Improved Collaborative Filtering Algorithm Based on Stacked Denoising AutoEncoders

Lei Jiang(✉), Jingping Song, and Tianhan Gao

Notheastern University Shenyang, Liaoning, China
1871089@stu.neu.edu.cn, songjp@swc.neu.edu.cn, gaoth@mail.neu.edu.cn

Abstract. With the rapid development of the mobile Internet, the increasing user data has brought about serious information overload. Recommendation system is a more effective solution to information overload. Collaborative filtering is one of the most widely used methods in recommendation systems. The traditional collaborative filtering algorithm performs the recommendation in terms of the rating matrix to calculate the similarity. While in most applications, the ratings of the users for the item is sparse, which leads to the issues of low recommendation accuracy and cold start. In addition, traditional collaborative filtering is based on the user's historical behavior neglecting auxiliary information of users and items. For new users, it is impossible to accurately predict the preferences. In this paper, the Stacked Denoising AutoEncoder is integrated into collaborative filtering. The ratings and auxiliary information are taken as input, and two Stacked Denoising AutoEncoder are explored to learn the implicit representation of users and items respectively. Thus the similarity between users and items can be calculated to make score prediction. In addition, the weight factor is introduced to control the proportion of the two score predictions to improve the sparsity of collaborative filtering. Experiments are done on the MovieLens dataset, where the accuracy of the proposed algorithm is proved to be significantly improved compared with several mainstream algorithms.

Keywords: Mobile Internet · Collaborative filtering · Stacked Denoising AutoEncoder · Recommendation · Deep learning

1 Introduction

In recent years, the mobile Internet and e-commerce industries have developed rapidly, and the amount of information and data traffic has exploded. People are facing serious information overload problems. In the context of this year-on-year development of Internet technology and communication technology, a good recommendation system is particularly important [1–4]. The recommendation system with the help of the mobile Internet platform uses the interactive information between users and items to help users find information of interest

I. You et al. (Eds.): MobiSec 2019, CCIS 1121, pp. 165–176, 2020.
https://doi.org/10.1007/978-981-15-9609-4_12

and solve the problem of information overload [5]. At the same time, the development of mobile Internet has also greatly promoted the rapid development of recommendation systems.

Collaborative filtering is one of the most widely used methods in recommendation systems [6],which predicts user preferences simply and effectively by discovering and exploiting the similarities between users and items through the rating matrix. The most widely employed models are user-based and item-based collaborative filtering. However, these shallow models cannot learn the deep features of users and items, limiting their scalability for recommendation. In recent years, deep learning techniques represented by neural networks have made considerable progress in the fields of image and speech [6]. Consequently, more and more research has been put forward to apply neural networks into collaborative filtering, where the autoencoder model, such as AutoRec [7–10] is the most ideal one. Compared with the traditional collaborative filtering algorithm, the recommendation accuracy of AutoRec is greatly improved. Unfortunately, AutoRec can't deal with the large-scale historical behavior data of users. Besides, the shallow model structure is hard to extract the deep hidden features of users and items.

This paper proposes a collaborative filtering recommendation algorithm based on improved Stacked Denoising AutoEncoder [11–14]. The hidden representation of users and items is learned from the ratings and auxiliary information through the Stacked Denoising AutoEncoder framework. The deep feature representation ability is extracted to address the inefficiency and sparsity issues of matrix decomposition in traditional collaborative filtering algorithms. In addition, the user and item dimensions are also taken into account, which is able to effectively alleviate the sparse data and cold start of new items, so as to improve the efficiency of the recommendation algorithm. Experiments are done on the movielens dataset and compared with several mainstream algorithms. The results show that the recommendation precision and recall rate of the proposed algorithm are significantly improved, and the cold start problem has been alleviated.

2 Preliminaries

2.1 Autoencoder

The autoencoder [15] is a type of neural network that is commonly used to learn the deep features of input data as shown in Fig. 1. The basic autoencoder consists of an input layer, a hidden layer, and an output layer. The input layer and the output layer have the same number of neurons, while the number of neurons in the hidden layer is typically smaller than the input layer and the output layer. The autoencoder tries to learn an identity function that makes the input and output as equal as possible. The automatic encoder is an unsupervised learning approach, which does not need to mark the training data.

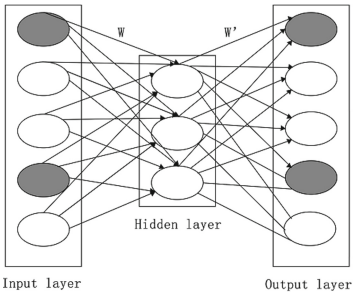

Fig. 1. The network structure of AutoEncoder

The AutoEncoder's working process is elaborated as below. Suppose the training set has sample ratings for m users $\{x_1, x_2, \cdots, x_m\}$, and the rating for each sample $x_i \in R^N$ is an N-dimensional vector. First, each sample rating is encoded to obtain the features of the hidden layer $h^i \in R^L$.

$$h^i = \sigma(Wx_i + b) \tag{1}$$

Where $W \in R^{L*N}$ is the weight matrix of the encoding part,b is the bias vectors, $\sigma(x) = 1/(1 + e^{-x})$ is Sigmoid function indicating that the Sigmoid operation is performed on each dimension of the input x after the encoding. The decoding operation is executed to restore $\hat{x} \in R^N$ from the hidden feature h^i of the L dimension as (2).

$$\hat{x} = \sigma(W'h_i + b') \tag{2}$$

Where $W' \in R^{N*L}$ is the weight matrix of the encoding part, b' is the bias vectors.The training process of the AutoEncoder is to constantly adjust the weight matrix W and W', the offset vector b and b' in order to minimize the objective function as (3).

$$E = \frac{1}{2m}\sum_{i=1}^{m}||x_i - \hat{x}_i|| + \frac{\lambda}{2}||W||^2 + \frac{\lambda}{2}||W'||^2 \tag{3}$$

Where $||x_i - \hat{x}_i||$ is the error term of the input data x and the output data $\hat{x}$ which is used to minimize the error between the output data and the original data. $\frac{\lambda}{2}||W||^2$ and $\frac{\lambda}{2}||W'||^2$ are regular terms, in order to avoid over-fitting the training data. Finally, the hidden layer features h^i are gained through the trained parameters, so that the hidden layer feature codes of the original data can be obtained.

2.2 Denoising AutoEncoder

The AutoEncoder performs pre-training of the model by minimizing the error between the input and output. However, it is easy to learn an identity function from the AutoEncoder due to problems such as model complexity, training set

data volume, and data noise. In order to solve this problem, Vincent proposed Denoising AutoEncoder(DAE) in terms of robustness [16] based on AutoEncoder. In order to prevent the over-fitting problem, random noise is added to the input data, and the process of encoding and decoding by adding noise data is reproduced input.In order to minimize the error between the reconstructed input and the original input, the purpose of DAE is to minimize the loss function.

2.3 Stacked Denoising AutoEncoder

Stacked Denoising AutoEncoder (SDAE) is a deep-structured neural network constructed by stacking multiple DAE [17]. SDAE is used to process larger data sets and extract deeper features of the input data. The training of SDAE network adopts the greedy layer-wise training approach proposed by Hitton [18]. The first layer of the network is trained to get the parameters. The hidden layer output obtained by the first layer is then used as the input of the second layer. When training the next layer, the parameters of the preceding layers remain unchanged. After the training of each layer is completed, the entire network is initialized by the weights during training separately. The output of the layer is used as reconstruction data. Finally, the optimization objective function as Eq. (3) is adopted to adjust the parameters.

3 The Proposed Algorithm

In order to address the data sparseness and cold start issues in traditional collaborative filtering algorithms, two SDAEs are employed to handle the user's ratings - user's auxiliary information and items scores - item auxiliary information [13,14,19] respectively in this paper. The hidden layer's feature is referred as the deep level feature of user and item, which is used to calculate the similarity between users and items.

3.1 User Similarity Calculation

The traditional collaborative filtering algorithms only consider user rating data when performing user similarity calculation, ignoring the user's auxiliary information. There is also a cold start problem for the new user. In addition, the traditional algorithms only consider the shallow features of the user, and cannot extract the deep hidden features of user and item,that results in the low accuracy during the similarity calculation. The proposed algorithm integrates deep neural network SDAE into collaborative filtering. Taking movie recommendation as an example. Suppose there are M users, N movies, and user u scores an integer of 1–5 for movie v, where that R^{m*n} is the user's rating matrix. Three auxiliary information of user, gender, age, and occupation are considered. After discretizing the user's age, the user information matrix $U \in R^{m*l}$ is obtained . Each node of at SDAE input layer represents user's rating on the current movie and the features of the current user. The input data is trained layer by layer

without label to get the parameters of each layer, which are used to extract the deep features of users. The user based network structure of SDAE is defined as U-SDAE, and the item based network structure of SDAE is defined as I-SDAE.

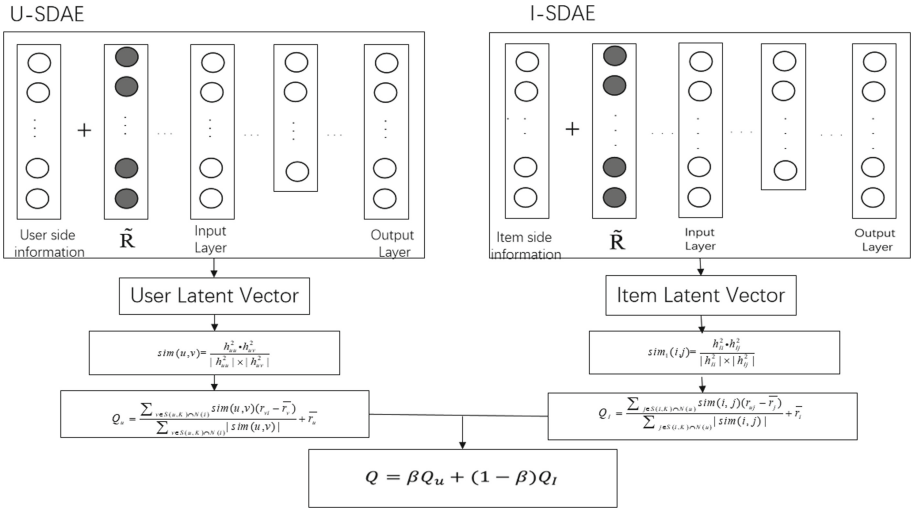

Fig. 2. Improved Collaborative Filtering based on Stacked Denoising AutoEncoders

As shown in Fig. 2, the network structure of SDAE in this paper consists of one input layer, two hidden layers, and one output layer. The algorithm inputs the user information matrix U^{m*l} to generate a user feature vector $U^i \in U^l$, where l is the number of neurons in the input layer, representing a user's score for n items and the characteristics of the current user. The parameters are trained using the automatic encoder training method as follows:

$$h_u^1 = \sigma(W_1 U^T + b_1) \tag{4}$$

$$h_u^2 = \sigma(W_1^{'} h_u^1 + b_1^{'}) \tag{5}$$

$$\hat{U} = \sigma(W_1^{''} h_u^2 + b_1^{''}) \tag{6}$$

Where $W_1 \in R^{k*l}$, $W_1^{'} \in R^{j*k}$ and $W_1^{''} \in R^{l*j}$ is weight matrix. h_u^1 and h_u^2 is the hidden layer feature of the user. $b_1 \in R^{m*1}$,$b_1^{'} \in R^{m*1}$, $b_1^{''} \in R^{m*1}$ are bias vectors. The objective function of learning user's potential features is defined as:

$$E = \frac{1}{2m} \sum_{i=1}^{m} ||U - \hat{U}|| + \frac{\lambda}{2} ||W_1^{'}||^2 + \frac{\lambda}{2} ||W_1^{''}||^2 \tag{7}$$

Where λ is a regularization parameter used to prevent overfitting. By continuously minimizing the objective function, the parameters $\{W_1, b_1\}$ of the first

layer and the output of the first hidden layer are obtained, which forms the input of the next layer. The above training process is continuously repeated to record the parameters of each layer $\{W_1, W_1^{'}, W_1^{''}, b_1, b_1^{'}, b_1^{''}\}$ The trained parameters are then used to calculate h_u^2 through formula (4) and formula (5) in order to compress the original l sample dimension into j dimensional features. Finally, user similarity is calculated with user's low-dimensional feature vector.

$$sim(u,v) = \frac{h_{uu}^2 \bullet h_{uv}^2}{|h_{uu}^2| \times |h_{uv}^2|} \tag{8}$$

Where h_{uu}^2 and h_{uv}^2 represent the j dimensional feature vectors compressed by user u and user v through the SDAE.

3.2 Item Similarity Calculation

In the recommendation system, the auxiliary information of an item is an important indicator to distinguish different items. The traditional collaborative filtering algorithm ignores the contribution of the item attribute to the similarity calculation. The proposed algorithm combines ratings and item attributes to calculate similarities between items. First, the item-attribute matrix is obtained by analyzing the item information. Assuming that the number of items is n and the number of attributes is r, the item-attribute matrix is shown in Table 1. Then, the user's rating matrix and the item attribute matrix are combined to obtain an item information matrix $I^{n \times p}$. Each node of the SDAE input layer represents the scores of the current item by m users and the attribute characteristics of the current item. The input data are trained layer by layer without label to gain the parameters of SDAE network. The parameters are used to extract the deep-seated features of the item. The structure of the SDAE network is similar to that of Fig. 2. The proposed algorithm inputs the item information matrix $I^{n \times p}$ to generate a user feature vector $I^i \in I^p$, where p is the number of neurons in the input layer, indicating that m users have scored the current item and attribute features of the current item. The training process of the I-SDAE model is basically the same as the U-SDAE. After the training is completed, the hidden layer feature h_I^2 of the item is calculated through the trained parameters, which is a feature of compressing the original sample from p dimensional to t dimensional. The learned low-dimensional features include the evaluation information obtained by the item and the attribute features of the item itself, that can express the features of the item in a deeper level. Finally, the low-dimensional feature vector of the learned item is used to calculate the item similarity :

$$sim_1(i,j) = \frac{h_{Ii}^2 \bullet h_{Ij}^2}{|h_{Ii}^2| \times |h_{Ij}^2|}, \tag{9}$$

where h_{Ii}^2 and h_{Ij}^2 represent t dimensional feature vectors that the item i and item j are compressed by the SDAE.

Table 1. Item-Attribute Sheet

	a_1		a_i		a_r
$Item_1$	0	$\cdots$	1	$\cdots$	1
$\cdots$					
$Item_i$	1	$\cdots$	0	$\cdots$	0
$\cdots$					
$Item_n$	1	$\cdots$	1	$\cdots$	0

3.3 Prediction of Comprehensive Score

This paper uses a domain-based scoring prediction algorithm, which first calculates the user-based score prediction. First, formula (8) to calculate the similarity of the user $sim(u, v)$, sort the similarity between the items, and get the set of nearest neighbors of the target user $U_u = \{U_{u1}, U_{u2}, \cdots, U_{uk}\}$, Then user u's score prediction Q_u to item i is:

$$Q_u = \frac{\sum_{v \in S(u,K) \cap N(i)} sim(u, v)(r_{vi} - \bar{r_v})}{\sum_{v \in S(u,K) \cap N(i)} |sim(u, v)|} + \bar{r_u} \tag{10}$$

Where $S(u, K)$ is a collection of K users most similar to the user u's interest, $N(i)$ is a set of users who have scored the item i, $sim(u, v)$ is the similarity between users, $\bar{r_u}$ is the average value of user u's score on all items, r_{vi} is user v's score on item i , $\bar{r_v}$ is the average value of user v ratings on all items he scored.

This paper considers the similarity of the items to predict the score. The Item-based scoring prediction algorithm refers to user u scoring for other items similar to item i. User u's scoring prediction Q_I for item i is:

$$Q_I = \frac{\sum_{j \in S(i,K) \cap N(u)} sim(i, j)(r_{uj} - \bar{r_j})}{\sum_{j \in S(i,K) \cap N(u)} |sim(i, j)|} + \bar{r_i} \tag{11}$$

Where $S(i, K)$ is the most similar set of item i, $N(u)$ is a collection of items that users have scored, $sim(i, j)$ is the similarity between items, $\bar{r_i}$ is the average score of item i. After getting the predicted scores for the two dimensions of user and item, the predicted score for the fusion can be calculated as follows:

$$Q = \beta Q_u + (1 - \beta) Q_I \tag{12}$$

Where: $\beta \in [0, 1]$ is the weight that controls the prediction scores, which should be adjusted in the experiment.

4 Experiments and Analysis

4.1 Datasets

In this paper, movielens dataset[1] is adopted to validate the related recommendation algorithms. The dataset has three scales, where we employ the 1 M scale, including 6040 users, 3883 movies, and 1000209 rating data. Each rating data includes user number, movie number, user rating data, and timestamp. In addition, the movie information includes the name and category of each movie, and the user information includes gender, age, and occupation. In the experiment, we choose 80% of the dataset as training set and the remaining 20% as test set.

4.2 Evaluation Goal

We take the precision rate and recall rate of the recommendation system as the evaluation goal [20]. The precision rate and recall rate is described 13 and 14 respectively:

$$Precision = \frac{\sum_u |R(U) \bigcap T(U)|}{\sum_u |R(U)|} \tag{13}$$

$$Recall = \frac{\sum_u |R(U) \bigcap T(U)|}{\sum_u |T(U)|} \tag{14}$$

Where: $R(U)$ is a list of recommendations for the user based on the behavior of the user on the training set, which is a list of behaviors of the user on the test set.

As shown in Table 2 ,the traditional user-based, item-based, AE, and SDAE schemes are choosen to make the comparative analysis with our proposed algorithm (SDAE-U-I).

4.3 Results Analysis

Figure 3 shows the recall rate as a function of weight. It can be seen from the figure that the β value is around 0.4 to 0.6, and the recall rate is better. In this paper we set the weight β to 0.5. When $\beta = 0$, the algorithm makes a score prediction based on the hidden features of the item learned by SDAE. When $\beta = 1$, the algorithm makes a score prediction based on the hidden features of the user learned by SDAE.

Figure 4 and Fig. 5 show the recall rate comparison between SDAE-U-I algorithm and other five algorithms under different number of neighbors. It can be seen from the figures that there is no linear relationship between the nearest neighbors and the recall rate of the recommended results, where the best number of nearest neighbors is between 80–100. Compared with user-based, item-based, and AE, the recall rate and precision rate of SDAE, SDAE-U, SDAE-I are significantly improved, indicating that the feature extraction effect of deep network

[1] https://grouplens.org/datasets/movielens/1m.

Table 2. Comparison between models

Model	Comparative analysis
Item-based	Item-based Collaborative Filtering. When calculating the similar-ity, the algorithm only uses the user's rating data, does not use the attribute characteristics of the item itself, nor extract the deepfeatures of the item, and does not consider the user's dimension
User-based	User-based Collaborative Filtering. When calculating the similar- ity, the algorithm only uses the user's rating data, does not use the user's own information, does not extract the user's deep-seated features, and does not consider the dimension of the item
AE	Collaborative filtering based on Autoencoder. The algorithm uses only one hidden layer to extract features, and does not integrate user and item dimensions
SDAE-U	SDAE-U integrates user's own information in collaborative filtering based on Stacked Denoising AutoEncoder, applies deep learning tocollaborative filtering, extracts user's deep-seated features, and alleviates the problem of data sparseness and cold start. But this method does not consider the item dimension
SDAE-I	SDAE-I combines the attributes of the item in collaborative filter-ing based on Stacked Denoising AutoEncoder, applies deep learning to collaborative filtering, extracts the deep features of the item, and alleviates the problem of data sparsity and cold start. But this method does not consider user dimension

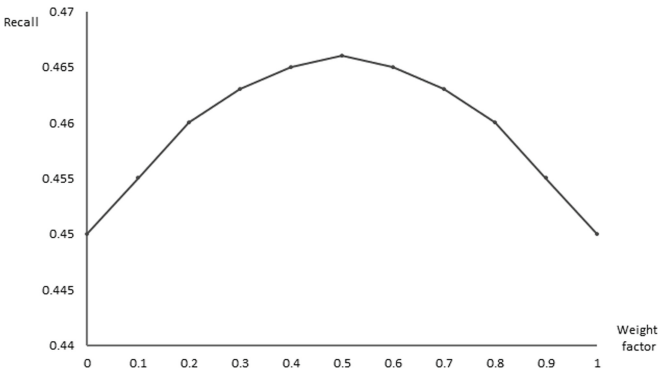

Fig. 3. Effect of different parameters β on recall rate

is better than that of shallow model and improves the quality of the recommendation system. In addition, compared with AE, SDAE-U, and SDAE-I models, SDAE-U-I has improved the precision and recall rate. When we recommend the same length item list, SDAE-U-I has higher precision and more accurate results, which shows that the recommended cold start problem has been alleviated. Moreover, it can be seen from the results that the user characteristics and item characteristics learned from deep network can better replace users and items. Compared with the recommendation algorithm which only considers one dimension of users or items, the recall rate and precision rate are improved, and the recommendation effect is improved.

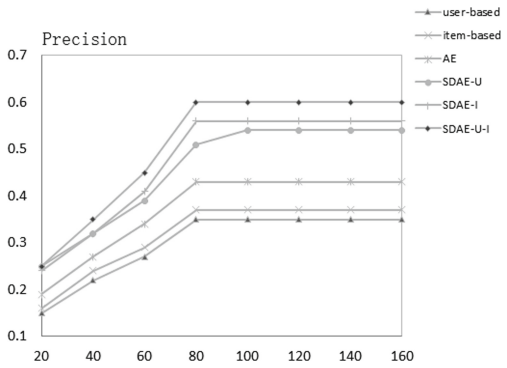

Fig. 4. Precision rate of each algorithm under different neighbors

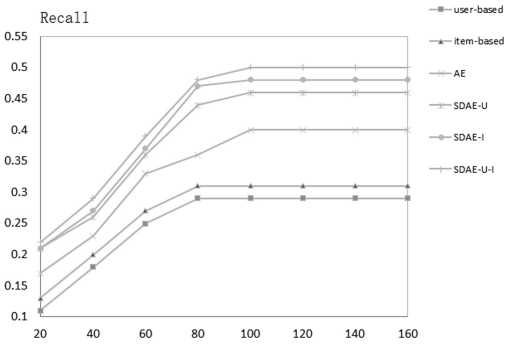

Fig. 5. Recall rate of each algorithm under different neighbors

5 Conclusion and Future Studies

This paper proposes an improved collaborative filtering algorithm with Stacked Denoising AutoEncoders. The information matrix of users and items is trained by two Stacked Denoising AutoEncoders. The hidden feature vectors of users and items are considered, which equipts the proposed algorithm with the recommendation ability for new users or new items. The experimental results show that compared with the traditional methods, the precision and recall rate of the proposed algorithm are higher. To some extent, the issues of data sparseness and the cold start of new items and new users are solved. In addition, it can be seen that the effect of extracting features from deep neural networks is better than that of shallow models. However, we spend a lot of time on data preprocessing, which needs to be improved. When the data volume of users and items gradually increases, how to optimize the computational efficiency of the recommendation algorithm and achieve real-time recommendation will be the focus of the future research.

References

1. Gupta, T., Choudhary, G., Sharma, V.: A survey on the security of pervasive online social networks (POSNs). J. Internet Serv. Inf. Secur. (JISIS) **8**(2), 48–86 (2018)
2. Applying big data processing and machine learning methods for mobile internet of things security monitoring. J. Internet Serv. Inf. Secur. 54–63 (2018)
3. Choudhary, G., Kim, J., Sharma, V.: Security of 5G-mobile backhaul networks: a survey. J. Wirel. Mob. Netw. Ubiquit. Comput. Dependable Appl. **9**(4), 41–70 (2019)
4. Lim, J., Shin, Y., Lee, S., Kim, K., Yi, J.H.: Survey of dynamic anti-analysis schemes for mobile malware. J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. (2019)
5. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2015)
6. Zhang, S., Yao, L.: Sun, A: Deep learning based recommender system: a survey and new perspectives. ACM Comput. Surv. (CSUR) **52**(1), 1–38 (2017)
7. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In: Proceedings of the 26th International Conference on World Wide Web, pp. 173–182 (2017)
8. Sedhain, S., Menon, A.K., Sanner, S., Xie, L: AutoRec: autoencoders meet collaborative filtering. In: Proceedings of the 24th International Conference on World Wide Web, pp. 111–112 (2015)
9. Wu, Y., DuBois, C., Zheng, A.X., Ester, M: Collaborative denoising auto-encoders for top-N recommender systems. In: Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, pp. 153–162 (2016)
10. Zheng, Y., Tang, B., Ding, W., Zhou, H.: A neural autoregressive approach to collaborative filtering. In: Proceedings of The 33rd International Conference on Machine Learning, pp. 764–773 (2016)
11. Vincent, P., Larochelle, H., Lajoie, I., et al.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J. Mach. Learn. Res. **11**(6), 3371–3408 (2010)

12. Strub, F., Mary, J.: Collaborative filtering with stacked denoising AutoEncoders and sparse inputs. In: NIPS Workshop on Machine Learning for eCommerce, Montreal, Canada. (2015). ffhal-01256422v1f
13. Dong, X., Yu, L., Wu, Z., et al.: A hybrid collaborative filtering model with deep structure for recommender systems. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
14. Wei, J., He, J., Chen, K., et al.: Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst. Appl. **69**, 29–39 (2017)
15. Bengio, Yoshua: Learning deep architectures for AI. Found. Trends Mach. Learn. **2**(1), 1–127 (2009)
16. Vincent, P., et al.: Extracting and composing robust features with denoising autoencoders. In: Machine Learning, Proceedings of the Twenty-Fifth International Conference, 5–9 Jun 2008
17. Wang, H., Shi, X., Yeung, D.Y.: Relational stacked denoising autoencoder for tag recommendation. In: Proceedingsof the 29th Conference on Artificial Intelligence, Austin, USA, pp. 3052–3058 (2015)
18. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
19. Zhuang, F., Zhang, Z., Qian, M., et al.: Representation learning via Dual-Autoencoder for recommendation. Neural Netw. **90**, 83–89 (2017)
20. Yuxiao, Z., Linyuan, L.: Summary of evaluation index of recommendation system. J. Univ. Electron. Sci. Technol. **41**(2), 163–175 (2012)