



HGAR: Hybrid Granular Algorithm for Rating Recommendation

Fulan Qian¹✉, Yafan Huang¹, Jianhong Li¹, Shu Zhao¹, Jie Chen¹,
Xiangyang Wang², and Yanping Zhang¹

¹ School of Computer Science and Technology, Anhui University,
Hefei 230601, Anhui, People's Republic of China
qianfulan@hotmail.com

² Anhui Electrical Engineering Professional Technique College,
Hefei 230051, Anhui, People's Republic of China

Abstract. Recommendation algorithms based on collaborative filtering show products which people might like and play an important role in personalized service. Nevertheless, the most of them just adopt explicit information feedback and achieve low recommendation accuracy. In recent years, deep learning methods utilize non-linear network framework to receive feature representation of massive data, which can obtain implicit information feedback. Therefore, many algorithms are designed based on deep learning to improve recommendation effects. Even so, the results are unsatisfactory. The reason is that they never consider explicit information feedback. In this paper, we propose a Hybrid Granular Algorithm for Rating Recommendation (HGAR), which is based on granulation computing. The core idea is to explore the multi-granularity of interaction information for both explicit and implicit feedback to predict the users ratings. Thus, we used Singular Value Decomposition model to get explicit information and implicit information can be received by multi-layer perception of deep learning. In addition, we fused the two part information when the two models are jointly trained. Therefore, HGAR can explore the multi-granularity of interaction information which learned explicit interaction information and mined implicit information in different information granular level. Experiment results show that HGAR significantly improved recommendation accuracy compared with different recommendation models including collaborative filtering and deep learning methods.

Keywords: Information granular · Information feedback · Rating recommendation

1 Introduction

As a tool to help users find useful information quickly, the recommendation algorithm solves information overload and implements personalized recommendation, so it has many application scenarios and commercial values. However, with the

rapidly growth of the amount of data, methods based on collaborative filtering encountered some problems. For example, users' preferences cannot be easily obtained, so it is impossible to achieve good recommendation accuracy. Many researchers try to find useful information to improve recommendation accuracy in big data. This kind of demand promotes the application and development of particle computing theory. As shown in Fig. 1, the interactive information can be divided into explicit information feedback and implicit information feedback in the user-item bipartite graph. Explicit information includes ratings, purchases, friends, follow-ups, and other information that actually happens. While implicit information feedback is the relationships and information hidden behind the actual data, such as browsing, clicking, adding to the shopping cart, etc. In general, explicit information can more directly reflect user preferences. However, explicit information is difficult to obtain and data volume is small. The amount of implicit feedback information is large, easy to obtain, but also can tap into the user's more interests. According to information granulation, we can think of information as consisting of explicit information and implicit information. Therefore, from the perspective of information granulation, by effectively mining the explicit information granule and implicit information granule, the recommendation effect can be better improved.

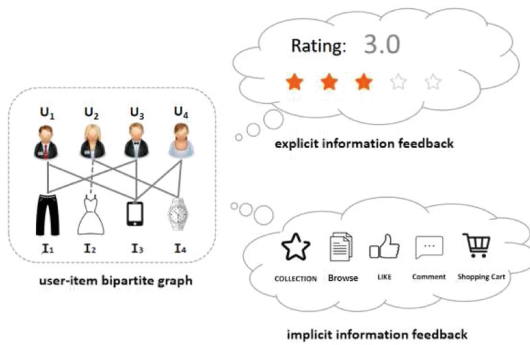


Fig. 1. An example of our proposed recommender system based on explicit and implicit information feedback

Since the information can split different granularity, we can use the idea of granulation to solve the problems in the recommendation system. For example, item-based and user-based recommendation algorithms actually granulate the user set or the item set in the form of targeted user's nearest neighbor. The granulation method is introduced in three-way decision, which uses the explicit information feedback to reflect the information granular. At present, many scholars try to solve the recommendation problem with three-way decision. Huang et al. [1] presented a three-way decision method for recommendation which considers the variable cost as a function of project popularity. Zhang et al. [2]

proposed a regression-based three-way recommender system that aims to minimize the average cost by adjusting the thresholds for different behaviors. Xu et al. [3] designed a model that adds a set of items that may be recommended to users. Zhang et al. [4] created a framework that integrates three-way decision and random forests to build recommender systems. Qian et al. [5] proposed a three-way decision collaborative recommendation algorithm based on user reputation by giving each user a corresponding reputation coefficient. These methods make rating prediction, but the accuracy is not good. Therefore, only relying on explicit information feedback is not a good solution.

Due to the powerful capacity of mining implicit information, deep learning techniques have gained much success in many domains. Therefore, much effort has been made to introduce deep learning techniques to rating recommendations. Cheng et al. [6] jointly trained wide linear models and deep neural networks to combine the benefits of memorization and generalization for recommender systems. Guo et al. [7] combines Factorization Machine (FM) with Deep Neural Networks (DNN) to improve the model ability of learning feature interactions. Covington et al. [8] proposed deep neural network to learn both user and item's embedding, which is generated from their corresponding features separately. However, the above method of deep learning uses implicit information feedback and does not consider explicit information feedback. Therefore, the recommended results could not receive superior accuracy.

To address the challenges we mentioned, in this paper, we propose a hybrid granular algorithm for rating recommendation (HGAR), by combining the advantages of explicit and implicit information feedback to achieve the effect of combinatorial optimization. Explicit information feedback is obtained by user ratings while implicit information is trained by deep learning framework. We can further get new granular by fusing these two information granularity. For a large number of data, HGAR reduced irrelevant information of data and extracted the most accurate user preferences to acquire better recommendation effect. Experiments demonstrate that our model outperforms the compared methods for rating recommendation.

The following sections of this paper are organized as follows: Sect. 2 introduces the problem formulations for quotient space attribute sets; Sect. 3 describes hybrid granular algorithm for rating recommendation in detail; Sect. 4 presents the experimental results and analysis; Sect. 5 is the conclusion of the full paper.

2 Problem Formulation

According to the idea of granulation, we turn the interactive information granulation into explicit information feedback and implicit information feedback. The granular computing theory abstracts the problems into triples to describe them, and then solving them from different granular. Then discussing the representation of different domain attribute in different granularity, and exploring the interdependence and transformation of these representations. In this paper, we define information granular notations of data.

Let $\{x_1, x_2, x_3, \dots, x_n\}$ denotes interactive information attribute, n is referred to as the number of attributes. For a recommendation system, X contains explicit information and implicit information based on previous discussions. So we can formulate the equation $X = X_1 + X_2$, X_1 is explicit information granule and X_2 is implicit information granule. Thus, we define $x_i \in X_j$ as interactive information attribute is classified into explicit and implicit, in which $i \in \{1, 2, 3, \dots, n\}$, $j \in \{1, 2\}$. And Y denotes the domain of the rating values. The domain of ratings is made on a 5-star scale (whole-star ratings only). Besides $f : X \rightarrow Y$ is a property function, and if f is a single value, then f can be used to define the partition. Generally speaking, we can easily figure out the structure of Y . For example, if Y is a set of real numbers or Euclidean space, we can define the corresponding classification in Y by using the information feedback of X (i.e. taking different information granularity for rating).

The method is as follows: define $X_j = \{x_i | f(x_i) \in Y\}$, $i \in \{1, 2, 3, \dots, n\}$, $j \in \{1, 2\}$. So $\{X_j\}$ is a partition of X . Specifically, the notion of explicit information granule can be defined as: $X_1 = f_{explicit}(x_i)$, and the corresponding method is described by the information particle as $Y = f(X_1)$. Similarly, the notion of implicit information granule is $X_2 = f_{implicit}(x_i)$. And $Y = f(X_2)$ is the method described by explicit information particles. To sum up, the final output Y is defined as: $Y = f(x_1) + f(x_2)$, the framework is shown in Fig. 2. In the following sections, we will introduce the detail operation of this algorithm framework.

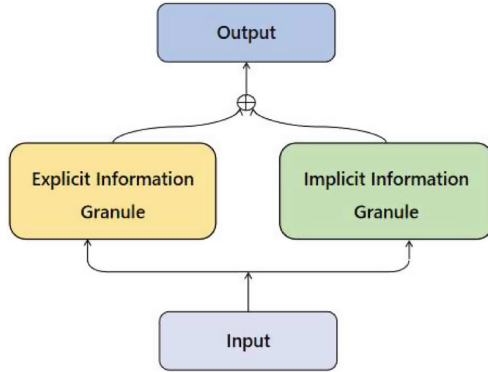


Fig. 2. The basic framework of HGAR

3 Hybrid Granular Algorithm for Rating Recommendation

In this section, we use Singular Value Decomposition (SVD) to represent the explicit information granule and Multi-layer Perceptron (MLP) to represent the implicit information granule. We first present how SVD and MLP worked separately and explain how they serve as a rating recommendation framework.

Figure 5 depicts the architecture of the proposed hybrid granular model. Then, we fuse these modules to predict ratings through the HGAR model which has been trained.

Embedding Layer. We adopt an embedding layer to present user and item. The user-id and the item-id are input information that needs to be preprocessed before entering the model. This is done by mapping the input information to a dense vector. In this way, we can obtain $uemb$ as a set of feature vector from user, and $iemb$ as a set of feature vector from item. The processing of the embedded layer is represented as follows:

$$uemb = embedding_lookup(userid) \quad (1)$$

$$iemb = embedding_lookup(itemid) \quad (2)$$

Where $embedding_lookup$ represents the embedding operation, $userid$ and $itemid$ are the input of embedding layer, $uemb$ and $iemb$ are the output vectors.

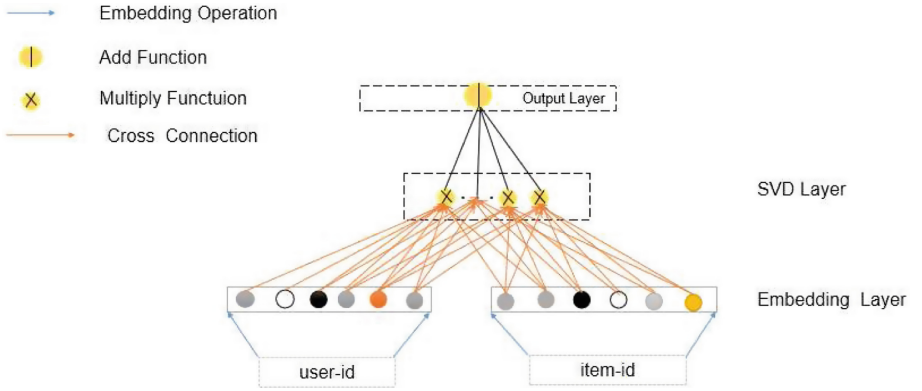


Fig. 3. The architectures of SVD layer for explicit information feedback

3.1 SVD Layer

In this layer, we take advantage of explicit feedback from user and item to implement rating prediction. The model of SVD layer is shown in Fig. 3. SVD is a matrix factorization method. The high dimensional user-item rating matrix is converted into two low dimensional user factor matrices and item factor matrices. In order to obtain feedback information to obtain the user's rating of the item. The formula is shown in:

$$X_1 = f_{explicit}(uemb, iemb) \quad (3)$$

where $f_{explicit}$ is \cdot operation.

The rating consists of four components: global average, user bias, item bias and user-item interaction. The following equation shows the calculation process:

$$\widehat{r}_{ui} = \mu + b_i + b_u + X_1 \quad (4)$$

Where the rating \widehat{r}_{ui} is the output of the SVD layer, μ denotes the overall average rating, b_i and b_u respectively indicate the observed deviations of user u and item i . Obviously, SVD directly adopts explicit information feedback (rating information) to adjust model prediction errors and to get better recommendation accuracy.

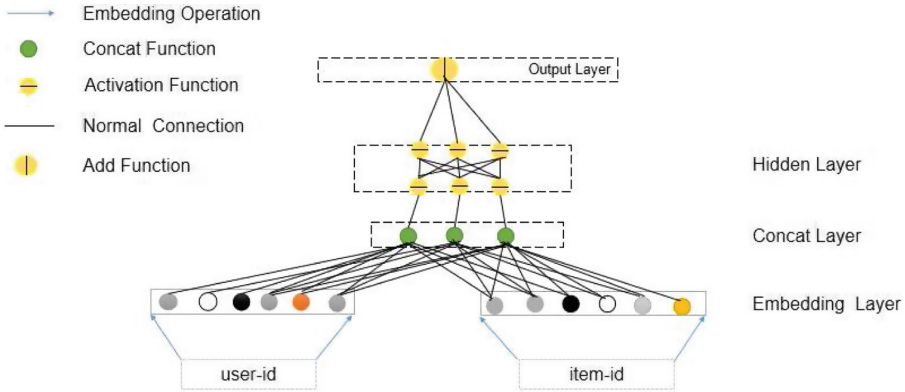


Fig. 4. The architectures of deep component for implicit information feedback

3.2 Deep Component

Contact Layer. Before mining the implicit information, we have preprocessed the embedding vector. After that, we need to adopt a contact layer to concatenate $uemb$ and $iemb$ into one vector. Mapping the two vectors to a vector space and reducing data dimension. The formulation is shown by:

$$\alpha = uemb \oplus iemb \quad (5)$$

where \oplus represents the concat operation, α is the output of contact layer.

Hidden Layer. The MLP model is designed to learn implicit information from hidden layer, as shown in Fig. 4. It consists of an input layer, an output layer and a number of hidden layers. In the process of model training, the embedded vector is randomly initialized firstly, and then the value of the embedded vector is trained to minimize the loss function. These low-dimensional dense embedding vectors are fed into the hidden layer of the neural network in the forward

channel. MLP can enhance the expressiveness of the model through multiple hidden layers, but it also increases the complexity of the model. High-dimensional features can be converted into a low-dimensional but dense valuable features by multi-layer. According to the definition of implicit information particles, the hidden layer denotes as:

$$X_2 = f_{implicit} \left(W^{(l+1)} \alpha^l + b^l \right) \quad (6)$$

$$\alpha^{l+1} = f(X_2) \quad (7)$$

Where $f_{implicit}$ denotes non-linear activation, l is the number of layer, W^l , b^l , α^l are the l -th weight, the l -th bias, the l -th input. f shows the linear activation function.

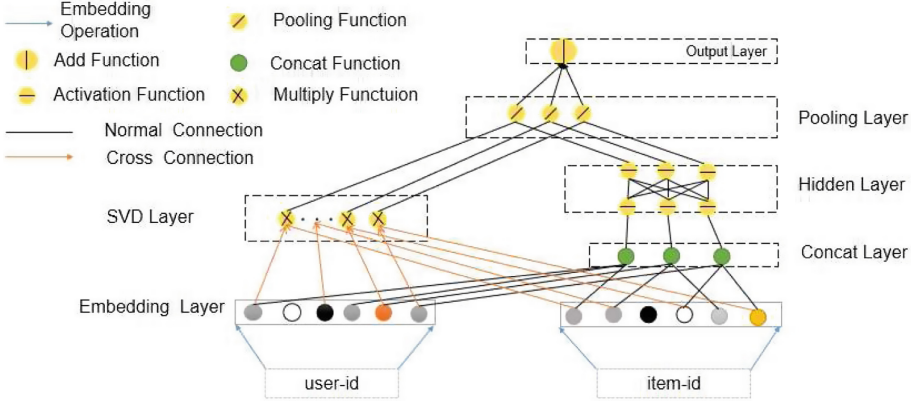


Fig. 5. The model of hybrid granular algorithm for rating recommendation

3.3 Joint Training of HGAR Model

Pooling Layer. Through the previous operation, we obtained the explicit rating and the implicit rating respectively. Now, we need to convert ratings with sum pooling to descend to 1-dimension. The operation is defined as follows:

$$m = \sum_i^n e_i, \forall i = 2, 3, \dots, n \quad (8)$$

where e_i represents the i -dimension vector of input, and m is the 1-dimension output.

Output Layer. Finally, we combine both explicit and implicit rating into a single vector representation to predict the final rating. The output after fusion is formulated as:

$$\widehat{R}_{u,i} = f(m_{ui}^{SVD}, m_{ui}^{MLP}) \quad (9)$$

Where $\widehat{R}_{u,i}$ denotes the user rating for a specific item, m_{ui}^{SVD} is the pooling result of the SVD model, m_{ui}^{MLP} is the pooling result of the deep component.

4 Experimental Analysis

In this section, we present our experimental setup and empirical evaluation. We aim to answer the following questions in our experiments:

Q1: How does HGAR perform in terms of efficiency and effectiveness, compared to other state-of-the-art methods based on explicit feedback?

Q2: How does HGAR perform as compared to the state-of-the-art deep learning methods based on implicit feedback?

Q3: How do Singular Value Decomposition (SVD) and Multi-layer Perceptron (MLP) affect the performance of HGAR?

4.1 Data Description

We perform experiments on two well-known and widely used datasets in recommendation: Movielens-100k and Movielens-1M. In Movielens-100k dataset, it contains nearly 100,000 rating records of 943 users on 1,682 movies. As Movielens-1M dataset contains UserIDs which ranged between 1 and 6040 and MovieIDs which ranged between 1 and 3952. Ratings are made on a 5-star scale (whole-star ratings only). Each user has at least 20 ratings. We divide the dataset into training and test set as 8:2, and we use 5-fold cross-validation to get the average results. The basic statistical information of two datasets are illustrated by Table 1.

Table 1. Statistics of the MovieLens datasets

	MovieLens 100k	MovieLens 1M
Users	943	6,040
Items	1,682	3,952
Ratings	100,000	1,000,209
Ratings of per user	106.4	165.6
Rating of per item	59.5	253.1
Rating Sparsity	93.7%	95.8%

4.2 Evaluation Metrics

We use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate the prediction performance of all algorithms.

$$MAE = \frac{\sum_{(u,i) \in N} |R_{u,i} - \widehat{R}_{u,i}|}{|N|} \quad (10)$$

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in N} (R_{u,i} - \widehat{R}_{u,i})^2}{|N|}} \quad (11)$$

where N denotes the whole number of ratings, $R_{u,i}$ denotes the rating user u gives to item i , and $\widehat{R}_{u,i}$ denotes the rating user u gives to item i as prediction. The smaller values of MAE and RMSE indicate the better performance.

4.3 Baselines

We compared our method with the following baseline methods, including the state-of-the-art recommendation methods and the proposed model with its two parts (SVD and MLP). Below we provide the names of algorithms as well as its brief introduction that will be used in the following experiments.

- **SVD**: A classical SVD algorithm based on user and item bias.
- **MLP**: A traditional neural network to solve the nonlinear problem that is trained by error backpropagation.
- **PMF** [9]: Probabilistic matrix factorization model, which is a widely used matrix factorization model.
- **BPMF** [10]: Bayesian probabilistic matrix factorization for recommendation.
- **RLMC** [11]: A new robust local matrix completion algorithm that characterizes the bias and variance of the estimator in a finite sample setting.
- **RegSVD** [12]: A rating prediction algorithm based on SVD.
- **PRMF** [13]: A novel recommendation method that can automatically learn the dependencies between users to improve recommendation accuracy.
- **TWDA** [5]: A three-way decision method to process the boundary region and divided all ratings in boundary region into positive region or negative region reasonably.
- **PRA** [14]: Probabilistic rating auto-encoder that uses autoencoder to generate latent user feature profiles.
- **CDAE** [15]: A novel method called collaborative denoising auto-encoder for top-N recommendation that utilizes the idea of denoising auto-encoders.
- **SR^{imp}** [16]: Exploiting users implicit social relationships for recommendation.
- **SVD++** [17]: Merging the latent factor model and neighborhood model for recommendation.
- **Wide and Deep** [6]: Jointly trained wide linear models and deep neural networks to combine the benefits of memorization and generalization for recommender systems.

- **Hybird IC-CRBMF** [18]: An improved item category aware conditional restricted Boltzmann machine frame model for recommendation by integrating item category information as the conditional layer.
- **HACF** [19]: A fundamentally new architecture of hierarchical autoencoder where each layer reconstructs and provides complimentary information.
- **HGAR**: Our proposed method combines SVD and MLP to obtain explicit and implicit information simultaneously, which further improves recommendation accuracy.

4.4 Comparison of Performance with Other State-of-the-art Methods Based on Explicit Feedback (Q1)

The Table 2 represents all MAE results of two data sets based on explicit information feedback. From the results, we can see clearly that: Results for MAE, HGAR outperforms all other methods based on explicit information feedback. To be specific, HGAR is equal to TWDA on Movielens-100k, but HGAR shows an improvement of 2% compared to TWDA on Movielens-1M. This shows that our model is better at large data sets. The results reveal that other methods only based on explicit feedback cannot obtain higher precision. Thus, our method of hybrid granular which combines explicit and implicit features has better performance on MAE.

Table 2. Experimental performance MAE metrics of HGAR compared to explicit feedback baselines on the MovieLens datasets.

	MovieLens 100k	MovieLens 1M
PMF	0.782	0.690
BPMF	0.881	0.680
RLMC	0.760	0.736
RegSVD	0.733	0.698
PRMF	0.721	0.673
TWDA	0.717	0.670
HGAR	0.717	0.668

4.5 Comparison of Performance with Other State-of-the-art Methods Based on Implicit Feedback (Q2)

Table 3 shows the performance of HGAR compared with other algorithms for implicit feedback. The benchmark algorithms, for example, SP, SVD++, Wide and Deep, they all take advantage of implicit information feedback for rating recommendation. We compared HGAR with them and obtained better experimental results. In particular, the result of HACF on Movielens-100k is the same as ours, but on the 1M dataset, our result is better. Similarly, on Movielens-1M, SVD++ is equal to us, but in the 100k dataset, we show an advantage.

Table 3. Experimental performance MAE metrics of HGAR compared to implicit feedback baselines on the MovieLens datasets.

	MovieLens 100k	MovieLens 1M
PRA	0.759	0.714
CDAE	0.735	0.691
SR ^{imp}	0.729	0.674
SVD++	0.726	0.668
Wide and Deep	0.723	0.671
Hybird IC-CRBMF	0.719	0.681
HACF	0.717	0.681
HGAR	0.717	0.668

Given all above analysis, our approach makes a good result on two public real-world datasets, which could explain that the granulation of explicit and implicit information plays an important role and brings a significant improvement.

4.6 The Impact of SVD and MLP (Q3)

SVD and MLP are two parts of our model, thus we experiment these two separate algorithms to make sure whether combination is better. From Table 4, we can see that HGAR makes significant improvements compared to the MLP, whatever MAE or RMSE on MovieLens-100k or 1M. Meanwhile, as shown in Table 4 compared to SVD, the MAE value of HGAR is better with 0.1% in MovieLens-1M and poorer with 0.5% in MovieLens-100k. In addition, the RMSE and MAE values of HGAR show good results in MovieLens 1M. Thus, we find that SVD only gets explicit feedback as well as MLP merely obtained implicit feedback. They all perform badly because merely from a single attribute perspective is not as good as from the idea of multi-granularity decomposition to recommend.

Table 4. Experimental performance of SVD and MLP on the MovieLens datasets.

Dataset	Movielens 100k		Movielens 1M	
	MAE	RMSE	MAE	RMSE
SVD	0.718	0.916	0.673	0.861
MLP	0.741	0.947	0.716	0.910
HGAR	0.717	0.921	0.668	0.856

5 Conclusion

In this paper, we proposed Hybrid Granular Algorithm for Rating Recommendation. Considering the large amount of data in the recommendation system, we

put the problem on the space of different granularity for analysis and research. To make full use of information granularity, we study the attributes of interactive information and conclude that it can be divided into explicit information and implicit information. In this way, the fine-grained and precise user preferences can be captured. Results on two public datasets show that the proposed model produces comparative performance compared to state-of-the-art methods based on explicit or implicit information feedback.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China (Grants #61673020 #61702003 #61876001), the Provincial Natural Science Foundation of Anhui Province (Grants #1808085MF175). The authors would like to thank the anonymous reviewers for their valuable comments.

References

1. Huang, J., Wang, J., Yao, Y., Zhong, N.: Cost-sensitive three-way recommendations by learning pair-wise preferences. *Int. J. Approx. Reason.* **86**, 28–40 (2017)
2. Zhang, H.R., Min, F., Shi, B.: Regression-based three-way recommendation. *Inf. Sci.* **378**, 444–461 (2016)
3. Xu, Y.-Y., Zhang, H.-R., Min, F.: A three-way recommender system for popularity-based costs. In: Polkowski, L., et al. (eds.) *IJCRS 2017. LNCS (LNAI)*, vol. 10314, pp. 278–289. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60840-2_20
4. Zhang, H.R., Min, F.: Three-way recommender systems based on random forests. *Knowl. Based Syst.* **91**, 275–286 (2016)
5. Qian, F., Min, Q., Zhao, S., Chen, J., Wang, X., Zhang, Y.: Three-way decision collaborative recommendation algorithm based on user reputation. In: Mihálydeák, T., et al. (eds.) *IJCRS 2019. LNCS (LNAI)*, vol. 11499, pp. 424–438. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22815-6_33
6. Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., et al.: Wide & deep learning for recommender systems. In: *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 561–568. ACM(2009)
7. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017)
8. Covington, P., Adams, J., Sargin, E.: Deep neural networks for youtube recommendations. In: *2016 ACM Conference on Recommender Systems*, pp. 191–198. ACM (2016)
9. Salakhutdinov, R.: Probabilistic matrix factorization. In: *2008 Advances in neural information processing systems*, pp. 1257–1264. ACM (2008)
10. Salakhutdinov, R., Mnih, A.: Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In: *2008 International Conference on Machine Learning*, pp. 880–887(2008)
11. Sabetsarvestani, Z., Kiraly, F., Miguel, R., Rodrigues, D.: Entry-wise matrix completion from noisy entries. In: *2018 European Signal Processing Conference (EUSIPCO)*, pp. 2603–2607. IEEE(2018)
12. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: *Proceedings of KDD cup and workshop*, vol. 2007, pp. 5–8. ACM (2007)

13. Liu, Y., Zhao, P., Liu, X., Wu, M., Li, X. L.: Learning user dependencies for recommendation. In: 2016 International Joint Conference on Artificial Intelligence, pp. 2379–2385 (2017)
14. Liang, H., Baldwin, T.: A probabilistic rating auto-encoder for personalized recommender systems. In: 2015 ACM International on Conference on Information and Knowledge Management, pp. 1863–1866. ACM (2015)
15. Wu, Y., DuBois, C., Zheng, A. X., Ester, M.: Collaborative denoising auto-encoders for top-n recommender systems. In: 2016 ACM International Conference on Web Search and Data Mining, pp. 153–162. ACM (2016)
16. Ma, H.: An experimental study on implicit social recommendation. In: 2013 ACM SIGIR conference on Research and development in information retrieval, pp. 73–82. ACM (2013)
17. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: 2008 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM (2008)
18. Liu, X., Ouyang, Y., Rong, W., Xiong, Z.: Item category aware conditional restricted boltzmann machine based recommendation. In: Arik, S., Huang, T., Lai, W.K., Liu, Q. (eds.) ICONIP 2015. LNCS, vol. 9490, pp. 609–616. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26535-3_69
19. Maheshwari, S., Majumdar, A.: Hierarchical autoencoder for collaborative filtering. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE (2018)