoints, where higher weights are given to nearby or similar datapoints, as determined by the kernel. For simplicity we consider a rectangular kernel, which gives equal weight to all datapoints for which the corresponding (α', β') are at distance no more than a specified threshold. This is also equivalent to a fixed threshold nearest neighbor algorithm, where the nearest neighbor set is defined by the distances in the latent space.

In our problem setting, we do not have knowledge of α. Instead we propose an algorithm that uses data to estimate a proxy for distance between the rows, which is then used to determine the set of nearest neighbors used to construct the final estimates. As a result, the crux of the algorithm and resulting analysis is to make sure that the estimated distances are estimated closely enough to add value to the final estimates with respect to the bias variance tradeoff. We construct distances that approximate the L_2 difference in the latent functional space, evaluated with respect to the column latent variables, given by

$$d^2(u, v) = \tfrac{1}{m} \textstyle\sum_{l\in[m]} \left(f(\alpha_u, \beta_l) - f(\alpha_v, \beta_l)\right)^2.$$

Since we don't have access to the latent function f, we instead estimate the function $f(\alpha_u, \cdot)$ associated to row u using the data in row u itself. While any nonparametric estimator could be used, we use the Nadaraya-Watson estimator with a rectangular kernel with respect to the infinity norm for ease of analysis [34]. Our algorithm has three steps, which we detail below.

Step 1: Initial row latent function estimates. For each row u, compute $\hat{f}(u, i)$ to approximate $f(\alpha_u, \beta_i)$ via the Nadaraya-Watson estimator on row u's data, according to

$$\hat{f}(u, i) = \frac{1}{W_{ui}} \sum_{j\in[m]} \mathbb{I}((u, j) \in \mathcal{E}') X_{uj} K\left(\tfrac{\beta_i - \beta_j}{h}\right)$$
$$\text{for} \quad W_{ui} = \textstyle\sum_{j\in[m]} \mathbb{I}((u, j) \in \mathcal{E}') K\left(\tfrac{\beta_i - \beta_i}{h}\right),$$

with bandwidth h and kernel function $K(b) = \mathbb{I}(\|b\| \le \frac{1}{2})$, where $\|\cdot\|$ denotes the infinity norm, $\|b\| = \max_l |b_l|$, and $\mathcal{E}_{ui}$ denotes the indicator function $\mathbb{I}((u, i) \in \mathcal{E})$.

Step 2: Pairwise row distance estimates. For each pair of rows u and v, compute $\hat{d}^2(u, v)$ to approximate $d^2(u, v)$ by comparing $\hat{f}(u, i)$ and $\hat{f}(v, i)$ across all $i \in [m]$, according to

$$\hat{d}^2(u, v) = \tfrac{1}{m} \textstyle\sum_{i\in[m]} (\hat{f}(u, i) - \hat{f}(v, i))^2 - \xi^2_{uv},$$

for ξ^2_{uv} computed to offset the bias that arises from the squared terms involving the observation noise (distributed $N(0, \sigma^2)$),

$$\xi^2_{uv} := \frac{\sigma^2}{m} \sum_{i\in[m]} \sum_{l\in[m]} \left(\tfrac{\mathbb{I}((u,l)\in\mathcal{E}')}{W^2_{ui}} + \tfrac{\mathbb{I}((v,l)\in\mathcal{E}')}{W^2_{vi}}\right) K^2\left(\tfrac{\beta_l - \beta_i}{h}\right).$$

Step 3: Nearest neighbor estimates. For each index pair (u, i), estimate F_{ui} using a fixed radius nearest neighbor estimator,

$$\hat{F}_{ui} = \frac{\sum_{v\in\mathcal{N}_1(u,\eta_1)} \sum_{j\in\mathcal{N}_2(i,\eta_2)} X_{vj} \mathbb{I}((v, j) \in \mathcal{E}'')}{|(\mathcal{N}_1(u, \eta_1) \times \mathcal{N}_2(i, \eta_2)) \cap \mathcal{E}'|},$$

where the neighborhood sets are defined as

$$\mathcal{N}_1(u,\eta_1) := \{v \in [n] : \hat{d}^2(u,v) \le \eta_1^2\}$$
$$\mathcal{N}_2(i,\eta_2) := \{j \in [m] : \|\beta_i - \beta_j\| \le \eta_2\}$$

for some chosen thresholds η_1, η_2.

Our algorithm has three tuning parameters, h, η_1, and η_2. The most costly step is computing the nearest neighbor estimates. This can be accelerated using approximate nearest neighbor algorithms, or by computing a block constant estimate resulting from clustering using the distances. The same performance guarantees can be achieved with reduced computational complexity by choosing an appropriate number of clusters.

IV. Theoretical Guarantees

We quantify the gain due to side information by bounding the mean squared error achieved by our algorithm relative to naive row regression. When there are few rows, i.e. $n = O((mp)^{d_1/(2\lambda+d_2)})$, then estimating each row separately using classical regression techniques obtains the minimax rate

$$\text{MSE} = O((mp)^{-2\lambda/(2\lambda+d_2)}).$$

Even if the row covariates α were observed, the achieved MSE from performing regression on the full matrix dataset would be $O((pmn)^{-2\lambda/(2\lambda+d_1+d_2)}$, which is worse than the MSE achieved from estimating on each row separately, as it ignores the additional structure in a matrix dataset which enforces that the covariates of all the datapoints are aligned along a grid corresponding to the rows and columns. Essentially, when n is small, each row is sufficiently different so that the bias from sharing data outweighs the benefits of variance reduction.

Our algorithm focuses on the more interesting regime where $n = \omega((mp)^{d_1/(2\lambda+d_2)})$. We will choose the bandwidth of our initial row regression estimates according to

$$h = \Theta\Big(\Big(\frac{pm}{\log mn}\Big)^{-\min(1/d_2, 2/(d_2+4\lambda))}\Big).$$

Theorem IV.1. *For $n = \omega((mp)^{d_1/(2\lambda+d_2)})$ and $n = O\left((mp)^{\min((2\lambda+d_1)/d_2,(2d_1+d_2)/(4\lambda+d2))}\right)$, our algorithm with $\eta_1 = \eta_2^\lambda = (pnm)^{-\lambda/(2\lambda+d_1+d_2)}$ achieves rate*

$$\text{MSE} = O\left((pmn)^{-2\lambda/(2\lambda+d_1+d_2)}\right).$$

For $n = \omega\left((mp)^{\min((2\lambda+d_1)/d_2,(2d_1+d_2)/(4\lambda+d2))}\right)$ our algorithm with $\eta_1 = 2h^\lambda$, $\eta_2 = h$ achieves rate

$$\text{MSE} = O\Big(\Big(\frac{pm}{\log mn}\Big)^{-\min(2\lambda/d_2, 4\lambda/(d_2+4\lambda))}\Big).$$

In the regime that $n = \omega((mp)^{d_1/(2\lambda+d_2)})$ and $n = O\left((mp)^{\min((2\lambda+d_1)/d_2,(2d_1+d_2)/(4\lambda+d2))}\right)$, our estimator in fact achieves the nonparametric minimax optimal rate of an oracle regression algorithm which observed both row and column covariates. This means that even without knowledge of the row covariates, the algorithm can learn the row distances from the data itself accurately enough to match the oracle.

For $n = \omega\left((mp)^{\min((2\lambda+d_1)/d_2,(2d_1+d_2)/(4\lambda+d2))}\right)$, our algorithm still improves upon the performance of naive row regression, but does not match the oracle. This is expected as the knowledge of the row covariates is more powerful when there is a large number of rows relative to the datapoints in each row. When the number of datapoints in each row is small relative to the number of rows, the performance of our algoritm is limited by the rate of estimating L_2 distance between the latent functions associated to pairs of rows. Our analysis for our algorithm is nearly tight, as there is a matching (up to polylog factors) minimax lower bound of $(mp)^{-4\lambda/(4\lambda+d_2)}$ for estimating the L_2 norm of a function when the β covariates are evenly spaced [35]. The term $(pm/\log mn)^{-2\lambda/d_2}$ arises from the randomness of the column covariates β.

Without access to side information, classical matrix estimation requires a sample complexity linear in $\max(n,m)$, i.e. $p = \omega(\min(n,m)^{-1})$ in order for a convergent estimator to exist, as any low rank matrix completion algorithm requires a growing number of observed entries in each row and column. In contrast, when we have access to column covariates, the sample complexity is only linear in n, i.e. $p = \omega(m^{-1})$. When m grows faster than n, then this could significantly reduce sample complexity. In particular, when m is large and $p = o(n^{-1})$, with high probability there will be columns which have zero entries observed. Using covariate knowledge, we can predict empty columns with other columns having similar covariates.

V. Proof Sketch

As the final estimate is constructed using fixed radius nearest neighbor, the primary piece of the proof is to show that the estimated distances concentrate, as stated in Lemma V.1.

Lemma V.1. *With prob $1 - o(1)$, for all $u, v \in [n]^2$,*

$$|\hat{d}(u,v) - d(u,v)| \le O\left(\frac{2^{3d_2/4}\sigma \log^{1/2}(n)}{(pm)^{1/2}h^{d_2/4}} + 2L(h/2)^\lambda\right)$$

for h satisfying $h = o(\log^{-2/d_2}(n))$.

To prove Lemma V.1, we will separately bound the error due to the additive Gaussian observation noise, and the error due to the randomness in sampling $\{\beta_i\}_{i\in[m]}$ and the indices in $\mathcal{E}'$. Let $\tilde{f}(u,i)$ and $\tilde{d}^2(u,v)$ denote the hypothetical distances estimated from comparing the Nadaraya-Watson estimator on row u's data assuming no observation noise,

$$\tilde{f}(u,i) = \tfrac{1}{W_{ui}} \textstyle\sum_{j\in\mathcal{E}'_u} K\left(\frac{\beta_i - \beta_j}{h}\right) f(\alpha_u, \beta_j)$$
$$\tilde{d}^2(u,v) = \tfrac{1}{m} \textstyle\sum_{l\in[m]} \left(\tilde{f}(u,l) - \tilde{f}(v,l)\right)^2.$$

The proof of Lemma V.1 bounds the error due to observation noise as captured by $|\hat{d}(u,v) - \tilde{d}(u,v)|$ using concentration inequalities that exploit the Gaussian distribution of the additive observation noise terms. We bound the error due to the sparse sampling as captured by $|\tilde{d}(u,v) - d(u,v)|$ using Holder-smoothness and regularity of the sampling model for the column covariates and the location of the observed entries.

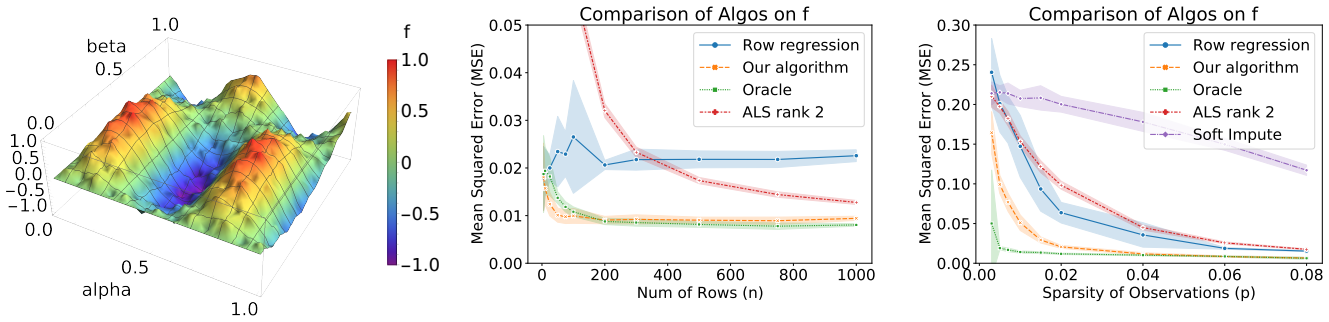

Fig. 1: Results of the experiments on the synthetic dataset: (Left) latent function f; (Center) plots MSE vs. number of rows n for different algorithms; (Right) plots MSE vs. sparsity p for different algorithms

By approximating (u, i) via averaging over nearby datapoints (v, j) for which $\beta_i - \beta_j$ is small and $d(u, v)$ is small, we can bound the MSE using the Holder smoothness property of f, the construction of $d(u, v)$, and the uniformity of the sampling model. Our proof slightly deviates from the typical bias variance calculations common to nearest neighbor estimators, as $d(u, v)$ only estimates the L_2 difference between the latent functions of u and v. A bound on the L_2 difference can translate into a very loose bound on the L_∞ distance especially in high dimensions. As a result, $d(u, v)$ being small does not imply that $f(\alpha_u, \beta)$ is close to $f(\alpha_v, \beta)$ for all values of β. However as our goal is to compute an aggregate MSE bound, we can still obtain a good bound on the MSE without having a tight L_∞ bound. In particular, under the good event that the distances estimates concentrate and the nearest neighborhood sets are sufficiently large, Lemma V.2 bounds the MSE of our estimator.

Lemma V.2. *Conditioned on* $\cap_{u,v\in[n]^2}\{|\hat{d}(u, v) - d(u, v)| \leq \Delta\}$, $\cap_{u\in[n]}\{|\mathcal{N}_1(u, \eta_1)| \geq z_1\}$ *and* $\cap_{i\in[m]}\{|\mathcal{N}_2(i, \eta_2)| \geq z_2\}$, *with respect to the randomness in* $\mathcal{E}''$ *and* $\{X_{ab}\}_{(a,b)\in\mathcal{E}''}$, *for* $p = \omega((z_1 z_2)^{-1})$, *with probability* $1 - o(1)$, *it holds that*

$$MSE = O\left(\frac{\sigma}{\sqrt{p z_1 z_2}} + L^2\eta_2^{2\lambda} + (\eta_1 + \Delta)^2\right).$$

The final result follows from appropriately choosing the parameters h, η_1, and η_2, which determine Δ, z_1, and z_2.

VI. Synthetic Experiments

We construct synthetic experiments to illustrate the performance of our algorithm in practice. We use the following latent function f to generate the ground truth matrices for our experiments, where the row covariates $\{\alpha_i\}_{i\in[n]}$ and column covariates $\{\beta_j\}_{j\in[m]}$ are sampled independently from $U[0, 1]$.

$$f(\alpha, \beta) = \sin(10\alpha)\sin(4\beta) + 0.2\,(\sin(40\alpha)\sin(40\beta))^3$$

The corresponding ground truth matrix is rank 2, where the latent factors of the low rank decomposition are non-linear functions of α and β, such that knowledge of α and β without knowledge of the nonlinear transformation does not reveal the latent row and column subspaces. For a given sparsity level p, each index $(i, j) \in [n] \times [m]$ is observed independently with probability p, upon which the associated datapoint X_{ij} is perturbed by additive noise distributed as $N(0, \sigma^2)$.

We compare our algorithm both to the naïve algorithm that estimates each row separately using kernel regression, and to an oracle kernel regression algorithm that is given knowledge of both α and β. We also compared our algorithm against classical matrix completion algorithms such as SoftImpute and alternating least squares (ALS) with rank parameter 2.

The center plot of Figure 1 shows the resulting MSE of each of the algorithms as a function of the number of rows n, where we set $m = 500$, $p = 0.05$, and $\sigma = 0.2$. The right plot of Figure 1 shows the resulting MSE of each of the algorithms as a function of the sparsity of observations p, where we set $m = 500$, $n = 200$, and $\sigma = 0.2$. SoftImpute performed significantly worse by an order of magnitude, such that it is not displayed in the right plot. For each combination of parameters, we generate 10 datasets, each of which is then given to each of the algorithms we benchmark. The line plot shows the MSE averaged over the 10 sampled datasets, and the shaded region shows the standard deviation of the resulting MSE from these 10 datasets. Our algorithm performs well across all values of n, matching the oracle at small values of n and still performing close to the oracle even at larger values of n. Our algorithm also performs well even at low levels of sparsity, significantly outperforming other benchmarks, and nearly matching the oracle at fairly low sparsity levels.

Our results and simulations show that simple nonparametric nearest neighbor style estimators can outperform low rank methods even in the class of low rank matrices, when the matrix is far from square, and when there is available covariate side information. As a point of reference, when $m = 500$, $n = 200$, the minimum sparsity such that we observe 2 datapoints in each row and column on average would be $p = 0.02$, which is the rough threshold after which there is sufficient information to fit a rank 2 model. Our algorithm performs well at this level of extreme sparsity, outperforming row regression and classical matrix completion. At a sparsity level of $p = 0.05$, we would expect that there is sufficient information to fit a low rank model at roughly $n = 40$. While our algorithm performs well at such low values of n, the matrix completion algorithms do not perform well until significantly higher values of n.

REFERENCES

[1] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[2] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[3] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *The Journal of Machine Learning Research*, vol. 11, pp. 2287–2322, 2010.

[4] S. Chatterjee, "Matrix estimation by universal singular value thresholding," *The Annals of Statistics*, vol. 43, no. 1, pp. 177–214, 2015.

[5] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2057–2078, 2010.

[6] R. Ge, J. D. Lee, and T. Ma, "Matrix completion has no spurious local minimum," in *Advances in Neural Information Processing Systems*, 2016, pp. 2973–2981.

[7] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proceedings of the 45th annual ACM symposium on Theory of computing*. ACM, 2013, pp. 665–674.

[8] C. Borgs, J. Chayes, C. E. Lee, and D. Shah, "Thy friend is my friend: Iterative collaborative filtering for sparse matrix estimation," in *Advances in Neural Information Processing Systems*, 2017, pp. 4715–4726.

[9] C. Borgs, J. T. Chayes, D. Shah, and C. L. Yu, "Iterative collaborative filtering for sparse matrix estimation," *Operations Research*, 2021.

[10] M. Xu, R. Jin, and Z.-H. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2301–2309.

[11] K. Zhong, P. Jain, and I. S. Dhillon, "Efficient matrix sensing using rank-1 gaussian measurements," in *International conference on algorithmic learning theory*. Springer, 2015, pp. 3–18.

[12] K.-Y. Chiang, C.-J. Hsieh, and I. S. Dhillon, "Matrix completion with noisy side information," in *Advances in Neural Information Processing Systems*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 3447–3455. [Online]. Available: http://papers.nips.cc/paper/5940-matrix-completion-with-noisy-side-information.pdf

[13] A. Eftekhari, D. Yang, and M. B. Wakin, "Weighted matrix completion and recovery with prior subspace information," *IEEE Transactions on Information Theory*, vol. 64, no. 6, pp. 4044–4071, 2018.

[14] M. Ghassemi, A. Sarwate, and N. Goela, "Global optimality in inductive matrix completion," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2226–2230.

[15] K.-Y. Chiang, I. S. Dhillon, and C.-J. Hsieh, "Using side information to reliably learn low-rank matrices from missing and corrupted observations," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 3005–3039, 2018.

[16] P. Jain and I. S. Dhillon, "Provable inductive matrix completion," *arXiv preprint arXiv:1306.0626*, 2013.

[17] J. Lu, G. Liang, J. Sun, and J. Bi, "A sparse interactive model for matrix completion with side information," *Advances in Neural Information Processing Systems*, vol. 29, pp. 4071–4079, 2016.

[18] Y. Guo, "Convex co-embedding for matrix completion with predictive side information," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[19] D. Bertsimas and M. L. Li, "Fast exact matrix completion: A unified optimization framework for matrix completion," *Journal of Machine Learning Research*, vol. 21, no. 231, pp. 1–43, 2020.

[20] M. Burkina, I. Nazarov, M. Panov, G. Fedonin, and B. Shirokikh, "Inductive matrix completion with feature selection," *Computational Mathematics and Mathematical Physics*, vol. 61, no. 5, pp. 719–732, 2021.

[21] T. Zhou, H. Shan, A. Banerjee, and G. Sapiro, "Kernelized probabilistic matrix factorization: Exploiting graphs and side information," in *Proceedings of the 2012 SIAM international Conference on Data mining*. SIAM, 2012, pp. 403–414.

[22] V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Matrix completion on graphs," *arXiv preprint arXiv:1408.1717*, 2014.

[23] N. Rao, H.-F. Yu, P. Ravikumar, and I. S. Dhillon, "Collaborative filtering with graph information: Consistency and scalable methods." in *Advances in Neural Information Processing Systems*, vol. 2. Citeseer, 2015, p. 7.

[24] M. Yin, J. Gao, and Z. Lin, "Laplacian regularized low-rank representation and its applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 3, pp. 504–517, 2015.

[25] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," *Advances in Neural Information Processing Systems*, vol. 29, pp. 847–855, 2016.

[26] S. Dong, P.-A. Absil, and K. A. Gallivan, "Preconditioned conjugate gradient algorithms for graph regularized matrix completion." in *ESANN*, 2019.

[27] S. Dong, P.-A. Absil, and K. Gallivan, "Riemannian gradient descent methods for graph-regularized matrix completion," *Linear Algebra and its Applications*, vol. 623, pp. 193–235, 2021.

[28] D. Song, C. E. Lee, Y. Li, and D. Shah, "Blind regression: Nonparametric regression for latent variable models via collaborative filtering," in *Advances in Neural Information Processing Systems*, 2016, pp. 2155–2163.

[29] Y. Li, D. Shah, D. Song, and C. L. Yu, "Nearest neighbors for matrix estimation interpreted as blind regression for latent variable model," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1760–1784, March 2020.

[30] C. Gao, Y. Lu, and H. H. Zhou, "Rate-optimal graphon estimation," *The Annals of Statistics*, vol. 43, no. 6, pp. 2624–2652, 2015.

[31] C. Gao, Y. Lu, Z. Ma, and H. H. Zhou, "Optimal estimation and completion of matrices with biclustering structures," *Journal of Machine Learning Research*, vol. 17, no. 161, pp. 1–29, 2016.

[32] O. Klopp, A. B. Tsybakov, and N. Verzelen, "Oracle inequalities for network models and sparse graphon estimation," *Annals of Statistics*, 2015.

[33] J. Xu, "Rates of convergence of spectral methods for graphon estimation," *arXiv preprint arXiv:1709.03183*, 2017.

[34] A. B. Tsybakov, "Introduction to nonparametric estimation. revised and extended from the 2004 french original. translated by vladimir zaiats," 2009.

[35] O. Lepski, A. Nemirovski, and V. Spokoiny, "On estimation of the l_r norm of a regression function," *Probability theory and related fields*, vol. 113, no. 2, pp. 221–253, 1999.

[36] M. Udell and A. Townsend, "Why are big data matrices approximately low rank?" *SIAM Journal on Mathematics of Data Science*, vol. 1, no. 1, pp. 144–160, 2019.