IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 17, 2018, accepted August 20, 2018, date of publication August 31, 2018, date of current version September 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2868083

Wide and Deep Model of Multi-Source Information-Aware Recommender System

WEIHUA YUAN[1,2], HONG WANG[1], (Fellow, IEEE), BAOFANG HU[1], LUTONG WANG[1], AND QIAN WANG[1]

[1]School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China
[2]School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China

Corresponding author: Hong Wang (wanghong106@163.com)

This work was supported in part by the National Natural Science Fund under Grant 61672329, Grant 61373149, and Grant 61472233, in part by the Technology Program of Shandong Province under Grant 2014GGB01617, in part by the Science and Technology Program Project of Shandong Colleges and Universities under Grant J17KA070, in part by the Doctoral Foundation of Shandong Jianzhu University under Grant XNBS1523, in part by the Excellent Course Project of Shandong Province under Grant 2012BK294, Grant 2013BK399, and Grant 2013BK402, and in part by the Educational Science Planning Project of Shandong Province under Grant ZK1437B010.

ABSTRACT Collaborative filtering recommendation suffers from the problems of high data sparsity, poor expansibility, cold start, and the difficulty of modeling user preferences, among which data sparsity is the greatest issue. Although our previous work on matrix completion model, named low rank non-negative matrix factorization and completion algorithm (LR-NMFC) and stochastic sub-gradient based low rank matrix completion algorithm, could effectively alleviate the sparsity problem, they customarily model the linear feature interactions instead of the complex nonlinear structures between users and items when making recommendations. To better depict user preferences and item features, we deepen the linear model LR-NMFC to establish a wide and deep model, which we named Wide and Deep model of Multi-source information-Aware recommender system (WDMMA), based on multi-source information composed of user-item interaction matrix, attributes, and context. The wide part mainly handles the linear interactions between users and items, while the deep part portrays the high-order nonlinear interactions. We pre-train both the wide and the deep part using LR-NMFC in the embedding layer. In the pooling layer, we define a pooling operation, AC-pooling, which is used to model the various interactions among users, items, attributes, and context information. Upon the pooling layer, we stack some hidden layers to capture the high-order nonlinear feature interactions. Experiments on two public datasets show that WDMMA can learn complex nonlinear feature patterns successfully and effectively and is beneficial to improve the recommendation performance. Therefore, it is an effective way to consider both linear user-item interactions and multi-source information-aware nonlinear interactions in a deep learning framework when making recommendations.

INDEX TERMS Collaborative filtering recommendation, data sparsity, deep learning, high-order nonlinear interactions, multi-source information.

I. INTRODUCTION

In the era of information explosion, recommender systems have been widely applied to various online service systems to alleviate the problem of information overload. One popular example is the Amazon product recommender system, a typical representative of item-based collaborative filtering [1]. Social networks like Facebook,[1] Myspace[2] and LinkedIn[3] etc. employ collaborative filtering to their systems for recommendation of new friends, groups and other social connections. According to Yelling in 2013,[4] approximately 75-80% of movies watched on Netflix[5] come from recommendations rather than by search engines. With the renaissance of artificial intelligence,[6] the development of

[1]http://www.facebook.com
[2]https://myspace.com/
[3]https://www.linkedin.com/
[4]https://www.huffingtonpost.com/entry/netflix-profiles_n_3685876
[5]https://www.netflix.com/
[6]https://sites.google.com/view/lianghu/home

machine learning involves almost all the fields from natural language processing to computer vision to game theory. Deep learning, one of the most advancing machine learning technologies, becomes the preferred solution when building recommender systems, here a case in point is the technical report from Netflix.[7]

Collaborative filtering (CF) recommendation is currently the most popular method among content-based, collaborative filtering based and hybrid recommendation methods, which has the advantages of simplified modeling and low data dependency. However, it still suffers from the problems of high data sparsity, poor expansibility, cold start and the difficulty of modeling user preferences. The main cause of these problems stems from the insufficiency of data [2], and matrix completion has become a popular and effective solution [3], [4]. Our previous work on the matrix completion model, the LR-NMFC (Low Rank Non-negative Matrix Factorization and Completion Algorithm) [6] and SS-LRMC (Stochastic Sub-gradient based Low Rank Matrix Completion Algorithm) [7], could effectively alleviate the above issues.

Despite the effectiveness of LR-NMFC and SS-LRMC, they are still modeling the linear feature interactions between users and items. However, in the real world, these interactions are customarily built on complex nonlinear structures [8]. In addition to the user-item interaction matrix, user preferences are also affected by some rich auxiliary information [9]. For example, to enhance the accuracy of targeted advertising, Saia *et al.* [10] proposed a novel segmentation strategy by a semantic analysis on the description of the items, while Boratto *et al.* [11] take the time effect, that is, the aging of items into account when modeling user preferences. In a word, there are various kinds of hetergeneous multi-source information available in recommender systems [12]:

1) The attribute information of users and items, such as user's demographic information (age, occupation and gender, etc.), as well as item's attribute information, such as movie genre, leading actors and the semantic information of item descriptions.
2) The rich contextual information related to user-item interactions, such as location, time, peers and user emotions.

Obviously, if we fuse these data with implicit or explicit feedback in recommender systems, we can better model user preferences and item features. However, the greatest problem confronted here is the expression of these multi-source information, and in many complex scenarios, it becomes increasingly difficult when incorporating them into recommender systems. Furthermore, like the sparse user-item interactions, the feature vectors produced by these data are also high-dimensional and sparse, which further increase the difficulty of modeling and recommending with them.

[7]https://www.re-work.co/blog/deep-learning-tony-jebara-machine-learning-research-netflix

Since deep learning can obtain a unified embedding expression through automatic feature learning [13], we decide to build a deep learning based recommendation framework. And due to the high sparsity of feature vectors that represent the multi-source data mentioned above, we ought to consider the impact of different feature interactions when modeling user preferences. Wide&deep [14] extracts low-order and higher-order feature interactions from the data simultaneously, but its wide deep part requires feature engineering. DeepFM [15] applies multiple feature combinations for modeling CTR prediction. Based on the above analysis, we decide to deepen our linear model LR-NMFC and build a wide and deep model of multi-source information-aware recommender system, abbreviated as WDMMA. When modeling user preferences and item features, we consider the effect of linear interactions as well as attribute and context aware higher-order nonlinear interactions. The fusion of multi-source information enhances the expressiveness of our model; meanwhile, the extraction of both low-order linear features and high-order nonlinear features improves the recommendation accuracy. The main contributions are as follows:

1) To better depict user preferences and item features, we propose WDMMA, a wide and deep recommendation model based on multi-source information. Its structure is composed of GNMF (Generalized Non-negative Matrix Factorization) and AC-MLP (Attribute and Context-aware Multi-Layer Perceptron). The wide part GNMF mainly handles the linear interactions between users and items, while the deep part AC-MLP portrays the high-order nonlinear feature interactions based on multi-source information.
2) We pre-train both the wide and the deep part using LR-NMFC in the embedding layer. Upon the embedding layer of AC-MLP is the pooling layer. We define a pooling operation, AC-pooling, to model the various interactions among users, items, attributes and context information. Upon the pooling layer, we stack some hidden layers to capture the underlying high-order nonlinear feature interactions.
3) Experiments on two public datasets show that, the process of pre-training can accelerate algorithm convergence, as well as yield better recommendation performance. In addition, it also indicates empirically that WDMMA can learn complex nonlinear feature patterns successfully and effectively, considering both the linear interactions and the high-order multi-source information-aware nonlinear interactions.

II. RELATED WORK

A. SYMBOLS AND NOTATIONS

We provide here a brief summary of the symbols and notations used in the paper, as shown in Table 1.

TABLE 1. Symbols and notations.

Symbol	Meaning
$X, Y, M \in R^{m \times n}$	the rating matrix of m users on n items.
$y_{ui}, \hat{y}_{ui}$	the rating of user u on item i, and $\hat{y}_{ui}$ is the corresponding prediction rating.
$X^{(j)} (1 \leq j \leq n)$	the column vector of X about the rating list of item j.
X^T, $\hat{X}$, $\|X\|_F$	the transpose of X, the low rank reconstruction solution of X and the Frobenius norm of X, respectively.
Ω	an index set, which is a uniformly random sampling set indicating positions of the non-zeroes in X.
P_Ω	the projection selection operator of Ω
p_u^G, q_i^G	the embedding expression of user u and item i in GNMF part of WDMMA, respectively.
p_u^M, q_i^M	the embedding expression of user u and item i in AC-MLP part of WDMMA, respectively.
A_u, A_i	the corresponding attributes of user u and item i, respectively.
C_{ui}	the corresponding context of the current interaction between user u and item i.
$g_{A_u}, g_{A_i}, g_{C_{ui}}$	the corresponding embedding expressions of A_u, A_i and C_{ui}, respectively.
σ	the activation function of sigmoid, hyperbolic tangent (tanh) and Rectifier (ReLu) and among others.
χ	the set of training examples.
Θ	the set of parameters in the model.

B. REALTED WORK

Collaborative filtering recommendations are mainly divided into rating prediction and ranking prediction, also known as Top-N recommendation, based on their forms of output [16]. Recently, Top-N recommendation based on implicit feedback has been extensively studied. The latent factor models like MF [17], PLSA [18], and LDA [19] and so forth, become increasingly popular due to its outstanding performance in large-scale recommendation tasks. The key idea behind the MF model is to decompose the user-item interaction matrix into a product of low rank user and item submatrices in a shared hidden space. And therefore, a target user's prediction on an item is calculated through the inner product of the corresponding implicit user factor and item factor:

$$y_{ui} = p_u^T q_i = \sum_{l=1}^{k} p_{ul} q_{il} \tag{1}$$

Although the MF method is very effective in collaborative filtering, the inner product of Eq.(1) does not consider any high-order non-linear feature interactions between user and item hidden factors, implicitly assuming they are independent of each other [8]. Therefore, this form of representation limits the expressiveness of the model and is insufficient to capture the complex structure of user-item interactions.

To improve model performance, many researchers incorporated attributes or time information in their basic matrix decomposition models. For example, Agarwal and Chen [20] extended their model to handle user or item attributes, while Koren and Yehuda [21] and Xiong *et al.* [22] considered the temporal influences. Although the above work demonstrated empirically their superiority over basic MF models, they still belong to the category of shallow methods and model the interactive relationships linearly. In large datasets, these methods tend to increase the model complexity while not significantly improving the performance. Therefore, to model the second-order feature interactions, Rendle proposed a fast and effective context-aware recommendation system [12] based on FM (Factorization Machine) [23]. However, FM is a linear model which captures the low-order feature interactions and cannot handle the nonlinear relationships and deep features.

Deep neural networks (DNNs) can extract implicit structures and intrinsic models from training data at different levels of abstraction and are able to approximate any continuous function [24]. DNNs have been successfully applied in computer vision [25], speech recognition [26], and natural language processing [27] and so forth; as a result, recommender systems based on deep learning have attracted increasingly more attention among researchers [28]. Related work mainly includes FNN proposed by Zhang *et al.* [29], PNN network by Qu *et al.* [31], wide&deep by Cheng *et al.* [14], deepFM by Guo *et al.* [15], and NCF by He *et al.* [8].

In deep recommender systems, we can combine the user-item interactions with various attributes and contextual information to model user preferences and to alleviate the impact of the sparsity issue. These information is customarily encoded as binary vectors with one-hot encoding, which is high-dimensional and sparse. We ought to consider the feature combinations among them if we want to establish an effective recommendation model, and moreover, there indeed exist complex higher-order nonlinear feature interactions. For instance, company staffs become accustomed to downloading APPs that provide food delivery services during meal time, which shows that a feature combination of APP type, user age, occupation and time context can be used for CTR prediction. Among which, user age, occupation, and APP type belong to attribute information, while timestamp is the context of current action.

Jian *et al.* [30] proposed a recommendation model which coupled the deep neural network SADE with the CF model, timeSVD++ for rating prediction, to address the issue of cold start. Liao *et al.* [32] proposed the social network embedding model, and they suggested enhancing network embedding expression by various attribute information, in addition to structural similarities. Embedding is originating from word embedding [34], a distributed representation that is automatically learned from data. It is a fully connected layer with one-hot encoding vectors as the input, and the number of nodes in the mid layer as the vector dimensions. Each dimension of this low-dimensional and dense vector has its practical meaning, and their values can be continuously updated during the training of DNNs [34].

Guo *et al.* propose deepFM [15], which combines the linearity of FM with the non-linearity of DNN, to extract high-order and low-order feature combinations

simultaneously for CTR prediction. However, its FM part adopts shared input and embedding expressions with its deep part, which could limit the scalability of the model and hinder the further improvement of accuracy. NCF [8] allows its integrated components to learn different user and item embeddings, respectively, but it does not account for any impacts of the abovementioned multi-source information. According to wide&deep model [14], both high-order and low-order features can bring about additional performance improvements, in contrast to considering either one alone. Hence, it is an efficient way to consider the various feature interactions between these data. In this way, we can better capture user preferences and item features, and improve the performance of recommender systems.

In addition, due to the non-convexity of the objective function in deep learning, optimizations based on gradient descent can only find its local optimal solution. The influence of initialization on the convergence and performance of the model is far reaching. Therefore, we can obtain good feature expressions through reasonable pre-training to guide the learning and optimization in the direction of the minimum [33]. In light of this, many researchers try to use pre-training to enhance the performance of their deep recommendation models. For example, Zhang *et al.* pre-trained the embedding layer of FNN [29] through feature embeddings learned by FM [23]. And it indicates empirically that the pre-trained FNN can learn complex nonlinear modes between features effectively. He *et al.* also used the pre-training results of GMF and MLP in NCF [8] to initialize their NeuMF framework.

Inspired by this, we use the latent user and item factors produced by our matrix completion model LR-NMFC [5], [6] to pre-train WDMMA. LR-NMFC is a low-rank matrix completion model based on non-negative decomposition to solve the sparse problem of the rating matrix:

$$\begin{aligned} &\min_{U,V} \|P_\Omega(X) - P_\Omega(UV)\|_F^2 \\ &s.t.\ P_\Omega(X) = P_\Omega(M), \\ &\quad rank(X) \le r, U_{ij} \ge 0 \quad and\ V_{ij} \ge 0 \end{aligned} \tag{2}$$

where M represents the set of known observations of the original rating matrix, $U \in R^{m\times r}$ and $V \in R^{r\times n}$ are the decomposition sub-matrices, and P_Ω is defined in (3):

$$P_\Omega(X) = \begin{cases} X_{ij}, & if\ \ (i,j) \in \Omega \\ 0, & otherwise \end{cases} \tag{3}$$

The multiplicative update process of U and V are derived by coordinating descent method, as shown in (4):

$$\begin{aligned} U_{jk}^{(n+1)} &\leftarrow U_{jk}^{(n)} \frac{\left(P_\Omega(X)V^T\right)_{jk}}{\left(P_\Omega(UV)V^T\right)_{jk}}, \\ V_{ki}^{(n+1)} &\leftarrow V_{ki}^{(n)} \frac{\left(U^T P_\Omega(X)\right)_{ki}}{\left(U^T P_\Omega(UV)\right)_{ki}} \end{aligned} \tag{4}$$

During iterations of LR-NMFC, we obtain the low-rank reconstruction solution $\hat{X}$ to predict the missing values in original matrix X; besides, we obtain the corresponding non-negative decomposition satisfying:

$$\hat{X} = UV \tag{5}$$

It indicates emperically that combining LR-NMFC with existing collaborative filtering algorithms can significantly improve the prediction accuracy.

III. THE PROPOSED MODEL WDMMA

A. MODEL DESCRIPTION

In WDMMA, we can combine the user-item interaction matrix with various attributes and contextual information when modeling user preferences. Its framework is shown in Fig. 1, which is composed of two parts: the generalized non-negative matrix factorization part named GNMF, and the deep neural part, named AC-MLP. WDMMA models both the linear interactions of GNMF and the higher-order nonlinear feature interactions of AC-MLP and tries to generate better predictions unifying the strengths of the two parts. And moreover, different from deepFM [15], WDMMA allows GNMF and AC-MLP to learn their respective embeddings and combines these two components at the prediction layer.

B. GNMF

The input layer of GNMF contains the one-hot encoding vectors for users and items, respectively, which are not only high-dimensional but also extremely sparse. Upon the input layer is the embedding layer which, according to [34], maps the sparse input to a low dimensional dense vector. In addition, each dimension of this dense vector has its practical meaning. Therefore, vectors of the embedding layer can be used as features; for example, He *et al.* [8] treats them as user and item hidden feature vectors of the latent factor models.

In GNMF, the embeddings for user u and item i are represented as p_u^G and q_i^G, respectively. Then, we can calculate the element-wise inner product between p_u^G and q_i^G to capture the low-order linear feature interactions, as shown in (6):

$$Z_L^G = p_u^G \odot q_i^G \tag{6}$$

where Z_L^G is used to model the importance of first order feature interactions between user u and item i, and $\odot$ represents the element-wise product between vectors. The predicted score $\hat{y}_{ui}^G$ can be computed according to (7):

$$\hat{y}_{ui}^G = \sigma\left(h^T Z_L^G\right) \tag{7}$$

among which σ is the non-linear activation function of Relu or sigmoid, and h is the weight which is learned from data.

Based on (5), we obtain the non-negative decomposition representation of the reconstruction matrix, where $U^{(i)}$ and $V^{(j)}$ represent the distributions of user interest and item features, respectively. Therefore, we pre-train WDMMA with normalized $U^{(i)}$ and $V^{(j)}$, separately assigning them to p_u^G and q_i^G. According to (6) and (7), we can generalize and deepen our LR-NMFC to a non-linear context.

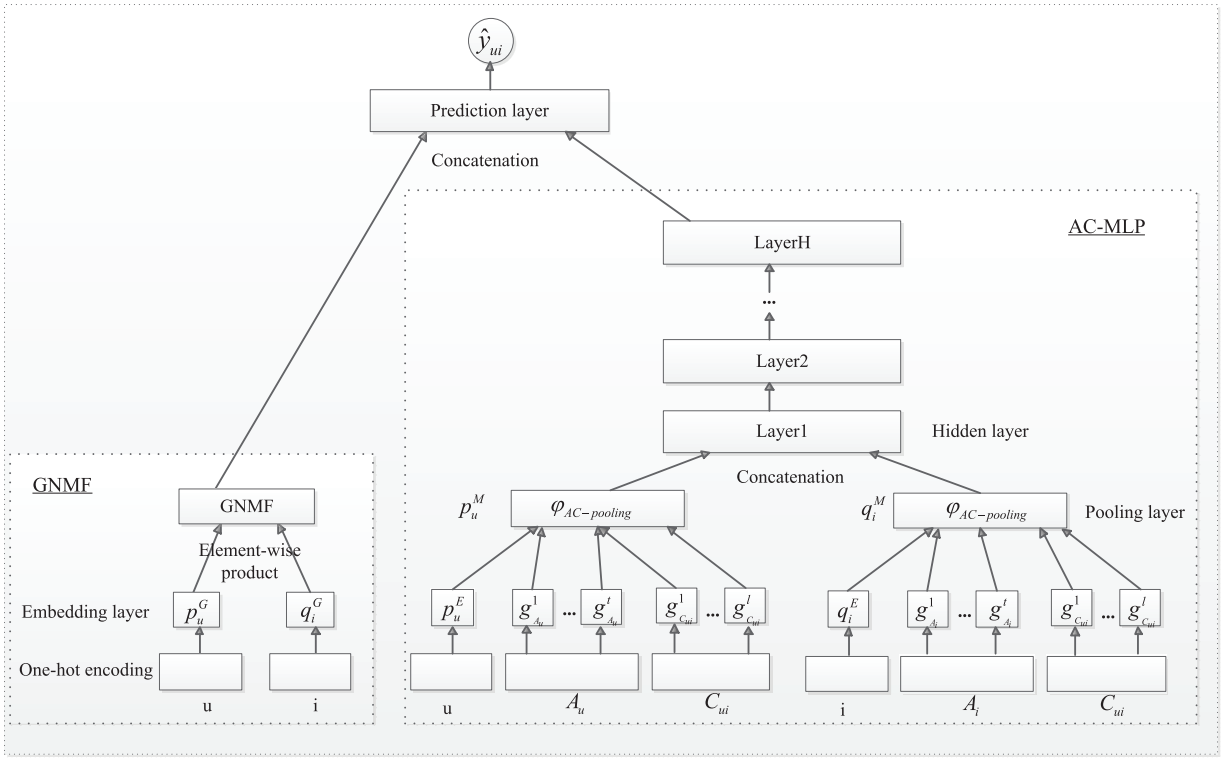

FIGURE 1. The structure of WDMMA

C. AC-MLP

1) THE INPUT LAYER AND EMBEDDING LAYER

In AC-MLP section, the input layer includes the one-hot encoding vectors of user u, his attributes A_u, item i, its attributes A_i, as well as the corresponding context of the current interaction C_{ui}, where A_u, A_i and C_{ui} might include either categorical or continuous variables. Upon the input layer is the embedding layer, which converts these sparse vectors into low-dimensional and dense expressions, represented as p_u^E, g_{A_u}, q_i^E, g_{A_i} and $g_{C_{ui}}$, respectively. We still pre-train p_u^E and q_i^E with normalized $U^{(i)}$ and $V^{(j)}$ as we do in GNMF.

Since only the non-zero binary feature vectors of g_{A_u}, g_{A_i} and $g_{C_{ui}}$ are stored, different user-item interactions might contain different numbers of non-zero attributes and context variables. As a result, the input of AC-MLP is a set of embedding vectors with varying length. DNNs traditionally require a fixed-length vector as input, and consequently, the embedding vectors should be preprocessed before they are fed into the hidden layers. To achieve this, these vectors ought to be converted into a fixed-length vector, and the important low-rank interactive information between them should be captured effectively.

In CNNs (conventional neural networks) [35], convolution and pooling are its two crucial operations. The pooling layer conducts dimensionality reduction on features of its convolution layer to form the final features, and above the pooling layer we stack some fully connected layers. Average and maximum pooling are two commonly used pooling operations, suitable for feature extraction in dense data representations as images and videos. However, the feature vectors generated by the heterogeneous multi-source data here are highly sparse and, consequently, the two pooling operations are no longer applicable to capture the important user-item interactions. To address this issue, He *et al.* proposed BI-Interaction pooling operation [36] and Wang proposed a pairwise-pooling operation [37]. These pooling operations are capable of capturing the interactions between user/item and its attributes with low computation complexity, making it convenient for the following deep layers to learn meaningful information. Inspired by this, to guarantee that the fully connected layers learn meaningful high-order nonlinear feature interactions, we define the AC-pooling operation, which is used to capture different interactions between users, items, attributes and contextual information.

2) POOLING LAYER

In AC-MLP, we defined three types of AC-pooling operations for user u, expressed as:

- $\varphi_{AC-pooling}\left(p_u^E, \left\{g^t_{A_u}\right\}\right)$: to capture pairwise correlations between users and their attributes, as well as all nested correlations between these attributes;

$$\varphi_{AC-pooling}\left(p_u^E, \{g^t_{C_{ui}}\}\right) = \sum_{t=1}^{|C_{ui}|} p_u^E \odot g^t_{C_{ui}} + \sum_{t=1}^{|C_{ui}|} \sum_{t'=t+1}^{|C_{ui}|} g^t_{C_{ui}} \odot g^{t'}_{C_{ui}} \quad (8)$$

- $\varphi_{AC-pooling}\left(p_u^E, \left\{g_{C_{ui}}^l\right\}\right)$: to capture pairwise correlation between users and context and all nested correlations between the context variables;

$$\varphi_{AC-pooling}\left(p_u^E, \left\{g_{C_{ui}}^l\right\}\right) = \sum_{l=1}^{|C_{ui}|} p_u^E \odot g_{C_{ui}}^l + \sum_{l=1}^{|C_{ui}|} \sum_{l'=l+1}^{|C_{ui}|} g_{C_{ui}}^l \odot g_{C_{ui}}^{l'} \quad (9)$$

- $\varphi_{AC-pooling}\left(\left\{g_{A_u}^t\right\}, \left\{g_{C_{ui}}^l\right\}\right)$: to capture pairwise correlations between attributes and context variables.

$$\varphi_{AC-pooling}\left(\left\{g_{A_u}^t\right\}, \left\{g_{C_{ui}}^l\right\}\right) = \sum_{t=1}^{|A_u|} \sum_{l=1}^{|C_{ui}|} g_{A_u}^t \odot g_{C_{ui}}^l \quad (10)$$

Based on the pooling operations of (8), (9) and (10), the embedding expression p_u^M of user u in AC-MLP can be described as:

$$p_u^M = \varphi_{AC-pooling}\left(p_u^E, \left\{g_{A_u}^t\right\}\right) + \varphi_{AC-pooling}\left(p_u^E, \left\{g_{C_{ui}}^l\right\}\right) + \varphi_{AC-pooling}\left(\left\{g_{A_u}^t\right\}, \left\{g_{C_{ui}}^l\right\}\right) \quad (11)$$

Similarly, the three pooling operations for item i are defined as:

- $\varphi_{AC-pooling}\left(q_i^E, \left\{g_{A_i}^t\right\}\right)$: to capture pairwise correlations between items and their attributes, as well as all nested correlations between these attributes;

$$\varphi_{AC-pooling}\left(q_i^E, \left\{g_{A_i}^t\right\}\right) = \sum_{t=1}^{|A_i|} q_i^E \odot g_{A_i}^t + \sum_{t=1}^{|A_i|} \sum_{t'=t+1}^{|A_i|} g_{A_i}^t \odot g_{A_i}^{t'} \quad (12)$$

- $\varphi_{AC-pooling}\left(q_i^E, \left\{g_{C_{ui}}^l\right\}\right)$: to capture pairwise correlation between items and context and all nested correlations between the context variables;

$$\varphi_{AC-pooling}\left(q_i^E, \left\{g_{C_{ui}}^l\right\}\right) = \sum_{l=1}^{|C_{ui}|} q_i^E \odot g_{C_{ui}}^l + \sum_{l=1}^{|C_{ui}|} \sum_{l'=l+1}^{|C_{ui}|} g_{C_{ui}}^l \odot g_{C_{ui}}^{l'} \quad (13)$$

- $\varphi_{AC-pooling}\left(\left\{g_{A_i}^t\right\}, \left\{g_{C_{ui}}^l\right\}\right)$: to capture pairwise correlations between attributes and context variables.

$$\varphi_{AC-pooling}\left(\left\{g_{A_i}^t\right\}, \left\{g_{C_{ui}}^l\right\}\right) = \sum_{t=1}^{|A_i|} g_{A_i}^t \odot \sum_{l=1}^{|C_{ui}|} g_{C_{ui}}^l \quad (14)$$

And then, the embedding expression q_i^M of item i in AC-MLP is represented as:

$$q_i^M = \varphi_{AC-pooling}\left(q_i^E, \left\{g_{A_i}^t\right\}\right) + \varphi_{AC-pooling}\left(q_i^E, \left\{g_{C_{ui}}^l\right\}\right) + \varphi_{AC-pooling}\left(\left\{g_{A_i}^t\right\}, \left\{g_{C_{ui}}^l\right\}\right) \quad (15)$$

By analysis of expressions in Eq.(8)-Eq.(15), we know that the AC-pooling operations encode the second-order feature interactions of users, items, attributes and context variables, and its output is a k-dimensional vector.

3) HIDDEN LAYERS

Above the pooling layer of AC-MLP, we stack some fully connected layers. We send p_u^M and q_i^M to the hidden layers to capture the higher-order nonlinear interacting features between the abovementioned multi-source data, which are defined as (16):

$$\begin{aligned} Z_1^M &= \phi_c\left(p_u^M, q_i^M\right) = \begin{bmatrix} p_u^M \\ q_i^M \end{bmatrix} \\ Z_2^M &= \sigma_2\left(W_2 Z_1^M + b_2\right) \\ Z_3^M &= \sigma_3\left(W_3 Z_2^M + b_3\right) \\ &\cdots \\ Z_H^M &= \sigma_H\left(W_H Z_{H-1}^M + b_H\right) \end{aligned} \quad (16)$$

where W_l, b_l, σ_l and Z_l^M represent weight matrix, bias vector, activation function and output vector of layer l, respectively, and for the activation function, we usually choose ReLu from sigmoid, hyperbolic tangent (tanh) and Rectifier (ReLu) and among others. During implementation, we use a tower network structure, and use fewer hidden units in higher layers so that the model can learn more attractive features [31]. And the predicted score y_{ui}^M, based solely on AC-MLP can be computed as:

$$\hat{y}_{ui}^M = \sigma\left(h^T Z_H^M\right) \quad (17)$$

D. MODEL PREDICTION AND OPTIMIZATION

To obtain the output of the prediction layer, we should leverage both the linear and the non-linear feature interactions, therefore the prediction list cannot be generated in light of (7) or (17) alone. Instead, we first conduct concatenation of the output vectors, that is, Z_L^G of GNMF with Z_H^M, the last hidden layer of AC-MLP, and then multiply the neural weight vector h of the prediction layer, to obtain prediction score $\hat{y}_{ui}$:

$$\hat{y}_{ui} = \sigma\left(h^T \begin{pmatrix} Z_L^G \\ Z_H^M \end{pmatrix}\right) \quad (18)$$

The proposed model WDMMA focuses on implicit feedback, and user's ratings on items are either 0 or 1. Since implicit feedback only provides a noise signal on the user's preference, 1 means that the user likes the item, and 0 indicates that he does not like the item or does not know the item at all [38]. Therefore, there is a natural loss of negative feedback in implicit datasets; to deal with it we can either treat all unobserved items as negative feedbacks or conduct negative sampling from the unobserved dataset [39]. Here, in WDMMA, we choose to extract negative items at a certain sampling rate according to the number of observations.

Since WDMMA fuses user-item interactions with various attributes and contextual information to realize predictions,

TABLE 2. Attributes of Movielens-1M.

Attributes	userID	gender	age	occupation	moiveID	genres	day-of-the-week	hour-of-the-day
Counts	6,040	2	7	21	3,900	18	7	24

we regard the whole predicting process as a regression problem, and its loss function is defined as:

$$L\left(y_{ui}, \hat{y}_{ui}\right) = \sum_{(u,i)\in\chi} \left(y_{ui} - \hat{y}_{ui}\right)^2 + \lambda/2\|\Theta\|^2 \qquad (19)$$

where y_{ui} is the true value, and $\hat{y}_{ui}$ the predicted value, Θ the set of parameters, and λ the regularization coefficient to prevent the model from over-fitting. χ is the set of training examples, consisting of both positive and negative feedbacks. We use mini-batch Adagrad [36], whose learning rate can be self-adjusted according to the training phase, as the optimizer to iteratively update all parameters until convergence.

The process of WDMMA is summarized as Algorithm 1.

Algorithm 1 Wide and Deep Model of Multi-Source Information-Aware Recommender System (WDMMA)

Require:
Embedding dimension m, batch size n;
Step size η, regularization parameter λ;
Maximum iterations T, user-item interaction matrix X;
One-hot encodings of u, A_u, i, A_i, and C_{ui};

Ensure:
Top-N recommendation list;

Initializing the weight matrices and bias vectors defined in Eq.(16) and Eq.(18) , with small random values;
Running LR-NMFC according to Eq.(4) to obtain normalized $U^{(i)}$ and $V^{(i)}$;
Pre-training p_u^G, p_u^E, and q_i^G, q_i^E with normalized $U^{(i)}$ and $V^{(j)}$, respectively;
for $i = 1$ to T **do**
  Dividing the dataset into multiple batches $\{\chi_L\}$;
  for $\chi_l \in \{\chi_L\}$ **do**
    Running GNMF part according to Eq.(6);
    Obtaining p_u^M and q_i^M based on Eq.(8)-Eq.(15);
    Feeding p_u^M and q_i^M to the fully connected layers based on Eq.(16) ;
    Obtaining predicting scores according to Eq.(18);
    Computing loss by Eq.(19);
    Updating the parameters using mini-batch Adagrad;
  end for
end for
return Top-N recommendation list according to $\hat{y}_{ui}$ in the testing set.

IV. EXPERIMENTAL RESULTS AND THE ANALYSIS

In this section, we first introduce the experimental setup and datasets. Then, we verify the performance of WDMMA by comparison with several state-of-the-art recommendation algorithms.

A. DATASETS

In the experiments, our datasets include MovieLens-1M [8] and Frappe.[9] Movielens-1M contains 1,000,209 ratings made by 6,040 users on 3,900 movies, as well as the corresponding timestamps, and each user should rate at least 20 movies. The ratings are integers ranging from 1 to 5, which we convert into implicit feedback with value 1 indicating one interaction between users and items. We extract two kinds of context from MovieLens, that is, day-of-week and hour-of-the-day. In addition, we also pick some user and movie attributes available in the datasets, as shown in Table 2. When conducting negative sampling for a certain user, we randomly select items that have no interaction with him under the same context, at a ratio of 1:2 according to the interactions in his behavioral history.

Since movie genres have multi-values here, we preprocess the genres column and retain its first value only. Each user-item interaction with the corresponding attributes and context information is converted into a feature vector by one-hot encoding, and we obtain 10,019 features in total. Fig. 2 demonstrates some related behavioral characteristics through statistical analysis of movie genres, gender, age and occupation.

The Frappe dataset contains 96,203 app usage logs in different contexts. In addition to userID and appID, each log contains several context variables, including cnt, daytime, weekday, isweekend, cost, home/work, weather, country, and city, etc., as shown in Table 3. We also convert each log into a feature vector using one-hot encoding, which yields a total of 5,382 features. We assign a value of 1, meaning that the user used this app in this context. We adopt the same negative sampling strategy as we do in Movielens dataset. Fig. 3 and Fig. 4 show the statistical behavioral characteristics related to attributes such as daytime, weather and country, respectively.

B. EVALUATION INDEX AND SETTING OF HYPER-PARAMETERS

We use RMSE to evaluate the performance of the algorithms; the smaller the RMSE, the better the algorithm performance:

$$RMSE = \sqrt{\frac{1}{|TEST|}\sum_{(u,v,R_{u,v})}\left(R_{u,v} - estimated\right)^2} \qquad (20)$$

In addition, we also conduct performance comparison with some state-of-the-art algorithms based on AUC(Area under the ROC curve): the larger the AUC, the better the algorithm performance.

For each user's behavior sequence, we take 70% as the training set, 10% as the validation set, and the remaining 20%

[8] https://grouplens.org/datasets/movielens/
[9] http://baltrunas.info/research-menu/frappe

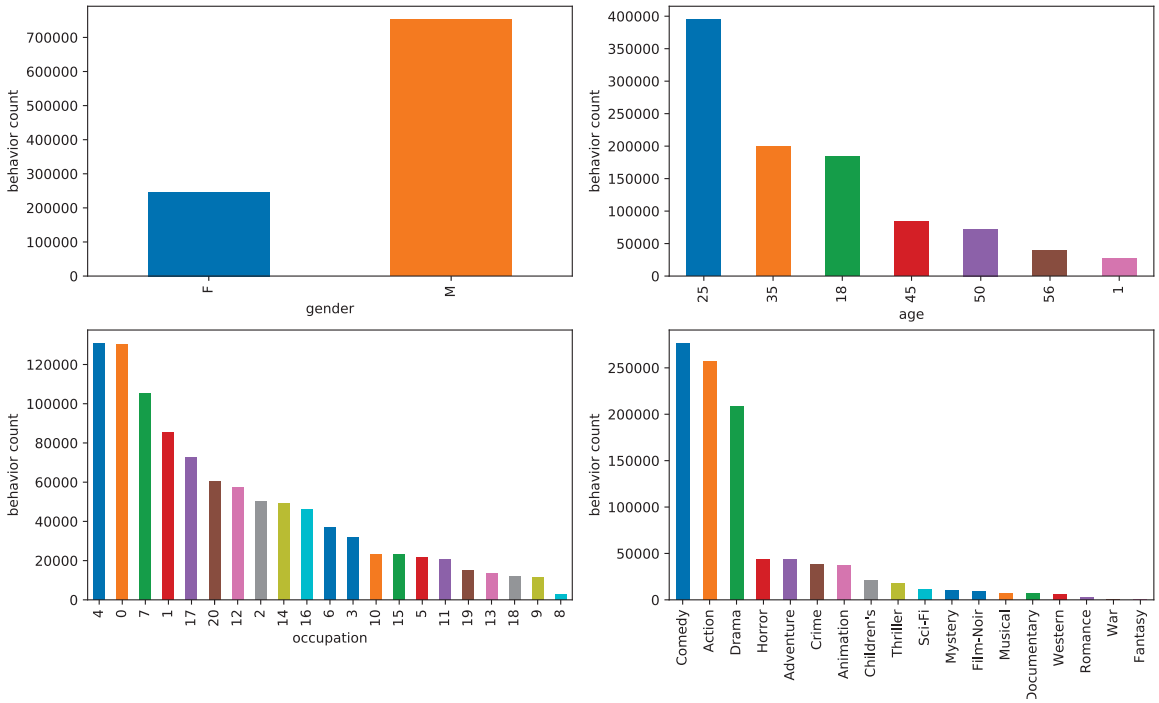

FIGURE 2. Behavior characteristics related to genres, gender, age and occupation in Movielens-1M

TABLE 3. Attributes of Frappe dataset.

Attributes	userID	appID	daytime	weekday	isweekend	cost	home/work	weather	country
Counts	957	4082	4	7	2	2	3	9	80

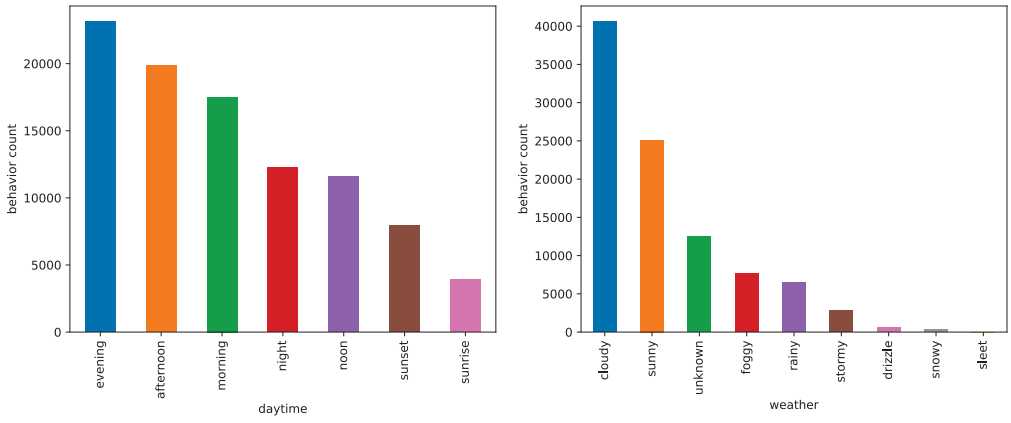

FIGURE 3. Behavior characteristics related to daytime and weather in Frappe

for testing. All the numerical results reported in this section are averaged over five runs. The training set is used to train the parameters of the model; the validation set is for tuning various hyper-parameters, for example, the exploration on the properties of WDMMA in section IV (subsection C, D and E) is conducted on the validation set; while the testing set is for performance comparison with some state-of-the-art algorithms. The range of parameters involved in the algorithm is as follows:

1) The range of the embedding dimension m is [16,32,64,128,256];
2) The range of the regularization coefficient λ is $[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1]$;

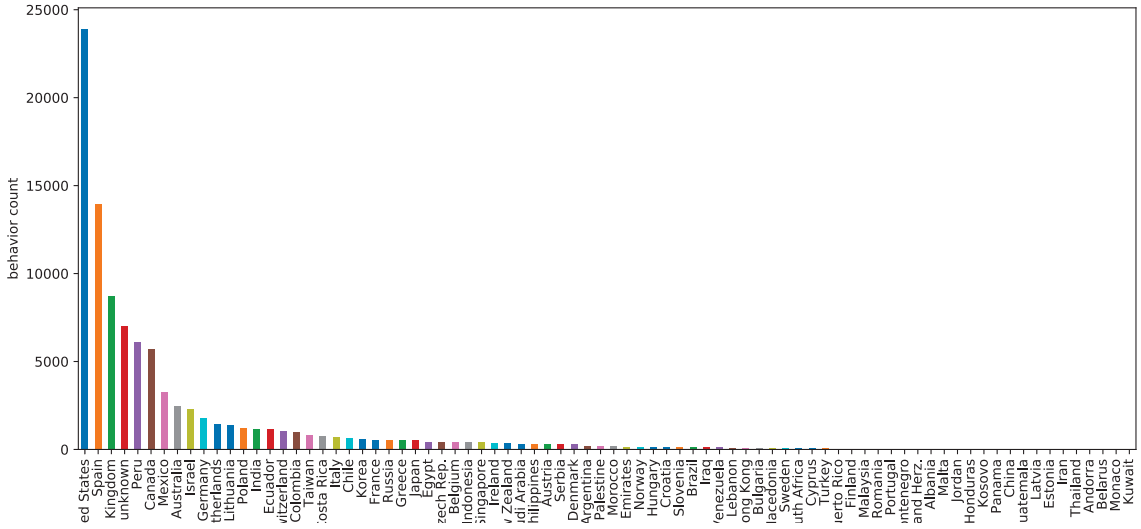

FIGURE 4. Behavior characteristics related to country in Frappe

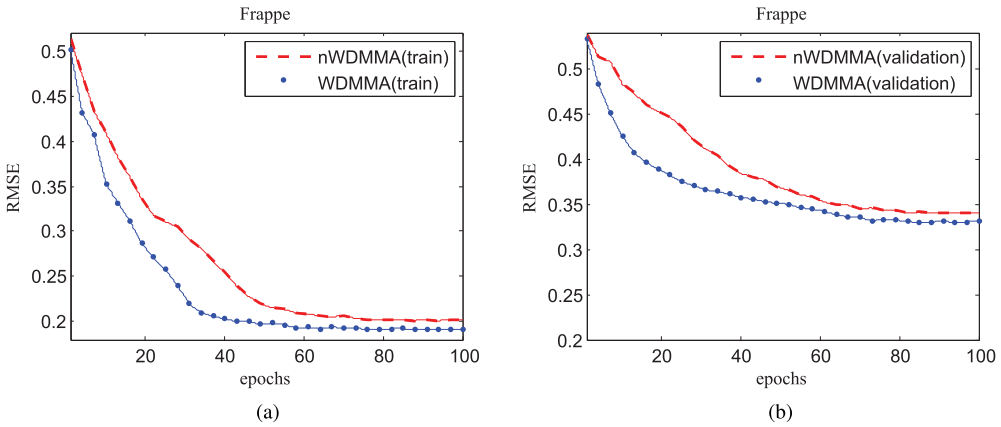

FIGURE 5. Effectiveness of the pre-training operation of WDMMA on Frappe dataset. (a) Effectiveness of the pre-training operation on the training set of Frappe. (b) Effectiveness of the pre-training operation on the validation set of Frappe.

3) The step size η of Adagrad is set in the range $\left[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\right]$;
4) The default value of batch size n is set to 128, which can be adjusted according to the size of the training set. Its value should be balanced between the training time and the convergence rate. The larger the value of n is, the faster the training, but the slower the convergence.

C. EFFECTIVENESS ANALYSIS AND VERIFICATION OF PRE-TRAINING OPERATION

Due to the nonlinearity of neural networks, the model easily falls into the local optimum when using SGD for optimization; therefore, we use pre-training to accelerate the training process. First, we run our LR-NMFC on the two datasets and then pre-train user and item embeddings with the corresponding normalized $U^{(i)}$ and $V^{(j)}$; for comparison the proposed algorithm without pre-training is represented as nWDMMA (here n means no pre-training). The comparable results of WDMMA and nWDMMA on Frappe and MovieLens are shown in Fig. 5 and Fig. 6, respectively. Among which the subgraphs of Figs. 5(a) and 6(a) are the comparisons of loss on the training sets between WDMMA and nWDMMA, while Figs. 5(b) and 6(b) are their comparisons on the validation sets.

From Figs. 5 and 6, we can see that WDMMA that is pre-trained by LR-NMFC converges much faster than nWDMMA. On the Frappe dataset, WDMMA tends to converge after approximately 30 epochs in Fig. 5(a), with the performance comparable to that of nWMDMA trained from scratch after approximately 60 epochs. As can be clearly seen from Fig. 5(b), the variation and comparison tendencies are almost the same on the validation set as they are on the training set.

Fig. 6(a) and Fig. 6(b) illustrate the comparison results between WDMMA and nWDMMA on Movielens-1M.

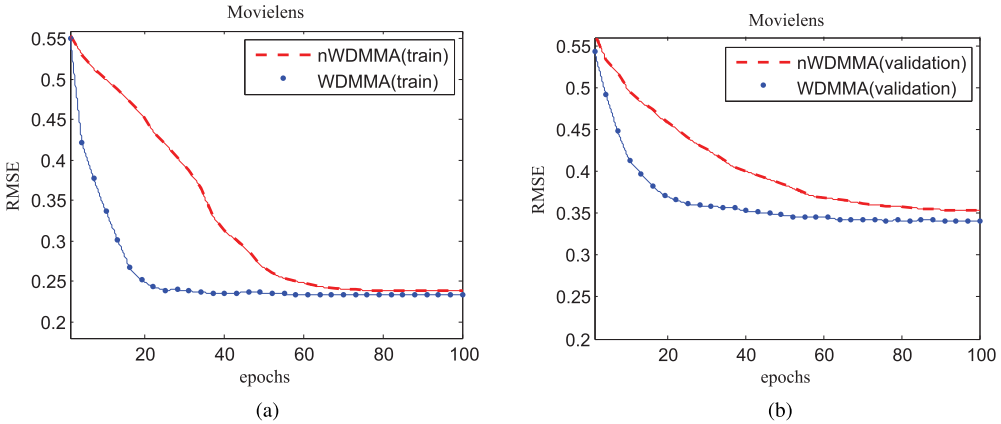

FIGURE 6. Effectiveness of the pre-training operation of WDMMA on Movielens-1M Dataset. (a) Effectiveness of the pre-training operation on the training set of Movielens-1M. (b) Effectiveness of the pre-training operation on the validation set of Movielens-1M.

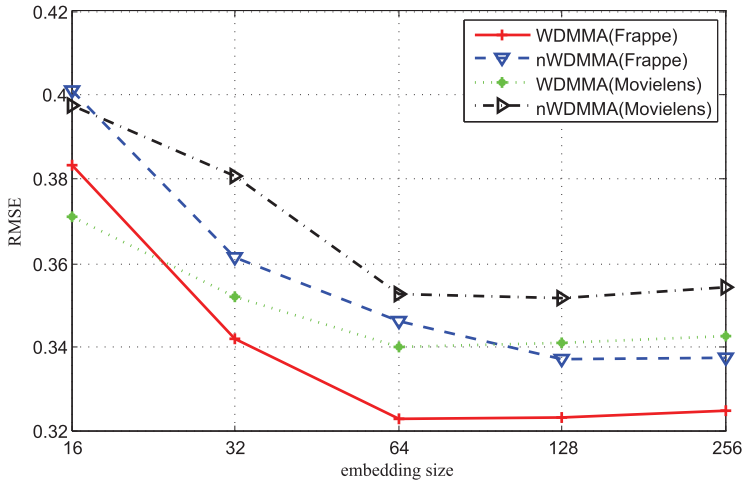

FIGURE 7. Performance comparison of WDMMA and nWDMMA for different embedding size *m*

As shown in the figures, the proposed algorithm with pre-training achieves better performance, for example, in Fig. 6(a) WDMMA is apt to converge after only about 20 epochs. In addition, pre-training does enhance the final performance of the algorithm, as it can be concluded from Figs. 5 and 6 that, the RMSE of WDMMA is lower than that of nWDMMA on both the training and validation sets.

D. PERFORMANCE COMPARISON OF WDMMA AND nWDMMA FOR DIFFERENT EMBEDDING SIZE m

The performance variations and comparison of WDMMA and nWDMMA with different number of embedding size m are depicted in Fig. 7. The figure demonstrates that WDMMA is superior to nWDMMA under different m, which further validates the effectiveness of the pre-training step by LR-NMFC. With the increase of m, the performance of each algorithm is gradually increasing, which means that increasing the embedding size can further enhance the expressiveness of the model. On Frappe, the relative improvement of WDMMA is 6.6% and 4%, in contrast to nWDMMA, when $m = 64$ or 128. Fig. 7 also indicates that both WDMMA and nWDMMA obtains optimal performance with $m = 64$, which is set as the default value of the embedding size in the subsequent experiments.

E. VALIDATION ANALYSIS OF THE AC-POOLING OPERATION

The AC-pooling operation is defined to capture various interactions between users, items, attributes and contextual variables, as shown in Eq.(8)-Eq.(15). To confirm whether this pooling operation has effectively caught related interactions, as well as its influence on the following hidden layers, we stack 0-5 hidden layers above the pooling layer, and represented the model as WDMMA-0 to WDMMA-5, respectively. Table 4 is their RMSE comparison on the validation sets.

It can be clearly seen from the table that WDMMA-0 has the worst performance on both the Frappe and Movielens

TABLE 4. Performance of WDMMA with different number of hidden layers.

Frappe	WDMMA-0	WDMMA-1	WDMMA-2	WDMMA-3	WDMMA-4	WDMMA-5
RMSE	0.3885	0.331	**0.3229**	**0.3237**	0.3341	0.3367
Movielens	WDMMA-0	WDMMA-1	WDMMA-2	WDMMA-3	WDMMA-4	WDMMA-5
RMSE	0.4036	0.3574	**0.3403**	**0.3421**	0.3465	0.3512

datasets. This is principally because without hidden layers, it directly maps the user and item embeddings to the output layer, which just captures the low-order feature interactions between users and items, attributes and context variables. The relative improvement of WDMMA-1 with one hidden layer is 16.9% and 11%, when compared with WDMMA-0, which illustrates the importance of capturing high-order feature interactions through DNNs.

By comparative analysis of the models WDMMA-1 to WDMMA-5, we find that the performance has no apparent and further improvement with the increasing number of hidden layers. Based on the result of Table 4, WDMMA-2 and WDMMA-3 perform the best. This strongly proved the effectiveness of the underlying AC-Pooling operation, since the model are capable of learning higher-order feature interactions effectively with only two or three hidden layers. In other words, the algorithm can achieve its best performance with only two hidden layers. This simplifies model optimization a lot and can avoid or relieve the underlying problem [8] of vanishing/exploding gradients, overfitting, etc. Based on the above analysis and results, we set two hidden layers for WDMMA in the subsequent experiments.

F. PERFORMANCE COMPARISON WITH SEVERAL STATE-OF-THE-ART ALGORITHMS

When compared with state-of-the-art methods, we use WDMMA-2 with two hidden layers and set embedding size $m = 64$; for the activation function, we choose ReLu. In wide&deep, we use the same network structure as proposed in [14]. The size of the hidden factor is set to 64 for NCF [8], FM [12] and LR-NMFC [6].

1) ItemKNN+LR-NMFC:
   For the implicit user-item interaction matrix we first run LR-NMFC to fill in the missing values. In the whole process all missing items are assigned to 0 without negative sampling, to verify the feasibility and validity of deep expansion for linear methods. Then the recommendation list is generated based on itemKNN [40] from the reconstructed matrix.
2) FM [12] and CARs [9]:
   Rendle *et al.* propsed a fast context-aware recommendation method [12], which we call it FM for short, based on factorization machines; while CARs is a context-aware recommendation system proposed by Adomavicis *et al.* The recommendation list is obtained based on user-item interactions and the related context. The algorithms here are used to verify the effectiveness of the proposed algorithm in context-aware recommendations.
3) Wide&deep [14] and NCF [8]:
   In wide&deep, we use original features for the wide part, and concatenate the feature embeddings for the deep part to model their interactions. We use a three-layer tower network structure as proposed in [14] :1024, 512 and 256. We use the NeuMF model proposed by He *et al.* in their NCF framework and represent the algorithm as NCF. In addition, in NCF the predictions are made purely based on user-item interactions without considering any attributes or context. Both wide&deep and NCF are used to verify the effectiveness of WDMMA in deep recommendations.

TABLE 5. The RMSE comparison with state-of-the-art algorithms.

	Frappe (RMSE)	MovieLens(RMSE)
itemKNN+LR-NMFC	0.4062	0.4837
wide&deep	0.3508	0.3872
FM	0.3634	0.4029
NCF	0.3589	0.3914
CARs	0.4027	0.4851
nWDMMA	0.3462	0.3527
WDMMA	**0.3229***	**0.3403***

Table 5 is the RMSE comparison of the above algorithms on Frappe and MovieLens, and here * means the statistical significance for $p < 0.05$ when comparing with the best baseline. It has been shown that WDMMA obtains the best performance on both datasets.

On Frappe, the RMSE of ItemKNN+LR-NMFC is slightly worse than that of CARs, since its recommendations are made based on the reconstructed user-item matrix of implicit feedback without any attributes or context variables. Thus it can be clearly seen that:

1) Algorithms of matrix completion perform worse on datasets with implicit feedback than on datasets with explicit feedback. One reason might be that in implicit feedback users' ratings on items are either 0 or 1, which cannot reflect the subjective differences of their rating behaviors. However, these differences tend to play a vital important role in low rank matrix completion.
2) The Frappe dataset has abundant context variables related to the usage of APP, and thus it follows that this multi-source information has a positive effect on modeling of user interest and item features. Meanwhile, on MovieLens, the performance of ItemKNN+LR-NMFC is slightly superior to CARs, because we only extract two kinds of context information here: day-of-week and hour-of-the-day, and recommendations are

generated mainly based on the user-item interaction matrix.

On the two datasets, the other algorithms have better performance than both ItemKNN+LR-NMFC and CARs, which further indicates the effectiveness of fusing attributes and context information in the modeling of recommender systems. The RMSE of wide&deep is lower than FM, and the performance of NCF also betters FM a lot. By analysis, we find that FM only models the second linear feature interactions between the context variables and user-item interactions; and we contribute the performance enhancement of NCF to the application of its MLP part, which learns high-order nonlinear feature interactions between users and items, instead of the conventional inner product between their embeddings. The superior performance of wide&deep indicates the effectiveness of considering the higher-order nonlinear feature combinations of attributes and context information.

WDMMA achieves the best performance on the two datasets, significantly outperforming wide&deep by a large margin. This highly encouraging result is mainly attributed to two aspects. One is that we pre-train the model using LR-NMFC, and the other is that, different from the simple concatenation operations, in WDMMA, we define three AC-pooling operations for users and items, respectively. The operation not only captures the interactions of embeddings at the lower layer but also helps to ease the following fully connected layers to learn more meaningful, high-order and nonlinear feature interactions.

Besides the comparison of RMSE, we also conduct extensive experiments to compare the AUC between these algorithms and the results are shown in Table 6. As shown in Table 6, our proposed WDMMA and nWDMMA achieves a higher AUC compared with the other algorithms, which further confirm the effectiveness of the proposed method.

TABLE 6. The AUC comparison with state-of-the-art algorithms.

	Frappe(AUC)	MovieLens(AUC)
itemKNN+LR-NMFC	0.6807	0.6750
wide&deep	0.7313	0.7106
FM	0.7060	0.6905
NCF	0.7250	0.7070
CARs	0.6804	0.6620
nWDMMA	0.7318	0.7121
WDMMA	**0.7369***	**0.7163***

V. CONCLUSIONS

In this paper, we deepen our linear model LR-NMFC and build a wide and deep model fusing a multi-source information-aware learning framework named WDMMA. We treat recommendation as a regression problem and pre-train the proposed model with embeddings learned from LR-NMFC. WDMMA considers both the linear interactions and the high-order nonlinear interactions fusing heterogeneous multi-source data between users and items. For one thing, the model improves recommendation accuracy by extracting low-order linear features and high-order nonlinear features concurrently; for another, it improves model expressiveness by fusing the multi-source data when depicting user and item expressions. Experiments on two public datasets indicate that in deep learning framework, WDMMA can effectively learn complex nonlinear feature interactions, and the pre-training with LR-NMFC not only speed up the algorithm convergence but also obtains better recommendation performance. Therefore, in recommender systems, it is an effective way to consider both linear user-item interactions and multi-source information-aware nonlinear interactions in deep learning framework. In future, we will continue our study on deep learning based recommendation, for example, the attention-based deep recommendation based on these hetergeneous multi-source information.

REFERENCES

[1] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003, doi: 10.1109/MIC.2003.1167344.

[2] L. Hu, L. Cao, J. Cao, Z. Gu, G. Xu, and D. Yang, "Learning informative priors from heterogeneous domains to improve recommendation in cold-start user domains," *ACM Trans. Inf. Syst.*, vol. 35, no. 2, p. 13, 2016, doi: 10.1145/2976737.

[3] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010, doi: 10.1137/080738970.

[4] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010, doi: 10.1109/TIT.2010.2046205.

[5] W. Yuan and H. Wang, "Using two-stage non-negative matrix factorization for topic recommendation in online social networks," *Comput. Model. Technol.*, vol. 18, no. 1, pp. 93–99, Jan. 2014.

[6] W. Yuan, H. Wang, and X. Du, "Collaborative filtering algorithm integrating non-negative matrix completion and subgroups partitioning," *J. China Mini-Micro Comput. Syst.*, vol. 38, no. 12, pp. 2645–2651, Dec. 2017, doi: 10.3969/j.issn.1000-1220.2017.12.005.

[7] W. Yuan, H. Wang, B. Hu, and Q. Sun, "A stochastic sub-gradient method for low rank matrix completion of collaborative recommendation," *Int. J. Performability Eng.*, vol. 13, no. 5, pp. 643–656, Sep. 2017, doi: 10.23940/ijpe.17.05.p9.643656.

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in *Proc. WWW*, Perth, WA, Australia, Apr. 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[9] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Proc. ACM Conf. Recommender Syst.*, New York, NY, USA, Oct. 2008, pp. 335–336, doi: 10.1145/1454008.1454068.

[10] R. Saia, L. Boratto, and S. Carta, "A latent semantic pattern recognition strategy for an untrivial targeted advertising," in *Proc. IEEE Int. Congr. Big Data*, New York, NY, USA, Aug. 2015, pp. 491–498, doi: 10.1109/BigDataCongress.2015.78.

[11] L. Boratto, S. Carta, G. Fenu, and R. Saia, "Semantics-aware content-based recommender systems: Design and architecture guidelines," *Neurocomputing*, vol. 254, pp. 79–85, Sep. 2017, doi: 10.1016/j.neucom.2016.10.079.

[12] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme, "Fast context-aware recommendations with factorization machines," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Beijing, China, Jul. 2011, pp. 635–644, doi: 10.1145/2009916.2010002.

[13] Y.-X. Peng *et al.*, "Cross-media analysis and reasoning: Advances and directions," *Frontiers Inf. Technol.*, vol. 18, no. 5, pp. 44–57, 2017.

[14] H. T. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Boston, MA, USA, Sep. 2016, pp. 7–10, doi: 10.1145/2988450.2988454.

[15] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell. Main Track*, 2017, pp. 1725–1731, doi: 10.24963/ijcai.2017/239.

[16] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[17] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009, doi: 10.1109/MC.2009.263.

[18] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004, doi: 10.1145/963770.963774.

[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[20] D. Agarwal and B. C. Chen, "Regression-based latent factor models," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Paris, France, Jun. 2009, pp. 19–28, doi: 10.1145/1557019.1557029.

[21] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun. ACM*, vol. 53, no. 4, pp. 447–456, 2009, doi: 10.1145/1557019.1557072.

[22] L. Xiong, X. Chen, T. K. Huang, J. G. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with bayesian probabilistic tensor factorization," in *Proc. SIAM Int. Conf. Data Mining*, Columbus, OH, USA, May 2010, pp. 211–222, doi: 10.1137/1.9781611972801.19.

[23] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2010, pp. 995–1000, doi: 10.1109/ICDM.2010.127.

[24] Y. Bengio, "Learning deep architecture for AI," *Foundations Trends Mach. Learn.*, 2009, pp. 21–127.

[25] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2018–2025, doi: 10.1109/ICCV.2011.6126474.

[26] L. Deng, O. Abdel-Hamid, and D. Yu, "A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 6669–6673, doi: 10.1109/ICASSP.2013.6638952.

[27] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12 pp. 2493–2537, Aug. 2011.

[28] S. Zhang, L. Yao, and A. Sun. (2017). "Deep learning based recommender system: A survey and new perspectives." [Online]. Available: https://arxiv.org/abs/1707.07435

[29] W. Zhang, T. Du, and J. Wang. (2016). "Deep learning over multi-field categorical data: A case study on user response prediction." [Online]. Available: http://arxiv.org/abs/1601.02376

[30] W. Jian, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Syst. Appl.*, vol. 69, pp. 29–39, Mar. 2017, doi: 10.1016/j.eswa.2016.09.040.

[31] Y. Qu *et al.*, "Product-based neural networks for user response prediction," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Barcelona, Spain, Dec. 2016, pp. 1149–1154, doi: 10.1109/ICDM.2016.0151.

[32] L. Liao, X. He, H. Zhang, and T. S. Chua, "Attributed social network embedding," *IEEE Trans. Knowl. Data Eng.*, to be published, doi: 10.1109/TKDE.2018.2819980.

[33] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Feb. 2010, doi: 10.1145/1756006.1756025.

[34] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efcient estimation of word representations in vector space," *Comput. Sci.*, 2013. [Online]. Available: https://arxiv.org/pdf/1301.3781.pdf

[35] L. O. Chua and T. Roska, "Cnn paradigm," *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.*, vol. 40, no. 3, pp. 147–156, 1993, doi: 10.1109/81.222795.

[36] X. He and T. S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Shinjuku, Tokyo, Japan, Aug. 2017, pp. 355–364, doi: 10.1145/3077136.3080777.

[37] X. Wang, X. He, L. Nie, and T. S. Chua, "Item silk road: Recommending items from information domains to social users," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Shinjuku, Tokyo, Japan, Aug. 2017, pp. 185–194, doi: 10.1145/3077136.3080771.

[38] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2009, pp. 263–272, doi: 10.1109/ICDM.2008.22.

[39] R. Pan *et al.*, "One-class collaborative filtering," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 502–511, doi: 10.1109/ICDM.2008.16.

[40] R. M. Bell and Y. Koren, "Improved neighborhood-based collaborative filtering," in *Proc. KDDCup Workshop*, vol. 5, 2007, no. 2, pp. 7–14, doi: 10.1007/s007790170019.

WEIHUA YUAN received the B.S. degree from the School of Information Management, Shandong Normal University, in 2000, the M.S. degree in computer application from Guizhou University in 2006, and the Ph.D. degree from Shandong Normal University in 2018 under the supervision of Dr. H. Wang. She has been with the School of Computer Science and Technology, Shandong Jianzhu University, since 2006. She has authored two books and more than 10 articles. Her research interests include recommender systems, machine learning, data mining, and big data cloud.

HONG WANG was born in 1966. She received the master's degree in software from the Department of Computer Science and Technology, Tianjin University, in 1991, the Ph.D. degree in computer applications from the Computing Institute, Chinese Academy of Sciences, in 2002, and the Postdoctoral degree from the Postdoctoral Station of Computer Science and Technology, Shandong University, in 2009. She is currently a Professor and the Doctoral Tutor with Shandong Normal University.

As a Project Leader, she has sponsored many scientific research projects, for example, the Shandong Young Scientist Award Fund, a Postdoctoral Grant Selection Program in Shandong Province, and the Natural Science Foundation of Shandong Province. She has also guided many joint venture projects for companies. In recent years, she has published more than 60 articles in academic journals at home and abroad, two software copyrights, and wrote many textbooks as a chief editor. Her research interests include data mining and machine learning. She received the Second Prize of outstanding achievements in computer applications in Shandong Province, the second prize of the Outstanding Scientific Research Achievement Award in Shandong Province, the Second Prize and Third Prize of the Shandong Provincial Science and Technology Award. She is currently the Chair of the National Natural Science Foundation of China and the Shandong Science and Technology Planning Project. She has cultivated four doctoral candidates and over 40 master students.

BAOFANG HU received the M.E. degree in computer software and theory from the School of Information Science and Engineering, Shandong Normal University, China, in 2006, where she is currently pursuing the Ph.D. degree under the supervision of Prof. H. Wang. Her research interests include data mining and complex networks.

LUTONG WANG received the bachelor's degree from the School of Information Science and Engineering, Shandong Normal University, Jinan, China, in 2017, where she is currently pursuing the master's degree in computer software and theory. Her research interests include machine learning and complex networks.

QIAN WANG currently pursuing the master's degree in computer software and theory with the School of Information Science and Engineering, Shandong Normal University, China, under the supervision of Prof. W. Hong. Her research interests include data mining and machine learning.