Latent Feature Modelling for Recommender Systems

Abdullah Alhejaili and Shaheen Fatima
Department of Computer Science
Loughborough University
Loughborough, United Kingdom
{A.E.S.Alhejaili, S.S.Fatima}@lboro.ac.uk

Abstract—Matrix factorization is one of the most successful model-based collaborative filtering approaches in recommender systems. Nevertheless, useful latent user features can lead to a more accurate recommendation. However, user privacy and cross-domains access restrictions challenge collection and analysis of such information. In this study, we propose a feature extraction method (WAFE) which leverages user-item interaction history to extract useful latent user features. We also propose a rating prediction approach that incorporates the local mean of users’ and items’ ratings. We evaluate our proposed model using two real-world benchmark datasets and compare its performance against the state-of-the-art matrix factorization collaborative filtering methods. Evaluation results show that proposed method outperforms the existing methods.

Index Terms—Matrix Factorization, Collaborative Filtering, Recommender System

I. INTRODUCTION

Recommender systems (RS) assign active users with useful suggestions on relevant items [1], where “item” refers to what the system provides, such as books, movies, songs and article papers. In RS, consumers receive several suggestions for related items offered to them based on some information that the system holds about items and users. This information might be specified by users about themselves, such as age and gender (users’ demographics) or derived from the item’s details, such as name, type and date of release (items’ content). Other information can be retrieved from the history of user-item interactions (users’ feedback on items) describing their behaviours. Users’ feedback can be implicit (such as clicks, adding to the cart, purchasing history), or explicit (such as ratings and reviews).

In RS, three main approaches are adopted, namely collaborative filtering (CF), content-based (CB) and hybrid. Collaborative filtering [2] is an RS approach in which users’ feedback on items is used to extract any similarities between different users. The similarities between different users are extracted based on their shared liked or disliked products. Content-based filtering [3] is another approach for designing RSs. In this approach, items’ contents are used to extract the similarities between items. Thus, a recommendation is made based on these similarities. Hybrid RSs [4] combine CF with CB to enhance their advantages and overcome their limitations.

CF can be categorised into memory-based and model-based. In the former, which is also called neighbourhood-based, neighbourhoods are defined for each user/item, in which the rating of the active user on an unrated item can be predicted based on the nearest neighbours’ ratings on that item [5]. This can be either item-based or user-based nearest neighbour model. In model-based approach, the users’ rating behaviours are analysed in order to learn an optimal prediction model given some features of users and items [5]. This is also called latent factors method [6], in which matrix factorization methods are one of the most successful approaches.

CF methodology has achieved a large acceptance in recommender systems’ field, thanks to its high accuracy. However, CF requires a reasonable number of user-item interactions (feedback assigned by users to items) to be able to make an accurate recommendation [7]. This leads to developing CF models that incorporate additional information along with the item-user rating history.

Matrix factorization (MF) is one of the most successful methods used for model-based CF in the literature [6]. In MF models, users and items are mapped to joint latent factors that can be learnt using user-item rating history. Nevertheless, more information can be involved in learning more useful latent features which can lead to achieving a more accurate recommendation. This additional information, mostly, is retrieved from external domains, such as social networks [8], [9]. However, when regarding user privacy and cross-domains access restrictions, there are unavoidable challenges in collecting and analysing such information [2], [10].

The challenges mentioned above have raised the need for investigating the possibility of improving the accuracy of CF by modelling useful users’ and items’ latent features without relying on external data. In this study, we aim to investigate the usefulness of modelling users and items latent features by analysing their rating behaviours. We propose a features extraction method, namely, Weighted Average based Feature Extraction (WAFE) that analyse user-item interaction history for generating users’ features. Then, we introduce a matrix factorisation collaborative filtering method that adopts WAFE to initialise the initial values of users’ and items’ latent features. We also propose a rating prediction approach that incorporates the local mean of users’ and items’ ratings. The
main contributions of this paper are as follows:

- We propose a method (WAFE) which leverages user-item latent features. Singular Value Decomposition (SVD) is a well-known technique for identifying latent factors [6], [11], which requires factorizing the rating matrix. In a recommender system, dealing with a sparse matrix $R$ is a major difficulty, as the number of unrated items by each user is high. It has been proposed in some studies [12], [13] that these missing values can be filled in with user/item rating mean. However, assigning all missing values based on the rating average may result in an inaccurate recommendation and high computational cost [14]. Besides, matrix decomposition can be performed using only known entries of $R$, by learning the optimal users and items latent factors that model these known entries. However, this cannot be applied to the rest of the missing entries, as it will result in overfitting. Hence, it has been suggested by some authors such as [15]–[17] to avoid overfitting through a regularised model such as:

$$\min_{p^u, q_i} \sum_{u \in \mathcal{U}} (r_{ui} - p^u_i q_i)^2 + \lambda \left( \|p_u\|^2 + \|q_i\|^2 \right)$$

where, $\kappa$ is the set of the $(u, i)$ pairs for all known $r_{ui}$ and $\lambda$ is a constant that controls the extent of the regularisation. The regularised squared error can be minimised using an optimiser such as gradient descent.

In practice, the number of items rated by each user is different, and so the number of times items have been rated. To deal with this variance, data can be adjusted by adding biases [17], as in equation 3:

$$b_{ui} = \mu + b_u + b_i$$

where, $\mu$ is the overall global average rating, $b_u$ and $b_i$ are the bias of user $u$ and item $i$ respectively. Hence, the rating estimation in equation 1 and the squared error function in equation 2 are extended by adding the biases as in equation 4 and 5 respectively.

$$\hat{r}_{ui} = \mu + b_u + b_i + p^u_i q_i$$

$$\min_{p^u, q_i} \sum_{u \in \mathcal{U}} (r_{ui} - \mu - b_u - b_i - p^u_i q_i)^2 + \lambda \left( \|p_u\|^2 + \|q_i\|^2 + b_u^2 + b_i^2 \right)$$

The regularised squared error function in equation 5 can be minimised using gradient descent so that the model parameters can be determined by looping over all known ratings and update them as follows:

$$b_u \leftarrow b_u + \gamma \cdot (e_{ui} - \lambda \cdot b_u)$$

$$b_i \leftarrow b_i + \gamma \cdot (e_{ui} - \lambda \cdot b_i)$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} - q_i - \lambda \cdot p_u)$$

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i)$$

where, $\gamma$ is a constant of the step size (learning rate), and $e_{ui}$ is the prediction error, which can be computed using equation 10.

$$e_{ui} = r_{ui} - \hat{r}_{ui}$$

SVD++ is proposed by [17] as an extension of SVD. SVD++ incorporates implicit feedback of users and items to enhance the rating prediction accuracy.

Authors in [18], presented a collaborative filtering model called Probabilistic Matrix Factorization (PMF) to cover the issue of learning from an imbalanced and sparse recommendation dataset (e.g., recommending items to users with a minimal rating history). The model shows an improvement of the recommendations for these types of users.

In [19], Wang and Blei proposed a collaborative topic regression (CTR) model, which combines traditional collaborative filtering with topic modelling using latent Dirichlet allocation (LDA) proposed by [20]. The model relies on the historical interaction data of users as well as the content of items to perform the recommendation.

The authors in [21] proposed a Collaborative Deep Learning model (CDL), which enhances the advantages of ratings collaborative filtering and content information learning using Stacked Denoising AutoEncoder (SDAE) [22] for learning input data representation.

Kim et al. in [23] proposed a context-aware recommendation model called convolutional matrix factorization (ConMF), which integrates a convolutional neural network and PMF. The main advantage of this model is using textual

The remaining of this paper is organised as follow. Section II, presents the related work and section III presents our proposed methods. We evaluate the methods in section IV and discuss the results in section V. The work is concluded in section VI.
information in the description of items to enhance the recommendation accuracy. The limitation of this approach is the adoption of information that is not available in all domains, for instance, MovieLens dataset.

As an extension of [23] work, a robust document context-aware hybrid method has been proposed by Kim et al. in [24]. The authors introduced a latent factor modelling method for items by adopting the items’ ratings alongside their description documents, taking into consideration the case of Gaussian noise, which occurs when users have different interaction rates. This can offer a better understanding of user-item relationships. However, the computational cost of collection and analyses the adopted description documents may be high. In our model, we do not consider the description documents; thus, it is not limited by a risk of high computational costs.

In [25], Nguyen proposed a model-based book recommender system employing user side information and applying some feature selection techniques to enhance the accuracy of the RS. The recommendation is considered as a binary classification task; ratings in the range of 6 to 10 are recommended and ratings less than 6 are not.

A hybrid collaborative filtering RS model is proposed by [26]. Their model uses an autoencoder to extract latent features from side information. The model uses matrix factorization for implicit feedback prediction. The model assumes that implicit feedback is given if the rating is six or higher.

In summary, many studies in the literature rely on retrieving more information from external domains such as textual description of items, or users’ personal information to model users and items latent features. At the same time, some other studies incorporate users’ implicit feedback or users’ rating probabilities along with explicit feedback to enhance the rating prediction accuracy. However, these studies are limited to i) requiring cross-domain access, ii) not considering users’ privacy, or iii) not incorporating users and items meaningful features values for modelling users and items latent features. In our research, we integrate useful user features without relying on retrieving information from an external domain or users’ personal information. Moreover, we incorporate the local mean of users and items rating to enhance rating prediction accuracy. To the best of our knowledge, this is the first study that combines i) modelling users and items latent features with meaningful values and ii) incorporating the local mean of users and items rating to enhance the rating prediction accuracy in model-based collaborative filtering recommender systems.

III. PROPOSED METHOD

Items and users features are incorporated to improve the performance of collaborative filtering methods. This incorporation enhances rating prediction accuracy. In standard SVD, the users and items latent features are randomly initialised, in which the optimal values are achieved using learning algorithm such as gradient descent. In our proposed method, instead of random initialisation for users and items latent features, we use meaningful values of users and items features and assign them as latent features. Besides, we incorporate the users and items local rating mean for formalising the bias along with the global (overall) rating mean. To this end, we first, introduce a novel method (WAFE) for extracting users’ features from items features and user-items interaction history. Then, WAFE-SVD is proposed, in which WAFE is used to extend SVD to incorporate the users and items features as latent features. Moreover, in order to incorporate the local rating average of users and items, we propose SVD with the local mean (SVD-LM) to enhance the prediction accuracy. Finally, WAFE-SVD and SVD-LM are combined into WAFE-SVD-LM. This further increases the prediction accuracy.

A. Weighted average based feature extraction method (WAFE)

Users and items latent factors are defined to represent the joint intersection between items and users. For example, in a movie recommendation dataset, one latent factor could be action, in which both user and item can be represented by this factor, i.e., to what extent this movie is belonging to action movies, and how much the user likes action movies. Based on the same concept, WAFE works on, first, defining the joint factors by extracting features of items, and then, link the users to these features by extracting their interests of these factors. Thus, the latent factors here can be represented by the defined items features. In the following text, we explain the mechanism of extracting items features and users features.

Items’ features can be extracted manually from the domain knowledge. Recall the example of a movies recommender system, information about movies, such as genre and year of release can be derived and represented in different ways, such as continues variable and categorical variables [27]. The former is used for features with values that have a meaningful scale and can be applied to mathematical operations such as averaging. At the same time, categorical variables represent features with values that assign the item to a particular category, such as the genre. It is also possible for some features to be transformed into another type of representation [27], [28]. In this study, we represent all items features as categorical variables using dummy variable technique. For example, in a movies recommender system dataset, assuming that there three possible genres of movies, i.e., action, comedy and drama, thus, if a film belongs to action category, its action feature is assigned to 1, and 0 otherwise, and so on for all other categories.

Following this methodology for all items features, each feature’s unique value represents a joint factor. Thus, the number of joint factors equals to the number of item’s features’ unique values. The following section is to explain how users can be linked to these joint factors.

1) Users features: The goal is to link each user with each joint factors, which are, initially, the items’ features categories, representing the users’ interest in these factors. This interest is calculated based on the users’ rating behaviour for all items that belong to each factor. More precisely, for each factor, we calculate the weighted average of the users rating for all items that belong to that factor. This is formally calculated as follow: Let:

\[
\text{Let:}
\]

...
• $U$: is the set of all users.
• $I$: is the set of all items.
• $R$: is the set of all ratings assigned by all users to all items.
• $I_u$: is the set items rated by user $u$.
• $R_u$: is the set of all ratings assigned by user $u$ to items
• $f_j$: is the $j^{th}$ item feature.
• $v_j$: is the $j^{th}$ joint factor.
• $R_u^j$: is the set of ratings assigned by user $u$ to all items

For each joint factor $v_j$, user $u$ is linked to that factor by $v_j^u$, in which the value of $v_j^u$ can be calculated using equation 11 as follow:

$$v_j^u = \left( \frac{1}{|R_u^j|} \sum_{r \in R_u^j} r \right) \cdot w_u$$

where, $w_u$ is the weight of the users rating, which can be calculated using equation 12 as follow:

$$w_u = \frac{|R_u|}{|R|}$$

B. WAFE-based SVD (WAFE-SVD)

Typically, SVD based collaborative filtering methods aim to minimise the rating prediction error by solving an optimisation problem (equation 5). A well-known technique of addressing this problem is to start with initialising users’ latent features $P$, and items’ latent features $Q$ with random values. Then, these initial values are updated in order to minimise the prediction error using a learning algorithm such as gradient descent, as explained in section II. In practice, this technique has achieved a reasonable prediction accuracy, thanks to the regularisation parameters and biases. However, the initialisation of $P$ and $Q$ plays a crucial role in increasing the prediction accuracy and reducing the computational time. Thus, we extend SVD to initialise users and items latent features with useful values. The proposed WAFE-SVD model utilises WAFE method to initialise the initial values of $P$ and $Q$. As explained in section III-A, WAFE produces users’ and items’ features that are linked via joint factors. In WAFE-SVD, the initial values of $P$ and $Q$ are assigned to these features. This initialisation reduces the initial prediction error, which will enhance the performance of the model.

C. SVD with local mean (SVD-LM)

SVD adopts the global mean of ratings in the estimated baseline, as shown in equation 3. The incorporation of users rating behaviours and items reputation can enhance the rating prediction accuracy. Users’ behaviours and item reputation are represented by their local rating averages. Thus, in SVD-LM, the local means of users and items ratings are involved in the rating estimation process. We extend the estimation formula in equation 4 as follow:

$$\hat{r}_{ui} = \frac{1}{3} \left( (\mu + b_u + b_i + p_u^T q_i) + \mu_u + \mu_i \right)$$

where $\mu_u$ and $\mu_i$ are the local rating average of user $u$ and item $i$ respectively. This is also to improve the accuracy of the rating prediction and reduce the number of the required iteration times for achieving the best accuracy of the model.

D. WAFE-based SVD with local mean (WAFE-SVD-LM)

The main advantage of WAFE-SVD is to initialise users and items latent features with meaningful values that have been extracted using WAFE, which utilises user-item rating behaviour. Besides, SVD-LM is advantaged by incorporating the local mean of users’ and items’ ratings. In WAFE-SVD-LM, we leverage the advantages of WAFE-SVD and SVD-LM to enhances the CF model by combining them into one method, as illustrated in algorithm 1.

Algorithm 1 WAFE-SVD-LM

Input: rating matrix $R$, items’ features, $\gamma$, $\lambda$ and epochs

Output: The predicted rating matrix $\hat{R}$

1: Extract users’ features using WAFE
2: $P \leftarrow$ users’ features
3: $Q \leftarrow$ items’ features
4: initialise $B_u, B_i$
5: $\mu \leftarrow$ overall rating average
6: $Mu \leftarrow$ users’ rating average
7: $Mi \leftarrow$ items’ rating average
8: for all known $r_{u,i}$ do
9: calculate $\hat{r}_{u,i}$ using equation 13
10: end for
11: $minRMSE \leftarrow$ RMSE, using equations 14
12: for epochs do
13: for all known $r_{u,i}$ do
14: calculate $\hat{r}_{u,i}$ using equation 13
15: calculate $e_{u,i}$ using equation 10
16: update $p_u, q_i, b_u$ and $b_i$ using equations (6-9)
17: end for
18: $thisRMSE \leftarrow$ RMSE, using equations 14
19: if $thisRMSE < minRMSE$ then
20: $minRMSE = thisRMSE$
21: $Q_{opt}, P_{opt}, B_{uopt}, B_{iopt} \leftarrow P, Q, B_u, B_i$
22: end if
23: end for
24: calculate $\hat{R}$ for $Q_{opt}, P_{opt}, B_{uopt}, B_{iopt}$ using equation 13

WAFE-SVD-LM starts with utilising WAFE for extracting users features and assign $P$ and $Q$ to users and items features, while users’ and items biases ($B_u, B_i$) are initialised with random values. Then, the local mean of users ($Mu$) and items ($Mi$) ratings are calculated along with the global (overall) rating average ($mu$). Using equation 13, $Mu$ and $Mi$ are incorporated to predict the rating. Then, $P, Q, B_u$ and $B_i$ are optimised using gradient descent. Finally, the optimal $P, Q, B_u$ and $B_i$ are found to be used to generate the predicted rating matrix $\hat{R}$.
IV. EVALUATION

The effectiveness of our proposed methods, i.e., SVD-LM, WAFE-SVD and WAFE-SVD-LM, have been examined in two steps. First, we examined the effectiveness of using WAFE and LM to the standard biased SVD model.

Second, we compare our methods, SVD-LM, WAFE-SVD and WAFE-SVD-LM with two baseline methods, i.e. SVD++ [17], an SVD extension method that incorporates implicit feedback of users and items, and PMF [18], probabilistic matrix factorization method. The performance comparison is performed in terms of the prediction accuracy achieved by each method taking into consideration the number of epochs (iterations) needed to achieve the best accuracy.

A. Datasets

In this experiment, two stable benchmark datasets, MovieLens-1M (ML-1M) and MovieLens-10M (ML-10M) have been used to examine the effectiveness of our proposed approach. MovieLens datasets have been widely used to evaluate recommender systems. ML-1M contains 6040 users, 3952 movies and more than 1 million ratings. Side information is provided in this dataset for users (e.g., gender, age) and movies (e.g. genre, year). ML-10M contains 72,000 users, 10,000 movies and 10 million ratings. Side information about movies are provided in ML-10M; however, users’ side information is not. In both datasets, each user has rated at least 20 movies, and each movie has been rated at least once. Both datasets have been randomly split into 80% for training and 20% for testing.

B. Evaluation metrics

Two evaluation metrics have been used to examine the accuracy of the four compared methods, i.e., root mean squared error (RMSE) and mean absolute error (MAE), which are calculated using equation 14 and equation 15.

\[
RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (\hat{r}_t - r_t)^2} \tag{14}
\]

\[
MAE = \frac{1}{T} \sum_{t=1}^{T} |\hat{r}_t - r_t| \tag{15}
\]

where \( T \) is the number of examples in the test set, \( \hat{r}_t \) is the predicted rating of the \( t^{th} \) test example, and \( r_t \) is the actual rating of the \( t^{th} \) test example.

C. Parameters setting

The parameters of this experiment include \( d \)-the number of latent factors, \( \lambda \) – the regularisation extent, \( \gamma \) – the step size (learning rate) and the number of epochs (iteration).

For WAFE-SVD and WAFE-SVD-LM, since users and items features are generated using WAFE, \( d \) is assigned to the number of the extracted features. In our experiment, 46 factors have been defined for ML-1M, and 47 factors have been defined for ML-10M dataset. On the other hand, for the methods SVD, SVD++, PMF and SVD-LM, we have tried the values \((10, 20, 30, 40, 50)\) for \( d \) and set each method to its optimal value. For PMF, we tried the values of \((0.5, 0.6, 0.7, 0.8, 0.9, 1)\) for \( \gamma \), \( \lambda_1 \) and \( \lambda_2 \), and found that the optimal values are \( \gamma = 1 \), \( \lambda_1 = 0.1 \) and \( \lambda_2 = 0.9 \). For all other methods, we have tried different values for the remaining parameters as follow. \( \lambda = (0.01, 0.02, 0.03, 0.04, 0.05) \) and \( \gamma = (0.005, 0.006, 0.007, 0.008, 0.009, 0.01) \), and assigned them to their optimal values as follows: for SVD++, we assigned \( \lambda_1 = 0.02, \lambda_2 = 0.02 \) and \( \gamma = 0.007 \); for the other methods, we assigned \( \lambda = 0.02 \) and \( \gamma = 0.005 \). The number of epochs is assigned to 100 with early stopping; we picked the best RMSE and MAE when achieved in earlier iteration.

V. RESULTS AND DISCUSSION

A. The effectiveness of WAFE and LM

The effectiveness of i) using WAFE to initialise the values of users and items latent features, and ii) incorporating the local mean (LM) of users and items rating for rating prediction have been examined by running the experiment for our three proposals, i.e., SVD-LM, WAFE-SVD and WAFE-SVD-LM in addition to the biased SVD. The results show a significant improvement in the prediction accuracy achieved by our methods. The best accuracy achieved by biased SVD is 0.8669 RMSE and 0.677 MAE for ML-1M dataset, and 0.8063 RMSE and 0.6198 MAE for ML-10M dataset. On the other hand, our proposed methods SVD-LM, WAFE-SVD and WAFE-SVD-LM have achieved improvement in the RMSE of 1%, 2% and 3% for ML-1M and 2%, 3% and 4% for ML-10M respectively. In addition, they achieved MAE improvement of 1%, 2% and 2% for ML-1M and 1%, 2% and 3% for ML-10M respectively. Results are showed in Table I.

TABLE I: The best RMSE and MAE obtained by our proposed methods WAFE-SVD, SVD-LM and WAFE-SVD-LM compared with PMF, SVD and SVD++ for MovieLens-1M (ML-1M) and MovieLens-10M (ML-10M) datasets.

| Method       | ML-1M   | ML-10M  |
|--------------|---------|---------|
|              | RMSE    | MAE     | RMSE     | MAE     |
| SVD          | 0.8669  | 0.677   | 0.8063   | 0.6198  |
| SVD++        | 0.8617  | 0.6708  | 0.7961   | 0.6096  |
| PMF          | 0.8583  | 0.6656  | 0.7905   | 0.608   |
| SVD-LM       | 0.8605  | 0.6685  | 0.7933   | 0.6077  |
| WAFE-SVD     | 0.8479  | 0.6619  | 0.7809   | 0.5976  |
| WAFE-SVD-LM  | 0.8386  | 0.6549  | 0.7689   | 0.5861  |

Fig. 1 shows the improvement of the accuracy for all methods in the first 20 epochs. However, different methods need more iterations in order to achieve the optimal accuracy. In practice, we found that our methods have significantly reduced the number of required epochs for achieving the
best accuracy for ML-1M and ML-10M datasets; results are shown in Table II and Table III. The results show that our WAFE-based methods, i.e., WAFE-SVD and WAFE-SVD-LM, have achieved the best RMSE and MAE in the first 25 epochs for both datasets, while all other methods needed at least 40 epochs to accomplish the best accuracy. For example, for ML-1M dataset, SVD-LM needed 55 epochs to achieve the best RMSE and 70 epochs for attaining the best MAE. However, SVD needed 85 epoch to accomplish the best RMSE and 100 epochs to achieve the best MAE. Fig. 2 illustrates a comparison of the best RMSE and MAE obtained by each method considering the number of epochs needed when running the experiment for ML-1M and ML-10M datasets. All our three proposed methods have achieved more accurate prediction in earlier iterations compared to the biased SVD, while our WAFE-based methods outperformed SVD-LM. This demonstrates the significant effectiveness of WAFE on reducing the prediction error and the time required to achieve the best accuracy. Moreover, as WAFE-SVD-LM is combining WAFE-SVD and SVD-LM, which incorporate features extracted by WAFE and the local mean of users and items ratings, it outperformed all compared methods.

B. Comparison with other methods

We compare our proposed methods SVD-LM, WAFE-SVD and WAFE-SVD-LM with three baseline methods, i.e., biased SVD, SVD++ and PMF, which are the state of the art of matrix factorization collaborative filtering methods. Results are shown in Table I.

1) Prediction accuracy: It is clear from the results that our WAFE-based methods, i.e., WAFE-SVD and WAFE-SVD-LM outperformed all the other compared methods. This demonstrates the effectiveness of assigning the values of users and items latent features with meaningful values instead of initialising them with random values. Besides, as our proposed feature extraction method (WAFE) relies on generating users’ and items’ features based on user-item rating interaction, initialising users’ and items’ latent features with these extracted values minimises the error in an early stage (the first epochs), as shown in Fig. 1.

2) Computational time: Using WAFE to initialise users’ and items’ latent features values allows the method to achieve the best accuracy at earlier stages compared to the other methods which adopt random initialisation of users’ and items’ latent features, as illustrated in Table II and Table III. The
Fig. 2: A comparison of the performance of all methods considering the number of epochs needed to achieve the best RMSE and MAE using MovieLens-1M (ML-1M) and MovieLens-10M (ML-10M) datasets.

TABLE II: The required number of epochs (iterations) to achieve the best RMSE and MAE of the compared methods for MovieLens-1M dataset.

| Method     | Best RMSE | Epochs | Best MAE | Epochs |
|------------|-----------|--------|----------|--------|
| SVD        | 0.8669    | 85     | 0.677    | 100    |
| SVD++      | 0.8617    | 52     | 0.6708   | 52     |
| PMF        | 0.8583    | 40     | 0.6656   | 49     |
| SVD-LM     | 0.8605    | 55     | 0.6685   | 70     |
| WAFE-SVD   | 0.8479    | 17     | 0.6619   | 19     |
| WAFE-SVD-LM| 0.8386    | 18     | 0.6549   | 20     |

TABLE III: The required number of epochs (iterations) to achieve the best RMSE and MAE of the compared methods for MovieLens-10M dataset.

| Method     | Best RMSE | Epochs | Best MAE | Epochs |
|------------|-----------|--------|----------|--------|
| SVD        | 0.8063    | 81     | 0.6198   | 87     |
| SVD++      | 0.7961    | 58     | 0.6096   | 74     |
| PMF        | 0.7905    | 41     | 0.608    | 46     |
| SVD-LM     | 0.7933    | 55     | 0.6077   | 62     |
| WAFE-SVD   | 0.7809    | 22     | 0.5976   | 24     |
| WAFE-SVD-LM| 0.7689    | 24     | 0.5861   | 25     |

overall performance for all the compared methods considering the prediction accuracy and the required number of epochs to achieve the best accuracy for each method is shown in Fig. 2. It is clear that our two proposed WAFE-based methods, i.e., WAFE-SVD and WAFE-SVD-LM, outperform all other methods.

3) Parameter tuning cost: Finding the optimal number of latent factor is a crucial task that needs to be tackled in order to achieve the optimal results of matrix factorization collaborative filtering methods. This can be performed by running the experiment several times, assigning the value of d to different numbers. For example, in this study, for all
methods that do not use WAFE, i.e., SVD, SVD++, PMF and SVD-LM, we tried the values 10, 20, 30, 40, 50 to assign $d$. Then, we assigned each method to its optimal value of $d$. This requires more effort and time. However, this step is not required when using WAFE as the number of latent factors $d$ is determined by WAFE, which is the number of user/item features.

VI. CONCLUSION

In this paper, we showed how user-item interaction history can be leveraged for feature extraction by proposing WAFE method. Besides, we proposed a matrix factorization collaborative filtering method that adopts WAFE in generating user and item latent features. Furthermore, we proposed a rating prediction approach that incorporates the local mean of users’ and items’ ratings. The proposed model has been evaluated by comparing its performance against the state-of-the-art collaborative filtering methods using two real-world datasets. The results showed that our proposed model outperformed the compared methods in terms of prediction accuracy and computational time.

In this paper, the proposed method has been applied to the domain of movie recommendation. However, the method can be applied to other domains, such as book recommendation and fashion products recommendation. In the future, we plan to extend our experiments to include benchmark datasets from different domains (e.g., Book-Crossing and Amazon-Fashion).

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, “Introduction to recommender systems handbook,” in Recommender systems handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 1–35.
[2] X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” Advances in artificial intelligence, vol. 2009, pp. 1–19, 2009.
[3] P. Lops, M. de Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 73–105.
[4] R. Burke, “Hybrid recommender systems: Survey and experiments,” User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, Nov 2002.
[5] C. C. Aggarwal et al., Recommender systems The textbook. Springer, 2016, vol. 1.
[6] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, vol. 42, no. 8, pp. 30–37, 2009.
[7] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in The adaptive web. Springer, 2007, pp. 291–324.
[8] T. Zhao, J. McAuley, and I. King, “Leveraging social connections to improve personalized ranking for collaborative filtering,” in Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, ser. CIKM ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 261–270.
[9] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, “Recommender systems with social regularization,” in Proceedings of the fourth ACM international conference on Web search and data mining, 2011, pp. 287–296.
[10] S. K. T. Lam, D. Frankowski, and J. Riedl, “Do you trust your recommendations? an exploration of security and privacy issues in recommender systems,” in Emerging Trends in Information and Communication Security, G. Müller, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 14–29.
[11] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, 2005.
[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Application of dimensionality reduction in recommender system—a case study,” Minnesota Univ Minneapolis Dept of Computer Science, Tech. Rep., 2000.
[13] D. Kim and B.-J. Yun, “Collaborative filtering based on iterative principal component analysis,” Expert Systems with Applications, vol. 28, no. 4, pp. 823 – 830, 2005.
[14] Y. Koren and R. Bell, Advances in Collaborative Filtering. Boston, MA: Springer US, 2015, pp. 77–118.
[15] R. Bell, Y. Koren, and C. Volinsky, “Modeling relationships at multiple scales to improve accuracy of large recommender systems,” in Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’07. New York, NY, USA: Association for Computing Machinery, 2007, p. 95–104.
[16] A. Paterek, “Improving regularized singular value decomposition for collaborative filtering,” in Proceedings of KDD cup and workshop, vol. 2007, 2007, pp. 5–8.
[17] Y. Koren, “Factorization meets the neighborhood: A multifaceted collaborative filtering model,” in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 426–434.
[18] A. Mnih and R. R. Salakhutdinov, “Probabilistic matrix factorization,” in Advances in neural information processing systems, 2008, pp. 1257–1264.
[19] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” in Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’11. New York, NY, USA: ACM, 2011, pp. 448–456.
[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” Journal of machine learning research, vol. 3, no. Jan, pp. 993–1022, 2003.
[21] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems,” in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015, pp. 1235–1244.
[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” Journal of machine learning research, vol. 11, no. Dec, pp. 3371–3408, 2010.
[23] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, “Convolutional matrix factorization for document context-aware recommendation,” in Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016, pp. 233–240.
[24] D. Kim, C. Park, J. Oh, and H. Yu, “Deep hybrid recommender systems via exploiting document context and statistics of items,” Information Sciences, vol. 417, pp. 72–87, 2017.
[25] T. S. Nguyen, “Model-based book recommender systems using naive bayes enhanced with optimal feature selection,” in Proceedings of the 2019 8th International Conference on Software and Computer Applications, ser. ICSA ’19. New York, NY, USA: ACM, 2019, pp. 217–222.
[26] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, “A hybrid collaborative filtering model with deep structure for recommender systems,” in Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[27] M. Kuhn and K. Johnson, Feature engineering and selection: A practical approach for predictive models. CRC Press, 2019.
[28] A. Zheng and A. Casari, Feature engineering for machine learning: principles and techniques for data scientists. “ O’Reilly Media, Inc.”, 2018.