

# Auxiliary Stacked Denoising Autoencoder based Collaborative Filtering Recommendation

Ruihui Mu<sup>1,2</sup>, Xiaoqin Zeng<sup>1</sup>

<sup>1</sup>College of Computer and Information, Hohai University, Nanjing 210098, P.R.China  
[e-mail: muruihui@126.com; zxq@hhu.edu.cn]

<sup>2</sup>College of Computer and Information Engineering, Xinxiang University, Xinxiang 453000, P.R.China

\*Corresponding author: Ruihui Mu

*Received January 5, 2020; revised March 13, 2020; accepted April 22, 2020;*

*published June 30, 2020*

---

## **Abstract**

In recent years, deep learning techniques have achieved tremendous successes in natural language processing, speech recognition and image processing. Collaborative filtering(CF) recommendation is one of widely used methods and has significant effects in implementing the new recommendation function, but it also has limitations in dealing with the problem of poor scalability, cold start and data sparsity, etc. Combining the traditional recommendation algorithm with the deep learning model has brought great opportunity for the construction of a new recommender system. In this paper, we propose a novel collaborative recommendation model based on auxiliary stacked denoising autoencoder(ASDAE), the model learns effective the preferences of users from auxiliary information. Firstly, we integrate auxiliary information with rating information. Then, we design a stacked denoising autoencoder based collaborative recommendation model to learn the preferences of users from auxiliary information and rating information. Finally, we conduct comprehensive experiments on three real datasets to compare our proposed model with state-of-the-art methods. Experimental results demonstrate that our proposed model is superior to other recommendation methods.

---

**Keywords:** Auxiliary information, collaborative filtering, data sparsity, recommender system, stacked denoising autoencoder

## 1. Introduction

With the increasing amount of information on the Internet, it has become increasingly difficult to find valuable and relevant information from vast amounts of data [1-2]. Recommender systems [3-6] have become effective tool to solve the problem of information overload by analyzing user's historical ratings, mining potential user preferences, and providing personalized recommendation services. In recent years, it has received extensive attention from academic and industrial. It has been applied in many fields, such as Amazon and Taobao, Netflix, Last.fm, Google, etc. [7] These Internet applications or e-commerce platform by the deployment of recommender systems, one aspect, it satisfies customer's individualized needs and reduces the problem of information overload. On the other hand, it increases user's loyalty and company's profits.

In recommender systems, collaborative filtering(CF) [8-11] methods are the most widely used recommendation techniques. CF methods predict user preferences by analyzing user's historical feedback information. Because CF methods only need user's feedback information and do not require user's profile information, CF methods are independent of specific application domain and can be generalized to different application domains. However, there are serious problems such as data sparsity and cold start in CF methods [12]. Data sparsity causes CF methods cannot calculate the similarity accurately between users or items, so seriously affects the predictive accuracy. Cold start problem refers to the fact that CF methods cannot find similar user or item accurately due to few rating data related to newly registered user and newly added item, and thus cannot provide personalized recommendations for newly registered users, or newly added items [13-15]. Recently, some hybrid recommendation methods [16] have been proposed by fusing auxiliary information such as users or items side information with CF methods.

Nowadays, deep learning has been applied in recommender systems [17]and achieved great successes. Deep learning can automatically extract low-level features from multi-source heterogeneous data and combine them into high-level features with higher representation. During this low-to-high learning process, the features change from sparse to dense, the hidden rules in the data will be automatically mined, which solves the problem of manual feature extraction in traditional recommender systems. Deep learning can take multi-source heterogeneous data as input and map different data to the same relatively low-dimensional feature space through an end-to-end non-linear deep neural network structure framework, thereby obtaining a unified low-dimensional representation of the data. On this basis, fusion with traditional recommendation algorithms can effectively utilize heterogeneous auxiliary information from multiple sources to alleviate the cold start and data sparsity problems in traditional recommender systems.

Autoencoder is one of the most representative deep learning methods for effectively learning user's preference, especially stacked denoising autoencoder (SDAE), have strong feature representation learning capabilities. They are mainly used to learn the latent feature representation of users or items through rating information of users or items in hybrid recommender systems based on autoencoders, then incorporate the latent feature representation into traditional recommender systems. Wang et al. [18] proposed autoencoders(AE) to perform CF recommendation by explicit data and Strub et al. [19] also used autoencoder(AE) to perform CF recommendation by implicit data. A collaborative denoising autoencoder(CDAE) model was proposed by Wu et al. [20] to perform CF recommendation. A correlative denoising autoencoder (DAE) model was proposed by Pan et al. [21] to perform CF recommendation by fusing ratings and trust information.

Although those methods achieve great recommendation performance, there are still exist the data sparsity and cold start problem. Those methods are difficult to learn user's preferences accurately by only using single information as input. In order to alleviate these problems, auxiliary information can be used as supplement to rating information to improve the recommendation performance.

To make best use of auxiliary information, we proposed a novel collaborative filtering recommendation model based on auxiliary stacked denoising autoencoder(ASDAE) which integrates auxiliary information with rating information as inputs to improve the accuracy of recommendation. The main contribution of this paper includes:

- 1) We proposed a novel method named auxiliary stacked denoising autoencoder based collaborative filtering recommendation, which integrates auxiliary information with rating information as inputs and learns effective user's preferences accurately.
- 2) Compared to the previous works, we make the best use of autoencoder model by fusing auxiliary information with rating information into SDAE, this model can significant alleviates the cold start problem and data sparsity problem.
- 3) In order to validate the performance of our proposed model, we conduct comprehensive experiments on three datasets ([MovieLens-100K](#), [MovieLens-1M](#) and [Epinions](#)) to make comparisons with state-of-the-art recommendation methods. Experimental results demonstrate that our proposed model is superior to other baseline methods.

The paper is organized as follows. we first give a brief overview of related literatures of autoencoders in Section 2. Then, we detail our proposed model completely in Section 3. Next, we describe and analyze experimental results in section 4. Finally, Section 5 concludes our work and outlines potential future work.

## 2. Related Work

In this section, we mainly introduce autoencoder(AE), denoising autoencoder(DAE) and stacked denoising autoencoder (SDAE).

### 2.1 Autoencoder

In 1986, Rumelhart et al. proposed autoencoder [22] (AE) for processing of high dimensional complex data. AE is an unsupervised machine learning algorithm that reconstructs the input data through two processes (Encoder and Decoder) to learn the latent feature of data. AE is a non-cyclic feedforward three-layer neural network structure: an input layer, a hidden layer, and an output layer, where the output layer and the input layer are the same size. Reconstruction is defined as the error between the input data and the output data is as small as possible, which forces the hidden layer to learn a good latent representation of the input data. Fig. 1 shows the structure of autoencoder.

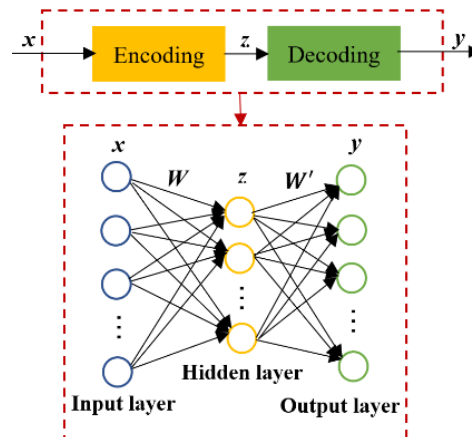


Fig. 1. The structure of AE

AE mainly consists of the following two processes:

- 1) The encoding process from the input layer to the hidden layer:

$$z = g(Wx + b) \quad (1)$$

- 2) The decoding process from the hidden layer to the output layer:

$$y = f(W'z + b') \quad (2)$$

Where  $x, y$  and  $z$  are the input data, the reconstruction of input data and the hidden representation of input data, respectively.  $W$  and  $W'$  are the weight matrix,  $b$  and  $b'$  are

the bias vector.  $g(\cdot)$  and  $f(\cdot)$  are the activation function. If  $W=W'$ , the weights are called the tied weights. It can be seen that the transformation between layers are "linear transformation" + "non-linear activation". The goal of AE is to make reconstruction of the input data, so its loss function combined with different data forms can be mean squared error (MSE) or cross entropy.

The objective loss function of AE can be defined as (3) or (4).

$$L_{AE} = \sum_{i=1}^n \|y_i - x_i\|_2^2 \quad (3)$$

$$L_{AE} = -\sum_{i=1}^n (x_i \log y_i + (1 - x_i) \log(1 - y_i)) \quad (4)$$

Where  $x$  denotes the input data,  $y$  represents the reconstruction of input data,  $n$  denotes the number of input data.

## 2.2 Denoising Autoencoder

Vincent et al. [23] proposed denoising autoencoder(DAE). Its' main idea is: based on the reconstruction of the original input data, it is also desirable to encode and decode added noising data, then output noiseless data, i.e., based on autoencoder, training the corrupted data to reconstruct the raw data. DAE deliberately corrupts the data which input to the network.

For example, the inputs  $x$  and its corrupted version  $\bar{x}$  by adding Gaussian noise to the data or erasing certain dimensions of the data; then, AE maps  $\bar{x}$  to  $z$  by encoder  $g(\cdot)$ , reconstructs  $x$  by decoder  $f(\cdot)$  and produces reconstruction  $\tilde{x}$ , i.e.,  $\tilde{x} = f(g(\bar{x}))$ . Where  $g(\cdot)$  and  $f(\cdot)$  denote encoding and decoding functions, respectively, usually non-linear activation function. Fig. 2 shows the structure of DAE.

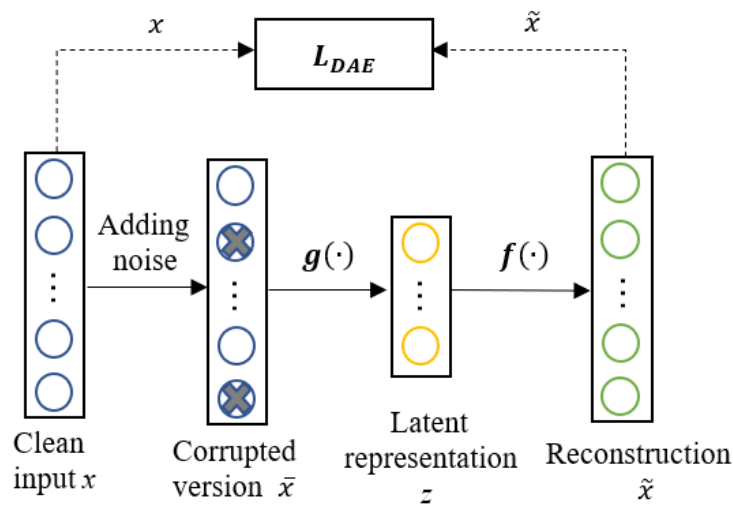


Fig. 2. The structure of DAE

Mean square error(MSE) or cross entropy can be used as loss function. Intuitively, DAE can resist the pollution of raw data to a certain extent, that is, the learned feature transformations are more robust. The goal of DAE is to minimize the loss function to minimize the error between the reconstructed data and the clean input data. The objective loss function of DAE can be defined as (5) or (6).

$$L_{DAE} = \alpha \sum_{i \in I(\bar{x})} (x_i - \tilde{x}_i)^2 + \beta \sum_{i \notin I(\bar{x})} (x_i - \tilde{x}_i)^2 \tag{5}$$

$$L_{DAE} = \alpha \left( - \sum_{i \in I(\bar{x})} [x_i \log \tilde{x}_i + (1 - x_i) \log(1 - \tilde{x}_i)] \right) + \beta \left( - \sum_{i \notin I(\bar{x})} [x_i \log \tilde{x}_i + (1 - x_i) \log(1 - \tilde{x}_i)] \right) \tag{6}$$

Where  $\alpha$  and  $\beta$  are hyperparameters, the parameter  $\alpha$  and  $\beta$  reweight the prediction errors and reconstruction errors, respectively.  $I(\bar{x})$  represents the set of corrupted components of  $x$ .

### 2.3 Stacked Denoising Autoencoder

In general, the ability of DAE to characterize complex problems is limited. The depth model learns the deeper representation of the original data layer by layer, and often can effectively extract effective information, and the number of model network layers is negatively correlated with the degree of abstraction of features. Therefore, stacking multiple DAE models into a depth model can better understand the hidden layer representation of the input data. SDAE [24] can learn the representation of the corrupted input data through learning to predict the clean input data. Such a model is called stacked denoising autoencoder(SDAE). The structure of SDAE is shown in Fig. 3.

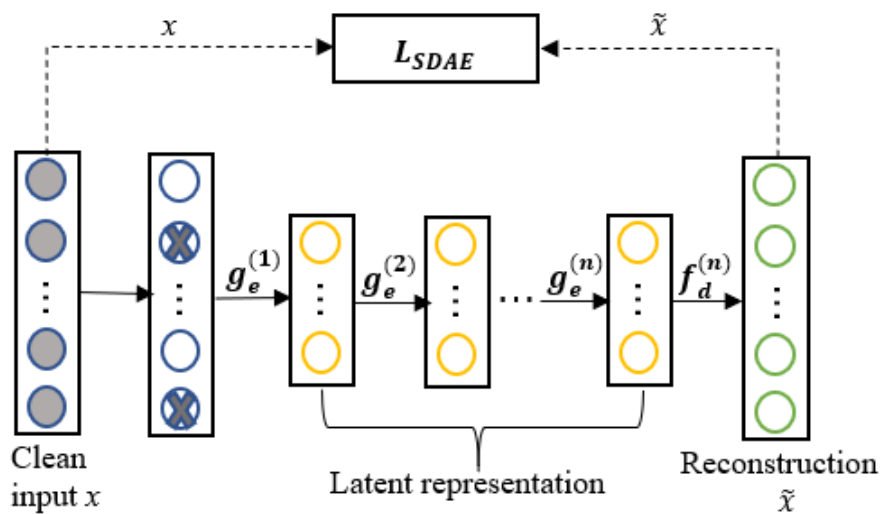


Fig. 3. The structure of SDAE

From **Fig. 3**, we can see that the output of first level DAE hidden layer is learned by encoder  $g_e^{(1)}$ , then it is used as input to train the second level DAE, and so on. Finally, using the real value of  $x$  and the reconstruction value  $\tilde{x}$  to conduct supervised learning, then further fine-tuning the parameters of the entire network. SDAE uses the layer-wise unsupervised pre-training method to train the network.

The specific process is described as follows:

Step1 Given the initial input, train the first level of DAE in an unsupervised manner, reducing the reconstruction error to a threshold which set before.

Step2 Use the output of the first DAE hidden layer as the input to the second level DAE and train the DAE in the same way as above. Repeat this step until all DAEs are initialized.

Step3 Output the hidden layer of the last DAE as a hidden layer of the entire model.

The objective loss function of SDAE can be defined as:

$$L_{SDAE} = \sum_{i=1}^n \|x_i - \tilde{x}_i\|_2^2 + \frac{\lambda}{2} (\sum \|W\|_2^2 + \sum \|b\|_2^2) \quad (7)$$

Where  $\lambda$  denotes the regularization parameter,  $W$  denotes the weight matrix,  $b$  represents the bias vectors.

### 3. Methodology

In this section, we describe our proposed model completely. Firstly, we introduce the framework of our proposed model. Secondly, we integrate auxiliary information with rating information into SDAE. Finally, we present ADAE process by integrating auxiliary information and rating information.

#### 3.1 The Framework of ASDAE

We proposed a novel approach called auxiliary stacked denoising autoencoder(ASDAE) based collaborative filtering, which integrates auxiliary information with rating information as inputs. The framework of ASDAE model as shown in **Fig. 4**. The red balls and blue balls represent auxiliary information and rating information respectively. Be different from traditional SDAE, we integrate auxiliary information with rating information into SDAE, alleviate the cold start problem and learn effective user's preferences accurately.

ASDAE model is based on SDAE, randomly discards certain elements in the input vector  $x$  to obtain the corrupted version  $\bar{x}$ , and then uses  $\bar{x}$  to reconstruct the input vector  $x$ . There are two methods to make corruption, one is to add Gaussian noise, the other is to add mask-out noise. In this paper, we use mask-out noise to make corruption, i.e. randomly set each of the dimensions of  $x$  to 0 with a probability  $q$ :  $p(\bar{x} = x/(1-q)) = 1-q$  or  $p(\bar{x} = 0) = q$ . The main idea of ASDAE is "a good feature can be obtained robustly from the corrupted input, and it can reconstruct the uncorrupted input". ASDAE can reduce the interference of invalid features.

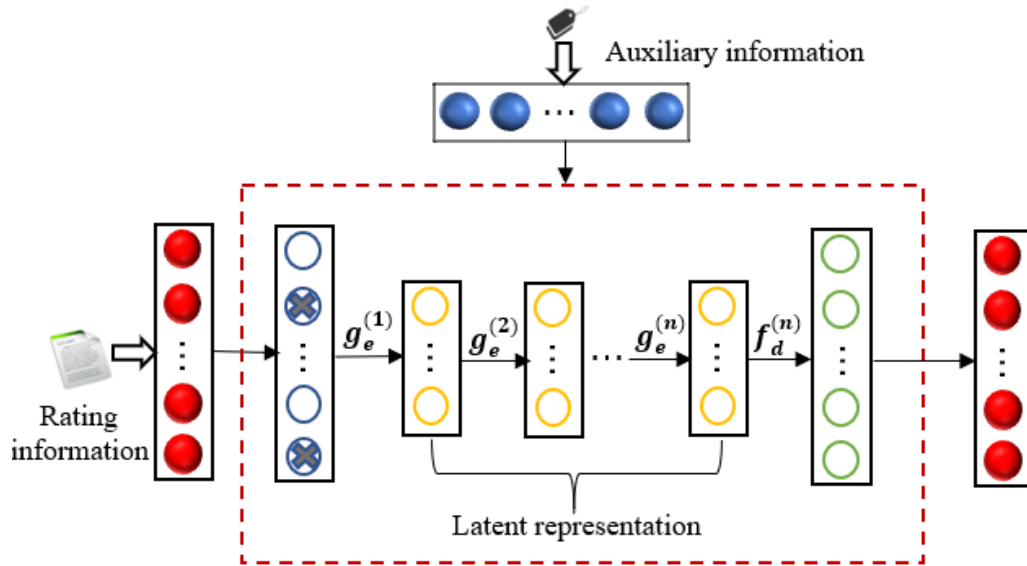


Fig. 4. The framework of our proposed method

We adopt an unsupervised layer-by-layer greedy training algorithm to train ASDAE, which is an effective method for training deep neural networks. The main idea of layer-by-layer greedy training is layer-by-layer training and overall fine-tuning, that is, we train only one layer of ASDAE at a time, and fix the parameters that have been trained, and then train the next layer; when we initialize the weights of ASDAE which obtained by training each layer of ASDAE separately after the training process of the  $n$ th layer of ASDAE is completed. We use the output of  $n$ th layer of ASDAE as the reconstruction rating vector  $x$ , and then utilize the optimization objective function (9) to fine-tune the parameters of the entire network by using the back-propagation algorithm. After the training process of ASDAE is completed, the output of ASDAE is used as the robust features of the original input samples.

### 3.2 Integrating Auxiliary Information with Rating Information

CF method has been widely used in recommender system for the reason of easy to implement and not limited by specific domain knowledge. However, CF method only makes use of user-item rating information, and it is very sparse, it suffers from data sparsity problem. Furthermore, for new user or new item, CF cannot find similar users or items only by the rating information. Cold start problem seriously affects the recommendation performance. Therefore, if we have more information about users or items besides rating information, we can greatly improve the recommendation accuracy. Thus, in order to improve the recommendation accuracy, we integrate auxiliary information with rating information into SDAE. We append auxiliary information to rating information as inputs and inject auxiliary information to hidden layer of the SDAE. Appending auxiliary



information to rating information as inputs to the SDAE can alleviate data sparsity and injecting auxiliary information to hidden layer of the SDAE can enhance the latent representation. Fig. 5 shows appending auxiliary information to rating information as inputs and injecting auxiliary information to every layer of the SDAE.

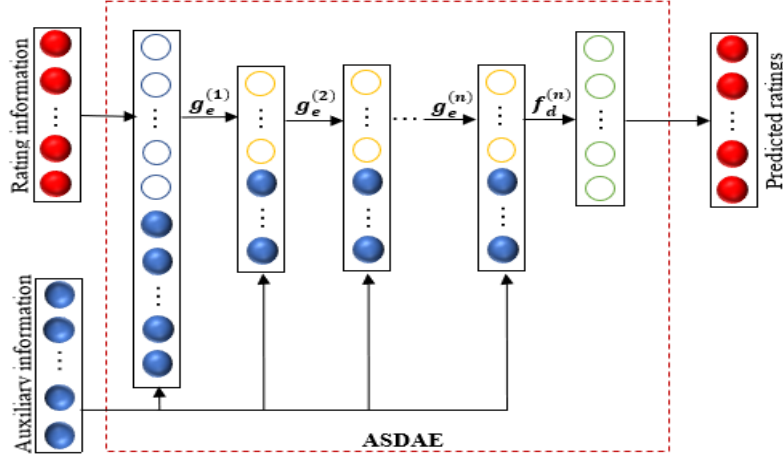


Fig. 5. The schematic diagram of fusing process

The predicted ratings are the representation of the output layer. Defined as in

$$\hat{R}^{(i)} = f(W_L \cdot g\{\dots W_1 \cdot g(\{r^{(i)}; a_i\} + b_1) \dots\}, a_i\} + b_L) \quad (8)$$

Where  $\hat{R}^{(i)}$  represents the predicted rating of user  $i$ ,  $\{r^{(i)}; a_i\}$  denotes concatenation of  $r^{(i)}$  and  $a_i$ ,  $W_l$  denote the weight matrix,  $b_l$  are the bias vectors,  $a_i$  denotes the auxiliary information of user  $i$ ,  $r^{(i)}$  denotes user  $i$ 's rating information,  $L$  denotes the number layers of ASDAE.  $g(\cdot)$  and  $f(\cdot)$  denote non-linear activation function, in our approach, we adopt sigmoid function.

In order to make best use of auxiliary information, Firstly, we concatenate rating information with auxiliary information as inputs to ASDAE. Then we inject auxiliary information to every hidden layer of ASDAE. Finally train ASDAE by backpropagation.

We suppose ASDAE consists 6 layers,  $L_0$  is the input layer, the initial input data  $x$  is the combination of the rating matrix and auxiliary matrix,  $L_c$  is the output layer,  $L_1$ ,  $L_2$ , and  $L_3$  are used for dimensionality reduction compression,  $L_4$  are used for performing decoding. During layer-by-layer training, after the  $i$ th layer ASDAE training is completed, the weight of the  $i$ th layer is fixed, and the output  $g_e^i(\cdot)$  and  $a_i$  are concatenated as the input of the  $i+1$  layer. After the training of all the layers is completed, the weight parameter set is obtained and as the initial weights of ASDAE. When the training of ASDAE is completed, the output of  $L_3$  is the latent feature of rating information.

### 3.3 Handling Sparse Rating Information

The problem of sparsity of rating information has always affected the performance of recommendation, we use ASDAE to handle the missing ratings of rating information. The process of handling missing rating of ratings information as shown in Fig. 6.

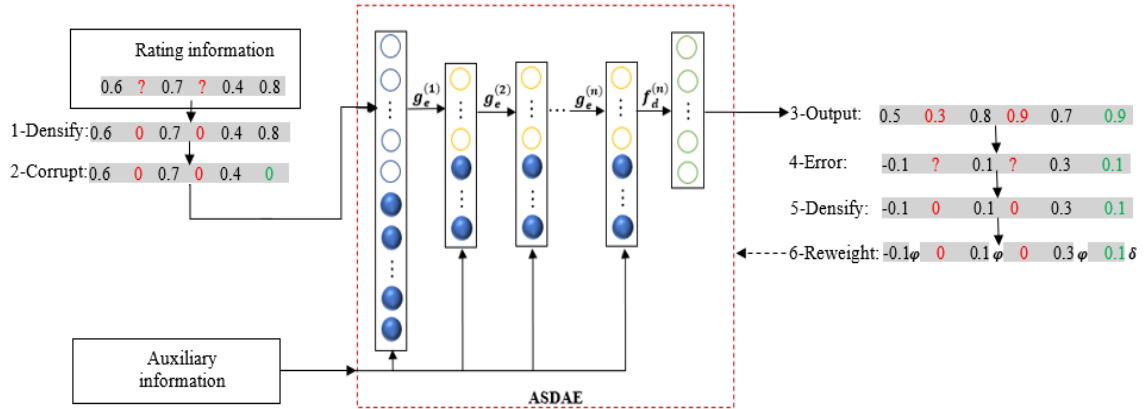


Fig. 6. The process of handling missing ratings of rating information

Firstly, we add the masking noise to the rating information. In our approach, we set missing ratings of rating information to zero, then get a dense rating information.

Secondly, we corrupt a small fraction of known ratings to zero, the corruption ratings simulate missing ratings in the training process of ASDAE.

Thirdly, we disregard unknown rating's loss by using a specific loss function to prevent the backpropagated errors of unknown rating.

Before backpropagation, we set missing ratings to zero error. The parameter  $\delta$  and  $\varphi$  are used for reweighting prediction errors and reconstruction errors, respectively. The parameter is a tuning parameter, which used for controlling the influence of adding noise. In order to emphasize the prediction of missing ratings over the reconstruction of known ratings, we set  $\delta$  is larger than  $\varphi$ .

We define the training loss function as:

$$\begin{aligned} \mathcal{L} = & \delta \left( \sum_{i \in I(o) \cap I(c)} [h(\{\tilde{r}^{(i)}; a_i\}) - r^{(i)}]^2 \right) \\ & + \varphi \left( \sum_{i \in I(o) \setminus I(c)} [h(\{\tilde{r}^{(i)}; a_i\}) - r^{(i)}]^2 \right) \\ & + \frac{\lambda}{2} \cdot \text{Regularization} \end{aligned} \quad (9)$$

Where  $\mathcal{L}$  denotes the loss function,  $\delta$  and  $\varphi$  are used for reweighting prediction errors and reconstruction errors, respectively.  $\lambda$  represents the regularization parameter. **Regularization** adopts L2 norm to avoid overfitting.  $r^{(i)}$  denotes user  $i$ 's rating information.  $\tilde{r}^{(i)}$  denotes the corrupted ratings of user  $i$ 's rating information,  $a_i$  denotes the auxiliary

information of user  $i$ ,  $\{\tilde{r}^{(i)}; a_i\}$  represents concatenation of  $\tilde{r}^{(i)}$  and  $a_i$ ,  $h(\cdot)$  denotes non-linear activation function, i.e., the output of ASDAE.  $I(o)$  denotes user-item rating information,  $I(c)$  denotes corrupted user-item rating information.

In Equation (9), **Regularization** =  $\sum \|W\|_2^2 + \sum \|b\|_2^2$ ,  $W$  denotes the weight matrix of ASDAE,  $b$  denotes the bias vectors. We initialize  $W_{i,j} \sim [-1/\sqrt{n}, 1/\sqrt{n}]$ , and use batch stochastic gradient descent to optimize. The batch size is  $p$ . The process of updating  $W$ , and  $b$  are as follows:

$$W' = W - \frac{\eta}{p} \sum_{i=1}^p \frac{\partial L(\{\tilde{r}^{(i)}; a_i\}, W, b)}{\partial W} \quad (10)$$

$$b' = b - \frac{\eta}{p} \sum_{i=1}^p \frac{\partial L(\{\tilde{r}^{(i)}; a_i\}, W, b)}{\partial b} \quad (11)$$

Where  $\eta$  represents the learning rate.

### 3.4 Complexity Analysis

In our approach, we use the open library TensorFlow to implement ASDAE and utilize batch stochastic gradient descent to optimize the ASDAE model. In our model, the inputs contain two kinds of information, i.e., user-item rating information and auxiliary information, so the sum of rating information and auxiliary information is the dimension of inputs. The time complexity of our model is  $O(n(m+q))$ . Where  $n$ ,  $m$  and  $q$  denote the number of users, the number of items and the number of auxiliary information, respectively. To sum up, our model has potential to scale up to large-scale data sets.

### 3.5 Comparisons of AE, DAE, SDAE and ASDAE

In our approach, we use SDAE to extract the latent feature of rating information and auxiliary information. Different from SDAE, besides inputting auxiliary information to the input layer, it's also been input to each hidden layer. We summarize the differences of AE, DAE, SDAE and ASDAE as shown in [Table 1](#).

**Table 1.** The comparisons of AE, DAE, SDAE and ASDAE

	The input layer	The hidden layer	The output layer	The main feature
AE	The high-dimensional input data	One hidden layer	The reconstruction of input data	Extracts the latent feature by reconstructing the input data
DAE	The corrupted version of high-dimensional input data	One hidden layer	The reconstruction of original input data	Forces the hidden layer to acquire more robust features and to prevent DAE from simply learning the identity function

SDAE	The corrupted version of high-dimensional input data	Two or more hidden layers	The reconstruction of original input data	Stacks several DAEs together to get higher level representations
ASDAE	The corrupted version of high-dimensional input data and the auxiliary information	Two or more hidden layers, and input the auxiliary information to each hidden layer	The reconstruction of user-item rating matrix	Stacks several DAEs together to reconstruct the user-item rating matrix

## 4. Experimental Results and Analysis

In this section, we detail and analyze experimental results. First, we introduce the datasets, then describe the baselines and experimental settings, followed by the evaluation metrics. Finally, the result and analysis of experiment are presented.

### 4.1 Data Sets

We use Epinions, MovieLens-100K, and MovieLens-1M as datasets to validate the performance of our proposed model. In Epinions, user can browse other user's ratings and reviews on products. Products are classified according to their category, such as movies, digital products, books and so on. In addition, some of users have trust relationships with others in Epinions. The trust relationship is directional and the trust value is binary in Epinions. Auxiliary information is the trust relationship of users. The trust relationship can be treated as matrix of user's tags. Epinions contains 922 267 ratings, 22 166 users and 296 277 items, and 355 813 trust relationships. The sparsity of the user-item rating matrix is 99.986%.

MovieLens-100K and MovieLens-1M contain information such as user's ratings and auxiliary information (users' age, sex, gender, occupation, and movies' category). Auxiliary information for MovieLens-100K and MovieLens-1M is a tag matrix  $T$ . MovieLens-100K contains 943 users and 1 682 movies, 100 000 ratings. MovieLens-1M contains 6 040 users and 3 706 movies, 276 256 ratings. We extract users with ratings larger than 3 for training and testing.

Since the huge dimensionality of the auxiliary information (exclude texts or pictures), we use matrix factorization to reduce the dimension of the matrix of tags or trust friendships. We adopt the method proposed by Yeung et al. [25] to directly learn the embedding of auxiliary information and utilize the left part of a matrix factorization of the tag matrix  $T$ . The movie category is represented by binary. Finally, we concatenate rating information with auxiliary information together.

The experimental environment is: Intel Core i9-9900K processor, 5GHz, RAM-16GB, Windows10 operating system.

## 4.2 Baselines and Experimental Settings

To verify our proposed model, we select the following methods as baselines:

- 1)PMF. Mnih and Salakhutdinov [26] proposed PMF, PMF represents probabilistic matrix factorization and only uses user-item rating matrix to generate recommendation.
- 2)TrustSVD [4]. This method adds explicit trust information to SVD to improve recommendation performance and is widely used method for rating prediction.
- 3)DAE. DAE [23] improves the recommendation accuracy by adding noise to AE, the method only uses user-item rating information as inputs.
- 4) I-AutoRec. This method is a collaborative filter algorithm based on autoencoder proposed in [27], it trains one autoencoder per item and sharing the weights between the different autoencoders.
- 5)CDAE [20]. This method injects users' preference into hidden layer of SDAE to improve recommendation performance.
- 6)CDL [25]. This method combines SDAE with probabilistic model for collaborative filtering and is a widely used algorithm. CDL achieves better recommendation performance.
- 7)ASDAE. Our proposed method integrates rating information with auxiliary information as inputs to the SDAE.

We randomly select 90% of the samples as training set and the remaining10% as test set. This random selection was performed 5 times independently and the average of 5 runs as the final result. We use batch stochastic gradient descent to optimize ASDAE and set the batch of size to 40.

For fair comparison, we set the parameters for different methods to achieve optimal performance. **Table 2** shows the parameter settings of perspective methods.

**Table 2.** Parameter settings of respective methods

Methods	Parameter settings
PMF	$\lambda_U=\lambda_V=0.03$ , dim=50, $\eta=0.05$
TrustSVD	$\lambda=0.9$ , $\lambda_t=0.5$
DAE	$\lambda_U=\lambda_V=0.1$ , $\eta=0.1$ , dim=300
I-AutoRec	$\lambda=0.9$ , $\eta=0.001$
CDAE	$\lambda=0.9$ , $\eta=0.001$ , dim=50
CDL	$\lambda=0.9$ , $\lambda_U = 1$ , $\lambda_V = 1000$ , $\eta=0.001$ , dim=50, noise=0.3, $\lambda_w = 0.0001$
ASDAE	$\delta = 0.6$ , $\varphi = 0.4$ , $\lambda=0.6$ , $\eta= 0.001$ , dim=50, noise=0.3, $\lambda_w = 0.5$

In our method, we use the sigmoid function as activation function, the layers of ASDAE is 6-layers.

### 4.3 Evaluation Metrics

There are many different metrics used to measure the performance of recommendation, such as root mean squared error (RMSE), mean absolute error (MAE), and normalized mean absolute error (NMAE). In this paper, we use RMSE and MAE to evaluate the rating prediction performance. The smaller of these evaluation metrics, the better of the performance of recommendation.

The definition of RMSE is:

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in R_{test}} |r_{ui} - \hat{r}_{ui}|^2}{|R_{test}|}} \quad (12)$$

Where  $r_{ui}$  and  $\hat{r}_{ui}$  represent the actual and the predicted rating for user  $u$  on item  $i$ , respectively;  $|R_{test}|$  represent the number of ratings in test set. A lower RMSE represents a higher predictive accuracy.

The definition of MAE is:

$$MAE = \frac{\sum_{(u,i) \in R_{test}} |r_{ui} - \hat{r}_{ui}|}{|R_{test}|} \quad (13)$$

Regarding the top- $k$  recommendation performance, we use the most commonly used evaluation metrics, such as Precision, Recall and F1\_score. For the target user  $u$ , precision represents the probability that the top- $k$  recommended items will hit in the test set. Recall represents the proportion of the set of  $k$  items recommended to user  $u$  that were actually liked. *F1\_score* represents the harmonic mean of precision and recall.

The definition of precision is:

$$P = \frac{|R(u) \cap T(u)|}{|R(u)|} \quad (14)$$

The definition of recall is:

$$R = \frac{|R(u) \cap T(u)|}{|T(u)|} \quad (15)$$

The definition of F1\_score is:

$$F1\_score = \frac{2PR}{P+R} \quad (16)$$

Where  $P$  and  $R$  denotes precision and recall, respectively.  $R(u)$  represents a recommended list of items for user  $u$ ,  $T(u)$  represents a set of items that have interactions with user  $u$  in the test set.

## 4.4 Experimental Results

### 4.4.1 Rating prediction performance comparison

In order to demonstrate the performance of respective methods, we conduct a series of experiments to compare with our proposed model. **Table 3** shows experimental results of respective methods.

**Table 3.** Rating prediction performance comparison on three datasets

Methods	MovieLens-100K		MovieLens-1M		Epinions	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
PMF	0.9217	0.7112	0.8788	0.6910	1.4665	0.9837
TrustSVD	0.8524	0.6572	0.7462	0.6032	1.1468	0.8598
DAE	0.8122	0.6249	0.7342	0.5887	1.1120	0.8475
I-AutoRec	0.8219	0.6387	0.7428	0.5968	1.272	0.8501
CDAE	0.7882	0.6024	0.6984	0.5613	1.0723	0.8288
CDL	0.7677	0.5912	0.6812	0.5532	1.0542	0.8075
ASDAE	<b>0.7453</b>	<b>0.5743</b>	<b>0.6701</b>	<b>0.5323</b>	<b>1.0301</b>	<b>0.7872</b>

From **Table 3**, we can find:

1) On three datasets, the accuracy of our proposed model is superior to the other state-of-the-art methods, and our method has excellent performance. For example, compared with the optimal results of CDL, CDAE, DAE, I-AutoRec, TrustSVD and PMF on MovieLens-1M dataset, the RMSE of our method improved 1.63%, 4.05%, 8.73%, 9.79%, 10.20% and 23.75%, respectively. ASDAE model can learn more accurate user's preference by integrating auxiliary information with rating information. That is, the accuracy of ASDAE is the best among all the baseline methods. The effective use of auxiliary information is more important for the fewer number of rating information especially for new users or items. CDAE also uses SDAE model, but it only injects users' preference into hidden layer of SDAE, so the performance of CDAE model is less than our proposed method. CDL combines SDAE with probabilistic model and is a strong baseline method. However, our proposed method can beat CDL on all three datasets.

2) The performance of ASDAE, CDL, CDAE are better than DAE on three datasets. For example, on MovieLens-100K dataset, compared with the optimal results of DAE, the RMSE of ASDAE, CDL and CDAE improved 8.24%, 5.48%, 2.95%, respectively. The reason is that ASDAE, CDL, CDAE are use SDAE model which stacks several DAEs together to learn users' latent representation. They have more accuracy than DAE.

3) We can observe that ASDAE, CDL, CDAE and DAE achieve better performance than I-AutoRec. For example, on Epinions dataset, compared with the optimal results of I-AutoRec, the RMSE of ASDAE, CDL, CDAE and DAE improved 8.73%, 6.56%, 4.94%,

1.37%, respectively. The reason is that I-AutoRec use AE for collaborative filtering. However, ASDAE, CDL, CDAE and DAE use DAE which corrupts the inputs before mapping them into the latent representation, they can learn more robust features than AE.

4) The performance of PMF is the worst among all approaches. Since PMF only uses rating matrix to factorize, ignores auxiliary information, it's also verifying that auxiliary information can greatly improve the accuracy of recommendation.

#### 4.4.2 Top-*k* recommendation performance comparison

We also conduct a series of experiments to compare our proposed method with the baseline methods on top-*k* recommendation performance. **Table 4** shows experimental results of respective methods.

**Table 4.** Top-*k* recommendation performance comparison three datasets

Methods	MovieLens-100K			MovieLens-1M			Epinions		
	P@20	R@20	F1_score	P@20	R@20	F1_score	P@20	R@20	F1_score
PMF	0.14	0.12	0.1292	0.15	0.14	0.1448	0.13	0.11	0.1192
TrustSVD	0.16	0.15	0.1548	0.17	0.16	0.1648	0.15	0.13	0.1393
DAE	0.20	0.18	0.1895	0.23	0.21	0.2195	0.18	0.17	0.1749
I-AutoRec	0.18	0.16	0.1694	0.20	0.18	0.1895	0.17	0.14	0.1535
CDAE	0.23	0.21	0.2195	0.24	0.22	0.2296	0.20	0.19	0.1949
CDL	0.25	0.24	0.2449	0.26	0.25	0.2549	0.23	0.21	0.2195
ASDAE	<b>0.27</b>	<b>0.25</b>	<b>0.2596</b>	<b>0.28</b>	<b>0.26</b>	<b>0.2696</b>	<b>0.25</b>	<b>0.22</b>	<b>0.2340</b>

From **Table 4**, we can find:

1) On three datasets, our proposed model is superior to the other state-of-the-art methods on three metrics and has the best top-*k* recommendation performance. For example, compared with the optimal results of CDL, CDAE, DAE, I-AutoRec, TrustSVD and PMF on MovieLens-1M dataset, the F1\_score of our method improved 5.77%, 17.44%, 22.80%, 42.29%, 63.54% and 86.15%, respectively. ASDAE model can learn more accurate user's preference by integrating auxiliary information with rating information. That is, the top-*k* recommendation performance of ASDAE is the best among all the baseline methods. The effective use of auxiliary information is more important for the fewer number of rating information especially for new users or items.

2) The top-*k* recommendation performance of ASDAE, CDL, CDAE are better than DAE on three datasets. For example, on MovieLens-100K dataset, compared with the optimal results of DAE, the P@20 of ASDAE, CDL and CDAE improved 35%, 25%, 15%, respectively. The reason is that ASDAE, CDL, CDAE are use SDAE model which stacks several DAEs together to learn users' latent representation. They have more accuracy top-*k* recommendation performance than DAE.



#### 4.4.3 Impact of parameter $\delta$ and $\varphi$

In our proposed method,  $\delta$  and  $\varphi$  are important parameters that affect the performance of recommendation. The parameter  $\delta$  affects the effect of adding noise to ASDAE ( $\delta + \varphi = 1$ ). In the training process, the known ratings are corrupted through a certain masking ratio to simulate missing ratings, then train the SDAE to predict them. The smaller  $\delta$  means that we are more dependent on reconstruction errors in our proposed model. In extreme case, such as  $\delta = 0$ , the proposed model in this paper will completely depend on reconstruction errors to learn users' latent feature vectors while ignoring prediction errors; conversely, the larger  $\delta$  means that we give more weight to prediction errors. For example,  $\delta = 1$  and  $\varphi = 0$  indicates that the method only uses prediction errors to learn users' latent feature vector; conversely,  $\delta = 0$  and  $\varphi = 1$  indicates that the method only uses reconstruction errors to learn users' latent feature vector. In this experiment, we set  $\delta$  from 0.05 to 0.95 with 0.05 increments, and the parameter  $\varphi = 1 - \delta$ . Fig. 7 shows the experimental results on three datasets.

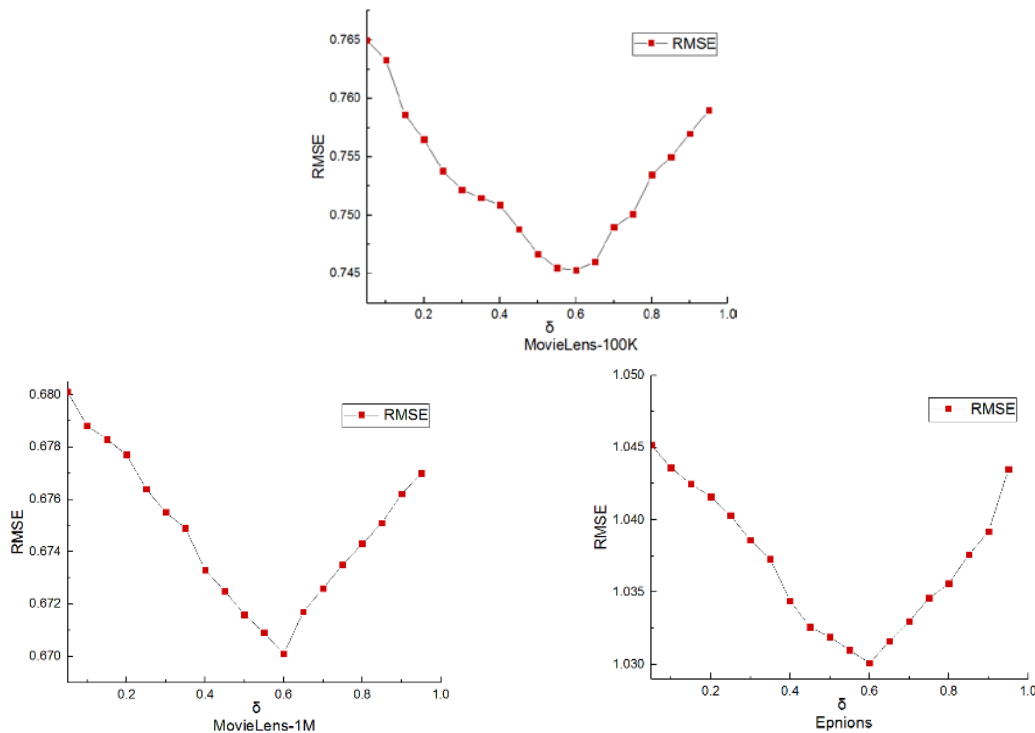


Fig. 7. Impact of parameter  $\delta$  and  $\varphi$

From Fig. 7, firstly, it can be seen that the parameter  $\delta$  affects the performance of recommendation. Secondly, the value of RMSE showed a similar trend: With the increasing of  $\delta$ , RMSE decreased first and the predictive accuracy increased. After reaching the

optimal, RMSE increased with the increasing of  $\delta$ , i.e., the predictive accuracy decreased. Finally, on the three datasets, our proposed method achieved best performance when  $\delta = 0.6$  and  $\varphi = 0.4$ .

#### 4.4.4 Impact of auxiliary information

In this experiment, we compare the recommendation performance of ASDAE with SDAE, as shown in Table 4. ASDAE integrates rating information with auxiliary information as inputs to the SDAE, while SDAE only uses rating information as inputs. All experimental parameters of ASDAE and SDAE are the same. From Table 5, we can find that ASDAE can achieve more accurate than SDAE by integrating auxiliary information with rating information. The improvement of ASDAE on three datasets are significant. The RMSE of ASDAE is improved by 7.86%, 7.20% and 6.46%, and the MAE of ASDAE is improved by 6.42%, 6.63% and 5.19% on three datasets, respectively. It's also verifying that the effective use of auxiliary information can greatly improve the performance of recommendation. All above, experimental results show that the predictive accuracy of our method is superior to the SDAE.

**Table 5.** Performance comparison of SDAE and ASDAE

Methods	MovieLens-100K		MovieLens-1M		Epinions	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
SDAE	0.8091	0.6137	0.7221	0.5571	1.1012	0.8303
ASDAE	0.7453	0.5743	0.6701	0.5223	1.0301	0.7872
Improved (%)	7.8853	6.4200	7.2012	6.2466	6.4565	5.1908

#### 4.4.5 Impact of masking noise

In this experiment, we set different ratio of masking noise for comparison on MovieLens-1M, as shown in Fig. 8. We can find that as the masking ratio increased, the recommendation performance increased. When masking ratio increased more than 0.3, the recommendation performance decreased. While  $\delta = 0.6$  and masking ratio=0.3, the performance of recommendation achieves the best.

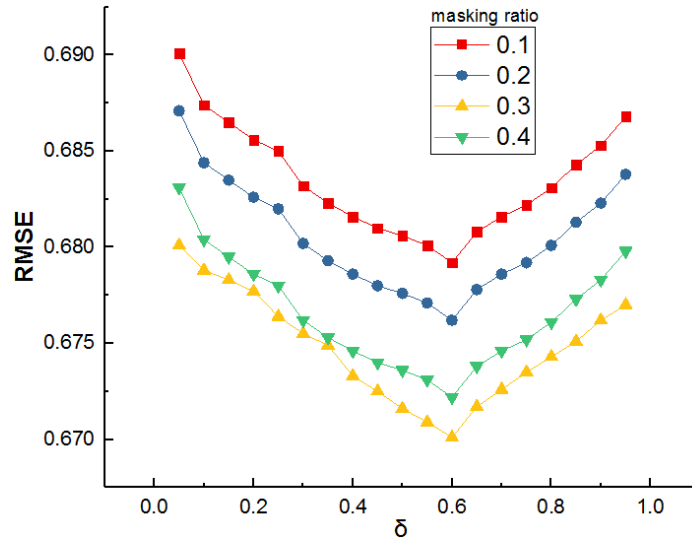


Fig. 8. Impact of different ratio of masking noise

#### 4.4.6 Training time comparison

In this experiment, we conduct a series of experiment on model training time for comparison. Experimental results of different methods on training time as shown in Table 6.

Table 6. The training time of different methods(MINUTE:SECOND)

Methods	MovieLens-100K	MovieLens-1M	Epinions
PMF	13:15	17:27	20:52
TrustSVD	17:27	20:38	26:36
DAE	23:45	28:19	34:58
I-AutoRec	24:20	29:27	35:41
CDAE	27:19	38:07	49:23
CDL	26:29	37:54	48:40
ASDAE	25:17	35:25	42:37

From Table 5, it can be seen that PMF has the least training time on all datasets. The reason is that PMF does not consider the effect of auxiliary information on the user latent feature vector when learning the latent feature of users or items. In terms of model training time, DAE and I-AutoRec are outperform other SDAE methods, such as CDAE, CDL, and ASDAE, this shows that SDAE methods introduced additional feature-assisted enhancement information (users' preference), so it took more time to perform data pre-processing and model training, but the performance of the SDAE methods was improved compared with DAE and I-AutoRec. At the same time, the time spent is still acceptable. Our method is more efficient than CDAE and CDL on the three datasets.

All above, our proposed method outperforms other baseline methods in all experiments. In terms of the efficiency of model learning, our method is also better than the current mainstream SDAE-based approaches.

## 5. Conclusion

In this paper, we proposed a novel deep learning method named auxiliary stacked denoising autoencoder(ASDAE) based collaborative filtering recommendation model, the model learns the preferences of users from auxiliary information and rating information effectively. Firstly, we integrated auxiliary information with user's rating information. Then, we designed a stacked denoising autoencoder based collaborative recommendation model to learn the preferences of users from auxiliary information and rating information. Finally, we compared our proposed method with baselines on the three datasets. Experimental results on real datasets showed that our proposed model outperforms state-of-the-art recommendation methods.

In future work, this paper only considered auxiliary information with rating information, some possible information may improve the recommendation performance, such as users' review information. User's review information also reflects the user's preference in some extent. The employment of user's review information as auxiliary information is the future research direction of this paper. In addition, in real scenarios, user often only provide implicit feedback information, such as purchase records, click records, etc. How to learn user's preference from the implicit feedback and use it to improve the recommendation performance is also the future research direction. Finally, different variants of autoencoders such as conditional variational autoencoders or marginalized denoising autoencoders are another the future research direction.

## Acknowledgements

This research was supported by the National Key Research and Development Plan Key Projects of China under Grant No.2017YFC0405800, the National Natural Science Foundation of China Grant (Nos. 60971088, 60571048).

We thank the anonymous referees for their helpful comments and suggestions on the initial version of this paper.

## References

- [1] Al-Sabahi Kamal, Zuping Zhang and Nadher M, "A hierarchical structured self-attentive model for extractive document summarization (HSSAS)," *IEEE ACCESS*, vol.6, pp.24205-24212, 2018. [Article \(CrossRef Link\)](#).

- [2] Peng Y, Zhu W and Zhao Y, "Cross-media analysis and reasoning: advances and directions," *Frontiers of Information Technology & Electronic Engineering*, vol.18, no.1, pp.44-57, Feb, 2017. [Article \(CrossRef Link\)](#).
- [3] Ruihui M, "A Survey of Recommender Systems based on Deep learning," *IEEE ACCESS*, vol.6, pp.69009 – 69022, 2018. [Article \(CrossRef Link\)](#).
- [4] G. Guo, J. Zhang and N. Yorke-Smith, "A Novel Recommendation Model Regularized with User Trust and Item Ratings," *IEEE Transactions on Knowledge and Data Engineering*, vol.28, no.7, pp.1607-1620, July, 2016.
- [5] Fang. H, Guo. G and Zhang. J, "Multi-faceted trust and distrust prediction for recommender systems," *Decision Support Systems*, vol.71, pp.37–47,2015. [Article \(CrossRef Link\)](#).
- [6] Li. X and She. J, "Collaborative variational autoencoder for recommender systems," in *Proc. of the SIGKDD 23th International Conference on Discovery Data Mining*, pp.305–314, 2017.
- [7] He X, He Z, Song J, Liu Z, Jiang Yu and Chua T, "Nais: Neural attentive item similarity model for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol.30, no.12, pp.2354-2366, 2018.
- [8] Sharma R, Gopalani D and Meena Y, "Collaborative Filtering based Recommender System: Approaches and research challenges," in *Proc. of International Conference on Computational Intelligence & Communication Technology*, pp.1-6, 2017.
- [9] Ming He, Qian Meng, and Shaozong Zhang, "Collaborative Additional Variational Autoencoder for Top-N Recommender Systems," *IEEE ACCESS*, vol.7, pp. 5707-5713, 2019. [Article \(CrossRef Link\)](#).
- [10] Yang Bo, Lei Yu, Liu Dayou and Liu Jiming, "Social collaborative filtering by trust," in *Proc. of the 23rd Joint Conference on Artificial Intelligence(IJCAI'13), Menlo Park, AAAI, CA*, pp.2747-2753, 2013. [Article \(CrossRef Link\)](#).
- [11] Linden G, Smith B and York J, "Amazon. Com recommendations: Item-to-item collaborative filtering," *Internet Computing*, vol.7, no.1, pp.76-80, 2003. [Article \(CrossRef Link\)](#).
- [12] Pereira A L V and Hruschka E R, "Simultaneous co-clustering and learning to address the cold start problem in recommender systems," *Knowledge Based Systems*, vol.82, pp.11-19, 2015. [Article \(CrossRef Link\)](#).
- [13] Guo Guibing, Zhang Jie and Yorke-Smith N, "TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proc. of 29th AAAI conference on artificial intelligence, AAAI Press*, pp.123–129, 2015. [Article \(CrossRef Link\)](#).
- [14] Yu. H, Shen Z, Miao. C, An B and C Leung, "Filtering trust opinions through reinforcement learning," *Decision Support Systems*, vol.66, pp.102–113,2014. [Article \(CrossRef Link\)](#).
- [15] Campos. P. G, Díez. F and Cantador. I, "Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols," *User Modeling and User-Adapted Interaction*, vol.24, no.1, pp. 67–119, Feb, 2014. [Article \(CrossRef Link\)](#).
- [16] Çano E and Morisio M, "Hybrid recommender systems: A systematic literature review," *Intelligent Data Analysis*, vol.21, no.6, pp.1487-1524, 2019. [Article \(CrossRef Link\)](#).

- [17] Rafailidis D and Crestani F, “Recommendation with social relationships via deep learning,” in *Proc. of the ACM SIGIR International Conference on Theory of Information Retrieval*, pp.151–158, 2017. [Article \(CrossRef Link\)](#).
- [18] Wang. H, Shi X and Yeung. D, “Collaborative Recurrent Autoencoder: Recommend while Learning to Fill in the Blanks,” in *Proc. of NIPS*, pp.77-88, 2016. [Article \(CrossRef Link\)](#).
- [19] Strub. F, Gaudel. R and Mary. J, “Hybrid recommender system based on autoencoders,” in *Proc. of RecSys DLRS workshop*, 2017. [Article \(CrossRef Link\)](#).
- [20] Wu. Y, DuBois. C, Zheng. A. X and Ester. M, “Collaborative Denoising Autoencoders for Top-N Recommender Systems,” in *Proc. of the Ninth ACM International Conference on Web Search and Data Mining, New York, NY, USA*, pp. 153–162, Feb, 2016. [Article \(CrossRef Link\)](#).
- [21] Pan Y, He F and Yu H, “Trust-aware top-N recommender systems with correlative denoising autoencoder,” *2017.arXiv:1703.01760*, 2017. [Article \(CrossRef Link\)](#).
- [22] Rumelhart D E, Hinton G E and Williams R J, “Learning representations by back-propagating errors,” *Nature*, vol.323, pp.533-536, 1986. [Article \(CrossRef Link\)](#).
- [23] Vincent P, Larochelle H, Bengio Y and Manzagol P, “Extracting and composing robust features with denoising autoencoders,” in *Proc. of the 25th International Conference on Machine Learning, Helsinki, Finland*, pp.1096-1103, 2008. [Article \(CrossRef Link\)](#).
- [24] Vincent P, Larochelle H, Lajoie I, Bengio Y and Manzagol P, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of Machine Learning Research*, vol.11, pp.3371-3408, 2010. [Article \(CrossRef Link\)](#).
- [25] Wang H, Wang N and Yeung D.Y, “Collaborative deep learning for recommender systems” in *Proc. of KDD*, 2014. [Article \(CrossRef Link\)](#).
- [26] Mnih A, Salakhutdinov R, “Probabilistic matrix factorization,” *Neural information processing systems*, pp,1257–1264, 2008.
- [27] Sedhain S, Menon AK, Sanner S and Xie L, “AutoRec: Autoencoders meet collaborative filtering,” in *Proc. of the 24th International Conference on World Wide Web, Florence, Italy*, pp.111–112, 2015. [Article \(CrossRef Link\)](#).



**Ruihui Mu** received his M.S. degree in computer technology from Huazhong University of Science and Technology, He is currently a Ph.D. student at Hohai University. He is an associate professor in Henan Xinxiang University. His research interests include deep learning, neural networks and personalized recommender systems.



**Xiaoqin Zeng** received his B.S. degree in computer software from Nanjing University, M.S. degree in computer applications from Southeast University in China, and his Ph.D. degree in computer science from the Hong Kong Polytechnic University. He is a professor at Hohai University, a doctoral supervisor, and director of the Institute of Intelligent Science and Technology. His research interests focus on machine learning, computing and intelligence, and neural networks.