Received: December 13, 2022. Revised: March 14, 2023.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

"Guess Why I didn't rate it": A New Preference-based Model for Enhanced Top-K Recommendation

Nouhaila Idrissi[1]* **Ahmed Zellou[2]** **Zohra Bakkoury[1]**

[1]*Mohammadia School of Engineering, Mohammed V University in Rabat, Morocco*
[2]*National Higher School of Computer Science and Systems Analysis, Mohammed V University in Rabat, Morocco*
* Corresponding author's Email: nouhaila_idrissi@um5.ac.ma

Abstract: In existing matrix factorization (MF)-based recommender systems, the user-item interaction matrix is factorized linearly into two low-ranked feature matrices to generate predictions or provide the users with personalized rankings of items. However, when MF methods are directly applied to sparse rating matrices, they cannot cope with the inherent structure of real-world user and item latent features. The efficiency of MF-based systems crucially depends on their capability to address sparsity issues. To this end, we propose, in this paper, a novel preference-based data imputation approach for effective MF-based Top-K recommendation. We apply MF on an imputed and denser rating matrix with only interesting items to users. We obtain these items by inferring the prior preferences of users, considering some biases that may impact their choices for items they interact with, and leveraging a powerful latent and non-linear feature extraction using a deep generative model. Experimental results on two real-world sparse datasets reveal that the proposed model significantly enhances the performance of Top-K recommendations and outperforms baselines with an average improvement margin of 6.45% and 3.91% in hit rate (HR) and normalized discounted cumulative gain (NDCG) evaluation metrics, respectively, averaging on all employed datasets.

Keywords: Recommender systems, Matrix factorization, Data imputation, Sparsity, Collaborative filtering.

1. Introduction

Collaborative filtering (CF) is woven into any recommendation system to mine personalized information, thus dealing with progressively expanding data and enhancing the user's experience. Such personalization is fulfilled by assuring an accurate rating prediction or a reliable ranking-based recommendation approach [1-3]. The purpose of the rating prediction task is to effectively predict ratings close to those given by the user to ensure an optimal minimization of the prediction error for unobserved interactions in the user-item matrix. The ranking-based system aims to provide users with Top-K personalized and sorted items according to their interests and preferences. Several state-of-the-art methods leverage matrix factorization (MF) techniques, including singular value decomposition (SVD), to learn efficient ranked lists of items for users [1, 4]. However, in real-world applications, the user-item interaction matrices are sparse, with only a few observed rating values given by users on items [2, 3]; hence using an MF technique to linearly model interaction data directly on a sparse rating matrix cannot deal with the non-linear and complex intrinsic structure of user and item latent features [2]. Imputation approaches are proposed to tackle sparsity issues by ensuring pre-processing for missing entries in the user-item matrix before generating ranked items for the target user [5-7]. Nevertheless, many data imputation-based models assume that unobserved rating information is missing at random (MAR) [8], considering that the probability of observing a rating does not rely on its value. Such an assumption is not met in a real-world application where users select items they want to interact with, especially those with higher pre-use preferences. Ignoring these missing interactions

Received: December 13, 2022. Revised: March 14, 2023.

would bias the model towards higher available rating values, thereby inaccurately ranking tail items [9]. On the other hand, several state-of-the-art imputation approaches consider unobserved items with empty values in the rating matrix as negative examples [10, 11] since users are likelier to assign ratings for items they are interested in, which would inaccurately increase the ranking of popular items.

This paper proposes PreFImp, a novel preference-based imputation approach for effective Top-K recommendation. We impute a denser rating matrix for the SVD model by exploring unrated positive items that may interest the users. We obtain these items by inferring prior preferences of users for items considering that some entries in the rating matrix can be missing not at random (MNAR) and leveraging a significant and powerful latent feature extraction using a deep generative model. The main contributions of this paper are:

- While the current collaborative methods apply MF directly on a sparse user-item interaction matrix, we introduce an efficient Top-K recommendation using MF on a denser rating matrix with only interesting items to users.
- We infer prior preferences of users for items by extracting reliable latent features of users and items using a deep generative model. We also take into account some biases that might impact the pre-use preferences of users.
- We consider the different rating distributions by calculating a threshold preference score for each target user; interesting unrated items are selected according to personalized preference scores.
- Experimental results on two real-world datasets demonstrate that the proposed PreFImp model outperforms the baseline methods with an average improvement margin of 6.45% and 3.91% in HR and NDCG, respectively.

The remainder of this paper is structured into four sections as follows. We first provide in section 2 a summary of the literature review. In section 3, the proposed preference-based imputation method is explained in detail. Section 4 reports experimental findings and compares the proposed model's performance with state-of-the-art approaches. Finally, section 5 presents the paper's conclusions and future directions. The notations used in this paper are briefly summarized and defined in Table 1.

2. Related work

The Top-K recommendation aims to determine a personalised sorted list of K items according to the user's preferences. The efficiency of such ranking-based systems crucially depends on how they deal with sparse and missing entries in the user-item rating matrix [2]. Data imputation-based methods are proposed to overcome rating sparsity without employing auxiliary information [5-7]. Yuan et al. [12] proposed the imputation-based SVD (ISVD) approach to generate and then include imputed ratings into the SVD model by inferring reliable neighbors for users and items. The PDMF imputation method [5] produces preliminary data to constrain the learning in MF. Although only selected neighbors' ratings are employed for imputation in ISVD and the correlations between the learned original, preliminary, and concatenated preferences are examined in PDMF, such neighborhood-based techniques do not estimate uncertainty in rating matrix's empty cells to impute missing values only to relevant positions [1].

Trust-based methods [6, 7] substitute unavailable evaluations with imputed entries by exploring the trust network information of users and items. Such trust-aware techniques can enhance the performance of Non-negative Matrix Factorization (NMF), especially for cold-start new users who need to assess more items [7, 13, 14]. For an efficient Top-K recommendation, some models consider the unobserved entries in the rating matrix as negative preferences [10, 11], while they are ignored in prediction-based approaches. By employing the one class collaborative filtering (OCCF) strategy [15], the zero-injection method [16] treats missing entries as negative evaluation examples with uniform weights and assigns single zero values to identified uninteresting items. However, such assumptions bias the experimentation model toward popular and positively favored items [8].

Even though the above-outlined imputation approaches outperform baseline collaborative techniques, they assume that missing values in the rating matrix are missing at random (MAR) or missing completely at random (MCAR) [8], which would still impede their performance in capturing complex correlations between users and items.

Unlike baseline imputation approaches, we assume that some entries in the rating matrix can be missing not at random (MNAR) since users in real-world applications choose items they want to interact with. Furthermore, we don't ignore unrated positive items by not treating all items with missing entries as negative examples but instead considering

Received: December 13, 2022. Revised: March 14, 2023.

Table 1. Important notations

Notation	Definition
$U = \{u_0 \dots u_n\}$ and $I = \{i_0 \dots i_m\}$	The set of n users and the set of m items
$R = U \times I$ and R^{ob}	The sparse rating matrix with r_{ui} entries and the observed rating matrix of user-item pairs
P	The observed prior preferences matrix with p_{ui} entries
I_u^{ob} and I_u^{un}	The set of observed items by user u and the set of unrated items by user u
I_u^{+} and I_u^{-}	The set of interesting items for user u and the set of uninteresting items for user u
I_u^{un+} and I_u^{un-}	The set of unrated positive items for user u and the set of unrated negative items for user u
$\hat{R}$	The reconstructed denser rating matrix with $\hat{r}_{ui}$ entries
L	The latent preference matrix with ℓ_{ui} entries
$\aleph(u)$	The set of most similar users to user u
$\hat{P}$	The imputed preference matrix with $\hat{p}_{ui}$ entries
$C\hat{P}$	The categorical preference matrix with $c\hat{p}_{ui}$ entries
$pos(i)$	The positivity of item i
$D(i)$	The density of item i
$R_e(i)$	The relevancy of item i
$\rho(u)$	The threshold preference score for user u
$\theta = \{\theta_1 \dots \theta_k\}$	The set of categorical preferences
$\mathcal{H}(u)$	The maximum value of categorical preferences of user u
$\mathcal{F}(u)$	The most frequent categorical preference of user u
R^{+}	The imputed rating matrix with only interesting items

that unrated items can be interesting or uninteresting to users.

3. Proposed approach

To overcome the challenges of traditional Top-K recommendation techniques, we propose PreFImp, a preference-based imputation approach with five main phases. First, we identify the observed prior preferences for users on items (Fig. 1(a)). Next, we aim for an accurate reconstruction of the sparse rating matrix using a deep generative model by learning efficient latent features of users and items (Fig. 1(b)). The reconstructed denser matrix is used to capture the latent preferences of users toward items by inferring a set of similar neighbors for each active user u (Fig. 1(c)). Then, we impute missing preferences of users for unrated items, i.e., $r_{ui} = null$ (Fig. 1(d)) by also taking into consideration some biases that might impact the prior preferences of users (i.e., item popularity and item overall relevancy). Finally, we identify the set of positive unrated items for the target user u by comparing the imputed preference $\hat{p}_{ui}$ of the user u for item i with u's estimated threshold preference score (Fig. 1(e)). The sets of positive items for each user are leveraged to impute a rating matrix to solve the sparsity problem in the Top-K recommendation.

3.1 Inference of observed prior preferences

To infer the observed prior preferences, we convert the rating matrix $R = (r_{ui})_{nxm}$ to a binary prior preferences matrix $P = (p_{ui})_{nxm}$ (Fig. 1(a)), where $p_{ui} = 1$ indicates that the user u has the highest pre-preference for the item i, since the rating of user u on the item i is not missing, i.e., $r_{ui} \epsilon\, R^{ob} \subseteq R$, where R^{ob} is a subset of user-item pairs with known ratings, i.e., $R^{ob} = \{r_{ui} \epsilon\, R \mid r_{ui} \neq null\}$. An item i that belongs to the set of observed items by user u, i.e., $i \,\epsilon\, I_u^{ob}$, must be interesting to u in the beginning, as he/she decided to experience. In other words, all rated items by the user u are interesting items to user u since they have led to a user-item interaction described by an available rating r_{ui}, i.e., $I_u^{ob} \subseteq I_u^{+}$. However, an item i that has not been rated by user u, i.e., $r_{ui} = null$, can be interesting or uninteresting to user u, i.e., $i \epsilon\, I_u^{un+}$ or $i \epsilon\, I_u^{un-}$. Therefore, the main challenge is accurately determining the prior preferences for items I_u^{un}.

3.2 Extraction of user and item features

In this stage, we aim to accurately reconstruct the sparse rating matrix $R = (r_{ui})_{nxm}$ using the deep belief network (DBN) [17] (Fig. 1(b)). We employ the deep generative model for missing rating prediction by learning relevant hidden features of user u and item i. The DBN is formed by stacking multiple probabilistic building blocks called restricted boltzmann machines (RBMs) [18, 19] used to apprehend one layer of latent features at a

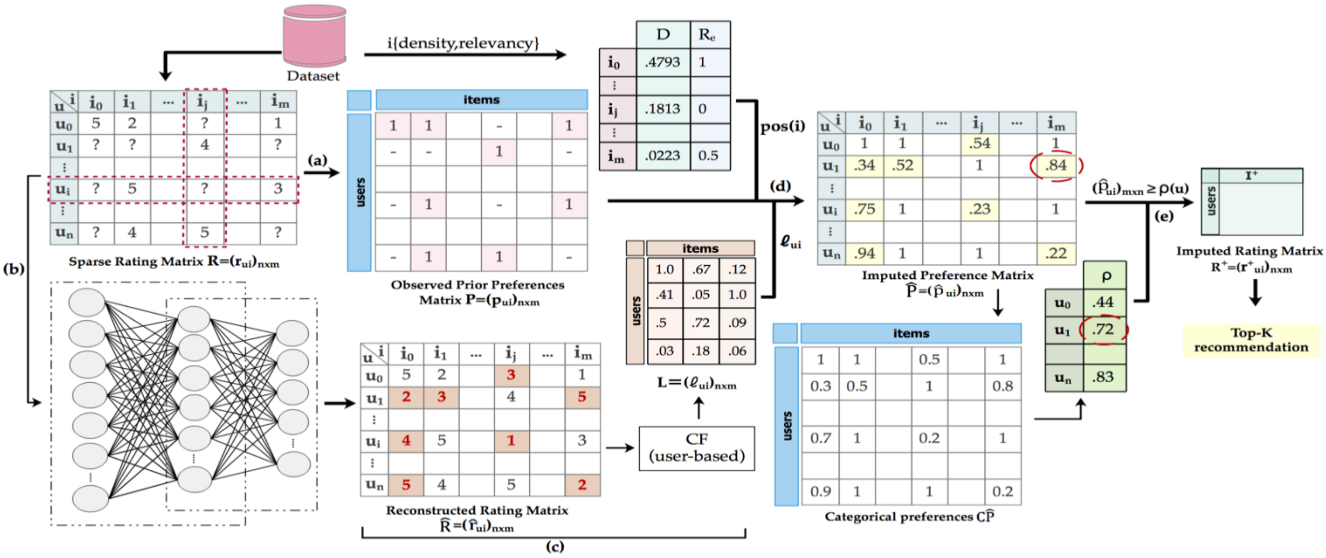

Figure. 1 Overview of the proposed PreFImp approach

time. An RBM is a bipartite graph-based model composed of two layers, a visible layer $\{v_1 \dots v_a \dots v_A\}$ with A visible units and a hidden layer $\{h_1 \dots h_b \dots h_B\}$ with B hidden nodes. Since no unit has a connection with another unit of the same layer, the values of units in the visible or hidden layer are independent concerning the units of the same layer. Therefore, the joint probability $P_\Theta(v,h)$ of each visible and hidden unit can be computed as follows:

$$P_\Theta(v,h) = \frac{e^{-E(v,h;\,\Theta)}}{\wp(\Theta)} \qquad (1)$$

Where $\wp(\Theta) = \sum_{v,h} e^{-E(v,h;\,\Theta)}$ defines the normalization factor and $E(v,h;\,\Theta)$ is the overall energy of the joint configuration $\{v,h\}$, which can be described by the following equation:

$$E(v,h;\,\Theta) = -\sum_{a=1}^{A}\sum_{b=1}^{B} W_{ab}\, v_a h_b - \sum_a c_v\, v_a - \sum_b c_h\, h_b \qquad (2)$$

Where W_{ab} is the weight between v_a and h_b; c_v and c_h are, respectively, the bias vectors for visible and hidden units that form the parameter set $\Theta = \{\mathrm{W}, c_v, c_h\}$ of the RBM. The probability of a visible sample is calculated over all the conditional probabilities of hidden vectors using Eq. (3):

$$P_\Theta(v) = \sum_h P_\Theta(v,h) = \frac{\sum_h e^{-E(v,h;\,\Theta)}}{\sum_{v,h} e^{-E(v,h;\,\Theta)}} \qquad (3)$$

The main goal of the RBM is maximizing the log-likelihood $logP_\Theta(v)$ to learn the weight matrix $W = \{W_{ab}\}$ that ensures an effective reconstruction of the sparse rating matrix R:

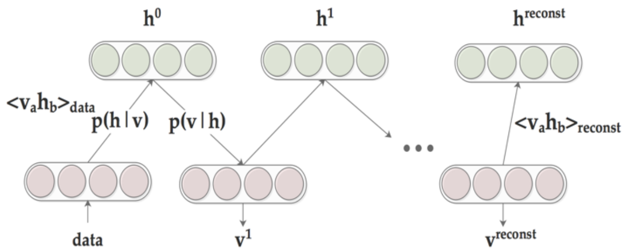

Figure. 2 Gibbs sampling

$$\frac{\partial logP_\Theta(v)}{\partial W_{ab}} = < v_a h_b >_{data} - < v_a h_b >_{reconst} \qquad (4)$$

$$\Delta W_{ab} = \eta < v_a h_b >_{data} - < v_a h_b >_{reconst} \qquad (5)$$

Where $<.>_{data}$ stands for the expectation under training data distribution, $<.>_{reconst}$ defines the expectation under reconstructed-data distribution obtained by the RBM model, and η is the learning rate. Since there are no hidden–hidden connections, the states of hidden units are conditionally independent. Given a visible vector v, the hidden unit value h_b can thereby be determined as follows:

$$P(h_b = 1|v) = \sigma\left(\sum_{a=1}^{A} W_{ab} v_a + c_h\right) \qquad (6)$$

Where $\sigma(x) = 1/1 + e^{-x}$ is the logistic sigmoid function, hence, $<.>_{data}$ can be easily obtained. On the other hand, $<.>_{reconst}$ can be computed by applying the contrastive divergence (CD) [17] to ensure the initialization of v using training data and then performing t steps of Gibbs sampling, as indicated in Fig 2, which speeds up the learning process.

Once the stacked RBMs of the DBN model are pre-trained, where each RBM's hidden layer $H = \{h_b\}$ performs as the visible layer $V = \{v_a\}$ for the

Received: December 13, 2022. Revised: March 14, 2023.

following RBM, the bottom-up weights of the unsupervised DBN are leveraged to initialize a multi-layer neural network, thus accomplishing a discriminative fine-tuning based on the back-propagation of error derivatives. At the end of this phase, the fine-tuned DBN model discovers optimal parameters that yield a denser reconstructed rating matrix $\hat{R} = (\hat{r}_{ui})_{nxm}$.

3.3 Inference of latent preferences

To learn the latent preferences matrix $L = (\ell_{ui})_{nxm}$ (Fig. 1(c)), we first infer a set of similar neighbors for each target user u. The degree to which two users, u and u', are similar is determined by considering the denser ratings learned by the DBN model $\hat{r} \epsilon \hat{R}$ on the common observed items by users u and u', i.e., $I^{ob}_{u,u'}$. The user-based similarity is computed as follows:

$$sim(u,u') = \frac{\sum_{i \,\epsilon I^{ob}_{u,u'}} (\hat{r}_{ui} - \overline{\hat{r}_u})(\hat{r}_{u'i} - \overline{\hat{r}_{u'}})}{\sqrt{\sum_{i \,\epsilon I^{ob}_{u,u'}} (\hat{r}_{ui} - \overline{\hat{r}_u})^2} \sqrt{\sum_{i \,\epsilon I^{ob}_{u,u'}} (\hat{r}_{u'i} - \overline{\hat{r}_{u'}})^2}} \quad (7)$$

Where $\overline{\hat{r}_u}$ and $\overline{\hat{r}_{u'}}$, are respectively the average of predicted ratings of users u and u'. We predict the latent preference ℓ_{ui} of the target user u for the item i by leveraging denser ratings $\hat{r} \epsilon \hat{R}$ assessed to the item i, by the most similar users to the active user u, i.e., $\aleph_i(u)$:

$$\ell_{ui} = \frac{\sum_{u' \epsilon \aleph_i(u)} sim(u,u') \hat{r}_{u'i}}{\sum_{u' \epsilon \aleph_i(u)} |sim(u,u')|} \quad (8)$$

Since similar users may vary in their use of rating scales (two users have the same degree of appreciation for item i, but the first rates it at 5 and the second at 4), the average of the reconstructed ratings of the target user u, i.e., $\overline{\hat{r}_u}$ as well as the average of denser evaluation scores of similar users are incorporated. The latent preference prediction ℓ_{ui}, is, therefore, computed as follows:

$$\ell_{ui} = \overline{\hat{r}_u} + \frac{\sum_{u' \,\epsilon \aleph} sim(u,u')(\hat{r}_{u'i} - \overline{\hat{r}_{u'}})}{\sum_{u' \epsilon \aleph_i(u)} |sim(u,u')|} \quad (9)$$

3.4 Imputation of the preference matrix

To impute the preference $\hat{p}_{ui}$ (Fig. 1(d)) of a user u for an item i with a missing rating, i.e., $i \,\epsilon\, I^{un}_u$, instead of considering that user u is more likely to have a low preference for the unrated item i, i.e., $i \,\epsilon\, I^{un-}_u$ [10, 11, 15, 16], and treat all the unobserved items as negative examples for user u, i.e., $I^{un}_u \subseteq I^-_u$, we leverage the predicted latent preference of the user u toward the item i, i.e., ℓ_{ui} learned from the denser rating matrix. We also take into consideration that the preferences $\hat{P} = (\hat{p}_{ui})_{nxm}$ are biased to item positivity $pos(i)$. The latter encompasses two major points that may impact the pre-preference of a user u toward an item i: (1) The popularity or the density of an item $D(i)$ may influence users' choice of items. By referring to the long-tail distribution of evaluated items in many real-world applications, one may see that a great number of feedback ratings are compressed into a small fraction of popular items [8, 9]. (2) A user's prior preference towards an item can also be associated with the average relevancy $R_e(i)$ of that item (a user is more likely to watch a movie that the majority of users have positively rated). The overall positivity of item i is calculated as follows:

$$pos(i) = \frac{D(i) + R_e(i)}{2} \quad (10)$$

Here $D(i)$ is obtained by computing the ratio of observed ratings on item i to the total number of users (i.e., $D(i) = N_R / N_{users}$), and $R_e(i)$ refers to one of the three relevancy classes for the item i, i.e., $\{0, 0.5, 1\}$ based on i's average rating. Therefore, the imputed preference of the user u for item i can be estimated as follows:

$$\hat{p}_{ui} = (w\, pos(i) + \ell_{ui})/1.5 \quad (11)$$

Where w is a weight parameter set as 0.5.

3.5 Imputation of rating matrix for Top-K recommendation

Now, we discuss how the imputed preferences $\hat{P} = (\hat{p}_{ui})_{nxm}$ can be used to achieve efficient Top-K recommendation results of any CF model. When applying matrix factorization techniques directly on a sparse rating matrix to generate a Top-K recommendation list for the user u, the linear dot-product is inadequate to learn the complex hidden structure of u 's interaction data. Furthermore, unobserved ratings are considered zero entries, which leads to ineffective learning of latent factors of users and items [2]. Taking into account these constraints, we apply SVD on the imputed rating matrix R^+ to generate accurate Top-K recommendation lists for users, with only positive items I^+ for each user. To identify the set of positive unrated items for the user u, i.e., I^{un+}_u, we first convert the imputed preference matrix $\hat{P} = (\hat{p}_{ui})_{nxm}$ to a categorical preference matrix $C\hat{P} =$

$(c\hat{p}_{ui})_{nxm}$, so that we can calculate the threshold preference score for the user u as follows:

$$\rho(u) = (\alpha \sum_{\theta' \in \{\theta_1 \dots \theta_k\}} \frac{\mathcal{H}(u) > dist(\theta)}{|\theta|} + \beta\, \mathcal{F}(u) + \delta\, \mathcal{H}(u))/2.5 \quad (12)$$

Given $\theta = \{\theta_1 \dots \theta_k\}$, the set of categorical preferences, i.e., $c\hat{p}_{ui} \in \theta$, $\mathcal{H}(u)$, the maximum value of categorical preferences $c\hat{p}$ of user u for unrated items I_u^{un}, i.e., $\mathcal{H}(u) = \max(\theta, c\hat{p} \neq 1)_u$, $\mathcal{H}(u) > dist(\theta)$ is the number of distinct categorical preferences that are ranked lower than $\mathcal{H}(u)$, i.e., $\theta' < \theta$. $\mathcal{F}(u)$ is the most frequent categorical preference of user u, and $|\theta|$ is used for normalization. Parameters α, β, and δ are set as 0.5, 0.5, and 1.5, respectively.

We conclude that an item i with missing rating is an item that may interest the user u, i.e., $i \in I_u^{un+}$ (Fig. 1(e)) by comparing the imputed preference $\hat{p}_{ui}$ of the user u for item i with the threshold score of user u, i.e., $\rho(u)$. Therefore, the set of unrated positive items of the user u is determined as follows:

$$I_u^{un+}(\rho(u)) = \{i | \rho(u) \leq rank(\hat{p}_{ui}) < 1,\ r_{ui} = null\} \quad (13)$$

Fig. 3 depicts the imputed preference matrix $\hat{P} = (\hat{p}_{ui})_{nxm}$ for users $\{u_0 \dots u_4\}$, where the colored cells are positive unrated items I^{un+} for users according to their threshold scores ρ. Note that if we set a unique threshold value ρ for all users, the positive items I^+ for each user will change. The proposed preference-based model considers the different rating distributions to infer accurate user preferences. For example, two users, u and u' might give the same rating of 3 to item i. Nevertheless, the significance of the evaluation value can be interpreted in various ways. On a scale of 1 to 5, the rating value of 3 may indicate the satisfaction of user u toward item i (in case of user u rarely gives a rating of 5). On the other hand, user u' could choose the rating value of 3 for a less interesting item. Furthermore, by employing the personalized preference score ρ for each user, we deal with the problem of grey sheep users who have unique and unusual interests [20], by tackling the challenge of creating precise profiles for such users.

Given $R^+ = (r_{ui}^+)_{nxm}$ the imputed rating matrix with only positive items, i.e., $i \in I_u^+$, SVD factorizes R^+ into two low-rank matrices, a z-dimensional matrix S that contains user factors, i.e., $s_u \in \mathbb{R}^z$ and a z-dimensional matrix Q of item factors, i.e., $q_i \in \mathbb{R}^z$. Prediction of the rating r_{ui}^* is calculated as the inner product of user and item related feature vectors:

	i_0	i_1	i_2	i_3	i_4	ρ
u_0	1	1	0.12	0.42	0.52	.44
u_1	0.34	0.52	0.79	0.69	0.8	.72
u_2	0.69	0.92	0.54	1	0.71	.68
u_3	0.45	0.94	1	0.75	0.55	.89
u_4	0.75	1	0.28	0.81	1	.74

Figure. 3 Positive unrated items according to imputed preferences and threshold scores of users

Table 2. Statistics of experimental datasets

Datasets	MovieLens 100K	MovieLens 1M
#Users	943	6 040
#Items	1 682	3 706
#Ratings	100 000	1 000 209
Sparsity	93.7%	95.53%

$$r_{ui}^* = s_u q_i^T \quad (14)$$

By applying SVD directly on the imputed ratings r_{ui}^+ of the matrix R^+, we can learn efficient latent correlations between users and items, which enhances the final Top-K recommendation results. This is achieved by minimizing the following objective function:

$$\min \sum_{(u,i)} (r_{ui}^+ - s_u q_i^T)^2 + \lambda(\|s_u\|^2 + \|q_i\|^2) \quad (15)$$

Where λ is used as a regularization parameter to prevent overfitting.

4. Experimental results

4.1 Dataset description

Experimental evaluations are conducted on two real-world benchmark datasets: MovieLens 100K and MovieLens 1M. Statistics and Sparsity levels of both datasets are shown in Table 2. The following equation is used to calculate the sparsity level:

$$Sparsity = 1 - \frac{N_{\#Ratings}}{N_{\#Users} \times N_{\#Items}} \quad (16)$$

4.2 Metrics

The hit rate (HR) and the normalized discounted

Table 3. Performance of the proposed model compared with baseline approaches

	MovieLens 100K				MovieLens 1M			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
ItemPop	0.2831	0.1892	0.4060	0.2264	0.3101	0.2126	0.44584	0.2562
BPR [10]	0.4550	0.3172	0.5801	0.3312	0.4810	0.3315	0.6577	0.3910
NeuMF [11]	0.4645	0.3183	0.6257	0.3704	0.5089	0.3562	0.6833	0.4124
NGCF [21]	0.4274	0.2889	0.5864	0.3402	0.5017	0.3437	0.6688	0.3977
DeepCF [22]	0.5009	<u>0.3502</u>	0.6819	0.3981	0.5508	0.3941	0.7253	0.4416
DAVE [23]	0.4995	0.3415	0.6723	0.3971	0.5417	0.3761	0.7185	0.4334
CFFNN [24]	<u>0.5113</u>	0.3345	<u>0.7012</u>	<u>0.4077</u>	0.5730	<u>0.4102</u>	<u>0.7511</u>	<u>0.4582</u>
TBRS [25]	0.4710	0.313	0.6452	0.3696	<u>0.5941</u>	0.4074	0.7306	0.4565
PreFImp	**0.5853**	**0.3993**	**0.7586**	**0.4348**	**0.6538**	**0.4448**	**0.8182**	**0.5039**
Improvement	7.4%	4.91%	5.74%	2.71%	5.97%	3.46%	6.71%	4.57%

cumulative gain ($NDCG$) are employed as ranking quality measures for the Top-K recommendation with K selected from $\{5,10\}$. $HR@K$ evaluates the number of correctly predicted items in the Top-K list in the test set, while $NDCG@K$ considers the hits' position.

4.3 Performance comparison and results discussion

We compare the performance of the proposed model against the following eight benchmark approaches:

- ItemPop is a non-personalized standard method that provides ranking recommendations solely based on items' popularity.
- Bayesian personalized ranking (BPR) [10] is an MF model that leverages pairwise learning for a personalized ranking-based recommendation.
- Neural matrix factorization (NeuMF) [11] combines the linear modelling of generalized matrix factorization with non-linear kernel-based multi-layer perceptron for extracting feature interaction.
- Neural graph collaborative filtering (NGCF) [21] proposes a propagation layer to incorporate the collaborative bipartite graph into embedding users and items.
- Deep collaborative filtering (DeepCF) [22] is a Deep Neural Network-based ranking approach that incorporates matching function learning with deep representation learning of users and items.
- Dual adversarial variational embedding (DAVE) [23] merges adversarial training with variational auto-encoder to learn multi-modality preferences from the user-item matrix.
- Cross feature fusion neural network (CFFNN) [24] apprehends users' preferences toward items based on a fusion mechanism of user and item features.
- Tensor based recommender system (TBRS) [25] combines collaborative filtering with sequential recommendation to model users as rectangles that capture their interests.

To determine the hyper-parameters' viability and choose their optimal values, several experiments are conducted. We train the DBN model using a momentum of 0.1, a batch size of 32 for all datasets. We set the learning rate as 0.0005 for MovieLens 100K and MovieLens 1M. For the Top-K recommendations, K is selected from $\{5,10\}$.

Table 3 shows the performance results of the proposed PreFImp model and baseline approaches in terms of HR and NDCG with respect to the ranking position K. Each column's highest score is bold, and the underlined value represents the second-best score. The experimental results demonstrate that the proposed model attains the best performance on both datasets, significantly outperforming all the state-of-the-art approaches in terms of all employed evaluation metrics.

Unlike the proposed PreFImp baselines, ItemPop merely considers the popularity of items and does not take into account users' personalized preferences; consequently, ItemPop results in the lowest ranking accuracy.

On MovieLens 100K, PreFImp demonstrates an average improvement margin of 6.57% against the best competitor (i.e., CFFNN) in HR results. Regarding NDCG, when setting K as 5, PreFImp surpasses the best approach among baselines (i.e., DeepCF) with 4.91% and achieves 2.71% relative improvement compared with the best competitor (i.e., CFFNN) when K=10. CFFNN is exploring several multilayer perceptrons (MLPs) for enhanced extraction and integration of non-linear features of users and items. DeepCF also applied MLP for complex matching function learning and capturing low-rank correlations between user-item pairs.

However, despite their effectiveness, CFFNN and DeepCF show inferior results compared with the PreFImp approach because they rely on simple MLP for feature extraction in contrast to the proposed PreFImp, which leverages a more powerful deep generative model to capture higher-level hidden feature interaction of users and items in the sparse rating matrix; thus effectively generating a reconstructed denser matrix to infer similar neighbors for the target user.

On MovieLens 1M, TBRS shows the best HR@5 ranking result but fails to surpass PreFImp. In fact, using rectangles to model users' preferences in TBRS makes up for typical single-sided user vector representations by allowing tensor-based modeling of numerous aspects of the user's interests. PreFImp also provides the higher HR@10 performance (i.e., 0.8182) and shows an average relative improvement of 4.01% than CFFNN, which has led to the best NDCG ranking results among benchmark approaches. We can also note that when the parameter K is set as 5 and 10, all the models have led performance results that can be placed in the following sequence (PreFImp > CFFNN > TBRS > DeepCF > DAVE > NeuMF > NGCF > BPR > ItemPop) DAVE exhibits the closest ranking results to DeepCF and performs almost better than other MF-based approaches but shows limited outcomes compared with PreFImp. This proves that building the new PreFImp with a pre-training phase for a reliable weight initialization and then a supervised fine-tuning step for further optimization provides an enhanced generalization performance which is more efficient than simply adding adversarial noise to improve the model's regularization. On the other hand, although NeuMF combines generalized matrix factorization (GMF) linearity with MLP non-linearity to model different structures of the rating matrix, it shows inferior results than other competitive deep learning-based techniques. By applying MF on the denser generated matrix, PreFImp does not ignore significant correlations of user-item interactions that the dot product can learn of hidden features. Moreover, PreFImp leverages a powerful architecture rather than dual embedding spaces that lead to overfitting, which might be a reason that hinders the NeuMF’s performance.

Besides, PreFImp vastly outperforms graph-based approaches (i.e., NGCF) that can leverage the bipartite graph structure of irrelevant user-item interactions. This may lead to a poor collaborative signal encoded in the embedding process, which conveys bias and disturbing user interest inference.

Instead of relying primarily on the traditional uniform negative sampler used in BPR to assume unseen items as negative instances, PreFImp considers that unrated items can be interesting or uninteresting to the user and estimates the latter's latent preferences towards these items, which is more effective for real-world sparse datasets that have a large number of unobserved items.

Unlike the baseline approaches, PreFImp aims for an accurate, personalized ranking-based recommendation for users by imputing their prior preferences for items. PreFImp also considers the different rating distributions and tackles the problem of unique and unusual interests of grey sheep users by calculating a personalized threshold preference score $\rho(u)$ for each target user, clearly demonstrating highly encouraging ranking accuracy results and confirming the efficiency of the proposed preference-based model in Top-K recommendation.

5. Conclusions and future work

In this paper, we introduced a novel preference-based imputation approach to overcome sparsity issues in item ranking-based models and enhance the Top-K recommendation performance. The proposed PreFImp method tackles traditional matrix factorization limitations by inferring reliable prior preferences of users for unrated items and leveraging a powerful and deep latent feature extraction to incorporate the imputed denser rating matrix with only interesting items for users in the Top-K recommendation. Moreover, the proposed technique not only considers the biases that might impact users' pre-use preferences but also considers the different rating distributions by calculating a threshold preference score for each target user, thus selecting effectively interesting unrated items based on personalized preference scores. Experimental evaluations with different sparsity levels prove that the proposed PreFImp model significantly enhances the Top 5 Recommendation and outperforms recent state-of-the-art approaches with 6.68% and 4.18% in HR and NDCG, respectively. For the Top 10 Recommendation, PreFImp achieves the higher accuracy results with an average improvement of 6.22% and 3.64% in HR and NDCG, respectively. Future directions include the hybridization of PreFImp with other feature-learning techniques to explore further latent representations of users and items under cold-start conditions.

Received: December 13, 2022. Revised: March 14, 2023.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Nouhaila Idrissi and Ahmed Zellou; methodology, Nouhaila Idrissi and Ahmed Zellou; software, Nouhaila Idrissi; validation, Nouhaila Idrissi, Ahmed Zellou, and Zohra Bakkoury; formal analysis, Nouhaila Idrissi and Ahmed Zellou; investigation, Nouhaila Idrissi; data curation, Nouhaila Idrissi; writing—original draft preparation, Nouhaila Idrissi; writing—review and editing, Nouhaila Idrissi, Ahmed Zellou, and Zohra Bakkoury; visualization, Nouhaila Idrissi; supervision, Ahmed Zellou and Zohra Bakkoury.

References

[1] S. Hwang and D. K. Chae, "An Uncertainty-Aware Imputation Framework for Alleviating the Sparsity Problem in Collaborative Filtering", In: *Proc. of International ACM Conf. On Information & Knowledge Management*, Atlanta, GA, USA, pp. 802-811, 2022.

[2] N. Idrissi and A. Zellou, "A systematic literature review of sparsity issues in recommender systems", *Social Network Analysis and Mining*, Vol. 10, No. 1, pp. 1-23, 2020.

[3] Hanafi, E. Pujastuti, A. Laksito, R. Hardi, R. Perwira, A. Arfriandi, and Asroni, "Handling Sparse Rating Matrix for E-commerce Recommender System Using Hybrid Deep Learning Based on LSTM, SDAE and Latent Factor", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 2, pp. 379-393, 2022, doi: 10.22266/ijies2022.0430.35.

[4] K. Muruganantham and S. Shanmugasundaram, "Recommender System using Distributed Improved Predictive Framework with Matrix Factorization and Random Forest", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, pp. 447-457, 2022, doi: 10.22266/ijies2022.0831.40.

[5] X. Yuan, L. Han, S. Qian, L. Zhu, J. Zhu, and H. Yan, "Preliminary data-based matrix factorization approach for recommendation", *Information Processing & Management*, Vol. 58, No. 1, p. 102384, 2021.

[6] W. S. Hwang, S. Li, S. W. Kim, and K. Lee, "Data imputation using a trust network for recommendation via matrix factorization", *Computer Science and Information Systems*, Vol. 15, No. 2, pp. 347-368, 2018.

[7] F. Alghamedy, X. Wang, and J. Zhang, "Imputing trust network information in NMF-based collaborative filtering", In: *Proc. of the ACMSE Conf*, pp. 1-8, 2018.

[8] B. Pradel, N. Usunier, and P. Gallinari, "Ranking with non-random missing ratings: influence of popularity and positivity on evaluation metrics", In: *Proc. of the Sixth ACM Conf. On Recommender Systems*, Dublin, Ireland, pp. 147-154, 2012.

[9] A. Yuliawati, H. Tohari, R. Mahendra, and I. Budi, "On the Long Tail Products Recommendation using Tripartite Graph", *International Journal of Advanced Computer Science and Applications*, Vol. 13, No. 1, 2022.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, L. S. Thieme, "BPR: Bayesian personalized ranking from implicit feedback", In: *Proc. of International Conf. on Uncertainty in Artificial Intelligence*, Quebec, Canada pp. 452-461, 2009.

[11] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering", In: *Proc. of International Conf. on World Wide Web*, pp. 173-182, 2017.

[12] X. Yuan, L. Han, S. Qian, G. Xu, and H. Yan, "Singular value decomposition based recommendation using imputed data", *Knowledge-Based Systems*, Vol. 163, pp. 485-494, 2019.

[13] N. Idrissi, A. Zellou, O. Hourrane, Z. Bakkoury, and E. H. Benlahmar, "Addressing cold start challenges in recommender systems: towards a new hybrid approach", In: *Proc. of International Conf. on Smart Applications, Communications and Networking (SmartNets)*, Sharm El Sheik, Egypt, pp. 1-6, 2019.

[14] N. Idrissi, A. Zellou, O. Hourrane, Z. Bakkoury, and E. H. Benlahmar, "A New Hybrid-Enhanced Recommender System for Mitigating Cold Start Issues", In: *Proc. of International Conf. on Information Management and Engineering*, London, United Kingdom, pp. 10-14, 2019.

[15] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang, "One-Class Collaborative Filtering", In: *Proc. of International Conf. on Data Mining*, pp. 502-511, 2008.

[16] W. S. Hwang, J. Parc, S. W. Kim, J. Lee, and D. Lee, "Told you i didn't like it": Exploiting uninteresting items for effective collaborative filtering", In: *Proc. of International Conf. on Data Engineering (ICDE)*, pp. 349-360, 2016.

Received: December 13, 2022. Revised: March 14, 2023.

[17] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets", *Neural Computation*, Vol. 18, No. 7, pp. 1527-1554, 2006.

[18] G. E. Hinton, "A practical guide to training restricted Boltzmann machines", *Neural Networks: Tricks of the Trade*, pp. 599-619, 2012.

[19] N. Idrissi, O. Hourrane, A. Zellou, and E. H. Benlahmar, "A Restricted Boltzmann Machine-based Recommender System For Alleviating Sparsity Issues", In: *Proc. of International Conf. on Smart Systems and Data Science (ICSSD)*, Rabat, Morocco, pp. 1-5, 2019.

[20] J. M. Radecka and B. Indurkhya, "A bias detection tree approach for detecting disparities in a recommendation model's errors", *User Modeling and User-Adapted Interaction*, pp. 1-37, 2022.

[21] X. Wang, X. He, M. Wang, F. Feng, and T. S. Chua, "Neural graph collaborative filtering", In: *Proc. of International ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp. 165-174, 2019.

[22] Z. H. Deng, L. Huang, C. D. Wang, J. H. Lai, and S. Y. Philip, "Deepcf: A unified framework of representation learning and matching function learning in recommender system", In: *Proc. of AAAI International Conf. on Artificial Intelligence*, pp. 61-68, 2019.

[23] Q. Yi, N. Yang, and P. Yu, "Dual adversarial variational embedding for robust recommendation", *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[24] R. Yu, D. Ye, Z. Wang, B. Zhang, A. M. Oguti, J. Li, B. Jin, and F. Kurdahi, "CFFNN: Cross feature fusion neural network for collaborative filtering", *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[25] B. Ma, B. Hao, F. Zhang, L. Gao, and X. Ren, "Research on recommendation method based on tensor similarity", In: *Proc. of International Symposium on Computer Technology and Information Science*, pp. 317-325, 2022.