Attention neural collaboration filtering based on GRU for recommender systems

Hongbin Xia^{1,2} · Yang Luo¹ · Yuan Liu^{1,2}

Received: 23 August 2020 / Accepted: 13 January 2021 / Published online: 30 January 2021
© The Author(s) 2021

Abstract

The collaborative filtering method is widely used in the traditional recommendation system. The collaborative filtering method based on matrix factorization treats the user's preference for the item as a linear combination of the user and the item latent vectors, and cannot learn a deeper feature representation. In addition, the cold start and data sparsity remain major problems for collaborative filtering. To tackle these problems, some scholars have proposed to use deep neural network to extract text information, but did not consider the impact of long-distance dependent information and key information on their models. In this paper, we propose a neural collaborative filtering recommender method that integrates user and item auxiliary information. This method fully integrates user-item rating information, user assistance information and item text assistance information for feature extraction. First, Stacked Denoising Auto Encoder is used to extract user features, and Gated Recurrent Unit with auxiliary information is used to extract items' latent vectors, respectively. The attention mechanism is used to learn key information when extracting text features. Second, the latent vectors learned by deep learning techniques are used in multi-layer nonlinear networks to learn more abstract and deeper feature representations to predict user preferences. According to the verification results on the MovieLens data set, the proposed model outperforms other traditional approaches and deep learning models making it state of the art.

Keywords Stacked denoising autoencoder · Gated recurrent unit · Attention mechanism · Collaborative filtering · Auxiliary information

Introduction

Nowadays, it is difficult to obtain useful information when the information available to users is exploding. To solve the problem of information overload, a good recommendation algorithm is very important. The recommendation system implements personalized services by mining users' historical behavior, text information, etc. for modeling. Traditional recommendation methods are mainly divided into three types: one is the recommendation method based on collaborative filtering (CF) [1, 2], and it can be divided into neighborhood-based CF [3] and model-based CF [4]. The

second is content-based recommendation method [5]. The third is the hybrid recommendation method [6]. Matrix factorization (MF) [7] is widely used in traditional collaborative filtering recommendation method. The idea of MF is to project the latent feature vectors of users and items into a shared latent space, and use the form of inner product to represent the user's interaction with the items, and to complete the scoring matrix. Mnih et al. [8] proposed a probability matrix factorization (PMF) to improve the recommendation performance. Although MF shows better results to a certain extent, there are still historical problems: data sparseness and cold start. Some researchers [9] are committed to integrating user and item auxiliary information in CF to generate more effective features, but this also has limitations and cannot extract deep-level latent features. Therefore, deep learning has received extensive attention from researchers.

In recent years, many scholars have proposed advanced algorithms to solve complex problems, including intelligent calculation methods [10] and deep learning algorithms. The combination of traditional recommendation algorithms and

✉ Hongbin Xia
hbxia@jiangnan.edu.cn

¹ School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, Jiangsu, People's Republic of China

² Jiangsu Key Laboratory of Media Design and Software Technology, Wuxi, Jiangsu, People's Republic of China

deep learning has been well received by researchers and has also received good results. Zhang et al. [11] introduced the research status of combining deep learning with traditional recommendation systems. He et al. [12] proposed the neural collaborative filtering (NCF). This model uses deep nonlinear structures instead of the inner product of traditional feature vectors to learn the latent vectors of users and items, but only considers the score data of users and items, and does not take into account the use of auxiliary information. Wang et al. [13] proposed a collaborative deep learning (CDL) method, which combines deep learning and CF for the ratings. Since deep learning has strong feature extraction capabilities, some researchers are beginning to use the combination of auxiliary information and deep learning to generate effective feature representations. For example, Dong et al. [14] applied additional Stacked Denoising Autoencoder (aSDAE) to extract users' auxiliary information. And it is good at extracting latent vectors without text information. Kim et al. [15] used convolutional matrix factorization (ConvMF) to extract items' feature. Pal et al. [16] applied long–short-term memory (LSTM) for text feature extraction, which was used to focus on the entire text information. Bansal et al. [17] used Gated Recurrent Unit (GRU) model to extract the latent features of the item text to improve the performance of collaborative filtering, which also has a significant effect on the cold start problem. With further deep learning research, an important breakthrough is to introduce the attention mechanism into deep learning, thereby emphasizing local key information. Yin et al. [18, 19] used the attention mechanism in the recommendation system to learn the user's recent interests from the user's short-term interaction records. Guo et al. [20] used the attention machine to make feature extraction for users and items. Liu et al. [21] combined aSDAE and convolutional neural networks (CNN) for feature extraction, and showed good results, but at the same time, ignored the establishment of long-term dependencies and the role of key information.

The above models all obtain in-depth expressions of users and items through deep learning, but these models are only based on the idea of matrix factorization. Although Yu et al. [22] proposed a deep hybrid recommendation system based on Auto-encoder (DHA-RS), which performed nonlinear interactive learning after modeling users and items, they did not consider establishing long-term dependencies and the role of focusing on key information. Based on existing research, this paper proposes an effective recommendation model (GRU_Attention Neural collaborative filtering, GANCF), which mainly combines aSDAE model, auxiliary information, GRU and a multi-layer deep neural network to build a hybrid recommendation model. The core idea of GANCF: first, aSDAE model combined with scoring data and the user's auxiliary information are used to model the user's latent vector. Second, GRU uses the item's auxiliary

information to model the latent vector of the item, which uses the attention mechanism to learning the weight information of the key words of the text. Finally, the latent vectors of the user and the item are used as the input of the deep neural network to perform multi-level feature learning to obtain the hidden features between the user and the item. The deep feature representation is used to optimize feature vectors to improve recommendation efficiency. Experimental results on the two datasets of Movielens show that the recommended performance of this model is better than the traditional recommended performance.

The main contributions of this paper are summarized as follows:

- We use attention mechanism to enhance the feature extractive ability of GRU. Attention mechanism learns the weights of words from context feature and selects informative words according to the weights of these words before words are fed into the next layer. By combining GRU with item auxiliary information, we propose the GANCF model, which can take into account the order and context of words simultaneously so that it can effectively extract the features of the documents and these features can be used for rating prediction as well.
- We use the user's and item's auxiliary information to alleviate data sparsity and cold start problems, and use multi-layer neural networks to learn user and item interaction information, and also learn deeper nonlinear interaction features.
- The structure of this paper is as follows: The first section is the introduction to the recommendation system. The second section introduces the related works of this article. The third section introduces the model GANCF in this paper. The fourth section introduces the experimental preparation, such as data set and evaluation metric, etc. The fifth section is the experimental results and analysis. The sixth section is a summary.

Related works

Related works about matrix factorization, attention mechanism, neural collaborative filtering and gated recurrent unit are introduced briefly in this section.

Problem definition

This paper takes explicit feedback as training and testing data to complete the recommendation task. We have n users, m items, and an extremely sparse rating matrix $R \in R_{n \times m}$. Each entry R_{ui} of R corresponds to user u 's rating on item i . Likewise, the auxiliary information matrix of users and items are denoted by X and Y , respectively. Let $p_u, q_i \in R$ be

user u 's latent factor vector and item i 's latent factor, respectively, where k is the dimensionality of the latent space. Therefore, the corresponding matrix forms of latent factors for users and items are $P = p_{[1:n]}$ and $Q = q_{[1:m]}$, respectively. Given the sparse rating matrix R and the side information matrix X as well as Y , our goal is to learn effective users' latent factors P and items' latent factors Q , and then to predict the missing ratings in R .

Matrix factorization

The most widely used method in the recommendation system is collaborative filtering, which can be divided into memory-based collaborative filtering and model-based collaborative filtering. Model-based collaborative filtering has attracted more attention from researchers. Matrix factorization has been widely used in personalized recommendation systems. Li et al. [23] proposed Bayesian probability matrix factorization, and Cheng et al. [24] proposed non-negative matrix factorization. These are all matrix factorization models.

The core idea of MF is to decompose a sparse scoring matrix into two-dimensional low-rank matrices, one representing the latent vector of user features $P_{n \times k}$ and the other representing the latent vector of item features $Q_{k \times m}$, and then projecting into a common hidden factor space to fill the scoring matrix. Suppose that the user's rating matrix for items is $R_{n \times m}$, including n users and m items.

The objective function of matrix factorization is defined as:

$$\arg \min_{P, Q} L(R, PQ^T) + \lambda (\|P\|_F^2 + \|Q\|_F^2), \quad (1)$$

in which R is the real score, L is loss function that is used to evaluate the difference between the true score and the experimental predicted score. In addition, $\|P\|_F$ and $\|Q\|_F$ denote the Frobenius norm of the matrix, and λ is regularization parameters that is usually set to alleviate model overfitting.

Neural collaborative filtering

NCF [11] proved that MF has certain limitations. Using simple fixed inner products in low-dimensional spaces to estimate complex user interactions with items can cause certain problems. MF only uses information in two dimensions, $userId$ and $itemId$, and it is difficult to integrate more useful features during the learning process, such as user preferences and item feature information. And MF can only perform simple linear inner products, and cannot perform complex nonlinear interactions. Therefore, consider using a deep neural network in the recommendation system, that is, using a multi-layer perceptron to learn the nonlinear interaction

expression between the user and the item, so as to solve the restrictive problem of matrix factorization.

Set user set U and item set V . Above the input layer is an embedding layer, which projects the sparse representations of users and items onto a dense fully connected layer to serve as the user's latent vector $P = \{p_1, p_2, \dots, p_n\}$ and the item's latent vector $Q = \{q_1, q_2, \dots, q_m\}$, respectively. Then, connecting P and Q in a specific way as the input of multiple hidden layers to learn the potential representation of user-item interaction. Finally, the hidden vector is mapped to the prediction score. Both MF and multi-layer perceptron can be interpreted using the NCF model. When the learned user and item features are internally produced, the MF model can be obtained. The model is shown in Fig. 1.

The latent vector of user u is P_u , the latent vector of item i is q_i , and the first-layer network mapping function is as follows:

$$\Phi_1(p_u, q_i) = p_u \Delta q_i, \quad (2)$$

where Δ denotes the connection mode of P_u and q_i , or the two are correspondingly multiplied, or the two are directly connected.

If Δ denotes correspondingly multiplied, the network mapped to the output layer after learning is as follows:

$$\hat{y}_{ui} = \alpha_{out}(h^T(p_u, q_i)), \quad (3)$$

in which $\hat{y}_{ui}$ denotes the element-wise product of vectors, $\hat{y}_{ui}$ denotes prediction score, h^T and α_{out} denote the weights the weight and activation function of the output layer, respectively.

Fig. 1 NCF model

if Δ denotes directly connected, the network mapped to the output layer after learning is as follows:

Fig. 2 GANCF model

$$\hat{y}_{ui} = f(P^T U_u, Q^T V_i | P, Q, \Theta_f) \quad (4)$$

$$f(P^T U_u, Q^T V_i) = \phi_{\text{out}}(\phi_X(\dots \phi_2(\phi_1(P^T U_u, Q^T V_i)))) \quad (5)$$

in which Θ_f denotes the model parameter of the interaction function f , ϕ_{out} denotes the mapping function of the model output layer, and ϕ_X denotes the mapping function of the X th hidden layer.

Gated recurrent unit

GRU can better capture the dependence of large time distance in time series. As a variant of LSTM, GRU combines the forget gate and the input gate into a single update gate. The cell state and the hidden state are also mixed, and other changes are added. The final model is simpler than the standard LSTM model. In the GRU network, there is no division of the internal state and the external state in the LSTM network, but directly by adding a linear dependency between the current network state h_t and the previous state of the network state h_{t-1} . The purpose is to solve the problem of gradient disappearance and gradient explosion.

1. *Update gate* It is used to control the degree to which the historical state information h_{t-1} at the previous moment and current input state x_t are brought into the current state h_t . The calculation process of the update gate is as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \quad (6)$$

in which x_t is the input state at time t , and U_z , W_z denote the weight of update gate, b_z denotes the bias of update gate. σ is sigmoid function and the output is between $[0, 1]$.

(2) *Reset gate* decide whether the candidate state $\tilde{h}_t$ at the current moment depends on the network state h_{t-1} at the

previous moment and how much it depends on. The calculation process of the reset gate is as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \quad (7)$$

in which U_r , W_r denote the weight of reset gate, b_r denotes the bias of reset gate.

The output z_t of the update gate is linearly multiplied with the historical state topic h_{t-1} and the candidate state $\tilde{h}_t$ to jointly determine the output h_t of the GRU. The calculation process is as follows:

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \quad (8)$$

$$h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t, \quad (9)$$

in which $*$ denotes multiplication by element.

GRU can not only realize the function of LSTM, but also has a more brief network structure and fewer parameters.

GANCF model

The structure of the GANCF model is shown in Fig. 2. The model consists of three parts: the left part is the latent feature matrix of the auxiliary information of the attention-GRU modeling item Q . The right part is the additional SDAE model based on the user auxiliary information to model the user's latent feature matrix P . The middle part uses the latent features of user and item as the input of the multi-layer neural collaborative filtering model to learn the nonlinear interaction characteristics of user and item, and finally makes the score prediction. X and Y denote user and item auxiliary information, respectively. R is user-item rating matrix, S is user rating information, and W^+ and W denote

Fig. 3 Overall framework **Fig. 4** aSDAE model

weight parameters of aSDAE and GRU-Attention, respectively. K is the latent vector dimension.

The overall flow diagram of the GANCF model is shown in Fig. 3.

User feature extraction

This paper uses the method of the literature [13] for user feature extraction and integrates auxiliary information into user input to generate latent vectors P for users. There are n users, m items, user rating sample set $S = \{s_1, s_2, \dots, s_n\}$, user auxiliary information set $X = \{x_1, x_2, \dots, x_n\}$, $\tilde{S}$ and $\tilde{X}$ are the noise-added damaged versions of the original input S and X , respectively. Given a user-item scoring matrix R , the user-item rating matrix R is transformed into a m -sample rating set. The structure is shown in Fig. 4.

Fig. 5 GRU-Attention model

For each hidden layer $l \in \{1, 2, \dots, L-1\}$ of the aSDAE model, the hidden layer l is represented as h_l :

$$h_l = g(W_l h_{l-1} + V_l \tilde{x} + b_l) \quad (10)$$

$$h_0 = \tilde{s}$$

The output of layer L is expressed as:

$$\hat{S} = f(W_L h_L + b_s) \quad (11)$$

$$\hat{X} = f(V_L h_L + b_{\hat{x}}), \quad (12)$$

The output of the $L/2$ layer is the user's latent vector, and the latent vector output by each user u is:

$$p_u = \text{asdae}((W, V), (X_u, S_u)), \quad (13)$$

in which W and V are the weight parameters of each layer, b is the bias of each layer. $g()$ and $f()$ are nonlinear activation

function. User auxiliary information includes user age, gender, position, etc. The attribute values of user auxiliary information are spliced into a vector by one-hot encoder. To learn the weight and bias parameters, each layer uses the back propagation algorithm.

Item feature extraction

The attention gated recurrent unit (GRU-Attention) obtains latent vectors from the item's document auxiliary information. Among them, the attention layer is introduced after the GRU extracts the text features, and each word vector is assigned a corresponding probability weight to further extract the text features. The framework of GRU-Attention is shown in Fig. 5.

Input layer The original document is converted into word vectors through the glove model and used as the input for the next layer. The text information d of l sentences composed of

n words, that is, $d = \{s_1, s_2, \dots, s_l\}$, where the i -th sentence is expressed as $s_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$.

GRU layer: GRU is used to learn when and to what extent to update the hidden state. The output word vector of the previous layer is used as the input sequence of the GRU. The update gate is used to control how much historical state h_{t-1} is to be kept in the output state h_t at the current moment. The reset gate determines whether the candidate state $\tilde{h}_t$ at the current moment depends on the network state h_{t-1} at the previous moment and how much it depends on. The output of the update gate is multiplied with the historical state h_{t-1} and the candidate state $\tilde{h}_t$ at time t to determine the GRU output h_t . The calculation process of GRU is as follows:

$$\begin{aligned} z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\ r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\ \tilde{h}_t &= \tanh(W \cdot [r_t * h_{t-1}], x_t) \\ h_t &= z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t, \end{aligned} \quad (14)$$

in which $*$ denotes multiplication by element. The GRU unit updates the current state through the state at the previous moment and the new candidate state.

Attention layer The output of the GRU is used as an input to extract the feature information of important words, and it creates a context vector for each word, and then performs a weighted summation of the context vector and the word feature vector, which can be expressed as:

$$\begin{aligned} u_{it} &= \tanh(W_w h_{it} + b_w) \\ a_{it} &= \frac{\exp(u_{it}^T u_w)}{\sum_i \exp(u_{it}^T u_w)} \\ s_i &= \sum_i a_{it} h_{it}, \end{aligned} \quad (15)$$

in which W_w denotes a weight coefficient, b_w denotes bias, u_w denotes a randomly initialized attention mechanism matrix, and s_i denotes a feature vector.

The entire attention GRU network structure accepts the original document of the item as input and outputs the latent vector of each item, which is defined as follows:

$$q_j = \text{agru}(W, Y_j), \quad (16)$$

in which W denotes the weight and bias vector, Y_j and q_j represent the original document and latent vector of the item, respectively.

Parameter learning

aSDAE is used when extracting user latent vectors. First, we combine the user's auxiliary information X and the user rating matrix R as the original input, and then perform

noise reduction to form the corresponding damaged version, which is converted into a low-dimensional user latent feature vector matrix P through the encoding process, that is, the middle layer is for the user's latent feature vector matrix. The original data is reconstructed by decoding. For n users and m items, the scoring matrix R is transformed into a set of m samples of $S^u = \{s_1^u, s_2^u, \dots, s_m^u\}$, where $s_i^u = \{R_{i1}, R_{i2}, \dots, R_{in}\}$ is the n -dimensional vector of user i on all items.

By minimizing reconstruction errors, the optimization goal is as follows:

$$L_u = \min \alpha \sum_i (s_i^u - \hat{s}_i^u)^2 + (1 - \alpha) \sum_i (x_i^u - \hat{x}_i^u)^2 + \lambda(f), \quad (17)$$

in which $f = \|W_l\|_F^2 + \|V_l\|_F^2$, $\hat{s}_i^u$ and $\hat{x}_i^u$ are the output of s_i^u and x_i^u on the model, respectively. W_l and V_l are weights, λ is a regularization parameter, and α is a trade-off parameter.

Second, GRU extracts the item latent feature matrix Q , adds an attention layer after the GRU layer to learn key weight information, and then uses dropout to prevent the hidden layer unit from adapting, and at the same time, it gives certain restrictions on the weight parameters to reduce the overfitting. The stochastic gradient descent is used to update the weights for each sample. The objective function is as follows:

$$L_i = \sum_j (Y_j - \text{agru}(W_g, Y_j))^2 + \lambda \|W_g\|_F^2, \quad (18)$$

In which Y_j is the input of item j .

Third, the user's latent feature matrix P and the item's latent feature matrix Q are connected as the input of the deep neural network, and then multi-level nonlinear interaction is performed, and finally the prediction score is output. The calculation process of the multi-layer network structure is shown in formula (19).

$$\begin{aligned} x_1 &= \phi_1(p_u, q_i) = \begin{bmatrix} p_u \\ q_i \end{bmatrix} \\ \phi_1(x_2) &= \sigma_2(W_2^T x_1 + b_2) \\ &\dots \\ \phi_L(x_{L-1}) &= \sigma_L(W_L^T x_{L-1} + b_L) \\ \hat{R}_{ui} &= \sigma_{\text{out}}(h^T \phi_L(x_{L-1})), \end{aligned} \quad (19)$$

Table 1 MovieLens statistics

Dataset	Number of users	Number of items	Sparsity (%)
ML-100 K	943	1682	93.76
ML-1 M	6040	3706	95.53

Table 2 User and item attributes

Dataset	User auxiliary information	Item auxiliary information
ML-100 K	Tags	Genre/description
ML-1 M	Gender/age/occupation	Genre/description

in which P_u denotes the latent vector of user u , q_i denotes the latent vector of item i , R_{ui} is the real score, λ is the regularization parameter, σ and W are the activation functions and weights of the hidden layer, respectively. σ_{out} and h are the activation functions and weights of the output layer, respectively.

Finally, the user latent feature matrix and the item latent feature matrix are used as the input of the deep neural network, and then multi-level nonlinear interaction is performed, and then the prediction score is output. The objective function is as follow:

$$L_c = \arg \min L_R(R, PQ^T) + \lambda(\|P\|_F^2 + \|Q\|_F^2) \\ L_R(R, PQ^T) = \sum_{u,i} (R_{ui} - p_u q_i^T)^2 \quad (20)$$

In which the latent vector of user u is p_u , the latent vector of item i is q_i , R_{ui} is the real score, and λ is the regularization parameter.

The loss function of the model consists of three parts: reconstruction error of user feature extraction, item modeling error and model prediction error. The optimization objective function of the model is as follows:

$$L = L_c + \mu L_u + \psi L_i, \quad (21)$$

in which μ and ψ are the trade-off parameters of the objective function. For the optimization of the objective function, back propagation is used to train the parameters of the neural network.

Experiment setup

In this section, we evaluate the performance of our GANCF model with two real-word datasets from different domains and compare our GANCF model with three state-of-the-art algorithms.

We use Python language for experiments, and use Python libraries such as pandas for data preprocessing, and apply Tensorflow (GPU) as the framework for deep learning. In Windows 10 64-bit operating system, Pycharm 2017, Inter(R) Core(TM) i7-8700 k CPU @3.70 GHz, 16 GB memory, and python 3.5, Comparative experiment analysis is performed.

Datasets

Movielens public data sets are widely used in movie recommendation systems. In this paper, MovieLens-100 k (ML-100 k) and MovieLens-1 M (ML-1 m) with auxiliary information are selected as experimental data sets, with a score range of 1–5 points. ML-100 k includes more than 100,000 pieces of rating data on 1682 items from 943 users. ML-1 m includes more than 1 million pieces of rating data for 3706 items by 6040 users, where the movie data rated by each user is greater than 20. User auxiliary information includes attributes such as age, occupation, and gender, which are converted into binary information. Item auxiliary information includes information such as movie description and movie genre. Tables 1 and 2 summarize the characteristics of the MovieLens datasets we used in the experiment. Use glove to convert its text information into word vectors. We split each dataset into the training set, validation set and test set at the ratio of 8:1:1, and each user (or item) in training set is ensured to have one rating at least.

Evaluation metric

Dacoudi [25] pointed out that there are three types of commonly used evaluation indicators in the recommendation system. Among them, to quantitatively evaluate our GANCF model, the recall rate (recall) is used to measure top-n recommendations, and the root mean square error (RMSE) is used as the evaluation index of algorithm accuracy [26], which is defined as:

$$\text{recall}@N = \frac{\sum_{u \in U} |R(u) \cap T(u)|@N}{\sum_{u \in U} |T(u)|}, \quad (22)$$

in which U denotes a set of users, N denotes the number of top N items recommended to users, $R(u)$ denotes a list of items recommended to user u , and $T(u)$ denotes a list of items watched by users.

The root-mean-squared error on test dataset is given by:

$$\text{RMSE} = \sqrt{\frac{\sum_{i,j \in T} (R_{ij} - \hat{R}_{ij})^2}{T}}, \quad (23)$$

in which T is the total number of ratings in the test set, R_{ij} is the real rating of user i in the test set for item j , and $\hat{R}_{ij}$ is the experimentally predicted rating.

Implementation details

In the experiment, the maximum interaction value is set to 200, learning rate is set to 0.001. In the aSDAE part, the noise rate is set to 0.4, the number of hidden layers is 3, the activation function is the *Sigmoid* function, and the batch

Fig. 6 MovieLens-recall@N

size is 256; in the attention GRU part sets the embedding dimension of each word to 200, the maximum length of the project document is 300, the dropout is 0.2, the batch size is 256, and the activation function uses *Relu*; in the neural collaborative filtering interaction part, the latent vector of the user and the latent vector of the item are combined as the embedded layer of the network. We defined the latent vector dimension as the number of neurons in the last neural collaborative filtering layer of neural collaborative filtering, a three-layer hidden layer structure is used to analyze the number of different potential factors [8, 16, 32, 64]. If the number of potential factors is 16, the network structure is 64- > 32- > 16.

Baselines

To verify the performance of the proposed GANCF model, the comparison model is as follows:

1. *NCF model* [11] After extracting the features of users and items, it is integrated into a multi-layer perceptron for nonlinear interactive learning.
2. *PHD model* [19] PHD is a hybrid model, which uses aSDAE and CNN models to extract user and item features, respectively.
3. *DHA-RS model* [20] The model uses two SDAE to fuse auxiliary information to model users and items, and then the modeled features are fused into a neural collaborative filtering model for scoring prediction.
4. *ANCF model* [19] The model is based on the NCF model and then uses attention to further extract feature information.

Fig. 7 The effect of the number of iterations

5. *aSDAE model* [13] The model uses two stacked denoising autoencoders to extract user and item features, but does not consider the auxiliary information of the item.
6. *GANCF-a model* The model uses two aSDAE based on GANCF to extract user and item features.
7. *GANCF model* GANCF is the model proposed in this paper. This model uses GRU to extract project features, attention to extract key text information, and a multi-layer neural network to learn nonlinear interactive features.

Table 3 RMSE

Model	RMSE
NCF	0.87735
ANCF	0.85447
aSDAE	0.86283
PHD	0.85914
DHA-RS	0.85075
GANCF-a	0.84862
GANCF	0.84471

Experiments

In this section, we evaluate the performance of GANCF model on two datasets and analyze the experimental results.

1. Discuss the performance of different N values in evaluating the standard recall. Experiment on two different data sets with different sparsity and different algorithms in the same environment.

It can be seen from Fig. 6 that the change trend of the evaluation indicator recall on both data sets is the same, and the recall value of several algorithms gradually increases with the increase of the N value. It is found that the evaluation index value of the NCF model is the smallest, because the NCF model does not consider the influence of auxiliary information on the recommendation performance, which makes the recommendation result poor. The other three models that use auxiliary information are significantly better than the NCF model, which shows the importance of integrating auxiliary information in the recommendation system. And the GANCF model is obviously superior to the other three algorithms on the two data sets, which shows that the model proposed in this paper improves the recommendation performance to a certain extent. It can also be clearly seen that the algorithm performs better on dense data sets than on sparse data sets. The similarity between the GANCF model and the ANCF model is that both use deep neural networks to learn complex non-linear interactive information and use attention mechanisms to obtain key feature information. The difference between the two is that the ANCF model does not take the auxiliary information into consideration, and the GANCF model integrates multiple deep learning models to make up for the shortcomings of the single model and enhance the feature extraction ability of the GANCF model. Therefore, it can be seen from the experimental results that the GANCF model is better than the ANCF model. At the same time, the GANCF proposed in this paper is better than other algorithms on sparse data sets, which solves the problem of data sparseness to a certain extent.

2. Discuss the changes of the RMSE values of several model methods under the number of iterations. The experiment was conducted on the ML-1 M data set, and the results are shown below.

Fig. 8 The influence of latent factors on the GANCF model

Figure 7 shows the performance of the RMSE of the GANCF model, NCF, aSDAE, DHA-RS, ANCF, GANCF-a and PHD model under the ML-1 m data set under different iterations. First, it can be seen from Fig. 7 that the overall trend of the four models is that the RMSE value gradually decreases with the increase of the number of iterations, and finally has a stable trend. But it can also be seen that too many iterations will increase the value of RMSE. This is because too many iterations will cause the model overfitting and reduce the recommended performance. Second, it is found that recommendation algorithms fused with auxiliary information, such as GANCF, DHA-RS, PHD, aSDAE and GANCF-a, has better performance than the NCF model of the recommendation algorithm without auxiliary information, indicating that adding certain auxiliary information can improve the recommendation performance of the model. At the same time, the GANCF results proposed in the text are superior to the PHD model, which shows that after incorporating the auxiliary information, multi-level deep interaction is performed, and learning deep-level nonlinear information can make the recommendation result better. GANCF's RMSE is superior to the DHA-RS model, indicating that selecting effective deep learning techniques to extract feature information is conducive to improving recommendation performance. That is, GRU and attention can learn long-term dependencies and learning on the basis of making up for the shortcomings of CNN feature extraction Feature weight information to important words. Finally, the GANCF model proposed in this paper can achieve very good experimental results in the case of fewer iterations, the model converges faster, and can have better operating efficiency.

Table 3 shows the average RMSE in ten iterations.

As shown in Table 3, compared with the aSDAE model, the performance of PHD model is increased by 0.43%,

Table 4 Parameter settings

Parameter	ML-1 m						
μ	1	1	1	3	3	3.5	3
ψ	150	300	500	150	250	250	500

Fig. 9 Influence of parameter value

indicating that proper integration of different models will improve the overall model performance. Compared with the GANCF-a model, the performance of DHA-RS model is reduced by 0.25%. The reason is that the GANCF-a model uses attention to obtain key feature information, realizes different attention to text information, and improves the performance of the model. Compared with the GANCF-a and ANCF models, the performance of GANCF model is increased by 0.46% and 1.16%, respectively. The former is mainly because the GANCF model integrates different deep learning technologies, while GANCF-a uses SDAE to extract features, which shows that the fusion of different models will produce different effects on the overall models. The latter is mainly because the ANCF model does not consider the influence of auxiliary information on the model. It can be seen that the GANCF model in this paper is better than other models, which also shows the effectiveness of our model in this paper.

3. Discuss the influence of the number of different latent factors on the model proposed in this paper under the condition of the number of iterations.

The experiment was conducted under the ML-1 m data set. To set an appropriate number of latent factors, a comparative experiment was conducted on the number of latent factors [8, 16, 32, 64]. The μ value is 1, the ψ value is 500, and the batch size is 256, and a three-layer hidden layer structure is used. The experimental results are shown in Fig. 8. It can be seen from Fig. 8 that the recommendation accuracy of the model can achieve better results when the

number of latent factors is 64, which shows that an appropriate increase in the number of latent factors can improve the recommendation performance. It can be seen that increasing the number of latent factors can improve the accuracy of the model, but if it exceeds a certain number, it will reduce the recommendation effect.

4. Discuss the influence of the trade-off parameters μ and ψ on the RMSE of the GANCF model. Experiments are conducted under the data set ML-1 m. The parameter setting results are shown in Table 4. The experimental results are shown in Fig. 9.

To verify the influence of parameter values on the GANCF model, the latent factor is set to 64 and the batch size is 256. The experimental results are shown in Fig. 9. On one hand, on the whole, the influence of different parameter values within a certain range on the model is not great, so the model has certain robustness. On the other hand, it can be seen from the figure that as the number of iterations increases, the value of RMSE gradually stabilizes. From a horizontal point of view, when the value of μ is small and the value of ψ gradually increases, the value of RMSE decreases, and the error decreases. From a longitudinal point of view, when the μ value increases, the model convergence speed is slow, and the model can achieve good results after 6–8 iterations. Therefore, the value of μ is appropriately reduced, and the value of ψ is increased, and this shows that appropriately increasing the weight of the feature vectors extracted by the GRU can improve the performance of the model.

Conclusions

In the information age, deep learning is widely used in many fields by many researchers. Compared with the traditional recommendation, the recommendation based on deep learning is to form a lower level feature to form a more abstract feature way to recommend, the purpose is to find out the effective representation of user and item data. In view of the sparse data and traditional recommendation, it is impossible to learn deep nonlinear interactive information. Based on deep learning, this paper fuses the auxiliary information of users and items to extract the latent feature vectors of them, and combines the latent vectors of users and items. The connection is used as the input of the deep network, and then a deeper non-linear

learning is performed to obtain better performance. The experimental results show that the GANCF model can get better results on the two data sets than others, which fully shows that considering auxiliary information and applying it to deeper learning can improve the recommendation performance of the model. But sometimes the interests of users change with the passage of time, so in future work, we will consider using the time prediction mechanism to add time factors to the model for research. In addition, consider integrating more multi-source heterogeneous data to build user and item models.

Author contributions The research results of this manuscript come from our joint collaborative research.

Funding This work was supported in part by the National Science and Technology Support Program of China (No. 61672264).

Availability of data and materials The data we use comes from public data sets: <https://grouplens.org/datasets/movielens/>. Movielens public data sets are widely used in movie recommendation systems. In this paper, MovieLens-100 k (ML-100 k) and MovieLens-1 M (ML-1 m) with auxiliary information are selected as experimental data sets, with a score range of 1–5 points. ML-100 k includes more than 100,000 pieces of rating data on 1682 items from 943 users. ML-1 m includes more than 1 million pieces of rating data for 3706 items by 6040 users, where the movie data rated by each user is greater than 20. User auxiliary information includes attributes such as age, occupation, and gender, which are converted into binary information.

Code availability The code of our paper is temporarily not available.

Compliance with ethical standards

Conflicts of interest/competing interests To the best of our knowledge, the named authors have no conflict of interest, financial or otherwise.

Ethics approval Ethics approval was not required for this research.

Consent to participate No one participated in the study of the manuscript.

Consent for publication Written informed consent for publication was obtained from all participants.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Elahi M, Ricci F, Rubens N (2014) A survey of active learning in collaborative filtering recommender systems. In: E-Commerce and Web Technologies—15th International Conference, EC-Web 2014, Munich, Germany, September 1–4, 2014. Proceedings, pp 29–50
2. Chen R, Hua Q, Chang Y, Wang B, Zhang L, Kong X (2018) A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access* 6:64301–64320
3. Kaleli C (2014) An entropy-based neighbor selection approach for collaborative filtering. *Knowl Based Syst* 56:273–280
4. Walter FE, Battiston S, Schweitzer F et al (2008) A model of a trust-based recommendation system on a social network. *Auton Agent Multi-Agent Syst* 16(1):57–74
5. Lops P, Jannach D, Musto C et al (2019) Trends in content-based recommendation: Preface to the special issue on Recommender systems based on rich item descriptions. *User Model User-Adap Inter* 29(2):239–249
6. Işık GTZ (2018) A hybrid movie recommendation system using graph-based approach. *Int J Comput Acad Res* 7(2):29–37
7. Li G, Yuchi J, Yang H et al (2019) A network delay factor model based on the hidden Markov Model and Latent Dirichlet Allocation. *IEEE Access* 7:133136–133144
8. R. Salakhutdinov, A. Mnih (2008) Probabilistic matrix factorization. In: Advances in neural information processing systems, pp 1257–1264
9. Shi Y, Larson M, Hanjalic A (2014) Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges. *ACM Comput Surv (CSUR)*. <https://doi.org/10.1145/2556270>
10. Wang Z, Ong Y, Sun J, Gupta A, Zhang Q (2019) A Generator for multiobjective test problems with difficult-to-approximate pareto front boundaries. *IEEE Trans Evol Comput* 23(4):556–571. <https://doi.org/10.1109/TEVC.2018.2872453>
11. Zhang S, Yao L, Sun A et al (2019) Deep learning based recommender system: A survey and new perspectives. *ACM Comput Surv (CSUR)*. <https://doi.org/10.1145/3285029>
12. He X N, Liao L Z, Zhang H W, et al (2017) Neural collaborative filtering. In: ACM, pp 173–182
13. H. Wang, N. Wang, and D.-Y. Yeung (2015) Collaborative deep learning for recommender systems. In: ACM SIGKDD. Discovery Data Mining (KDD), pp 1235–1244
14. Dong X, Yu L, Wu Z, Sun Y, et al (2017) A hybrid collaborative filtering model with deep structure for recommender systems. In: AACL, pp 1309–1315
15. Kim D, Park C, Oh J, et al (2016) Convolutional matrix factorization for document context-aware recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, pp 233–240
16. Zhang L, Wang S, Liu B (2018) Deep learning for sentiment analysis: A SURVEY. *Wiley Interdiscipl Rev Data Min Knowl Discov*. <https://doi.org/10.1002/widm.1253>
17. Bansal T, Belanger D, McCallum A (2016) Ask the GRU: Multi-Task Learning for Deep Text Recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems RecSys'16, pp 107–114
18. Yin W, S H, Xiang B, et al (2016) ABCNN: attention-based convolutional neural network for modeling sentence pairs. In: Transactions of the Association for computational linguistics, pp 259–272
19. Fu M, Qu H, Moges D et al (2018) Attention based collaborative filtering. *Neurocomputing* 311:88–89

20. Yanli G, Zhongmin Y (2020) Recommended system: attentive neural collaborative filtering. *IEEE Access* 99:125953–125960
21. Liu J, Wang D, Ding Y (2017) PHD: a probabilistic model of hybrid deep collaborative filtering for recommender systems *Asian Conference on Machine Learning*, pp 224–239
22. Yu L, Wang S, Shahrulkh KM, He JY (2018) A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering. *Big Data Min Anal* 1(3):211–221
23. Li J, Bioucas-Dias J M, Plaza A (2012) Collaborative nonnegative matrix factorization for remotely sensed hyperspectral unmixing. In: *Geoscience and Remote Sensing Symposium (IGARSS)*, pp 3078–3081
24. Cheng H T, Koc L, Harmsen J, et al (2016) Wide & deep learning for recommender systems. In: *Proceedings of the 1st workshop on deep learning for recommender systems*. ACM, pp 7–10
25. Davoudi A, Chatterjee M (2016) Modeling trust for rating prediction in recommender systems. In: *SIAM Workshop on machine learning methods for recommender systems 2016*
26. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.