Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 829–836

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

An Efficient Deep Learning Approach for Collaborative Filtering Recommender System

Mohammed Fadhel Aljunid[1], Manjaiah DH[b]

[a]*Department of Computer Science, Mangalore University,India*
[b]*Department of Computer Science,Mangalore University,India*

Abstract

Owing to the enormous growth in information over the past few decades, the world has become a global village. The recommendation system remains the most widely used type of commercial websites. The personalized recommender system is of paramount importance in modeling user's preference on items based on their past interactions (e.g., ratings and clicks), known as collaborative filtering (CF) technique. Although CF is very important among the algorithms used in recommendation systems, it suffers some setbacks such as the sparsity of matrix ratings, scalability, and integrals nature of data. Several research studies have shown that the above-mentioned obstacle could be tackled with the help of matrix factorization (MF) techniques. In spite of the fact that the technique is likely to suffer from lack of some meaningful signals by using a low ranked approximation as well as lack of sparsity in times of denser singular vectors. Recently, deep learning techniques have proven to learn good representation in natural language processing, image classification, and so on. In this work, we propose a deep learning method of collaborative recommender systems (DLCRS). We have made a comparative study of the proposed method and existing methods. Experimental results demonstrate that our approach gives improved results compared to already existing methods. We empirically evaluate DLCRS on two famous datasets: 100K and 1M Movielens.

Keywords: Recommender System , Collaborative Filtering , Matrix Factorization , Deep Learning , Movielens Datasets;

1. Introduction

The advancement of artificial intelligence and machine learning technologies has brought intelligent products that are essential in providing access to various endeavors of peoples' day-to-day life. Effective and useful information

* Corresponding author. Tel.: +917304385644 .
E-mail address: ngm505@yahoo.com

10.1016/j.procs.2020.04.090

from massive internet data could be obtained from intelligent recommendation function of personalized recommender systems thereby making it applicable in sundry network platforms which include, movies, music as well as shopping platforms. One of the paramount forms of a recommendation system is the recommendation algorithm which is a useful determinant in identifying the system performance and the quality of the recommendation results. The commonly used algorithm may be sub-divided into two main classes: content-based filtering (CBF) techniques [1] and collaborative filtering (CF) techniques [2, 3]. CBF analyses and construct portraits of users and items through extra information, such as user-item profiles, and content analysis to construct a recommendation system. In many instances, there is difficulty in obtaining useful information to construct portrait which poses limitations in its applications and performance. CF algorithm is among the most reported algorithm in building recommender systems. Their non-requirement on information about users or items makes them different from CBF techniques and so. On the other hand, the CF algorithm performs accurate recommendation which is based on interactive information between users and items which include browsing, rating, and clicks alone [27]. However, despite the simplicity and effectiveness of this method, the performance of the algorithm is limited due to the high sparsity of data. Therefore, there is a need for alternative methods to improve recommendation performance. Recently, artificial neural networks have recorded remarkable successes recorded in the areas like speech recognition [6], natural language processing [7], and computer vision [8] via Deep Neutral Networks. In spite of these great successes, studies on recommendation systems using these technologies are in their infant stage. There are few research work which is a recommendation system based on deep learning [9]. However, virtually all these models involve the use of additional features like audio information and text content for performance improvement. This poses difficulty in obtaining the aforementioned information for most recommendation systems. In this paper, we developed an efficient deep learning method of collaborative recommender system (DLCRS) that is independent of involving the use of any extra information apart from the interaction between users and items. DLCRS achieves much better performance with repeat to the conventional approaches. The DLCRS model consistently achieves lower RMSE then existing approaches.

The rest of the paper is organized as follows. Section 2 described the related work; a methodology is described in Section 3. Experimental and results are given in Section 4. Finally, a conclusion and future work are given in Section 5.

2. Related Work

The collaborative filtering (CF) technique may be sub-divided according to the following categories viz; memory-based methods and model-based methods as reported by Breese et al.[10]. In memory-based CF, the similarities between users [11] or items [12] are utilized in making recommendations. The memory-based CF method is the most reported method due to its effectiveness and ease with which it is implemented. However, seems difficult which is due to the similarity calculation as well as the scale expansion of the recommendation system. Furthermore, owing to the high data sparsity of the data, there is a limitation in the overall performance of the CF approach. Various recommendation systems which are based on CF model were developed such as latent semantic models [13], regression-based models [12], Bayesian models [14], MF models [16] and clustering models [15], in order to ameliorate the above-mentioned predicaments. Among the various CF technologies, MF techniques are the most popular approach. This approach performs the modeling of users and items to vectors which are similar in dimension. this is taken as a representation of the latent features of the users/items. The concept of this approach may be expressed as Probabilistic Matrix Factorization (PMF) [19], Non parametric Probabilistic Principal Component Analysis (NPCA) [17] and Singular Value Decomposition (SVD) [18]. Nonetheless, there is inefficiency in the latent features learned by the MF technique more especially in the case of the rating matrix is very sparse. Conversely, a greater achievement in natural languages processing and computer vision field has been recorded under the auspices of deep learning techniques. Because of the great potentials exhibited by deep learning methods, current researches report the application of deep learning method into the fields of recommendations. The restricted Boltzmann machine is utilized to replace the conventional MF technique to carry out the CF has been reported by Salakhutdinov et al. [20]. Expansion of work incorporating the correlation between users and between items has been well documented Georgiev et al. [21]. There are also other studies that have developed several approach based on deep learning, however, their main focus was on recommendation system for [5, 22]. the most widely reported approach that are used in learning the music content features are the deep trust networks and traditional convolution neural networks. Other recommendation system such as hierarchical Bayesian model has been proposed by Wang et al. [23]. This is in addition to music recommendation. in addressing the rating information, this model put into cognizance, a deep learning model to obtain content features and a traditional CF models. It may be observed that methods based on deep learning techniques execute recommendations through deeper learning of the content features of items such as text content and the spectrum of music. However, these methods are less used in an event where the content of the items is difficult to obtain. Attempt to propose a new recommendation framework based on deep learning was reported by He, Xiangnan, et al.[24]. The method reported the presentation of users and items using one-hot encoding method of user and items ID. It is obvious that, the method utilizes only the ID information amid the training phase of the model. This poses difficulty in the use of large amount of prior information. There is therefore a difficulty in the effectiveness of feature learning.

3. Proposed Methodology

In this section, we introduce an efficient deep learning approach for collaborative recommender system. The DLCRS model learns the latent features of users and movies as input and uses the foreward propagation method to predict the ratings score.

3.1. Problem Formulation

Let N represents set of users and M represents set of items and the users binary rating matrix $R = [y_{ij}]^{|N|[M]}$, which the elements y_{ij} represent whether user i rates item j, and if user i has a rating record for item j , then $(y_{ij} = 1)$, otherwise, $(y_{ij} = 0)$,

$$y_{ij} = \begin{cases} 1, & \text{if interaction (user i, item j) observed;} \\ 0, & \text{Otherwise} \end{cases} \tag{1}$$

The aim of movie recommendation system for implicit feedback is to generate a movie list that reflects the users preference.

3.2. Proposed Method

The concept in matrix factorization (MF) approaches is that the behaviors of a user can be decided by the intrinsically hidden factors which are embeddings. It is a well-known fact that embeddings are low dimensional hidden factors for the vectors movies and users.

Here, the reason for using deep learning neural networks is that it is similar to that of the MF technique. In the MF technique, the given sparse matrix decomposes into the multiplicative of two low-rank orthogonal matrices. In deep learning neural network experimental setup, we don't need them to be orthogonal, hence it has to learn the values of the embedding matrix by itself from the features matrix.

He et.al[24] proposed a user and movie latent features that are looked up from the embedding matrices in a particular movie-user combination. The explicit characteristics such as a movie's genre and/or the hidden features of movies may be explained by the latent movie factors. A user latent factors measure how much the user likes a movie in terms of the corresponding latent factors. As illustrated in Figure 1 a DLCRS model concatenates the user and movie identities as one-hot vectors i and j respectively. A fully-connected layer deep learning neural network is used as the embedding layer to learn the lower-dimension the embedding process is formulated as shown in equation 3:

$$x_0 = concatenate(U_i, V_j) = \begin{bmatrix} U_i \\ V_j \end{bmatrix} \tag{2}$$

The weight matrices $W_i \in R^{k \times |i|}$ and $W_j \in R^{k \times |j|}$ are integral part of all layers between the input and embedding layers. The DLCRS approach matches both users and movies to the latent factor space with the same dimensionality k. Further, the embedding vectors U and V are fed into a multiplication layer which conducts the element-wise product of U and V. It then outputs a linear interaction vector R that represents the linear user-movie interactions. We formulate it as shown in equation3:

$$R = U \otimes V = (U_1V_1, U_2V_2, ..., U_kV_k) \tag{3}$$

Unlike the MF model that works by user-movie ratings via inner-products, in the deep learning models, it feeds the vector R into a multilayer fully-connected neural network to deeply learn the high-level abstraction of user-movie interactions[32]. After training DLCRS, matrices W_i and W_j represent the latent factors for all users and movies. With the one-hot encoded representation of users and movies, each column of W_i and W_j represents a certain user and movie's latent factors U and V respectively. For a certain user i, a dimension j measures the degree of interest a user has into corresponds factor of the movies. For a certain movie j, each dimension of j measures the extent to which the movie has these factors. Hence, an output vector of R of the element product layer considers the interaction of a user and movie. The fully-connected layer maps the output to show the nonlinear of the model using an activation function namely is relu formulate it as shown in equation 4:

$$\begin{aligned} x_1 &= a(W_1^T x_0 + b_1) \\ x_2 &= a(W_2^T x_1 + b_2) \\ &\cdots \\ x_L &= a(W_L^T x_L + b_L) \end{aligned} \tag{4}$$

where $W_1, W_2, ..., W_L$ and b_1 , b_2 , . . . , $b = L$ represents the weight matrices and biases of each layer, a indicates the activation layers which is ReLU, and $x_1, x_2, ..., x_L$ denote the output of each layer activated by the ReLU function.

DLCRS predicts the probability of reactions of user-movie by using a logistic Sigmoid function to obtain the output int the interval of [0, 1], where 1 represents a user favors a movie, and 0 indicates no reactions, by converting the multi-scale ratings to binary as shown in the equation. 5

$$\phi = \frac{1}{1 + e^{-z}} \tag{5}$$

The user-movies interaction prediction with respect to the probability is shown in equation 6:

$$P_\Theta(y = 1|i, j) \tag{6}$$

where Θ is the neural network weights. We use $\hat{y}$ as the prediction output as shown in equation 7:

$$\hat{y} = Sigmoid(w_L^T x_L + b_L) \tag{7}$$

In our experiments, we take negative samples from the unseen interactions data in the matrix R by a uniform negative sampling technique as declared by zhang et all.[31]. For example in the training dataset $D = \left\{< (u^i, v^i), y^i >\right\}$ and the corresponding predicted output $\hat{y}_i$ (here i denotes the i^{th} example and $i \in 1, 2, 3, ..., |D|$) the loss of the model is

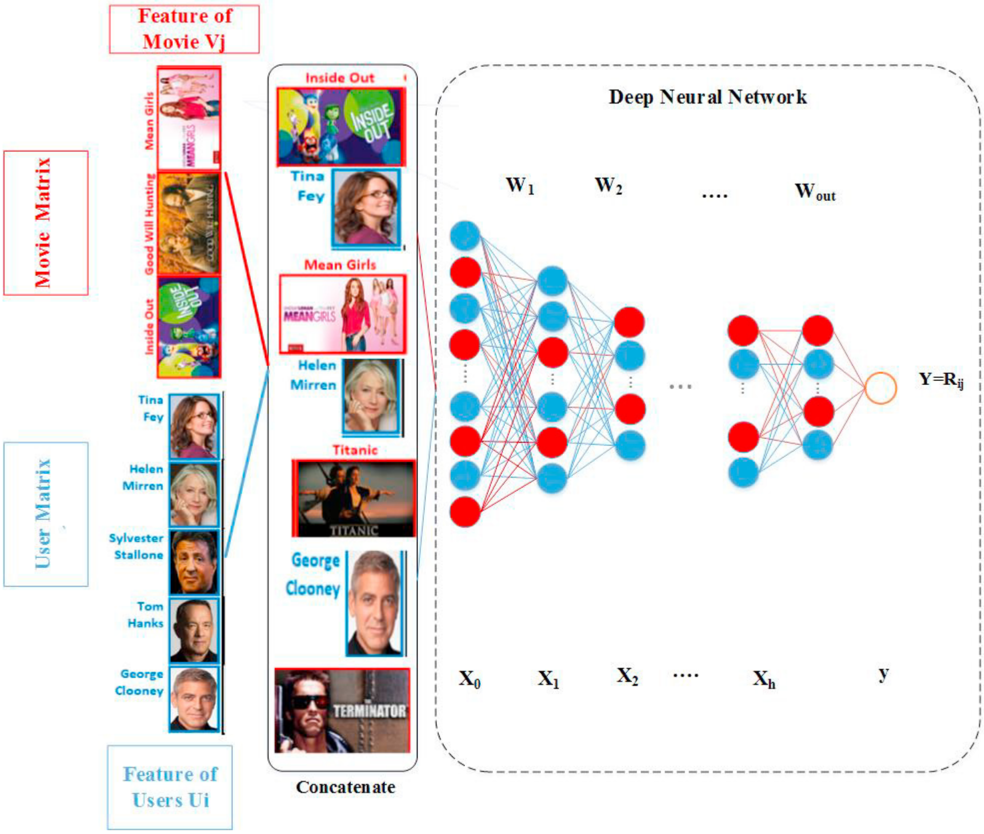


Fig. 1. User and Movie Feature Combined together as Input to Proposed Deep Learning Model, the probability of the ratings will be obtained from the output layer.

computed using equation 8, that computes the difference between the predicted $\hat{y}$ and actual y value.

$$C_f(\hat{y}^i, y^i) = -y^i \log(\hat{y}^i) - (1 - y^i)\log(1 - \hat{y}^i) \tag{8}$$

Finally, we train the proposed deep learning model using the cost function shown in equation 9:

$$L = \frac{1}{n}\sum_{i=1}^{n} C_f\left((\hat{y}^i, y^i)\right) \tag{9}$$

4. Experimental and Results

In the following sub-sections, we detailed our experiments and results. An experimental setup is introduced in sub-section 4.1 , data description is given in sub-section 4.2. Lastly, results and discussion are presented in sub-section 4.3.

4.1. Experimental Setup

We carried out all of our experiments and techniques on the Ubuntu 16.04 system running on Intel® Core™ i5-2400 CPU 3.10 GHz 4 processors and a hard disk of 500 GB. We used the latest released anaconda python 3.5 to simulate the models.

4.2. Data Description

To validate the proposed method we have used two real datasets namely MovieLens 100k and MovieLens1M dataset which are publicly available in (http://www.movielens). The ratings of the datasets are given as an integer from 1 to 5 representing the worst and the best case ratings. The detail description of the datasets is shown in Table 1. Root mean square error (RMSE) metric was used to evaluate the proposed method, (see equation 10) which is known to put more weight on prediction with the larger error.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i,j}(y_{i,j} - \hat{y}_{i,j})^2} \tag{10}$$

Whereby n is the total number of predicted movies, y_{ij} represents the predicted value for user i on item j and $\hat{y}_{ij}$ is the predicted

Table 1. Statistics of MovieLens datasets

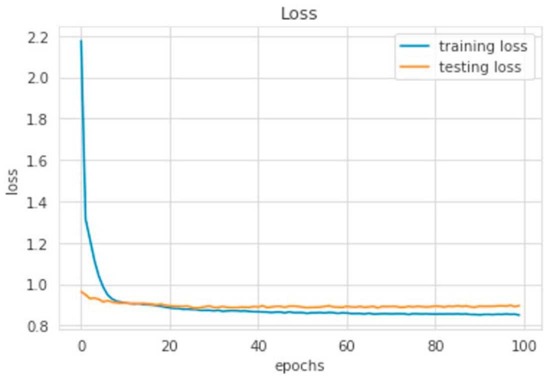
	MovieLens-1M	MovieLens-100K
No of users	6,041	943
No of items	3,952	1,682
No of ratings	1,000,208	100,001
No of ratings per user	165.8	106.6
No of ratings per movies	253.08	59.5
Ratings sparsity	95.80%	93.71%

4.3. Results and Discuss

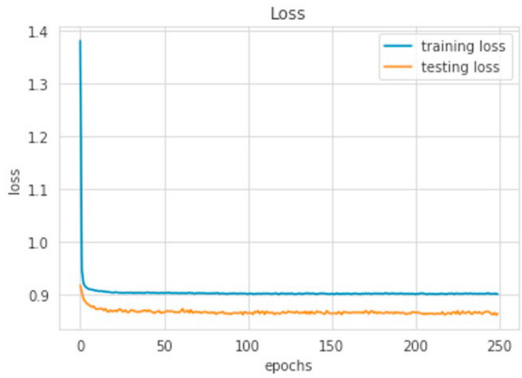
We applied the RMSE measurement metric in this experiment and the results are shown in Figure 2 and Table2. Initially, we randomly split the dataset into train and test data, 80% of the data as the training set and the remaining 20% as the test set. Figure 2(i) illustrates that the RMSE values for the train and test data versus the number of training epochs. To prevent over-fitting, the dropout layer has added to the input and hidden layers of the proposed model. We run the models for 100 epochs, with 64 batch-size. We have concatenated the output of the movies and users' layers to a single layer followed by three fully connected hidden layers with 150 neurons for the first layer, 100 for the second layer and 50 third layer, each before predict the value of the required rating learned the corresponding weights of features, Adam optimization algorithm is used. It is expected that overfitting may occur with a continuously increasing number of epochs.

Figure 2(ii) illustrates the MovieLens 1M dataset that the value of RMSE for train and test data against the number of training epochs . The model for 250 epochs was running with 64 batch-size. Also it is palpable that lower value of RMSE is obtained as training progress. The curve is little bit different compared to 100k dataset due to difference in number of users, items and rating.

Table 2 illustrates the experimental result in terms of different recommendation models on both datasets. Our proposed model is out-performer compared to conventional methods such as User Avg, Movie Avg, Movie-User Avg users-based cosine similarity, items-based cosine similarity, SVD, and the matrix factorization (MF).



(i) Movielens 100K Datasets



(ii) Movielens 1M Datasets

Fig. 2. Performances of the Proposed Deep learning Model by Evaluation Metric of RMSE on Movielens Dataset

The performance measure of MovieLens 1M dataset which presents on the right column of Table 2 is better than MovieLens 100k, which is due to larger training data. The proposed model achieves with lower RMSE with respect to existing approaches. The comparison is given in Figure 2.

Table 2. Prediction Performance Comparison of the Proposed Method with Existing Methods by Evaluation Metric of RMSE on MovieLens Datasets

Model	Movilens 100K	Movilens 1M
User Avg	1.042	1.036
Movie Avg	1.024	0.980
SVD	1.006	0.965
User-based Cosine Similarity	0.995	1.007
Averaged Movie-User Avg	0.983	0.957
Movie-based Cosine Similarity	0.956	0.959
Dot Product (MF)	0.954	0.954
Our Model (DLCRS)	**0.917**	**0.903**

5. Conclusion and Future Work

In the literature, collaboration filtering (CF) plays a vital role in designing and developing recommendation systems. It has a limitation that it suffers the sparsity of the data which represents matrix ratings, scalability, and integrals nature of data. In this paper, we propose a deep learning model for a collaborative recommender system (DLCRS). We have made a comparative study of the proposed deep learning method with existing methods. Our experimental results show that the proposed method depicts better performance compared to the existing methods, which proved that the application of a deep learning approach in the recommender system is a successful attempt. We validated the model on 100k and 1M MovieLens dataset. As future work, we are planning to extend to another deep learning approach such as autoencoder for recommender system and attempt to further improve the performance.

References

[1] FadhelAljunid M, Manjaiah DH. A survey on recommendation systems for social media using big data analytics. International Journal of Latest Trends in Engineering and Technology. 2017:48-58.

[2] Aljunid MF, Manjaiah DH. Movie Recommender System Based on Collaborative Filtering Using Apache Spark. InData Management, Analytics and Innovation 2019 (pp. 283-295). Springer, Singapore.
[3] Aljunid MF, Manjaiah DH. An Improved ALS Recommendation Model Based on Apache Spark. International Conference on Soft Computing Systems 2018 Apr 19 (pp. 302-311). Springer, Singapore.
[4] Zhang L, Luo T, Zhang F, Wu Y. A recommendation model based on deep neural network. IEEE Access. 2018 Jan 5;6:9454-63.
[5] Van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. InAdvances in neural information processing systems 2013 (pp. 2643-2651).
[6] Richardson F, Reynolds D, Dehak N. Deep neural network approaches to speaker and language recognition. IEEE signal processing letters. 2015 Apr 6;22(10):1671-5.
[7] Arisoy E, Sainath TN, Kingsbury B, Ramabhadran B. Deep neural network language models. InProceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT 2012 Jun 8 (pp. 20-28). Association for Computational Linguistics.
[8] Cireşan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification. arXiv preprint arXiv:1202.2745. 2012 Feb 13.
[9] Batmaz Z, Yurekli A, Bilge A, Kaleli C. A review on deep learning for recommender systems: challenges and remedies. Artificial Intelligence Review. 2019 Jun 1;52(1):1-37.
[10] Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. InProceedings of the Fourteenth conference on Uncertainty in artificial intelligence 1998 Jul 24 (pp. 43-52). Morgan Kaufmann Publishers Inc..
[11] Herlocker JL, Konstan JA, Borchers A, Riedl J. An algorithmic framework for performing collaborative filtering. In22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1999 1999 Aug 1 (pp. 230-237). Association for Computing Machinery, Inc.
[12] Sarwar BM, Karypis G, Konstan JA, Riedl J. Item-based collaborative filtering recommendation algorithms. Www. 2001 May 1;1:285-95.
[13] Hofmann T, Puzicha J. Latent class models for collaborative filtering. InIJCAI 1999 Jul 31 (Vol. 99, No. 1999).
[14] Miyahara K, Pazzani MJ. Collaborative filtering with the simple Bayesian classifier. InPacific Rim International conference on artificial intelligence 2000 Aug 28 (pp. 679-689). Springer, Berlin, Heidelberg.
[15] Liu J, Jiang Y, Li Z, Zhang X, Lu H. Domain-sensitive recommendation with user-item subgroup analysis. IEEE Transactions on Knowledge and Data Engineering. 2015 Oct 26;28(4):939-50.
[16] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009 Aug 1(8):30-7.
[17] Yu K, Zhu S, Lafferty J, Gong Y. Fast nonparametric matrix factorization for large-scale collaborative filtering. InProceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval 2009 Jul 19 (pp. 211-218). ACM.
[18] Sarwar B, Karypis G, Konstan J, Riedl J. Application of dimensionality reduction in recommender system-a case study. Minnesota Univ Minneapolis Dept of Computer Science; 2000 Jul 14.
[19] Mnih A, Salakhutdinov RR. Probabilistic matrix factorization. InAdvances in neural information processing systems 2008 (pp. 1257-1264).
[20] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering. InProceedings of the 24th international conference on Machine learning 2007 Jun 20 (pp. 791-798). ACM.
[21] Van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation. InAdvances in neural information processing systems 2013 (pp. 2643-2651).
[22] Wang X, Wang Y. Improving content-based and hybrid music recommendation using deep learning. InProceedings of the 22nd ACM international conference on Multimedia 2014 Nov 3 (pp. 627-636). ACM.
[23] Wang H, Wang N, Yeung DY. Collaborative deep learning for recommender systems. InProceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining 2015 Aug 10 (pp. 1235-1244). ACM.
[24] He X, Liao L, Zhang H, Nie L, Hu X, Chua TS. Neural collaborative filtering. InProceedings of the 26th international conference on world wide web 2017 Apr 3 (pp. 173-182). International World Wide Web Conferences Steering Committee.
[25] Liao CL, Lee SJ. A clustering based approach to improving the efficiency of collaborative filtering recommendation. Electronic Commerce Research and Applications. 2016 Jul 1;18:1-9.
[26] Elahi M, Ricci F, Rubens N. A survey of active learning in collaborative filtering recommender systems. Computer Science Review. 2016 May 1;20:29-50.
[27] Zhang L, Luo T, Zhang F, Wu Y. A recommendation model based on deep neural network. IEEE Access. 2018 Jan 5;6:9454-63.
[28] Deng S, Huang L, Xu G, Wu X, Wu Z. On deep learning for trust-aware recommendations in social networks. IEEE transactions on neural networks and learning systems. 2016 Feb 19;28(5):1164-77.
[29] Koren Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. InProceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining 2008 Aug 24 (pp. 426-434). ACM.
[30] Lee DD, Seung HS. Algorithms for non-negative matrix factorization. InAdvances in neural information processing systems 2001 (pp. 556-562).
[31] Zhang Q, Cao L, Zhu C, Li Z, Sun J. CoupledCF: Learning Explicit and Implicit User-item Couplings in Recommendation for Deep Collaborative Filtering. InIJCAI 2018 Jan 1 (pp. 3662-3668).
[32] Zhang Q, Wang L, Meng X, Xu K, Hu J. A Generic Framework for Learning Explicit and Implicit User-Item Couplings in Recommendation. IEEE Access. 2019 Aug 27;7:123944-58.