How to Impute Missing Ratings? Claims, Solution, and Its Application to Collaborative Filtering

Youngnam Lee
Hanyang University
Seoul, Korea
youngnam.lee@vuno.co

Sunju Park
Yonsei University
Seoul, Korea
boxenju@yonsei.ac.kr

Sang-Wook Kim
Hanyang University
Seoul, Korea
wook@hanyang.ac.kr

Xing Xie
Microsoft Research Asia
Beijing, China
xing.xie@microsoft.com

ABSTRACT

Data sparsity is one of the biggest problems faced by collaborative filtering used in recommender systems. *Data imputation* alleviates the data sparsity problem by inferring missing ratings and imputing them to the original rating matrix. In this paper, we identify the limitations of existing data imputation approaches and suggest three new claims that all data imputation approaches should follow to achieve high recommendation accuracy. Furthermore, we propose a deep-learning based approach to compute imputed values that satisfies all three claims. Based on our hypothesis that most pre-use preferences (e.g., impressions) on items lead to their post-use preferences (e.g., ratings), our approach tries to understand via deep learning how pre-use preferences lead to post-use preferences differently depending on the characteristics of users and items. Through extensive experiments on real-world datasets, we verify our three claims and hypothesis, and also demonstrate that our approach significantly outperforms existing state-of-the-art approaches.

CCS CONCEPTS

• Information systems → Recommender systems;

KEYWORDS

Recommender Systems; Collaborative Filtering; Data Sparsity; Data Imputation

ACM Reference Format:

Youngnam Lee, Sang-Wook Kim, Sunju Park, and Xing Xie. 2018. How to Impute Missing Ratings? Claims, Solution, and Its Application to Collaborative Filtering. In *Proceedings of The Web Conference 2018 (WWW 2018)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3178876.3186159>

1 INTRODUCTION

The *recommender system* (RS), which recommends a set of items personalized to a target user that she/he is likely to prefer, has been

extensively studied in the academic society as well as aggressively utilized by many companies such as Google, Netflix, eBay, and Amazon [1][5][20][7]. *Collaborative filtering* (CF) is one of the most successful and widely-used approaches in RS [33][29][13][14]. In order to provide the top- N recommendations, CF relies on past user behaviors. For example, it learns the rating of a user on an item based on the rating given by other users similar to her/him in preferences.

The majority of the entries in a user-item rating matrix R are *missing ratings*, because most users give ratings only on a small number of items [43]. The sparsity of the rating matrix makes it difficult for CF to learn the ratings of a user on items, thereby contributing to the poor recommendation accuracy of CF. This *data sparsity problem* is one of the biggest problems faced by CF. To alleviate the data sparsity problem, a wide variety of approaches, such as acquiring additional information from a trust network [10][17][11][18][19] or crowdsourcing [2][42][8] and performing data imputation [3][36][21], have been extensively studied.

Data imputation aims to infer missing ratings and to impute them to the original rating matrix. This approach has been widely used, because it does not require additional data from other sources (e.g., trust networks and crowdsourcing). Since the imputed value is not a *real rating* but an *inferred rating*, however, there may exist an error. Inferring the imputed value of a missing rating accurately is the key to reducing such errors and eventually improving the accuracy of CF.

Depending on the imputed values, existing data imputation approaches can be classified into two groups: one (called *group M*) that infers missing ratings *with multiple values* [36][27][26] and the other (called *group S*) [43][3][21] that infers missing ratings *with a single value*. Both groups have their limitations. Group M is able to capture the fact that a preference of user u on item i could be different, depending on the characteristics of u and i . However, it mostly assigns missing ratings with *high values*, which contradicts the observation that users tend to have low preferences on items with missing ratings [21][43]. Group S could avoid the problem of group M, because it gives a low rating on most items with missing ratings, but fails to reflect the characteristics of users and items in imputed ratings.

In this paper, we suggest three new claims to overcome the limitations of existing data imputation approaches and verify our claims through experiments with real-world datasets.

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW 2018, April 23–27, 2018, Lyons, France

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

ACM ISBN 978-1-4503-5639-8/18/04.

<https://doi.org/10.1145/3178876.3186159>

- Claim 1 (C1): Most of the imputed values should be a *low rating*, because the items with missing ratings are mostly *uninteresting items*.
- Claim 2 (C2): The imputed values of interesting items should be *higher* than those of *uninteresting items*.
- Claim 3 (C3): The imputed values should be within the range that *users can give* to items.

Although the three claims are simple, they are *easily applicable* to any data imputation approaches and *very effective* in providing high recommendation accuracy. We show the validity of three claims via extensive experiments in Section 3. Furthermore, we propose a novel approach that satisfies all three claims. The key idea of this approach is to infer the post-use preference using the pre-use preference based on the hypothesis below.

Hypothesis: Most pre-use preferences on items lead to their post-use preferences.

User u 's *pre-use preference* on item i (e.g., impression) is determined by i 's *external features* that u can obtain *without actually using it*, such as the genre or director of a movie. On the other hand, u 's *post-use preference* on i (e.g., rating) is determined by i 's *inherent features* that u does not know *before using it*, such as the storyline or choreography of a movie. After using the item, u expresses her/his post-use preference on i *through a rating*.

Note that, the inherent and external features are *not independent*. Most inherent features are tightly related to external features. For example, the storyline or choreography of a movie is mostly under the influence of the director and the genre's characteristics. In this paper, we show the recommendation using *pre-use preferences only* achieves accuracy higher than classic recommendation approaches (e.g., itemKNN and SVD). In addition, it is possible to improve the recommendation accuracy further by considering the characteristics of the users and the items. Because the degree to which the pre-use preference leads to the post-use preference is different depending on the characteristics of the users and the items.

Our imputation approach is as follows. First, it infers the pre-use preferences on all user-item pairs based on the original rating matrix R . Second, it extracts the features (i.e., characteristics) of the user and the item using the *variational autoencoder* (VAE). Third, it trains the model that infers the post-use preference based on the pre-use preference and the extracted features using the *multi-layer perceptron* (MLP). Finally, we infer the post-use preference for every missing rating (i.e., $r_{u,i} = \text{null}$) using the trained model, and impute the inferred post-use preferences to R . To show the effectiveness of the proposed approach, we perform experiments with three real-life datasets. The results demonstrate that our approach universally and consistently outperforms existing state-of-the-art approaches.

In summary, the main contributions of this paper are listed as follows:

- We identify the limitations of existing data imputation approaches.
- We suggest three new claims that all data imputation approaches should follow to achieve high recommendation accuracy.

- We propose a novel deep-learning-based approach that satisfies all three claims and relies on a hypothesis that most pre-use preferences on items lead to their post-use preferences.
- Through extensive experiments on real-world datasets, we verify our claims and hypothesis, and also demonstrate that our proposed approach outperforms significantly existing state-of-the-art approaches.

The paper is organized as follows. Section 2 reviews existing approaches of data imputation and points out their limitations. Section 3 validates our claims via extensive experiments. Section 4 proposes our imputation approach, based on the variational autoencoder and the multi-layer perceptron, to an accurate inference of imputed values and explains its components in detail. Section 5 shows the superiority of our approach in comparison with existing state-of-the-art approaches via experimental evaluation. Finally, Section 6 summarizes and concludes the paper.

2 LIMITATION OF EXISTING APPROACHES

Existing data imputation approaches can be categorized into two groups. The first group (group M), including AutAI [26] and AdaM [27], infers missing ratings as *multi-values*. After performing CF (e.g., userKNN, itemKNN) based on existing ratings, it determines missing ratings as the values predicted by CF. The intuition of group M is that *missing ratings would be similar to existing ratings*. On the other hand, the second group (group S), including Zero-injection [21] and AllRank [43], infers missing ratings as a *single-value*: 0 in Zero-injection and 2 in AllRank. The intuition of group S is that, if $r_{u,i}$ of user u on items i in R is missing, it *implicitly represents* u 's *negative preference* on i . Therefore, it imputes a single *low rating* to missing ratings.

According to [21], a user has a *small* number of *interesting items* and a very *large* number of *uninteresting items*. Interesting items are those items that a user would be interested in and expected to use eventually (i.e., user's pre-use preference on the item is high). Uninteresting items are such items that a user *would not be interested in* and thus not be expected to use (i.e., user's pre-use preference on the item is low). Due to time and cost constraints, a user experiences only a few interesting items; most items are left as uninteresting items.

Keeping the above characteristics of missing ratings in mind, we analyze the limitations of existing data imputation approaches. Despite being *mostly uninteresting items*, most of missing ratings are inferred as *high ratings* by group M, because existing ratings in R are mostly high ratings. Compared to group M, group S has a high accuracy of recommendation because it focuses on a large number of uninteresting items: it regards *all (or most)* items having missing ratings as uninteresting items. However, the approaches of group S have two limitations. First, it ignores a small number of interesting items. *Regardless of whether item i is an interesting item or uninteresting item*, missing rating $r_{u,i}$ in R is inferred as the *same single-value* [43][3]. Second, Zero-injection [21] and pureSVD [3] in group S impute missing ratings as 0. Note that the meaning of the imputed value is the rating that a user is likely to give to an item. However, 0 is even lower than 1, the lowest rating that the

Figure 1: Distribution of existing ratings and imputed values in the three real-world datasets.

Figure 2: The accuracy with different imputed values of interesting items in the three real-world datasets.

user can give. Therefore, imputing 0 *exaggerates* users' negative preference.

3 VALIDATION OF OUR CLAIMS

In this section, we show the limitations of existing data imputation approaches and verify the validity of our claims suggested in Section 1 via a series of experiments.

3.1 Experimental Setup

We use three real-world datasets, whose statistics are reported in Table 1. Because of clear differences in sparsity, the number of ratings, and rating distribution, the datasets are effective for cross-checking the validation of our experiments. The MovieLens is a rating dataset collected and made available from the MovieLens web site¹ by the GroupLens Research. The CiaoDVD is a crawled dataset in 2013 from the DVD category² of Ciao that is the leading platform to get outstanding reviews from real consumers on millions of products. The Watcha is a dataset privately released in 2014 by an online movie-streaming company³ similar to Netflix. The ratings of the three real-world datasets are all in integer values from 1 (worst) to 5 (best). In our experiments, if the dataset is not explicitly

mentioned, the MovieLens dataset is used by default, which has been widely used for evaluating CF algorithms.

For the top- N recommendation, we vary the value of N ($N = 5, 10$, and 20) [9]. Three metrics are used to measure the accuracy of the top- N recommendation: precision, recall, and normalized Discounted Cumulative Gain (nDCG) [10]. For a user u , precision $P_u@N$ and recall $R_u@N$ are computed by $\frac{|Rel_u \cap Rec_u|}{|Rec_u|}$ and $\frac{|Rel_u \cap Rec_u|}{|Rel_u|}$, respectively, where Rec_u denotes a set of N items recommended by CF to u , and Rel_u denotes a set of items considered relevant to u . We consider the items with rating 5 (best) as relevant items, (i.e., ground truth), following [3]. nDCG is used to reflect *ranked positions* of items in Rec_u . Let y_k represent a binary variable

Table 1: Statistics of three real-world datasets

	MovieLens	CiaoDVD	Watcha
Sparsity	95%	99%	97%
# of users	943	996	1,391
# of items	1,682	1,359	1,927
# of ratings	100,000	18,640	101,073
Avg. of all ratings	3.56	4.03	3.68
Avg. # of ratings per user	106.04	18.72	72.66
Min. # of ratings per user	20	1	16
Max. # of ratings per user	737	319	513

¹movielens.org

²dvd.ciao.co.uk

³watcha.net

Table 2: Comparison of group M and group S (MovieLens)

Metrics	AdaM (group M)	Zero-injection (group S)
$P@5$	0.051	0.207
$P@10$	0.046	0.166
$P@20$	0.042	0.125
$R@5$	0.043	0.218
$R@10$	0.076	0.325
$R@20$	0.132	0.450
$nDCG@5$	0.058	0.274
$nDCG@10$	0.066	0.297
$nDCG@20$	0.085	0.332

for the k -th item i_k in Rec_u . If $i_k \in Rel_u$, y_k is set as one; otherwise, y_k is set as zero. Then, $nDCG_u@N$ is computed by $\frac{DCG_u@N}{IDCG_u@N}$, where $DCG_u@N = \sum_{k=1}^N \frac{2^{y_k-1}}{\log_2(k+1)}$, and $IDCG_u@N$ means an ideal $DCG_u@N$ where y_k is set as one for every item $i_k \in Rec_u$. Finally, $P@N$, $R@N$, and $nDCG@N$ are the averages of $P_u@N$, $R_u@N$ and $nDCG_u@N$ for all users, respectively. In our experiments, if the metric is not explicitly mentioned, $P@5$ is used because other metrics and other N showed very similar tendencies. For all experimental results, we have performed five cross-validations.

3.2 Validating C1

C1 is based on the limitation of group M. Despite being mostly uninteresting items, most of missing ratings are *inferred high* by group M, because existing ratings in rating matrix R are mostly high [43][21]. Therefore, we verify the following: (1) *existing ratings and imputed values inferred by CF are high ratings*, and (2) *most imputed values should be low ratings*.

To verify (1), Figure 1 shows the distribution of existing ratings and imputed values inferred by CF⁴. As shown, both existing ratings and imputed values are *mostly higher* than or equal to 3. To verify (2), we show that the accuracy of recommendations is improved when imputed values are low ratings. First, we impute the *same* value (e.g., 1 to 5) to rating matrix R and then recommend top- N items for each user by CF. To demonstrate the validity of C1 independently of CF methods, we use the most widely used itemKNN and SVD in CF. In Figure 2, the accuracy of recommendations is high only when imputed values are *low ratings* (i.e., 1 and 2) in both itemKNN and SVD. Also, the accuracy of group S that imputes low values is *much higher* than that of group M that imputes high values as in Table 2. We use AdaM [27] and Zero-injection [21] for comparisons because they show the highest accuracy in each group. Through these three experiments, we identify the limitation of group M and also validate C1.

3.3 Validating C2

C2 is based on the first limitation of group S. Group S ignores a small number of interesting items. Whether item i is an interesting item or an uninteresting item to user u , all the missing ratings $r_{u,i}$

⁴We used itemKNN [5] in CF here; we also observed the same tendency when we used userKNN and other CF.

Table 3: The accuracy with respect to different imputed values of interesting items (MovieLens)

Metrics	AllRank ($I_{un} = 2$)				
	$I_{un}-1.0$	$I_{un}-0.5$	$I_{un}+0$	$I_{un}+0.5$	$I_{un}+1.0$
$P@5$	0.081	0.128	0.184	0.207	0.207
$P@10$	0.063	0.100	0.147	0.164	0.164
$P@20$	0.047	0.075	0.108	0.122	0.123
$R@5$	0.067	0.119	0.195	0.216	0.219
$R@10$	0.099	0.174	0.287	0.322	0.324
$R@20$	0.141	0.241	0.397	0.446	0.446
$nDCG@5$	0.103	0.170	0.247	0.278	0.281
$nDCG@10$	0.104	0.175	0.264	0.297	0.300
$nDCG@20$	0.114	0.192	0.294	0.330	0.334

Metrics	pureSVD ($I_{un} = 0$)				
	$I_{un}-1.0$	$I_{un}-0.5$	$I_{un}+0$	$I_{un}+0.5$	$I_{un}+1.0$
$P@5$	0.113	0.146	0.161	0.173	0.181
$P@10$	0.087	0.115	0.128	0.138	0.147
$P@20$	0.066	0.087	0.099	0.106	0.112
$R@5$	0.102	0.142	0.163	0.178	0.189
$R@10$	0.150	0.215	0.246	0.268	0.289
$R@20$	0.212	0.302	0.360	0.388	0.415
$nDCG@5$	0.147	0.190	0.214	0.230	0.242
$nDCG@10$	0.151	0.202	0.229	0.246	0.263
$nDCG@20$	0.166	0.225	0.260	0.280	0.298

are imputed by a *single-value*. For example, pureSVD [3] and AllRank [43] impute 0 and 2 to rating matrix R regardless of interesting and uninteresting items, respectively. C2 asserts that, although the number of interesting items is small, the accuracy of recommendations is improved when it is considered. To verify C2, we show that the accuracy of AllRank and pureSVD is improved when C2 is applied to them. More specifically, we let AllRank and pureSVD impute higher ratings to *interesting items* than uninteresting items while having the rest of their schemes remain the same. We define interesting items as the top 10% of missing ratings according to the pre-use preferences [21].

In Table 3, I_{un} indicates the imputed values for uninteresting items. Thus, $I_{un}+0.5$ is 2.5 in AllRank and 0.5 in pureSVD; $I_{un}+1.0$ and $I_{un}+0.5$ (resp. $I_{un}-1.0$ and $I_{un}-0.5$) mean that the imputed values of interesting items are higher (resp. lower) than those of uninteresting items; $I_{un}+0$ means that the imputed values of interesting items and those of uninteresting items are same, that is, original AllRank and pureSVD. Table 3 reports that the accuracy of AllRank and pureSVD is improved when they impute *higher* values to interesting items than those to uninteresting items and is decreased when they impute lower values to interesting items than those to uninteresting items in terms of all metrics. The experimental result identifies the first limitation of group S and validates C2.

3.4 Validating C3

C3 is based on the second limitation of group S. Zero-injection in group S infers missing ratings as 0 without considering the meaning of imputed values. Note that the meaning of an imputed value is the

Table 4: The accuracy with respect to different imputed values ranging from 0 to 5 (MovieLens)

Metrics	Zero-injection					
	0	1	2	3	4	5
$P@5$	0.207	0.224	0.234	0.224	0.159	0.031
$P@10$	0.166	0.175	0.182	0.176	0.127	0.03
$P@20$	0.125	0.130	0.134	0.131	0.096	0.03
$R@5$	0.218	0.235	0.244	0.230	0.148	0.029
$R@10$	0.325	0.337	0.347	0.330	0.224	0.058
$R@20$	0.450	0.464	0.468	0.453	0.323	0.124
$nDCG@5$	0.274	0.303	0.316	0.304	0.209	0.037
$nDCG@10$	0.297	0.319	0.331	0.319	0.219	0.046
$nDCG@20$	0.332	0.353	0.362	0.350	0.244	0.067
pureSVD						
	0	1	2	3	4	5
$P@5$	0.161	0.167	0.184	0.194	0.105	0.005
$P@10$	0.128	0.136	0.147	0.149	0.08	0.004
$P@20$	0.099	0.103	0.108	0.111	0.06	0.003
$R@5$	0.163	0.176	0.195	0.197	0.076	0.002
$R@10$	0.246	0.264	0.287	0.276	0.113	0.003
$R@20$	0.360	0.379	0.397	0.383	0.161	0.006
$nDCG@5$	0.214	0.224	0.247	0.259	0.13	0.005
$nDCG@10$	0.229	0.242	0.264	0.268	0.128	0.005
$nDCG@20$	0.260	0.274	0.294	0.295	0.137	0.005

rating that a user is likely to give to an item. However, 0 is lower than 1, which is the lowest rating that users can give. Therefore, imputing 0 *exaggerates* users' negative preference. C3 asserts that the imputed values should be within the range that users can give to items in reality. To verify C3, we let the approaches which impute 0 (i.e., Zero-injection and pureSVD) impute values within the *rating range* (i.e., from 1 to 5), rather than 0 while letting the rest of their approaches remain the same.

Table 4 shows the accuracy with respect to different imputed values ranging from 1 to 5. Both Zero-injection and pureSVD show higher accuracy when imputed values are 1, 2, and 3 rather than 0. This result identifies that imputing 0 *exaggerates* users' negative preference too much and thus validates C3. In addition, the accuracy drops *drastically* when the imputed values are 4 and 5, which confirms again that *most of the missing ratings are uninteresting items*.

The validation of our claims in this section clearly points out the limitations of existing imputation approaches and demonstrates the possibility of improving the accuracy of recommendation when our *three claims* are satisfied.

4 PROPOSED IMPUTATION APPROACH

In this section, we present our imputation approach in detail: the challenges (Section 4.1); the overview of the proposed approach (Section 4.2); how to infer the pre-use preference of a user on an item (Section 4.3); how to extract the features of a pair of a user and an item (Section 4.4); finally, how to infer the post-use preference of a user on an item for imputation (Section 4.5).

4.1 Challenges

How do we determine post-use preferences from pre-use preferences? A straightforward answer would be simple: to use pre-use preferences as post-use preferences. This, however, would not work for two reasons. First, their scales are different. Pre-use preferences are ranged from 0 to 1 [21], while post-use preferences are from 1 to 5. Second, even if the scales are converted, post-use preferences should be based not only on pre-use preferences but also on the characteristics of a target user and a target item. Although most pre-use preferences on items lead to their post-use preferences, the degree of correlation depends on the user and the item.

4.2 Overview

Our approach consists of the following four steps. First, we infer the pre-use preferences on all user-item pairs (Figure 3(a)). Note that, even if a user u 's pre-use preference on an item i and a user v 's pre-use preference on an item j is equal, u 's post-use preference on i and v 's post-use preference on j could be different, depending on the characteristics of the users (u and v) and the items (i and j). To address this issue, we extract the features (i.e., characteristics) of the user and the item (Figure 3(b)). Next, we train the model which infers the post-use preference using those features and the pre-use preference together. Specifically, the model is trained with the existing ratings in R and infers u 's post-use preference on i , given u 's pre-use preference on i and features of u and i (Figure 3(c)). Finally, we infer the post-use preference for every missing rating (i.e., $r_{u,i} = \text{null}$) using the trained model and impute the inferred post-use preferences to R to solve the data sparsity problem.

4.3 Inference of Pre-use Preferences

Inferring the pre-use preference of a user on an item with an existing rating is straightforward. *Interesting item $_{u,i}$* is the item i that a user u would be interested in and expected to experience. Therefore, u 's pre-use preference on i with an *existing rating* should be high. On the other hand, inferring u 's pre-use preference of i with a *missing rating* is non-trivial, because missing ratings encompass both interesting and uninteresting items.

This is the *One-Class Collaborative Filtering (OCCF) problem* [22][12][39][24][40][44][23]. The OCCF problem occurs when the rating is a single, usually positive feedback, such as purchases and bookmarks. A cell in the feedback matrix is either missing a rating (i.e., null) or indicating interesting (i.e., 1). Therefore, ambiguity arises in the interpretation of missing ratings. A missing rating may mean *positive feedback*; the user was not aware of the existence of the item but would have liked it if she/he had known. If a missing rating means *negative feedback*, the user knew the existence of the item but did not like it [30][38][28].

To infer pre-use preferences (Figure 3(a)) on those items with missing ratings, we first convert a rating matrix R to a binary matrix B [31][15]. In matrix B , $b_{u,i}$ is 1 if $r_{u,i}$ exists (i.e., *interesting item $_{u,i}$*); $b_{u,i}$ is 0 if $r_{u,i}$ is missing (i.e., *uninteresting item $_{u,i}$*). Since not all missing ratings indicate uninteresting items, however, we employ *weighted Alternative Least Squares* (wALS) [37]. Given matrix B , wALS aims at approximating B with a pre-use preference matrix C , where $c_{u,i} \in [0,1]$ indicates u 's pre-use preference on i , with minimizing the loss function in Eq. (1). The closer to 1, the higher

Figure 3: Overview of our approach.

the pre-use preference is.

$$\mathcal{L}(C) = \sum_u \sum_i w_{u,i} (b_{u,i} - c_{u,i})^2, \quad (1)$$

where $w_{u,i} \in [0,1]$ indicates the weight of $b_{u,i}$. The weight matrix W assignment is based on one of the most successful and widely-used approaches [21][12]. If $b_{u,i}$ is 1 (i.e., observed rating), $w_{u,i}$ is assigned with the highest weight, 1. If $b_{u,i}$ is 0 (i.e., missing rating), $w_{u,i}$ is proportional to the number of ratings given by u . In [12], it is demonstrated, as a user has a more number of positive feedbacks, it is more likely that she/he does not like the other (i.e., missing) items. Therefore, the missing rating of such a user is more likely to indicate a negative feedback.

Since matrix C is a matrix multiplication of a matrix $X \in \mathbb{R}^{m \times d}$ and a matrix $Y \in \mathbb{R}^{n \times d}$, we can re-write the loss function in Eq. (1) as Eq. (2). d is a low rank of matrix B , and m and n are the numbers of users and items, respectively [34][16][41].

$$\mathcal{L}(X, Y) = \sum_u \sum_i w_{u,i} \{ (b_{u,i} - X_{u(\cdot)} Y_{i(\cdot)}^T)^2 + \lambda_1 (\|X_{u(\cdot)}\|_F^2 + \|Y_{i(\cdot)}\|_F^2) \}, \quad (2)$$

where $\|\cdot\|_F$ denotes the Frobenius norm and λ_1 is the regularization parameter ($\lambda_1 = 0.015$, following [21]). In order to minimize the loss function, wALS first initializes matrix X with random values (step 1). Then, while fixing X , it computes Y using Eq. (3) (step 2). After solving Y , it computes X while fixing Y , using Eq. (4) (step 3). wALS repeats steps 2 and 3 until convergence. After the convergence, each $c_{u,i}$ in C has user u 's pre-use preference inferred on item i .

$$X_{u(\cdot)} = B_{u(\cdot)} \tilde{W}_{u(\cdot)} Y \{ Y^T \tilde{W}_{u(\cdot)} Y + \lambda_1 (\sum_i w_{u,i}) I \}^{-1}, \quad \forall 1 \leq u \leq m, \quad (3)$$

$$Y_{i(\cdot)} = B_{(\cdot)i}^T \tilde{W}_{(\cdot)i} X \{ X^T \tilde{W}_{(\cdot)i} X + \lambda_1 (\sum_u w_{u,i}) I \}^{-1}, \quad \forall 1 \leq i \leq n, \quad (4)$$

where $\tilde{W}_{u(\cdot)}$ is a diagonal matrix with elements of $W_{u(\cdot)}$ on the diagonal, and matrix I is a $d \times d$ identity matrix.

4.4 Extracting User and Item Features

We use the *variational autoencoder* (VAE) [32], one of the most popular approaches in the generative model, to extract the features of user u and the item i (Figure 3(b)) [4][25][35]. VAE allows learning from a large dataset compared to other generative model approaches (e.g., Markov chain Monte Carlo sampling) and shows better reconstructive performance. Being unsupervised learning, VAE does not require labeled data (i.e., ratings) during training and thus is able to utilize all user-item pairs in R [6]. The goal of VAE is to maximize the probability distribution of data x in Eq. (5).

$$P(x; \theta) = \int P(x|z; \theta) P(z) dz. \quad (5)$$

In our case, x is a pair of user and item vectors. User vectors and item vectors are rows and columns, respectively, of rating matrix R , where missing ratings are treated as 0s. z is the latent feature of a user-item pair and is used as an input when inferring the imputed value. High $P(x)$ indicates z represents data x better. Therefore, maximizing $P(x)$ is important in improving the accuracy of our approach.

However, the integral is intractable in Eq. (5), as it needs to be evaluated over all of z . To solve this problem, VAE approximates the posterior distribution using variational inference. As a result, Eq. (5) can be rewritten as Eq. (6). Eq. (6) is a loss function of VAE and can be rewritten as Eqs. (7) and (8), since the KL divergence is non-negative.

$$\log P(x) = D_{KL}(Q(z|x; \phi) || P(z|x; \theta)) + \mathcal{L}(\theta, \phi; x) \quad (6)$$

$$\begin{aligned} \log P(x) &\geq \mathcal{L}(\theta, \phi; x) \\ &= \mathbb{E}_{Q(z|x; \phi)} [-\log Q(z|x; \phi) + \log P(x, z; \theta)] \end{aligned} \quad (7)$$

$$\begin{aligned} \mathcal{L}(\theta, \phi; x) &= -D_{KL}(Q(z|x; \phi) || P(z; \theta)) + \\ &\quad \mathbb{E}_{Q(z|x; \phi)} [\log P(x|z; \theta)] \end{aligned} \quad (8)$$

Given x corresponding to a pair of u and i , our approach samples z , which minimizes Eq. (8), for x by using $P(z|x; \theta)$ learned by VAE. As shown in Figure 3(b), the sampled z , together with u 's pre-use preference on i is used when inferring u 's post-use preference on i .

4.5 Inference of Post-use Preferences

To infer u 's *post-use preference* (Figure 3(c)), we use the *features* of user u and item i extracted by VAE and u 's *pre-use preference* on i . Our approach is to train the model to predict how u 's pre-use preference on i leads to u 's post-use preference on i according to the features of u and i . For learning, we use the multi-layer perceptron (MLP) that uses multiple layers and non-linear functions. Since it is supervised learning, MLP requires labeled data (i.e., post-use preferences or ratings). In our case, MLP uses existing ratings in R for training. The model parameters ϕ is trained by minimizing the loss function in Eq (9).

$$\mathcal{L}(\phi) = \sum_u \sum_i (r_{u,i} - \hat{r}_{u,i})^2, \quad (9)$$

where $r_{u,i}$ is the existing rating and $\hat{r}_{u,i}$ is the value predicted by the model. If $\hat{r}_{u,i}$ is less than 1 and larger than 5 (rarely occurring cases), it is truncated to 1 and 5, respectively, following C3. After training the model, we predict the post-use preference for every missing rating (i.e., $r_{u,i} = \text{null}$) and impute the predicted values to R .

4.6 Discussions

Our approach is *CF-agnostic*, that is, applicable to any CF methods in the recommendation, because it requires the imputation to the rating matrix only. Furthermore, the recommendation accuracy increases as it satisfies our three claims. Figure 4 reports the distribution of imputed values by our approach, indicating that most imputed values are low ratings (C1). Figure 5 shows pre-use preferences are strongly (positively) correlated to post-use preferences. Because our approach infers post-use preferences by referring to pre-use preferences, the imputed values (i.e., inferred post-use preferences) of interesting items are mostly higher than those of uninteresting items (C2). Imputed values are in the range of ratings that any user actually can give to items (C3).

5 EXPERIMENTS

In this section, we verify our hypothesis and then evaluate the effectiveness of our approach with real-world datasets. All experimental setup is equal to that stated in Section 3. Our experiments are to answer the following two key questions:

- Q1: Do pre-use preferences lead to post-use preferences? (Is our hypothesis valid?)

Figure 4: Distribution of imputed values inferred by our approach (MovieLens).

- Q2: How much does our approach outperform existing state-of-the-art approaches? (in various situations with data sparsity, including basic top- N recommendation, long-tail item's recommendation, and recommendation to cold-start users)

5.1 Q1. Validating Our Hypothesis

This section presents two experiments whose results support the validity of our hypothesis that pre-use preferences on items mostly lead to their post-use preferences. In the first experiment, we directly compare the correlation between pre-use preferences and post-use preferences on items. A positive correlation signifies that our hypothesis is valid.

Figure 5 reports the correlation between pre-use preferences and post-use preferences on items, where the x-axis indicates the range of pre-use preferences on items and the y-axis does the average of post-use preferences on items having the pre-use preferences in the corresponding range. It clearly shows the positive correlation between the two axes, especially in the range of pre-use preferences between 0.4 and 1.0. Given that most items (93.3% in exact) belong to the pre-use preferences range between 0.4 and 1.0, this result supports that our hypothesis is valid.

In the second experiment, we show that a high level of accuracy can be achieved *even when only pre-use preferences are used* in top- N recommendations. In other words, each user is recommended with the items having the top- N *pre-use preferences* inferred for her/him, without any CF. If our hypothesis is valid, the recommendation should provide reasonable accuracy. We compare the accuracy of

Figure 5: Correlation analysis between pre-use preferences and post-use preferences (MovieLens).

Table 5: The accuracy of top- N recommendations using pre-use preferences (MovieLens)

Metrics	itemKNN	SVD	top- N pre-use preferences
$P@5$	0.039	0.069	0.192
$P@10$	0.041	0.059	0.151
$P@20$	0.039	0.048	0.115
$R@5$	0.030	0.056	0.202
$R@10$	0.059	0.091	0.299
$R@20$	0.111	0.144	0.421
$nDCG@5$	0.043	0.084	0.254
$nDCG@10$	0.053	0.089	0.272
$nDCG@20$	0.071	0.105	0.306

recommendation based on the top- N pre-use preferences with two existing CF approaches, itemKNN [5] and SVD [34]. Both itemKNN and SVD use existing ratings only, without employing any data imputation.

Table 5 reports the accuracy results. Surprisingly, the recommendation using pre-use preferences only (i.e., no further CF) achieves accuracy, 389% higher than itemKNN and 176% higher than SVD, in terms of $P@5$. Indeed, its remarkable accuracy is comparable to that of existing state-of-the-art approaches. Based on the results of two experiments above, as the answer to Q1, it is verified that pre-use preferences lead to post-use preferences.

5.2 Q2. Performance Comparison

This section evaluates how much our approach outperforms existing state-of-the-art approaches. The accuracy evaluation is conducted under various situations, including basic top- N recommendation, long-tail item's recommendation, and recommendation to cold-start users. To demonstrate the effectiveness of our approach independently of CF methods, we use the most widely used itemKNN and SVD in CF. We compare two variants of our approach with three existing state-of-the-art approaches (i.e., pureSVD [3], AllRank [43], and Zero-injection [21]) in group S because the accuracy of group S is higher than those of group M as shown in Table 2.

Figure 6 shows that two variants of our approach universally and consistently outperform existing imputation approaches with all datasets and all metrics. This indicates the imputed values inferred by our approach are more beneficial than those by other existing imputation approaches since they satisfy all the three claims. Moreover, Our approach+SVD achieves 16.4% higher accuracy than

Table 6: Comparison of existing state-of-the-art approaches and our approach

	Imputed values	Imputing targets
pureSVD	0	All missing ratings
AllRank	2	All missing ratings
Zero-injection	0	Ratings only for uninteresting items
Our approach	[1, 5]	All missing ratings

Zero-injection, which has been reported to provide the best accuracy [21] among existing imputation approaches, in terms of $P@5$ in MovieLens.

We also validate our approach under the situations where the problem of data sparsity is even more serious: (1) long-tail item's (i.e., items having a few ratings) recommendation and (2) recommendation to cold-start users (i.e., users giving a few ratings) [9]. To set up the first situation, we divide items into *top-head items* and *long-tail items* by their popularity (i.e., the number of ratings they have) [36]. For MovieLens, top-head items are 5.9% of all items (i.e., 100 items) but involve 30% of all ratings. Then, we only recommend those items belonging to long-tail items to users. We set up the second situation, following [9] and [36]. We randomly select 200 users as cold-start users. Then, we also randomly select a small k

Table 7: The accuracy of long-tail item's recommendations (MovieLens)

Metrics	pureSVD	AllRank	Zero-injection	Our approach+itemKNN	Our approach+SVD
$P@5$	0.099	0.104	0.108	0.112	0.122
$P@10$	0.076	0.082	0.087	0.088	0.096
$P@20$	0.057	0.062	0.066	0.068	0.072
$R@5$	0.183	0.199	0.190	0.194	0.220
$R@10$	0.133	0.144	0.144	0.145	0.160
$R@20$	0.191	0.202	0.206	0.209	0.227
$nDCG@5$	0.127	0.137	0.138	0.142	0.158
$nDCG@10$	0.132	0.143	0.145	0.146	0.163
$nDCG@20$	0.146	0.157	0.160	0.162	0.180

Table 8: The accuracy of top- N recommendations to cold-start users (MovieLens)

Metrics	$k=5$				
	pureSVD	AllRank	Zero-injection	Our approach+itemKNN	Our approach+SVD
$P@5$	0.054	0.063	0.029	0.079	0.080
$P@10$	0.047	0.053	0.020	0.066	0.062
$P@20$	0.040	0.045	0.012	0.053	0.053
$R@5$	0.053	0.062	0.040	0.083	0.078
$R@10$	0.096	0.103	0.063	0.137	0.125
$R@20$	0.144	0.165	0.079	0.215	0.207
$nDCG@5$	0.066	0.075	0.043	0.107	0.097
$nDCG@10$	0.079	0.085	0.048	0.118	0.107
$nDCG@20$	0.094	0.105	0.052	0.141	0.132
	$k=10$				
	pureSVD	AllRank	Zero-injection	Our approach+itemKNN	Our approach+SVD
$P@5$	0.066	0.071	0.063	0.080	0.090
$P@10$	0.057	0.064	0.053	0.072	0.075
$P@20$	0.048	0.053	0.038	0.063	0.063
$R@5$	0.071	0.072	0.077	0.087	0.096
$R@10$	0.119	0.132	0.122	0.153	0.152
$R@20$	0.190	0.192	0.180	0.248	0.246
$nDCG@5$	0.086	0.092	0.084	0.111	0.118
$nDCG@10$	0.099	0.109	0.097	0.129	0.132
$nDCG@20$	0.122	0.128	0.113	0.159	0.161

Figure 6: The accuracy of basic top- N recommendations (MovieLens, CiaoDVD, and Watcha).

number of ratings (k is set as 5 and 10) for each cold-start user and discard the rest of the ratings. Considering the average number of ratings per user in Table 1, the selective values of k are small. We finally recommend items only to the cold-start users.

Tables 7 and 8 show the accuracy of recommendations under these two data sparsity situations in MovieLens. Similar to the results of basic top- N recommendation, two variants of our approach universally and consistently outperform existing imputation approaches with all metrics. The overall results provide the answer to Q2.

6 CONCLUSIONS

In this paper, we identified the limitations of existing data imputation approaches and suggested three new claims that all data imputation approaches should follow for more accurate recommendations. We also hypothesized that most pre-use preferences on

items lead to their post-use preferences. Based on our hypothesis, we proposed a novel imputation approach based on deep learning. Through extensive experiments with real-world datasets, we demonstrated that our proposed approach is quite effective, achieving up to 16.4% higher accuracy compared to the *best* among the state-of-the-art approaches. Our approach also outperforms all the other approaches under more difficult settings, such as long-tail item's recommendation and recommendation to cold-start users.

7 ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT: Ministry of Science and ICT) (No. NRF-2017R1A2B3004581). Also, this work was supported by the Naver Corporation, where Jung-Tae Lee and Jaeho Choi gave us good comments in a practical point of view, which helped us greatly in performing this research successfully.

REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. on Knowledge and Data Engineering* 17 (2005), 734–749.
- [2] S. Chang, F. M. Harper, and L. G. Terveen. 2016. Crowd-Based Personalized Natural Language Explanations for Recommendations. In *Proc. ACM Int'l Conf. on Recommender Systems*. 175–182.
- [3] P. Cremonesi, Y. Koren, and R. Turrin. 2010. Performance of Recommender Algorithms on Top-N Recommendation Tasks. In *Proc. ACM Int'l Conf. on Recommender Systems*. 39–46.
- [4] C. Doersch. 2016. Tutorial on Variational Autoencoders. *arXiv preprint arXiv:1606.05908* (2016).
- [5] B. Sarwar et al. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *Proc. Int'l Conf. on World Wide Web*. 285–295.
- [6] D. Kingma et al. 2014. Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems*. 3581–3589.
- [7] D. K. Chae et al. 2018. On Identifying k-Nearest Neighbors in Neighborhood Models for Efficient and Effective Collaborative Filtering. *Neurocomputing* 278 (2018), 124–143.
- [8] J. Lee et al. 2013. Alleviating the Sparsity in Collaborative Filtering using Crowdsourcing. In *Proc. of Workshop on Crowdsourcing and Human Computation for Recommender Systems*.
- [9] J. Lee et al. 2016. Improving the Accuracy of Top-N Recommendation using a Preference Model. *Information Sciences* 348 (2016), 290–304.
- [10] J. W. Ha et al. 2012. Top-N Recommendation through Belief Propagation. In *Proc. ACM Int'l Conf. on Information and Knowledge Management*. 2343–2346.
- [11] M. H. Jang et al. 2016. PIN-TRUST: Fast Trust Propagation Exploiting Positive, Implicit, and Negative Information. In *Proc. ACM Int'l Conf. on Information and Knowledge Management*. 629–638.
- [12] R. Pan et al. 2008. One-Class Collaborative Filtering. In *Proc. IEEE Int'l Conf. on Data Mining*. 502–511.
- [13] S. C. Lee et al. 2015. A Graph-Based Recommendation Framework for Price-Comparison Services. In *Proc. Int'l Conf. on World Wide Web*. 59–60.
- [14] S. C. Lee et al. 2017. A Single-Step Approach to Recommendation Diversification. In *Proc. Int'l Conf. on World Wide Web*. 809–810.
- [15] S. Rendle et al. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proc. Int'l Conf. on Uncertainty in Artificial Intelligence*. 452–461.
- [16] V. Sindhwani et al. 2010. One-Class Matrix Completion with Low-Density Factorizations. In *Proc. IEEE Int'l Conf. on Data Mining*. 1055–1060.
- [17] W. S. Hwang et al. 2013. Exploiting Trustors as well as Trustees in Trust-Based Recommendation. In *Proc. ACM Int'l Conf. on Information and Knowledge Management*. 1893–1896.
- [18] W. S. Hwang et al. 2014. Data Imputation using a Trust Network for Recommendation. In *Proc. Int'l Conf. on World Wide Web*. 299–300.
- [19] W. S. Hwang et al. 2015. On Exploiting Trustors in Trust-Based Recommendation. *Journal of Internet Technology* 16, 4 (2015), 755–765.
- [20] W. S. Hwang et al. 2016. Efficient Recommendation Methods using Category Experts for a Large Dataset. *Information Fusion* 28 (2016), 75–82.
- [21] W. S. Hwang et al. 2016. “Told You I Didn’t Like It”: Exploiting Uninteresting Items for Effective Collaborative Filtering. In *Proc. IEEE Int'l Conf. on Data Engineering*. 349–360.
- [22] X. He et al. 2016. Fast Matrix Factorization for Online Recommendation with Implicit Feedback. In *Proc. ACM Int'l Conf. on Research and Development in Information Retrieval*. 549–558.
- [23] Y. C. Lee et al. 2018. gOCCF: Graph-Theoretic One-Class Collaborative Filtering Based on Uninteresting Items. In *Proc. AAAI Conf. on Artificial Intelligence*.
- [24] Y. Li et al. 2010. Improving One-Class Collaborative Filtering by Incorporating rich User Information. In *Proc. ACM Int'l Conf. on Information and Knowledge Management*. 959–968.
- [25] Y. Pu et al. 2016. Variational Autoencoder for Deep Learning of Images, Labels and Captions. In *Advances in Neural Information Processing Systems*. 2352–2360.
- [26] Y. Ren et al. 2012. The Efficient Imputation Method for Neighborhood-Based Collaborative Filtering. In *Proc. ACM Int'l Conf. on Information and Knowledge Management*. 684–693.
- [27] Y. Ren et al. 2013. AdaM: Adaptive-Maximum Imputation for Neighborhood-Based Collaborative Filtering. In *Proc. IEEE Int'l Conf. on Advances in Social Networks Analysis and Mining*. 628–635.
- [28] Y. Yao et al. 2014. Dual-Regularized One-Class Collaborative Filtering. In *Proc. ACM Int'l Conf. on Information and Knowledge Management*. 759–768.
- [29] Z. Gantner et al. 2011. MyMediaLite: A Free Recommender System Library. In *Proc. ACM Int'l Conf. on Recommender Systems*. 305–308.
- [30] R. He and J. McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proc. Int'l Conf. on World Wide Web*. 507–517.
- [31] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proc. IEEE Int'l Conf. on Data Mining*. 263–272.
- [32] D. Kingma and M. Welling. 2013. Auto-Encoding Variational Bayes. In *arXiv preprint arXiv:1312.6114*.
- [33] Y. Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *Proc. ACM Int'l Conf. on Knowledge Discovery and Data Mining*. 426–434.
- [34] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42 (2009), 30–37.
- [35] X. Li and J. She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *Proc. of ACM Int'l Conf. on Knowledge Discovery and Data Mining*. 305–314.
- [36] H. Ma, I. King, and M. R. Lyu. 2007. Effective Missing Data Prediction for Collaborative Filtering. In *Proc. ACM Int'l Conf. on Research and Development in Information Retrieval*. 39–46.
- [37] S. Nathan and J. Tommi. 2003. Weighted Low-Rank Approximations. In *Proc. Int'l Conf. on Machine Learning*. 720–727.
- [38] R. Pan and M. Scholz. 2009. Mind the Gaps: Weighting the Unknown in Large-Scale One-Class Collaborative Filtering. In *Proc. ACM Int'l Conf. on Knowledge Discovery and Data Mining*. 667–676.
- [39] W. Pan and L. Chen. 2013. GBPR: Group Preference Based Bayesian Personalized Ranking for One-Class Collaborative Filtering. In *Proc. Int'l Joint Conf. on Artificial Intelligence*. 2691–2697.
- [40] S. Rendle and C. Freudenthaler. 2014. Improving Pairwise Learning for Item Recommendation from Implicit Feedback. In *Proc. Int'l Conf. on Web Search and Data Mining*. 273–282.
- [41] R. Salakhutdinov and A. Mnih. 2008. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems*. 1257–1264.
- [42] B. Smyth, R. Rafter, and S. Banks. 2016. Harnessing Crowdsourced Recommendation Preference Data from Casual Gameplay. In *Proc. ACM Int'l Conf. on User Modeling Adaptation and Personalization*. 95–104.
- [43] H. Steck. 2010. Training and Testing of Recommender Systems on Data Missing Not at Random. In *Proc. ACM Int'l Conf. on Knowledge Discovery and Data Mining*. 713–722.
- [44] M. Volkovs and G. W. Yu. 2015. Effective Latent Models for Binary Feedback in Recommender Systems. In *Proc. ACM Int'l Conf. on Research and Development in Information Retrieval*. 313–322.