Statistica Sinica:

A Zero-Imputation Approach in Recommendation Systems with Data Missing Heterogeneously

Jiashen Lu, Kehui Chen

University of Pittsburgh

Abstract: One of the main goals of recommendation systems is to predict unobserved ratings. The majority of existing methods often implicitly assume that all entries are missing at random and homogeneous, i.e., ratings are revealed with the same probability. Studies show that this assumption is often too strong in real data applications. In this article, we propose a Zero-imputation method to solve the prediction problems under heterogeneous missing situations. Our algorithm has a closed form solution, scalable to large data sets and can be extended to work for the cold start prediction problems, where one needs to predict for a new user or a new item that does not have any prior ratings. We provide theoretical guarantees of the proposed method and demonstrate its good performance in data analysis as well as simulations.

Key words and phrases: Bipartite Graph, Cold Start, Missing Values.

1. Introduction

A recommendation system is often represented by a rating matrix $S \in \mathbb{R}^{n\times m}$ where rows index users and columns index items, and the entries of the matrix correspond to users' ratings for items. Missing is very common in these types of data, i.e., only the ratings to a small portion of items are observed. One of the main goals of recommendation systems is to predict these unobserved missing scores.

Two types of predicting approaches exist in the literature, content-based filtering and collaborative filtering. Content-based filtering recommends items by comparing "key" features of items with users' profile (Lops et al., 2011), which often requires domain knowledge. Collaborative filtering makes use of the observed "collaborative" interaction data to make the predictions. Feuerverger et al. (2012) provides a nice review of some popular approaches. The majority of existing methods and theory in collaborative filtering approach assume or implicitly utilize the setting that missing is at random and homogeneous, i.e., entries are revealed with the same probability, and therefore the main part of the loss function is the average loss over observed entries (Webb, 2006; Paterek, 2007; Koren et al., 2009). Some other methods try to recover the missing ratings under the uniform missing probability assumption in an exact sense, meaning that they treat the observed entries are fixed without measurement errors (Candès and Recht, 2009; Keshavan et al., 2009, 2010; Recht, 2011; Mazumder et al., 2010). However, the probability of missing in recommendation systems are often heterogeneous. For example, those entries with higher underlying ratings may be more likely to be observed (Harper and Konstan, 2015; Marlin and Zemel, 2009). With heterogeneous missing data, averaging over only observed ratings may lead to a bias in approximating the loss function for the complete data (Ma and Chen, 2019; Dai et al., 2019; Schnabel et al., 2016; Wang et al., 2018, 2019; Mao et al., 2021).

Let R denote the missing matrix where $R_{i,j} = 1$ if element $S_{i,j}$ is observed and 0 otherwise, and let Ω be the set of entries that are observed. Homogeneous missing means that $R_{i,j}$ follows a Bernoulli distribution with a constant observation rate. We here assume that $R_{i,j} \sim Ber(O_{i,j})$ and is independent of others given $O_{i,j}$. The complete loss function for a recommendation system takes the form of $\sum_{i=1}^{n} \sum_{j=1}^{m} \mathcal{L}(S_{i,j}, \hat{S}_{i,j})$. In practice, regularization methods and modeling assumptions may be applied to modify the observed loss function $\sum_{(i,j)\in\Omega} \mathcal{L}(S_{i,j}, \hat{S}_{i,j})$ so that it may be close to the full loss function even in the case of heterogeneous missing. For example, Bi et al. (2017) cluster items and users into sub-groups based on their missing patterns and covariate patterns. There are two existing approaches target directly at the full loss function. One is the inverse propensity scoring (IPS) approach (Schnabel et al., 2016; Wang et al., 2019; Imbens and Rubin, 2015). The IPS loss function takes the form of $\sum_{(i,j)\in\Omega} \frac{1}{O_{i,j}} \mathcal{L}(S_{i,j}, \hat{S}_{i,j})$, and is proved to be an unbiased estimate of the full loss function assuming $O_{i,j}$s are known. One known challenge of the IPS approach is that it is not stable when small observation probabilities occur (Rubin, 2001; Schafer and Kang, 2008). Existing works have therefore utilized parametric models, low-rank models or other regularization methods for the estimation of the weighting matrix (Negahban and Wainwright, 2012; Klopp, 2014; Cai et al., 2016; Ma and Chen, 2019; Mao et al., 2021). Another approach is an error-imputation-based (EIB) method, where one estimates the loss $\mathcal{L}(S_{i,j}, \hat{S}_{i,j})$ for unobserved entries (i, j) (Steck, 2010; Wang et al., 2019; Dai et al., 2019). For example, Dai et al. (2019) propose to leveraging information from observed neighbors to impute the errors for missing entries, where the neighborhoods are constructed using user and item networks as well as relevant covariates. All of these methods need to first construct the loss function and iteratively solve optimization problems depending on the specific loss function.

In this paper, we propose a different approach, which we call Zero-imputation. For illustration, let us assume that S is a binary matrix with 1 representing "like", and 0 representing "dislike". We assume that $\mathbb{E}(S_{i,j}) = P_{i,j}$ and the entries are independently formed given $P_{i,j}$. The goal is to estimate $P_{i,j}$ and use that as the prediction for the non-observed entries. Given $O_{i,j}$, $P_{i,j}$ can be estimated by $\frac{\mathbb{E}(S_{i,j}R_{i,j})}{\mathbb{E}(S_{i,j}R_{i,j})+\mathbb{E}((1-S_{i,j})R_{i,j})}$. Although the matrix S is not entirely observable (contains many "NA" values), the matrix $S \circ R$ is available by imputing missing values with 0, and the matrix $(1 - S) \circ R$ can be obtained by first flipping the binary values and then imputing the missing values with 0. Here "$\circ$" denotes the matrix element-wise product (Hadamard product). We then use a soft-thresholding SVD to recover the mean matrix from the binary outcome matrices $S \circ R$ and $(1-S) \circ R$. Predicting ordered scale ratings can be decomposed into several parallel tasks using this binary model. Comparing to existing approaches, the merits of the proposed approach are three-fold. First the proposed approach utilized the "flip" relation of the paired $S \circ R$ and $(1-S) \circ R$ and estimate the inverse weighting matrix as $\mathbb{E}(S \circ R) + \mathbb{E}((1-$

$S) \circ R)$. This provides a self-stabilization and guarantees that the resulting estimate of the probability is between 0 and 1. Second, while most of the IPS methods apply the inverse weighting to the loss function and need an iterative optimization approach, we impute missingness with zero and directly estimate the mean of two fully-observed binary matrix, which can be achieved using a soft-thresholding SVD approach with simple tuning, and end up with a closed form solution. With minimal assumptions, we are able to obtain its rate of convergence for heterogeneous missing cases. Third, the simple form of the Zero-imputation approach naturally extends to the cold start problems, where one needs to predict for a new user or a new item that does not have any prior ratings. Details can be found in Section 3.

In Section 4 and Section 5, we illustrate the proposed approach for predicting unobserved values with heterogeneous missing and new users'/items' ratings using the Movielens data sets and simulated data sets. Theoretical proofs can be found in the Appendix.

2. Zero-imputation approach in predicting order-scaled ratings

Let $S \in \mathbb{R}^{n \times m}$ be the score rating matrix, in which n represents the total number of people and m total number of items. We assume that each entry takes an order-scaled rating in $\{1, 2, \dots, K\}$. The data contains an incomplete matrix S with a large proportion of missing values. Let R denote the data recording matrix where $R_{i,j} = 1$ if element (i, j) is observed and 0 otherwise. We assume that $R_{i,j} \sim Ber(O_{i,j})$ and is independent of others given $O_{i,j}$.

For each $2 \leq k \leq K$, we construct two binary matrices, $A^{(k)}$ and $A_{(k)}$, where the upper matrix $A^{(k)}_{i,j} = 1$ if and only if $S_{i,j}$ is observed and $S_{i,j} \geq k$, and the lower matrix $A_{(k);i,j} = 1$ if and only if $S_{i,j}$ is observed and $S_{i,j} < k$. By definition, $A^{(k)}_{i,j} + A_{(k);i,j} = R_{i,j}$, and in both matrices, the missing values are always imputed with zero. The two matrices have the "flip" relation on observed ratings such that if one matrix is dichotomized as 0 and 1, then the other is dichotomized as 1 and 0. Given missing parameters $O_{i,j}$, for $2 \leq k \leq K$,

$$P(S_{i,j} \geq k) = P(A^{(k)}_{i,j} = 1)/O_{i,j} = \frac{\mathbb{E}(A^{(k)}_{i,j})}{\mathbb{E}(R_{i,j})} = \frac{\mathbb{E}(A^{(k)}_{i,j})}{\mathbb{E}(A^{(k)}_{i,j}) + \mathbb{E}(A_{(k);i,j})}, \tag{2.1}$$

and then we predict the rating using $\mathbb{E}(S_{i,j}) = 1 + \sum_{k=2}^{K} P(S_{i,j} \geq k)$. We call the estimation approach based on Equation (2.1) the Zero-imputation method. We note that the sum of $\mathbb{E}(A^{(k)}_{i,j})$ and $\mathbb{E}(A_{(k);i,j})$ equals $O_{i,j}$. We use Equation (2.1) approach since it provides a self-stabilization and guarantees that the resulting estimate of the probability is between 0 and 1.

Discussion of the missing heterogeneous assumption. Equation (2.1) holds under the assumption that given $O_{i,j}$, $\{R_{i,j}\}$ is independent of $\{S_{i,j}\}$. This is satisfied since $R_{i,j}$ is independently generated from $Ber(O_{i,j})$. Although we require that $R_{i,j}$ is independent of the ratings $S_{i,j}$ given $O_{i,j}$, we allow the underlying missing probability $O_{i,j}$ to freely change over different entries, and may change with $\mathbb{E}(S_{i,j})$ or other parameters. This is much more flexible than the conventional Missing Completely At Random (MCAR) notion. The conventional missing terminologies are mainly developed for parametric settings where one has i.i.d. samples and a set of low dimensional parameters. MCAR will then correspond to a homogeneous missing case where all the data are revealed with the same probability. Here we have relational data with $n \times m$ entries and allow each entry to have its own missing parameter $O_{i,j}$. This kind of completely heterogeneous missing is impossible to estimate in the conventional non-relational data. In the traditional framework of missing data, Missing At Random (MAR) setting is used to relax the MCAR assumption so that the missing probability can vary. In the recommendation systems, researchers found that those entries with higher underlying ratings may be more likely to be observed. Some authors (Marlin and Zemel, 2009; Chi and Li, 2019) tried to use MAR to model this phenomenon where the missing probability is allowed to be different among entries but can only through a function of the observed ratings. Heterogeneous missing is more flexible to accommodate these features in data sets. For example, in our simulations, missing probability $O_{i,j}$ is a decreasing function of the expectation of the observed or unobserved ratings.

At this end, we only need to estimate the mean of a fully-observed binary matrix, i.e., $\mathbb{E}(A^{(k)})$ or $\mathbb{E}(A_{(k)})$. There are well developed methods for this task, which enjoy computational advantages with theoretical guarantee. We choose to apply the soft singular value thresholding approach (Cai et al., 2010; Xu, 2018). The estimation is a modification of matrix SVD, where we replace the original singular values with the soft-thresholded values.

Let $\{\cdot\}_+ = \max\{0, \cdot\}$ be the positive part function. Let $A^{(k)} = \sum_{1 \le i \le (m \wedge n)} \hat{\sigma}_i^k \hat{U}_i^k (\hat{V}_i^k)^T$ be the Singular Value Decomposition (SVD) of matrix $A^{(k)}$ where $\hat{\sigma}_i^k$ is the i-th singular value, $\hat{U}_i^k$ is the corresponding left singular vector, and $\hat{V}_i^k$ is the right singular vector. Similarly let $A_{(k)} = \sum_{1 \le i \le (m \wedge n)} \hat{\sigma}_{k,i} \hat{U}_{k,i} \hat{V}_{k,i}^T$ be the SVD of matrix $A_{(k)}$. We summarize our Zero-imputation method in Algorithm 1.

Algorithm 1 Zero-imputation method for predicting unobserved ratings

Input: Observed S; a dimension p; minimum observation probability $\varepsilon_{n,m}$.

Output: Complete rating matrix $\hat{S}$.

```
Parallel for k in 2,..., $K$ do
    Obtain $A^{(k)}, A_{(k)}$ by truncation and Zero-imputation.
    $A^{(k)} = \sum_{1 \le i \le (m \wedge n)} \hat{\sigma}_i^k \hat{U}_i^k (\hat{V}_i^k)^T$.    ▷ SVD of upper-truncation matrix
    $\hat{A}^{(k)} = \sum_{1 \le i \le (m \wedge n)} \{\hat{\sigma}_i^k - \lambda^k\}_+ \hat{U}_i^k (\hat{V}_i^k)^T$.    ▷ Soft-thresholding using $\lambda^k = \hat{\sigma}_{p+1}^k$
    $A_{(k)} = \sum_{1 \le i \le (m \wedge n)} \hat{\sigma}_{k,i} \hat{U}_{k,i} \hat{V}_{k,i}^T$.    ▷ SVD of lower-truncation matrix
    $\hat{A}_{(k)} = \sum_{1 \le i \le (m \wedge n)} \{\hat{\sigma}_{k,i} - \lambda_k\}_+ \hat{U}_{k,i} \hat{V}_{k,i}^T$.    ▷ Soft-thresholding using $\lambda_k = \hat{\sigma}_{k,p+1}$
end Parallel
$\hat{S}^k = \frac{\hat{A}^{(k)}}{\max\{\hat{A}^{(k)} + \hat{A}_{(k)}, \varepsilon_{n,m}\}}$.    ▷ Scale back
$\hat{S} = 1 + \sum_{k=2}^{K} \hat{S}^k$.    ▷ Prediction
```

Remark 1. Instead of soft-thresholding, one may also use a hard-thresholding method, where one directly cuts off singular values at λ and do not take the differences. Our theoretical results are also valid for the hard-thresholding procedure.

Remark 2. As specified in Theorem 1, to be able to consistently estimate S, we require that the minimum of observation probability $O_{i,j}$ is lower bounded away from zero. In the algorithm one can specify a very small number as the minimum observation probability to stabilize the results in step 8. Also each element in $\hat{A}^{(k)}$ and $\hat{A}_{(k)}$ should be non-negative since it is an estimation of probability. In our numerical results, we used $\varepsilon_{n,m} = 10^{-4}$, and a sensitivity analysis showed that the results are almost identical for $\varepsilon = 10^{-4}, 10^{-5}$ and 10^{-6}. The data is allowed to be more sparse (higher missing rate) as n and m grow, and accordingly the choice of $\varepsilon_{n,m}$ should match the approximate sparsity level of the data.

In the asymptotic theory, one can apply the universal threshold value $\lambda = C_0\sqrt{\delta_{n,m} m \vee n}$, where C_0 is some positive constant greater than 2 and often chosen as 2.01 (Chatterjee et al., 2015) and $\delta_{n,m}$ is the sparsity parameter. In our algorithms, we first use 5-fold cross-validation to choose a thresholding dimension p, and then set the soft-thresholding values to be $\lambda^k = \hat{\sigma}^k_{p+1}$ and $\lambda_k = \hat{\sigma}_{k,p+1}$, where $\hat{\sigma}^k_{p+1}$ and $\hat{\sigma}_{k,p+1}$ are the $(p+1)$-th singular value of $A^{(k)}$ or $A_{(k)}$. We also note that the problem is not assumed to be low-rank; therefore the selected thresholding dimension p could be large. For example, the average value of p is 60 in our simulations with $(n, m) = (3000, 1500)$.

The proposed Zero-imputation algorithm can be decomposed into $2 \times (K - 1)$ parallel tasks because of the independence of each parallel procedure. In each individual task, sparsity matrix appears since we impute all missing values with zero. For large sparse matrices, we can make use of existing tools to efficiently solve the truncated SVD procedure (for example, using the "svds" function in R package RSpectra).

Optional one-step update. We can further improve the Zero-imputation estimator using refinement methods developed for matrix completion. In recommendation systems, common methods such as the regularized SVD (Webb, 2006; Paterek, 2007) usually incorporate ANOVA-type mean correction; therefore we recommend to consider a one-step de-bias approach following the strategy proposed in Chen et al. (2019). Specifically, let $\hat{S}$ be the original Zero-imputation estimation, we may apply the soft singular value thresholding again on the matrix $\hat{S} - \frac{1}{\hat{R}} \circ P_\Omega(\hat{S} - S)$, where $\hat{R}$ is the estimate of the missing matrix and $P_\Omega(B_{i,j}) = B_{i,j}$ if (i, j) is observed and 0 otherwise. The resulting matrix is $\hat{S}_{update}$.

Zero-imputation for continuous ratings. One may directly apply the Zero-imputation approach to $S \in [a, b]$. First scale it into $S^{'} \in [0, 1]$ by subtracting a and then divided by $b - a$. Then Equation (2.1) is modified as

$$\mathbb{E}(S^{'}_{i,j}) = \frac{\mathbb{E}(A^U_{i,j})}{\mathbb{E}(A^U_{i,j}) + \mathbb{E}(A_{L;i,j})},$$

where $A^U_{i,j} = S^{'}_{i,j}$ if observed and 0 otherwise and $A_{L;i,j} = 1 - S^{'}_{i,j}$ if observed and 0 otherwise. The prediction for unobserved values are $\hat{S} = \widehat{\mathbb{E}(S')} \times (b - a) + a$. We focus on working with the binary indicator of $S_{i,j} \geq k$ for two main reasons: first Bernoulli random variables are fully characterized by their expectations, so we can discuss the Bipartite Graph Root Distribution in the cold start problem with minimal assumptions; second, the classification of $S_{i,j}$ at a cut-off value k is often of interest. Our numerical experiments show that directly targeting at $P(S_{i,j} \geq k)$ delivers better classification results.

In the following, we derive the theoretical property of Zero-imputation estimator. In recommendation systems, the observation probabilities $O_{i,j}$ could be very small and produce sparse bipartite graph. It is therefore of interest to set up the asymptotic theorems that can allow sparser graph with growing sample size. To this end, we add a **"sparsity parameter"** $\delta_{n,m}$ to the sampling scheme such that $O_{i,j} = \delta_{n,m}\tilde{O}_{i,j}$, $\mathbb{E}(A^{(k)}) = \delta_{n,m}\tilde{P}^{(k)}$ and $\mathbb{E}(A_{(k)}) = \delta_{n,m}\tilde{P}_{(k)}$, where $\tilde{O}_{i,j}$, $\tilde{P}^{(k)}$, $\tilde{P}_{(k)}$ take values between 0 and 1 and are considered to be at a constant level. In the following, we use $\sigma_i(\tilde{P}^{(k)})$ to denote the i-th singular value of $\tilde{P}^{(k)}$ and use C to denote positive constant values.

Theorem 1. *For results simplicity, we assume $m \leq n$. Let $\hat{S}^k_{i,j}$ be the estimator of $P(S_{i,j} \geq k)$ using the Zero-imputation method mentioned in Algorithm 1. Assume that the sparsity parameter satisfies $\delta_{n,m} \geq C_1 \frac{log(n)}{n}$ and $m\delta_{n,m} \to \infty$ and $min_{i,j}\tilde{O}_{i,j} = \tilde{C}_2 > 0$. For all C_1, there exist C_0, C_2 and C_3 such that if the singular value threshold λ in Algorithm 1 is $C_0\sqrt{\delta_{n,m}n}$ and the lower truncation of observation probability $\varepsilon_{n,m}$ is $C_2\delta_{n,m}$, smaller than $\tilde{C}_2\delta_{n,m}$, then with probability at least $1 - n^{-C_3}$, we have for $2 \leq k \leq K$,*

$$\frac{1}{mn}\sum_{i,j}\left(\hat{S}^k_{i,j} - P(S_{i,j} \geq k)\right)^2 \leq \min_{0 \leq r \leq m}\{\frac{C_4 r}{m\delta_{n,m}} + \frac{C_5}{mn}\sum_{i \geq r+1}\sigma_i^2(\tilde{P}^{(k)})\}. \tag{2.2}$$

Remark 3. The condition $m\delta_{n,m} \to \infty$ is used in other matrix estimation work, such as Theorem 2.1 in Chatterjee et al. (2015), and Theorem 1.1 in Keshavan et al. (2010).

Intuitively, we need the number of observations to be at least in the order of $n \log n$ so that with high probability, each row and column have at least one observation (Candès and Tao, 2010). Under Bernoulli sampling of the set of observed entries, this essentially requires $nm\delta_{n,m}$ to be of order $n \log n$, which implies $m\delta_{n,m} \to \infty$. If m and n are in the same order, the sparsity level can reach the lower bound $\delta_{m,n} = Clog(n)/n$ and the (main term of) convergence rate is $\frac{1}{log(n)}$, which matches the state-of-the-art results in sparse matrix completion.

Remark 4. Theorem 1 provides a general bound to the error. The rate of convergence depends on the structure of the singular values. Corollary 1 and Corollary 2 provide the convergence rates for a finite rank structure and a polynomial decay structure.

Remark 5. The one-step update we mentioned earlier can be shown to have the same general bound with smaller pre-constants. Refer to Theorem 3 in Chen et al. (2019) for relevant discussions.

Xu (2018) and Chatterjee et al. (2015) provided asymptotic results for singular value thresholding approaches for binary matrix completion with a homogeneous observation probability. We modified some of their proofs to prove the above result and the error bound is comparable to Xu (2018) and improved upon Chatterjee et al. (2015). For example, if we assume that the singular values decay in a polynomial rate as $\sigma_r \asymp \frac{\sqrt{mn}}{r^\alpha}$ for some $\alpha > 1$, then the error is in the order of $(\frac{1}{m\delta_{n,m}})^{1-\frac{1}{2\alpha}}$, which slightly improves upon the bounds in Theorem 1.1 in Chatterjee et al. (2015) and is comparable to the bound proved in Corollary 1 in Xu (2018). If the singular values vanish to zero after a finite number, then the error is in the order of $\frac{1}{m\delta_{n,m}}$, which matches the result in Xu (2018). Recall that $\mathbb{E}(S_{i,j}) = 1 + \sum_{k=2}^{K} P(S_{i,j} \geq k)$. For the above mentioned two singular value structures, it is straightforward to prove the following convergence results for $\hat{S}_{i,j} = 1 + \sum_{k=2}^{K} \hat{S}_{i,j}^{k} = 1 + \sum_{k=2}^{K} \hat{P}(S_{i,j} \geq k)$.

Corollary 1. *Given conditions in Theorem 1, if all matrices $\tilde{P}$ has finite rank, then $\frac{1}{mn}\sum_{i,j}(\hat{S}_{i,j} - \mathbb{E}(S_{i,j}))^2 = O_p(\frac{1}{m\delta_{n,m}})$.*

Corollary 2. *Given conditions in Theorem 1, if for all matrices $\tilde{P}$, the singular values decay in a polynomial rate as $\sigma_r \asymp \frac{\sqrt{mn}}{r^\alpha}$ for some $\alpha > 1$, then $\frac{1}{mn}\sum_{i,j}(\hat{S}_{i,j} - \mathbb{E}(S_{i,j}))^2 = O_p((\frac{1}{m\delta_{n,m}})^{1-\frac{1}{2\alpha}})$.*

3. Bipartite Graph Root Distribution (BGRD) and the Cold Start Problem

The cold start problem refers to the problem of predicting the rating for new users or new items where we don't have any observed scores yet. It naturally can be divided into three sub problems: item-cold start, user-cold start and both-cold start. The rating matrix S is then separated into four parts: Old-Old, Old-New, New-Old and New-New, as seen below,

$$S = \begin{array}{c} \\ \text{Old-user} \\ \text{New-user} \end{array} \begin{array}{c} \begin{array}{cc} \text{Old-item} & \text{New-item} \end{array} \\ \left(\begin{array}{cc} S_{(1)} & S_{(2)} \\ S_{(3)} & S_{(4)} \end{array} \right) \end{array}. \tag{3.3}$$

Cold start problem asks to infer the ratings in $S_{(2)}$, $S_{(3)}$, and $S_{(4)}$ given the observations in $S_{(1)}$ and any available covariates of users and items. To efficiently use covariate information to solve the "cold start" problems, we utilize the bipartite graph root distribution (BGRD) theory, which states that each binary matrix, if viewed as an exchangeable random graph, can be generated by first generating independent user latent positions $\{u_i, 1 \leq i \leq n\}$ from a distribution F_1 and independent item latent positions $\{v_j, 1 \leq j \leq m\}$ from a distribution F_2, and then generating the (i, j)-th entry from a Bernoulli distribution with parameter $u_i^T v_j$. Our approach first estimates $\{u_i : 1 \leq i \leq n\}$ and $\{v_j, : 1 \leq j \leq m\}$ from $S_{(1)}$ using the Zero-imputation algorithm and regards these as training data for the bipartite graph root distribution. Then we utilize a nonparametric regression framework to predict the latent positions (u_0, v_0) for a new entry. The last step is to project (u_0, v_0) to the set of weighted summation estimates to ensure that all the resulting inner products $u^T v$ will be between 0 and 1, and satisfy the BGRD requirement. Before we talk about the details of the algorithm, we first state the existence and identifiability of the bipartite graph root distribution, and derive the canonical form of u_i and

v_j. These results are adapted from the graph root distribution developed in Lei (2021) for network data analysis.

Definition 1. Let K be a separable Hilbert space and F_1, F_2 are two probability measures on K. A probability measure $F = F_1 \times F_2$ is called a bipartite graph root distribution (BGRD) if for any two points $u \sim F_1$ and $v \sim F_2$, we have

$$P(u^T v \in [0, 1]) = 1.$$

BGRD is naturally connected to the concept of graphon for a random two-way binary array $A = (A_{i,j})$. The Aldous-Hoover Theorem (Aldous, 1981; Hoover, 1982) says that any separately exchangeable binary array can be generated by first i.i.d. sampling $\{s_i\}$ and $\{t_j\}$ from Uniform $(0, 1)$, then generate $A_{i,j}$ by a Bernoulli distribution with probability $W(s_i, t_j)$ for a graph function (graphon) W: $[0, 1]^2 \to [0, 1]$.

Considering square-integrable graphons $W(s, t) \in L^2([0, 1]^2)$, we have the functional SVD,

$$W(s, t) = \sum_r \lambda_r \phi_r(s) \psi_r(t). \tag{3.4}$$

A graphon W with SVD in Equation (3.4) is said to admit strong decomposition if

$$\sum_r \lambda_r \phi_r^2(s) < \infty, \sum_r \lambda_r \psi_r^2(t) < \infty \quad \text{a.e..}$$

Theorem 2. *(Existence of BGRD) Any exchangeable bipartite random graph generated by a graphon W that admits strong SVD can be generated by a BGRD.*

To avoid ambiguity due to scaling, we restrict ourselves to equally-weighted BGRD.

Definition 2. A BGRD is called equally-weighted if the second moments of u and v are matched, i.e., $\mathbb{E}uu^T = \mathbb{E}vv^T$.

It is clear that an equally-weighted BGRD remains equally-weighted after rotation. To deal with ambiguity due to rotation, we first define the following equivalence class.

Definition 3. We say two equally-weighted BGRDs F and G are equivalent up to orthogonal transforms, written as $F \overset{o.t.}{=} G$, if there is an orthogonal transform Q such that $(u, v) \sim F \Leftrightarrow (Qu, Qv) \sim G$.

Theorem 3. *(Identifiability of BGRD) Two square-integrable equally-weighted BGRDs F and G give the same exchangeable bipartite random graph sampling distribution if and only if $F \overset{o.t.}{=} G$.*

Since all equally-weighted BGRD are identifiable up to a rotation Q, we call a representative in the class canonical if the second moments for u and v are diagonal matrices.

Now for a binary matrix in each parallel step, according to Algorithm 1, the estimate of the underlying probability matrix takes the form $\sum_{1\leq i\leq p}(\hat{\sigma}_i - \lambda)\hat{U}_i\hat{V}_i^T$, where $p = \max\{i : \sigma_i > \lambda\}$. Assume we have n_1 users and m_1 items in $S_{(1)}$, our canonical representation of the latent positions are as follows,

$$\hat{u} = [\hat{u}_1, \ldots, \hat{u}_{n_1}]^T = [\sqrt{\hat{\sigma}_1 - \lambda}\hat{U}_1, \ldots, \sqrt{\hat{\sigma}_p - \lambda}\hat{U}_p] \in \mathbb{R}^{n_1 \times p}, \tag{3.5}$$

and

$$\hat{v} = [\hat{v}_1, \ldots, \hat{v}_{n_1}]^T = [\sqrt{\hat{\sigma}_1 - \lambda}\hat{V}_1, \ldots, \sqrt{\hat{\sigma}_p - \lambda}\hat{V}_p] \in \mathbb{R}^{m_1 \times p}.$$

Each row represents the estimated p dimensional latent position of the user or item. We would like to use the training points and node covariates/attributes to predict the new user and new item's latent positions in each parallel step $2 \leq k \leq K$. We take new users for illustration, and new items' estimation is similar.

Given the estimates for old users $\{\hat{u}_i\}_{i=1}^{n_1}$ and the user's covariate $\{X_i\}_{i=1}^{n}$, where n_1 is the number of old users, the best estimation, in terms of the mean prediction error, for new user's latent position is $\mathbb{E}[u|X]$. According to the definition of conditional expectation, this can be approximated by a weighted version of empirical data, i.e. $\sum_{i=1}^{n_1} w_i u_i$, where the weights $\{w_i\}$ depend on the joint distribution of u and X as well as the marginal distribution of X, and may have a complex form involving all the available data. One observation here is that as long as the estimated latent positions take this weighted summation form, all the resulting inner products $u^T v$ will be between 0 and 1, and satisfy the BGRD requirement. This motivates us to consider the following two-step approach. First, use a nonparametric statistical learning method to estimate u given X, denoting the learned position as u^*. In a second step, we project u^* to the set of weighted estimates. Specifically, we try to find the weighted version that is closest to the learning-based prediction in terms of the link probability.

Recall the notations that $\hat{u} \in \mathbb{R}^{n_1 \times p}$, $\hat{v} \in \mathbb{R}^{m_1 \times p}$ are the estimated latent positions, and $u^* \in \mathbb{R}^{p \times 1}$ is the statistical learning based prediction for a new user. Then the estimated position $\tilde{u} = \hat{u}^T w \in \mathbb{R}^{p \times 1}$ could be obtained by solving the following optimization problem,

$$\min_{\tilde{u}} \ \frac{1}{2}\|\hat{v}\tilde{u} - \hat{v}u^*\|^2 \tag{3.6}$$

$$s.t. \quad \begin{cases} \tilde{u} = \hat{u}^T w \\ \sum_{i=1}^{n_1} w_i = 1 \\ w_i \geq 0 \qquad\qquad (i = 1, \cdots, n_1). \end{cases}$$

The above optimization problem is convex and has a unique solution in terms of $\tilde{u}$, but the constrain set is complex to deal with. Solving Equation (3.6) is equivalent to minimizing $\frac{1}{2}\|\hat{v}\hat{u}^T w - \hat{v}u^*\|^2 + \lambda \mathbb{I}_C\{w\}$ in terms of w, where $\mathbb{I}_C$ is the set indicator function and C stands for the probability simplex. This is a convex optimization problem, and we can apply the Projected Gradient Descent algorithm to solve the above problem by updating weights from iteration t to $t+1$ as $w_{t+1} = \Pi_C(w_t - \eta \nabla g(w_t))$, where Π_C is the projection to simplex operator that can be computed using the algorithm discussed in Wang and Carreira-Perpinán (2013), η is the learning rate and ∇g is the gradient of the quadratic function that appeared in the objective function. While the solution may not be unique in terms of w in the case that $n_1 > m_1$; they still correspond to the unique solution $\tilde{u}$. In our numerical studies, we used the Random Forest method (Breiman, 2001) to predict each dimension in u^*. We do not see a big difference in whether or not the projection step is used, as the random forest output often is very close to a weighted estimator. If some learning methods directly produce a u^* in the form of a weighted summation $\hat{u}^T w$, the projection step is not needed.

We summarize our method for user's cold start rating estimation in Algorithm 2, and the method for new item's or both new can be analogously derived.

Algorithm 2 Zero-imputation method for predicting new users' ratings

```
Input: Observed rating matrix $S_{(1)} \in \mathbb{R}^{n_1 \times m_1}$; a dimension $p$; minimum observation
    probability $\varepsilon_{n_1,m_1}$; covariate matrix $X$.
Output: Predicted rating matrix $\hat{S}_{(3)} \in \mathbb{R}^{n_2 \times m_1}$.
 1: Parallel for k in 2,... $K$ do
 2:     Obtain $A^{(k)}, A_{(k)}$ by truncation and Zero-imputation.
 3:     $A^{(k)} = \sum_{1 \le i \le (m_1 \wedge n_1)} \hat{\sigma}_i^k \hat{U}_i^k (\hat{V}_i^k)^T$.        ▷ SVD of upper-truncation matrix
 4:     Obtain the canonical form of the latent positions $\hat{u}^k$, $\hat{v}^k$ according to Equation (3.5).
 5:     Obtain $u^{k,*} \in \mathbb{R}^{n_2 \times p}$ by multivariate learning methods such as random forests.
 6:     Obtain $\tilde{u}^k \in \mathbb{R}^{n_2 \times p}$ according to Equation (3.6).
 7:     Repeat steps 3-6 for $A_{(k)}$.
 8: end Parallel
 9: $\hat{S}^k_{(3)} = \frac{\tilde{u}^k \hat{v}^{kT}}{\max\{\tilde{u}^k \hat{v}^{kT} + \tilde{u}_k \hat{v}_k^T, \varepsilon_{n,m}\}}$.        ▷ Scale back
10: $\hat{S}_{(3)} = 1 + \sum_{k=2}^{K} \hat{S}^k_{(3)}$.        ▷ Prediction
```

4. Movie-Lens Data Analysis

We use the Movie-lens 100k (ML-100k) and Movie-lens 1M (ML-1M) data sets (`https://grouplens.org/datasets/movielens/`) to illustrate our method. The ML-100k data set contains 100k ratings from 943 users and 1682 movies. Each user has rated at least 20 movies, the overall average rating is 3.53.

For the ML-1M data set, which involves over 1 million rating scores from 6040 users and 3952 movies, the average score is 3.58 and each user has at least 20 ratings. The distributions of the ratings are shown in Figure 1.

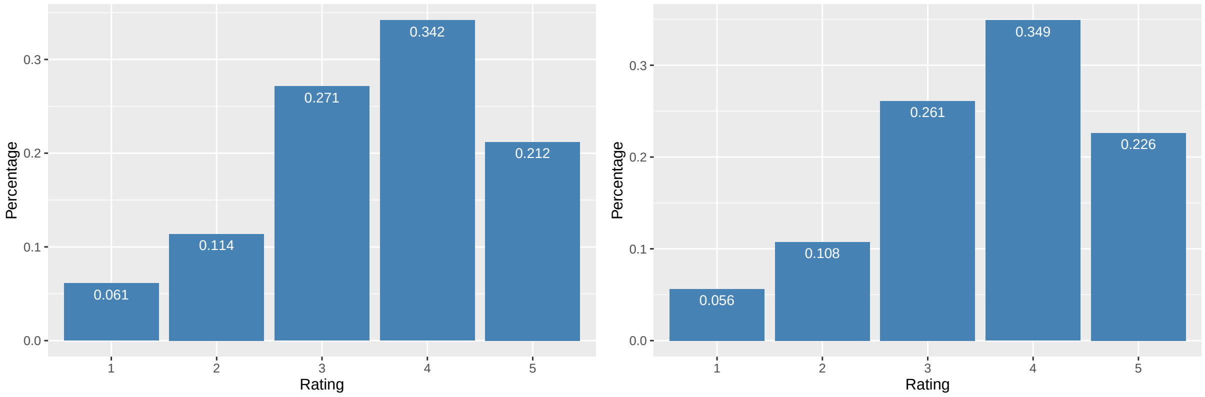

Figure 1: Rating frequency plot for Movie-lens data: ML-100k on the left and ML-1M on the right.

Both data have a large number of missing values with the observation rate about 5%. The missing is suspected to be heterogeneous with higher ratings more likely to be observed (Harper and Konstan, 2015). We heuristically check the missing pattern by regressing the observation probabilities $O_{i,j}$ on the ratings $S_{i,j}$. The observation probabilities are estimated by applying the soft-thresholding SVD method on the binary recording matrix R. Figure 2 shows the estimated observing probability by ratings in the ML-1M data set.

We can see from the graph that the average observation probabilities seem to be higher in higher ratings.

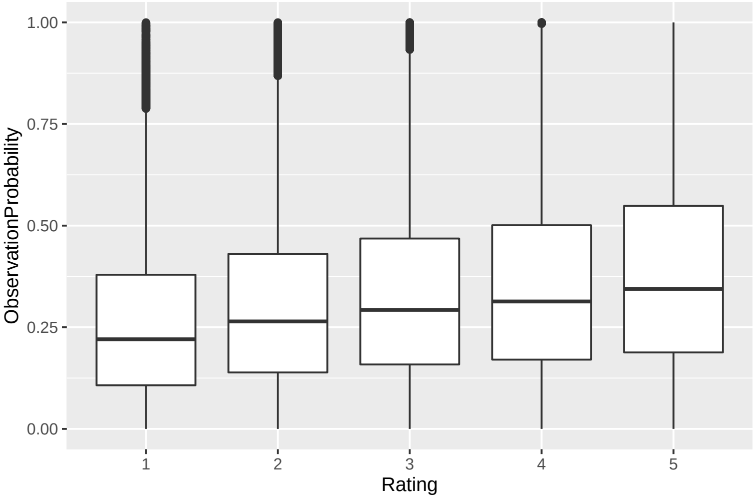

Figure 2: Box plot of the estimated observation probabilities by rating for the ML-1M data.

There are many methods in the literature for predicting unobserved entries in the recommendation systems under homogeneous missing schemes. Based on our knowledge, very few of them may work for heterogeneous missing or for completely cold start problems. As a popular comparison, we include the results of the regularized SVD method

Table 1: Prediction error for unobserved values in the ML-100k and ML-1M data sets. Here “Zero-imputation”, “Zero-imputation-1”, “rSVD”, “gSVD”, “1BITMC-rSVD”, “ItemImpute” and “UserImpute” refer to the proposed method, one-step update of Zero-imputation, regularized SVD (Paterek, 2007), group SVD (Bi et al., 2017), propensity score de-biased rSVD (Ma and Chen, 2019), movie-based mean imputation and user-based mean imputation, respectively.

	ML-100k		ML-1M	
	RMSE	MAE	RMSE	MAE
Zero-imputation	.9246	.7233	.8650	.6774
Zero-imputation-1	.9065	.7213	.8501	.6713
rSVD	.9415	.7355	.8848	.6941
gSVD	.9054	.7112	.8748	.6869
1BITMC-rSVD	.9143	.7197	.8684	.6810
ItemImpute	1.023	.8159	.9799	.7831
UserImpute	1.042	.8336	1.036	.8295

with ANOVA-type mean correction (Webb, 2006; Paterek, 2007), denoted as “rSVD” and implemented through R package rrecsys. This method is originally developed for predicting unobserved entries with homogeneous missing schemes and is popular due to its relatively simple objective function and competitive performance. In view of heterogeneous missing, we include the propensity score adjustment approach as a comparison (Ma and Chen, 2019). In particular, the inverse propensity scores estimated from one bit matrix completion (Davenport et al., 2014) is used as weights for de-biasing the rSVD method, denoted as "1BITMC-rSVD" and implemented based on the public code `https://mdav.ece.gatech.edu/software/`. We also include the results from group-specific SVD (Bi et al., 2017), denoted as "gSVD", and implemented based on the public code `https://sites.google.com/site/xuanbigts/software`. This method utilizes missing patterns and/or users' and items' covariates to create groups and provide more accurate latent positions than rSVD for new users and items. Naive mean imputations based on observed values are also included as baseline comparisons. We denote the one-step update of the Zero-imputation method as "Zero-imputation-1". Methods are tuned as suggested by the original paper to provide best results.

To evaluate the performance, we randomly split the overall observed scores into 90% for training and 10% for testing. The performance is measured by the Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE),

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{M}(\hat{s}_i - s_i)^2}{M}},$$

$$\text{MAE} = \frac{\sum_{i=1}^{M}|\hat{s}_i - s_i|}{M},$$

where $\{s_i\}_{i=1}^{M}$ represent the ratings in the unobserved set (or the new sets in completely cold start problems) and M is the test set size.

Table 2: Prediction error for cold start problems in the ML-100k and ML-1M data sets. Here "Zero-imputation", "rSVD", "gSVD", "1BITMC-rSVD", "MeanImpute" refer to the proposed method, regularized SVD (Paterek, 2007), group SVD (Bi et al., 2017), propensity score de-biased rSVD (Ma and Chen, 2019), and the corresponding mean imputation, respectively.

		Item-Cold		User-Cold		Both-Cold	
		RMSE	MAE	RMSE	MAE	RMSE	MAE
ML-100k	Zero-imputation	.9836	.7724	.9640	.7716	1.038	.8280
	rSVD	1.067	.8618	.9803	.7783	1.097	.9167
	gSVD	1.030	.8227	.9606	.7734	1.066	.8608
	1BITMC-rSVD	1.075	.8779	.9642	.7777	1.105	.9277
	MeanImpute	1.043	.8322	.9645	.7765	1.097	.9165
ML-1M	Zero-imputation	.9324	.7382	.9699	.7727	1.018	.8193
	rSVD	1.090	.9014	.9781	.7811	1.131	.9613
	gSVD	.9998	.8021	.9740	.7799	1.058	.8647
	1BITMC-rSVD	1.103	.9131	.9791	.7877	1.143	.9725
	MeanImpute	1.036	.8313	.9742	.7791	1.117	.9366

Table 3: Classification for scores greater than or equal to 4 versus less than 4. The AUC and overall accuracy are evaluated on the test set.

	ML-100k		ML-1M	
	AUC	Accuracy	AUC	Accuracy
Zero-imputation	.792	.725	.818	.747
rSVD	.700	.703	.731	.737
gSVD	.724	.728	.732	.739
1BITMC-rSVD	.708	.705	.721	.728
ItemImpute	.650	.654	.673	.681
UserImpute	.625	.630	.636	.645

Table 1 records the performance of different methods for the within sample unobserved predictions. We see that the performances of different methods are generally close except for the two mean imputation methods. All of the methods have a better accuracy in the larger data set. The proposed Zero-imputation method, “gSVD” method and ” “1BITMC-rSVD” method produce slightly better results than the “rSVD” method as they account for the heterogeneous missing.

For the completely cold start problem, the public movie-lens data include user covariates named age, gender, and occupation, as well as one item covariate named movie genre. We believe that it is easy to obtain more attributes for movie other than movie genre, such as directors, actors and so on. These covariates contain information of the general popularity and general quality of the movie. To better illustrate the cold start problem, we created two movie covariates to roughly mimic the general popularity and quality. The first is constructed as the total number of ratings of the movie. The second is the total number of ratings above 3. Here "rSVD", "1BITMC-rSVD" are not designed to handle the cold start problem, we simply use the average of the user's/item's sample position estimated from the Old-Old data to predict the new user's or new item's latent positions, and then predict the ratings by the inner product of latent positions. For gSVD, we use 10-means method based on the user/items' covariate to generate the group labels.

We randomly select 10% of users and movies for the cold start sections and use the other 90% in the training. Table 2 summarizes the performance of different methods on the cold start problems in the two data sets. Unsurprisingly, the proposed method and the "gSVD" method perform better than other methods, and the proposed method performs the best overall.

One by-product of the proposed Zero-imputation method is the binary classification of ratings being "good" vs "bad" for any cut value k. We can classify $S \geq 4$ vs $S < 4$ using the estimated $A^{(4)}$. Table 3 displays the classification results of our method as well as the other methods. The proposed Zero-imputation method performs better in terms of AUC and the overall accuracy. The overall accuracy is computed at a cut-off value that the empirical proportions of ones match.

5. Simulations

In this section, we conduct a simulation study, where the data is generated to match the features observed in the Movie-lens data.

We use three different sample sizes, namely, small (1500×800), medium (3000×1500) and large (5000×2500). The small and large cases correspond to the ML-100k and ML-1M sample sizes respectively. We first generate non-missing rating matrix S^0, and a masking procedure R, and then use $S^0 \circ R$ as the observed data.

Following the simulation setting in previous papers, we generate the rating matrix as follows. First generate users' latent positions $\{u_i\}$ from a 12-dimension normal distribution $\mathcal{N}((0.5 \times 1_6, -0.1 \times 1_6)^T, \Sigma)$, where $\Sigma_{i,j} = 0.6^2 I\{i = j\}$. The items' position $\{v_j\}$ are generated by $\mathcal{N}((0.5 \times 1_6, 0.1 \times 1_6)^T, \Sigma)$. Here $S^0_{i,j}$ is generated by first sampling from $\mathcal{N}(u_i^T v_j, 0.6^2)$, then clipping it into the interval [1,5], and finally rounding the number into the nearest integer in {1,2,3,4,5}. We consider a heterogeneous missing scenario where we have a higher chance to observe a higher score. The observed probability that were used to generate R is $(0.022, 0.02, 0.02, 0.05, 0.1)^T$ for scores 1 to 5 respectively. The RMSE and MAE are evaluated on all unobserved entries and averaged over 50 simulations. Regarding the computational time, for $(n, m) = (5000, 2500)$, one single simulation for the proposed method takes 6.3 seconds, the "rSVD" method takes 1.6 seconds, the "gSVD" method runs more than 20 seconds, and "1BIT-rSVD" takes more than 6 minutes. These values include the time used for tuning parameter selections. While "rSVD" method is the fastest, it does not have a special treatment for the heterogeneous missing scheme, and produces a larger error in both data analysis and simulations. The results are run on a PC with 8-core Intel Core i7-10700F processor and 32GB RAM.

For the cold start problems, we create two covariates. The first one is the average of the first six latent dimensions of u/v and the second covariate is a normal nuisance variable $\mathcal{N}(0, 0.6^2)$.

Table 4: Prediction error for unobserved values with heterogeneous missing in the simulated data (the number in the parenthesis is the standard deviation).

	(1500,800)		(3000,1500)		(5000,2500)	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Zero-imputation	.9954(.013)	.8017(.011)	.9421(.006)	.7536(.006)	.8890(.004)	.7082(.003)
Zero-imputation-1	.9750(.012)	.7420(.014)	.9197(.005)	.7048(.006)	.8555(.004)	.6566(.005)
rSVD	1.004(.038)	.7645(.050)	.9808(.032)	.7444(.045)	.9630(.019)	.7304(.032)
gSVD	.9847(.011)	.7703(.010)	.9649(.006)	.7347(.006)	.9356(.004)	.7146(.004)
1BITMC-rSVD	1.002(.010)	.7937(.011)	.9790(.006)	.7752(.015)	.8748(.011)	.6667(.010)
ItemImpute	1.151(.016)	.9249(.015)	1.143(.009)	.9220(.009)	1.141(.006)	.9207(.006)
UserImpute	1.167(.017)	.9331(.016)	1.151(.011)	.9255(.010)	1.147(.006)	.9241(.006)

Table 4 shows the result for the unobserved entries and Table 5 shows the result for the cold start problem with sample size $(n, m) = (5000, 2500)$. The results for other

Table 5: Prediction error for cold start problems in the simulated data with sample size $(n, m) = (5000, 2500)$ (the number in the parenthesis is the standard deviation).

	Item-Cold		User-Cold		Both-Cold	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
Zero-imputation	.9813(.023)	.7582(.023)	.9680(.017)	.7475(.017)	.9772(.026)	.7646(.027)
rSVD	1.101(.020)	.8927(.029)	1.089(.025)	.8854(.033)	1.184(.033)	.9907(.041)
gSVD	1.018 (.018)	.8015(.016)	1.008(.018)	.7959(.017)	1.058(.029)	.8571(.028)
1BITMC-rSVD	1.082(.016)	.8804(.012)	1.077(.018)	.8762(.014)	1.176(.031)	.9709(.023)
MeanImpute	1.151(.014)	.9283(.013)	1.144(.016)	.9237(.013)	1.254(.020)	1.123(.019)

sample sizes show similar pattern. The results are consistent to what we see in the Movie-lens data. All of the methods have reasonable performances for unobserved entry prediction and improve as the sample size grows. Comparing to "Zero-imputation", "Zero-imputation-1", "gSVD" and "1BITMC-rSVD", the "rSVD" method does not account for the heterogeneous missing and shows larger error and larger variation. The one-step update for the Zero-imputation method outperforms all other methods. The proposed method and the "gSVD" method work reasonably well for the cold start predictions. We see that the proposed method shows a sharper improvement in larger data sets and in cold start problems.

Supplementary Materials

Contain proofs of Theorem 1, Corollary 2, Theorem 2, and Theorem 3.

References

Aldous, D. J. (1981), "Representations for partially exchangeable arrays of random variables," *Journal of Multivariate Analysis*, 11, 581–598.

Bi, X., Qu, A., Wang, J., and Shen, X. (2017), "A group-specific recommender system," *Journal of the American Statistical Association*, 112, 1344–1353.

Breiman, L. (2001), "Random forests," *Machine learning*, 45, 5–32.

Cai, J.-F., Candès, E. J., and Shen, Z. (2010), "A singular value thresholding algorithm for matrix completion," *SIAM Journal on optimization*, 20, 1956–1982.

Cai, T., Cai, T. T., and Zhang, A. (2016), "Structured matrix completion with applications to genomic data integration," *Journal of the American Statistical Association*, 111, 621–633.

Candès, E. J. and Recht, B. (2009), "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, 9, 717–772.

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

Chatterjee, S. et al. (2015), “Matrix estimation by universal singular value thresholding,” *Annals of Statistics*, 43, 177–214.

Chen, Y., Fan, J., Ma, C., and Yan, Y. (2019), “Inference and uncertainty quantification for noisy matrix completion,” *Proceedings of the National Academy of Sciences*, 116, 22931–22937.

Chi, E. C. and Li, T. (2019), “Matrix completion from a computational statistics perspective,” *Wiley Interdisciplinary Reviews: Computational Statistics*, 11, e1469.

Dai, B., Wang, J., Shen, X., and Qu, A. (2019), “Smooth neighborhood recommender systems,” *The Journal of Machine Learning Research*, 20, 589–612.

Davenport, M. A., Plan, Y., Van Den Berg, E., and Wootters, M. (2014), “1-bit matrix completion,” *Information and Inference: A Journal of the IMA*, 3, 189–223.

Feuerverger, A., He, Y., and Khatri, S. (2012), “Statistical significance of the Netflix challenge,” *Statistical Science*, 27, 202–231.

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

Hoover, D. N. (1982), “Row-column exchangeability and a generalized model for probability,” *Exchangeability in probability and statistics (Rome, 1981)*, 281–291.

Imbens, G. W. and Rubin, D. B. (2015), *Causal inference in statistics, social, and biomedical sciences*, Cambridge University Press.

Keshavan, R., Montanari, A., and Oh, S. (2009), "Matrix completion from noisy entries," *Advances in neural information processing systems*, 22.

Keshavan, R. H., Montanari, A., and Oh, S. (2010), "Matrix completion from a few entries," *IEEE transactions on information theory*, 56, 2980–2998.

Klopp, O. (2014), "Noisy low-rank matrix completion with general sampling distribution," *Bernoulli*, 20, 282–303.

Koren, Y., Bell, R., and Volinsky, C. (2009), "Matrix factorization techniques for recommender systems," *Computer*, 42, 30–37.

Lei, J. (2021), "Network representation using graph root distributions," *The Annals of Statistics*, 49, 745–768.

Lops, P., De Gemmis, M., and Semeraro, G. (2011), "Content-based recommender systems: State of the art and trends," *Recommender systems handbook*, 73–105.

Ma, W. and Chen, G. H. (2019), "Missing not at random in matrix completion: The effectiveness of estimating missingness probabilities under a low nuclear norm assumption," *Advances in Neural Information Processing Systems*, 32, 14900–14909.

Mao, X., Wong, R. K., and Chen, S. X. (2021), "Matrix Completion under Low-Rank Missing Mechanism," *Statistica Sinica*, 31, 1–26.

Marlin, B. M. and Zemel, R. S. (2009), "Collaborative prediction and ranking with non-random missing data," in *Proceedings of the third ACM conference on Recommender systems*, pp. 5–12.

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

Negahban, S. and Wainwright, M. J. (2012), "Restricted strong convexity and weighted matrix completion: Optimal bounds with noise," *The Journal of Machine Learning Research*, 13, 1665–1697.

Paterek, A. (2007), "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and workshop*, vol. 2007, pp. 5–8.

Recht, B. (2011), "A simpler approach to matrix completion." *Journal of Machine Learning Research*, 12.

Rubin, D. B. (2001), "Using propensity scores to help design observational studies: application to the tobacco litigation," *Health Services and Outcomes Research Methodology*, 2, 169–188.

Schafer, J. L. and Kang, J. (2008), "Average causal effects from nonrandomized studies: a practical guide and simulated example." *Psychological methods*, 13, 279.

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. (2016), "Recommendations as treatments: Debiasing learning and evaluation," in *international conference on machine learning*, PMLR, pp. 1670–1679.

Steck, H. (2010), "Training and testing of recommender systems on data missing not at random," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 713–722.

Wang, M., Gong, M., Zheng, X., and Zhang, K. (2018), "Modeling dynamic missingness of implicit feedback for recommendation," *Advances in neural information processing systems*, 31, 6669.

Wang, W. and Carreira-Perpinán, M. A. (2013), "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application," *arXiv preprint arXiv:1309.1541*.

Wang, X., Zhang, R., Sun, Y., and Qi, J. (2019), "Doubly robust joint learning for recommendation on data missing not at random," in *International Conference on Machine Learning*, PMLR, pp. 6638–6647.

Webb, B. (2006), "Netflix update: Try this at home," *Blog post sifter. org/simon/journal/20061211. html*.

Xu, J. (2018), "Rates of convergence of spectral methods for graphon estimation," in *International Conference on Machine Learning*, PMLR, pp. 5433–5442.

Department of Statistics, University of Pittsburgh, PA, 15260

E-mail: jil235@pitt.edu, khchen@pitt.edu