IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 8, 2020, accepted December 22, 2020, date of publication January 25, 2021, date of current version February 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053291

Integrating Stacked Sparse Auto-Encoder Into Matrix Factorization for Rating Prediction

YIHAO ZHANG, CHU ZHAO, MIAN CHEN, AND MENG YUAN
School of Artificial Intelligence, Chongqing University of Technology, Chongqing 400054, China

Corresponding author: Yihao Zhang (yhzhang@cqut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702063, in part by the Natural Science Foundation of Chongqing under Grant cstc2019jcyj-msxmX0544, and in part by the Scientific Research Project of Chongqing Education Commission under Grant KJQN202001136.

ABSTRACT Currently, collaborative filtering technology has been widely used in personalized recommender systems. The problem of data sparsity is a severe challenge faced by traditional collaborative filtering methods based on matrix factorization techniques. A lot of improved collaborative filtering methods have been proposed to alleviate the data sparsity problem; However, due to the sparsity of the user rating matrix, the latent factor learned by these improved methods may be not efficient. In this paper, we propose a novel recommendation algorithm named SSAERec by integrating stacked sparse auto-encoder into matrix factorization for rating prediction, which can learn effective representation from user-item rating matrix. Extensive experiments on three real-world datasets demonstrate the proposed method outperforms other baselines in the rating prediction task.

INDEX TERMS Sparse auto-encoder, collaborative filtering, data sparsity, rating prediction.

I. INTRODUCTION

E-commerce websites recommend new products to potential customers according to their previous ratings on items [1]. Therefore, the accuracy of ratings prediction is critical for E-companies' ability to provide high-quality recommendation services, which leads to increase profile and improve service quality. Matrix Factorization (MF) is one of the well-know collaborative filtering methods [2], [3], which has been successfully utilized to perform rating prediction tasks in recommender systems [4]. By factorizing the user-item rating matrix, the MF method represents users' interests and items' features as latent factor vectors in a common latent space. As improved versions of the basic MF method, these methods of probabilistic matrix factorization (PMF) and non-negative matrix factorization (NMF) can partly alleviate the problem of data sparsity [5]–[7]. However, there are often millions users and items in the E-commerce websites, users only rate each small part of items when faced with a large number of items, so the rating (or interaction) matrix composed of these users and items is extremely sparse. The problem of data sparsity makes the collaborative filtering methods (i.e., the MF method) in recommender system often fail to achieve satisfied experimental results.

In order to solve the problem of data sparsity, researchers have paid extensive attention to efficient feature extraction methods and side information (i.e., social networks, items category etc.) [8], [9]. Traditional matrix factorization models use side information to improve recommendation accuracy [10], [11]; however, it is difficult for those methods to capture complex relationship between users and items as well as learn the effective latent factors from side information, because of the limited learning capacity and the sparse nature of the side information. With the success of deep learning technique in related fields, as an efficient feature extraction tool, it has been employed to learn effective latent representation in recommender systems [12]. Multi-layer perceptron (MLP) as a feed-forward neural architecture with multiple hidden layers, which is adopted to learn hierarchical feature representations from side information to improve recommendation accuracy [13]. Convolution Neural Network (CNN) can be interpreted as an improvement variant of MLP and has been demonstrated success in the fields of computer vision and natural language processing. In addition, in the research of recommender systems, researchers also strive to utilize CNN to perform recommendation tasks,

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

where they employ CNN to capture feature from additional information to solve the data sparsity problem [14]. The graph convolution neural networks are also employed to perform recommendation by utilizing side information (i.e., social network, knowledge graphs, and protein-interaction networks, etc) [15]. Although those methods have achieved encouraging performances than the traditional MF methods [16], [17]; however, the performance of those methods is limited by the quality of side information.

Recently, the auto-encoder has been exploited into recommender systems, which has been demonstrated great success across a wide range of recommendation tasks [18]. Motivated by these methods, a novel rating prediction method based on sparse auto-encoder has been proposed, which can learn effective representation from user-item rating matrix. Experiment results also show that the rating prediction method based on stacked auto-encoder significantly outperforms other baselines on some real-world datasets.

In order to address the data sparsity problem, we propose a novel structure model called stacked sparse auto-encoder for rating prediction in recommender systems (SSAERec). We construct the SSAERec model by stacking multiple sparse auto-encoder to form a deep structure and fusing it into matrix factorization technique, in which the stacked sparse auto-encoders are used to learn latent the latent factors from user-item ratings matrix. The major contributions of this paper are summarized below.

1) We analyze the current methods for solving the data sparsity problem, and present a novel hybrid collaborative filtering model, which incorporates deep representation learning and matrix factorization technique. This model can extract latent factors from user-item ratings matrix and capture implicit relationship between users and items.

2) We propose a novel rating prediction method named SSAERec, which is a variant of sparse auto-encoder by stacking multiple sparse auto-encoder to form a deep structure and integrating it into SVD++ model.

3) We conduct comprehensive experiments on three real-world datasets, which demonstrates that the SSAERec method outperforms other baseline methods in rating prediction, and can efficiently improve the recommendation accuracy and address the problem of data sparsity.

The reminder of the paper is organized as follows. Section II discusses the related work on deep learning methods and auto-encoder techniques used for recommender systems. Section III illustrates the proposed model architecture in detail. In section IV, we give out the experiment results with analysis. Finally, we conclude the paper and explain possible future work.

II. RELATED WORK

With the effort of academia and industry, deep learning techniques have been widely utilized into recommender systems [19]. Researchers have exploited deep learning methods to extract complex features from side information to alleviate the data sparsity problem [20]. Seo *et al.* [21] employ CNN to learn features from review text of items for rating prediction. Kim *et al.* [14] propose convolutional matrix factorization method which integrates convolutional neural network (CNN) into probabilistic matrix factorization (PMF) and uses side information to improve the rating prediction accuracy. Tang *et al.* [22] develop a new neural method called UWCVM which directly exploits user information to predict ratings.

At present, auto-encoder techniques have been proven to be effective for recommender systems, and many researchers have paid extensive attentions to auto-encoder for improving recommendation accuracy. Ouyang *et al.* [23] employ auto-encoder into recommender systems and propose an auto-encoder based collaborative filtering model (ACF), while ACF fails to handle non-integer ratings. Strub and Mary [24] introduce a collaborative filtering neural network model (CFN) in which they utilize stacked denosing auto-encoder to learn more robust features and integrate the side information to alleviate cold start problem. Wu *et al.* [25] design a novel method for top-N recommendation, called collaborative denoising auto-encoder model (CDAE), which is a neural network with single hidden layer. Yi *et al.* [26] propose a supervised neural recommendation (SNR) model, which employs the stacked auto-encoder to extract the features of input and then reconstructs the input to do recommendation as well as the side information is leveraged to improve recommendation performance in framework. Incorporating auto-encoder into traditional recommendation techniques (e.g. matrix factorization, probability matrix factorization, factorization machine) and other deep learning models for recommender systems is a mainstream trend. Li *et al.* [27] combine probabilistic matrix factorization with marginalized denoising stacked auto-encoder to form a deep architecture, which can learn effective latent factor from side information. Dong *et al.* [28] present a novel deep learning model, which is a variant of stacked denoising auto-encoder and can integrate the side information into the learned latent factors efficiently. Zhang *et al.* [29] introduce a new hybrid model by generalizing contractive auto-encoder paradigm into matrix factorization framework to overcome the problem of data sparsity. Chen and Rijke [30] propose to simultaneously recover user ratings and side information by using a variational auto-encoder (VAE). Wang *et al.* [31] combine recurrent neural network (RNN) with auto-encoder for recommendation task and proposed collaborative recurrent auto-encoder (CRAE). Zhang *et al.* [32] present a hybrid deep learning framework termed as Collaborative Knowledge Base Embedding(CKE) in which the stacked denoising anto-encoder and the stacked convolutional auto-encoder are incorporated into model.

Analyzing the current recommendation techniques, we found that existing recommendation algorithms prefer to exploit side information for improving the accuracy of recommendation model; however, those additional information is unavailable in some scenarios. Another serious problem is that directly exploiting user information for recommendation may trigger privacy-related problems. In this paper, we present a novel recommendation architecture by integrating stacked sparse auto-encoder into SVD++ for solving the problem of data sparsity. In implementation, we utilize stacked sparse auto-encoder to extract high-level features from the datasets and incorporate it into matrix factorization model.

III. MODEL ARCHITECTURE

A. LATENT FACTOR MODEL

BiasedSVD is one kind of latent factor models, which is an effective collaboration filtering method. The computed processing of the algorithm is as

$$\hat{r}_{u,i} = \mu + b_u + b_i + p_u {q_i}^T, \tag{1}$$

where μ is the average of rating, and b_u is the bias of users u, which represents irrelevant ratings factor with item in users; b_i is the bias of items i. $p_u \in R^{n\times f}$ and $q_i \in R^{m\times f}$ represent the latent preference of users u and latent property of items i, respectively, f represents the dimensionality of latent factor.

High-quality explicit feedback is the most convenient available information for recommender systems. However, due to the sparsity of user rating matrix, we often cannot obtain clear explicit feedback information. Therefore, the recommender system only utilizes implicit feedback information (e.g., click record, browse record, and purchase history) to build models. SVD++ is a variant algorithm based on SVD [33], which incorporates implicit feedback information in the model. The SVD++ is defined as

$$\hat{r}_{u,i} = \mu + b_u + b_i + {q_i}^T (p_u + |R(u)|^{-\frac{1}{2}} \sum_{j\in R(u)} y_j), \tag{2}$$

where $R(u)$ is a set of items rated by users, which represents implicit feedback provided by users u, $y_i \in R^f$ is the implicit factor vector.

B. SPARSE AUTO-ENCODER

The auto-encoder is an unsupervised machine learning algorithm for feature extraction and dimensionality reduction, which consists of the encoder and decoder component. For each input vector $x^{(i)}$ from a training set $\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$, the encoder $f(\cdot)$ takes given $x^{(i)}$ and maps it to a hidden representation $h^{(i)}$, then the decoder $g(\cdot)$ maps the hidden representation back to a reconstructed version of $x^{(i)}$. The process is defined as (3) and (4), where W_1 and W_1' are weight matrices, b_1 and b_1' are bias vectors. Given a training set of m instances, we define the loss function as

$$h^{(i)} = f(W_1 x^{(i)} + b_1), \tag{3}$$

$$y^{(i)} = g(W_1' h^{(i)} + b_1'), \tag{4}$$

$$J(W, b) = \left[\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\|y^{(i)} - x^{(i)}\|^2\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(W_{ji}^{(l)})^2, \tag{5}$$

In the (5), the first term represents the average sum-of-squares error and the second term is the regularization term which is utilized to prevent over-fitting. In the (5), λ represents the weight decay parameter, n_l represents the number of layers, s_l represents the neuron number in layer l, and $W_{ji}^{(l)}$ is the connect weight matrix.

The parameters of auto-encoder can be fixed by minimizing error of objection function, then the model will output the reconstruct feature $g(f(x))$, which remains a majority of information of original data and approximately equal input data, such that $g(f(x)) \approx x$.

However, in extreme cases, the auto-encoder can perfectly reconstruct the input data, and it is impossible to learn effective functions from the data by using simple identification functions. With adding constraints in the hidden layer to force the hidden layer to be different from the input layer, the auto-encoder model can reconstruct the input data and learn more robust feature representations.

Sparse auto-encoder is an improved auto-encoder method, which increases some sparsity restrictions in the hidden layer of traditional auto-encoder. By limiting the output of most hidden layers, the auto-encoder network can achieve a sparsity effect. According to different activation functions, the concept of sparsity is distinctive. For example, when the output of hidden layer of sigmoid function is closed to 0, the model network is considered to be constrained; and the network is considered to be constrained when the output of tanh function is closed to -1. In order to achieve the sparsity effect, the sparse auto-encoder will constrain the average activation value of the neuron output in the hidden layer and utilizes the KL divergence to force it to be closed to a default value, which will be added to the loss function as a penalty. The loss function of sparse auto-encoder is defined as

$$J_{spare}(W, b) = J(W, b) + \beta \sum_{i=1}^{m} KL(\rho\|\hat{\rho}_i), \tag{6}$$

where β represents the weight used to control sparsity penalty factor, which is a random value in [0, 1]. $\sum_{j=1}^{m} KL(\rho\|\hat{\rho})$ is defined as

$$\sum_{j=1}^{m} KL(\rho\|\hat{\rho}) = \sum_{j=1}^{m} \rho \log\frac{\rho}{\hat{\rho}_i} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_i}, \tag{7}$$

where $\rho = \frac{1}{m}\sum_{i=1}^{m}(a_j(x_i))$ represents the average activation of all training instances in hidden layer neuron j, a_j is the activation value in hidden layer neurons, and m represents the number of hidden units per layer.

C. STACKED SPARSE AUTO-ENCODER

Existing studies have demonstrated that stacking multiple layers together can generate abundant feature representation in hidden layers, and therefore leads to better performance for various learning tasks. The marginalized stacked denoising auto-encoders (mDAE) stacks several denoising auto-encoders together to learn effective latent representation [34], inspired by mDAE, we stack multiple sparse auto-encoders to form a deep network structure which termed as stacked sparse auto-encoder (SSAE). For each hidden layer $l \in \{1, \ldots, L-1\}$ of the SSAE model, the hidden layer output y_l is defined as follow:

$$y_l = f(W_h h_{L-1} + b_l), \tag{8}$$

where L is the total number of layer. Note that the encoder part of model consists of the first $L/2$ layers and the last $L/2$ layers of the model is decoder. In SSAE model,we assume that only one hidden layer should be close to the latent factor, and the $L/2$ layer will generate the latent factor. The SSAE utilizes a deep network to reconstruct the input and minimize the squared loss between the reconstructed outputs. The loss function for SSAE is computed as follow, which is similar to (6):

$$J_{spare}(W_l, b_l) = J(W_l, b_l) + \beta \sum_{i=1}^{m} KL(\rho \| \hat{\rho}_i), \tag{9}$$

we employ back-propagation algorithm to learn weight matrix W_l and bias vector b_l of each layer.

D. OUR RECOMMENDATION MODEL

It is difficult for traditional latent factor model to extract efficient and effective feature representation from rating data. Therefore, we are preparing to utilize the SSAE method to extract feature representations as

$$SSAE(M_i) = s(W_i M_i + b_i), \tag{10}$$

where SSAE(M_i) represents the low-dimension feature representation which is extracted by SSAE method, and M_i is the original feature vector. The extracted feature representation will be used to replace item latent vector in SVD algorithm. The processing of our algorithm can be described as

$$\hat{r}_{u,i} = \mu + b_u + b_i + p_u(\beta \cdot SSAE(M_i) + \alpha_i)^T, \tag{11}$$

where β is a hyper parameter to normalize $SSAE(M_i)$, α_i is the latent item-based offset vector. Decomposing user latent vector is feasible; however, due to the limit of user privacy agreement, which is not sensible to utilize users' profile.

Although SVD algorithm has utilized compact and efficient feature representations extracted by SSAE method, the performance of algorithm is still poor and cannot produce satisfied recommendation accuracy. In order to solve these problems, we propose a novel deep recommendation algorithm integrating SSAE algorithm to SVD++ algorithm, which takes the implicit feedback information into account for mitigating the data sparsity problem. We show the model structure of SSAERec in Fig. 1. The recommendation model based on SSAE method is formulated as

$$\hat{r}_{u,i} = \mu + b_u + b_i + (\beta \cdot SSAE(M_i) + \alpha_i)^T (p_u + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j), \tag{12}$$

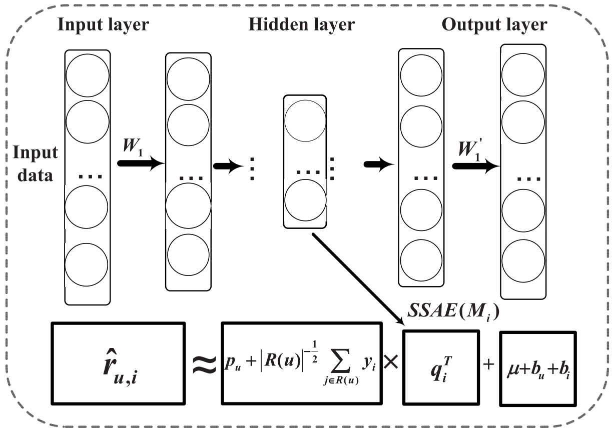


FIGURE 1. The architecture of stacked sparse auto-encoder SVD++ model.

E. TRAINING AND OPTIMIZATION

The task of our model is to predict the user's ratings of items. The input of the model is a sparse rating matrix, and the corresponding output is the users' missing rating for items. The parameters of our model are trained by minimizing regularized squared error loss, and the loss function is

$$minLoss = \sum (r_{u,i} - \hat{r}_{u,i})^2 + \lambda \cdot Z, \tag{13}$$

where λ is the learning rate, Z represents regularization terms that is employed to prevent overfitting and gradient explosion problem. The term Z is described as

$$Z = b_u^2 + b_i^2 + \|\alpha\|^2 + \|p_u\|^2 + \sum_{j \in R(u)} \|y_j\|^2. \tag{14}$$

Our model consists of two parts: feature extraction and rating prediction. Optimization method of adaptive moment estimation (Adam) is used to optimize stacked sparse auto-encoder, which is an algorithm for first-order gradient-based optimization of stochastic objective functions based on adaptive estimates of lower-order moments and can update neural network weights iteratively based on training data. For rating prediction model, parameters can be learned by stochastic gradient descent (SGD). In implementation, we firstly utilize SGD algorithm to compute the prediction error as

$$e_{u,i} = \hat{r}_{u,i} - r_{u,i}, \tag{15}$$

then, we update parameters by choosing a opposite direction of gradient to move. The update rules for SSAERec are as follow:

$$b_u := b_u + \gamma_1 (e_{u,i} - \lambda_1 \cdot b_u) \tag{16}$$

$$b_i := b_i + \gamma_1 (e_{u,i} - \lambda_1 \cdot b_i) \tag{17}$$

$$p_u := p_u + \gamma_2 (e_{u,i} \cdot (\beta \cdot cae(M_I) + \alpha_i) - \lambda_2 \cdot p_u) \tag{18}$$

$$\alpha_i := \alpha_i + \gamma_2 (e_{u,i} \cdot (p_u + |R(u)|^{-\frac{1}{2}} \sum_{j \in R(u)} y_j) - \lambda_2 \cdot \alpha_i) \tag{19}$$

$$\forall j \in R(u): y_j := y_j + \gamma_2 \cdot (e_{u,i} \cdot |R(u)|^{-\frac{1}{2}} \cdot (\beta \cdot SAE(M_i) + \alpha_i) - \lambda_2 \cdot y_i) \quad (20)$$

where γ_1 and γ_2 represent different learning rate, λ_1 and λ_2 are regularization weights. By setting the value of parameters, the relative importance of each feedback will be decided and computed automatically in the model.

IV. EXPERIMENTS

A. EVALUATION DATASETS

We employ three datasets which come from different domains to demonstrate our model, i.e., Ciao,[1] MovieLens-100k and MovieLens-1M,[2] which are usually used to evaluate collaboration filtering algorithms in recommender system filed. The MovieLens-100k dataset consists of 100k ratings, 943 users and 1682 movies. The MoiveLens-1M dataset contains one million ratings from 6040 users and 3706 movies. The Ciao dataset contains 700k ratings from 17615 users and 16121 items, which is a new movie rating dataset collected from Ciao website.[3] The MovieLens-100k and MovieLens-1M datasets are most frequently used in many excellent papers and the Netflix competition. Each rating is a number between 1 (worst) and 5 (best), and all datasets are extremely sparsity. The detailed statistics of selected datasets are presented in Table1.

TABLE 1. Datasets statistics.

Datasets	Users#	Items#	Ratings#	Sparsity
Ciao	17515	16121	72665	99.62%
ml-100k	943	1682	60089	93.72%
ml-1m	6040	3706	1000209	95.53%

B. EVALUATION METRICS

In order to evaluate the performance of algorithm, we split each dataset into training set and test set with different split percentage. We employ two commonly rating evaluation metrics (i.e., Root Mean Square Error (*RMSE*) and Mean Absolute Error (*MAE*)) to evaluate our model. *RMSE* is used to measure the deviation between observed value and true value:

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i)\in T} (\hat{r}_{u,i} - r_{u,i})^2} \quad (21)$$

where $|T|$ is the number of ratings in the testing dataset, $r_{u,i}$ denotes the ground-truth rating of user u for item i, $\hat{r}_{u,i}$ is the prediction rating calculated by recommendation algorithm. MAE is used to measure the average absolute error, which is defined as follows:

$$MAE = \frac{\sum_{(u,i)\in|T|} (\hat{r}_{u,i} - r_{u,i})}{|T|} \quad (22)$$

[1]https://www.librec.net/datasets.html
[2]https://grouplens.org/datasets/movielens/1m/
[3]http://dvd.ciao.co.uk

C. BASELINES

We leverage the python library of tensorflow to implement our SSAERec algorithm, and compare it with the following four baseline algorithms:

- PMF [5]: The algorithm assumes that feature matrix of user u and item v obeys Gaussian distribution by maximum posterior probability (MAP) and maximum likelihood estimate (MLE).
- ConvMF [14]: a novel context-aware hybrid recommendation model, which consists of Convolutional Neural Network (CNN) and Probabilistic Matrix Factorization (PMF) model. ConvMF employs CNN to extract feature from dataset and integrates it into PMF to enhance the rating prediction accuracy.
- DHA-RS [35]: DHA-RS is a hybrid deep learning framework, which incorporates stacked auto-encoder with neural collaborative filtering and utilize auxiliary information to predict user preferences.
- CAVAE [36]: CAVAE integrates additional variational auto-encoder and probabilistic matrix factorization and is a hybrid deep learning model.

We select the above four methods which are usually used in rating prediction field. The PMF is representative of traditional collaborative filter method. The ConvMF is the state-of-the-art hybrid recommendation model by fusing PMF and CNN. The DHA-RS combines stacked auto-encoder with neural collaborative filtering. The CAVAE model integrates auto-encoder network structure into probabilistic matrix factorization technique to enhance rating prediction accuracy.

In order to verify the validity of each part of SSAERec, we design the ablation experiment and are preparing to compare SSAERec with the following two algorithms:

- SVD++ [33]: SVD++ is a combined model that fuses the advantages of both neighborhood and latent factor approaches in the latent space.
- SAERec: This model is a basic version of SSAERec, which integrates sparse auto-encoder (only one hidden layer) with SVD++.

D. EXPERIMENTAL RESULTS AND ANALYSIS

We implement our SSAERec algorithm using the python library of tensorflow.[4] In our experiments, we set different proportion of training dataset and test dataset to demonstrate our model. For the hyper-parameter of SSAERec, $\gamma_1 = 0.001$, $\gamma_2 = 0.005$, $\lambda_1 = 0.001$, $\lambda_2 = 0.008$ and $\beta = 0.1$. For ConvMF, DHA-RS and CAVAE methods, the model parameters are set according to their mentioned in the author's paper. Each method is executed five times, and we take the average RMSE and MAE as the result. The performance of RMSE and MAE to all adopted methods is described in table 2 and table 3.

Table 2 shows the average RMSE of PMF, ConvMF, DHA-RS, CAVAE and Our SSAERec with different percentage

[4]https://github.com/haomiaocqut/ReSys_SSAERec

TABLE 2. The average RMSE of our algorithm compared with other baselines, with different percentages of training data on ml-100K and ml-1m datasets.

Method	90%		80%		50%	
	ml-100k	ml-1m	ml-100k	ml-1m	ml-100k	ml-1m
PMF	0.9343±0.0127	0.9185±0.0045	0.9234±0.0030	0.8934±0.0027	0.9498±0.0024	0.9056±0.0015
CovnMF	0.9143±0.0089	0.8567±0.0067	0.9225±0.0067	0.8659±0.0031	0.9167±0.0034	0.9012±0.0012
DHA-RS	0.9221±0.0024	0.8432 ±0.0056	0.9132±0.0053	0.8526±0.045	0.9217±0.0044	0.8807±0.0023
CAVAE	0.9034±0.0017	0.8623 ±0.0028	0.9157±0.0046	0.8575±0.045	0.9134±0.0035	0.8705±0.0017
SSAERec	**0.9021±0.0023**	**0.8521±0.0063**	**0.9123±0.0047**	**0.8478±0.0032**	**0.9159±0.0021**	**0.8643±0.0011**

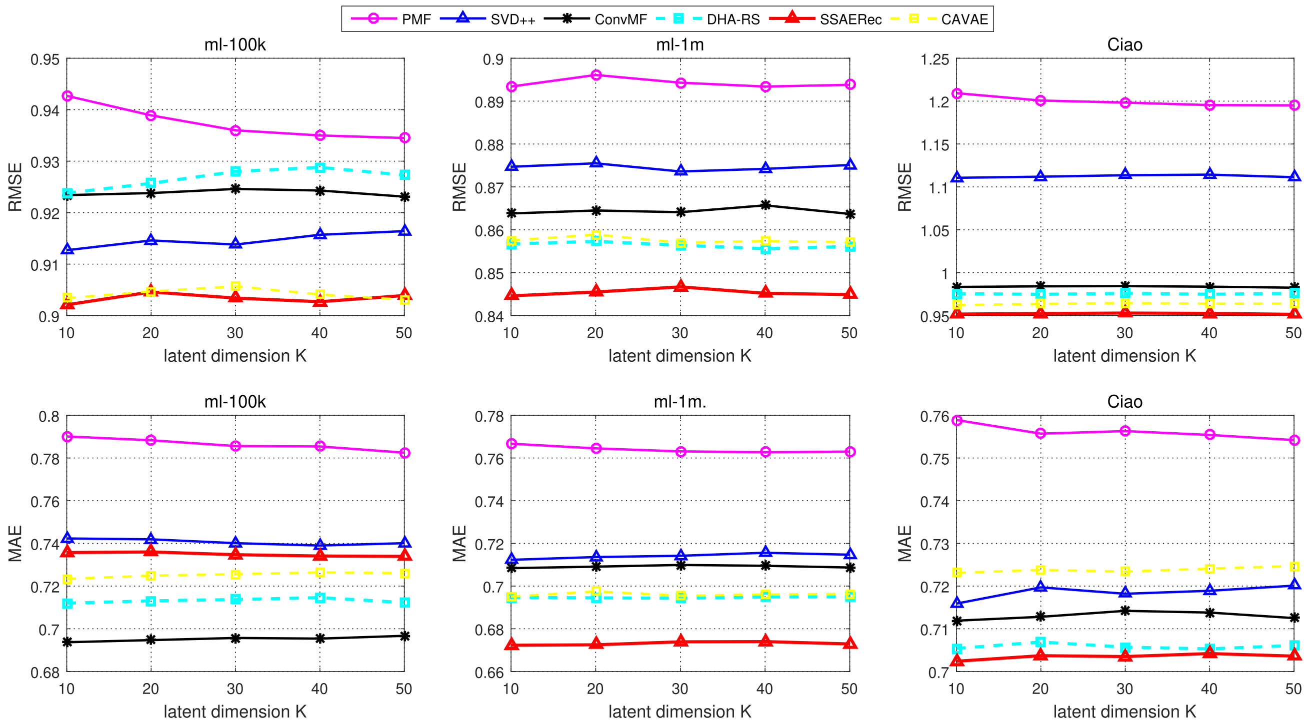


FIGURE 2. The RMSE and MAE performance of our algorithm compared with other baselines on three datasets, with latent dimension ranging between 10 and 50.

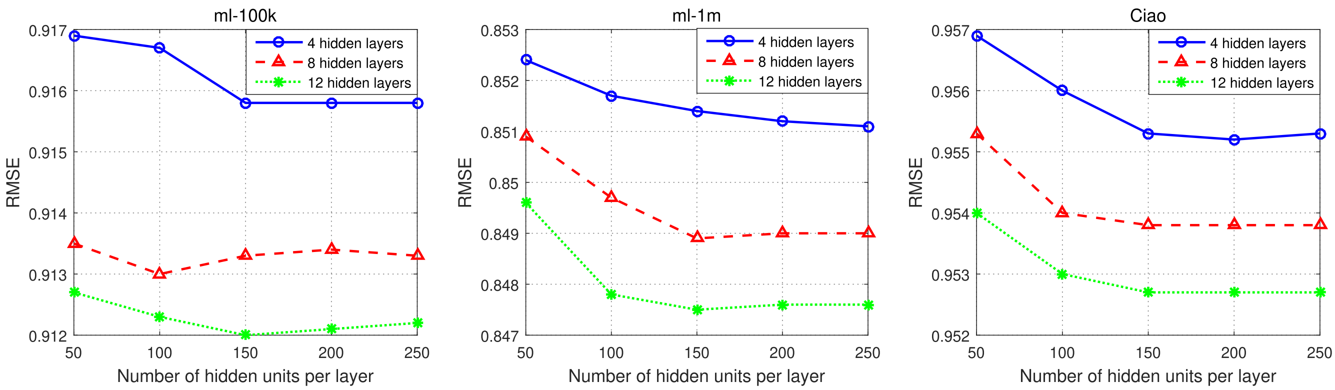


FIGURE 3. The RMSE performance of SSAERec with different hidden layers and the hidden units per layer.

of training data on two different real-world datasets. Analyzing the result, we can find that SSAERec achieves better performance than other baseline algorithms. In detail, on ml-1m dataset (when the proportion of training dataset is 80%), the RMSE metric of our SSAERec has decreased to 0.8478, and the improvement is 2.09% compared with state-of-the-art algorithm (ConvMF). The improvement is 1.86% and 1.13% compared with state-of-the-art algorithms

TABLE 3. The average MAE of compared algorithms with different percentages of training data on ml-100K, ml-1m and Ciao datasets.

Method	ml-100k	ml-1m	Ciao
PMF	0.792	0.762	0.843
ConvMF	0.693	0.708	0.711
DHA-RS	0.710	0.694	0.705
CAVAE	0.721	0.695	0.723
SSAERec	**0.723**	**0.672**	**0.702**

TABLE 4. The average RMSE of compared algorithms on ml-100K, ml-1m and Ciao datasets.

Method	SVD++	SAERec	SSAERec	Improvement	
ml-100k	0.9327	0.9215	**0.9123**	2.19%	1.00%
ml-1m	0.8734	0.8620	**0.8478**	2.93%	1.65%
Ciao	1.0021	0.9723	**0.9534**	4.86%	1.94%

(i.e., DHA-RS and CAVAE) on ml-1m dataset (when the proportion of training dataset is 50%). Especially, the deep structure such as ConvMF, DHA-RS and CAVAE outperform transitional MF methods, which demonstrates deep structure can learn effective latent factor from data.

Table 3 shows the performance of all adopted algorithms in the metric MAE. It is easy to see that our SSAERec algorithm performs better than other baseline algorithms, our SSAERec model achieves lower MAE on ml-1m dataset and Ciao dataset compared with other four algorithms. Such as, the MAE metric of our SSAERec has reduce to o.723, 0.672 and 0.702 on ml-100k, ml-1m and Ciao dataset respectively. The improvement is 12.08%, 12.00% and 16.73% compared with traditional matrix factorization method (PMF) on above three datasets. However the ConcMF and DHA-RS methods work better than our SSAERec model with MAE 0.693 and 0.721 respectively. To sum up, the MAE metric in experimental results show it is effective that we proposed stacked sparse auto-encoder SVD++ for improving prediction accuracy without auxiliary information.

Table 4 shows the ablation experiment results, we compare SVD++, SAERec and SSAERec on ml-1m dataset with RMSE metric. Analyzing the results, we can see that SAERec and SSAERec algorithms achieve the best performance compared with traditional matrix factorization method SVD++. At smae time, comparing SSAERec with SAERec the RMSE metric reduces from 0.8620 to 0.8478 and the improvement is 1.65% on ml-1m dataset, which indicates the effectiveness of our proposed deep learning model SSAERec.

Fig. 2 shows the performance of all the competitive algorithms with respect to different numbers of latent dimension. We test the values of dimension as [10,20,30,40,50], respectively. Analyzing the result from Fig. 2, We have almost the same conclusion, regardless of the latent dimension setting, SSAERec algorithm is far superior to all other algorithms on adopted three datasets. On ml-100k dataset, our SSAAERec model is weaker the ConvMF algorithm in MAE metric, but our SSAERec algorithm performs far better than the ConvMF algorithm in RMSE metric. In detail, the RMSE metric of SSAERec algortithm reduce to 0.8478 on ml-1m, which is far better than other algorithms.To sum up, the result demonstrates that stacked sparse auto-encoder can effectively mitigate the data sparsity problem and improve rating prediction accuracy without side information.

In Fig. 3, on ml-1m dataset, we show the effect of different stacked hidden layers on the performance of our SSAERec algorithms with RMSE metric. Analyzing the results from Fig. 3, it is easy to see that as we increase the number of hidden layers and the number of hidden units per layer, the RMSE metric keep reducing. In detail, when stacked 12 hidden layers together, our SSAERec achieve best performance on three datasets and the RMSE metric reduce to 0.9210, 0.8475 and 0.9534, respectively. And we also find that leveraging 100 or 150 hidden units per layer is suitable for our SSAERec algorithm.

V. CONCLUSION

In this paper, we propose a deep collaborative filtering framework (SSAERec), which incorporates the latent matrix factorization and the stacked sparse auto-encoder. SSAERec is a hybrid collaborative filtering model and learns latent factor representation from user-item ratings. In addition, Stacked sparse auto-encoder is utilized to extract item representation and SVD++ is used to further incorporate the feedback information. Extensive experiment results on Ciao, MovieLens-100k and MovieLens-1M demonstrate that the effectiveness of the latent factor representation learned by our model. Our model achieves better performance than existing deep learning hybrid models such as ConvMF, and outperforms other related algorithms on datasets.

In future, we are particularly interested in integrating attention mechanism into auto-encoder. The attention mechanism can focus on and highlight specific parts of the input data, which has been successfully utilized in computer vision and natural language processing.

REFERENCES

[1] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems: Introduction and Challenges*. Boston, MA, USA: Springer, 2015, pp. 1–34.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[3] Z.-D. Chen, C.-X. Li, X. Luo, L. Nie, W. Zhang, and X.-S. Xu, "SCRATCH: A scalable discrete matrix factorization hashing framework for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2262–2275, Jul. 2020.

[4] D. Bokde, S. Girase, and D. Mukhopadhyay, "Matrix factorization model in collaborative filtering algorithms: A survey," *Procedia Comput. Sci.*, vol. 49, pp. 136–146, Jan. 2015.

[5] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. NIPS*, Vancouver, BC, Canada, 2008, pp. 1257–1264.

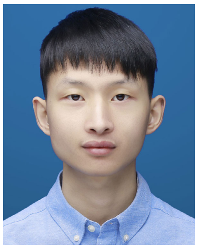
[6] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*, Vancouver, BC, Canada, 2001, pp. 556–562.

[7] N. F. Al-Bakri and S. H. Hashim, "Reducing data sparsity in recommender systems," *Al-Nahrain J. Sci.*, vol. 21, no. 2, pp. 138–147, Jun. 2018.

[8] L. H. Son, "Dealing with the new user cold-start problem in recommender systems: A comparative review," *Inf. Syst.*, vol. 58, pp. 87–104, Jun. 2016.

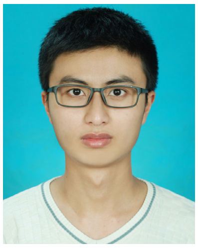
[9] Z. Sun, Q. Guo, J. Yang, H. Fang, G. Guo, J. Zhang, and R. Burke, ''Research commentary on recommendations with side information: A survey and research directions,'' *Electron. Commerce Res. Appl.*, vol. 37, Sep. 2019, Art. no. 100879.
[10] R. P. Adams, G. E. Dahl, and I. Murray, ''Incorporating side information in probabilistic matrix factorization with Gaussian process,'' in *Proc. 26th Conf. UAI*, Catalina Island, CA, USA, Jul. 2010, pp. 1–9.
[11] T. Zhao, J. McAuley, and I. King, ''Leveraging social connections to improve personalized ranking for collaborative filtering,'' in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, Shanghai, China, Nov. 2014, pp. 261–270.
[12] S. Zhang, L. Yao, A. Sun, and Y. Tay, ''Deep learning based recommender system: A survey and new perspectives,'' *ACM Comput. Surveys*, vol. 52, no. 1, pp. 1–38, Feb. 2019.
[13] Z. Xu, C. Chen, T. Lukasiewicz, Y. Miao, and X. Meng, ''Tag-aware personalized recommendation using a deep-semantic similarity model with negative sampling,'' in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Indianapolis, IN, UAS, Oct. 2016, pp. 1921–1924.
[14] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, ''Convolutional matrix factorization for document context-aware recommendation,'' in *Proc. 10th ACM Conf. Recommender Syst.*, Boston, MA, USA, Sep. 2016, pp. 233–240.
[15] R. van den Berg, T. N. Kipf, and M. Welling, ''Graph convolutional matrix completion,'' 2017, *arXiv:1706.02263*. [Online]. Available: http://arxiv.org/abs/1706.02263
[16] Z. Cheng, X. Chang, L. Zhu, R. C. Kanjirathinkal, and M. Kankanhalli, ''MMALFM: Explainable recommendation by leveraging reviews and images,'' *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–28, Mar. 2019.
[17] X. He, T. Chen, M.-Y. Kan, and X. Chen, ''TriRank: Review-aware explainable recommendation by modeling aspects,'' in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Melbourne, VIC, Australia, Oct. 2015, pp. 1661–1670.
[18] G. Zhang, Y. Liu, and X. Jin, ''A survey of autoencoder-based recommender systems,'' *Frontiers Comput. Sci.*, vol. 14, no. 2, pp. 430–450, Apr. 2020.
[19] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, ''A review on deep learning for recommender systems: Challenges and remedies,'' *Artif. Intell. Rev.*, vol. 52, no. 1, pp. 1–37, Jun. 2019.
[20] Y. Chen, X. Zhao, and M. de Rijke, ''Top-N recommendation with high-dimensional side information via locality preserving projection,'' in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tokyo, Japan, Aug. 2017, pp. 985–988.
[21] S. Seo, J. Huang, H. Yang, and Y. Liu, ''Interpretable convolutional neural networks with dual local and global attention for review rating prediction,'' in *Proc. 11th ACM Conf. Recommender Syst.*, Como, Italy, Aug. 2017, pp. 297–305.
[22] D. Tang, B. Qin, T. Liu, and Y. Yang, ''User modeling with neural network for review rating prediction,'' in *Proc. 24th IJCAI*, 2015, pp. 1340–1346.
[23] Y. Ouyang, W. Liu, W. Rong, and Z. Xiong, ''Autoencoder-based collaborative filtering,'' in *Proc. 21th ICONIP*, Kuching, Malaysia, 2014, pp. 284–291.
[24] F. Strub and J. Mary, ''Collaborative filtering with stacked denoising autoencoders and sparse inputs,'' in *Proc. NIPS Workshop Mach. Learn. eCommerce*, Montreal, QC, Canada, 2015, pp. 150–158.
[25] 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*, San Francisco, CA, USA, Feb. 2016, pp. 153–162.
[26] B. Yi, X. Shen, Z. Zhang, J. Shu, and H. Liu, ''Expanded autoencoder recommendation framework and its application in movie recommendation,'' in *Proc. 10th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Chengdu, China, 2016, pp. 298–303.
[27] S. Li, J. Kawale, and Y. Fu, ''Deep collaborative filtering via marginalized denoising auto-encoder,'' in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Melbourne, VIC, Australia, Oct. 2015, pp. 811–820.
[28] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, ''A hybrid collaborative filtering model with deep structure for recommender systems,'' in *Proc. 31th AAAI*, New York, NY, USA, 2017, pp. 1309–1315.
[29] S. Zhang, L. Yao, and X. Xu, ''AutoSVD++: An efficient hybrid collaborative filtering model via contractive auto-encoders,'' in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Tokyo, Japan, Aug. 2017, pp. 957–960.
[30] 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.*, Vancouver, BC, Canada, Oct. 2018, pp. 3–9.
[31] H. Wang, X. Shi, and D. Y. Yeung, ''Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks,'' in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 415–423.
[32] F. Zhang, N. J. Yuan, D. Lian, X. Xie, and W.-Y. Ma, ''Collaborative knowledge base embedding for recommender systems,'' in *Proc. 22nd Int. Conf. Knowl. Discovery Data Mining (ACM SIGKDD)*, San Francisco, CA, USA, Aug. 2016, pp. 353–362.
[33] Y. Koren, ''Factorization meets the neighborhood: A multifaceted collaborative filtering model,'' in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Las Vegas, NV, USA, 2008, pp. 426–434.
[34] M. Chen, Z. Xu, K. Weinberger, and F. Sha, ''Marginalized denoising autoencoders for domain adaptation,'' in *Proc. 29th ICML*, Edinburgh, U.K., 2012, pp. 1–8.
[35] L. Yu, W. Shuai, and M. S. Khan, ''A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering,'' *Big Data Mining Anal.*, vol. 1, no. 3, pp. 211–221, 2018.
[36] M. He, Q. Meng, and S. Zhang, ''Collaborative additional variational autoencoder for Top-N recommender systems,'' *IEEE Access*, vol. 7, pp. 5707–5713, 2019.

YIHAO ZHANG received the B.S. degree from Xinyang Normal University, in 2006, the M.S. degree from the Kunming University of Science and Technology, in 2010, and the Ph.D. degree from Chongqing University, in 2014, all in computer science. He worked as a Visiting Scholar with the National University of Singapore, in 2019. He is currently an Associate Professor with the School of Artificial Intelligence, Chongqing University of Technology. His research interests include recommendation systems, natural language processing, and machine learning.



CHU ZHAO received the B.S. degree from the Chongqing University of Education, in 2019. He is currently pursuing the master's degree with the School of Artificial Intelligence, Chongqing University of Technology. His research interests include machine learning and recommendation systems.

MIAN CHEN was born in Chaozhou, China, in 1995. She received the B.S. degree from the Guangdong University of Finance and Economics, in 2018. She is currently pursuing the master's degree with the Chongqing University of Technology, under the supervision of Assoc. Prof. Zhang. Her research interests include machine learning and recommendation systems.



MENG YUAN received the B.S. degree from Kunming University, in 2019. He is currently pursuing the master's degree with the School of Artificial Intelligence, Chongqing University of Technology. His research interests include machine learning and recommendation systems.