

Adaptive matrix completion for the users and the items in tail

Mohit Sharma
University of Minnesota
Twin Cities, MN, USA
sharm163@umn.edu

George Karypis
University of Minnesota
Twin Cities, MN, USA
karypis@umn.edu

ABSTRACT

Recommender systems are widely used to recommend the most appealing items to users. These recommendations can be generated by applying collaborative filtering methods. The low-rank matrix completion method is the state-of-the-art collaborative filtering method. In this work, we show that the skewed distribution of ratings in the user-item rating matrix of real-world datasets affects the accuracy of matrix-completion-based approaches. Also, we show that the number of ratings that an item or a user has positively correlates with the ability of low-rank matrix-completion-based approaches to predict the ratings for the item or the user accurately. Furthermore, we use these insights to develop four matrix completion-based approaches, i.e., Frequency Adaptive Rating Prediction (FARP), Truncated Matrix Factorization (TMF), Truncated Matrix Factorization with Dropout (TMF + Dropout) and Inverse Frequency Weighted Matrix Factorization (IFWMF), that outperforms traditional matrix-completion-based approaches for the users and the items with few ratings in the user-item rating matrix.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; **Personalization**; **Recommender systems**;

KEYWORDS

Recommender systems; Collaborative filtering; Matrix completion; Matrix factorization

ACM Reference Format:

Mohit Sharma and George Karypis. 2019. Adaptive matrix completion for the users and the items in tail. In *Proceedings of the 2019 World Wide Web Conference (WWW '19)*, May 13–17, 2019, San Francisco, CA, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3308558.3313736>

1 INTRODUCTION

Recommender systems are used in e-commerce, social networks, and web search to suggest the most relevant items to each user. Recommender systems commonly use methods based on Collaborative Filtering [15], which rely on historical preferences of users over items in order to generate recommendations. These methods

This work was supported in part by NSF (1447788, 1704074, 1757916, 1834251), Army Research Office (W911NF1810344), Intel Corp, and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.

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 '19, May 13–17, 2019, San Francisco, CA, USA

© 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-6674-8/19/05.

<https://doi.org/10.1145/3308558.3313736>

predict the ratings for the items not rated by the user and then select the unrated items with the highest predicted ratings as recommendations to the user.

In practice, a user may not rate all the available items, and hence we observe only a subset of the user-item rating matrix. For the task of recommendations, we need to complete the matrix by predicting the missing ratings and select the unrated items with high predicted ratings as recommendations for a user. The matrix completion approach [3] assumes that the user-item rating matrix is low rank and estimates the missing ratings based on the observed ratings in the matrix. The state-of-the-art collaborative filtering methods, e.g., Matrix Factorization (MF) [10] are based on the matrix completion approach.

Assuming that the user-item rating matrix is low-rank, it was shown that in order to accurately recover the underlying low-rank model of a $n \times n$ matrix of rank r , at least $O(nr \log(n))$ entries in the matrix should be sampled uniformly at random [4]. However, most real-world rating matrices exhibit a skewed distribution of ratings as some users have provided ratings to few items and certain items have received few ratings from the users. This skewed distribution may result in insufficient ratings for certain users and items, and can negatively affect the accuracy of the matrix completion-based methods.

This paper investigates how the skewed distribution of ratings in the user-item rating matrix affects the accuracy of the matrix completion-based methods and shows by extensive experiments on different low-rank synthetic datasets and as well as on real datasets that the matrix completion-based methods tend to have poor accuracy for the items and the users with few ratings. Moreover, this work illustrates that as we increase the number of latent dimensions, the prediction performance for the items and the users with sufficiently many ratings continues to improve, whereas the accuracy of the items and the users with few ratings degrades. This suggests that because of over-fitting, the matrix completion-based methods for large number of latent dimensions do not generalize well for the items and the users with few ratings.

Building on this finding, we develop four matrix completion-based approaches that explicitly consider the number of ratings received by an item or provided by a user to estimate the rating of the user on the item. Specifically, we introduce (i) Frequency Adaptive Rating Prediction (FARP) method, which uses multiple low-rank models for different frequency of the users and the items; (ii) Truncated Matrix Factorization (TMF) method, which estimates a single low-rank model that adapts with the number of ratings a user and an item has; (iii) Truncated Matrix Factorization with Dropout (TMF + Dropout) method, which is similar to TMF but probabilistically select the ranks for the users and the items; and (iv) Inverse Frequency Weighted Matrix Factorization (IFWMF) method,

which weighs the infrequent users and items higher during low-rank model estimation. Extensive experiments on various datasets demonstrate the effectiveness of the proposed approaches over traditional MF-based methods by improving the accuracy for the items (up to 53% improvement in RMSE) and the users (up to 8% improvement in RMSE) with few ratings.

2 RELATED WORK

The current state-of-the-art methods for rating prediction are based on matrix completion, and most of them involve factorizing the user-item rating matrix [7, 9, 10]. In this work, our focus is on analyzing the performance of the matrix completion-based MF approach and use the derived insights to develop an approach that performs better for the users and the items with few ratings in the user-item rating matrix. These approaches estimate user-item rating matrix as a product of two low-rank matrices known as the user and the item latent factors. If for a user u , the vector $\mathbf{p}_u \in \mathbb{R}^r$ denotes the r dimensional user’s latent factor and similarly for the item i , the vector $\mathbf{q}_i \in \mathbb{R}^r$ represents the r dimensional item’s latent factor, then the predicted rating ($\hat{r}_{u,i}$) for user u on item i is given by

$$\hat{r}_{u,i} = \mathbf{p}_u \mathbf{q}_i^T. \quad (1)$$

The user and the item latent factors are estimated by minimizing a regularized square loss between the actual and predicted ratings

$$\underset{\mathbf{p}_u, \mathbf{q}_i}{\text{minimize}} \quad \frac{1}{2} \sum_{r_{ui} \in R} (r_{ui} - \mathbf{p}_u \mathbf{q}_i^T)^2 + \frac{\beta}{2} (\|\mathbf{p}_u\|_2^2 + \|\mathbf{q}_i\|_2^2), \quad (2)$$

where R is the user-item rating matrix, r_{ui} is the observed rating of user u on item i , and parameter β controls the Frobenius norm regularization of the latent factors to prevent overfitting.

In another related work [21], it was shown that the lack of uniform distribution of ratings in the observed data could lead to folding, i.e., the unintentional affinity of dissimilar users and items in the low-rank space estimated by matrix completion-based methods. For example, the absence of an explicit rating between a child viewer and a horror movie could lead to an estimation of the corresponding latent factors such that both the child viewer and the horror movie are close in the low-rank space leading to the erratic recommendation of horror movies to the child viewer. We believe that the non-uniform distribution of the ratings is the reason behind this phenomenon.

The non-uniform distribution of ratings can also be viewed as an instance of rating data *missing not at random* (MNAR) [12]. The proposed solutions to MNAR model the missing data to improve the generated recommendations [13, 17, 19, 20]. However, in our work, we focus on the skewed distribution of ratings which often comes as a result of either new items or new users that are added to the system, or the items that are not popular to get many ratings. We use our analysis to develop a matrix completion-based approach to improve rating prediction for the users who have provided few ratings on items or for the items that have received few ratings from the users.

Table 1: Datasets used in experiments

Dataset	users	items	ratings	μ_u^a	σ_u^b	μ_i^a	σ_i^b	% [†]
Flixster (FX)	147K	48K	8.1M	55	226	168	934	1e-1
Movielens (ML)	229K	26K	21M	92	190	786	3269	3e-1
Yahoo Music (YM)	143K	136K	9.9M	69	199	73	141	5e-4
Netflix (NF)	354K	17K	9.5M	27	59	535	1693	1e-3

^a Average ratings per user (μ_u) or per item (μ_i).

^b Standard deviation of ratings per user (σ_u) or per item (σ_i).

[†] The percentage of observed ratings in the dataset.

3 IMPACT OF SKEWED DISTRIBUTION

As described in Section 1, the matrix completion-based methods can accurately recover the underlying low-rank model of a given low-rank matrix provided entries are observed uniformly at random from the matrix. However, the ratings in the user-item rating matrix in real-world datasets represent a skewed distribution of entries because some users have provided ratings to few items and certain items have received few ratings from the users.

In order to study how the skewed distribution of ratings in real datasets affects the ability of matrix completion to accurately complete the matrix (i.e., predict the missing entries) we performed a series of experiments using synthetically generated low-rank rating matrices. In order to generate a rating matrix $R \in \mathbb{R}^{n \times m}$ of rank r we followed the following protocol. We started by generating two matrices $A \in \mathbb{R}^{n \times r}$ and $B \in \mathbb{R}^{m \times r}$ whose values are uniformly distributed at random in $[0, 1]$. We then computed the singular value decomposition of these matrices to obtain $A = U_A \Sigma_A V_A^T$ and $B = U_B \Sigma_B V_B^T$. We then let $P = \alpha U_A$ and $Q = \alpha U_B$ and $R = PQ^T$. Thus, the final matrix R of rank r is obtained as the product of two randomly generated rank r matrices whose columns are orthogonal. The parameter α was determined empirically in order to produce ratings in the range of $[-10, 10]$.

We used the above approach to generate full rating matrices whose dimensions are those of the two real-world datasets, i.e., Flixster (FX) and Movielens (ML), shown in Table 1. For each of these matrices we select the entries that correspond to the actual user-item pairs that are present in the corresponding dataset and give it as input to the matrix completion algorithm. For each dataset we generated five different sets of matrices using different random seeds and we performed a series of experiments using synthetically generated low-rank matrices of rank 5 and 20. For each rank, we report the average of performance metrics in each set from the estimated low-rank models over all the synthetic matrices.

3.1 Results

3.1.1 Effect of item frequency in synthetic datasets. In order to investigate if the number of ratings an item has, i.e., item frequency, has any influence on the accuracy of the matrix completion-based methods for the item, we ordered all the items in decreasing order by their frequency in the rating matrix. Furthermore, we divided these ordered items into ten buckets and for a user computed the RMSE for items in each bucket based on the error between the predicted rating by the estimated low-rank model and the ground-truth rating. We repeated this for all the users and computed the average of the RMSE of the items in each bucket over all the users.

Figure 1 shows the RMSEs across the buckets along with the average frequency of the items in the buckets. As can be seen in the figure, the predicted ratings for the frequent items tend to have lower RMSE in contrast to infrequent items for all the datasets.

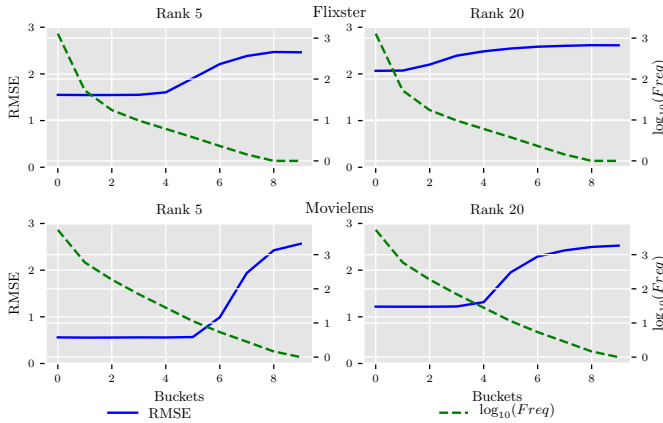


Figure 1: RMSE of the predicted ratings as the frequency of the items decreases.

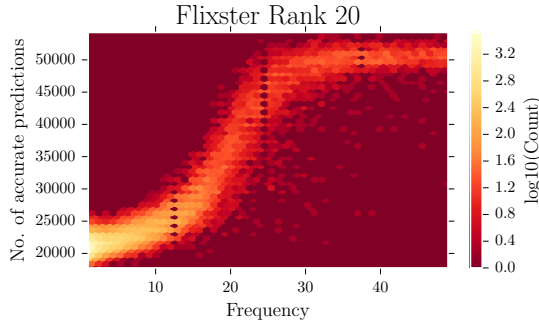


Figure 2: Scatter map of items having different frequency against their number of accurate predictions (Mean absolute error (MAE) ≤ 0.5) for low-rank model with rank 20.

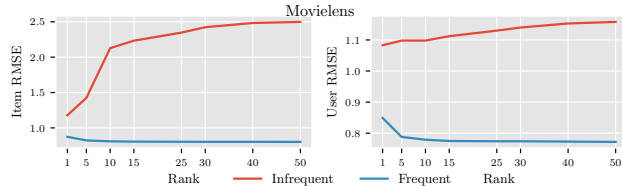


Figure 3: Variation in Test RMSE with increase in rank for the items and the users with different frequency.

Figure 2 shows the scatter map of items in FX having different frequency against the number of instances where the absolute difference between the original and the predicted rating, i.e., *Mean Absolute Error (MAE)*, is ≤ 0.5 . As can be seen in the figure, the number of accurate predictions is significantly lower for items having fewer ratings (≤ 20) compared to that of the items having a large number of ratings (≥ 30). The lower error of the frequent items is because they have sufficient ratings to estimate their latent factors accurately. Hence for the real datasets, items appearing at the top in ordering by frequency and having high predicted scores will form a reliable set of recommendations to a user.

3.1.2 Effect of frequency on accuracy in real datasets. In order to assess the finding that the infrequent items are not estimated accurately by the matrix completion method, we evaluated matrix completion on a random held-out subset of the real datasets. We followed the standard procedure of cross-validation and exhaustive grid search for hyperparameters for model selection. We computed

RMSE over the infrequent items in the test split, i.e., the items that have few ratings in the training split. For the analysis, we ordered the items in increasing order by the number of ratings in training splits. Next, we divided these ordered items into quartiles and identified the items in the first and the last quartile as the infrequent and the frequent items, respectively.

Figures 3 show the RMSE for the items and the users in the test for the Movielens (ML) dataset. We can see that the RMSE of the frequent items (or users) is lower than that of the infrequent items (or users). Furthermore, we observed similar trends in the remaining datasets (results not shown here due to space constraints). These results suggest that the matrix completion method fails to estimate the preferences for the infrequent items (or users) accurately in the real datasets. Also, the RMSE of the infrequent items increases with the increase in the rank while that of frequent items decreases with the increase in the rank. Similarly, the RMSE of the infrequent users increases with the increase in the rank. The increase in RMSE with the increase in ranks suggests that infrequent items or infrequent users may not have sufficient ratings to estimate all the ranks accurately thereby leading to the error in predictions for such users or items. The finding that infrequent items or infrequent users have better accuracy for fewer ranks follows from the result that $O(nr \log(n))$ entries are required to recover the underlying low-rank model of a $n \times n$ matrix of rank r [4], and therefore for fewer entries (e.g., infrequent users or infrequent items) we may recover only fewer ranks of the underlying low-rank model accurately.

4 METHODS

The analysis presented in the previous section showed that as the underlying rank of the low-rank model that describes the data increases, the error associated with estimating such a low-rank model from the skewed data increases for the infrequent users and the infrequent items. We use these observations to devise multiple approaches to improve the accuracy of the low-rank models for such users and items.

4.1 Frequency Adaptive Rating Prediction (FARP)

Since the error of the predictions from the estimated low-rank models increases for the infrequent users or items in skewed data, we propose to estimate lower dimensional latent factors for the infrequent users or items, and estimate higher dimensional latent factors for the frequent users or items. In this approach, we propose to learn multiple low-rank models with different ranks from all the available data and while predicting the rating of a user on an item we select the model that performed the best for the infrequent user or the item associated with the rating. Hence, the predicted rating of user u on item i is given by

$$\hat{r}_{ui} = \mathbf{p}_{uk} \mathbf{q}_{ik}^T, \quad (3)$$

where \mathbf{p}_{uk} and \mathbf{q}_{ik} are the user and the item latent factors from the k th low-rank model. For example, if $f_u < f_i$ then we select the k th low-rank model for prediction such that it has the best performance for users having frequency f_u , and similarly if $f_i < f_u$ then we select the model with the best performance for items with frequency f_i .

The user and items can be assigned to different low rank models based on the number of ratings that exists for them in the dataset. One approach that we investigated is to order the users and the items by the number of ratings and divide them into equal quartiles and save the best performing model for each quartile.

4.2 Truncated Matrix Factorization (TMF)

An alternate approach we develop is to estimate only a subset of the ranks for these users or items. In this approach, the estimated rating for user u on item i is given by

$$\hat{r}_{u,i} = \mathbf{p}_u(\mathbf{q}_i \odot \mathbf{h}_{u,i})^T, \quad (4)$$

where \mathbf{p}_u denotes the latent factor of user u , \mathbf{q}_i represents the latent factor of item i , $\mathbf{h}_{u,i}$ is a vector containing 1s in the beginning followed by 0s, and \odot represents the elementwise Hadamard product between the vectors. The vector $\mathbf{h}_{u,i}$ is used to select the ranks that are *active* for the (u, i) tuple. The 1s in $\mathbf{h}_{u,i}$ denote the active ranks for the (u, i) tuple.

4.2.1 Frequency adaptive truncation. One approach that we investigated for selecting the active ranks, i.e., $\mathbf{h}_{u,i}$, for a user-item rating is based on the frequency of the user and the item in the rating matrix. In this approach, for a given rating by a user on an item, first, we determine the number of ranks to be updated based on either the user or the item depending on the one having a lower number of ratings. In order to select the ranks, we normalize the frequency of the user and the item, and use a non-linear activation function, e.g., sigmoid function, to map this frequency of the user or the item in $[0, 1]$. Finally, we used this mapped value as the number of active ranks selected for the update of the user and the item latent factors. The number of active ranks to be selected is given by

$$k_{u,i} = \frac{r}{1 + e^{-k(f_{\min} - z)}}, \quad (5)$$

where r is the dimension of the user and the item latent factors, $f_{\min} = \min(f_u, f_i)$, f_u is the frequency of user u , f_i is the frequency of item i , k controls the steepness of the sigmoid function and z is the value of the sigmoid's midpoint. The use of such a function assists in identifying the users or the items that can not be estimated accurately using all the ranks and we can only estimate few ranks more accurately for such users or items. The active ranks for a user or an item can be chosen either from the beginning of all ranks or end of all ranks or can be chosen arbitrarily among all ranks until the same active ranks are used consistently for the user and the item. For ease of discussion and simplicity, we will assume that active ranks are chosen from the beginning of all ranks. Hence, the active ranks to be selected are given by

$$\mathbf{h}_{u,i}[j] = \begin{cases} 1, & \text{if } j \leq k_{u,i} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

We will refer to this method as Truncated Matrix Factorization (TMF).

4.2.2 Frequency adaptive probabilistic truncation. An alternative way to select the active ranks is to assume that the number of active ranks follows a Poisson distribution with parameter $k_{u,i}$. This method is similar to Dropout [18] technique in neural networks, where parameters are selected probabilistically for updates during

learning of the model. Similar to regularization it provides a way of preventing overfitting in learning of the model. The active ranks to be selected are given by

$$\mathbf{h}_{u,i}[j] = \begin{cases} 1, & \text{if } j \leq \theta_{u,i} \\ 0, & \text{otherwise,} \end{cases}$$

where $\theta_{u,i} \sim \text{Poisson}(k_{u,i})$. We will call this method as Truncated Matrix Factorization with Dropout (TMF + Dropout). Similar to Equation 2, the parameters of the model, i.e., the user and the item latent factors can be estimated by minimizing a regularized square loss between the actual and the predicted ratings.

4.2.3 Rating prediction. After learning the model the predicted rating for user u on item i for TMF model is given by

$$\hat{r}_{u,i} = \mathbf{p}_u(\mathbf{q}_i \odot \mathbf{h}_{u,i})^T, \quad (7)$$

where the active ranks, i.e., $\mathbf{h}_{u,i}$, is given by Equation 6. The predicted rating for the user and the item under TMF + Dropout model is given by the least number of ranks for whom the cumulative distribution function (CDF) for Poisson distribution with parameter k_{ui} obtains approximately the value of 1. The active ranks, i.e., $\mathbf{h}_{u,i}$, for prediction under TMF + Dropout are given by

$$\mathbf{h}_{u,i}[j] = \begin{cases} 1, & \text{if } j \leq s \\ 0, & \text{otherwise,} \end{cases}$$

where s is the least number of ranks for whom the CDF, i.e. $P(x \leq s) \approx 1$ and $x \sim \text{Poisson}(k_{u,i})$.

Unlike FARP, which requires us to estimate multiple models, TMF estimates a single model however it involves tuning of more hyperparameters in comparison to FARP.

4.3 Inverse Frequency Weighted Matrix Factorization (IFWMF)

In addition to the above approaches, we explored a weighted matrix factorization-based approach where we weigh the reconstruction error higher for the infrequent users and items. We propose to estimate the user and the item latent factors by minimizing a regularized weighted square error loss between the actual and the predicted ratings

$$\underset{\mathbf{p}_u, \mathbf{q}_i}{\text{minimize}} \quad \frac{1}{2} \sum_{r_{ui} \in R} w_{ui} (r_{ui} - \mathbf{p}_u \mathbf{q}_i^T)^2 + \frac{\beta}{2} (\|\mathbf{p}_u\|_2^2 + \|\mathbf{q}_i\|_2^2), \quad (8)$$

where the the weight w_{ui} is given by

$$w_{ui} = \frac{1}{1.0 + \rho f_{\min}}, \quad (9)$$

where ρ is a constant, $f_{\min} = \min(f_u, f_i)$, f_u and f_i are the normalized frequency of user u and item i , respectively. Essentially, we weigh the error in predictions more for the infrequent users and the infrequent items. This resembles the weighted matrix factorization [7, 8] where the weight of the reconstruction error is proportional to the confidence on the observed rating of user u on item i however in our method we weigh the error inversely proportional to the frequency of ratings observed for user u or item i . This is similar to the inverse propensity model-based approach [17], where the propensity is proportional to the frequency of the user

or the item. The up-weighting of the reconstruction error associated with the infrequent users or the infrequent items may lead to over-fitting as we only have few ratings for these users and items.

5 EXPERIMENTAL EVALUATION

5.1 Comparison algorithms

We compared our proposed approaches against the the state-of-the-art Matrix Factorization [9] and LLORMA [11] method. LLORMA assumes that the different parts of the user-item rating matrix can be approximated by different low-rank models and the complete user-item rating matrix is approximated as a weighted sum of these individual low-rank models. We have used the LibRec [5] software package to compare the proposed methods against the LLORMA approach.

5.2 Model selection

We performed grid search to tune the dimensions of the latent factors, regularization hyper-parameters, constant (ρ), and sigmoid function's parameters, i.e., k and z . We searched for regularization weights (λ) in the range [0.001, 0.01, 0.1, 1, 10], dimension of latent factors (r) in the range [1, 5, 10, 15, 25, 50, 75, 100], constant (ρ) in the range [1, 10, 50], steepness constant (k) in the range [1, 5, 10, 20, 40], and mid-point (z) in the range [-0.75, -0.50, -0.25, 0, 0.25, 0.50, 0.75]. The final parameters were selected based on the performance on the validation split. For LLORMA, we varied the number of local models (l_m) in the range [1, 5, 8, 15, 25, 50].

For FARP, we ordered the users in ascending order by frequency and divided them into equal quartiles. For each quartile, we saved the best performing model, i.e., the model having the lowest RMSE for all the users in that quartile in the validation split. Similarly, we ordered the items in ascending order by frequency and divided them into equal quartiles. Similar to users, we saved the best performing model for each quartile of items. At the time of prediction of rating for a user on an item, we choose the the model associated with the quartile of the user if the user is having lower number of ratings than the item, and if the item is having lower number of ratings than the user than we choose the model associated with the quartile of the item .

5.3 Datasets

In addition to Flixster (FX) [22] and Movielens 20M (ML) [6] datasets, we evaluated our proposed methods on subsets of the Yahoo Music (YM) [1, 16] and Netflix (NF) [2] datasets that we created in order to have a skewed distribution. These datasets were generated as follow. First, for each user we randomly selected the number of ratings that we want to sample from the user's ratings and randomly sampled these ratings for all users. Next, from the sampled ratings in previous step, for each item we randomly selected the number of ratings that an item has received from users in sampled ratings and randomly sampled these ratings for all the items. After following the above two steps, the sampled ratings from these datasets follows a skewed distribution and characteristics of all the datasets used in experiments are presented in Table 1.

5.4 Evaluation methodology

To evaluate the performance of the proposed methods we divided the available ratings in different datasets into training, validation and test splits by randomly selecting 20% of the ratings for each of the validation and the test splits. The validation split was used for model selection, and the model that was selected was used to predict ratings on the test split. We repeated this process three times and report the average RMSE across the runs.

In addition to computing RMSE obtained by different methods for the ratings in the test split, we also investigated the performance of the proposed approaches for the items and the users with a different number of ratings in the training split. To this end, we ordered the items and the users in increasing order by their number of ratings in training split and divided them equally into quartiles. We will report the RMSE achieved by different methods for ratings in the test split for the users and the items in these quartiles.

6 RESULTS AND DISCUSSION

6.1 Performance for rating prediction on entire dataset

Table 2 shows the results achieved by the proposed methods on the various datasets. As can be seen in the table for the task of rating predictions for all the ratings in the test splits the proposed approaches perform better than the MF method for FX, ML, YM and NF datasets. Interestingly, the proposed approaches have performed even better than the state-of-the-art LLORMA method and this suggests that LLORMA can be further improved by estimating local low-rank models that considers the skewed distribution of ratings in datasets. We found the difference between the predictions of different methods to be statistically significant (p -value ≤ 0.01 using two sample t -test). The performance is significantly better for FX in comparison to that of other datasets. Additionally, on FX dataset, the MF method outperforms the LLORMA method and a possible reason for this is that because of the skewed distribution LLORMA is not able to estimate a model that is as accurate as a global MF model. Moreover, LLORMA needs a significantly large number of local low-rank models in comparison to the proposed approaches. Also, by comparing the number of latent dimensions used by the models shown in Table 2 we can see that, for ML, TMF + Dropout and LLORMA needs significantly fewer ranks, i.e., 25, in comparison to that of MF, i.e., 100, to achieve the same performance and we believe that this could be because of MF overfitting the ML dataset for higher dimension of latent factors.

6.2 Performance for the users and the items with different number of ratings

Table 2 also shows the performance achieved by the different methods across the different quartiles of users and items. By comparing the performance of the different schemes we can see that the proposed methods significantly outperform the MF method and state-of-the-art LLORMA method for lower quartiles for majority of the datasets. This illustrates the effectiveness of the developed methods for the users and the items with few ratings. The better performance of the proposed methods for the users and the items with few ratings is because we can estimate accurately only a few ranks

Table 2: Test RMSE of the proposed approaches for different datasets. The RMSE for the users and the items in different quartiles order by their frequency. Q1 refers to the quartile containing the least frequent users or items followed by remaining in Q2, Q3, and Q4. Table 4 shows the average number of test ratings in different quartiles for different datasets.

	Flixster (FX)						Movielens (ML)					
	MF	LLORMA	IFWMF	FARP	TMF	TMF + Dropout	MF	LLORMA	IFWMF	FARP	TMF	TMF + Dropout
No. of low-rank models	1	20	1	8	1	1	1	8	1	8	1	1
Rank	15	50	15	NA [†]	10	15	100	25	100	NA [†]	100	25
All test ratings	0.864	0.871	0.867	0.863	0.851	<u>0.847</u>	0.806	0.806	0.804	<u>0.797</u>	0.804	0.804
Item Q1	1.302	1.705	1.289	1.258	<u>1.252</u>	<u>1.256</u>	2.527	1.501	2.382	<u>1.178</u>	2.377	2.115
Item Q2	0.961	1.099	0.961	0.962	<u>0.944</u>	<u>0.944</u>	1.619	0.974	1.449	<u>0.937</u>	1.499	1.123
Item Q3	0.800	0.863	0.801	0.798	0.785	<u>0.780</u>	0.891	0.841	0.869	<u>0.855</u>	0.866	0.851
Item Q4	0.864	0.865	0.867	0.863	0.851	<u>0.847</u>	0.799	0.805	0.799	<u>0.795</u>	0.798	0.801
User Q1	1.292	1.388	1.261	<u>1.246</u>	1.247	1.260	1.174	1.092	1.120	1.083	1.115	<u>1.078</u>
User Q2	1.177	1.255	1.156	<u>1.143</u>	1.144	1.151	0.975	<u>0.964</u>	0.965	0.965	<u>0.964</u>	0.968
User Q3	0.974	1.002	0.969	0.967	<u>0.964</u>	<u>0.964</u>	0.853	0.856	<u>0.852</u>	0.853	<u>0.852</u>	0.863
User Q4	0.853	0.852	0.857	0.853	0.841	<u>0.836</u>	0.767	0.774	0.769	<u>0.761</u>	0.769	0.769

	Yahoo Music (YM) [*]						Netflix (NF) [*]					
	MF	LLORMA	IFWMF	FARP	TMF	TMF + Dropout	MF	LLORMA	IFWMF	FARP	TMF	TMF + Dropout
No. of low-rank models	1	25	1	8	1	1	1	25	1	8	1	1
Rank	100	5	75	NA [†]	75	75	75	25	75	NA [†]	75	75
All test ratings	1.170	1.177	1.162	<u>1.152</u>	1.163	1.164	0.906	0.903	0.901	<u>0.901</u>	0.903	0.904
Item Q1	1.245	1.250	1.238	1.235	1.239	<u>1.234</u>	1.067	1.059	1.054	1.047	1.051	1.060
Item Q2	1.165	1.171	1.159	<u>1.152</u>	1.161	1.162	1.018	1.006	1.007	<u>1.005</u>	1.008	1.014
Item Q3	1.154	1.164	1.149	<u>1.139</u>	1.151	1.152	0.967	0.964	0.956	<u>0.955</u>	0.959	0.962
Item Q4	1.170	1.172	1.161	<u>1.150</u>	1.162	1.163	0.896	0.893	<u>0.891</u>	0.892	0.893	0.894
User Q2	1.429	1.452	1.412	1.408	1.407	<u>1.395</u>	1.281	1.378	<u>1.258</u>	1.274	1.259	1.256
User Q3	1.236	1.231	1.220	<u>1.210</u>	1.224	1.225	1.018	1.011	<u>1.008</u>	1.006	1.010	1.013
User Q4	1.155	1.160	1.148	<u>1.138</u>	1.150	1.15	0.879	<u>0.874</u>	0.875	0.875	0.877	0.877

[†] The ranks used for FARP are in Table 3.

^{*} Due to sampling, the test splits are not having any ratings for users in Q1 for these datasets.

Table 3: Rank used by FARP for different datasets.

	Item				User			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
FX	1	1	15	15	1	1	1	15
ML	1	5	10	100	1	15	30	100
YM	75	75	100	100	5	5	100	100
NF	50	50	50	75	50	50	50	75

Table 4: Average number of ratings in quartiles of test splits.

	Item				User			
	Q1 (10 ³)	Q2 (10 ³)	Q3 (10 ³)	Q4 (10 ⁶)	Q1 (10 ³)	Q2 (10 ³)	Q3 (10 ³)	Q4 (10 ⁶)
FX	4.8	6.3	22.0	1.4	16.5	15.9	54.7	1.4
ML	2.7	9.5	92.5	3.7	111.2	194.8	533.5	2.9
YM	83.4	175.4	303.2	1.2	NA [†]	38.9	181.4	1.6
NF	13.6	40.1	136.0	1.5	NA [†]	32.0	209.8	1.4

[†] Due to sampling, the test splits do not have ratings for users in Q1.

for them, and unlike MF and LLORMA the proposed approaches are effective in model estimation or predicting the ratings for these users and items.

Among the proposed approaches, the TMF-based approaches (TMF and TMF + Dropout), perform better in most of the quartiles in FX dataset. Surprisingly, FARP consistently performs better than the MF and LLORMA across most of the datasets, and this is promising as compared to TMF-based approaches FARP has fewer hyperparameters to tune. Specifically, in TMF-based approaches we have to tune regularization weights (λ), dimension of latent factors (r), steepness constant (k) and mid-point (z), while in FARP we have to tune only three parameters, i.e., number of low-rank

models, regularization weights (λ) and dimension of latent factors (r). This might be of interest to practitioners because multiple low-rank models under FARP can be estimated in parallel using vanilla MF widely available in off-the-shelf packages, e.g., SPARK [23] and scikit-learn [14].

While the IFWMF method performs better than MF in lower quartiles, the other proposed approaches, i.e., FARP and TMF-based methods, outperform IFWMF for most of the quartiles in all the dataset thereby illustrating the effectiveness of FARP and TMF-based methods in preventing over-fitting and generating better predictions.

7 CONCLUSION

In this work, we have investigated the performance of the matrix completion-based low-rank models for estimating the missing ratings in real datasets and its impact on the item recommendations. We showed in Section 3 that the matrix completion-based methods because of skewed distribution of ratings fail to predict the missing entries accurately in the matrices thereby leading to an error in predictions and thus affecting item recommendations. Based on these insights we presented different methods in Section 4, which considers the frequency of both the user and the item to estimate the low-rank model or for predicting the ratings. The experiments on real datasets show that the proposed approaches significantly outperforms the state-of-the-art Matrix Factorization method for rating predictions for the users and the items having few ratings in the user-item rating matrix.

REFERENCES

- [1] 2008. Yahoo! Music User Ratings of Songs with Artist, Album, and Genre Meta Information, v. 1.0. <https://webscope.sandbox.yahoo.com/>
- [2] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, Vol. 2007. New York, NY, USA, 35.
- [3] Emmanuel J. Candès and Benjamin Recht. 2009. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics* 9, 6 (2009), 717–772. <https://doi.org/10.1007/s10208-009-9045-5>
- [4] Emmanuel J. Candès and Terence Tao. 2010. The Power of Convex Relaxation: Near-optimal Matrix Completion. *IEEE Trans. Inf. Theor.* 56, 5 (May 2010), 2053–2080. <https://doi.org/10.1109/TIT.2010.2044061>
- [5] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. LibRec: A Java Library for Recommender Systems.. In *UMAP Workshops*, Vol. 4.
- [6] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (itiis)* 5, 4 (2016), 19.
- [7] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining. Ieee*, 263–272.
- [8] Noam Koenigstein and Yehuda Koren. 2013. Towards scalable and accurate item-oriented recommendations. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 419–422.
- [9] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.
- [10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug. 2009), 30–37. <https://doi.org/10.1109/MC.2009.263>
- [11] Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2013. Local low-rank matrix approximation. In *International Conference on Machine Learning*. 82–90.
- [12] Roderick JA Little and Donald B Rubin. 2014. *Statistical analysis with missing data*. Vol. 333. John Wiley & Sons.
- [13] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*. ACM, 5–12.
- [14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
- [15] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*. ACM, New York, NY, USA, 11.
- [16] Sebastian Schelter, Stephan Ewen, Kostas Tzoumas, and Volker Markl. 2013. All roads lead to rome: optimistic recovery for distributed iterative data processing. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, 1919–1928.
- [17] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. *arXiv preprint arXiv:1602.05352* (2016).
- [18] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
- [19] Harald Steck. 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 (KDD '10)*. ACM, New York, NY, USA, 713–722. <https://doi.org/10.1145/1835804.1835895>
- [20] Harald Steck. 2013. Evaluation of Recommendations: Rating-prediction and Ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems (RecSys '13)*. ACM, New York, NY, USA, 213–220. <https://doi.org/10.1145/2507157.2507160>
- [21] Doris Xin, Nicolas Mayoraz, Hubert Pham, Karthik Lakshmanan, and John R Anderson. 2017. Folding: Why Good Models Sometimes Make Spurious Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 201–209.
- [22] R. Zafarani and H. Liu. 2009. Flixster dataset, Social Computing Data Repository at ASU. <http://socialcomputing.asu.edu/datasets/Flixster>
- [23] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Thagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. 2016. Apache Spark: A Unified Engine for Big Data Processing. *Commun. ACM* 59, 11 (Oct. 2016), 56–65. <https://doi.org/10.1145/2934664>