Nonparametric Predictions for Network Links and Recommendation Systems

by

Jiashen Lu

Bachelor of Science, Fudan University, 2015

Master of Science, University of Michigan, Ann Arbor, 2017

Submitted to the Graduate Faculty of

the Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Jiashen Lu

It was defended on

April 28th 2022

and approved by

Kehui Chen, Department of Statistics, University of Pittsburgh

Satish Iyengar, Department of Statistics, University of Pittsburgh

Lucas Mentch, Department of Statistics, University of Pittsburgh

Jing Lei, Department of Statistics and Data Science, Carnegie Mellon University

Nonparametric Predictions for Network Links and Recommendation Systems

Jiashen Lu, PhD

University of Pittsburgh, 2022

In this thesis, we develop methodologies to make nonparametric predictions in relational data. Prominent examples of relational data include user-user network interactions and user-item recommendation systems. For social networks, we follow a new latent position framework and develop prediction methods in pure cold-start scenarios where the new nodes do not have any observed links to start with. For recommendation systems, we first develop a Zero-imputation method to address the challenges of heterogeneous missing and then make predictions for missing values and for new users or items. We explore some applications of this Zero-imputation method in the context of social network with missing edges. In particular, we are interested in inferences in network regression models. We compare our approach with existing methods through simulations and apply our method to one real Friends and Lifestyle data that study the influence of social network on alcohol and drug use behaviors among teenagers.

Keywords: Link predictions, Graph Root Distribution, Bipartite Graph, Cold-Start, Missing Imputation, Network AutoRegressive Model.

Table of Contents

List of Tables

List of Figures

Preface

I am grateful to all who help me during the five years' study in Pittsburgh.

First, I would like to thank my advisor Dr. Kehui Chen. I have learned a lot about research under her supervision. Ideas arise from sparkles in the mind, but they need to be lit and developed in a systematical way. This is a tough process for a newly proposed method as well as a challenging process for one to transform from student to researcher. This thesis then becomes the witness.

Second, I would like to thank all the committee members, Dr. Satish Iyengar, Dr. Lucas Mentch, and Dr. Jing Lei for the helpful comments and suggestions on this thesis work. It is enlightening for me to discuss the idea in this thesis with them as they can provide different perspectives that I am not aware of.

Third, I would like to thank my college peers and the statistics department. The weekly seminars provided by the department enable me to have a taste of other statistical areas. More importantly, speakers in the seminar often set examples for what qualifies a great work.

Last but not least, I would like to thank my parents. This work is impossible to be completed without their continuous supports.

1.0 Introduction

The study of relational data has attracted lots of attentions recently since it has many real world applications. In practice, relational data can be expressed as user-user interactions and user-item evaluations. In social networks, for example, one often observes friendship relations between users (Newman, 2018) and are interested in predicting new potential links between users. In user-item relational data such as the Netflix data (Bennett et al., 2007), customers have already viewed and rated multiple movies (Koren et al., 2009), and the task is to predict the unobserved scores for existing users and movies. A more challenging task is to predict ratings for new users and new movies.

We mainly focus on exchangeable relational data in this thesis. The exchangeable requirement means that the distribution of the data is unaffected by row or column permutations. Heuristically, it means the order of the nodes does not matter. Relational data can be naturally viewed as a two dimensional array of random variables $\{A_{i,j}\}$, in which i indexes row nodes and j indexes column nodes. In social networks, node means users and $A_{i,j} = 1$ if person i and person j are friends and 0 otherwise. In score predictions, row means users and column means items and each entry $A_{i,j}$ is a rating from user i to item j.

Link prediction aims to predict unobserved links between node pairs in the data set. Latent space approach is popular in the literature (Hoff et al., 2002; Airoldi et al., 2008) to solve the network link prediction problem. The general idea of these approaches is to first find a vector embedding of each node based on observed data, denoted as Z, and then find a bivariate mapping function ϕ such that $\phi(Z_i, Z_j)$ determines the link probability between two nodes (Lü and Zhou, 2011). In the case

that additional node covariates X are available, the mapping function may take X as input as well. This framework gives large flexibility in terms of modeling but how to determine the latent positions and the mapping function in a sensible way remains challenging and application specific. Recently, Lei (2021) proposed the concept of Graph Root Distribution (GRD) to study the exchangeable network data. It proves that there exists a one-to-one correspondence between GRD and graphon under mild conditions. The GRD framework naturally leads to a canonical form of the latent positions and explicitly characterizes the mapping function ϕ.

In Chapter 2, we first review the framework of GRD and then use it to solve the cold-start link prediction problem in network data. The cold-start problem refers to predict links for new users who do not have any observed interactions with existing users. Due to the heuristic and complex form of the latent positions and the link function, most of the existing work do not work for the pure cold-start problems. We build node covariates, and the latent positions from the GRD framework into a regression model with simplex constrains, and illustrate its good performance using numerical experiments as well as data examples.

In Chapter 3, we try to extend this new prediction approach to rating predictions in recommendation systems. The problem is more challenging than the one in Chapter 2 in two perspectives. First, we have two sets of nodes: users and items. Therefore the data we study is in fact a bipartite graph. We will extend the work of Lei (2021) and establish the framework of Bipartite Graph Root Distribution (BGRD). Second, there are many missing entries in the recommendation systems since most people only interact with a few items. For example, the famous Netflix data (Bennett et al., 2007) only has 1% observations and 99% of the entries are missing. Most of the existing work focus on the within-sample predictions for unobserved entries and train on observed entries only (Koren et al., 2009; Paterek,

2007; Webb, 2006). If missingness is heterogeneous, i.e., the missing probability is different for each entry, approaches that target at the observed risk function will be biased (Ma and Chen, 2019). We first propose a Zero-imputation method to solve the prediction problem under heterogenous missing and extended it to the cold-start prediction problems. We provide theoretical guarantees of the proposed method, and demonstrate its good performance in data analysis as well as simulations.

In Chapter 4, we explore how the proposed Zero-imputation method can help improve the inferences in network regression models when the observed social network is incomplete. In Chapter 5, we summarize our thesis work and discuss some possible future extensions.

2.0 Feature-based Network Link Predictions under Graph Root Distribution

2.1 Introduction

Network data is common nowadays since it can represent the pairwise relations between a group of people (nodes). Based on the observation of all existing links between nodes, data often take the form of a square matrix, which is often called the adjacency matrix, with binary elements. Link prediction problems are one of the key problems in network analysis. It aims to predict unobserved links between node pairs in the data set. The problem of link prediction in networks has attracted lots of attentions in recent years (Lü and Zhou, 2011; Clauset et al., 2008; Al Hasan and Zaki, 2011; Wu et al., 2018). It has broad applications in many fields, such as predicting friendship connections in social relations (Lazega et al., 2001; Leskovec and Mcauley, 2012), recommending co-purchase items on shopping websites, finding scientific relations within protein-protein interactions (Barzel and Barabási, 2013), and predict links in terrorist network (Anil et al., 2015). There exist different methods for predicting unobserved links, where one often assume that the training data set is completely observed, i.e., the network is completely observed between a set of training nodes, and for a set of testing nodes, one has only partially observed links and the task is to predict the unobserved links. The pure cold-start problem is known to be more challenging because the new nodes do not easily fit into the training framework.

The latent position view (Hoff et al., 2002; Handcock et al., 2007; Airoldi et al., 2008) provides an intuitive way to solve the link prediction problem and may be

extended to cold-start problems. The general framework for the latent approach is to first find a representation of each person from observed data and then consider a link function to predict link probabilities between node pairs (Lü and Zhou, 2011; Zhao et al., 2017). Different embedding ways include heuristic random walk (Perozzi et al., 2014; Grover and Leskovec, 2016), spectral decomposition (Lei and Rinaldo, 2015; Rohe et al., 2011), Bayesian method (Durante et al., 2017) and so on. Additional node features may help to improve the accuracy of predictions, especially in cold-start problems. How to combine the node features and the latent positions in prediction problems is an interesting research topic. One idea is to model the link probability as a function of the covariates as well as latent positions (Baldin and Berthet, 2018; Liu, 2019), and another idea is to model the latent positions as a function of the covariates (Wei et al., 2017). There exist many different methods whose performance is usually very good in the training data, acceptable in the testing data with partial link observations, and deteriorates fast in pure cold-start problems.

In this chapter, building upon a newly developed GRD framework (Lei, 2021), we develop a more streamlined method for the network link prediction, which does not make parametric or heuristic assumptions. Its simple form is particularly suitable for predictions in the cold-start problems.

2.2 Brief review of Graph Root Distribution (GRD)

In the framework of GRD Lei (2021), each binary matrix, if viewed as an exchangeable random graph, can be generated by first generating independent user latent positions Z from a distribution F on a Krein space (defined below) and then generating the (i, j)-th entry from a Bernoulli distribution with the link probability

as the Krein inner product between two latent positions.

Definition 2.2.1. *A Krein space $\mathcal{K} = \mathcal{H}_+ \ominus \mathcal{H}_-$, is the direct sum of two Hilbert space $\mathcal{H}_+$ and $\mathcal{H}_-$, called positive and negative part respectively, for each element $(x; y), (x'; y') \in \mathcal{K}$, the inner product is defined as*

$$\langle (x; y), (x'; y') \rangle_{\mathcal{K}} = xx' - yy'. \tag{1}$$

Remark 2.2.2. *We see from the definition of the Krein space that each latent position has one positive and one negative part. The inner product on a Krein space is different from the usual inner product in an Euclidean space.*

A graph root distribution is a probability distribution living on the Krein space.

Definition 2.2.3. *A graph root distribution is a probability measure on a Krein space $\mathcal{K}$, so that if z_1, $z_2 \in \mathcal{K}$*

$$P(\langle z_1, z_2 \rangle_{\mathcal{K}} \in [0, 1]) = 1. \tag{2}$$

Lei (2021) proved the existence and identifiability of the GRD, and derived a canonical form of the latent position vector, which motivates the estimation of latent position as follows.

2.3 Latent position estimation

In Lei (2021), the author considered estimating latent positions by truncating the weighted eigen vectors decomposed from A. Rewrite A in its eigen-value decomposition form

$$A = \sum_{j=1}^{n_1} \widehat{\lambda}_j \widehat{a}_j \widehat{a}_j^T - \sum_{j=1}^{n-n_1} \widehat{\gamma}_j \widehat{b}_j \widehat{b}_j^T, \tag{3}$$

where $\{\widehat{\lambda}_j\}$ are the positive eigen values in decreasing order, $\{\widehat{\gamma}_j\}$ are the absolute value of the negative eigen values in decreasing order, $\{\widehat{a}_j\}$, $\{\widehat{b}_j\}$ are the corresponding eigen vectors and n_1 is the number of all positive singular values in matrix A. When truncated at dimensions p_1 and p_2, the latent positions takes the form

$$\widehat{z}_i = [\widehat{\lambda}_1^{1/2}\widehat{a}_{1,i}, \ldots, \widehat{\lambda}_{p_1}^{1/2}\widehat{a}_{p_1,i}; \widehat{\gamma}_1^{1/2}\widehat{b}_{1,i}, \ldots, \widehat{\gamma}_{p_2}^{1/2}\widehat{b}_{p_2,i}]^T.$$

In practice, we find that soft singular-value thresholding approach (Koltchinskii et al., 2011; Cai et al., 2010) performs better than this type of hard-threshold approach, although the theoretical convergence rate are the same (more details in Section 3.2). The soft thresholding 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, then for some threshold λ, the estimated position for person i is

$$\begin{aligned}\widehat{z}_i = [&(\widehat{\lambda}_1 - \lambda)_+^{1/2}\widehat{a}_{1,i}, \ldots, (\widehat{\lambda}_{n_1} - \lambda)_+^{1/2}\widehat{a}_{n_1,i}; \\ &(\widehat{\gamma}_1 - \lambda)_+^{1/2}\widehat{b}_{1,i}, \ldots, (\widehat{\gamma}_{n-n_1} - \lambda)_+^{1/2}\widehat{b}_{n-n_1,i}]^T.\end{aligned}$$

Here λ is a tuning parameter. In the following simulations and real data analysis, we first split dataset into training and testing, then separate the training set into training-training and the tuning set. We search over a grid of λ, fitting over the training-training set and predict on the tuning test set and choose the best λ that minimize the mean squared error on the tuning set.

2.4 Constrained regression prediction

This step differs from the classical regression method because we require the resulting prediction value to satisfy the GRD requirement. Suppose in Section 2.3, our estimated positions are $\{\widehat{z}_i\}_{i=1}^n \in \mathbb{R}^{p_1+p_2}$, in which p_1 and p_2 are the positive and negative truncation dimension respectively. The best estimation $\widetilde{z}$, in terms of the mean prediction error, for a new user's latent position that associated with covariate X is $\mathbb{E}[z|X]$. According to the definition of conditional expectation, this can be approximated by a weighted version of empirical data, i.e. $\sum_{i=1}^n w_i z_i$, where the weights $\{w_i\}$ depend on the joint distribution of z and X as well as the marginal distribution of X. One observation here is that as long as the estimated latent positions take this weighted summation form, all the resulting inner products $\langle z, \widetilde{z}\rangle_{\mathcal{K}}$ will be between 0 and 1, and satisfy the GRD requirement. This motivates us to consider the following two-step approach. First, use a nonparametric statistical learning method to estimate z given X, denoting the learned position as z^*. In a second step, we project z^* 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.

Suppose $z^* \in \mathbb{R}^{(p_1+p_2)\times 1}$ is the estimated position for the new node using some learning method. Let Q denote the signature matrix, that is, a $p_1 + p_2$ diagonal matrix with first p_1 elements equal to 1 and last p_2 elements equal to -1. Then the estimated position $\widetilde{z} = \widehat{Z}^T w \in \mathbb{R}^{(p_1+p_2)\times 1}$ could be obtained by solving the following

optimization problem,

$$\min_{\widetilde{z}} \frac{1}{2} \|\widehat{Z}Q\widetilde{z} - \widehat{Z}Qz^*\|^2 \tag{4}$$
$$s.t. \quad \begin{cases} \widetilde{z} = \widehat{Z}^T w \\ \sum w_i = 1 \\ w_i \geq 0 \qquad (i = 1, 2, 3, \cdots, n). \end{cases}$$

The above optimization problem is convex and has a unique solution in terms of $\widetilde{z}$, but the constrain set is complex to deal with. Solving (4) is equivalent to minimizing $\frac{1}{2}\|\widehat{Z}Q\widehat{Z}^T w - \widehat{Z}Qz^*\|^2 + \lambda \mathbb{I}_C\{w\}$ 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. Condat (2017) discussed a few more algorithms to solve this type of constrained regression problem and finding the most efficient algorithm is still an open problem. While the solution may not be unique in terms of w but they still correspond to the unique solution $\widetilde{z}$. In our numerical studies, we used the Random Forest method (Breiman, 2001) to predict each dimension in z^*. The resulting z^* takes a weighted summation form for each dimension, but is not a weighted summation of the multivariate response. If some learning methods directly produce a z^* in the form of a weighted summation $\widehat{Z}^T w$, the projection step is not needed.

2.5 Classifications

Once we obtained all the predicted positions $\widetilde{z}$ for all the new nodes, the link prediction probability could be easily computed following the Krein inner product. However, we still need to classify the probability into 0 and 1 as the task here is to predict the possible friends for the new person. As we're interested in the top-N recommendations as well as maintaining sensible overall classification performance, we use the following classification rule. Let $\widehat{p}_1, \ldots, \widehat{p}_n$ be the estimated link probability, then

(1) Estimate the total link numbers N by the estimator $\sum_{i=1}^{n} \widehat{p}_i$.

(2) All the top-N links are determined to be 1.

(3) For all the others, generate a Bernoulli variable with parameter that equals to the estimated probability.

2.6 Simulations

2.6.1 Evaluation Criterion

The performance of the proposed method is measured by Area Under the Receiver Operating Characteristic (ROC) Curve (AUC), Precision, Sensitivity and Specificity.

In the link classification problem, for any binary classifier C, we can construct the two-by-two confusion matrix the way as Table 1. Then the True Positive rate (TPR) and False Negative rate (FPR) associated with C are defined as,

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}}, \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}}$$

where P and N are the total Positive and Negative cases in the set. ROC graphs are two-dimensional graphs in which TPR is plotted on the Y axis and FPR is plotted on the X axis. Let $\{(\widehat{Y}_i, Y_i)\}_{i=1}^n$ be the (Predicted probability, True-label) pairs with sample size n. For every threshold c, we can define a classifier C according to the rule: $\mathbb{I}\{\widehat{Y}_i > c\}$. Therefore such classifier C can be related to one point in ROC space using the confusion matrix. If we vary all thresholds, we can obtain a curve. AUC is the area under this curve.

Table 1: Confusion matrix for the binary classifier, "1" is classified as Positive and "0" as Negative.

		True class	
		1	0
Classification	1	True Positive (TP)	False Positive (FP)
Class	0	False Negative (FN)	True Negative (TN)
Column totals		P	N

AUC has widely been used to evaluate the performance of binary classifiers (Bradley, 1997). The AUC of a classifier can be explained as the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance (Fawcett, 2006). A random guess will give a 0.5 AUC value.

For any classifier C, Precision, Sensitivity (also called Recall) and Specificity are another three measures of the classification quality that can be computed using the confusion matrix.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{P}} \tag{6}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{N}} \tag{7}$$

Heuristically, Precision means, among all those who predicted to be 1, how many are in truth to be 1 and Sensitivity means, among all those indeed 1, how many are predicted to be 1. The finite version of Precision (P@N) is defined when the top-N are classified as 1. Therefore, if we're interested in top-N recommendations, we may look at P@N, and if we're interested the overall recommendation quality, AUC, Sensitivity and Specificity should be used.

2.6.2 Simulation Settings

We generate experimental data sets that are close to the Blog data and Facebook data (These data are analyzed in Section 2.7). For all the simulation part, the sample size is 1000. We always divide the data into 70% as old users in the training and 30% as new users in the testing. And we report the performance on the test set. All results are based on 100 replications.

We consider one lower dimension case and one higher dimension case. For the lower dimension case, we first decompose the observed adjacency matrix from the data using SVD and obtain users' latent positions, then truncate the positive and negative part using $p_1 = 5$ and $p_2 = 3$. For the higher dimension case, we truncate into $p_1 = 25$ and $p_2 = 10$. Then we generate the simulated network according to the inner product between positive and negative parts (inner product on Kerin Space). For the regression step in the prediction, we take the original covariates from the Blog data and Facebook data. Here we do not know any specific form of relationship

between the latent positions and the covariates. To evaluate the performance of the proposed method under different types of node covariate information, we tried three variations.

1. Randomly replace half of the original covariates with independent Normal random variables $\mathcal{N}(0, 1)$.
2. Randomly replace half of the original covariates with independent Bernoulli random variables $\mathcal{B}(0.5)$.
3. Replace all the original covariates with independent Normal random variables $\mathcal{N}(0, 1)$.

For Blog data, the original covariates includes the blog category each user belongs to. There are overall 60 categories and one user may belongs to multiple categories. Table 2 shows the classification results of the proposed method under four different settings of covariates. We also include an original result where the true latent positions were used to calculate the link probability and make the classification. We see that these predictions are intrinsically difficult and the proposed method is not bad comparing to the oracle method. The original covariates provide some information on the link probability. When we only have nuisance covariates, the estimated latent positions are basically the sample mean of the training data.

For Facebook data simulations, the side information is the one-hot encoding of some categorical variables. As to protect privacy, the exact meaning is unknown in the original data. There are 576 binary covariates in total. Table 3 displays the results in both lower and higher dimension settings. Similar to the observations in the Blog data, our method performs reasonably well and the original side information is useful for making predictions in cold-start problems.

Table 2: Performance of the proposed method in lower dimension and higher dimension Blog-simulation data sets, compared with the results using oracle link probability and the results replacing with three nuisance covariate variations.

		AUC	P@5	P@10	Sensitivity	Specificity
Low Dimension	Oracle	65.9	31.2	28.2	20.1	91.0
	Proposed method (original covariates)	60.4	23.0	21.3	24.4	85.9
	With half Normal covariates	57.6	17.9	17.2	19.5	87.2
	With half Binomial covariates	57.7	18.8	17.7	18.6	88.4
	With complete random covariates	57.6	17.3	17.1	19.0	87.9
High Dimension	Oracle	64.2	43.7	38.6	24.1	88.8
	Proposed method (original covariates)	59.3	27.9	25.7	25.4	84.8
	With half Normal covariates	57.1	23.4	22.0	20.8	86.1
	With half Binomial covariates	57.1	22.9	21.9	21.1	86.0
	With complete random covariates	57.1	23.0	21.6	21.2	85.7

Table 3: Performance of the proposed method in lower dimension and higher dimension Facebook-simulation data sets, compared with the results using oracle link probability and the results replacing with three nuisance covariate variations.

		AUC	P@5	P@10	Sensitivity	Specificity
Low Dimension	Oracle	78.7	43.2	42.2	47.5	91.2
	Proposed method (original covariates)	72.9	34.2	34.0	31.1	89.9
	With half Normal covariates	69.7	31.3	30.7	27.4	87.9
	With half Binomial covariates	70.0	30.5	30.2	28.0	87.6
	With complete random covariates	66.4	22.7	22.6	25.9	85.6
High Dimension	Oracle	64.8	63.4	61.7	49.5	91.2
	Proposed method (original covariates)	69.5	41.1	39.4	32.2	88.4
	With half Normal covariates	64.8	33.2	31.5	26.8	86.3
	With half Binomial covariates	64.3	30.8	29.1	26.9	85.5
	With complete random covariates	59.9	22.5	21.9	23.4	83.9

2.7 Real Data Analysis

We consider four real data sets in this section, namely, Disney, Facebook, Blog and Enron. Disney data set consists of 124 nodes and 28 features. Each node represents a product in category Disney sold on Amazon. Two nodes are connected if two products are usually being co-purchased. Attributes for each product include Amazon price, Rating ratios, Avg Ratings, Number of reviews, Sales Rank and so on. Facebook data set consists of 1045 nodes and 576 binary covariates. The average link number for each node is 56.8. The average link probability is 5.5%. Blog data

set consists of 88784 bloggers with friendship relations. Two nodes are connected if there is a link between two blogs. We first cluster the nodes by first fitting to a Stochastic Block Model and choose the second largest community as the data set. The selected data set has size 5282. The average link number is 171.5 and the link probability is 3.2%. There are 60 user-category node features. Enron data set is a CMU email address network set with 13533 nodes and 18 attributes. Each node represents an email address. The attributes include Average number CCed, Average number replied, Average content length and so on. The average link number for each node is 26.2. The average link probability is 0.2%. We compare our proposed method with two other methods, Probabilistic NoiseResilient Representation Consensus Learning (PNRCL) and Max Margin NoiseResilient Representation Consensus Learning (MMNRCL) that are proposed in Wei et al. (2017). Probabilistic and Max Margin are two different ways to estimate the node latent positions given observed data and NRCL is a regression method for learning the linear relations between the estimated positions and the covariates. Table 4 reports the performance of our methods and the other two methods. We see that in large data sets such as Blog and Enron, our method out-performs the other two in AUC. While in small sets, our method performs similarly to the other two methods.

Table 4: Performance of the proposed method, compared with PNRCL (Wei et al., 2017) and MMNRCL (Wei et al., 2017) in Disney, Facebook, Blog, and Enron data sets.

		AUC	Precision@5	Sensitivity@5
Disney	Proposed Method	73.4	26.2	24.9
	PNRCL	79.2	35.2	35.2
	MMNRCL	75.2	28.3	29.2
Facebook	Proposed Method	80.8	50.7	5.1
	PNRCL	83.2	40.8	3.8
	MMNRCL	83.0	41.2	3.7
Blog	Proposed Method	68.0	20.6	0.8
	PNRCL	61.8	17.5	0.6
	MMNRCL	60.9	18.3	0.6
Enron	Proposed Method	85.1	8.7	4.4
	PNRCL	72.7	5.9	3.1
	MMNRCL	75.9	5.3	2.4

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

3.1 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 $\sum_{i=1}^{n}\sum_{j=1}^{m}\mathcal{L}(S_{i,j}, \widehat{S}_{i,j})$. In practice, regularization methods and modeling assumptions may be applied to modify the observed loss function $\sum_{(i,j)\in\Omega}\mathcal{L}(S_{i,j}, \widehat{S}_{i,j})$ 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 $\sum_{(i,j)\in\Omega}\frac{1}{O_{i,j}}\mathcal{L}(S_{i,j}, \widehat{S}_{i,j})$, 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 $\mathcal{L}(S_{i,j}, \widehat{S}_{i,j})$ 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 chapter, 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 3.3.

In Section 3.4, Section 3.5, and Section 3.6, we illustrate the proposed approach for predicting unobserved values with heterogeneous missing and new users'/items' ratings using the Movie-lens data sets, the Hotel recommendation data sets and simulated data sets. Theoretical proofs can be found in Section 3.7. Further sensitivity analysis results can be found in Section 3.8.

3.2 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 $\{1, 2, \ldots, K\}$. 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 $2 \leq k \leq K$,

$$P(S_{i,j} \geq k) = P(A_{i,j}^{(k)} = 1)/O_{i,j} = \frac{\mathbb{E}(A_{i,j}^{(k)})}{\mathbb{E}(R_{i,j})} = \frac{\mathbb{E}(A_{i,j}^{(k)})}{\mathbb{E}(A_{i,j}^{(k)}) + \mathbb{E}(A_{(k);i,j})}, \tag{8}$$

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 (8) the Zero-imputation method. We note that the sum of $\mathbb{E}(A_{i,j}^{(k)})$ and $\mathbb{E}(A_{(k);i,j})$ equals $O_{i,j}$. We use Equation (8) 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 (8) 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 $A^{(k)} = \sum_{1\le i\le (m\wedge n)} \widehat{\sigma}_i^k \widehat{U}_i^k (\widehat{V}_i^k)^T$ be the Singular Value Decomposition (SVD) of matrix $A^{(k)}$ where $\widehat{\sigma}_i^k$ is the i-th singular value, $\widehat{U}_i^k$ is the corresponding left singular vector, and $\widehat{V}_i^k$ is the right singular vector. Similarly let $A_{(k)} = \sum_{1\le i\le (m\wedge n)} \widehat{\sigma}_{k,i} \widehat{U}_{k,i} \widehat{V}_{k,i}^T$ 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 $\widehat{S}$.

1: **Parallel for k** in 2,..., K **do**

2: Obtain $A^{(k)}, A_{(k)}$ by truncation and Zero-imputation.

3: $A^{(k)} = \sum_{1\le i\le (m\wedge n)} \widehat{\sigma}_i^k \widehat{U}_i^k (\widehat{V}_i^k)^T$. ▷ SVD of upper-truncation matrix

4: $\widehat{A}^{(k)} = \sum_{1\le i\le (m\wedge n)} \{\widehat{\sigma}_i^k - \lambda^k\}_+ \widehat{U}_i^k (\widehat{V}_i^k)^T$. ▷ Soft-thresholding using $\lambda^k = \widehat{\sigma}_{p+1}^k$

5: $A_{(k)} = \sum_{1\le i\le (m\wedge n)} \widehat{\sigma}_{k,i} \widehat{U}_{k,i} \widehat{V}_{k,i}^T$. ▷ SVD of lower-truncation matrix

6: $\widehat{A}_{(k)} = \sum_{1\le i\le (m\wedge n)} \{\widehat{\sigma}_{k,i} - \lambda_k\}_+ \widehat{U}_{k,i} \widehat{V}_{k,i}^T$. ▷ Soft-thresholding using $\lambda_k = \widehat{\sigma}_{k,p+1}$

7: **end Parallel**

8: $\widehat{S}^k = \frac{\widehat{A}^{(k)}}{\max\{\widehat{A}^{(k)} + \widehat{A}_{(k)}, \varepsilon_{n,m}\}}$. ▷ Scale back

9: $\widehat{S} = 1 + \sum_{k=2}^K \widehat{S}^k$. ▷ Prediction

Remark 3.2.1. *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 3.2.2. *As specified in Theorem 3.2.3, 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 $\widehat{A}^{(k)}$ and $\widehat{A}_{(k)}$ 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 $\lambda^k = \widehat{\sigma}^k_{p+1}$ and $\lambda_k = \widehat{\sigma}_{k,p+1}$, where $\widehat{\sigma}^k_{p+1}$ and $\widehat{\sigma}_{k,p+1}$ 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 $\widehat{S}$ be the original Zero-imputation estimation, we may apply the soft singular value thresholding again on the matrix $\widehat{S} - \frac{1}{\widehat{R}} \circ P_\Omega(\widehat{S} - S)$, where $\widehat{R}$ 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 $\widehat{S}_{update}$.

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 (8) 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 $\widehat{S} = \widehat{\mathbb{E}(S')} \times (b-a) + a$. 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 $O_{i,j} = \delta_{n,m}\widetilde{O}_{i,j}$, $\mathbb{E}(A^{(k)}) = \delta_{n,m}\widetilde{P}^{(k)}$ and $\mathbb{E}(A_{(k)}) = \delta_{n,m}\widetilde{P}_{(k)}$, where $\widetilde{O}_{i,j}$, $\widetilde{P}^{(k)}$, $\widetilde{P}_{(k)}$ take values between 0 and 1 and are considered to be at a constant level. In the following, we use $\sigma_i(\widetilde{P}^{(k)})$ to denote the i-th singular value of $\widetilde{P}^{(k)}$ and use C to denote positive constant values.

Theorem 3.2.3. *For results simplicity, we assume $m \leq n$. Let $\widehat{S}^k_{i,j}$ 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 $min_{i,j}\widetilde{O}_{i,j} = \widetilde{C}_2 > 0$. 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 $\widetilde{C}_2\delta_{n,m}$, then with probability at least $1-n^{-C_3}$, we have for $2 \le k \le K$,

$$\frac{1}{mn}\sum_{i,j}\left(\widehat{S}^k_{i,j} - P(S_{i,j} \ge k)\right)^2 \le \min_{0\le r\le m}\{\frac{C_4 r}{m\delta_{n,m}} + \frac{C_5}{mn}\sum_{i\ge r+1}\sigma_i^2(\widetilde{P}^{(k)})\}. \qquad (9)$$

Remark 3.2.4. *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 3.2.5. *Theorem 3.2.3 provides a general bound to the error. The rate of convergence depends on the structure of the singular values. Corollary 3.2.7 and Corollary 3.2.8 provide the convergence rates for a finite rank structure and a polynomial decay structure.*

Remark 3.2.6. *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 $\widehat{S}_{i,j} = 1 + \sum_{k=2}^{K} \widehat{S}_{i,j}^k = 1 + \sum_{k=2}^{K} \widehat{P}(S_{i,j} \geq k)$.

Corollary 3.2.7. *Given conditions in Theorem 3.2.3, if all matrices $\widetilde{P}$ has finite rank, then $\frac{1}{mn}\sum_{i,j}(\widehat{S}_{i,j} - \mathbb{E}(S_{i,j}))^2 = O_p(\frac{1}{m\delta_{n,m}})$.*

Corollary 3.2.8. *Given conditions in Theorem 3.2.3, if for all matrices $\widetilde{P}$, the singular values decay in a polynomial rate as $\sigma_r \asymp \frac{\sqrt{mn}}{r^\alpha}$ for some $\alpha > 1$, then $\frac{1}{mn}\sum_{i,j}(\widehat{S}_{i,j} - \mathbb{E}(S_{i,j}))^2 = O_p((\frac{1}{m\delta_{n,m}})^{1-\frac{1}{2\alpha}})$.*

3.3 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,

$$S = \begin{array}{c} \\ \text{Old-user} \\ \text{New-user} \end{array} \begin{array}{c} \begin{array}{cc} \text{Old-item} & \text{New-item} \end{array} \\ \begin{pmatrix} S_{(1)} & S_{(2)} \\ S_{(3)} & S_{(4)} \end{pmatrix} \end{array}. \tag{10}$$

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 3.3.1. *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,

$$W(s,t) = \sum_r \lambda_r \phi_r(s) \psi_r(t). \tag{11}$$

A graphon W with SVD in (11) 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 3.3.2. *(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 3.3.3. *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 3.3.4. *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 3.3.5. *(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 $\sum_{1 \le i \le p} (\widehat{\sigma}_i - \lambda) \widehat{U}_i \widehat{V}_i^T$, 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,

$$\widehat{u} = [\widehat{u}_1, \ldots, \widehat{u}_{n_1}]^T = [\sqrt{\widehat{\sigma}_1 - \lambda}\widehat{U}_1, \ldots, \sqrt{\widehat{\sigma}_p - \lambda}\widehat{U}_p] \in \mathbb{R}^{n_1 \times p}, \tag{12}$$

and

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

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 $2 \le k \le K$. We take new users for illustration, and new items' estimation is similar.

Given the estimates for old users $\{\widehat{u}_i\}_{i=1}^{n_1}$ 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 $\widehat{u} \in \mathbb{R}^{n_1 \times p}$, $\widehat{v} \in \mathbb{R}^{m_1 \times p}$ 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 $\widetilde{u} = \widehat{u}^T w \in \mathbb{R}^{p \times 1}$ could be obtained by solving the following optimization problem,

$$\min_{\widetilde{u}} \ \frac{1}{2}\|\widehat{v}\widetilde{u} - \widehat{v}u^*\|^2 \tag{13}$$
$$s.t. \quad \begin{cases} \widetilde{u} = \widehat{u}^T w \\ \sum_{i=1}^{n_1} w_i = 1 \\ w_i \geq 0 \qquad (i = 1, \cdots, n_1). \end{cases}$$

The above optimization problem is convex and has a unique solution in terms of $\widetilde{u}$, but the constrain set is complex to deal with. Solving (13) is equivalent to minimizing $\frac{1}{2}\|\widehat{v}\widehat{u}^T w - \widehat{v}u^*\|^2 + \lambda \mathbb{I}_C\{w\}$ 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 $\widetilde{u}$. 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 $\widehat{u}^T w$, 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 ε_{n_1,m_1}; covariate matrix X.

Output: Predicted rating matrix $\widehat{S}_{(3)} \in \mathbb{R}^{n_2 \times m_1}$.

1: **Parallel for k** in 2,... K **do**

2: Obtain $A^{(k)}, A_{(k)}$ by truncation and Zero-imputation.

3: $A^{(k)} = \sum_{1 \le i \le (m_1 \wedge n_1)} \widehat{\sigma}_i^k \widehat{U}_i^k (\widehat{V}_i^k)^T$. ▷ SVD of upper-truncation matrix

4: Obtain the canonical form of the latent positions $\widehat{u}^k$, $\widehat{v}^k$ according to (12).

5: Obtain $u^{k,*} \in \mathbb{R}^{n_2 \times p}$ by multivariate learning methods such as random forests.

6: Obtain $\widetilde{u}^k \in \mathbb{R}^{n_2 \times p}$ according to (13).

7: Repeat steps 3-6 for $A_{(k)}$.

8: **end Parallel**

9: $\widehat{S}^k_{(3)} = \frac{\widetilde{u}^k \widehat{v}^{kT}}{\max\{\widetilde{u}^k \widehat{v}^{kT} + \widetilde{u}_k \widehat{v}_k^T, \varepsilon_{n,m}\}}$. ▷ Scale back

10: $\widehat{S}_{(3)} = 1 + \sum_{k=2}^{K} \widehat{S}^k_{(3)}$. ▷ Prediction

3.4 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.

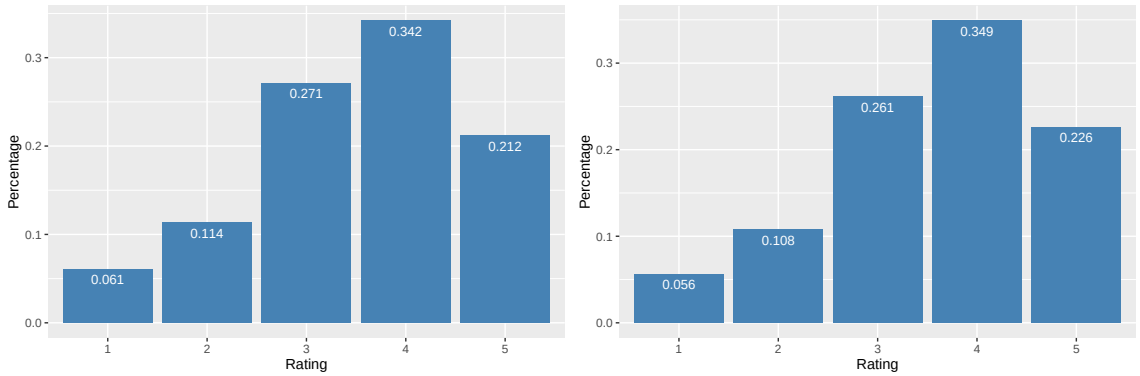

Figure 1: Rating frequency plot for Movie-lens data: ML-100k on the left and ML-1M on the right.

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.

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 reg-

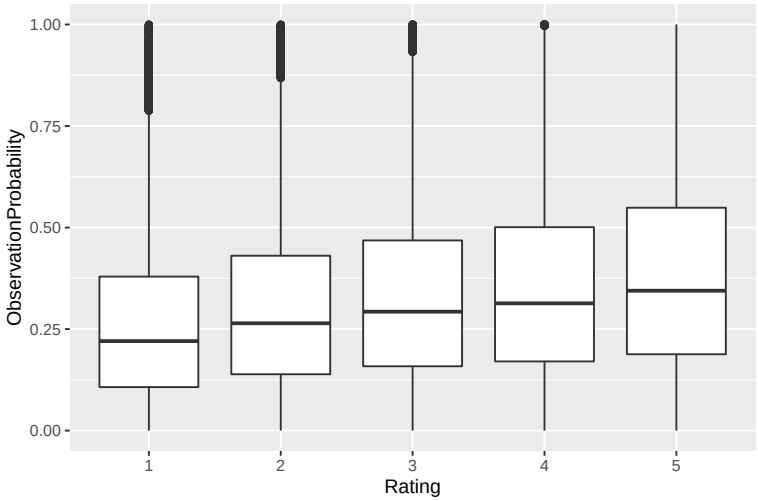

Figure 2: Box plot of the estimated observation probabilities by rating for the ML-1M data.

ularized SVD method 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/`

Table 5: 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

`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),

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

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

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 5 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

Table 6: 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

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 6 summarizes the performance of different methods on the cold start problems in the two data sets. Unsurprisingly, the proposed method

Table 7: 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

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 7 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.

3.5 Hotel Data Analysis

Antognini and Faltings (2020) collect 50,264,531 hotel reviews from Trip Advisor in nineteen years all over the world. More than 21 million users and 365 thousands hotels are involved in this data set and more than 99.9% observations are missing. This dataset provides another opportunity to explore the performance of the proposed prediction method in a sparser setting.

In this section, we filter the original dataset so that the selected hotels are all located in the west regions of the United States, which involves California, Washington, Nevada, and Oregon. The inclusion criteria also require that the users need to rate at least 20 hotels and the hotels need to have at least 50 ratings. We comprise the data size here because methods such as gSVD and 1BITMC-rSVD have quite heavy computational burden when the sample size grows. At the end, the dataset has 6,273 hotels and 2,191 users, comparable to the movie-lens 1M data. With 51,606 observations, the sparsity rate is 0.37%. The observed average rating is 3.89. Figure 3 shows the distribution of the observed scores.

To explore whether entries are missing uniformly at random, we first estimate the observation probabilities by applying the soft-thresholding SVD method on the binary recording matrix R. Figure 4 shows the Box plots of the estimated observation probabilities for each of the rating category. The average observation probabilities seem to be different in each rating and rating 4 has the highest probability to be observed.

We then compare the performance of different methods on this sparse hotel dataset. Table 8 shows the test set RMSE and MAE results when we partition the dataset into 90% training and 10% testing. First, we observe that the de-bias Zero-Imputation-1 method performs the best among all competitors, and IBITMC-

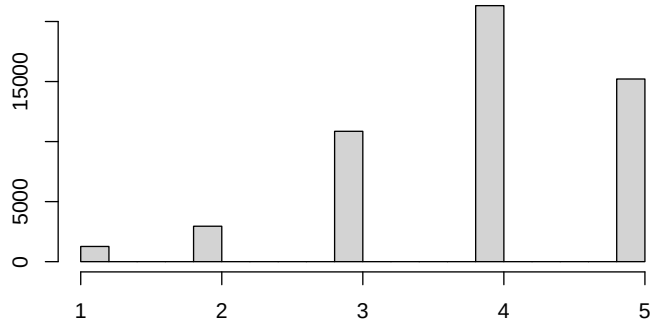

Figure 3: Histogram of the hotel recommendation set. X-axis represents the rating. Y-axis represents the number of observations.

rSVD method also performs well. The gSVD method is not as good, partly because this method is hard to tune. Second, the item-impute and user-impute method are not bad. In the movie-lens data, we see that these two straightforward imputation method have a much larger estimation error comparing to other approaches. Each person's criteria for rating hotels may not be as much different as rating movies. Therefore Recommendation Systems may be more useful in areas where people can have more subjective evaluation criteria.

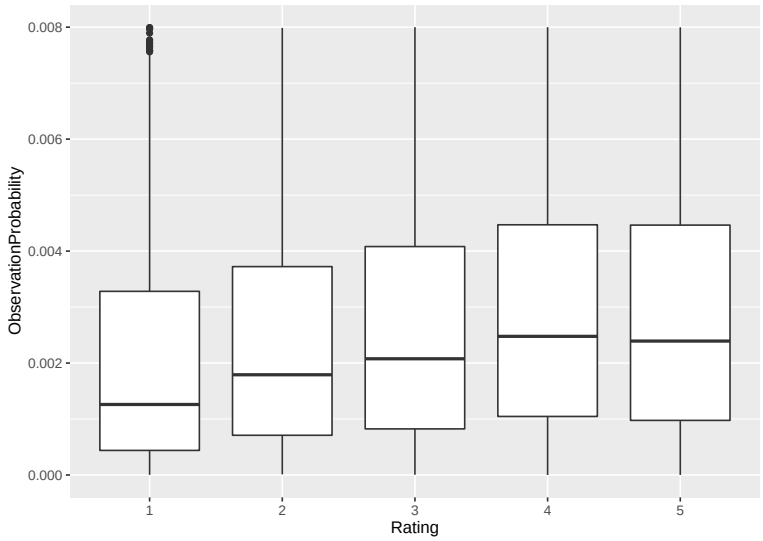

Figure 4: Box plot of the estimated observation probabilities by rating for the hotel data.

3.6 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 $\mathcal{N}((0.5 * 1_6, -0.1 * 1_6)^T, \Sigma)$, where $\Sigma_{i,j} = 0.6^2 I\{i = j\}$. The items'

Table 8: RMSE and MAE results from different methods on the hotel dataset

	RMSE	MAE
Zero-impute	.9201	.6801
Zero-impute-1	.8943	.6869
rSVD	.9355	.7140
1BITMC-rSVD	.8999	.6928
gSVD	.9275	.7121
ItemImpute	.9538	.7378
UserImpute	.9529	.7342

position $\{v_j\}$ are generated by $\mathcal{N}((0.5 * 1_6, 0.1 * 1_6)^T, \Sigma)$. 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 9: 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 10: 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)

Table 9 shows the result for the unobserved entries and Table 10 shows the result for the cold start problem with sample size $(n, m) = (5000, 2500)$. The results for other 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.

3.7 Appendix 1: Proofs

Proof of Theorem 3.2.3. First consider

$$\frac{1}{mn}\|\widehat{\mathrm{A}} - \mathrm{P}\|_F^2, \tag{14}$$

where $\widehat{\mathrm{A}}$ is soft-threshold estimator, $\mathrm{P} = \mathbb{E}(\mathrm{A})$ is the population parameter matrix, and $\|\cdot\|_F$ denotes the matrix Frobenius norm. Here A is a general notation for the truncation matrix $A^{(k)}$ or $A_{(k)}$. The proof of this part mainly follows Lemma 1 in Xu (2018). Let the error matrix $\mathrm{E} = \mathrm{A} - \mathrm{P}$ and let $\|\mathrm{E}\|$ denote the spectral norm of E. With the notation that $\mathrm{P} = \delta_{n,m}\widetilde{\mathrm{P}}_{i,j}$, we have $\mathrm{Var}(\mathrm{E}_{i,j}) = \delta_{n,m}\widetilde{\mathrm{P}}_{i,j} - \delta_{n,m}^2\widetilde{\mathrm{P}}_{i,j}^2 \leq \delta_{n,m}$. Let σ_r and $\sigma_r(\mathrm{A})$ be the r-th singular values of $\widetilde{\mathrm{P}}$ and A. By Lemma 2 in Xu (2018),

we know that there exist some positive constants c_1 and η', such that the following event happens with probability at least 1-n^{-c_1}.

$$Event = \{\|\mathrm{E}\| \leq \eta'\sqrt{\delta_{n,m}n}\}. \tag{15}$$

Note that to apply this lemma, we need the assumption that $\delta_{n,m}$ is lowered bounded by $c_2\frac{\log(n)}{n}$ for some positive constant c_2, i.e., $\delta_{n,m} \geq c_2\frac{\log(n)}{n}$.
On (15), consider the singular value threshold for some positive constant c_0,

$$\lambda = (1+c_0)\eta'\sqrt{\delta_{n,m}n}, \tag{16}$$

which means we only keep the singular values of A that are greater than λ for the soft-threshold procedure and $\|\mathrm{E}\| \leq \frac{1}{1+c_0}\lambda$. Consider

$$\ell = \sup\{r : \delta_{n,m}\sigma_r \geq \frac{c_0}{1+c_0}\lambda\}. \tag{17}$$

If $\ell = m$, it is easy to check the result. Now assume $\ell < m$, by Weyl's Theorem,

$$\sigma_{\ell+1}(\mathrm{A}) \leq \delta_{n,m}\sigma_{\ell+1} + \|\mathrm{E}\| < \lambda,$$

which implies the rank of $\widehat{\mathrm{A}}$ is bounded by ℓ. Let P_ℓ denote the best rank ℓ approximation to P, then

$$\begin{aligned}
\|\widehat{\mathrm{A}} - \mathrm{P}\|_F^2 &\leq 2\|\widehat{\mathrm{A}} - \mathrm{P}_\ell\|_F^2 + 2\|\mathrm{P}_\ell - \mathrm{P}\|_F^2 \\
&\leq 4\ell\|\widehat{\mathrm{A}} - \mathrm{P}_\ell\|^2 + 2\delta_{n,m}^2\sum_{i=\ell+1}\sigma_i^2 \\
&\leq 16\ell\lambda^2 + 2\delta_{n,m}^2\sum_{i=\ell+1}\sigma_i^2 \\
&\leq 16\min_{0\leq r\leq m}\{r\lambda^2 + (\frac{1+c_0}{c_0})^2\sum_{i=r+1}\delta_{n,m}^2\sigma_i^2\}.
\end{aligned}$$

The second to last inequality holds since

$$\|\widehat{\mathrm{A}} - \mathrm{P}_\ell\| \leq \|\widehat{\mathrm{A}} - \mathrm{A}\| + \|\mathrm{A} - \mathrm{P}\| + \|\mathrm{P} - \mathrm{P}_\ell\| \leq 2\lambda. \tag{18}$$

The last inequality holds since $\delta_{n,m}\sigma_{\ell+1} \leq \frac{c_0}{1+c_0}\lambda$ and by the definition that the last line in inequality has minimum value at ℓ. Therefore, on event (15), there exist some constants C_1, C_2, such that

$$\frac{1}{mn}\|\widehat{\mathrm{A}} - \mathrm{P}\|_F^2 \leq \frac{C_1 \min\limits_{0\leq r\leq m}\{r\lambda^2 + C_2 \sum\limits_{i=r+1} \delta_{n,m}^2\sigma_i^2\}}{mn}. \tag{19}$$

Recall that for each rating k, we recover the upper probability

$$\begin{aligned}
\widehat{S}_{i,j}^k =& \frac{\widehat{A}_{i,j}^{(k)}}{\max\{\widehat{A}_{i,j}^{(k)} + \widehat{A}_{(k)i,j}, \varepsilon_{n,m}\}} \\
=& \frac{\mathbb{E}(A_{i,j}^{(k)})}{\mathbb{E}(A_{i,j}^{(k)}) + \mathbb{E}(A_{(k);i,j})} + f_x(\xi,\eta)(\widehat{A}_{i,j}^{(k)} - \mathbb{E}(A_{i,j}^{(k)})) + \\
& f_y(\xi,\eta)(\max\{\widehat{A}_{i,j}^{(k)} + \widehat{A}_{(k);i,j}, \varepsilon_{n,m}\} - \mathbb{E}(A_{i,j}^{(k)}) - \mathbb{E}(A_{(k);i,j})),
\end{aligned}$$

where $f(x,y) = \frac{x}{y}$, $f_x(x,y) = \frac{1}{y}$, $f_y(x,y) = \frac{-x}{y^2}$ and $(\xi,\eta)^T$ is some point in the line segment between the true value and the estimated value, i.e. there exists some value t between 0 and 1 such that $[\xi,\eta]^T = t[\mathbb{E}(A_{i,j}^{(k)}), \max\{\mathbb{E}A_{i,j}^{(k)} + \mathbb{E}\widehat{A}_{(k);i,j}, \varepsilon_{n,m}\}]^T + (1-t)[\widehat{A}_{i,j}^{(k)}, \max\{\widehat{A}_{i,j}^{(k)} + \widehat{A}_{(k);i,j}, \varepsilon_{n,m}\}]^T$. The expectation element $\mathbb{E}_{i,j}$ corresponds to $\mathrm{P}_{i,j}$ that appeared previously. The absolute value of two partial derivatives are bounded by $\frac{1}{\eta}$, since $\frac{\xi}{\eta^2} \leq \frac{1}{\eta}$.

Note that η is a point between true observation probability and the estimated probability. By the assumption that the true value is lower bounded by $c\delta_{n,m}$ and

assumption that $\varepsilon_{n,m}$ is $c'\delta_{n,m}$ ($c' < c$), the partial derivatives are upper bounded by $\frac{1}{c_3\delta_{n,m}}$ for some constant c_3. So the overall MSE is

$$\frac{1}{mn}\sum_{i,j}(\widehat{S}^k_{i,j} - P(S_{i,j} \geq k))^2 \leq \min_{0\leq r\leq m}\{\frac{C_3 r}{m\delta_{m,n}} + \frac{C_4\sum_{i=r+1}\sigma_i^2}{mn}\}. \tag{20}$$

□

Proof of Corollary 3.2.8. From the proof in theorem 3.2.3, we know that for the minimum point ℓ, we have $\delta_{n,m}\sigma_\ell \geq c\sqrt{\delta_{n,m}n}$ and $\delta_{n,m}\sigma_{\ell+1} < c'\sqrt{\delta_{n,m}n}$. Use the assumption that $\sigma_\ell \asymp \frac{\sqrt{mn}}{\ell^\alpha}$, we have $\ell \asymp (m\delta_{n,m})^{1/(2\alpha)}$. Therefore the first term $\frac{\ell}{m\delta_{n,m}}$ in MSE is in the order of $(\frac{1}{m\delta_{n,m}})^{1-\frac{1}{2\alpha}}$. For the singular value summation term, using the fact that

$$\sum_{r=\ell+1}^{n\wedge m} r^{-2\alpha} = O(\frac{1}{\ell^{2\alpha-1}}),$$

we conclude that the second term in MSE is in the order of $(\frac{1}{m\delta_{n,m}})^{1-\frac{1}{2\alpha}}$. □

Proof of Theorem 3.3.2. Suppose the corresponding graphon W admits strong SVD in the form of

$$W(s,t) = \sum_i \lambda_i\phi_i(s)\psi_i(t).$$

Let s and t be i.i.d. $Unif(0,1)$, and let $u(s) = [u_1(s),\ldots,u_r(s),\ldots]^T$ in which $u_r(s) = \sqrt{\lambda_r}\phi_r(s)$, and $v(t) = [v_1(t),\ldots,v_r(t),\ldots]^T$ in which $v_r(t) = \sqrt{\lambda_r}\psi_r(t)$. The norm of each random variable is finite by the strong decomposition assumption. Moreover, $W(s,t) = u(s)^T v(t)$ almost everywhere. The sampling distribution generated by W with dimension n, m is, by Aldous-Hoover Theorem, first samples $s_1,\ldots,s_n$ and $t_1,\ldots,t_m$ from i.i.d. $Unif(0,1)$, then generate Bernoulli random variables with parameters $W(s_i,t_j)$. This is, by the construction, the same as first independently sampling from the BGRD distribution to get $u(s_i)$ and $v(t_j)$, then

form the exchangeable arrays by their inner-products, where F_1 is the probability measure induced by $u(s) : [0,1] \to K$ with $s \sim Unif(0,1)$ and F_2 is the probability measure induced by $v(t) : [0,1] \to K$ with $t \sim Unif(0,1)$. □

Proof of Theorem 3.3.5. ($\Leftarrow$) Since orthogonal transform maintains inner product, this direction is clear.

($\Rightarrow$) By Proposition 3.5 in Lei (2021), for a distribution F_1 on a separable Hilbert space K, there exists an inverse transform sampling, i.e., a measurable function $u : [0,1] \to K$ such that if $s \sim Unif(0,1) \Rightarrow u(s) \sim F_1$. Therefore, for a sampling point in BGRD $F = F_1 \times F_2$, we can write it as $(u(s), v(t))$, where u and v are inverse transform samplings, and $s, t \sim Unif(0,1)$. By equally-weighted assumption and without loss of generality, we assume that (u, v) have the same diagonal second moment matrix Λ. Analogously, we denote a sample point from G by $(\widetilde{u}(s), \widetilde{v}(t))$, and their moment matrix $\widetilde{\Lambda}$.

Define the graphon W corresponding to F as

$$\begin{aligned} W(s,t) &= \langle u(s), v(t) \rangle \\ &= \sum_j \lambda_j \lambda_j^{-1/2} u_j(s) \lambda_j^{-1/2} v_j(t), \end{aligned}$$

where λ_j is the jth diagonal value in Λ. Note that the above is the SVD decomposition of W. We can define $\widetilde{W}$ similarly for G. Since F and G lead to the same sampling distribution of binary arrays, we have

$$W(s,t) \overset{d}{=} \widetilde{W}(s,t).$$

By Theorem 4.1 in Kallenberg (1989), we have $\forall\ j$, $\lambda_j = \widetilde{\lambda}_j$ and there exists unitary operator Q with $Q_{j,j'} = 0$ for $\lambda_j \neq \lambda_{j'}$, such that for any measurable set A,

$$P(\Lambda^{-1/2} u \in A) = P(Q \Lambda^{-1/2} \widetilde{u} \in A),$$

$$P(\Lambda^{-1/2}v \in A) = P(Q\Lambda^{-1/2}\widetilde{v} \in A).$$

Therefore

$$\begin{aligned} P(u \in A) &= P(\Lambda^{-1/2}u \in \Lambda^{-1/2}A) \\ &= P(Q\widetilde{u} \in A). \end{aligned}$$

The same result holds for v. Therefore $F \overset{o.t.}{=} G$. □

3.8 Appendix 2: Further Results on Model Tuning and the Sensitivity Analysis of the Minimum Observation Probability

In our Algorithm 1 in Section 3.2, we only take two input parameters. One is the thresholding dimension p, which is a tuning parameter selected by 5-fold cross-validation, and the other is the lower bound ε on the missing probability.

The soft-thresholding values are then set to be $\lambda^k = \widehat{\sigma}^k_{p+1}$ and $\lambda_k = \widehat{\sigma}_{k,p+1}$, where $\widehat{\sigma}^k_{p+1}$ and $\widehat{\sigma}_{k,p+1}$ are the $(p+1)$-th singular value of $A^{(k)}$ or $A_{(k)}$. In our numerical experiments, tuning each λ^k and λ_k individually does not significantly reduce the error. The comparisons in our simulation settings and in the movie-lens data are shown in Table 11 and Table 12. 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 $(n, m) = (3000, 1500)$.

Table 11: Threshold dimension tuning comparison in simulation setting.

	$(n,m)=(1500,800)$		$(n,m)=(3000,1500)$		$(n,m)=(5000,2500)$	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
One p for all	.9747	.7422	.9127	.7039	.8566	.6558
Independently-tuned λ_k and λ^k	.9707	.7404	.9094	.7148	.8536	.6555

Table 12: Threshold dimension tuning comparison in Movie-lens data set.

	ML-100K		ML-1M	
	RMSE	MAE	RMSE	MAE
One p for all	.9070	.7212	.8526	.6774
Independently-tuned λ_k and λ^k	.9049	.7168	.8505	.6734

Table 13: Selection of ε sensitivity analysis in simulation setting.

	$(n,m)=(1500,800)$		$(n,m)=(3000,1500)$		$(n,m)=(5000,2500)$	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
$\varepsilon=10^{-3}$	.9945	.8097	.9343	.7619	.8776	.7126
$\varepsilon=10^{-4}$	.9754	.7439	.9138	.7036	.8571	.6597
$\varepsilon=10^{-5}$	.9752	.7392	.9130	.6991	.8559	.6556
$\varepsilon=10^{-6}$	.9752	.7388	.9129	.6987	.8558	.6552

Table 14: Selection of ε sensitivity analysis in Movie-lens data set.

	ML-100K		ML-1M	
	RMSE	MAE	RMSE	MAE
$\varepsilon = 10^{-3}$	.9013	.7134	.8491	.6746
$\varepsilon = 10^{-4}$	.9045	.7155	.8534	.6787
$\varepsilon = 10^{-5}$	.9053	.7163	.8555	.6811
$\varepsilon = 10^{-6}$	.9054	.7164	.8561	.6816

For the selection of ε, a sensitivity analysis for our simulation setting and the movie-lens data are included in Table 13 and Table 14. We can see that the results are very similar for $\varepsilon = 10^{-4}, 10^{-5}$ and 10^{-6}. We used 10^{-4} because we regarded this as a small enough lower bound for the observation rate in our data example and the simulated example. The data is allowed to be more sparse (higher missing rate) as n and m grow, and accordingly the choice of ε should match the approximate sparsity level of the data.

4.0 Applications in Network Regression with Missing Edges

4.1 Introduction

In this chapter, we mainly focus on the application of the Zero-imputation method to network regression models in which the social network matrix has missing entries. Most social network inferences rely on the assumption that all links are observed and reliable, which may not hold true in practice (Handcock et al., 2007). In practice, sometimes the missing entries in social networks are not differentiated from zero entries, i.e., entries really with no links. In other words, all missing edges are filled in with 0 and proceed as if the data is complete. However, this method may lead to severe bias in network inference. In this chapter, we explore the possibility of using Zero-imputation method to improve the inference of network regression models when the observed network matrix is incomplete. In Section 4.2, we compare the proposed imputation method with the naive Always-Zero method under the network autoregressive model (NAR), and in Section 4.3, we investigate the network autoregressive error model (NARE).

4.2 Network Autoregressive Model

Let n be the total sample size, and $A \in \mathbb{R}^{n\times n}$ be the complete social network matrix. Let $Y \in \mathbb{R}^n$ be the response variable, and $X \in \mathbb{R}^{n\times p}$ be the covariate matrix.

The network autoregressive (NAR) model in matrix form is

$$Y = \rho AY + X\gamma + \varepsilon. \tag{21}$$

In this model, individual i's response Y_i is not only affected by its own nodal effect $X_i^T\gamma$, but also impacted by other people's responses through the network connection (ρAY). Here ρ is the coefficient that measures the network effect. The NAR model has been used to study the peer pressure effect in many areas, such as financial activities (Zhu et al., 2020) and school achievements (Bramoullé et al., 2009). In spatial statistics, we can represent the geographical connections using an adjacency matrix G such that if two points (i, j) are adjacent, then entry $G_{i,j} = 1$ otherwise 0. Because of the similarity between matrix G and network matrix A, NAR model is closely related to the Spatial Auto Regression (SAR) model in spatial econometrics (Anselin and Bera, 1998; LeSage and Pace, 2009). Network autoregressive models have been extended to deal with times series data in which we observe time-stamped responses $\{Y_t\}$ for each node in the network (Zhu et al., 2017, 2021). One may further impose specific structures to better characterize the data dependence. Examples include group classified responses (Zhu and Pan, 2020) and community structured network A (Chen et al., 2020). One can also explore the social network effect in applications where the response is a time to event variable (Su et al., 2019). All these methods assume that the network matrix A is fully observed.

Since the network term ρAY introduces a non-zero correlation among subjects, ordinary least squares estimator can not be used here (Anselin, 1988). Instead one can use Maximum Likelihood Estimator (MLE) to consistently estimate model parameters (Lee, 2004).

In practice, social network data is often collected by sample surveys and therefore involves non-responses and missing data. For example, in some hard-to-reach population such as injecting drug users, network data is collected through respondent-driven sampling techniques. As a result, one can have access to only a portion of network entries (Gile, 2011). Ignoring missing ties may lead to severe bias in network social network analysis (Handcock et al., 2007; Huisman, 2009).

Network matrix is usually constructed by recording the observed connection pairs. Entry $A_{i,j} = 0$ could imply that two people have no connections. It is also possible that this entry is in fact missing because each person only reported a limited number of connections. For missing entries, we mainly investigate two approaches. The first method (Always-Zero) is to fill all missing entries with zero as if the network matrix is fully observed. This is a naive approach but often used in practice without explicitly considering the missing issue. The second approach is to apply the Zero-imputation method proposed in Chapter 3 to fill in all missing entries before carrying out subsequent analysis. In the NAR model, the network effect ρ is of particular interest. We explore the power and type I error of the inference for ρ, using the Zero-imputation method and the Always-zero method. We also explore the power and Type I error for other model parameters γ.

4.2.1 Simulations of the Network AutoRegressive Model

We set the sample size to be 150, comparable to the data example in Section 4.2.2, and we include three covariates, one standard normal, one uniform on 0 and 1, and one binary variable with Bernoulli parameter 0.7. The NAR model γ parameters are $(1.5, 0.2, 0, 0.50)^T$, and the error variance is 1. For each $i \neq j$, element $A_{i,j}$ is a Bernoulli random variable, generated independently by parameter $\frac{\delta}{\|x_i - x_j\|_2}$. Here δ

is a parameter that controls the network density. The quantity in the denominator is the Euclidean distance between subject i and subject j's covariates. If two people are close in covariate's Euclidean distance, then their connection probability will be large. We fixed the overall network density at 0.3 in the simulations and our simulation repetition number is 1000. NAR model fittings are implemented through R package spdep.

For missing mechanism in this simulation, we let $d_0 \in \{0.15, 0.25, 0.35\}$ be the parameter that controls the overall missing rate. The missing probability for each entry (i, j) is chosen as $\frac{d_0}{\|x_{i,1:2} - x_{j,1:2}\|_2}$. Here $x_{i,1:2}$ means the first and the second covariate for subject i. Therefore, the missing rates are related to a subset of individual's covariates. We mainly compare our Zero-Imputation method with the Always-Zero method. As a reference, we also present the results of MLE using the complete non-missing data.

We first explore the case when there is no network autoregressive effect, i.e., $\rho = 0$ in (21). Table 15 shows the Type I results of ρ. All methods can control the Type I error pretty well under different missing rates. When there is in fact no network effect, the estimations for model parameter γ are not affected much by the imputation method, as shown in Table 16. This result is as expected.

Table 15: Simulated NAR ρ estimation table for different methods when population $\rho = 0$.

Missing Rate	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\rho}$ ($\times 10^{-4}$)	TypeI	$\widehat{\rho}$	TypeI	$\widehat{\rho}$	TypeI	$\widehat{\rho}$	TypeI
.199	1.4	.048	-.7	.053	-.8	.051	-.7	.049
.310	0.2	.052	-1.0	.054	.7	.055	-1.0	.055
.407	-1.2	.051	-1.7	.047	-1.9	.047	-1.9	.047

Table 16: Simulated NAR γ_1 estimation table when population $\rho = 0$. The true $\gamma_1 = 0.2$.

Missing Rate	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\gamma}_1$	Power	$\widehat{\gamma}_1$	Power	$\widehat{\gamma}_1$	Power	$\widehat{\gamma}_1$	Power
.199	.199	.686	.199	.681	.201	.681	.199	.680
.310	.199	.672	.195	.673	.197	.662	.196	.673
.407	.199	.662	.198	.662	.200	.661	.199	.660

When population $\rho = 0.01$, Table 17 displays the NAR ρ estimation results under different missing rates. The Zero-imputation-1 method has a much higher power to detect an auto-regressive signal over the Always-Zero method, especially when the missing rate is high.

Table 18 shows the inference results for a non-zero parameter $\gamma_1 = 0.2$. Our method outperforms the Always-Zero method with a higher power to detect the

signal. Table 19 shows the inference results for $\gamma_2 = 0$, The zero-imputation-1 method has the much smaller Type I errors than the Always-Zero method but still shows an inflation in Type I error when the missing rate is high. This result is also understandable. For NAR model in the matrix form listed below, the estimation of γ is affected by an additional term $(I - \rho A)^{-1}$

$$Y = (I - \rho A)^{-1} X\gamma + (I - \rho A)^{-1}\varepsilon.$$

Therefore, inference for parameter γ is found to be sensitive to the value of $\widehat{\rho}$. When the missing rate is high, the zero-imputation method can alleviate the missing problem but not completely solve the missing problem.

Table 17: Simulated NAR ρ estimation table for different methods when population $\rho = 0.01$.

Missing Rate	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\rho}$	Power	$\widehat{\rho}$	Power	$\widehat{\rho}$	Power	$\widehat{\rho}$	Power
.199	.009	.985	.009	.968	.011	.932	.009	.961
.310	.010	.988	.008	.960	.011	.782	.008	.960
.407	.010	.982	.008	.945	.009	.527	.008	.943

Table 18: Simulated NAR γ_1 estimation table when population $\rho = 0.01$. The true $\gamma_1 = 0.2$.

Missing Rate	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\gamma}_1$	Power	$\widehat{\gamma}_1$	Power	$\widehat{\gamma}_1$	Power	$\widehat{\gamma}_1$	Power
.199	.202	.710	.201	.678	.210	.620	.201	.675
.310	.200	.690	.196	.660	.196	.600	.196	.654
.407	.201	.690	.197	.646	.195	.596	.196	.640

Table 19: Simulated NAR γ_2 estimation table when population $\rho = 0.01$. The true $\gamma_2 = 0$.

Missing Rate	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\gamma}_2$	TypeI	$\widehat{\gamma}_2$	TypeI	$\widehat{\gamma}_2$	TypeI	$\widehat{\gamma}_2$	TypeI
.199	.005	.051	.061	.058	.065	.086	.061	.061
.310	.003	.053	.064	.077	.120	.158	.064	.080
.407	.009	.051	.075	.086	.143	.196	.075	.086

4.2.2 Teenage Friends and Lifestyle Study Data under Auto-Regression Model

The teenage Friends and Lifestyle Study (Bush et al., 1997; Michell and West, 1996) aimed to study the influence of social network (peer pressure effect) on the

alcohol and drug use behaviors among teenagers. The data were recorded during two years period from 1995 to 1997. It contains 160 students and the friendship network is formed by asking students to name up to six friends. The observed network density is 2.29%. Researchers also collected several important covariates that may affect the student's alcohol behavior, including age, tobacco use (order scale variable from 1 to 3), drug use (order scale variable from 1 to 4), and money (pocket money per week). Table 20 displays the basic summary of these variables and Figure 5 shows the histogram of the students' alcohol consumption. The whole data set is public and is available at `https://www.stats.ox.ac.uk/~snijders/siena/`.

Table 20: Variable Summary of the teenage Friends and Lifestyle Study data

Variable	Summary
Age	Min: 12.40 Median:13.40 Max: 14.60
Tobacco	Mean:1.26
Cannabis	Mean:1.49
Money	Min: 0 Mean:9.21 Max: 40

Although the original social network data is complete, it is also possible that some entries are indeed missing since students can only name up to six friends. In this data analysis, for illustration purpose, we remove a portion of the entries and treat them as missing data. We can investigate how different imputation methods perform on the incomplete synthesis dataset. For simplicity, we set the missing probability for each entry to be $p \in \{0.10, 0.20\}$. To better compare the results, we include the complete

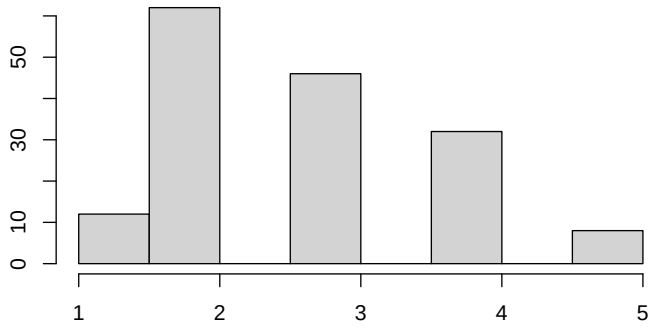

Figure 5: Histogram of the alcohol consumption in the teenage Friends and Lifestyle Study data.

data analysis result on the first row in Table 21 and Table 22. We have several interesting findings here. First, we can see from Table 21 that the network effect ρ can concluded to be significant at level $\alpha = 0.05$ using the complete data set. Our Zero-imputation method has consistent results in the network effect detection with the complete data analysis. However, the Always-Zero method produces a different conclusion if $\alpha = 0.05$ is used. Second, from Table 22, we can see that under the complete data analysis, Tobacco and Cannabis are found to be significant predictors for teenager's alcohol consumption under NAR model, and Cannabis has a very small p-value. The results are still consistent when different missing imputation methods are used for the missing data.

Table 21: NAR model results for the estimation of ρ with missing teenage Friends and Lifestyle Study data. NAR model is fitted with selected variables.

Missing Rate	Method	$\widehat{\rho}$	P-value
0	Complete Analysis	.025	.040
.10	Zero-impute-1	.021	.047
	Always-Zero	.022	.080
	Zero-impute	.022	.040
.20	Zero-impute-1	.025	014
	Always-Zero	.022	.089
	Zero-impute	.026	.010

Table 22: NAR Model results for Age, Tobacco, Cannabis, and Money with missing teenage Friends and Lifestyle Study data.

Missing Rate	Method	Age	P-value	Tobacco	P-value	Cannabis	P-value	Money	P-value
0	Complete Analysis	-.073	.734	.288	.042	.435	0	.016	.086
.10	Zero-impute-1	-.067	.757	.303	.033	.424	0	.016	.086
	Always-Zero	-.075	.731	.292	.040	.434	0	.016	.094
	Zero-impute	-.066	.759	.303	.033	.425	0	.017	.082
.20	Zero-impute-1	-.082	.702	.298	.034	.434	0	.016	.100
	Always-Zero	-.075	.731	.301	.034	.428	0	.016	.103
	Zero-impute	-.075	.726	.302	.032	.431	0	.016	.096

4.3 Network Autoregressive Error Model

With the same notation in Section 4.2, we now study the following network autoregressive error model (NARE),

$$\begin{cases} Y = X\beta + u \\ u = \lambda A u + \varepsilon \end{cases} \tag{22}$$

In the above model, $\beta \in \mathbb{R}^p$ is the model parameter of interests, u is the marginal error term, assumed to have an error-autoregressive structure with coefficient λ, and ε follows $\mathcal{N}(0, \sigma^2)$. We can also write the model in the following marginal form,

$$Y = X\beta + u, u \sim \mathcal{N}(0, \sigma^2(I - \lambda A)^{-2}).$$

Parameter λ characterizes the network autoregressive effect of the error terms. If we observe the complete network matrix A, then we can first use Maximum Likelihood (ML) to estimate model parameters σ and λ. We use $\widehat{\sigma}$ and $\widehat{\lambda}$ to denote the ML estimators.

Let

$$\widehat{\Sigma} = \widehat{\sigma}^2(I - \widehat{\lambda}A)^{-2}$$

be the plug-in estimator for marginal variance-covariance of u. Parameter β then can be estimated by the Generalized Least Squares (GLS) estimator,

$$\widehat{\beta} = (X^T\widehat{\Sigma}^{-1}X)^{-1}X^T\widehat{\Sigma}^{-1}Y.$$

Theoretically speaking, estimator $\widehat{\beta}$ is always unbiased even if the variance-covariance structure is not correctly identified, but the inference of β may be affected if the network effect is not correctly estimated. If there is no autoregressive effect in errors,

i.e., $\lambda = 0$, then model (22) will reduce to the linear regression model. If we treat A as the spatial connection, it then becomes the Spatial Error Model, see Anselin et al. (2001) for detailed discussions.

Following the notations in Chapter 3, we will explore the Zero-imputation method and the Always-zero method to fill in missing entries if the network is incomplete. Let $\widehat{A}$ be the imputed network matrix. We will then conduct the subsequent analysis as if this is the observed complete network A, and then compare the results with the complete data analysis.

4.3.1 Simulations of the Network Autoregressive Error Model

Let the sample size be 150 and we include three covariates, one standard normal, one uniform on 0 and 1, and one binary variable with Bernoulli parameter 0.7. The model parameters are $(\beta, \sigma)^T = (1.5, 0.05, 0, 0.50, 0.2)^T$. For each $i \neq j$, element $A_{i,j}$ is a Bernoulli random variable, generated independently by parameter $\frac{\delta}{\|x_i - x_j\|_2}$. Here δ is the parameter that controls the network density. The quantity in the denominator is the Euclidean distance between subject i and subject j's covariates. If two people are close in covariate's Euclidean distance, then their connection probability will be large. We fixed the overall network density at 0.12 in the simulations. All simulations are repeated 1000 times. NARE model fittings are implemented through R package sna.

For missing mechanism in this simulation, we let $d_0 \in \{0.10, 0.15, 0.20\}$ be the parameter that controls the overall missing rate. The missing probability for each entry (i, j) is chosen as $\frac{d_0}{\|x_{i,1:2} - x_{j,1:2}\|_2}$. To better understand the estimation impact of missingness, we first explore the case when there is no error-autoregressive effect, i.e., $\lambda = 0$. Table 23 and Table 24 display the estimation and the inference results under

this case. Type I errors are inflated a little bit for all methods. This may be due to the small sample size. It is known that restricted maximum likelihood (REML) may provide better results than MLE in autoregressive structure estimations, but it is not implemented in the package we use. All methods are similar in the inference of β when $\lambda = 0$, which is expected since the model then has nothing to do with network matrix A.

Table 23: Simulated NARE λ estimation table for different methods when population $\lambda = 0$.

Missing Rate	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\lambda}$	TypeI	$\widehat{\lambda}$	TypeI	$\widehat{\lambda}$	TypeI	$\widehat{\lambda}$	TypeI
.128	-.006	.066	-.006	.062	-.008	.064	-.006	.062
.187	-.007	.066	-.006	.069	-.008	.070	-.006	.067
.242	-.007	.065	-.007	.066	-.009	.064	-.007	.068

Table 24: Simulated NARE β_1 estimation table when population $\lambda = 0$. The true $\beta_1 = 0.05$.

Missing Rate	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\beta}_1$	Power	$\widehat{\beta}_1$	Power	$\widehat{\beta}_1$	Power	$\widehat{\beta}_1$	Power
.128	.049	.938	.049	.937	.049	.936	.049	.937
.187	.050	.939	.050	.939	.050	.940	.050	.939
.242	.049	.940	.049	.933	.049	.932	.049	.933

When $\lambda = 0.07$, Table 25 displays results of λ estimation. In general, the Zero-impute method higher power over the Always-Zero method. However, we still see some bias in the estimation of λ even the Zero-imputationis used. Table 26 shows the results for the non-zero β_1. Although point estimations for β_1 are all unbiased for different methods, our method has a higher power. Table 27 shows the results for $\beta_2 = 0$, the zero-imputation-1 method has smaller Type I errors than the Always-Zero method, although it shows a Type I error inflation when the missing rate is high. Overall, the estimation and inference of λ in NARE is not as good as the parameter ρ in the NAR model. NAR is more popular and more commonly studied in the literature.

Table 25: Simulated NARE λ estimation table for different methods when population $\lambda = 0.07$.

Missing Rate	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\lambda}$	Power	$\widehat{\lambda}$	Power	$\widehat{\lambda}$	Power	$\widehat{\lambda}$	Power
.128	.067	.866	.056	.716	.047	.672	.057	.712
.187	.067	.866	.052	.672	.042	.592	.052	.678
.242	.067	.874	.045	.612	.051	.550	.045	.612

Table 26: Simulated NARE β_1 estimation table when population $\lambda = 0.07$. The true $\beta_1 = 0.05$.

Missing Rate	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\beta}_1$	Power	$\widehat{\beta}_1$	Power	$\widehat{\beta}_1$	Power	$\widehat{\beta}_1$	Power
.128	.049	.826	.049	.798	.050	.792	.049	.794
.187	.049	.828	.049	.788	.051	.778	.049	.781
.242	.051	.836	.051	.776	.051	.762	.051	.778

Table 27: Simulated NARE β_2 estimation table when population $\lambda = 0.07$. The true $\beta_2 = 0$.

Missing Probability	Complete		Zero-impute-1		Always-Zero		Zero-impute	
	$\widehat{\beta}_2$	TypeI	$\widehat{\beta}_2$	TypeI	$\widehat{\beta}_2$	TypeI	$\widehat{\beta}_2$	TypeI
.128	.005	.062	.005	.088	.005	.116	.005	.090
.187	.004	.066	.004	.098	.004	.150	.004	.102
.242	.003	.062	.001	.112	-.002	.194	.002	.114

5.0 Summary and Discussion

Prediction of unobserved links is one of the main goals in relational data analysis. In Chapter 2, we develop a nonparametric prediction method for pure cold start users following a latent position approach. Simulations and the real data analysis show that our proposed method outperforms the existing methods in classification characteristics such as AUC. In Chapter 3, we develop a Zero-imputation method under heterogeneous missing situations in Recommendation Systems to predict all missing ratings. Our algorithm has a closed form solution, scalable to large data sets and can be extended to work for the cold start prediction problems. Simulations and the real data analysis show that our Zero-imputation method has a sharper improvement in larger data sets and in cold-start problems. Most work on network regression analysis assume that the network matrix is fully observed. However, this assumption is often violated in practice. In Chapter 4, we utilize the Zero-imputation method to impute all network missing values in the context of network Autoregressive (NAR) model. Numerical experiments show that our method has a higher power than other approaches to detect the network effect.

We now discuss a possible extension of this thesis. In classic Recommendation Systems, data set can be seen as an user-item rating matrix with large portion of missing values. In Chapter 3, we discussed the Zero-imputation method for learning ratings S,

$$S : \text{Users} \times \text{Items} \to S_0$$

where S_0 is some ordered score set such as $\{1,2,3,4,5\}$. In some applications, multiple ratings and aspects are of interest. For example, in the restaurant domain users can

rate food taste, location, service as well as the overall score for the restaurant, and in hotel recommendations (Jannach et al., 2012), one can rate cleanliness, sleep quality, front-desk and so on. Therefore it is possible to explore the multi-dimensional ratings S',

$$S' : \text{Users} \times \text{Items} \to S_1 \times \cdots \times S_T.$$

Here T is the total number of aspects. The heterogeneous missing problem also appears in the multi-criteria ratings since people can choose to rate one or more particular aspect scores. It may be of interest to extend the Zero-imputation method to this multi-dimensional problem, where we need to estimate a sparse binary tensor.

6.0 Bibliography

Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008), "Mixed membership stochastic blockmodels," *Journal of machine learning research*.

Al Hasan, M. and Zaki, M. J. (2011), "A survey of link prediction in social networks," in *Social network data analytics*, Springer, pp. 243–275.

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

Anil, A., Kumar, D., Sharma, S., Singha, R., Sarmah, R., Bhattacharya, N., and Singh, S. R. (2015), "Link prediction using social network analysis over heterogeneous terrorist network," in *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, IEEE, pp. 267–272.

Anselin, L. (1988), *Spatial econometrics: methods and models*, vol. 4, Springer Science & Business Media.

Anselin, L. and Bera, A. K. (1998), "Introduction to spatial econometrics," *Handbook of applied economic statistics*, 237.

Anselin, L. et al. (2001), "Spatial econometrics," *A companion to theoretical econometrics*, 310330.

Antognini, D. and Faltings, B. (2020), "HotelRec: a Novel Very Large-Scale Hotel Recommendation Dataset," *arXiv preprint arXiv:2002.06854*.

Baldin, N. and Berthet, Q. (2018), "Optimal link prediction with matrix logistic regression," *arXiv preprint arXiv:1803.07054*.

Barzel, B. and Barabási, A.-L. (2013), "Network link prediction by global silencing of indirect correlations," *Nature biotechnology*, 31, 720.

Bennett, J., Lanning, S., et al. (2007), "The netflix prize," in *Proceedings of KDD*

cup and workshop, New York, vol. 2007, p. 35.

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

Bradley, A. P. (1997), "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern recognition*, 30, 1145–1159.

Bramoullé, Y., Djebbari, H., and Fortin, B. (2009), "Identification of peer effects through social networks," *Journal of econometrics*, 150, 41–55.

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

Bush, H., West, P., and Michell, L. (1997), "The role of friendship groups in the uptake and maintenance of smoking amongst pre-adolescent and adolescent children: Distribution of frequencies," *Glasgow, Scotland: Medical Research Council*.

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, E. Y., Fan, J., and Zhu, X. (2020), "Community network auto-regression for high-dimensional time series," *arXiv preprint arXiv:2007.05521*.

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.

Clauset, A., Moore, C., and Newman, M. E. (2008), "Hierarchical structure and the prediction of missing links in networks," *Nature*, 453, 98.

Condat, L. (2017), "Least-squares on the simplex for multispectral unmixing," *Res. Rep, GIPSA-Lab, Univ. Grenoble Alpes, Grenoble, France*.

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.

Durante, D., Dunson, D. B., and Vogelstein, J. T. (2017), "Nonparametric Bayes modeling of populations of networks," *Journal of the American Statistical Association*, 112, 1516–1530.

Fawcett, T. (2006), "An introduction to ROC analysis," *Pattern recognition letters*, 27, 861–874.

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

Gile, K. J. (2011), "Improved inference for respondent-driven sampling data with application to HIV prevalence estimation," *Journal of the American Statistical Association*, 106, 135–146.

Grover, A. and Leskovec, J. (2016), "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864.

Handcock, M. S., Raftery, A. E., and Tantrum, J. M. (2007), "Model-based clustering

for social networks," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170, 301–354.

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

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002), "Latent space approaches to social network analysis," *Journal of the american Statistical association*, 97, 1090–1098.

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

Huisman, M. (2009), "Imputation of missing network data: Some simple procedures," *Journal of Social Structure*, 10, 1–29.

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

Jannach, D., Gedikli, F., Karakaya, Z., and Juwig, O. (2012), "Recommending hotels based on multi-dimensional customer ratings," in *Information and communication technologies in tourism 2012*, Springer, pp. 320–331.

Kallenberg, O. (1989), "On the representation theorem for exchangeable arrays," *Journal of Multivariate Analysis*, 30, 137–154.

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.

Koltchinskii, V., Lounici, K., Tsybakov, A. B., et al. (2011), "Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion," *The Annals of*

Statistics, 39, 2302–2329.

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

Lazega, E. et al. (2001), *The collegial phenomenon: The social mechanisms of cooperation among peers in a corporate law partnership*, Oxford University Press on Demand.

Lee, L.-F. (2004), "Asymptotic distributions of quasi-maximum likelihood estimators for spatial autoregressive models," *Econometrica*, 72, 1899–1925.

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

Lei, J. and Rinaldo, A. (2015), "Consistency of spectral clustering in stochastic block models," *The Annals of Statistics*, 43, 215–237.

LeSage, J. and Pace, R. K. (2009), *Introduction to spatial econometrics*, Chapman and Hall/CRC.

Leskovec, J. and Mcauley, J. J. (2012), "Learning to discover social circles in ego networks," in *Advances in neural information processing systems*, pp. 539–547.

Liu, B. (2019), "Statistical learning for networks with node features," Ph.D. thesis, University of Michigan, Ann Arbor.

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

Lü, L. and Zhou, T. (2011), "Link prediction in complex networks: A survey," *Physica A: statistical mechanics and its applications*, 390, 1150–1170.

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.

Michell, L. and West, P. (1996), "Peer pressure to smoke: the meaning depends on the method," *Health education research*, 11, 39–49.

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.

Newman, M. (2018), *Networks*, Oxford university press.

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

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014), "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710.

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

Rohe, K., Chatterjee, S., and Yu, B. (2011), "Spectral clustering and the high-dimensional stochastic blockmodel," *The Annals of Statistics*, 39, 1878–1915.

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.

Su, L., Lu, W., Song, R., and Huang, D. (2019), "Testing and estimation of social network dependence with time to event data," *Journal of the American Statistical Association*.

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*.

Wei, X., Xu, L., Cao, B., and Yu, P. S. (2017), "Cross view link prediction by learning noise-resilient representation consensus," in *Proceedings of the 26th International Conference on World Wide Web*, International World Wide Web Conferences Steering Committee, pp. 1611–1619.

Wu, Y.-J., Levina, E., and Zhu, J. (2018), "Link prediction for egocentrically sampled

networks," *arXiv preprint arXiv:1803.04084*.

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

Zhao, Y., Wu, Y.-J., Levina, E., and Zhu, J. (2017), "Link prediction for partially observed networks," *Journal of Computational and Graphical Statistics*, 26, 725–733.

Zhu, X., Cai, Z., and Ma, Y. (2021), "Network functional varying coefficient model," *Journal of the American Statistical Association*, 1–12.

Zhu, X., Huang, D., Pan, R., and Wang, H. (2020), "Multivariate spatial autoregressive model for large scale social networks," *Journal of Econometrics*, 215, 591–606.

Zhu, X. and Pan, R. (2020), "Grouped network vector autoregression," *Statistica Sinica*, 30, 1437–1462.

Zhu, X., Pan, R., Li, G., Liu, Y., and Wang, H. (2017), "Network vector autoregression," *The Annals of Statistics*, 45, 1096–1123.