

Received June 24, 2021, accepted July 10, 2021, date of publication July 14, 2021, date of current version July 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3097207

Novel SDDM Rating Prediction Models for Recommendation Systems

DIAO ZHOU¹, SHENGNAN HAO¹, HAIYANG ZHANG², CHENXU DAI¹, YONGLI AN¹, ZHANLIN JI^{1,5}, (Member, IEEE), AND IVAN GANCHEV^{3,4,5}, (Senior Member, IEEE)

¹Department of Artificial Intelligence, North China University of Science and Technology, Tangshan 063009, China

²Department of Computer Science, The University of Sheffield, Sheffield S10 2TN, U.K.

³Department of Computer Systems, University of Plovdiv "Paisii Hilendarski", 4000 Plovdiv, Bulgaria

⁴Institute of Mathematics and Informatics–Bulgarian Academy of Sciences, 1040 Sofia, Bulgaria

⁵Telecommunications Research Centre (TRC), University of Limerick, Limerick, V94 T9PX Ireland

Corresponding authors: Zhanlin Ji (zhanlin.ji@ncst.edu.cn) and Ivan Ganchev (ivan.ganchev@ul.ie)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFE0135700, and in part by the Bulgarian National Science Fund (BNSF) under Grant КП-06-ИП-КИТАЙ/1.

ABSTRACT The accuracy of behavioral interactive features is a key factor for improving the performance of rating prediction. In order to deeply explore the potential rules of user behavior and enhance the accurate representation of interactive features, this paper proposes two rating prediction models, based on the spatial dimension and distance measurement (SDDM), under the premise of taking the mean value of the user behavior history as a user feature, and obtaining the interactive features of an item and a user by calculating the distance between them in each feature dimension. In the proposed SDDM-Var and SDDM-PCC models, the variance and the Pearson correlation coefficient (PCC) are respectively utilized to evaluate the user's attention to each feature dimension as to further obtain the weight vector of the interactive features. Finally, in order to improve the generalization ability of the proposed models, the rating prediction is accomplished by means of a specially designed multi-layer full-connection neural network. The conducted experiments with two public MovieLens datasets demonstrate the superior rating prediction performance of the proposed models in comparison with the existing baseline models, in terms of the root mean square error (RMSE), by achieving values of 0.865 and 0.872 on MovieLens 100K, and 0.839 and 0.832 on MovieLens 1M, respectively for SDDM-Var and SDDM-PCC.

INDEX TERMS Variance, Pearson correlation coefficient (PCC), rating prediction, recommendation system, neural network, Item2Vec.

I. INTRODUCTION

It is quite difficult for users in the 'Big Data' era to quickly find and obtain valuable knowledge from the massive information volumes presented by multiple sources. The emergence of recommendation systems has provided a generic solution to this problem. Such systems are now widely used in different fields, such as e-commerce, video and music streaming, news delivery, etc., and huge profits have been made by many Internet companies as a result of this [1], [2].

In the field of personalized recommendation systems, the most widely utilized approach is collaborative filtering (CF) [3], which is based on the user behavior sequence.

The associate editor coordinating the review of this manuscript and approving it for publication was Ehab Elsayed Elattar¹.

CF provides recommendations based on computing the similarity between users or items, e.g., by using the cosine similarity or the Pearson correlation coefficient (PCC). CF is simple, quick, and efficient. However, it is difficult for the traditional CF to meet the current needs due to its limitation in generalization ability. For this, matrix factorization (MF) [4] has been proposed, which maps users and items into the same potential feature space according to the rating matrix. The rating prediction can be then realized by calculating the inner product of the potential feature vectors of a user and an item. MF with gradient descent proved to deliver an obvious performance improvement compared to traditional CF. However, the simple inner product operation still heavily limits the generalization ability of MF. With the deep learning (DL) technology developed in the past few years, the deep neural

networks (NNs) have evolved as a feasible way to improve the generalization ability of recommendation systems. For instance, by linking classical MF with NN, the neural CF (NCF), proposed by He *et al.* in [5], uses multi-layer perceptron to replace the inner product operation of MF, which effectively enhances the generalization ability of the model.

NCF is able to achieve better recommendation performance only at the expense of regularly training the potential feature vectors of a user and an item. However, due to the large number of users and items, such regular training consumes lots of computing resources and time. As a solution to this problem, in some recent research works, the user features have been obtained according to the mean value of the user behavior history [6], [7]. However, this approach often reduces the accuracy of rating prediction. Consequently, some other researchers attempted to introduce time decay and attention mechanism for achieving more accurate user feature representation [8], [9]. Our previous research efforts in the field of rating prediction are presented in [10], [11].

In this paper, two rating prediction models, based on spatial dimension and distance measurement (SDDM), are proposed. The variance and PCC are used to deeply explore the variation rules of potential feature vectors from the user behavior history and a multi-layer full-connection NN is adopted to boost the accuracy of rating prediction and the generalization ability of the proposed models. The initial item features are obtained by pre-training, based on the Item2Vec model proposed in [12] for training item feature vectors. The user feature is the mean value of the potential feature vectors of the item in the user behavior history, while the interactive feature represents the difference (distance) between the item feature and the user feature in each dimension. A weight vector is constructed by taking the variance of the user behavior history and PCC as the interactive feature to alleviate the limitation of using historical mean value as a user feature. Different from traditional CF that uses PCC to measure the correlation between users or items [13], the research presented in this paper adopts PCC to measure the correlation between potential features, and rating level and variance to evaluate the stability of potential features in various dimensions. The analysis of results, obtained from the conducted experiments based on two public MovieLens datasets by using the mean absolute error (MAE) and root mean square error (RMSE) as evaluation metrics, indicates that two proposed SDDM models are equipped with excellent rating prediction ability.

In summary, the research work presented in this paper was focused on solving two main problems:

- 1) How to enhance the interactive feature representation by exploring the variation rules of the user potential features.
- 2) How to improve the generalization ability of the recommendation model as to more accurately predict the user rating of items.

In order to solve these problems, we have conducted a lot of research and experiments. For this paper, the main contributions are as follows:

- 1) Two SDDM rating prediction models are proposed, whereby the interactive features of users and items are obtained by virtue of the calculation of the distance between them in each dimension.
- 2) The utilization of the variance and PCC is put forward to deeply explore the variation rules of user potential features. Based on this, a weight vector of interactive features is constructed for the accurate representation of interactive features.
- 3) Under the premise of using the Item2Vec model for item feature pre-training, a specially designed multi-layer full-connection neural network is utilized to improve the generalization ability and rating prediction performance.

II. RELATED WORK

A. CONVENTIONAL RECOMMENDATION APPROACHES

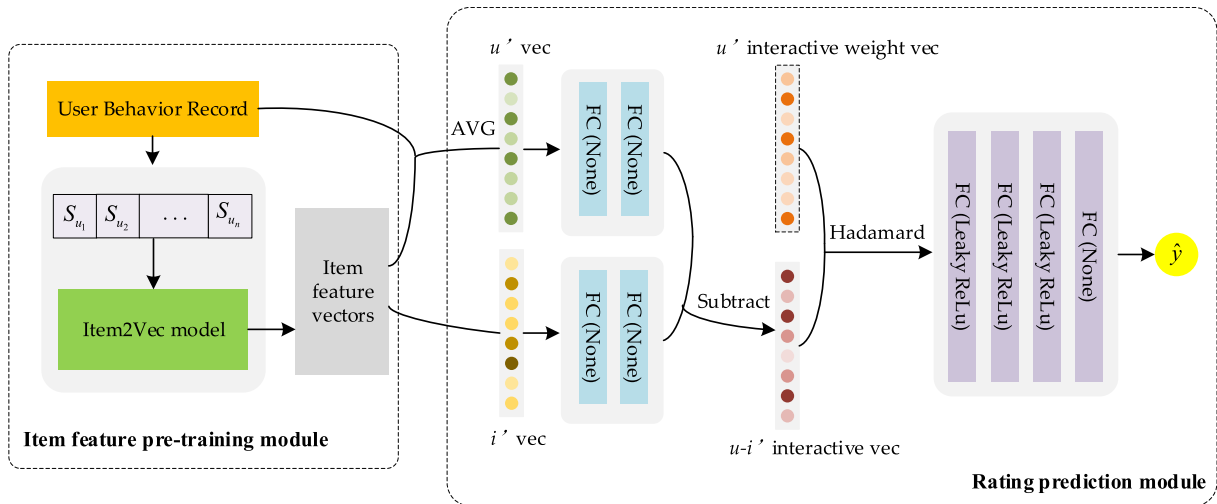
The conventional recommendation approaches could be divided into three groups: content-based filtering (CBF), CF, and hybrid approaches utilizing both content and collaborative information.

CBF [14] depends on the item portrait and user behavior. It can search for similar items under the portrait information of interesting items in the user history and recommend them to the user. CBF is widely used in industry due to its simplicity and efficiency.

However, when user–item ratings are the only information available for analysis, CF [15]–[17] is favored over CBF. A variety of CF recommendation models have been developed, such as SVD [18], SVD++ [19], and other MF-based models. Compared with the most basic CF, these models can improve the recommendation- and generalization performance to a large extent. CF and MF are still hot research topics in the area of recommendation systems.

B. DEEP LEARNING-BASED RECOMMENDATION APPROACHES

Deep learning (DL) has been successfully applied in the field of computer vision and natural language processing (NLP). A lot of research work has verified the excellent performance of DL in dealing with regression and classification tasks, e.g., [20], [21]. A new trend in recent years is represented by the DL-based recommendation approaches, initially developed by some large Internet technology companies. For instance, the wide & deep model, proposed by Google [22], is provided with the advantages of the logical regression model and the NN model, resulting in better memory- and generalization abilities. The NN-based recommendation model, proposed by YouTube in the field of video streaming [23], is suitable for large-scale data screening recommendation scenarios. The deep interest network (DIN) model, proposed by Alibaba, can capture the user's interest direction by utilizing the attention mechanism, leading to a better recommendation performance [24]. Academia has also paid a lot of attention to the use of DL in recommendation systems.



- FC (Leaky ReLU): full-connection NN layer using Leaky ReLU as an activation function
- FC (None): full-connection NN layer without activation function
- u' vec: user feature vector
- i' vec: item feature vector
- u' interactive weight vec: user weight vector of interactive features
- Hadamard: Hadamard product
- $u-i'$ interactive vec: user-item interactive feature vector
- Subtract: Distance metric
- AVG: Average pooling

FIGURE 1. The overall structure of the proposed SDDM models.

For instance, Kuang *et al.* proposed the DMF-CDR model, utilizing a multi-layer perceptron to learn the feature representation of users and items, and adding a cross-domain information to alleviate the sparsity problem of CF and to improve the recommendation performance [25]. Based on a graph neural network (GNN), Xian *et al.* proposed the ReGNN model for recommendation tasks, which combines a repeated search mechanism and achieved more accurate prediction by modeling the repeated exploration behavior pattern of users [26]. These and other DL-based recommendation models, developed in recent years, demonstrated the advantages of the DL technology in the field of recommendation systems.

C. ITEM2VEC MODEL

A common approach in this field is to construct corresponding potential feature vectors for each user and item, regarded as an embedding process. Being an improved version of Word2Vec [27], [28], Item2Vec [12] is an excellent item embedding training model, which can better adapt to different recommendation scenarios. Basically, Item2Vec maps all items to a potential feature space according to the user behavior history and calculates the similarity between items using the cosine of the angle. In addition, the simple full-connection layer structure in Item2Vec ensures its fast-computing speed and excellent embedding performance. The optimization objectives of Item2Vec are defined as follows:

$$\frac{1}{K} \sum_{i=1}^K \sum_{j \neq i}^K \log p(w_j | w_i), \quad (1)$$

where w_i and w_j are the center item and its corresponding surrounding items in any behavior sequence, and K is the sequence length.

III. SDDM MODELS

A. PROBLEM DEFINITION

Given a set of users $U = \{u_1, u_2, \dots, u_n, \dots, u_N\}$, a set of items $I = \{i_1, i_2, \dots, i_m, \dots, i_M\}$, and a set $S_{u_n} = \{S_{u_n}^{i_1}, S_{u_n}^{i_2}, \dots, S_{u_n}^{i_m}, \dots, S_{u_n}^{i_M}\}$ of the interaction behavior of user u_n with these items, the purpose of the proposed SDDM models is to predict the future interaction behavior $S_{u_n}^{i_m}$ of user u_n towards item i_m , i.e., to predict any rating values $S_{u_n}^{i_m}$ missing in S_{u_n} .

B. OVERVIEW

The proposed SDDM models consist of the following two modules (Fig. 1):

- 1) *Item feature pre-training module* – based on Item2Vec [12], this module maps all items to the feature space by inputting the user behavior record in order to get the item feature vectors that express user- and item features existing in a multi-dimensional space. The user behavior record refers to the behavior history sets, generated so far based on the user interaction with items, containing the rating values given by the user to different items (generally, a separate rating value is assigned by the user to each item feature). The behavior record could be interpreted as a sentence composed of different words, following the Word2Vec concept.

To enable faster convergence of the rating prediction algorithm executed in the other module, this module obtains the feature vector of an item by calculating the co-occurrence probability of different items by utilizing, for pre-training, the Item2vec model instead of a common embedding layer.

- 2) *Rating prediction module* – based on a specially designed multi-layered full-connection NN, this module is used to improve the generalization ability of the proposed models by enhancing the feature representation. The input user- and item feature vectors are respectively processed by the first two layers of the neural network and then the difference between the two corresponding feature dimensions is calculated to obtain the interactive feature vector of the particular user and item. Finally, the Hadamard product calculation is performed on the interactive feature vector and the interactive weight, and the predicted rating is obtained through the final four layers of the full-connection NN. To compensate the shortcomings of the full-connection NNs, such as over-fitting, regularization parameters are added in the process of model training. The implementation of the designed multi-layer full-connection NN is described in detail in Subsection III.F.

C. PRE-TRAINING OF ITEM FEATURES

Pre-training of item features can be done in different ways, e.g., by utilizing the SVD model [18] or an embedding model [12], [27], [29]. The latter approach was utilized in the proposed SDDM models for the pre-training of item features, based on Item2Vec.

In the original Item2Vec model, all items in the behavior record of user u_n are regarded as a sentence for training without distinguishment between high and low ratings. However, the SDDM models, proposed in this paper, focus on *explicit* feedback behavior, so for each user, the items in his/her behavior records are first divided into different sentences according to their rating value given by the user, so that the items in the same sentence are equally graded. Then these sentences are inputted into the Item2Vec model for training and getting the set $V_{(I)} = \{v_{i_1}, v_{i_2}, \dots, v_{i_m}, \dots, v_{i_M}\}$, which contains the feature vectors of all items. Each vector v_{i_m} is a dense vector with dimension j , where j represents the total number of item features.

In the proposed SDDM models, the interactive features of a user and an item are calculated through the difference between them in each dimension. The interactive features need to reflect the similarity between the user- and item features to a certain extent. However, the feature vector generated by Item2Vec cannot meet this requirement because it measures the similarity between the two kinds of features through the cosine of the angle and the difference in each dimension cannot directly reflect the level of similarity. Consequently, it is necessary to normalize the calculation of all generated item feature vectors so that each

vector can be transformed into a unit vector with a module length of 1. So, in the SDDM models presented in this paper, the following normalization is applied for each vector $v_{i_m} = (v_{i_m}^{(1)}, v_{i_m}^{(2)}, \dots, v_{i_m}^{(j)})$:

$$v'_{i_m} = \frac{1}{\sqrt{\sum_{c=1}^j (v_{i_m}^{(c)})^2}} (v_{i_m}^{(1)}, v_{i_m}^{(2)}, \dots, v_{i_m}^{(j)}) \quad (2)$$

D. REPRESENTATION OF USER FEATURES

User feature is the mean value of all positive rating values given by a user (and presented in his/her behavior history record) to a particular feature of items. Given a set $V_{(I)}^{u_n^+} = \{v_1, v_2, \dots, v_q, \dots, v_Q\}$, $V_{(I)}^{u_n^+} \subseteq V_{(I)}$, of feature vectors of items containing only positive rating values given by user u_n , where the feature vector of item q in the set is $v_q = (v_q^{(1)}, v_q^{(2)}, \dots, v_q^{(j)})$, the interest feature matrix $M_{u_n^+}$ of user u_n can be presented as follows:

$$M_{u_n^+} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_Q \end{bmatrix} = \begin{bmatrix} v_1^{(1)} & v_1^{(2)} & \dots & v_1^{(j)} \\ v_2^{(1)} & v_2^{(2)} & \dots & v_2^{(j)} \\ \vdots & \vdots & \ddots & \vdots \\ v_Q^{(1)} & v_Q^{(2)} & \dots & v_Q^{(j)} \end{bmatrix} \quad (3)$$

Then, the feature vector of user u_n can be obtained by average pooling of the item feature vectors:

$$v_{u_n} = \frac{1}{Q} \sum_{q=1}^Q (v_q^{(1)}, v_q^{(2)}, \dots, v_q^{(j)}) \quad (4)$$

It can be considered that the user- and item feature vectors are in the same potential space as the former is obtained by average pooling of the item feature vectors present in the user behavior history record. A sample 3D feature space is shown in Fig. 2, where the blue arrows represent the feature vectors v_1 and v_2 of two items in the user behavior sequence, whereas the red arrow represents the feature vector of user v_{u_n} resulted from the average pooling of the feature vectors of these two items.

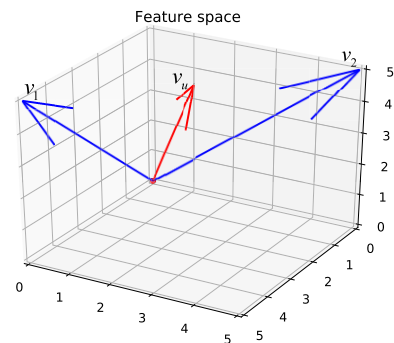


FIGURE 2. A sample 3D feature space.

E. FEATURE INTERACTION

Feature interaction is about getting an interactive feature vector according to the user- and item feature vectors under

interaction rules, which is similar to the concatenation operation performed on the user- and item features. The user- and item feature vectors are mapped respectively through the first two layers of the designed full-connection NN as to enhance feature representation prior to the implementation of feature interaction. The process can be represented as follows:

$$\begin{aligned} v_{u_n} &\leftarrow \psi_{None}(v_{u_n}); \\ v_{i_m} &\leftarrow \psi_{None}(v_{i_m}), \end{aligned} \quad (5)$$

where ψ_{None} denotes the full-connection NN and *None* means that the activation function is not set.

Interactive feature is a vector that is composed of the distance between the user features and item features in each dimension. In this step, the user- and item feature vectors could be each regarded as a point in the potential space and the distance between the two points in each dimension could be deemed as a reflection of the similarity or difference between the two entities to some extent. Thus, the interactive feature vector of item i_m and user u_n can be expressed as follows:

$$v_{i_m-u_n} = \left(\left| v_{i_m}^{(1)} - v_{u_n}^{(1)} \right|, \left| v_{i_m}^{(2)} - v_{u_n}^{(2)} \right|, \dots, \left| v_{i_m}^{(j)} - v_{u_n}^{(j)} \right| \right). \quad (6)$$

However, it is hard to achieve a good performance only by using the above interactive features as a basis for rating prediction because: (i) the user features are obtained by average pooling at the beginning, which leads to some inaccuracy; and (ii) not all dimensions would be of equal importance/interest to users, so the distance on some dimensions cannot serve as a real basis for rating prediction. Therefore, it is necessary to determine which dimensions of interactive features are meaningful and to weaken the influence of meaningless or less significant dimensions. Consequently, two different methods – the variance and PCC – are used in this paper to capture the user's interest in each dimension, based on the user behavior history record, which results in two different models, called SDDM-Var and SDDM-PCC, respectively. The two ways of calculation of the weight $w_{u_n}^{(k)}$ of the k^{th} dimension for user u_n are described in the following two subsections.

1) CALCULATION OF DIMENSIONS' WEIGHT USING VARIANCE

According to (3), the variance of each column in matrix $M_{u_n^+}$ is calculated under the circumstance of considering only the positive feedback of user u_n . The smaller the variance is, the more stable the value of the corresponding dimension and the stronger the reference value. Since the purpose of variance calculation is to get the weight of interactive features, the smaller the variance of the dimension is, the larger the weight value. Therefore, the weight $w_{u_n}^{(k)}$ of the k^{th} dimension for user u_n can be obtained as:

$$var_{u_n}^{(k)} = \frac{\sum_{q=1}^Q \left(v_q^{(k)} - \bar{v}^{(k)} \right)^2}{Q}; \quad (7)$$

$$w_{u_n}^{(k)} = e^{-\gamma * var_{u_n}^{(k)}}, \quad (8)$$

where $\bar{v}^{(k)}$ represents the mean value at the K dimension in the behavior history record of user u_n , and γ is a user-defined parameter.

2) CALCULATION OF DIMENSIONS' WEIGHT USING PCC

When PCC is used for calculating the dimensions' weight, all rating values (both negative and positive) given by a user (and presented in his/her behavior history record) are considered, i.e., not just the positive rating values. So, the interactive feature vector of item i_m and user u_n becomes:

$$v_{i_m-u_n} = \left(v_{i_m}^{(1)} - v_{u_n}^{(1)}, v_{i_m}^{(2)} - v_{u_n}^{(2)}, \dots, v_{i_m}^{(j)} - v_{u_n}^{(j)} \right). \quad (9)$$

Then, the weight $w_{u_n}^{(k)}$ of the k^{th} dimension for user u_n can be obtained by using PCC as:

$$w_{u_n}^{(k)} = \frac{\sum_{m=1}^M \left(v_{i_m}^{(k)} - \bar{v}^{(k)} \right) \left(r_{i_m}^{u_n} - \bar{r}_{u_n} \right)}{\sqrt{\sum_{m=1}^M \left(v_{i_m}^{(k)} - \bar{v}^{(k)} \right)^2 \sum_{m=1}^M \left(r_{i_m}^{u_n} - \bar{r}_{u_n} \right)^2}}, \quad (10)$$

where $r_{i_m}^{u_n}$ is the rating value given by user u_n to item i_m and \bar{r}_{u_n} is the average value of all ratings in the behavior history record of user u_n .

The process of interactive weight calculation is formally presented as Algorithm 1.

Algorithm 1 Calculation of Dimensions' Weight for User u_n

Input: i_vec_List (feature vector of items in the behavior history record), i_rat_List (items rating record), γ (user-defined parameter), dim (dimension of feature vector)

Output: dw (dimensions' weight)

1: **start:**

2: **Define** dw as a list

3: **for** $k = 1$ to dim

4: $X = (i_vec_List_1^{(k)}, i_vec_List_2^{(k)}, \dots, i_vec_List_n^{(k)})$

5: **if** using PCC:

6: $Y = (i_rat_List_1^{(k)}, i_rat_List_2^{(k)}, \dots, i_rat_List_n^{(k)})$

7: $w^{(k)} = \text{Pearson}(X, Y)$

8: **end if**

9: **if** using Variance:

10: $w^{(k)} = \exp(-\gamma * \text{Var}(X))$

11: **end if**

12: append $w^{(k)}$ to dw

13: **end for**

14: **return** dw

15: **end**

After calculating the weight of each dimension (either by using the variance or PCC), the final interactive feature vector is obtained as follows:

$$v_{i_m-u_n} \leftarrow \left(v_{i_m-u_n}^{(1)} * w_{u_n}^{(1)}, v_{i_m-u_n}^{(2)} * w_{u_n}^{(2)}, \dots, v_{i_m-u_n}^{(j)} * w_{u_n}^{(j)} \right). \quad (11)$$

F. OUTPUT OF RATING PREDICTION

In the proposed SDDM models, a specially designed multi-layer full-connection NN is used for rating prediction. The output \hat{y} of the corresponding rating prediction module (c.f. Fig. 1) is:

$$\hat{y} = f_{LReLU}(\dots f_{LReLU}(g(\psi_{None}(\dots \psi_{None}(v_{u_n})), \psi_{None}(\dots \psi_{None}(v_{i_m})), w_{u_n}))), \tau b_{u_n} + (1 - \tau) b_{i_m}, \quad (12)$$

where f_{LReLU} denotes a full-connection NN layer using Leaky ReLU as an activation function, ψ_{None} denotes a full-connection NN layer without activation function, g denotes the calculation process of the final interactive feature vector, b_{u_n} and b_{i_m} are the user- and item offset terms (c.f. [4]), and τ ($0 \leq \tau \leq 1$) is the offset term coefficient. When $\tau > 0.5$, the rating value is inclined to user personalization, otherwise, it tends to item popularization.

In the training process of the rating prediction module, the output of the l^{th} layer of the full-connection NN can be expressed as in the process of forward propagation as:

$$Y_l = f_a(X_l * W_l^T + B_l), \quad (13)$$

where X_l denotes the input of the l^{th} layer of the full-connection NN, f_a is the activation function, and W_l and B_l are the weight and bias, respectively, which are constantly updated during the training.

In the process of backward propagation, the mean square error (MSE) function is used as a loss function and L_2 regularization is added to prevent the model from overfitting:

$$Loss = \frac{1}{k} \left(\sum_{i=1}^k (y_i - \hat{y}_i)^2 + \sum_{l=1}^L \varphi_{L2}(W_l, \lambda) \right), \quad (14)$$

where y_i and \hat{y}_i are the real- and predicted rating values, respectively, k is the number of samples of the current training batch, L is the total number of NN layers, and λ is the regularization coefficient.

The rating prediction process is formally presented as Algorithm 2.

IV. EXPERIMENTS AND RESULTS

A. DATASETS

To evaluate the rating prediction performance of the proposed models in comparison with the existing baselines, corresponding experiments were conducted on two public movie rating datasets – MovieLens 100K (ML-100K) and MovieLens 1M (ML-1M).¹ ML-100K contains 100,000 ratings from 1000 users on 1700 movies, whereas ML-1M contains one million ratings from 6000 users on 4000 movies. In the experiments, the ratings in each dataset were divided into five groups, corresponding to the rating values, ranging from 1 to 5. In terms of the user rating behavior, only users who provided at least 20 movies ratings were considered in the experiments.

¹ <https://grouplens.org/datasets/movielens/>

Algorithm 2 Rating Prediction

Input: $train_set$ $\{u_vec$ (user feature vector), i_vec (item feature vector), b_u (user rating bias), b_i (item rating bias), u_dw (user dimensions' weight), y_{label} (rating label)}, τ (bias control parameter for users and items), eps (number of epochs), $batchsize$ (number of samples per training session)

Output: *Rating Model* (NN parameters)

1: **start:**

2: *Rating Model* initialization

3: **for** i in range (eps):

4: **for** j in range ($\text{len}(train_set) / batchsize$):

5: $u_vec' = \varphi_{None}^{(2)}(\varphi_{None}^{(1)}(u_vec))$

6: $i_vec' = \varphi_{None}^{(4)}(\varphi_{None}^{(3)}(i_vec))$

7: Calculate v_{i-u} based u_vec' and i_vec' , using (6) and (9)

8: $v_{i-u} \leftarrow \text{Hadamard}(v_{i-u}, u_dw)$, c.f. (11)

9: $\hat{y} = f_{LReLU}^{(n)}(\dots f_{LReLU}^{(1)}(v_{i-u})) + \tau * b_u + (1 - \tau) * b_i$

10: Calculate $Loss(\hat{y}, y_{label})$

11: Update all NN parameters in $f_{LReLU}^{(k)}, \varphi_{None}^{(k)}$

12: **end for**

13: **end for**

14: Save *Rating Model*

15: **end**

B. EVALUATION METRICS

For evaluating the rating prediction performance of different models included in the comparison, the standard metrics RMSE and MAE were used, defined as:

$$RMSE = \sqrt{\frac{1}{K} \sum_{i=1}^K (y_i - \hat{y}_i)^2}; \quad (15)$$

$$MAE = \frac{1}{K} \sum_{i=1}^K |y_i - \hat{y}_i|. \quad (16)$$

C. MODELS FOR COMPARISON

The two proposed models, SDDM-Var and SDDM-PCC, were compared to the following eight baseline models:

- IGMC [30] – an inductive graph-based matrix completion model, which can achieve good recommendation performance without using any auxiliary information.
- GC-MC [31] – a graph autoencoder model, based on distinguishable messages passed on a bidirectional interaction graph.
- Factorized EAE [32] – a DL-based model for cross-domain recommendation, with good generalization performance.
- NNMF [33] – a recommendation model combining NN and MF, whereby the inner product operation of MF is replaced by a multi-layer perceptron.
- MetricF [34] – a measurement decomposition model, based on the Euclidean distance meeting inequality attributes to measure the explicit proximity between

users and items for rating prediction or personalized ranking.

- AutoSVD++ [35] – a hybrid model, which integrates a compression autoencoder into the MF framework.
- HGAR [36] – a recommendation model, which combines the SVD algorithm and a multi-layer perception while also considering both implicit and explicit information.
- SSAERec [37] – a rating prediction model, based on a stacked sparse autoencoder and MF.

D. EXPERIMENT SETTINGS

The experiments with the proposed models were carried out under the DL framework Pytorch. Each dataset was divided according to the level of rating by means of stratified random sampling, in which the training set and the test set account for 80% and 20%, respectively. In addition, the experiments with the proposed models were conducted in two steps – pre-training of item features and rating prediction. When using Item2Vec to pre-train item features, the training method Skip-gram was applied with 150 iterations. In the rating prediction step, the learning rate, the regularization coefficient, and the bias term coefficient were set to $\alpha = 0.005$, $\lambda = 0.005$, and $\tau = 0.2$, respectively. For calculating the weight of dimensions based on the variance, the value of parameter γ was set to 15.

E. RESULTS

1) PERFORMANCE COMPARISON OF SDDM-VAR AND SDDM-PCC MODELS

Fig. 3 and Fig. 4 show the MAE and RMSE downward trend, respectively, with increasing the number of epochs² for the two proposed models, SDDM-Var and SDDM-PCC, based on the ML-100K and ML-1M datasets, whereby the item- and user feature dimensions³ are set to 128 for both models. From Fig. 3, the following observations can be made: (i) with increasing the number of epochs, the decrease of MAE is more stable for both models on ML-100K; (ii) on ML-1M, SDDM-PCC needed 35 epochs to start performing stably better than SDDM-Var (in terms of MAE); however, on ML-100K, SDDM-PCC is generally outperformed by SDDM-Var; (iii) on ML-1M, the MAE values for both models are lower than that on ML-100K, as a result of the much greater amount of data available on ML-1M, which allows the neural network to better learn the sample features of data and achieve better rating prediction performance. From Fig. 4, it can be observed that the downward trend of RMSE closely follows that of MAE depicted in Fig. 3. Here, 30 epochs were needed for SDDM-PCC to start performing stably better (in terms of RMSE) than SDDM-Var on ML-1M. However, on ML-100K, SDDM-Var is overall outperformed by SDDM-PCC.

²Epoch -a parameter, defining the number of times a model has worked through the entire training set.

³Feature dimension represents the amount of information expressed by a feature vector.

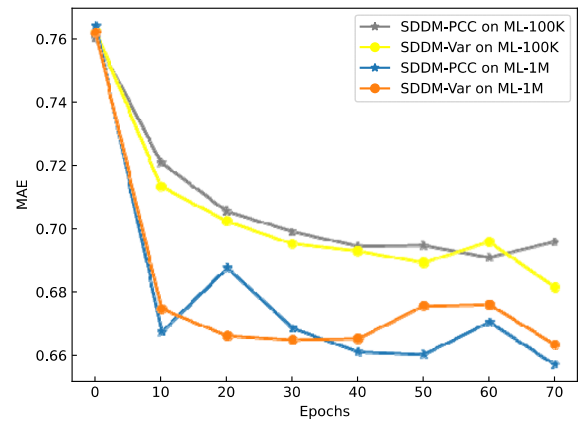


FIGURE 3. The dependence of MAE on the number of epochs for the SDDM-Var and SDDM-PCC models, on datasets ML-100K and ML-1M.

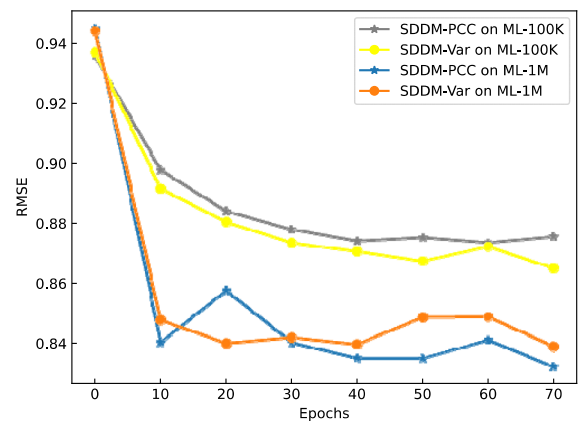


FIGURE 4. The dependence of RMSE on the number of epochs for the SDDM-Var and SDDM-PCC models, on datasets ML-100K and ML-1M.

As different values of feature dimension (d) may result in different prediction performance, additional experiments were carried out to investigate this. Tables 1 and 2 show the obtained MAE and RMSE results, respectively, for both proposed models after 70 epochs. From these results, it can be seen that both MAE and RMSE get their highest value when $d = 32$, after which they stably decrease, reaching a minimum when $d = 128$, except for the SDDM-PCC model applied on the ML-100K dataset, where the minimum is reached earlier – for MAE when $d = 64$, and for RMSE when $d = 96$, respectively. Thus, $d = 128$ is recommended for use as a good balance between the calculation time/cost and the prediction performance, as further increase of the feature dimension leads to a significant increase of the calculation time and cost.

2) PERFORMANCE COMPARISON OF ALL MODELS

The rating prediction performance comparison of all considered models (in terms of RMSE) is presented in Table 3. The RMSE results for the proposed models, SDDM-Var and SDDM-PCC, were obtained based on the conducted

TABLE 1. The MAE results for the proposed SDDM models, for different values of feature dimension (d).

Feature Dimension	SDDM-PCC		SDDM-Var	
	ML-100K	ML-1M	ML-100K	ML-1M
$d=32$	0.703	0.661	0.695	0.670
$d=64$	0.688	0.658	0.693	0.667
$d=96$	0.689	0.661	0.683	0.666
$d=128$	0.695	0.657	0.681	0.663

TABLE 2. The RMSE results for the proposed SDDM models, for different values of feature dimension (d).

Feature Dimension	SDDM-PCC		SDDM-Var	
	ML-100K	ML-1M	ML-100K	ML-1M
$d=32$	0.887	0.836	0.874	0.847
$d=64$	0.876	0.834	0.873	0.841
$d=96$	0.872	0.835	0.867	0.840
$d=128$	0.875	0.832	0.865	0.839

TABLE 3. Rating prediction performance comparison of models in terms of RMSE.

Model	ML-100K	ML-1M
IGMC [30]	0.905	0.857
GC-MC [31]	0.905	0.832
Factorized EAE [32]	0.920	0.860
NNMF [33]	0.903	0.843
MetricF [34]	0.891	0.836
AutoSVD++ [35]	0.904	0.848
HGAR [36]	0.921	0.856
SSAERec [37]	0.912	0.847
SDDM-Var	0.865	0.839
SDDM-PCC	0.875	0.832

experiments, whereas the RMSE values for the baselines were all taken from the corresponding papers. As most of these papers do not provided MAE results, this metric was not used in this comparison.

The RMSE results in Table 3 clearly demonstrate that both proposed models outperform all the baseline models, on both datasets, with only one exception on ML-1M, namely the GC-MC model, which shows equal rating prediction performance with the leader SDDM-PCC. This superior performance of the proposed models is due to the fact that they

fully consider the changes of item feature vectors in the user behavior history records in each dimension, which, compared to the baselines, allows them to more accurately grasp the users' interests in each feature dimension.

V. CONCLUSION

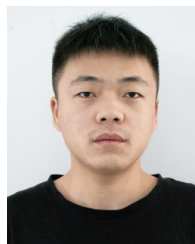
This paper has put forward two recommendation models, based on the spatial dimension and distance measurement (SDDM), for rating prediction by means of utilizing a specially designed multi-layer full-connection neural network. In the proposed models, the interactive features are obtained by calculating the distance between the user feature and item feature in each feature dimension. In addition, in order to achieve better prediction performance, the user's attention to different feature dimensions is fully considered and the interaction weight is obtained by utilizing the variance and the Pearson correlation coefficient (PCC), in the presented models, SDDM-Var and SDDM-PCC, respectively. The conducted experiments on two public MovieLens datasets confirmed that both proposed models outperform (in terms of RMSE) the baseline models considered.

Although the rating prediction performance of the proposed models is sufficiently high, there is still room for further improvement. For instance, extra user features, such as age and gender, as well as other relevant item attributes can be taken into account to improve the performance of the models. Extra work is also needed as to achieve feature crosses. All this will be a subject of our future research work. In addition, we plan to optimize both models to work faster with high values of feature dimension, as to make them more suitable for real-time recommendations.

REFERENCES

- [1] T. Arekar, M. Sonar, and N. Uke, "A survey on recommendation system," *Int. J. Comput. Appl.*, vol. 2, pp. 1–5, 2015.
- [2] H. Hwangbo, Y. S. Kim, and K. J. Cha, "Recommendation system development for fashion retail e-commerce," *Electron. Commerce Res. Appl.*, vol. 28, pp. 94–101, 2018.
- [3] L. H. Ungar and D. P. Foster, "Clustering methods for collaborative filtering," in *Proc. AAAI Workshop Recommendation Syst.*, USA, 1998, pp. 114–129.
- [4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, Australia, 2017, pp. 173–182.
- [6] N. Lu, Y. Ohsawa, and T. Hayashi, "Learning sequential behavior for next-item prediction," in *Proc. 33rd Annu. Conf. Jpn. Soc. Artif. Intell.*, Japan, 2019, pp. 1–4.
- [7] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A dynamic recurrent model for next basket recommendation," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tuscany, Italy, 2016, pp. 729–732.
- [8] Y. Li, W. Chen, and H. Yan, "Learning graph-based embedding for time-aware product recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, 2017, pp. 2163–2166.
- [9] W. Pei, J. Yang, Z. Sun, J. Zhang, A. Bozzon, and D. M. J. Tax, "Interacting attention-gated recurrent networks for recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Singapore, 2017, pp. 1459–1468.
- [10] H. Zhang, I. Ganchev, N. S. Nikolov, and M. Stevenson, "UserReg: A simple but strong model for rating prediction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Toronto, ON, Canada, 2021, pp. 3595–3599.

- [11] H. Zhang, N. S. Nikolov, and I. Ganchev, "Exploiting user feedbacks in matrix factorization for recommender systems," in *Proc. 7th Int. Conf. Model Data Eng.*, Barcelona, Spain, 2017, pp. 235–247.
- [12] O. Barkan and N. Koenigstein, "Item2vec: Neural item embedding for collaborative filtering," in *Proc. IEEE 26th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Salerno, Italy, 2016, pp. 1–6.
- [13] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Hong Kong, 2001, pp. 285–295.
- [14] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, vol. 4321. 2007, pp. 325–341.
- [15] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in *Proc. 6th Pacific Rim Int. Conf. Artif. Intell.*, Australia, 2000, pp. 679–689.
- [16] S. H. S. Chee, J. Han, and K. Wang, "Retree: An efficient collaborative filtering method," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*, Munich, Germany, 2001, pp. 141–151.
- [17] Y. Ding and X. Li, "Time weight collaborative filtering," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage.*, Berlin, Germany, 2005, pp. 485–492.
- [18] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system—a case study," in *Proc. ACM-SIGKDD Conf. Knowl. Discovery Databases (WebKDD)*, Boston, MA, USA, 2000, pp. 1–12.
- [19] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, USA, 2008, pp. 426–434.
- [20] Y. Chen et al., "The image annotation algorithm using convolutional features from intermediate layer of deep learning," *Multimedia Tools Appl.*, vol. 80, no. 3, pp. 4237–4261, 2021.
- [21] J. F. DeRose, J. Wang, and M. Berger, "Attention flows: Analyzing and comparing attention mechanisms in language models," *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1160–1170, 2021.
- [22] H.-T. Cheng et al., "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, USA, 2016, pp. 7–10.
- [23] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Boston, MA, USA, 2016, pp. 191–198.
- [24] G. Zhou et al., "Deep interest network for click-through rate prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, U.K., 2018, pp. 1059–1068.
- [25] H. Kuang, W. Xia, X. Ma, and X. Liu, "Deep matrix factorization for cross-domain recommendation," in *Proc. IEEE 5th Adv. Inf. Technol., Electron. Automat. Control Conf. (IAEAC)*, Chongqing, China, 2021, pp. 2171–2175.
- [26] X. Xian, L. Fang, and S. Sun, "ReGNN: A repeat aware graph neural network for session-based recommendations," *IEEE Access*, vol. 8, pp. 98518–98525, 2020.
- [27] T. Mikolov, G. Corrado, C. Kai, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, USA, 2013, pp. 1–12.
- [28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, USA, vol. 2, 2013, pp. 3111–3119.
- [29] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2014, pp. 701–710.
- [30] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," in *Proc. 8th Int. Conf. Learn. Represent.*, Addis Ababa, Ethiopia, 2020, pp. 1–12.
- [31] R. V. D. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," in *Proc. 24th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, London, U.K., 2018, pp. 1–7.
- [32] J. Hartford, D. Graham, K. Leyton-Brown, and S. Ravanbakhsh, "Deep models of interactions across sets," in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, pp. 1909–1918.
- [33] G. Karolina Dziugaite and D. M. Roy, "Neural network matrix factorization," 2015, *arXiv:1511.06443*. [Online]. Available: <http://arxiv.org/abs/1511.06443>
- [34] S. Zhang, L. Yao, Y. Tay, X. Xu, X. Zhang, and L. Zhu, "Metric factorization: Recommendation beyond matrix factorization," 2018, *arXiv:1802.04606*. [Online]. Available: <http://arxiv.org/abs/1802.04606>
- [35] S. Zhang, L. Yao, and X. Xu, "AutoSVD++ an efficient hybrid collaborative filtering model via contractive auto-encoders," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tokyo, Japan, 2017, pp. 957–960.
- [36] F. Qian et al., "HGAR: Hybrid granular algorithm for rating recommendation," in *Proc. Int. Joint Conf. Rough Sets*, Havana, Cuba, 2020, pp. 267–279.
- [37] Y. Zhang, C. Zhao, M. Chen, and M. Yuan, "Integrating stacked sparse auto-encoder into matrix factorization for rating prediction," *IEEE Access*, vol. 9, pp. 17641–17648, 2021.



DIAO ZHOU was born in 1994. He received the B.S. degree from the Swan College, Central South University of Forestry and Technology, China, in 2018. He is currently pursuing the master's degree with the North China University of Technology, China. His research interests include deep learning and recommender systems.



SHENGNAN HAO received the B.S. degree from the North China University of Science and Technology, China, in 1996, and the M.S. degree from the Beijing University of Technology, China, in 2009. She joined the North China University of Science and Technology in 1996 and became an Associate Professor in 2009. Her current research interests include complex systems, impulsive systems, and stochastic control.



HAIYANG ZHANG received the B.S. degree from the School of Software Engineering, Jilin University, China, in 2013, and the Ph.D. degree from the University of Limerick, Ireland, in 2018.

She is currently a Research Associate with the Department of Computer Science, University of Sheffield, U.K. Her current research interests include recommender systems, data mining, collaborative filtering, and natural language processing (NLP).



CHENXU DAI received the B.S. and M.S. degrees from the School of Computer Science and Technology, North China University, China. She is currently a Research Associate with the North China University. Her current research interests include neural networks, intelligent optimization algorithms, and deep learning.



YONGLI AN received the Ph.D. degree in information science from Beijing Jiaotong University, Beijing, China, in 2015. She is currently with the College of Information Engineering, North China University of Science and Technology, Tangshan, China. Her current research interests include multi-antenna processing technology, meta-material antenna, interference cancellation technology, and large-scale MIMO technology.



ZHANLIN JI (Member, IEEE) received the M.Eng. degree from Dublin City University, Ireland, in 2006, and the Ph.D. degree from the University of Limerick, in 2010. He is currently a Professor with the North China University of Science and Technology, China, and an Adjunct Researcher in the Telecommunications Research Centre (TRC), University of Limerick, Ireland. He has authored/coauthored more than 100 research papers in refereed journals and conferences. His research interests include Ubiquitous Consumer Wireless World (UCWW), the Internet of Things (IoT), cloud computing, big data management, and data mining.



IVAN GANCHEV (Senior Member, IEEE) received the degree (*summa cum laude*) in engineering and the Ph.D. degree (*summa cum laude*) from the Saint-Petersburg University of Telecommunications, in 1989 and 1995, respectively.

He is currently a Full Professor with the University of Plovdiv "Paisii Hilendarski," an International Telecommunications Union (ITU-T) Invited Expert, and an Institution of Engineering and Technology (IET) Invited Lecturer. He was involved in more than 40 international and national research projects. He has served on the TPC of more than 330 prestigious international conferences/symposia/workshops, and has authored/coauthored one monographic book, three textbooks, four edited books, and more than 280 research papers in refereed international journals, books, and conference proceedings. He is also on the editorial board of multiple international journals and served as a guest editor for multiple international journals.

• • •