A Recommendation Algorithm for Improved Residual Networks Based on Matrix Factorization

Chengzhi Mao and Zhifeng Wu(✉)

School of Information Technology Engineering, Tianjin University of Technology and Education, Tianjin, China
277691544@qq.com, zhifeng.wu@163.com

Abstract. Recommender system is widely used in e-commerce, news consulting, social networking, tourism, film, music and other fields because it can effectively deal with the problem of information overload. According to the practical problems in the recommendation system, various recommendation algorithms are produced. Traditional recommendation algorithms are divided into content-based recommendation, collaborative filtering based recommendation and hybrid recommendation. Collaborative filtering algorithms are favored because they can extract and process relevant information features to accurately predict user preferences. However, collaborative filtering algorithms generally have problems such as sparse data, cold start and data scalability. This paper presents the Associated Residual Matrix Factorization (ARMF) model, which can solve the common cold start problem and improve the recommended performance. The model uses associative residual neural networks (ARNs) to model higher-order interactions between the user and the underlying features of the item and to avoid network layer gradients exploding or disappearing and overfitting problems. In order to verify the validity and correctness of the model, this paper conducts tests on Movielens100k and Movielens10M datasets, and the experimental results show that the prediction results are 17%~ 23% better than other comparison algorithms.

Keywords: Recommendation algorithm · Matrix factorization · residual network

1 Introduction

With the rapid development of information technology, data is constantly iterated and expanded, and there are more and more application software based on data, which is very dependent on data [1]. The problem of "Information Overload" occurs when software can no longer handle the growing amount of data [2]. It is the key to solve the problem of "information overload" that how to find out the important information data to meet the demand in the massive data. Recommendation System (RS) [3, 4] can effectively filter information data and help users retrieve information that meets their needs in a personalized way [5], thus alleviating the problem of information overload. The core of the recommendation system is the recommendation algorithm. The traditional.

Y. Cao and X. Shao (Eds.): DIONE 2023, LNICST 515, pp. 27–36, 2025.
https://doi.org/10.1007/978-3-031-80713-8_2

recommendation algorithm is divided into three categories: content-based recommendation, collaborative filtering recommendation and hybrid recommendation. The content-based recommendation system takes the user history record as reference, mining the relationship between unknown record and known record to make recommendation. The recommendation system based on Collaborative Filtering (CF) is based on the preferences of similar users as a reference, mining the correlation between different users' preferences, and recommending possible favorite items for users with high correlation. Based on hybrid recommendation is based on both content recommendation and collaborative filtering recommendation.

Collaborative filtering algorithm is a common recommendation algorithm in the recommendation system. Among many collaborative filtering based recommendation algorithms, matrix decomposition algorithm [6] has become the most common collaborative filtering recommendation algorithm due to its simplicity and ease of implementation. Compared with other collaborative filtering algorithms, matrix decomposition algorithm can effectively solve the data sparsity problem. The user rating matrix is decomposed into user potential relationship matrix and item potential relationship matrix, that is, users and items are mapped into the shared potential feature vector space, and the relationship between users and items is expressed through the inner product between mapping vectors [7, 8]. In order to further improve the performance of the algorithm, many researchers have improved it. Badrul et al. [9] proposed a singular value decomposition (SVD) algorithm to learn the feature matrix, but SVD algorithm is easy to overfit. Later, Pan et al. proposed a regularized least squares optimization algorithm WR-MF with case weights [10]. The overfitting problem of the model is solved. Considering the asymmetric similarity between items, literature [11] proposes to perform matrix decomposition through trust perception to improve the accuracy of the recommendation system. Literature [12] and [13] respectively proposed to use the relationship between items and item attributes to perform matrix decomposition and an asymmetric similarity method among users to perform matrix decomposition, so as to achieve the recommended effect. However, these algorithms adopt the simple inner product method to construct the relationship between users and items, which may not be enough to effectively express the nonlinear relationship between features.

In order to solve the nonlinear problem of the model, the researchers introduced human Deep Neural Networks (DNN) for feature extraction. He et al. [14] introduced neural networks on the basis of Generalized Matrix Factorization (GMF). On the basis of MF, the full connection layer of neural network is added to the model to give the model the ability to learn the nonlinear relationship between users and items, and to improve the model expression ability. However, single-layer neural networks are not effective at capturing higher-order interactions between users and items that contain richer information. To solve this problem, Tian et al. [15] proposed a Deep Matrix Factorization model (DMF) combining deep neural network and matrix factorization technology. DMF adds multiple hidden layers after the fully connected layer of the neural network to model high-level interactions between users and items.

Based on the high order interaction characteristic that deep neural networks can build models, adding multiple hidden layers to different models has become a research craze. The VGG model was proposed by the Visual Geometry Group of Oxford University [16].

The model increases the network layer to 19 layers for the first time, and successfully improves the accuracy of image recognition. However, the accuracy of the network may decrease as the number of layers of the network increases. In deep learning, the increase in the number of network layers is generally accompanied by three major problems: computing resource consumption, easy overfitting of models, and gradient disappearance/gradient explosion.

In order to solve the above problems, the Associated Residual Matrix Factorization (ARMF) model is proposed in this paper. Different from the traditional residual network, the ARMF model retains the features of the previous layer network, adds the output of the current hidden layer, adjusts the number of neurons between the front and back layers by using biased weights, and iterates to obtain the final output.

The main contributions of this paper are as follows:

(1) Put forward the ARMF model, establish the high-order nonlinear relationship of the potential relationship between users and items, solve the overfitting of the model and the problem of gradient explosion or gradient disappearance during training, and improve the accuracy of recommendation.
(2) The biased weight is proposed to adjust the number of neurons between different layers, so as to solve the problem that the residual calculation cannot be constructed due to the different number of neurons in the network layer.
(3) The model algorithm is applied to movie recommendation to recommend more favorite movies for users.

2 ARMF

2.1 Problem Definition

The main task of recommendation algorithm is to recommend interesting items for users according to the characteristic information of users and items. Assume that the user set $U = \{u_1, u_2, u_3, \ldots, u_M\}$, item set $I = \{i_1, i_2, i_3, \ldots, N\}$, M is the total number of users, N is the number of items. The raw Rating matrix $Rating_{M \times N}$ represents the actual rating given to the item by the user. Given that user $u \in U$ and item $i \in I$, if user u does not score item i, we need to extract the characteristic information of the potential relationship between user and item according to the existing scoring information in the actual scoring matrix $R_{M \times N}$, and then restore the new scoring matrix by means of matrix inner product to predict the user's score on the item.

As for the general formula of recommendation algorithm, Huang et al. [17] gave the following definition:

$$\forall u \in U, i_u^{\prime} = \underset{i \in I}{argmax} S(u, i) \tag{1}$$

In formula (1), U is a set of users, and u represents a certain user; I indicates a set of items, and i indicates an item. $S(\cdot)$ represents the recommendation algorithm, and $i_u\prime$' represents the item i of interest to user u.

2.2 ARMF Model

In order to build a higher-order nonlinear relationship between users and item potential relationship features and store the previous layer of user or item potential relationship features, this paper proposes an associated residual matrix decomposition model, the framework of which is shown in Fig. 1.

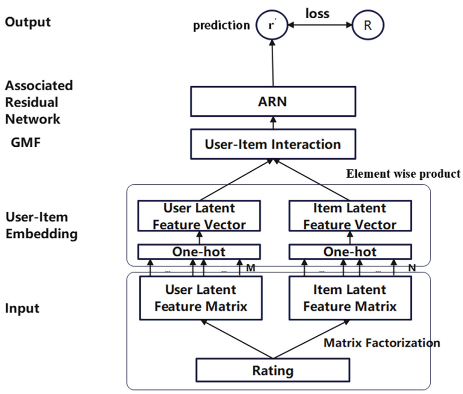

Fig. 1. ARMF Model

The framework is mainly composed of 5 parts: Input (Input Layer), user-item Embedding (User, item embedding Layer), GMF, Associated Residual Network (ARN Layer), Output (Output Layer). The Input layer decomposes the original score matrix of the data set through matrix decomposition technology, so as to obtain the feature matrix of the potential relationship between users and items. Then on the user-item Embedding layer, the one-hot coding technology is used to extract the Latent Feature Vector of each User or item from the latent feature matrix of the User and the Item respectively, that is, the User Latent Feature Vector. Item Latent Feature Vector. In the GMF layer, each user interacts fully with the underlying characteristics of the item. The principle is: the product of the corresponding elements of the user's potential feature vector and the item's potential feature vector. The Associated Residual Network layer uses the improved residual neural network to iteratively train the potential relationship between the user and the item, thereby establishing the higher-order nonlinear relationship between the user and the item and preventing gradient explosion or gradient disappearance in the model. Finally, a user's predicted rating of the item is obtained in the output layer. Each layer of the model framework is detailed below.

(1) Input layer. The raw rating matrix of the user's rating of the item. A matrix of rows M columns N is obtained by preprocessing the scores of the users in the data set. Where M represents the number of users and N represents the number of items. The potential feature matrix of users and items is obtained by matrix decomposition technique. The calculation formula (2) and (3) is as follows:

$$User\_input_{M \times K} = MF(\text{Rating}) \tag{2}$$

$$Item\_input_{N \times K} = MF(\text{Rating}) \quad (3)$$

Where, MF stands for Matrix Factorization; $User\_input_{M \times K}$ and $Item\_input_{N \times K}$ represent user and item potential feature matrices, respectively. Rating represents the original rating matrix.

(2) user-item Embedding (Embedding Layer). In order to get the feature vector implied by each user and item, one-hot coding technique is used in this layer. The potential feature vectors $user\_vector_i$(i=1 ~ M), $item\_vector_j$(j=1 ~ N)for each user and item are extracted from $User\_input_{M \times K}$ and $Item\_input_{N \times K}$ respectively. The calculation formula (4) and (5) are as follows:

$$user\_vector_i = User\_OneHot_i \cdot User\_input_{M \times K} \quad (4)$$

$$item\_vector_j = Item\_OneHot_j \cdot Item\_input_{N \times K} \quad (5)$$

Where, $user\_vector_i$ represents the potential eigenvector of the i th user; $item\_vector_j$ represents the potential eigenvector of the j th item; $User\_OneHot_i$ Indicates the one-Hot code of the i th user. $Item\_OneHot_j$ represents the one-Hot encoding of the j th item (as shown in Fig. 2).

Fig. 2. User and item one-hot encoding.

(3) GMF layer. In this paper, Generalized Matrix Factorization (GMF) is used to establish the interaction between users and the underlying features of the item. Multiply the potential eigenvector $user\_vector_i$ and $item\_vector_j$ on the user-item Embedding layer to obtain the K-dimensional vector, as shown in formula (6):

$$\text{UI}_{\text{GMF}} = \varphi\left(user\_vector_i, item\_vector_j\right) = user\_vector_i \odot item\_vector_j \quad (6)$$

Here, $\odot$ denotes the multiplication of the corresponding bits of $user\_vector_i$ and $item\_vector_j$.

(4) Associated Residual Network (ARN). This layer uses an improved residual neural network to establish higher-order nonlinear relationships between user and item potential relationship features. Deep neural networks have very good performance in information feature extraction, so they are often used in classification and regression tasks. However, as the number of layers of the neural network increases, there may be a problem of gradient disappearance or explosion due to gradient accumulation. The associated residual neural network in the model in this paper adds the feature information of the previous layer to each hidden layer, which is used to prevent the model from overfitting and gradient explosion or gradient disappearance. In addition, in order to ensure that the information features of each hidden layer are equal to those of the previous layer, the associated residual network integrates the output results of the previous layer with

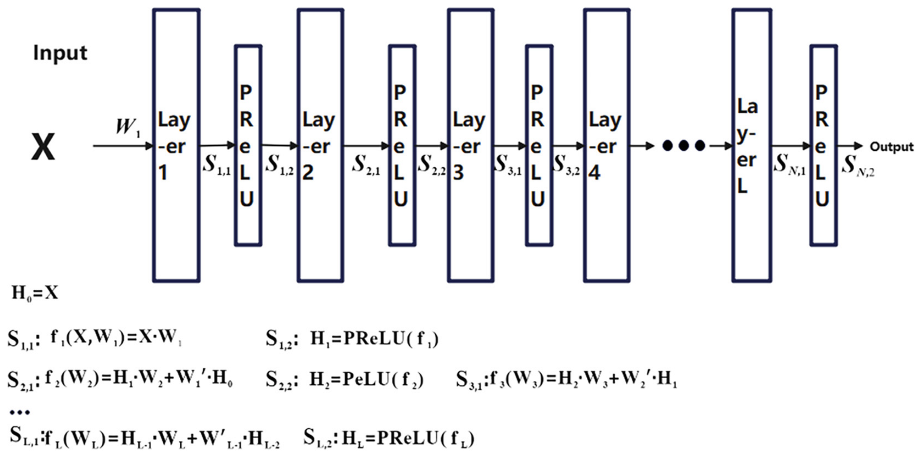

Fig. 3. ARes-Net

a biased weight to adjust the matching of the information features of the current layer and the previous layer. The structure is detailed in Fig. 3.

$$\begin{aligned}
&H_0 = X \\
&f_1(H_0, W_1) = H_0 \cdot W_1 \\
&\cdots \\
&f_L(H_{L-1}, W_L) = H_{L-1} \cdot W_L + W'_{L-1} \cdot H_{L-2} \\
&H_L = PReLu(L)
\end{aligned} \tag{7}$$

Formula (7) is the calculation process of forward propagation of the associated residual network layer. Where H_0 represents the initial input, which is $\mathrm{UI}_{\mathrm{GMF}}$. $f_L(H_{L-1}, W_L)$ (or f_L) represents the output of the L-hidden layer neurons, H_{L-1} represents the output result of the L-1 hidden layer through the activation function PReLU, and W_L represents the weight matrix of the L-1 hidden layer. $W_{L-1}\prime$ represents the bias matrix of the L-1 layer (the default value is 1), in order to regulate the number of neurons in the previous layer, and to store the information characteristics of the previous layer. This article uses PreLU [18] as the activation function for each hidden layer. Compared with other activation functions such as sigmod and ReLU, PReLU designs a learnable parameter for the negative semi-axis on the basis of ReLU, which makes the model have better fitting ability and reduces the risk of model overfitting.

(5) Output layer. This layer obtains the prediction result r' of the user for the item. The calculation is as in Eq. (8):

$$r\prime = ARN(\mathrm{UI}_{\mathrm{GMF}}) = H_L \tag{8}$$

H_L is the output result of ARN layer. $\mathrm{UI}_{\mathrm{GMF}}$ is the output result of the GMF layer, which is the input of the ARN layer.

In order to optimize the ARMF model, this paper uses the mean square error loss function with added regularization to train the model. The calculation is given in Eq. (9)

$$Loss = \frac{1}{Num} \sum_{i=1}^{Num} (R - r)^2 \tag{9}$$

Where Num represents the actual rating of the item by the user; R is the user's actual rating of the item.

3 Experimental

In this paper, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are used to verify the feasibility and effectiveness of the model on two public datasets in the movie field, Movielens100 K and Movielens1M.

3.1 The Dataset

The Movielens dataset collects the ratings of users to movies in a wide range of movie fields, mainly including user and movie ids, movie attributes, user ratings to movies, etc. In this paper, the public datasets Movielens100K and Movielens1M are used to train the model and analyze the model performance.

- The Movielens100K dataset contains 100,000 ratings (1–5) from 943 users for 1,682 items. Each user rated at least 20 movies.
- The Movielens10M dataset contains 10,000209 ratings (ratings 1–5) from 6,040 users for 3,952 items. Each user rated at least 20 movies.

3.2 Evaluation Indicators

Among many offline metrics of recommender systems, Root Mean Square Error (MSE) and Mean Absolute error (RMAE) are two commonly used evaluation metrics. Both are measures of the difference between the predicted value and the actual value, and they are used to assess how well the model fits the given data. In this paper, the above two evaluation metrics are used to evaluate the performance of the ARMF model. The calculation is given in Eqs. (10) and (11):

$$RMSE = \sqrt{\frac{1}{T} \sum_{i=1}^{T} (R - r)^2} \tag{10}$$

$$MAE = \frac{1}{T} \sum_{i=1}^{T} |R - r| \tag{11}$$

Where T resents the user's rating of the item in the test set; R is the actual rating the user gave the item. r denotes the score predicted by the model.

3.3 Comparison Experiment and Parameter Setting

In this paper, the scoring matrix in the data set is divided into the training set and the test set according to the ratio of 8:2, and the evaluation indicators RMSE and MAE are used to evaluate the performance of the model. In order to verify the accuracy of the recommendation, this paper selects SVD + +, NMF and NCF as comparison experiments.

The ARMF model in this paper only needs to input the user's rating matrix for the item, and no other information features will be introduced. Experimental parameter setting: Adam optimizer was used as the model trainer. Three hidden layers (number of neurons: 100, 200, 10) were selected for the correlation residual network. The bias weight value $W\prime$ in the residual network is 1. The activation function is PReLU. BatchNorm was added between the output of each layer of the network and the activation function to normalize the data. The normalization method was used to adjust the data gap of the actual scoring matrix and the model prediction input. The learning rate lr for model training is set to 0.1.

3.4 Analysis of Experimental Results

In order to verify the influence of the bias weight $W\prime$ in the associated residual network on the performance of the model. In the experiment, $W\prime$ pairs were adjusted on the dataset Movielens100K. When $W\prime$ is less than or equal to 0.01, RMSE is 0.2413 and MAE is 0.1985. When $W\prime$ is between 0.01 and 1, RMSE and MAE increase. When $W\prime$ is 1, RMSE is 0.7659 and MAE is 0.7195. $W\prime$ was fixed as 1, and the values of RMSE and MAE fluctuated around 0.1 after multiple experiments. Table 1 shows the relationship between the value of $W\prime$ and the performance of the model.

Table 1. Values of $W\prime$

$W\prime$	RMSE	MAE
0.01	0.2413	0.1985
1	0.7659	0.7195

In this paper, SVD + +, NMF [19], NCF and Constrained Bayesian Probabilistic Matrix Factorization (CBPMF) [20] algorithms are used to conduct comparative experiments on the data sets Movielens100K and Movielens1M respectively. The experimental results are shown in Table 2. It is known from Table 2 that the improved method in this paper is significantly better than other algorithms on the dataset movielens100K. In particular, the RMSE of movielens100K is improved by at least 17% compared with other algorithms.

Table 2. Results of each method on different datasets

Methods	Movielens100K		Movielens1M	
	RMSE	MAE	RMSE	MAE
SVD + +	0.9146	0.6617	0.8720	0.6980
NMF	1.0136	4.1686	0.9321	1.2163
NCF	0.9234	0.7265	0.9242	0.7310
CBPMF	0.8530	0.8033	0.8533	0.8087
ARMF	0.7659	0.7195	0.9121	0.7012

4 Conclusion

Matrix factorization techniques applied to recommender systems can simply extract the implicit relationship features between users and items from the given user-item explicit relationship matrix (rating matrix). These implicit relationship features become important indirect conditions for mining users 'preferences. However, in order to solve the cold start and sparsity problems of the recommendation system, the simple inner product method of the implicit relationship feature matrix (latent relationship matrix) between users and items cannot construct the high-order interaction relationship between users and items. The application of deep learning technology can solve the problem of high-order interaction. Neural networks, for example, extract deep features by mapping raw input data into high-dimensional or low-dimensional feature Spaces. Finally, nonlinear activation functions are used to improve the expression ability of the network.

However, with the increase of the number of layers, the neural network may have the problem of gradient disappearance or gradient explosion during backpropagation training, which leads to the degradation of model performance. To solve this problem, the residual network adds a residual part to the original direct feature mapping, so that the model always retains a residual term (residual part) in the process of direction propagation, so as to prevent the gradient from exploding or disappearing.

In this paper, an adjustment bias term is designed according to the idea of dimensionality reduction of the number of neurons in the traditional residual network. The bias term adjusts the number of neurons in the current layer and the previous layer by inner product with the output of the previous layer, so as to achieve the purpose of the residual.

However, this paper only considers the potential relationship characteristics between users and items, and does not add more diverse feature information, such as the relationship between users, the relationship between items, the dynamic rating of users and so on. These will be important directions that can be studied in the future.

References

1. Marz, N., Warren, J.: Big Data: Principles and Best Practices of Scalable Real-Time Data Systems. Manning Publications Co., New York (2015)

2. Qingwen, L.: Research on Recommendation Algorithm Based on Collaborative Filtering. University of Science and Technology of China, Hefei (2013)
3. Zhou, H.H.,Liuy, J.,Zhang, W.Q.,et al.: A survey of recommender system applied in E-commerce. Appl. Res. Comput. **21** (1), 8–12 (2004)
4. Huang, L.W.,Fu, M.S.,Li, F., et al.: A deep reinforcement learning based long-term recommender system. Knowl.-Based Syst. **13**, 106706 (2021)
5. Cui, Z., et al.: Personalized recommendation system based on collaborative filtering for IoT scenarios. IEEE Trans. Serv. Comput. **13**(4), 685–695, 1 July–August 2020. https://doi.org/10.1109/TSC.2020.2964552
6. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009). https://doi.org/10.1109/MC.2009.263
7. He, X.N., Zhang, H.W., Kan, M.Y., Chua, T.S.: Fast matrix factorization for online recommendation with implicit feedback. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 549−558. ACM, New York (2016). https://doi.org/10.1145/2911451.2911489]
8. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426−434. ACM, New York (2008). https://doi.org/10.1145/1401890.1401944]
9. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Incremental singular value decomposition algorithms for highly scalable recommender systems. In: Proceedings of the 5th International Conference on Computer and Information Science, pp. 27−28 (2002)
10. Pan, R., et al.: One-class collaborative filtering. In: Proceedings of the 2008 8th IEEE International Conference on Data Mining, pp. 502−511. IEEE, Piscataway (2008). https://doi.org/10.1109/ICDM.2008.16
11. Jianjun, L., Jianjun, L., Jianjun, L., et al.: Social recommendation algorithm based on trust and distrust information. J. Inf. Syst. **32**(4), 1–38 (2014)
12. Li Fei, X., Fei, C.F.: Coupling Term Matrix Decomposition, pp. 1–14. Springer, Cham (2014)
13. Tao, H., Jianjun, Z.: Asymmetric user similarity recommendation system based on matrix decomposition. Knowl.-Based Syst. **83**, 51–57 (2015)
14. He, X., Liao, L., Zhang, H., et al.: Neural collaborative filtering. In: International World Wide Web Conferences Steering Committee (2017). https://doi.org/10.1145/3038912.3052569
15. Zhen, T., Pan, L., Pu, Y., Rui, W.: Deep matrix factorization recommendation algorithm. J. Softw. **32**(12), 3917–3928 (2021). (in Chinese)
16. Liu, S., Deng, W.: Very deep convolutional neural network based image classification using small training sample size. In: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, Malaysia, pp. 730–734 (2015). https://doi.org/10.1109/ACPR.2015.7486599
17. Li-wei, H., Bi-tao, J., Shou-ye, L., et al.: Survey of recommender systems based on deep learning. Chin. J. Comput. **41**(7), 1619–1647 (2018). (in Chinese)
18. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, pp. 1026–1034 (2015). https://doi.org/10.1109/ICCV.2015.123
19. Lee, D.: .Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems, vol. 13 (2001)
20. Hao, Y., Ma, W., Wang, B.: Probabilistic matrix factorization algorithm based on specific user constraints. J. Nanjing University (Natural Sciences), **57**(5), 818–827 (2021)