IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 13, 2020, accepted February 10, 2020, date of publication March 9, 2020, date of current version March 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979255

Hybrid Collaborative Recommendation via Dual-Autoencoder

BINGBING DONG [1,2,3], YI ZHU [1,2,3,4], LEI LI [1,2,3], (Senior Member, IEEE), AND XINDONG WU [1,2,3,5], (Fellow, IEEE)

[1]Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, Hefei 230009, China
[2]School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China
[3]Institute of Big Knowledge Science, Hefei University of Technology, Hefei 230009, China
[4]School of Information Engineering, Yangzhou University, Yangzhou 225009, China
[5]Mininglamp Academy of Sciences, Mininglamp Technology, Beijing 100084, China

Corresponding author: Xindong Wu (xwu@hfut.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000901, in part by the National Natural Science Foundation of China under Grant 61906060 and Grant 91746209, in part by the Fundamental Research Funds for the Central Universities under Grant JZ2018HGBH0279, and in part by the Project funded by the China Postdoctoral Science Foundation under Grant 2018M630704.

ABSTRACT With the rapid increase of internet information, personalized recommendation systems are an effective way to alleviate the information overload problem, which has attracted extensive attention in recent years. The traditional collaborative filtering utilizes matrix factorization methods to learn hidden feature representations of users and/or items. With deep learning achieved good performance in representation learning, the autoencoder model is widely applied in recommendation systems for the advantages of fast convergence and no label requirement. However, the previous recommendation systems may take the reconstruction output of an autoencoder as the prediction of missing values directly, which may deteriorate their performance and cause unsatisfactory results of recommendation. In addition, the parameters of an autoencoder need to be pre-trained ahead, which greatly increases the time complexity. To address these problems, in this paper, we propose a Hybrid Collaborative Recommendation method via Dual-Autoencoder (HCRDa). More specifically, firstly, a novel dual-autoencoder is utilized to simultaneously learn the feature representations of users and items in our HCRDa, which obviously reduces time complexity. Secondly, embedding matrix factorization into the training process of the autoencoder further improves the quality of hidden features for users and items. Finally, additional attributes of users and items are utilized to alleviate the cold start problem and to make hybrid recommendations. Comprehensive experiments on several real-world data sets demonstrate the effectiveness of our proposed method in comparison with several state-of-the-art methods.

INDEX TERMS Recommendation system, matrix factorization, semi-autoencoder.

I. INTRODUCTION

With the rapid development of the Internet and the huge surge in data volume, we have gradually entered the era of information overload, which makes both information consumers and information producers face great challenges particularly. Recommendation system is an important tool for addressing these challenges and bridging the chasm between users and information, which simultaneously helps users to find valuable information and recommend information to interested users for achieving a win-win situation between information consumers and producers.

Collaborative Filtering (CF) is a widely used recommendation system approach. In general, existing collaborative filtering recommendation methods can roughly be categorized into three classes: user-based collaborative filtering, item-based one and model-based one. The user-based collaborative filtering mainly considers the similarity between different users, and recommends items that similar user like to the target user [1]. The item-based collaborative filtering is similar to the user-based collaborative filtering, except that it turns to find the similarity between different items [2]. Model based

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Xiao.

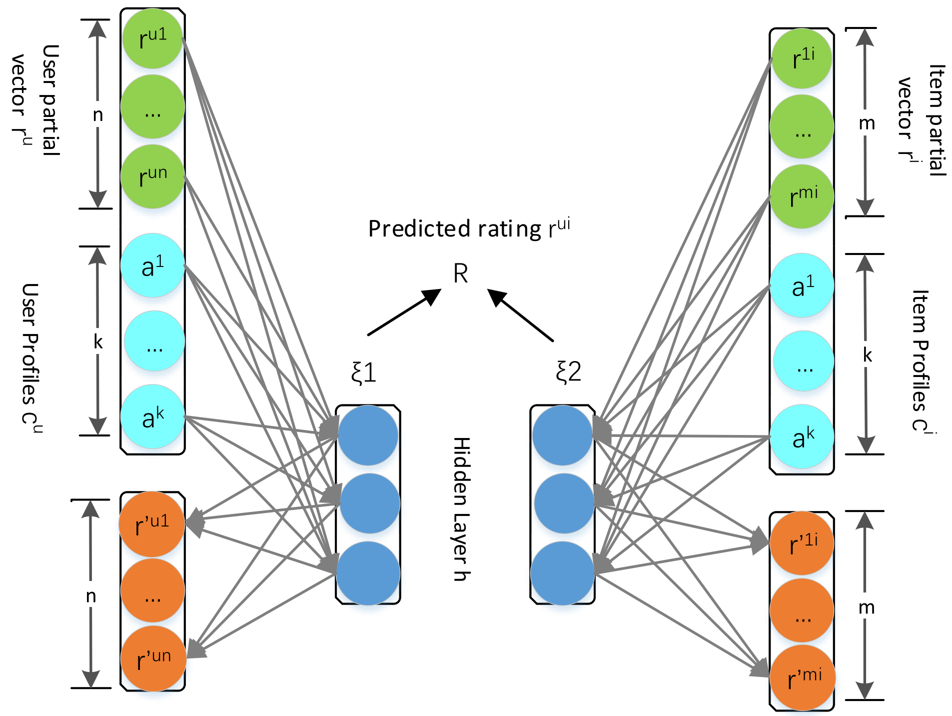

FIGURE 1. Illustration of Hybrid Collaborative Recommendation via Dual-Autoencoder.

collaborative filtering is currently popular recommendation method, which aims to factorize the rating matrix into user and item matrices for personalized recommendation, such as matrix factorization [3]. However, the matrix factorization method usually directly decomposes the user rating matrix to obtain the hidden features of the user and the item, and it is difficult to obtain the nonlinear features of the user and the item very well. Moreover, the CF method is greatly limited when the user raing matrix is sparse. Zhuang *et al.* [4] showed that traditional matrix factorization may not make full use of the rating matrix, and proved the effectiveness of auto-encoder in learning features. Therefore, in order to make full use of the rating information, the autoencoder model is leveraged in proposed HRCDa to generate hidden features of users and items, and the additional attributes of users and items is combined to alleviate the matrix sparse problem.

With the substantial performance of learning abstract and powerful feature representations, the methods of deep learning have achieved far-ranging consequences in various fields. Some deep learning methods have been applied to recommendation systems for improving predictive performance [4]–[8]. For example, Liang and Baldwin [8] utilized an autoencoder model to learn the user latent feature matrix for achieving fairly good performance in recommendation. Zhuang *et al.* [4] proposed a Dual-Autoencoder model to generate latent user and item feature matrices. Even though the aforementioned methods combine the autoencoder with the CF, the weight and offset parameters of the network need to be pre-trained in most of these methods, and then the sparse rating matrix is imported into the autoencoder to generate the prediction matrix.

To address the aforementioned problems, we propose a Hybrid Collaborative Recommendation method via Dual-Autoencoder (HCRDa for short). To the best of our knowledge, this is the first recommendation method to embed matrix factorization into the learning process of autoencoder. As far as we know, most of the methods based on autoencoders consider the user and item ratings separately [9], [10], and we use the matrix factorization method to consider both user and item ratings. The framework of our proposed method is shown as Figure 1. More specifically, the model of dual-autoencoder is employed to simultaneously learn the low-dimensional feature representations of users and items, which obviously reduce the time complexity of the method. Meanwhile, the additional attributes about users and items is added into the input layer for alleviating the cold start problem. Finally, we multiply the feature matrix of the user and the item learned when the algorithm converges to obtain the prediction matrix. Our key contributions are summarized as follows:

- We propose a recommendation framework named HCRDa. We use two novel auto-encoders, which take the additional attributes of users and items in combination with the corresponding rating information as input, while learning the hidden features of users and items to reduce time complexity and alleviate the cold start problem of collaborative filtering.

TABLE 1. The notation and descriptions.

NOTATIONS	DESCRIPTIONS
R	The rating matrix $R \in R^{n\times m}$
R'	The prediction matrix $R' \in R^{n\times m}$
m	The number of users
n	The number of items
h	The number of hidden neurons
y_i	The number of attributes information of item
y_u	The number of attributes information of user
r^{ui}	the rating to item $i \in \{1...n\}$ given by user $u \in \{1...m\}$
r^u	The column of the rating matrix
r^i	The row of the rating matrix
a^u	The attribute features of user u
a^i	The additional features of item i
r^U	The partial observed vectors for all users
r^I	The partial observed vectors for all items
A^U	The attribute features for all users
A^I	The additional features all items
Ω	The set of observed ratings
x	Raw input data
x'	Refactoring output data
ξ	The latent representation of users or items
Q, p, Q_u, p_u	The offset parameters of encoding layer in Autoencoder
Q', p', Q'_u, p'_u	Autoencoder decoding layer offset parameter
T	Transposition of matrix

- The matrix factorization method is the first embedded into the learning process of the autoencoder to better obtain the hidden features of users and items and improve the quality of recommendation.
- The comprehensive experiments on several data sets in the real world demonstrate the effectiveness of our proposed HCRDa.

The remainder of this paper is organized as follows. We introduce the notations frequently used in this paper and related preliminary knowledge in the second section, and then the framework of hybrid collaborative recommendation via dual-autoencoder is introduced in Section 3. In Section 4, we describe the experimental setup and the experimental results of several data sets in detail. In section 5, we briefly introduce the related work, and in section 6, we summarize our proposed method and introduce the future work.

II. NOTATIONS AND PRELIMINARIES

In this section, we first introduce some frequently used notations as presented in Table 1. Then, we introduce some preliminaries knowledge.

A. AUTOENCODER

Autoencoder is an unsupervised model which attempted to reconstruct the input data in the output layer [11]. The Autoencoder consists of a three-layer network in which the number of neurons in the input layer is equal to the number of neurons in the output layer, and the number of neurons in the middle layer is generally smaller than that of the input layer and the output layer. The reconstructed representations are generated at the output layer for each training sample through the network. The purpose of model is to minimize the discrepancy between input and output data by the reconstructed representations. In a three-layer autoencoder, the process can be shown as in Eq. (1) and Eq. (2):

$$\xi = f\left(Qx + p\right) \tag{1}$$

$$x' = g\left(Q'\xi + p'\right) \tag{2}$$

where Q and Q' are weight matrices whose dimensions are $Q \in R^{h\times n}$ and $Q' \in R^{n\times h}$ respectively, p and p' are bias vectors whose dimensions are $p \in R^h$ and $p' \in R^n$ respectively, and f and g are universal activation functions such as tanh, sigmoid and identity. Given m input instances $\{x_i\}_{i=1}^{i=m}$, the goal of the autoencoder is to reconstruct the input data for minimizing the gap between the output and the input data, so the expression of objective function is shown as in Eq. (3):

$$\begin{aligned}\min_{Q,Q',p,p'} \sum_{i=1}^{m} \left\|x_i - x_i'\right\|^2 \\ = \min_{Q,Q',p,p'} \sum_{i=1}^{m} \left\|x_i - g\left(Q'\left(f\left(Qx_i + p\right)\right) + p'\right)\right\|^2\end{aligned} \tag{3}$$

In our method, sigmoid function is adopted for the encoding activation of the autoencoder, and identity function is adopted for the decoding activation, which are two widely used nonlinear activation functions.

B. SEMI-AUTOENCODER

Almost all variants of autoencoder can be applied to recommended systems, such as denoising autoencoder [12], [13], variational autoencoder [14]–[16], contractive autoencoder [17], [18] and marginalized autoencoder [19]. However, the dimensions of input layer and output layer are required to be equal in traditional autoencoder model. To address this problem, Zhang *et al.* presented a model of Hybrid Collaborative Recommendation via Semi-Autoencoder [9], which utilized the additional information for latent representation learning by breaking the limitation of the output and input dimensionality.

The input layer and output layer in semi-autoencoder can own different dimensions, which is shown as Fig 2. In order to make better use of additional information, the dimension of input layer is greater than or equal to the dimension of output layer, the structure of semi-autoencoder is shown as the right half of Fig 2. There are two main advantages of semi-autoencoder compared to autoencoder: firstly, the different feature representations and reconstructions can be captured flexibly by sampling different subsets from the inputs, secondly is the convenience of semi-autoencoder to incorporate additional information in the input layer.

III. METHODOLOGY

In this section, we first introduce the motivation for the proposed method, and then the whole framework of proposed HCRDa is formulated in detail.

A. MOTIVATION

Since the deep learning methods have achieved excellent performance in recommendation systems, the requirement to large quantities of labeled data and the existence of high time complexity prevent the further development of supervised deep network, such as Convolutional Neural Network (CNN)

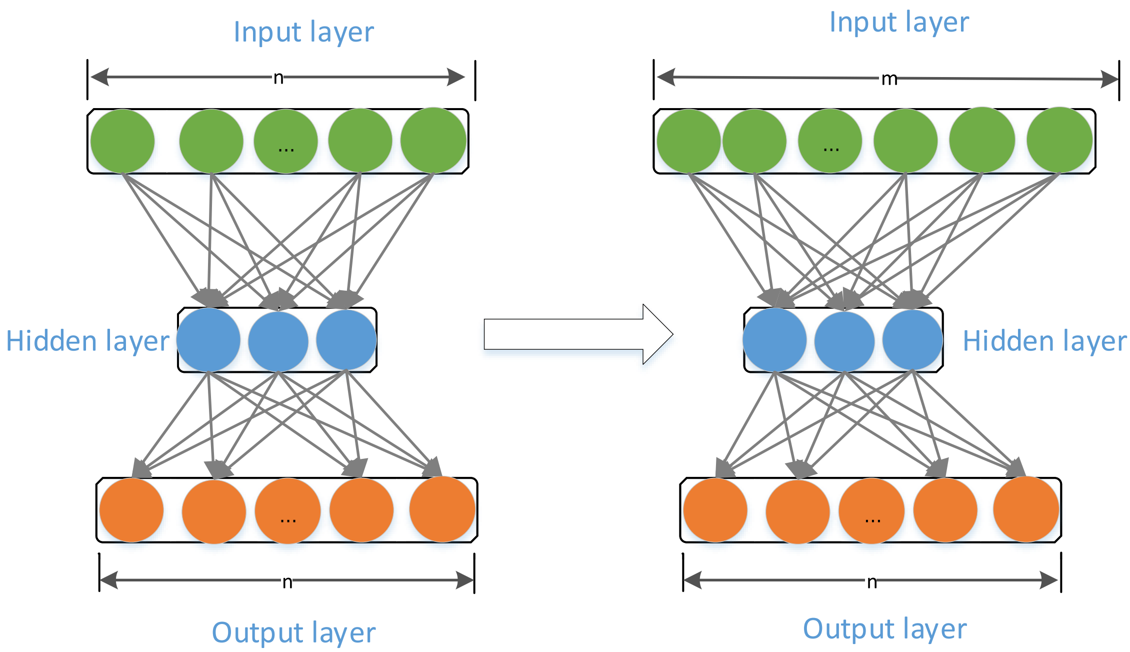

FIGURE 2. Illustration of Autoencoder and Semi-Autoencoder.

and Recurrent Neural Network (RNN). Compared to supervised deep learning methods, there are far less attention paid in autoencoder model for recommendation. However, autoencoder has proven to able to learn substantial representations in many fields, such as image classification [20]. Along this line, we propose to introduce an autoencoder in recommendation systems and achieve satisfied results. Traditional matrix factorization generally mapped a high-dimensional matrix to two low-dimensional matrices. Compared with traditional matrix factorization methods, autoencoder-based methods can bring nonlinear parts to the model and improve the fitting ability. On the other hand, a variety of features can be easily exploited to improve the accuracy of the model.

Besides this, there are two main problems in the existing autoencoder recommendation methods. The first is the training of parameters in autoencoder, and the existing autoencoder recommendation methods such as ReDa [4] and REAP [21] where all pre-train the weights and bias parameters of the network firstly, which caused excessive convergence time and greatly increased the time complexity. To address this problem, the sparse rating matrix is directly input into the autoencoder to learn the feature representations of the user and item in our proposed method, and the rating matrix for predicating which reduces the complexity compared to the previous model is obtained.

The second problem is the prediction of missing values for ratings. Most of the previous autoencoder based recommendation systems take the reconstruction output as the prediction of missing values directly, which may deteriorate performance and achieve the unsatisfied results of recommendation. To address this problem, the matrix factorization technique is embedded into the training process of the autoencoder, and the reconstruction rating matrix of users and items is leveraged to predict missing values. In addition, the additional auxiliary information is combined in our proposed method for alleviating the cold start problem.

B. THE PROPOSED HCRDa

Our proposed HCRDa is introduced with details in this section. Given the rating matrix $R \in R^{n\times m}$ with n items and m users. The observed rating vector r^i is combined with the additional information vector a^i of the item as the input of the autoencoder to learn the hidden feature representation of item i. The $In\left(r^i, a^i\right) \in R^{(m+y_i)}$ is the concatenation of r^i and a^i, and $In\left(r^I, A^I\right) \in R^{n\times(m+y_i)}$ is the concatenation of r^I and A^I, where r^I refers to all item rating vectors for a total of n items, and A^I refers to the additional information vector for all items. The $In\left(r^I, A^I\right)$ is input into the semi-autoencoder to learn the hidden representations of the item ξ_i. The purpose of the autoencoder is to approximate the initial input, and the original intention of the semi-autoencoder in this paper is to make the output approximate the partial input, that is, the rating data. So the reconstruction loss of the semi-autoencoder that deals with the item, f_i of Eq.(6), is the first term of our objective function. The encoding and decoding processes of item are shown as Eq.(4), Eq.(5) and Eq.(6):

$$\xi_i = f\left(In\left(r^I, A^I\right)Q + p\right) \tag{4}$$

$$r'^I = g\left(Q'\xi_i + p'\right) \tag{5}$$

$$f_i = \min_{Q,Q',p,p',r^{ui\in\Omega}} \left\| r^I - r'^I \right\|^2 \tag{6}$$

Notably, the functions f and g are nonlinear functions where f is the sigmoid function and g is the identity function.

$Q \in R^{(m+y_i)\times h}$ and $Q' \in R^{h\times m}$ are the weight matrices of autoencoder, $p \in R^h$ and $p' \in R^m$ are the bias terms, and h is the dimensions of hidden layer.

Similarly, the $In\left(r^U, A^U\right) \in R^{m\times(n+y_u)}$ is input into the semi-autoencoder, where r^U is the observed rating vector and A^U is the attribute vector of all users. Similar to the first part of the objective function, the reconstruction loss of the autoencoder that processes user ratings, f_u of Eq. (9), is used as the second term of our objective function. The encoding and decoding processes of user are as shown as Eq. (7), Eq. (8) and Eq. (9):

$$\xi_u = f\left(In\left(r^U, A^U\right)Q_u + p_u\right) \tag{7}$$

$$r'^U = g\left(Q'_u \xi_u + p'_u\right) \tag{8}$$

$$f_u = \min_{Qu,Qu',pu,pu',r^{ui\in\Omega}} \left\| r^U - r'^U \right\|^2 \tag{9}$$

where $Q_u \in R^{(n+y_u)\times h}$ and $Q'_u \in R^{h\times n}$ are the weight matrices of autoencoder, and $p_u \in R^h$ and $p'_u \in R^n$ are the bias terms. In order to better learn the hidden features of users and items simultaneously, we combine the method of matrix factorization. The matrix factorization technique is embedded in the learning process of the autoencoder, so that the final prediction matrix $\xi_i {\xi_u}^T$ approaches the original rating matrix, and Eq. (10) is used as the third term of our objective function.

$$f_{mf} = \min_{Q,Q',p,p',Q_u,Q'_u,p_u,p'_u,r^{ui\in\Omega}} \left\| R - \xi_i {\xi_u}^T \right\|^2 \tag{10}$$

In order to avoid over-fitting, we add the regularization item of weight matrix with $l2$ norm as the last term of our objective function, which is shown as Eq. (11), where γ is a trade-off parameter,

$$f_l = \frac{\gamma}{2}\left(\|Q\|^2 + \left\|Q'\right\|^2 + \|Q_u\|^2 + \left\|Q'_u\right\|^2\right) \tag{11}$$

In summary, the final objective function of our proposed framework is shown as Eq. (12):

$$\min_{Q,Q',p,p',Q_u,Q'_u,p_u,p'_u,r^{ui\in\Omega}} f_i + f_u + f_{mf} + f_l \tag{12}$$

The stochastic gradient descent is used in the proposed method to optimize the Eq. (12). Until the algorithm converges, we can get the prediction matrix R' as Eq. (13). The specific implementation steps of our proposed HCRDa algorithm are summarized in Algorithm 1.

$$R' = \xi_i {\xi_u}^T \tag{13}$$

IV. EXPERIMENTS

In this section, we conduct extensive experiments on three real-word datasets to systemically demonstrate the effectiveness of our proposed methods for recommendation.

Algorithm 1 HCRDa Algorithm

Input: The rating matrix $R \in R^{n\times m}$, the number of hidden neurons h, and the dimention of user and item attribute vector is y_u and y_i, respectively.
Parameter: Trade-off parameter γ.
Output: The prediction matrix $R' = \xi_i {\xi_u}^T$.

1: Get the attribute information vector a_u for each user;
2: Get the attribute information vector a_i for each item;
3: Get the splicing vectors $In\left(r^U, A^U\right)$ and $In\left(r^I, A^I\right)$ of the attribute vectors of the users and the items and the corresponding rating vectors;
4: Initialize Q, Q_u, Q', Q'_u by truncating a normal-distributed random number, and set p, p_u, p', p'_u to 0 vectors.
5: Input $In\left(r^U, A^U\right)$ and $In\left(r^I, A^I\right)$ to two semi-autoencoders;
6: Minimize Eq. (12) using a stochastic gradient descent algorithm until the algorithm converges;
7: **return** $R' = \xi_i {\xi_u}^T$.

A. DATA SETS

We conduct experiments on three real world datasets for evaluating the performance of our proposed HCRDa. The details of these three datasets are summarized in Table2.

- The first is the stable benchmark dataset that is widely used to evaluate the performance of recommendation system: Movielens 100K.[1] Movielens 100K contains additional information about the user and the item, the item's additional information used in the experiment includes the genre and year of release, and the user's additional attributes includes age, occupation and gender.
- The second data set is MovieTweetings [22], which is a collection of movie rating data from Twitter. This is an extremely sparse rating data set, and the scale of rating has changed from the traditional 1-5 to 1-10. In the experiment, we take a 10K snapshot of this dataset and retain users who have rated at least ten items. The same item's additional information as Movielens 100K dataset is applied in the experiment, but without combining user profiles.
- The last dataset is FilmTrust, which is a collection of movie rating data from the website FilmTrust [23]. The scale of rating has changed from the traditional 1-5 to 0.5-4. This dataset provides trust value between users in social network-based recommendation systems, but does not provide user and item attribute information. Therefore, we only use the rating information of this dataset to make recommendations in the experiment. Since the data set provides the trust value between users, we conduct the experiment and compare the results

[1] http://files.grouplens.org/datasets/movielens/ml-100k.zip

TABLE 2. Statistics of three datasets.

Dataset	Number of items	Number of users	Number of ratings	Desity (%)
MovieLens 100K	1682	943	100000	6.30
MovieTweetings 10K	3096	123	2233	0.59
FilmTrust	2071	1508	35497	1.14

between our proposed HCRDa and the method using trust matrix factorization technology.

B. COMPARED METHODS AND IMPLEMENTATION DETAILS

1) BASELINE METHODS

Since the matrix factorization methods that embedded in the training process of the autoencoder are different from the previous matrix factorization methods and autoencoder recommendation methods, we first compare our proposed HCRDa with the traditional matrix factorization method, and then compare HCRDa with state-of-the-art methods based on autoencoder, including:

- NMF (Non-negative Matrix Factorization) [24]: the basic recomemendation of algorithms for Non-negative Matrix Factorization.
- PMF (Probabilistic Matrix Factorization) [25]: the recommendation algorithms for Probabilistic Matrix Factorization.
- BPMF (Bayesian Probabilistic Matrix Factorization) [26]: the recommendation algorithms for Probabilistic Matrix Factorization.
- SVD++ [27]: SVD++ combines the matrix factorization model with the neighborhood model to propose a multifaceted collaborative filtering model.
- ReDa (Recommendation via Dual-Autoencoder) [4]: representation learning via Dual-Autoencoder, which uses autoencoder to generate latent user and item feature matrixs.
- HRSA (Hybrid collaborative recommendation via semi-autoencoder)[2] [9]: HRSA generalize Semi-Autoencoder into a hybrid collaborative filtering model for rating prediction as well as personalized top-n recommendations.
- TrustMF [28]: TrustMF adopts matrix decomposition technology to make recommendation based on the dual sparse data of users' rating information and social trust network data.
- TrustSVD [29]: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings.

2) IMPLEMENTATION DETAILS

The baselines NMF, PMF, BPMF, SVD++, TrustMF and TrustSVD implemented by the Toolkit Librec.[3] The Reda method has no source code, and its experimental results are obtained from reported papers. The bias matrix of the autocoder is generated randomly from a normal distribution of the specified mean and standard deviation, and the initial value of the bias vector is set to zero. We detect different hidden layer neurons for the HCRDa model, and set the number of hidden neurons to 500. After some preliminary test, the final learning rate is set to 0.001, and trade-off parameter γ is set as 1.

[2] https://github.com/cheungdaven/semi-ae-recsys

[3] http://www.librec.net/download.html

TABLE 3. RMSE and MAE on Movielens100K.

Setting	Metric	NMF	PMF	BPMF	SVD++	ReDa	HRSA	HCRDa
70%	MAE	0.7572	0.7396	**0.7132**	0.7240	0.7248 ±0.0067	0.7185 ±0.0041	0.7199 ±0.0031
	RMSE	0.9614	0.9324	**0.9101**	0.9245	0.9231 ±0.0081	0.9109 ±0.0030	0.9176 ±0.0009
80%	MAE	0.7527	0.7332	0.7044	0.7161	0.7203 ±0.0043	0.7073 ±0.0070	**0.6751 ±0.0007**
	RMSE	0.9531	0.9324	0.9000	0.9145	0.9190 ±0.0056	0.8971 ±0.0054	**0.8645 ±0.0010**
90%	MAE	0.7519	0.7286	0.6976	0.7109	0.7153 ±0.0094	0.7051 ±0.0034	**0.6089 ±0.0099**
	RMSE	0.9499	0.9226	0.8880	0.9052	0.9114 ±0.0093	0.8961 ±0.0106	**0.7879 ±0.0061**

TABLE 4. RMSE and MAE on MovieTweetings 10K.

Setting	Metric	NMF	PMF	BPMF	SVD++	HRSA	HCRDa
70%	MAE	1.3636	1.4262	1.3411	1.2306	2.4629 ±0.0300	**0.7519 ±0.0208**
	RMSE	1.8314	1.8945	1.7547	1.6285	2.8488 ±0.0389	**1.0513 ±0.0225**
80%	MAE	1.3768	1.3987	1.3681	1.2267	2.4756 ±0.0265	**0.6496 ±0.0807**
	RMSE	1.8533	1.8636	1.7568	1.6213	2.8383 ±0.0084	**0.9255 ±0.0794**
90%	MAE	1.2373	1.341	1.2898	1.1397	2.2989 ±0.0487	**0.6173 ±0.0497**
	RMSE	1.7247	1.7526	1.7105	1.5296	2.7186 ±0.0504	**0.8588 ±0.0537**

3) EVALUATION METRICS

We use these methods to evaluate their performance in terms of their respective Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) on the test set.

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R^{test}} \left(R_{u,i} - R'_{u,i}\right)^2}{|N\left(R^{test}\right)|}} \tag{14}$$

$$MAE = \frac{\sum_{(u,i)\in R^{test}} \left|R_{u,i} - R'_{u,i}\right|}{|N\left(R^{test}\right)|} \tag{15}$$

where R^{test} denotes the testing rating matrix, R and R' are the original matrix and the prediction matrix respectively, $(u, i) \in R^{test}$ denotes the rating which user u gives the item i in the testing dataset, and $N\left(R^{test}\right)$ is the number of ratings in R^{test}. The smaller the values of MAE and RMSE, the better the performance of the method.

C. EXPERIMENTAL RESULTS

For each data set, different proportions of the training set are randomly selected from the samples, and the rest

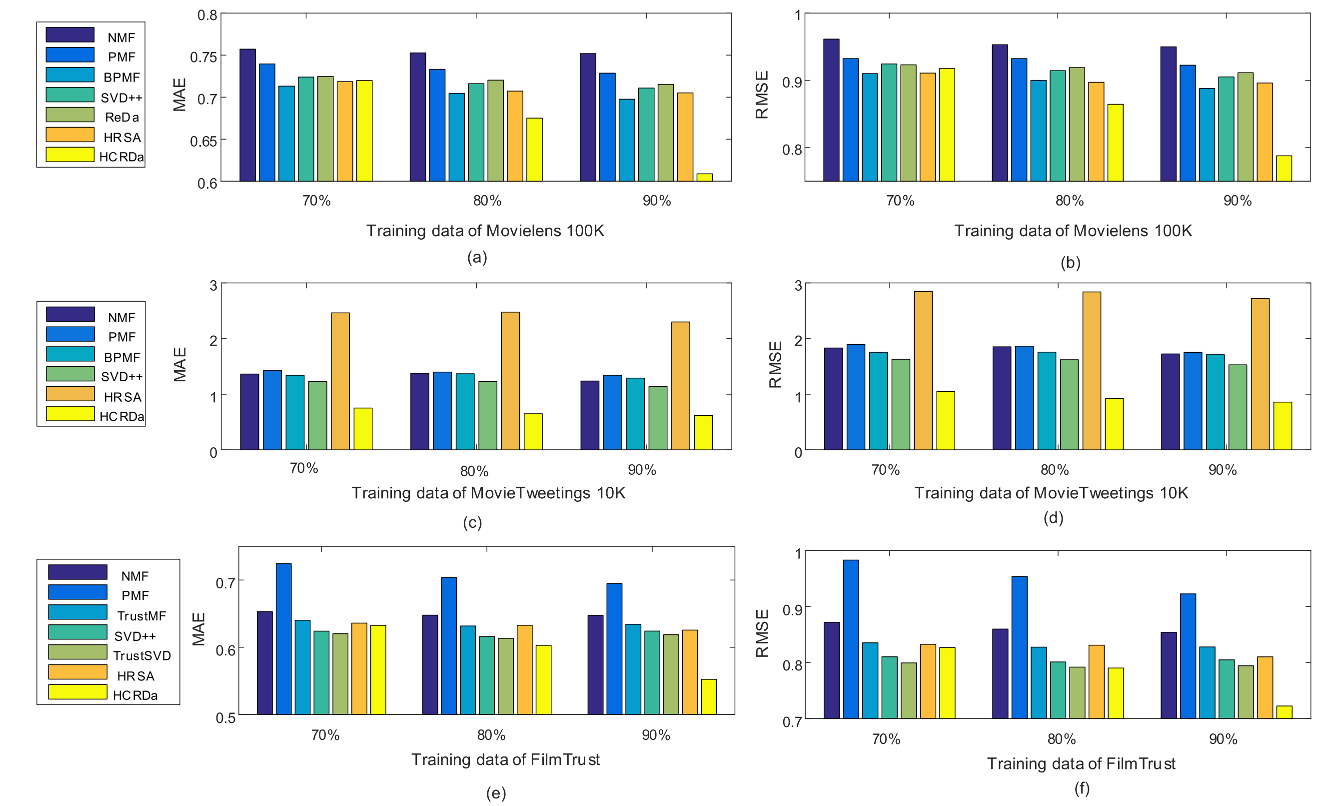

FIGURE 3. Recommendation quality of different algorithms tested on three datasets.

TABLE 5. RMSE and MAE on FilmTrust.

Setting	Metric	NMF	PMF	TrustMF	SVD++	TrustSVD	HRSA	HCRDa
70%	MAE	0.6531	0.7244	0.6402	0.6240	**0.6202**	0.6360 ±0.0039	0.6326 ±0.0127
	RMSE	0.8718	0.9829	0.8353	0.8104	**0.7995**	0.8326 ±0.0073	0.8268 ±0.0109
80%	MAE	0.6478	0.7039	0.6317	0.6159	0.6133	0.6327 ±0.0115	**0.6029 ±0.0023**
	RMSE	0.8599	0.9536	0.8275	0.8012	0.7919	0.8311 ±0.0112	**0.7903 ±0.0015**
90%	MAE	0.6476	0.6948	0.6342	0.6242	0.6188	0.6256 ±0.0010	**0.5524 ±0.0009**
	RMSE	0.8539	0.9226	0.8280	0.8049	0.7943	0.8103 ±0.0001	**0.7225 ±0.0037**

are used as the testing set. Tables 3, 4 and 5 record the experimental results on the three datasets respectively. We have the following observations from experimental results:

- Fig. 3 clearly reflects that, in general, with the increase of training data, the performance of all algorithms will be better.
- The performance of HCRDa exceeds the traditional matrix factorization methods NMF, PMF, BPMF and SVD ++ (except for one case, the performance of BPMF exceeds HCRDa), which reveals the ability of autoencoder model to learn effective features of users and items. When the training rate reaches 80% and above, HCRDa outperforms all comparison algorithms on three datasets.
- Only when the sample ratio for training data is 70%, the recommendation methods based on deep learning, such as ReDa, HRSA and our HCRDa, may perform worse than the traditional matrix factorization algorithm BPMF and TrustSVD. This may be because the deep learning based recommendation methods require a large amount of data for training the network to learn the better feature representations.
- The weight parameters of ReDa algorithm are initialized by stacked autoencoders with the input of rating matrix firstly. However, our proposed HCRDa randomly initializes parameters, which reduces the complexity of the algorithm and achieves better results than ReDa.
- HRSA makes user-based and item-based recommendations based on user and item rating matrix and additional attribute information respectively, without considering the idea of collaborative filtering. HCRDa considers the hidden vectors of users and items at the same time, surpassing the performance of HRSA, indicating the effectiveness of embedding matrix factorization into the learning process of autoencoders.
- The histogram in Fig. 3 intuitively reflects that with the increase of the training set, our HCRDa algorithm has a more significant improvement on the three data sets compared to other comparison algorithms.

- All experimental results show that the accuracy of all algorithms will increase with the increase of the number of ratings. The height of our algorithm HCRDa in (c) and (d) of Fig. 3 is almost less than half of the height of other comparison algorithms. The significant effect of HCRDa on 10K indicates that our algorithm is less affected by rating density than baselines, so HRCDa has an advantage in processing sparse data.

D. ANALYSIS OF PROPERTIES IN HCRDa

1) IMPACT OF DATA SPARSITY

The sparsity of data has always been one of the factors affecting the effectiveness of recommendation methods. Table 4 and the middle row of Fig. 3 reflects the experimental results of various methods on snapshots of MovieTweetings in a particularly sparse dataset. Our algorithm did not improve significantly on the movielens 100K dataset, and improved the most on 90% of the 100K. Compared with the comparison algorithms such as NMF, PNF, BPMF, SVD++, ReDa and HRSA, our algorithm improved the RMSE evaluation criteria by 17.1%, 14.6%, 11.3%, 13.0%, 13.6% and 12.1%, respectively. On other training sets, our algorithms have slightly improved RMSE evaluation standards than other comparison algorithms (except the BPMF algorithm on 70% of the 100K set). However, on 10K datasets, our algorithm improved the RMSE performance of other comparison algorithms by around 50%. For example, when the sample ratio for training data is 80%, the percentage improvement of our algorithm in RMSE relative to the comparison algorithm is 50.1, 50.3, 47.3, 42.9 and 67.3 respectively. For more sparse datasets, the more obvious advantages of our proposed method. The experimental results of the matrix factorization methods on the MovieTweetings 10K dataset are not much different, but SVD++ has the highest accuracy. This may be because SVD++ not only considers the user's rating information but also combines the implicit information to improve the experimental performance. The HRSA that uses only one autoencoder to reconstruct the user's rating matrix has a much worse result on this data set, which means the autocoder may not be able to learn the hidden features of the user and the item well. Although our method also uses the autoencoder model to learn the features of users and items, we combine matrix factorization technology during the training process, so that the autoencoder can better learn the hidden features of users and items, and improve the experimental performance.

2) IMPACT OF TRUST VALUE

Table 5 and the last line of Fig. 3 shows the experimental results of all methods on the dataset filmtrust. On this data set, we have added the TrustSVD++ contrast algorithm, which is the first algorithm that combines SVD++ and social trust information to obtain an excellent effect. The TrustSVD model outperforms other comparison methods on this dataset. When the training rate is 80% and 90%, our proposed HCRDa achieves satisfactory results, even without the trust value between users. Especially when the training rate is 90%, our algorithm improves 15.4%, 21.7%, 12.7%, 10.2%, 9.0%, and 10.8% compared with other comparison algorithms including TrustSVD++ on the RMSE evaluation standard. However, with only 70% of the training set, the trust value between users may play a certain role, and TrustSVD achieved a leading accuracy rate. However, the algorithm is only 3.3% more accurate on RMSE than we are. To sum up, we can achieve or even surpass the algorithms using explicit information, implicit information and trust value information by using rating information and additional information.

TABLE 6. Impact of side information on Movielens 100K.

Movielens100K	Metric	HCRDa without side information	HCRDa	Improvement(%)
70%	MAE	0.7399 ±0.0053	0.7199 ±0.0031	2.778
	RMSE	0.9394 ±0.0058	0.9176 ±0.0009	2.376
80%	MAE	0.6937 ±0.0055	0.6751 ±0.0007	2.755
	RMSE	0.8855 ±0.0144	0.8645 ±0.0010	2.429
90%	MAE	0.6405 ±0.0092	0.6089 ±0.0099	5.190
	RMSE	0.8241 ±0.0035	0.7879 ±0.0061	4.594

3) IMPACT OF SIDE INFORMATION

The movielens 100K dataset has both user attribute information and item description information, so experiments on the impact of side information are performed on this dataset. From Table 6, we can see that the additional information of this dataset has a certain degree of impact on RMSE and MAE. The impact of additional information is modest, the biggest improvement rate on RMSE is 4.594%. Even without side information, our algorithm outperforms all comparison algorithms in training rates of 80% and 90% for 100K data sets.

Overall, the experimental results on three data sets show that our proposed HCRDa has higher accuracy than start-of-the-art methods, and can better solve the data sparse problem.

V. RELATED WORK

Recommendation system has attracted a vast amount of attention and research in recent decades, and has been applied widely in multiple fields especially in the e-commerce site. In this section, we introduce traditional recommendation system method, deep learning based recommendation system method and autoencoder based recommendation system method.

A. TRADITIONAL RECOMMENDATION SYSTEM

The recommendation system recommends to users which they might like by evaluating the user's preferences for the item [30]. Traditional recommendation system methods can be generally fall into three categories [30]: content-based recommendation methods, collaborative filtering, and hybrid recommendation methods. The content-based recommendation system [31] constructs the user's preference characteristics according to the product information that the user likes in the past, calculates the similarity between the candidate item and the user preference feature, and recommends the most similar item to the user. The content-based recommendation methods face the problems of limited content analysis, overspecialization and new user issues [30]. The collaborative filtering methods discover the user's preferences by mining historical interaction data between user and item, group the users based on different preferences and recommend potential interested items to users. Matrix factorization is the most widely applied algorithm in traditional collaborative filtering methods, which factorizes the rating matrix into two matrices: the feature matrix of user and the feature matrix of item, and one row and one column of each matrix are taken as the inner product for prediction [3]. Since collaborative filtering only considers the information of user's rating, it is easy to cause a cold start problem. Hybrid recommendation methods combine two or more previous recommended methods together for recommendation [31]–[33].

B. DEEP LEARNING BASED RECOMMENDATION SYSTEM

With the substantial performance of deep learning in the fields of computer vision and speech recognition, it demonstrates the ability to learn latent and more powerful feature representations [34]. Recently, deep learning has already been applied in recommendation systems and achieved the encouraging results. Deep learning models commonly used in the recommended system include Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Autoencoder (AE). One typical deep recommendation method based on MLP is neural collaborative filtering (NCF) [5], which is a general framework for collaborative filtering based on neural networks. Most CNN-based recommendation systems utilize their powerful data processing capabilities to extract features from unstructured media data, such as image feature extraction [35], [36], text feature extraction [37], and audio and video feature extraction [38], [39]. RNN has a strong ability to process sequential data, so it is generally used for session-based recommendations [40], [41]. In the CNN based and RNN based recommendation methods, there exist some problems such as requiring a large number of labeled data and long training time.

C. AUTOENCODER BASED RECOMMENDATION SYSTEM

The autoencoder based recommendation methods have achieved sound performance for the superiority of no label requirement and fast convergence speed. The existing autoencoder based recommendation systems can be generally divided into two types [11]: one is to learn the latent feature representations of the user and the item with autoencoder, and the other is to fill the missing value of the original matrix in the reconstruction layer of autoencoder. In the first method, the parameters of autoencoder need to be pre-trained, which increases the time complexity greatly. For example, Zhuang *et al.* proposed the model of Recommendation via Dual-Autoencoder (ReDa) [4], two autoencoders were employed to train the original rating matrix for initializing the bias parameters, and then the gradient descent method was used to optimize the objective function for obtaining the final prediction matrix. The essence of the ReDa model is to obtain the hidden features of users and items by autoencoders. In the second method, the reconstructed value is directly utilized as the predicted value without combining the traditional recommended learning method.

VI. CONCLUSION

In order to meet the needs of users for items, most traditional methods use matrix factorization to obtain the hidden features of users and/or items. With the superiority of deep learning in feature learning, recommendation algorithms based on deep learning spring up. In our proposed framework, we stitched the additional information vectors and rating vectors of users and items as inputs to the two autoencoders, alleviating the cold start problem. Two novel autocoders are used to learn both users and items hidden features to reduce time complexity. In addition, we incorporate the matrix factorization method into the objective function of the training autoencoder to better obtain the hidden features of users and items. Experimental results on three real data sets show the superiority of our method.

At the same time, experimental results show that the deep learning-based recommendation method require a lot of training data, it can achieve better results when the training set is large, but when the training set is small, it may not be as satisfactory as the traditional matrix decomposition recommendation method. In future work, we will try to improve the accuracy of using autoencoder recommendations when the training set is small, possibly through more network layers. We will also consider incorporating more additional information, such as trust values between users and implicit feedback.

REFERENCES

[1] Z.-D. Zhao and M.-S. Shang, "User-based collaborative-filtering recommendation algorithms on Hadoop," in *Proc. 3rd Int. Conf. Knowl. Discovery Data Mining*, Jan. 2010, pp. 478–481, doi: 10.1109/WKDD.2010.54.

[2] J. Jiang, J. Lu, G. Zhang, and G. Long, "Scaling-up item-based collaborative filtering recommendation algorithm based on Hadoop," in *Proc. IEEE World Congr. Services*, Washington, DC, USA, Jul. 2011, pp. 490–497.

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

[4] F. Zhuang, Z. Zhang, M. Qian, C. Shi, X. Xie, and Q. He, "Representation learning via dual-autoencoder for recommendation," *Neural Netw.*, vol. 90, pp. 83–89, Jun. 2017, doi: 10.1016/j.neunet.2017.03.009.

[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*, Perth, WA, Australia, 2017, pp. 173–182.

[6] C. Chen, X. Meng, Z. Xu, and T. Lukasiewicz, "Location-aware personalized news recommendation with deep semantic analysis," *IEEE Access*, vol. 5, pp. 1624–1638, 2017, doi: 10.1109/ACCESS.2017.2655150.

[7] T. M. Phuong, T. C. Thanh, and N. X. Bach, "Neural session-aware recommendation," *IEEE Access*, vol. 7, pp. 86884–86896, 2019, doi: 10.1109/ACCESS.2019.2926074.

[8] H. Liang and T. Baldwin, "A probabilistic rating auto-encoder for personalized recommender systems," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Melbourne, VIC, Australia, 2015, pp. 1863–1866.

[9] S. Zhang, L. Yao, X. Xu, S. Wang, and L. Zhu, "Hybrid collaborative recommendation via semi-autoencoder," in *Proc. Int. Conf. Neural Inf. Process.*, Guangzhou, China, 2017, pp. 185–193.

[10] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "AutoRec: Autoencoders meet collaborative filtering," in *Proc. 24th Int. Conf. World Wide Web-WWW Companion*, Florence, Italy, 2015, pp. 111–112.

[11] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, p. 5, Feb. 2019, doi: 10.1145/3285029.

[12] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Sydney, NSW, Australia, 2015, pp. 1235–1244.

[13] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for Top-N recommender systems," in *Proc. 9th ACM Int. Conf. Web Search Data Mining (WSDM)*, San Francisco, CA, USA, 2016, pp. 153–162.

[14] Y. Chen and M. de Rijke, "A collective variational autoencoder for top-N recommendation with side information," in *Proc. 3rd Workshop Deep Learn. Recommender Syst. (DLRS)*, Vancouver, BC, Canada, 2018, pp. 3–9.

[15] X. Li and J. She, "Collaborative variational autoencoder for recommender systems," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Halifax, NS, Canada, 2017, pp. 305–314.

[16] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, Lyon, France, 2018, pp. 689–698.

[17] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, Bellevue, WA, USA, 2011, pp. 833–840.

[18] 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. (SIGIR)*, Tokyo, Japan, 2017, pp. 957–960.

[19] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Melbourne, VIC, Australia, 2015, pp. 811–820.

[20] Y. Zhu, X. Wu, P. Li, Y. Zhang, and X. Hu, "Transfer learning with deep manifold regularized auto-encoders," *Neurocomputing*, vol. 369, pp. 145–154, Dec. 2019, doi: 10.1016/j.neucom.2019.08.078.

[21] F. Zhuang, D. Luo, N. J. Yuan, X. Xie, and Q. He, "Representation learning with pair-wise constraints for collaborative ranking," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, Cambridge, U.K., 2017, pp. 567–575.

[22] S. Dooms, T. De Pessemier, and L. Martens, "MovieTweetings: A movie rating dataset collected from Twitter," in *Proc. Workshop Crowdsourcing Hum. Comput. Recommender Syst.*, Jan. 2013, p. 43.

[23] J. Golbeck and J. Hendler, "FilmTrust: Movie recommendations using trust in Web-based social networks," in *Proc. 3rd IEEE Consum. Commun. Netw. Conf. (CCNC)*, 2016, vol. 96, no. 1, pp. 282–286.

[24] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[25] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst., 2008*, pp. 1257–1264.

[26] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, Helsinki, Finland, 2008, pp. 880–887.

[27] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Diego, CA, USA, 2008, pp. 426–434.

[28] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1633–1647, Aug. 2017, doi: 10.1109/TPAMI.2016.2605085.

[29] G. Guo, J. Zhang, and N. Yorke-Smith, "TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, 2015, pp. 123–129.

[30] 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, doi: 10.1109/TKDE.2005.99.

[31] Y. Shih and D. Liu, "Product recommendation approaches: Collaborative filtering via customer lifetime value and customer demands," *Expert Syst. Appl.*, vol. 35, nos. 1–2, pp. 350–360, Jul./Aug. 2008, doi: 10.1016/j.eswa.2007.07.055.

[32] B. M. Kim, Q. Li, C. S. Park, S. G. Kim, and J. Y. Kim, "A new approach for combining content-based and collaborative filters," *J. Intell. Inf. Syst.*, vol. 27, no. 1, pp. 79–91, Jul. 2006.

[33] A. Albadvi and M. Shahbazi, "A hybrid recommendation technique based on product category attributes," *Expert Syst. Appl.*, vol. 36, no. 9, pp. 11480–11488, Nov. 2009, doi: 10.1016/j.eswa.2009.03.046.

[34] Y. Zhu, X. Hu, Y. Zhang, and P. Li, "Transfer learning with stacked reconstruction independent component analysis," *Knowl.-Based Syst.*, vol. 152, pp. 100–106, Jul. 2018, doi: 10.1016/j.knosys.2018.04.010.

[35] W. Yu, H. Zhang, X. He, X. Chen, L. Xiong, and Z. Qin, "Aesthetic-based clothing recommendation," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, Lyon, France, 2018, pp. 649–658.

[36] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu, "What your images reveal: Exploiting visual contents for Point-of-Interest recommendation," in *Proc. 26th Int. Conf. World Wide Web (WWW)*, Perth, WA, Australia, 2017, pp. 391–400.

[37] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, Cambridge, U.K., 2017, pp. 425–434.

[38] J. Lee, S. Abu-El-Haija, B. Varadarajan, and A. P. Natsev, "Collaborative deep metric learning for video understanding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, London, U.K., Jul. 2018, pp. 481–490.

[39] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2643–2651.

[40] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proc. 10th ACM Conf. Recommender Syst. (RecSys)*. Boston, MA, USA: ACM, 2016, pp. 241–248.

[41] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM)*, Cambridge, U.K., 2017, pp. 495–503.

BINGBING DONG received the B.S. degree from the Anhui University of Finance and Economics. She is currently a Graduate Student of the Key Laboratory of Knowledge Engineering With Big Data (Hefei University of Technology), and the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. Her research interests are in data mining and big knowledge engineering.

YI ZHU received the B.S. degree from Anhui University, the M.S. degree from the University of Science and Technology of China, and the Ph.D. degree from the Hefei University of Technology. He is currently an Assistant Professor with the School of Information Engineering, Yangzhou University, China. His research interests are in data mining and knowledge engineering.

LEI LI (Senior Member, IEEE) received the bachelor's degree in information and computational science from Jilin University, Changchun, China, in 2004, the master's degree in applied mathematics from the Memorial University of Newfoundland, St. John's, Canada, in 2006, and the Ph.D. degree in computing from Macquarie University, Sydney, Australia, in 2012. He is currently an Associate Professor with the Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), and the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China. He has published over 70 peer-reviewed articles in prestigious journals and top international conferences, including the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON SERVICES COMPUTING, *Knowledge-Based Systems*, *World Wide Web Journal*, *Pattern Recognition Letters*, AAAI, IJCAI, ICDM, ICSOC, and IEEE ICWS. His research interests include graph computing, social computing, data mining, and intelligent computing.

XINDONG WU (Fellow, IEEE) received the Ph.D. degree in artificial intelligence from The University of Edinburgh. He is currently a Chang Jiang Scholar with the School of Computer Science and Information Engineering, Hefei University of Technology, China, and also a Chief Scientist with the Mininglamp Academy of Sciences, Minglamp Technology, Beijing, China. His research interests include data mining, knowledge-based systems, and web information exploration. He is a Fellow of the AAAS. He is also the Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM). He is the Editor-in-Chief of *Knowledge and Information Systems* and of Springer book series, *Advanced Information and Knowledge Processing* (AIKP).