Statistica Sinica **34** (2024), 421-438
doi:https://doi.org/10.5705/ss.202021.0429

# A ZERO-IMPUTATION APPROACH IN RECOMMENDATION SYSTEMS WITH DATA MISSING HETEROGENEOUSLY

Jiashen Lu* and Kehui Chen

*University of Pittsburgh*

*Abstract:* One of the main goals of recommendation systems is to predict unobserved ratings. The majority of existing methods implicitly assume that all entries are missing at random and homogeneous, that is, the ratings are revealed with the same probability. However, studies have shown that this assumption is often too strong in real-data applications. We propose a zero-imputation method for solving prediction problems under heterogeneous missing situations. Our algorithm has a closed-form solution, is scalable to large data sets, and can be extended to include the cold-start prediction problems, where one needs a prediction for a new user or item with no prior ratings. We provide theoretical guarantees for the proposed method and demonstrate its good performance in a data analysis and in 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 the users' ratings for the items. One of the main goals of a recommendation system is to predict unobserved missing ratings in this matrix.

Two approaches to such predictions exist in the literature, namely, content-based filtering and collaborative filtering. Content-based filtering recommends items by comparing "key" features of the items with the users' profiles (Lops, De Gemmis and Semeraro (2011)), which often requires domain knowledge. Collaborative filtering uses observed "collaborative" interaction data to make predictions; see Feuerverger, He and Khatri (2012) for a review of some popular approaches. The majority of existing methods and theory in the collaborative filtering approach assume or implicitly use the setting that missing is at random and homogeneous, that is, entries are revealed with the same probability. Therefore, the main part of the loss function is the average loss over the observed entries (Webb (2006); Paterek (2007); Koren, Bell and Volinsky (2009)). Other methods try to recover missing ratings by assuming a uniform missing probability, in an exact sense, meaning that they treat the observed entries as fixed and

*Corresponding author.

without measurement errors (Candès and Recht (2009); Keshavan, Montanari and Oh (2010, 2009); Recht (2011); Mazumder, Hastie and Tibshirani (2010)). However, the probability of missing values in recommendation systems is often heterogeneous. For example, 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 observed ratings only may lead to bias when approximating the loss function for the complete data set (Ma and Chen (2019); Dai et al. (2019); Schnabel et al. (2016); Wang et al. (2018, 2019); Mao, Wong and Chen (2021)).

Let $R$ denote the missing matrix, where $R_{i,j} = 1$ if element $S_{i,j}$ is observed, and zero otherwise, and let $\Omega$ be the set of entries that are observed. Homogeneous missing data means that $R_{i,j}$ follows a Bernoulli distribution with a constant observation rate. Here, we 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, researchers may apply regularization methods and modeling assumptions to modify the observed loss function $\sum_{(i,j)\in\Omega}\mathcal{L}(S_{i,j},\hat{S}_{i,j})$ to be close to the full loss function, even in the case of heterogeneous missing data. For example, Bi et al. (2017) cluster items and users into sub-groups based on their missing patterns and covariate patterns. Two existing approaches directly target the full loss function. The first is the inverse propensity scoring (IPS) approach (Schnabel et al. (2016); Wang et al. (2019); Imbens and Rubin (2015)), which has a IPS loss function of the form $\sum_{(i,j)\in\Omega}(1/O_{i,j})\mathcal{L}(S_{i,j},\hat{S}_{i,j})$, and has been proved to be an unbiased estimate of the full loss function, assuming $O_{i,j}$ are known. A disadvantage of the IPS approach is that it is not stable when small observation probabilities occur (Rubin (2001); Schafer and Kang (2008)). As a result, parametric models, low-rank models, and other regularization methods have been used to estimate the weighting matrix (Negahban and Wainwright (2012); Klopp (2014); Cai, Cai and Zhang (2016); Ma and Chen (2019); Mao, Wong and Chen (2021)). The second approach is the 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 leveraging information from observed neighbors to impute the missing entries, where neighborhoods are constructed using user and item networks, as well as relevant covariates. These methods all first construct a loss function, and then iteratively solve optimization problems using this 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 one representing "like," and zero representing "dislike." We assume that $\mathbb{E}(S_{i,j}) = P_{i,j}$, and that the entries are formed independently, given $P_{i,j}$. The goal is to estimate $P_{i,j}$, and use it as a prediction for the non-observed entries. Given $O_{i,j}$, we estimate $P_{i,j}$ as $\mathbb{E}(S_{i,j}R_{i,j})/[\mathbb{E}(S_{i,j}R_{i,j}) + \mathbb{E}\{(1-S_{i,j})R_{i,j}\}]$. Although the matrix $S$ is not

entirely observable (i.e., it contains many "NA" values), the matrix $S \circ R$ can be obtained by imputing missing values with zero, and the matrix $(1 - S) \circ R$ can be obtained by first flipping the binary values, and then imputing the missing values with zero. Here, "$\circ$" denotes the matrix element-wise product (Hadamard product). We then use a soft-thresholding singular value decomposition (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. The proposed approach has three advantages over existing approaches. First, it uses the "flip" relation of the paired $S \circ R$ and $(1-S)\circ R$, and estimates the inverse weighting matrix as $\mathbb{E}(S\circ R)+\mathbb{E}\{(1-S)\circ R\}$. This provides self-stabilization and guarantees that the resulting estimate of the probability is between zero and one. Second, most IPS methods apply the inverse weighting to the loss function and need an iterative optimization approach. In contrast, we impute missingness with zero, and directly estimate the mean of two fully observed binary matrices. This can be achieved using a soft-thresholding SVD approach with simple tuning, and results in a closed-form solution. Using minimal assumptions, we can obtain its rate of convergence for heterogeneous missing cases. Third, the simple form of the zero-imputation approach extends naturally to cold-start problems, where we need a prediction for a new user or item with no prior ratings. Details can be found in Section 3.

In Section 4 and 5, we demonstrate the proposed approach by predicting heterogeneous unobserved values and new users' ratings using the Movie-lens and simulated data sets, respectively. All theoretical proofs can be found in the online Supplementary Material.

## 2. Zero-Imputation Approach for Predicting Order-Scaled Ratings

Let $S \in \mathbb{R}^{n\times m}$ be a score rating matrix, where $n$ represents the total number of people, and $m$ is the total number of items. We assume that each entry takes an order-scaled rating in $\{1, 2, \ldots, K\}$. The data contain 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 zero otherwise. We assume that $R_{i,j} \sim Ber(O_{i,j})$ and is independent of others, given $O_{i,j}$.

For each $2 \le k \le K$, 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 $S_{i,j} \ge k$, 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 a "flip" relation on observed ratings, such that if one matrix is dichotomized as zero and one, then the other is dichotomized as one and zero. Given the missing parameters $O_{i,j}$, for $2 \le k \le K$,

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

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 (2.1) the zero-imputation method. Note that the sum of $\mathbb{E}(A^{(k)}_{i,j})$ and $\mathbb{E}(A_{(k);i,j})$ is equal to $O_{i,j}$. We use (2.1), because it provides self-stabilization and guarantees that the resulting estimate of the probability is between zero and one.

**Discussion of the missing heterogeneous assumption.** (2.1) holds under the assumption that given $O_{i,j}$, $\{R_{i,j}\}$ is independent of $\{S_{i,j}\}$. This is satisfied because $R_{i,j}$ is generated independently 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 that have independent and identically distributed (i.i.d.) samples and a set of low-dimensional parameters. MCAR then corresponds to the homogeneous missing case in which all 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 missingness is impossible to estimate in conventional nonrelational data. In the traditional framework of missing data, the missing at random (MAR) setting is used to relax the MCAR assumption, so that the missing probability can vary. In recommendation systems, researchers have found that entries with higher underlying ratings may be more likely to be observed. Marlin and Zemel (2009); Chi and Li (2019) use MAR to model this phenomenon, where the missing probability is allowed to vary among entries, but only through a function of the observed ratings. Heterogeneous missingness is more flexible in terms of accommodating these features in data sets. For example, in our simulations, the missing probability $O_{i,j}$ is a decreasing function of the expectation of the observed or unobserved ratings.

Now, we need only to estimate the mean of a fully observed binary matrix, that is, $\mathbb{E}(A^{(k)})$ or $\mathbb{E}(A_{(k)})$. There are numerous methods for this task that enjoy computational advantages with a theoretical guarantee. We choose to apply the soft singular value thresholding approach (Cai, Candès and Shen (2010); Xu (2018)). The estimation is a modification of a 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 $A^{(k)} = \sum_{1 \leq i \leq (m \wedge n)} \hat{\sigma}^k_i \hat{U}^k_i (\hat{V}^k_i)^T$ be the SVD of the matrix $A^{(k)}$, where $\hat{\sigma}^k_i$ is the $i$th singular value, $\hat{U}^k_i$ is the corresponding left-singular vector, and $\hat{V}^k_i$ is the right-singular vector. Similarly, let $A_{(k)} = \sum_{1 \leq i \leq (m \wedge n)} \hat{\sigma}_{k,i} \hat{U}_{k,i} \hat{V}^T_{k,i}$ 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 $\hat{S}$.

1: **Parallel for k** in $2,\ldots,K$ **do**
2: Obtain $A^{(k)}, A_{(k)}$ by using truncation and the zero-imputation approach.
3: $A^{(k)} = \sum_{1\le i\le(m\wedge n)} \hat{\sigma}_i^k \hat{U}_i^k (\hat{V}_i^k)^T$. ▷ SVD of upper-truncation matrix
4: $\hat{A}^{(k)} = \sum_{1\le i\le(m\wedge n)} (\hat{\sigma}_i^k - \lambda^k)_+ \hat{U}_i^k (\hat{V}_i^k)^T$. ▷ Soft-thresholding using $\lambda^k = \hat{\sigma}_{p+1}^k$
5: $A_{(k)} = \sum_{1\le i\le(m\wedge n)} \hat{\sigma}_{k,i} \hat{U}_{k,i} \hat{V}_{k,i}^T$. ▷ SVD of lower-truncation matrix
6: $\hat{A}_{(k)} = \sum_{1\le i\le(m\wedge n)} (\hat{\sigma}_{k,i} - \lambda_k)_+ \hat{U}_{k,i} \hat{V}_{k,i}^T$. ▷ Soft-thresholding using $\lambda_k = \hat{\sigma}_{k,p+1}$
7: **end Parallel**
8: $\hat{S}^k = \hat{A}^{(k)} / \max(\hat{A}^{(k)} + \hat{A}_{(k)}, \varepsilon_{n,m})$. ▷ Scale back
9: $\hat{S} = 1 + \sum_{k=2}^K \hat{S}^k$. ▷ Prediction

**Remark 1.** Instead of soft-thresholding, we may use a hard-thresholding method, where we cut off singular values directly at $\lambda$, and do not take the differences. Our theoretical results remain valid in this case.

**Remark 2.** As specified in Theorem 1, to be able to consistently estimate $S$, we require that the minimum of the observation probability $O_{i,j}$ be 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. In addition, each element in $\hat{A}^{(k)}$ and $\hat{A}_{(k)}$ should be nonnegative, because it is an estimation of a probability. In our numerical results, we use $\varepsilon_{n,m} = 10^{-4}$. A sensitivity analysis showed that the results are almost identical for $\varepsilon = 10^{-4}, 10^{-5}$, and $10^{-6}$. The data are allowed to be more sparse (i.e., have a 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 two, often chosen as 2.01 (Chatterjee and others (2015)), and $\delta_{n,m}$ is the sparsity parameter. In our algorithms, we first use five-fold cross-validation to choose a thresholding dimension $p$, and then set the soft-thresholding values to $\lambda^k = \hat{\sigma}_{p+1}^k$ and $\lambda_k = \hat{\sigma}_{k,p+1}$, where $\hat{\sigma}_{p+1}^k$ and $\hat{\sigma}_{k,p+1}$ are the $(p+1)$th singular values of $A^{(k)}$ and $A_{(k)}$, respectively. Note too that the problem is not assumed to be low rank; therefore, the selected thresholding dimension $p$ can 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. A sparsity matrix appears for each individual task, because we impute all missing values with zero. For large sparse matrices, we can use existing tools to efficiently solve the truncated SVD procedure (e.g., the "svds" function in the R package RSpectra).

**Optional one-step update.** We can further improve the zero-imputation estimator by using refinement methods developed for matrix completion. In recommendation systems, common methods such as regularized SVD (Webb (2006); Paterek (2007)) usually incorporate an ANOVA-type mean correction. Therefore we recommend considering a one-step debiased 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 $\hat{S} - 1/\hat{R} \circ P_\Omega(\hat{S} - S)$, 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 zero otherwise. The resulting matrix is $\hat{S}_{update}$.

**Zero-imputation for continuous ratings.** One may apply the zero-imputation approach directly to $S \in [a, b]$. First, scale it to $S^{'} \in [0, 1]$ by subtracting $a$ and then dividing by $b - a$. Then, (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 zero otherwise, and $A_{L;i,j} = 1 - S^{'}_{i,j}$ if observed, and zero otherwise. The prediction for unobserved values is $\hat{S} = \widehat{\mathbb{E}(S')} \times (b-a) + a$. 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. Therefore, we can discuss the bipartite graph root distribution (BGRD) for 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 targeting $P(S_{i,j} \geq k)$ directly delivers better classification results.

In the following, we derive the theoretical properties of the zero-imputation estimator. In recommendation systems, the observation probabilities $O_{i,j}$ can be very small, resulting in a sparse bipartite graph. Therefore, it is of interest to set up asymptotic theorems that that allow the sparsity of a graph to increase with the 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 $\tilde{O}_{i,j}$, $\tilde{P}^{(k)}$, and $\tilde{P}_{(k)}$ take values between zero and one, 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 a positive constant value.

**Theorem 1.** *For 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 given in Algorithm 1. Assume that the sparsity parameter satisfies $\delta_{n,m} \geq C_1 log(n)/n$, and that $m\delta_{n,m} \to \infty$ 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 $\lambda$ in Algorithm 1 is $C_0\sqrt{\delta_{n,m}n}$ and the lower truncation of the observation probability $\varepsilon_{n,m}$ is $C_2\delta_{n,m}$, which is smaller than $\tilde{C}_2\delta_{n,m}$, then*

*with probability at least* $1 - n^{-C_3}$*, we have for* $2 \le k \le K$*,*

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

**Remark 3.** The condition $m\delta_{n,m} \to \infty$ is used in other matrix estimation work, such as Theorem 2.1 in Chatterjee and others (2015), and Theorem 1.1 in Keshavan, Montanari and Oh (2009). 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 of 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 $1/log(n)$, which matches the state-of-the-art results in sparse matrix completion.

**Remark 4.** Theorem 1 provides a general bound for 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, respectively.

**Remark 5.** The one-step update mentioned earlier can be shown to have the same general bound, with smaller pre-constants; see Theorem 3 in Chen et al. (2019) for a relevant discussion.

Xu (2018) and Chatterjee and others (2015) provide asymptotic results for singular value thresholding approaches for binary matrix completion with a homogeneous observation probability. We modify some of their proofs to prove the above result. The error bound is comparable with that of Xu (2018) and improves on that of Chatterjee and others (2015). For example, if we assume that the singular values decay at a polynomial rate $\sigma_r \asymp \sqrt{mn}/r^\alpha$, for some $\alpha > 1$, then the error is in the order of $\{1/(m\delta_{n,m})\}^{1-1/(2\alpha)}$, which improves slightly on the bounds in Theorem 1.1 in Chatterjee and others (2015), and is comparable with 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 $1/(m\delta_{n,m})$, which matches the result in Xu (2018). Recall that $\mathbb{E}(S_{i,j}) = 1 + \sum_{k=2}^K P(S_{i,j} \ge k)$. For the above-mentioned two singular value structures, it is straightforward to prove the following convergence results for $\hat{S}_{i,j} = 1 + \sum_{k=2}^K \hat{S}^k_{i,j} = 1 + \sum_{k=2}^K \hat{P}(S_{i,j} \ge k)$.

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

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

## 3. The BGRD and the Cold-Start Problem

The cold-start problem refers to predicting the rating for a new user or item when we do not yet have any observed scores. The problem 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:

$$S = \begin{array}{c} \\ \text{Old-user} \\ \text{New-user} \end{array} \begin{array}{c} \begin{array}{cc} \text{Old-item} & \text{New-item} \end{array} \\ \begin{pmatrix} S_{(1)} & S_{(2)} \\ S_{(3)} & S_{(4)} \end{pmatrix} \end{array}. \tag{3.1}$$

The cold-start problem asks that we 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 use the covariate information efficiently to solve cold-start problems, we use BGRD theory. This theory states that each binary matrix, if viewed as an exchangeable random graph, can be generated by first generating independent user latent positions $\{u_i, 1 \le i \le n\}$ from a distribution $F_1$ and independent item latent positions $\{v_j, 1 \le j \le m\}$ 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 $\{u_i : 1 \le i \le n\}$ and $\{v_j, : 1 \le j \le m\}$ from $S_{(1)}$ using the zero-imputation algorithm, and regards these as training data for the BGRD. Then, we use a nonparametric regression framework to predict the latent positions $(u_0, v_0)$ for a new entry. The last step projects $(u_0, v_0)$ onto the set of weighted summation estimates to ensure that all the resulting inner products $u^T v$ are between zero and one, and satisfy the BGRD requirement. Before discussing the algorithm, we first state the existence and identifiability of the BGRD, 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 $F_1$ and $F_2$ be two probability measures on $K$. A probability measure $F = F_1 \times F_2$ is called a BGRD if, for any two points $u \sim F_1$ and $v \sim F_2$, we have

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

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

$$W(s, t) = \sum_r \lambda_r \phi_r(s) \psi_r(t). \tag{3.2}$$

A graphon $W$ with the SVD in (3.2) 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 a strong SVD can be generated by a BGRD.*

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

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

An equally weighted BGRD clearly remains equally weighted after a 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 $F \overset{o.t.}{=} G$.*

Because all equally weighted BGRDs 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, from Algorithm 1, the estimate of the underlying probability matrix takes the form $\sum_{1 \le i \le p} (\hat{\sigma}_i - \lambda) \hat{U}_i \hat{V}_i^T$, where $p = \max\{i : \sigma_i > \lambda\}$. Assume we have $n_1$ users and $m_1$ items in $S_{(1)}$. Then, our canonical representations of the latent positions are as follows:

$$\hat{u} = [\hat{u}_1, \ldots, \hat{u}_{n_1}]^T = [\sqrt{\hat{\sigma}_1 - \lambda}\hat{U}_1, \ldots, \sqrt{\hat{\sigma}_p - \lambda}\hat{U}_p] \in \mathbb{R}^{n_1 \times p}, \tag{3.3}$$

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 use the training points and the node covariates/attributes to predict the latent positions of a new user and a new item in each parallel step $2 \le k \le K$. Here, we focus on new users. The estimation of new items 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 the 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, that is, $\sum_{i=1}^{n_1} w_i u_i$, where the weights $\{w_i\}$ depend on the joint distribution of $u$ and $X$ and on the marginal distribution of $X$, and may have a complex form involving all the available data. Note that as long as the estimated latent positions take this weighted summation form, all the resulting inner products $u^T v$ will be between zero and one, and satisfy the BGRD requirement. This motivates us to consider the following two-step approach. First, we use a nonparametric statistical learning method to estimate $u$ given $X$, denoting the learned position as $u^*$. In the second step, we project $u^*$ onto 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 that $\hat{u} \in \mathbb{R}^{n_1 \times p}$ and $\hat{v} \in \mathbb{R}^{m_1 \times p}$ 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 $\tilde{u} = \hat{u}^T w \in \mathbb{R}^{p \times 1}$ can be obtained by solving the following optimization problem:

$$\min_{\tilde{u}} \ \frac{1}{2}\|\hat{v}\tilde{u} - \hat{v}u^*\|^2 \qquad (3.4)$$
$$s.t. \quad \begin{cases} \tilde{u} = \hat{u}^T w \\ \sum_{i=1}^{n_1} w_i = 1 \\ w_i \geq 0 \qquad (i = 1, \ldots, n_1). \end{cases}$$

This optimization problem is convex and has a unique solution in terms of $\tilde{u}$, but the constraint set is complex to deal with. Solving (3.4) is equivalent to minimizing $(1/2)\|\hat{v}\hat{u}^T w - \hat{v}u^*\|^2 + \lambda \mathbb{I}_C\{w\}$ in terms of $w$, where $\mathbb{I}_C$ is the set indicator function, and $C$ is the probability simplex. Because this is a convex optimization problem, we can apply the projected gradient descent algorithm to solve it by updating the weights from iteration $t$ to $t+1$ as $w_{t+1} = \Pi_C(w_t - \eta \nabla g(w_t))$, where $\Pi_C$ is the projection to a simplex operator that can be computed using the algorithm in Wang and Carreira-Perpinán (2013), $\eta$ is the learning rate, and $\nabla g$ is the gradient of the quadratic function that appears in the objective function. Although the solution may not be unique in terms of $w$ when $n_1 > m_1$, they still correspond to the unique solution $\tilde{u}$. In our numerical studies, we use the random forest method (Breiman (2001)) to predict each dimension in $u^*$. Using the projection step does not appear to make a big difference, because the random forest output is often very close to a weighted estimator. If a learning method directly produces $u^*$ in the form of a weighted summation $\hat{u}^T w$, the projection step is not needed.

We summarize our method for a user's cold-start rating estimation in Algorithm 2. The method for a new item or when they are both new can be derived analogously.

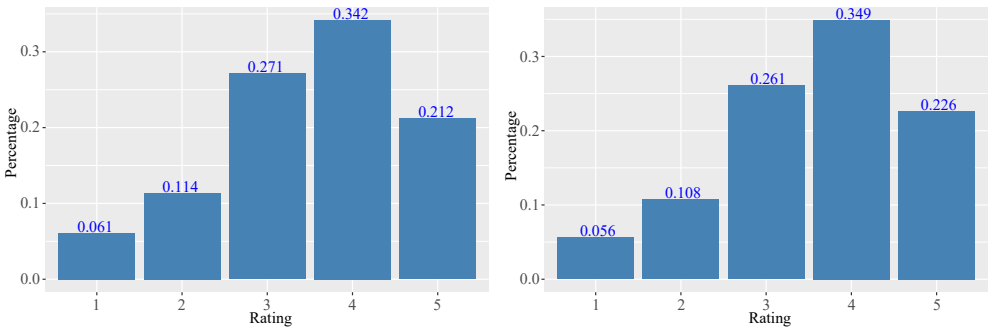


Figure 1. Rating frequency plot for Movie-lens data: ML-100k on the left and ML-1M on the right.

**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 using truncation and the zero-imputation approach.
3: $A^{(k)} = \sum_{1 \le i \le (m_1 \wedge n_1)} \hat{o}_i^k \hat{U}_i^k (\hat{V}_i^k)^T$. ▷ SVD of upper-truncation matrix
4: Obtain the canonical form of the latent positions $\hat{u}^k$, $\hat{v}^k$ from (3.3).
5: Obtain $u^{k,*} \in \mathbb{R}^{n_2 \times p}$ by applying multivariate learning methods, such as random forests.
6: Obtain $\tilde{u}^k \in \mathbb{R}^{n_2 \times p}$ from (3.4).
7: Repeat steps 3–6 for $A_{(k)}$.
8: **end Parallel**
9: $\hat{S}_{(3)}^k = \tilde{u}^k \hat{v}^{kT} / \max(\tilde{u}^k \hat{v}^{kT} + \tilde{u}_k \hat{v}_k^T, \varepsilon_{n,m})$. ▷ Scale back
10: $\hat{S}_{(3)} = 1 + \sum_{k=2}^{K} \hat{S}_{(3)}^k$. ▷ 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 demonstrate our method. The ML-100k data set contains 100k ratings from 943 users about 1,682 movies. Each user has rated at least 20 movies, and the overall average rating is 3.53.

The ML-1M data set contains over 1 million rating scores from 6,040 users about 3,952 movies. Here, the average score is 3.58, and each user has at least 20 ratings. The distributions of the ratings are shown in Figure 1.

Both data sets have a large number of missing values, with the observation rate being about 5%. The missingness is suspected to be heterogeneous, with higher ratings more likely to be observed (Harper and Konstan (2015)). We heuristically check the missingness pattern by regressing the observation

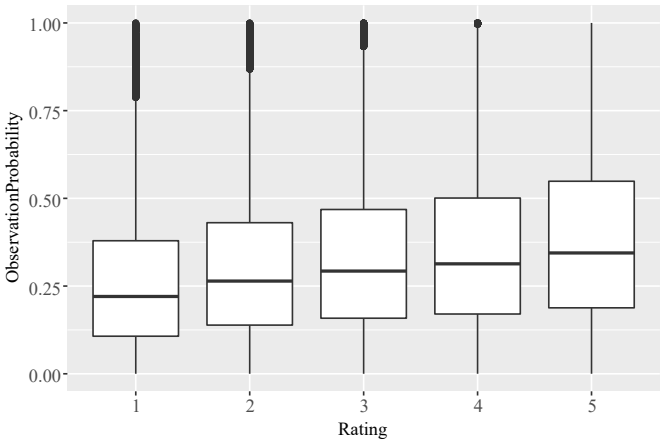


Figure 2. Box plot of the estimated observation probabilities by rating for the ML-1M data set.

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 debiased 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 | 0.9246 | 0.7233 | 0.8650 | 0.6774 |
| Zero-imputation-1 | 0.9065 | 0.7213 | 0.8501 | 0.6713 |
| rSVD | 0.9415 | 0.7355 | 0.8848 | 0.6941 |
| gSVD | 0.9054 | 0.7112 | 0.8748 | 0.6869 |
| 1BITMC-rSVD | 0.9143 | 0.7197 | 0.8684 | 0.6810 |
| ItemImpute | 1.023 | 0.8159 | 0.9799 | 0.7831 |
| UserImpute | 1.042 | 0.8336 | 1.036 | 0.8295 |

probabilities $O_{i,j}$ on the ratings $S_{i,j}$. The observation probabilities are estimated by applying the soft-thresholding SVD method to the binary recording matrix $R$. Figure 2 shows the estimated observation probabilities by rating in the ML-1M data set.

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

Numerous methods exist for predicting unobserved entries in recommendation systems under homogeneous missing schemes. Based on our knowledge, very few of these methods work for heterogeneous missingness or for completely cold-start problems. For comparison, we include the results of the regularized SVD method with an ANOVA-type mean correction (Webb (2006); Paterek (2007)),

denoted as "rSVD" and implemented using the R package rrecsys. This method was originally developed to predict unobserved entries with homogeneous missing schemes, and is popular because of its relatively simple objective function and competitive performance. In view of heterogeneous missingness, we include the propensity score adjustment approach as a comparison (Ma and Chen (2019)). In particular, we use the inverse propensity scores estimated from a one-bit matrix completion (Davenport et al. (2014)) as weights to debias the rSVD method. This is denoted as "1BITMC-rSVD" and is implemented based on the public available code at `https://mdav.ece.gatech.edu/software/`. We also include the results from a group-specific SVD (Bi et al. (2017)), denoted as "gSVD," and implemented using the public available code at `https://sites.google.com/site/xuanbigts/software`. This method uses missing patterns and/or users' and items' covariates to create groups and provide more accurate latent positions than those of rSVD for new users and items. We also include naive mean imputations based on observed values as baseline comparisons. We denote the one-step update of the zero-imputation method as "Zero-imputation-1." The methods are tuned as suggested in the original works to provide the 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 using 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 $\{s_i\}_{i=1}^{M}$ represents the ratings in the unobserved set (or the new sets in completely cold-start problems), and $M$ is the test set size.

Table 1 records the performance of the various methods for the within-sample unobserved predictions. As shown, the methods perform similarly, except for the two mean-imputation methods. All of the methods have better accuracy with the larger data set. The proposed zero-imputation method, "gSVD" method, and " "1BITMC-rSVD" method produce slightly better results than those of the "rSVD" method, because they account for heterogeneous missing data.

For the completely cold-start problem, the public movie-lens data include the user covariates age, gender, and occupation, and a covariate called movie genre. We believe that it would be straightforward to obtain additional attributes for movie other than movie genre, such as directors, actors, and so on, all of which contain information about the general popularity and 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 for the movie. The second is the total number of ratings

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

| | | Item-Cold | | User-Cold | | Both-Cold | |
|---|---|---|---|---|---|---|---|
| | | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| ML-100k | Zero-imputation | 0.9836 | 0.7724 | 0.9640 | 0.7716 | 1.038 | 0.8280 |
| | rSVD | 1.067 | 0.8618 | 0.9803 | 0.7783 | 1.097 | 0.9167 |
| | gSVD | 1.030 | 0.8227 | 0.9606 | 0.7734 | 1.066 | 0.8608 |
| | 1BITMC-rSVD | 1.075 | 0.8779 | 0.9642 | 0.7777 | 1.105 | 0.9277 |
| | MeanImpute | 1.043 | 0.8322 | 0.9645 | 0.7765 | 1.097 | 0.9165 |
| ML-1M | Zero-imputation | 0.9324 | 0.7382 | 0.9699 | 0.7727 | 1.018 | 0.8193 |
| | rSVD | 1.090 | 0.9014 | 0.9781 | 0.7811 | 1.131 | 0.9613 |
| | gSVD | 0.9998 | 0.8021 | 0.9740 | 0.7799 | 1.058 | 0.8647 |
| | 1BITMC-rSVD | 1.103 | 0.9131 | 0.9791 | 0.7877 | 1.143 | 0.9725 |
| | MeanImpute | 1.036 | 0.8313 | 0.9742 | 0.7791 | 1.117 | 0.9366 |

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

| | ML-100k | | ML-1M | |
|---|---|---|---|---|
| | AUC | Accuracy | AUC | Accuracy |
| Zero-imputation | 0.792 | 0.725 | 0.818 | 0.747 |
| rSVD | 0.700 | 0.703 | 0.731 | 0.737 |
| gSVD | 0.724 | 0.728 | 0.732 | 0.739 |
| 1BITMC-rSVD | 0.708 | 0.705 | 0.721 | 0.728 |
| ItemImpute | 0.650 | 0.654 | 0.673 | 0.681 |
| UserImpute | 0.625 | 0.630 | 0.636 | 0.645 |

above three. Note that because "rSVD" and "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 latent position of a new user or item, and then predict the ratings as the inner product of these latent positions. For gSVD, we use the 10-means method based on the covariates of the users/items to generate the group labels.

We randomly select 10% of users and movies for the cold-start sections, and use the other 90% for training. Table 2 summarizes the performance of the various methods for the cold-start problems in the two data sets. Unsurprisingly, the proposed method and the "gSVD" method outperform the 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" versus "bad," for any cut value $k$. We

can classify $S \geq 4$ versus $S < 4$ using the estimated $A^{(4)}$. Table 3 displays the classification results for our method and the other methods. The proposed zero-imputation method performs better than the other methods in terms of the AUC and overall accuracy. The overall accuracy is computed at a cut-off value such that the empirical proportions of ones match.

## 5. Simulations

In this section, we conduct a simulation study. The data are 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 the nonmissing rating matrix $S^0$ and a masking procedure $R$, and then use $S^0 \circ R$ as the observed data.

Following the simulation settings in previous papers, we generate the rating matrix as follows. First, generate the users' latent positions $\{u_i\}$ from a 12-dimensional 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' positions $\{v_j\}$ are generated from $\mathcal{N}((0.5 \times 1_6, 0.1 \times 1_6)^T, \Sigma)$. Here, we generate $S^0_{i,j}$ 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 to the nearest integer in {1,2,3,4,5}. We consider a heterogeneous missing scenario, where we have a higher chance of observing a higher score. The observed probabilities used to generate $R$ are $(0.022, 0.02, 0.02, 0.05, 0.1)^T$ for scores 1 to 5, respectively. The RMSE and MAE are evaluated on all unobserved entries and averaged over 50 simulations. With regard to the computational time, for $(n, m) = (5000, 2500)$, a single simulation for the proposed method takes 6.3 seconds, the "rSVD" method takes 1.6 seconds, the "gSVD" method takes more than 20 seconds, and "1BIT-rSVD" takes more than six minutes. These values include the time taken to select the tuning parameters. Although the "rSVD" method is the fastest, it does not have a special treatment for the heterogeneous missing scheme, and produces a larger error in both the data analysis and the simulations. The results are run on a PC with an 8-core Intel Core i7-10700F processor and 32 GB RAM.

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

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 the other sample sizes show similar patterns. The results are consistent with those based on the Movie-lens data. All of the methods exhibit reasonable performance for unobserved entry prediction and improve as the sample size grows. Compared with "Zero-imputation," "Zero-imputation-1," "gSVD," and

Table 4. Prediction errors for unobserved values with heterogeneous missingness in the simulated data (the number in parentheses is the standard deviation).

| | (1500,800) | | (3000,1500) | | (5000,2500) | |
|---|---|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| Zero-imputation | 0.9954(0.013) | 0.8017(0.011) | 0.9421(0.006) | 0.7536(0.006) | 0.8890(0.004) | 0.7082(0.003) |
| Zero-imputation-1 | 0.9750(0.012) | 0.7420(0.014) | 0.9197(0.005) | 0.7048(0.006) | 0.8555(0.004) | 0.6566(0.005) |
| rSVD | 1.004(0.038) | 0.7645(0.050) | 0.9808(0.032) | 0.7444(0.045) | 0.9630(0.019) | 0.7304(0.032) |
| gSVD | 0.9847(0.011) | 0.7703(0.010) | 0.9649(0.006) | 0.7347(0.006) | 0.9356(0.004) | 0.7146(0.004) |
| 1BITMC-rSVD | 1.002(0.010) | 0.7937(0.011) | 0.9790(0.006) | 0.7752(0.015) | 0.8748(0.011) | 0.6667(0.010) |
| ItemImpute | 1.151(0.016) | 0.9249(0.015) | 1.143(0.009) | 0.9220(0.009) | 1.141(0.006) | 0.9207(0.006) |
| UserImpute | 1.167(0.017) | 0.9331(0.016) | 1.151(0.011) | 0.9255(0.010) | 1.147(0.006) | 0.9241(0.006) |

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

| | Item-Cold | | User-Cold | | Both-Cold | |
|---|---|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| Zero-imputation | 0.9813(0.023) | 0.7582(0.023) | 0.9680(0.017) | 0.7475(0.017) | 0.9772(0.026) | 0.7646(0.027) |
| rSVD | 1.101(0.020) | 0.8927(0.029) | 1.089(0.025) | 0.8854(0.033) | 1.184(0.033) | 0.9907(0.041) |
| gSVD | 1.018 (0.018) | 0.8015(0.016) | 1.008(0.018) | 0.7959(0.017) | 1.058(0.029) | 0.8571(0.028) |
| 1BITMC-rSVD | 1.082(0.016) | 0.8804(0.012) | 1.077(0.018) | 0.8762(0.014) | 1.176(0.031) | 0.9709(0.023) |
| MeanImpute | 1.151(0.014) | 0.9283(0.013) | 1.144(0.016) | 0.9237(0.013) | 1.254(0.020) | 1.123(0.019) |

"1BITMC-rSVD," the "rSVD" method does not account for the heterogeneous missingness and shows larger errors and larger variations. The one-step update for the zero-imputation method outperforms all other methods. The proposed method and the "gSVD" method work reasonably well for cold-start predictions. We see that the proposed method shows a sharper improvement in larger data sets and in cold-start problems.

## Supplementary Material

The online Supplementary Material contains proofs of Theorem 1, Corollary 2, Theorem 2, and Theorem 3.

## Acknowledgments

Kehui Chen's work is partly supported by NSF DMS 2210402. We thank the associated editor and the two reviewers for the helpful comments and suggestions, which greatly improve the quality of the manuscript.

## 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. *The 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**, Article 19.

Hoover, D. N. (1982). Row-column exchangeability and a generalized model for probability. In *Exchangeability in Probability and Statistics* (Edited by G. Koch and F. Spizzichino), 281–291.

Imbens, G. W. and Rubin, D. B. (2015). *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press.

Keshavan, R. H., Montanari, A. and Oh, S. (2009). Matrix completion from a few entries. *2009 IEEE International Symposium on Information Theory*, 324–328.

Keshavan, R. H., Montanari, A. and Oh, S. (2010). Matrix completion from noisy entries. *Journal of Machine Learning Research* **11**, 2057–2078.

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**, 2005–2030.

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*, 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*, 5–8.

Recht, B. (2011). A simpler approach to matrix completion. *The Journal of Machine Learning Research* **12**, 3413–3430.

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**, 279–313.

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N. and Joachims, T. (2016). Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on Machine Learning* **48**, 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*, 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: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 *Proceedings of the 36th International Conference on Machine Learning* **PMLR 97**, 6638–6647.

Webb, B. (2006). Netflix update: Try this at home. *Project Sifter*. Blog. Web: `https://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 80**, 5433–5442.

Jiashen Lu

Department of Statistics, University of Pittsburgh, Pittsburgh, PA 15260, USA.

E-mail: jil235@pitt.edu

Kehui Chen

Department of Statistics, University of Pittsburgh, Pittsburgh, PA 15260, USA.

E-mail: khchen@pitt.edu

(Received September 2021; accepted June 2022)