IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 6, 2020, accepted October 27, 2020, date of publication November 18, 2020, date of current version November 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3038770

Collaborative Filtering Recommendation Algorithm Based on Attention GRU and Adversarial Learning

HONGBIN XIA[1,2], JING JING LI[1], AND YUAN LIU[1,2]
[1]School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China
[2]Jiangsu Key Laboratory of Media Design and Software Technology, Jiangnan University, Wuxi 214122, China

Corresponding author: Hongbin Xia (hbxia@jiangnan.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672264.

ABSTRACT Aiming at the problem that the traditional collaborative filtering algorithm using shallow models cannot learn the deep features of users and items, and the recommendation model is very susceptible to the counter-interference of its parameters; this paper proposes a matrix-factorization recommendation model that combines adversarial learning and attention-gated recurrent units (AGAMF). Firstly, the gated recurrent unit based on the attention mechanism is used to extract the user's latent vector from the user's auxiliary side information. Secondly, the convolutional neural network is used to extract the item's latent vector from the item's auxiliary side information. Finally, adversarial disturbances are introduced on the latent factors of users and items to quantify the loss of the model under parameter disturbances, and the latent vectors of users and items are integrated into the probability matrix factorization to predict the user's rating of the item. Experiments were performed on two real data sets MovieLens-1M and MovieLens-10M, and the RMSE, MAE and Recall indicators were used for evaluation. Experiments prove that the model proposed in this paper is robust and can effectively alleviate the problem of data sparsity. Compared with other related recommendation algorithms, our model has a significant improvement in recommendation performance.

INDEX TERMS Adversarial learning, attention mechanism, gated recurrent unit, convolutional neural network, probabilistic matrix factorization, collaborative filtering.

I. INTRODUCTION

In recent years, with the rapid development of Internet technology, the ways for users to obtain data have become more and more abundant. However, the explosive growth in the amount of information has brought about the problem of "information overload". Faced with noisy data, users may not be able to accurately select effective information. Therefore, the recommendation system is a necessary tool to help users obtain effective information. Traditional approaches include collaborative filtering methods [1]–[3], which use similar preferences among similar users to discover users' potential preferences for items, and are vulnerable to cold start problems and data sparsity problems. And content-based methods [4], [5], mining other items with similar attributes for recommendation based on user historical behaviors often encounters the problem of difficulty in feature extraction. Otherwise, hybrid recommendation methods [6], [7], considering that a single recommendation method has its own shortcomings, combine different recommendation algorithms for mixed recommendation.

Matrix factorization [8] is a widely used model-based CF method with good scalability and accuracy, the matrix factorization recommendation method has attracted more and more attention. The method expresses the user's rating information on the item in the form of a matrix, mines the low-dimensional latent space through the factorization operation of the matrix, and re-representing users and items in the low-dimensional space, and then expresses the correlation between users and items by the inner product of the latent feature vectors of users and items. In order to solve the sparsity problem of scoring data, Mnih proposed a probability matrix-factorization method [9]. Since deep learning has a powerful ability to learn the essential characteristics of data sets from

The associate editor coordinating the review of this manuscript and approving it for publication was Bohui Wang.

samples, more and more researches focused on combining traditional matrix factorization with deep learning models. Additional Stacked Denoising Autoencoder (aSDAE) [10] is good at extracting effective latent features from auxiliary side information and obtaining the implicit relationship between users and items. It extends the Stacked Denoising Autoencoder [11], takes additional auxiliary side information as input and integrates it closely with matrix factorization. Kim used Convolutional Neural Network (CNN) to capture the contextual information of item description documents to improve the accuracy of score prediction [12]. Considering that most current methods assume that there is a multi-linear interaction between latent factors, and small random disturbances of linear model parameters will lead to large backward errors. He [13] added adversarial perturbations to the each embedding vector of user and item in the matrix factorization can improve the robustness of the model. In recent years, neural networks based on the attention mechanism have been widely used in natural language processing. Zhou [14] proposed to use attention-based bidirectional long and short-term memory networks to capture key semantic information. Zhang [15] introduced the attention mechanism in the normalized matrix decomposition to analyze the user's different attention to item attributes to obtain more accurate user preferences. Wang [16] proposed a knowledge graph attention network to mine higher-order relationships (connecting two items with one or more link attributes).

Liu [17] proposed a probabilistic model that combined a stacked denoising autoencoder and a convolutional neural network, and showed good results. However, since the input document of the AutoEncoder contains many noise data without keywords, it is impossible to automatically distinguish keywords and capture sequence information, and also it is extremely vulnerable to be interfered by model parameters during model training. In response to these problems, based on Liu [17], a recommendation model (Adversarial GRU-Attention Matrix Factorization, AGAMF) is proposed by combining adversarial learning and GRU-Attention mechanism. Through the gated recurrent unit based on the attention mechanism and the convolutional neural network, adversarial perturbations are enforced on embedding factors of users and items to quantify the loss of the model under parameter perturbations. The latent vectors of users and items are integrated into the probability matrix factorization to predict user ratings. This work solves the problem of data sparsity and enhances the robustness of the model by optimizing feature vectors. The main contributions of our method are as follows:

1. Use GRU based on attention mechanism to enhance user feature extraction ability, obtain contextual semantic relationship of documents and highlight keyword information.
2. Adversarial perturbations are enforced on embedding factors of users and items to quantify the loss of the model under parameter disturbances. Stable the model fitting process and enhance the robustness of the model.
3. Integrating GRU-Attention and CNN into the PMF framework, and applying regularization parameters of users and items to balance the rating information and auxiliary side information, effectively alleviating the problem of data sparsity.

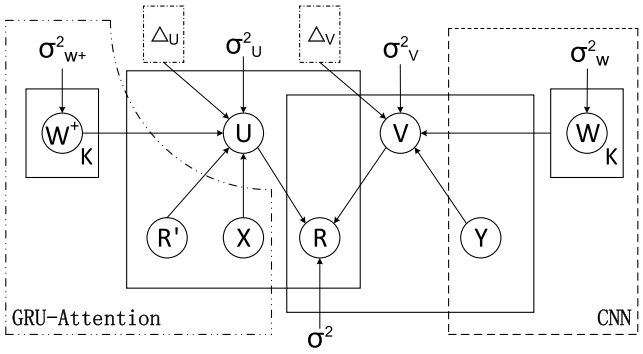


FIGURE 1. Overview of the AGAMF model.

II. ADVERSARIAL GRU-ATTENTION MATRIX FACTORIZATION MODEL

Figure 1 shows the overview of the probabilistic model for AGAMF, which integrates GRU-Attention and CNN into PMF, and the perturbations are enforced on each embedding vector of user and item.

In which R' represents the observed rating matrix, R represents the predicted rating matrix, X represents user auxiliary side information, such as user ID, gender, age, and occupation. Y represents item auxiliary side information, such as movie type and movie description. W and W^+ represent the weight of CNN and GRU-Attention. Δu and Δv respectively represent the adversarial perturbation enforced on the embedding vectors of users and items. K is dimension of the latent vector and σ^2 is the variance of the Gaussian normal distribution.

From a probabilistic point of view, the conditional distribution over predicted ratings can be given by:

$$p\left(R|\,U,V,\sigma^2\right)=\prod_{i}^{N}\prod_{j}^{M}N\left(R_{ij}|\,(u_i+\Delta u)^T(v_j+\Delta v),\sigma^2\right)^{I_{ij}} \tag{1}$$

In which $N(x|\,\mu,\sigma^2)$ is the probability density function of the Gaussian normal distribution with mean μ and variance σ^2.

A. MATRIX FACTORIZATION

Generally, MF model can learn latent factors of users and items in the user-item matrix, which are further used to predict new ratings between users and items. For clarity, we include the most common formulation of MF as follow:

$$\boldsymbol{L}=\sum_{i}^{N}\sum_{j}^{M}\boldsymbol{I_{ij}}(\boldsymbol{R_{ij}}-(\boldsymbol{u_i}+\boldsymbol{\Delta u})^{\boldsymbol{T}}(\boldsymbol{v_j}+\boldsymbol{\Delta v}))^2 +\boldsymbol{\lambda_U}\sum_{i}^{N}||U||_{\boldsymbol{F}}^{\boldsymbol{2}}+\boldsymbol{\lambda_V}\sum_{j}^{M}||\boldsymbol{V}||_{\boldsymbol{F}}^{\boldsymbol{2}} \tag{2}$$

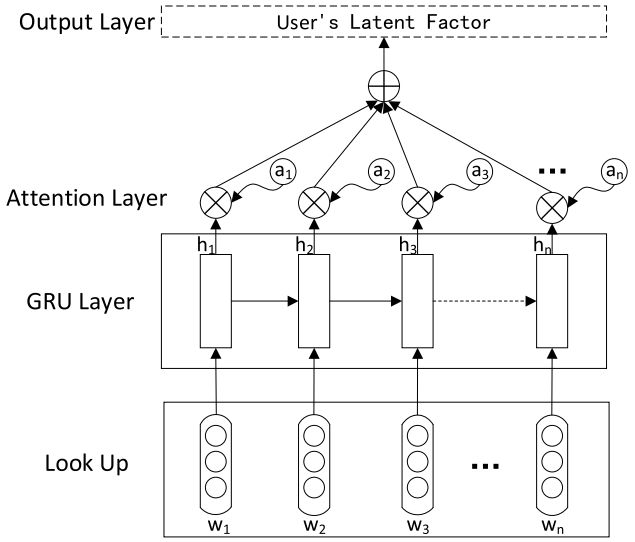


FIGURE 2. The gated recurrent unit based on the attention-mechanism architecture.

$\boldsymbol{\lambda_U}$ and $\boldsymbol{\lambda_V}$ are regularization parameters that are usually set to alleviate model overfitting, in which $\boldsymbol{I_{ij}}$ is an indicator function that is equal to 1 if $\boldsymbol{R_{ij}} > \boldsymbol{0}$, otherwise 0. In addition, $||\boldsymbol{U}||_{\boldsymbol{F}}$ and $||\boldsymbol{V}||_{\boldsymbol{F}}$ denote the Frobenius norm of the matrix.

B. GRU-ATTENTION

GRU has strong memory capabilities in time series, and it can learn the dependence of longer sequences of context without being limited to local features. Compared with LSTM, the network structure is simplified, the model parameters are less and the training rate is increased. The GRU consists of a reset gate, an update gate and a memory unit. We use GRU-Attention to obtain user latent vector U from the user's auxiliary side information, as shown in Figure 2.

Input Layer: The user's auxiliary side information is pre-trained using the Skip-Gram model in Word2Vec, then use Lookup to convert the words in the document into the corresponding pre-trained word vector $\{w_1, w_2, \ldots, w_n\}$ and use it as the input of the next layer.

GRU Layer: The sequence of the information input at $s$time is $w_1, w_2, , w_s$. The hidden layer output state h_s is obtained by updating $h_{s-1}$of GRU at $s-1$time. At different times, the hidden layer $w_1, w_2, , w_s$ corresponding to the GRU for each word output vector is $h_1, h_2, \ldots, h_s \in R_{n\_hid}$, n_hid is the number of neurons in the hidden layer of the GRU. h_s is input as a sentence feature vector to the next layer of the network. The feature extraction of text information is expressed as:

$$h_s = GRU\,(w_s)\,, \quad s \in [1, n] \tag{3}$$

Attention Layer: The attention mechanism of the relation classification task is used to capture the key semantic information in the sentence, and the word-level features at each moment are combined into a sentence feature vector, which can be expressed as:

$$e = tanh(w^T H) \tag{4}$$

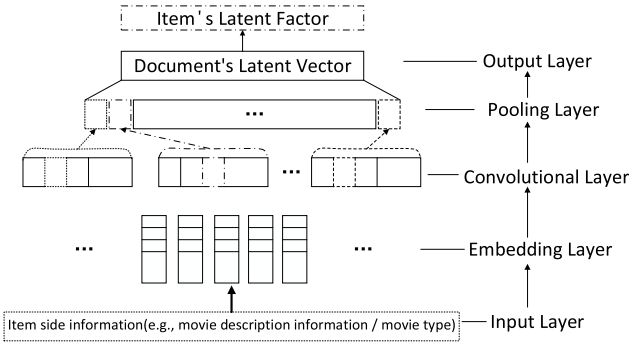


FIGURE 3. Convolutional neural network architecture.

$$\alpha = soft\ max(e) \tag{5}$$

$$\tau = H\alpha^T \tag{6}$$

H is a matrix composed of the output vector $[h_1, h_2, \ldots, h_s]$ of the GRU layer, where $H \in R^{d^w \times T}$, d^w is the dimension of the word vector, w is a trained parameter vector, and w^T is a transpose. The representation τ of the sentence is formed by a weighted sum of context vector and word feature vector.

The GRU network structure based on the attention mechanism accepts the user's original document as input and outputs the latent vector of each user, which is defined as follows:

$$u_i = agru(W^+, X_i) + \varepsilon_i \tag{7}$$

$$\varepsilon_i = N(0, \sigma_U^2 I) \tag{8}$$

ε_i is Gaussian noise, which is used to further optimize the user's latent vector.

For each weight parameter W_k^+ in W^+, the conditional distributions of W^+ and user latent vector U are:

$$\rho\left(W^+ \left| \sigma_{w^+}^2 \right.\right) = \prod_k^{|w_k^+|} N\left(w_k^+ | \, 0, \sigma_{w^+}^2\right) \tag{9}$$

$$\rho\left(U | \, W^+, X, \sigma_U^2\right) = \prod_i^N N\left(u_i | \, agru(W^+, X_i), \sigma_U^2\right) \tag{10}$$

C. CONVOLUTIONAL NEURAL NETWORK

The objective of our CNN architecture is to obtain documents' latent vectors from documents of items, which are used to compose the items' latent factors with epsilon variables. Figure 3 reveals our CNN architecture that contains five layers: 1) input layer, 2) embedding layer, 3) convolution layer, 4) pooling layer, 5) output layer.

Input Layer: Input information of the movie type and movie description.

Embedding Layer: Convert the original document into a number matrix according to the word length, the document matrix $D \in R^{p \times l}$ is as follows:

$$D = \begin{bmatrix} & | & & | & & | & \\ \cdots & w_{i-1} & & w_i & & w_{i+1} & \cdots \\ & | & & | & & | & \end{bmatrix}$$

In which, l represents the length of the document, p represents the embedding dimension of each word w_i.

Convolutional Layer: extract features of project text information. The contextual feature $c_i^j \in R$ is extracted by jth shared weight $W_c^j \in R^{p\times ws}$, whose window size ws determines the number of surrounding words.

$$c_i^j = f(W_c^j * D_{(:,i:(i+ws-1))} + b_c^j) \tag{11}$$

$*$ represents the convolution operator, $b_c^j \in R$ represents the bias of W_c^j, $f()$ represents the nonlinear activation function, and uses *relu* to avoid the problem of gradient disappearance. Then, a contextual feature vector $c_i^j \in R^{l-ws+1}$ of a document with W_c^j is constructed by:

$$c^j = [c_1^j, c_2^j, \cdots, c_i^j, \cdots c_{l-ws+1}^j] \tag{12}$$

A shared weight can only capture one type of context feature vector. Therefore, multiple shared weights are used to capture multiple types of context feature vectors to generate n_c context feature vectors with W_c (e.g., W_c^j where $j = 1, 2, \cdots n_c$).

Pooling Layer: Extract representative features from the convolutional layer, and process variable-length documents by constructing a pooling operation of fixed-length feature vectors.

$$d_f = [max\ (c^1), max\ (c^2), \ldots,\ max\ (c^j), \ldots, max\ (c^{n_c})] \tag{13}$$

In which, c^j is the context feature vector of $l-ws+1$ length extracted by the jth shared weight W_c^j.

Output Layer: We project d_f on the k-dimensional space of the project's latent factors, and generate the latent vector of the document by using conventional nonlinear projection.

$$s = tanh\ (W_{f2} tanh\ (W_{f1} d_f + b_{f1})\} + b_{f2}) \tag{14}$$

$W_{f1} \in R^{f\times n_c}$, $W_{f2} \in R^{k\times f}$ is the projection matrix, $b_{f1} \in R^f$, $b_{f2} \in R^k$ is the bias of W_{f1}, W_{f2}, where $s \in R^k$.

The CNN network structure accepts the original document of the project as its input and outputs the latent vector of each project, as follows:

$$v_j = cnn(W, Y_j) + \varepsilon_j \tag{15}$$

$$\varepsilon_j = N(0, \sigma_V^2 I) \tag{16}$$

ε_j is Gaussian noise, which is used to further optimize the items' latent vector.

For each weight parameter W_k in W, σ_w^2 is weight parameter variance, σ_V^2 is item latent vector variance, the conditional distributions of W and item latent vector V are:

$$\rho\left(W \left| \sigma_w^2\right.\right) = \prod_k^{|w_k|} N\left(w_k \right| 0, \sigma_w^2\left.\right) \tag{17}$$

$$\rho\left(V \left| W, Y, \sigma_V^2\right.\right) = \prod_j^M N\left(v_j \right| cnn(W, Y_j), \sigma_V^2\left.\right) \tag{18}$$

D. ADVERSARIAL LEARNING

The concept of robustness usually refers to the degree that an algorithm can resist the profile injection attack. However, few works have focused on the robustness of recommender system, which may fail to capture fine-grained and stable results due to noise data. Small random perturbations on the parameters of linear models can lead to large backward errors. Here we propose the AGAMF model, and this work is inspired by the recent developments of adversarial machine learning techniques [18]–[20]. Generally speaking, it was found that normal supervised training process makes a classier vulnerable to adversarial examples [21], which revealed the potential issue of an unstable model in generalization. Then researchers proposed adversarial training methods which augment the training process by dynamically generating adversarial examples to address the issue.

Building upon adversarial learning techniques [22]–[25], our approach injects adversarial perturbations to the model parameters based on neural networks in matrix factorization recommendations. Intuitively, the adversarial perturbations tend to attack model parameters, while the model parameters aim to defense against those perturbations for self-improvement. We formulate a unified objective function to take both adversarial perturbations and model parameters into account. As such, our method reaps the benefits of neural networks and matrix factorization, while enhancing the robustness of a recommender model, and thus improves its eventual performance.

III. AGAMF MODEL PARAMETERS OPTIMIZATION

This paper refers to the parameter optimization method of [17], and uses the maximum posterior estimation to optimize the parameters. The posterior probability of the parameters is:

$$\begin{aligned}&\max_{U,V,W^+,W} p\left(U, V, W^+, W | R, X, Y, \sigma^2, \sigma_U^2, \sigma_V^2, \sigma_{W^+}^2, \right.\\ &\left.\sigma_W^2, \Delta u, \Delta v\right)\\ &= \max_{U,V,W^+,W} [p\left(R|U, V, \sigma^2\right) p\left(U|W^+, X, \sigma_U^2, \Delta u\right)\\ &\times p(W^+|\sigma_{W^+}^2) p\left(V|W, Y, \sigma_V^2, \Delta v\right) p\left(W|\sigma_W^2\right)]\end{aligned} \tag{19}$$

The negative logarithm of (19) can be redefined as follows:

$$\begin{aligned}&L(U, V, W^+, W)\\ &= \sum_i^N \sum_j^M \frac{I_{ij}}{2}(R_{ij} - (u_i + \Delta u)^T (v_j + \Delta v))^2\\ &+ \frac{\lambda_U}{2} \sum_i^N \|(u_i + \Delta u) - agru(W^+, X_i)\|_F^2\\ &+ \frac{\lambda_V}{2} \sum_j^M \left\|(v_j + \Delta v) - cnn(W, Y_j)\right\|_F^2\\ &+ \frac{\lambda_{W^+}}{2} \sum_k^{|w_k^+|} \|w_k^+\|_2^2 + \frac{\lambda_W}{2} \sum_k^{|w_k|} \|w_k\|_2^2\end{aligned} \tag{20}$$

The coordinate ascending method is adopted to iteratively optimize the latent variables while fixing other variables, in which λ_U is σ^2/σ_U^2, λ_V is σ^2/σ_V^2, λ_{W^+} is $\sigma^2/\sigma_{W^+}^2$, λ_W is σ^2/σ_W^2. U and V are updated through the optimization function L until convergence, which is expressed as:

$$u_i \leftarrow (VI_iV^T + \lambda_U I_k)^{-1}(VR_i + \lambda_U agru(W^+, X_i)) \tag{21}$$

$$v_j \leftarrow (UI_jU^T + \lambda_V I_k)^{-1}(UR_j + \lambda_V cnn(W, Y_j)) \tag{22}$$

where λ_U and λ_V are regularization parameters that are usually set to alleviate model overfitting. I_i, I_j is a diagonal matrix of I_{ij} ($i = 1, 2, , N, j = 1, 2, , M$), when user i has a rating on item j, $I_{ij} = 1$, otherwise it is 0. W^+, W are related to GRU-Attention and CNN models, and cannot be optimized like U and V. When U and V are temporarily constant, we observe that L can be interpreted as a squared error function with L2 regularized terms as follows:

$$\Phi(W^+) = \frac{\lambda_U}{2}\sum_i^N \left\|(u_i + \Delta u) - agru(W^+, X_i)\right\|_F^2 + \frac{\lambda_{W^+}}{2}\sum_k^{|w_k^+|}\left\|w_k^+\right\|_2^2 + c \tag{23}$$

$$\nabla_{w_k^+}\Phi(W^+) = -\lambda_U\sum_i^N((u_i + \Delta u) - \nabla_{w_k^+} agru(W^+, X_i) + \lambda_{W^+} w_k^+ \tag{24}$$

$$\Phi(W) = \frac{\lambda_V}{2}\sum_j^M \left\|(v_j + \Delta v) - cnn(W, Y_j)\right\|_F^2 + \frac{\lambda_W}{2}\sum_k^{|w_k|}\|w_k\|_2^2 + c \tag{25}$$

$$\nabla_{w_k}\Phi(W) = -\lambda_V\sum_j^M((v_j + \Delta v) - \nabla_{w_k} cnn(W, Y_j) + \lambda_W w_k \tag{26}$$

The overall optimization process (U, V, W^+ and W are alternatively updated) is repeated until convergence. With optimized U, V, W^+ and W, finally we can predict ratings of users on items:

$$\hat{R}_{ij} \approx E[R_{ij}|(u_i+\Delta u)^T(v_j+\Delta v), \sigma^2] = (u_i+\Delta u)^T(v_j+\Delta v) = \left((agru(W^+, X_i)+\varepsilon_i)+\Delta u\right)^T\left((cnn(W, Y_j)+\varepsilon_j)+\Delta v\right) \tag{27}$$

The optimization process is as follows:

Algorithm 1 AGAMF

Input: user-item rating matrix R; user/item side information X, Y, adversarial perturbation Δu, Δv
Output: optimized latent variables$U$, V, W^+, W

Step1: Initialize U, V, W^+ and W randomly
Step2: For $i \le N$ do:
  Initialize U by $u_i \rightarrow agru(W^+, X_i)$
  End for
Step3: For $j \le M$ do:
  Update V by
  $v_j \rightarrow (UI_jU^T + \lambda_V I_k)^{-1}(UR_j + \lambda_V cnn(W, Y_j))$
  $V \rightarrow v_j + \Delta v$
  End for
Step4: For $j \le M$ do:
  Perform backpropagation and update W through (26)
  End for
Step5: For $i \le N$ do:
  Update U by
  $u_i(VI_iV^T + \lambda_U I_k)^{-1}(VR_i + \lambda_U agru(W^+, X_i))$
  $Uu_i + \Delta u$
  End for
Step6: For $i \le N$ do:
  Perform backpropagation and update W^+ through (24)
  End for
Step7: until convergence
Step8: until satisfying early stopping using a validation set

IV. EXPERIMENTS

A. EXPERIMENTAL SETTINGS

In order to verify the recommended performance of the AGAMF model proposed in this paper. We use keras as the deep learning framework and employ tensorflow as the background. Adam is selected as our optimizer to learn model parameters. The comparative experiment is under the environment of Windows 10 x64-based processors, Pycharm 2018, Inter(R) Core (TM) i7-8700k CPU @ 3.70GHz, 16 GB memory, and python 3.7.

1) DATASETS

The Movielens-1M and Movielens-10M public datasets are widely used in movie scoring prediction. Each user in the dataset has at least 20 rating data, which rating a movie using a 1 (worst) to 5 (best) scale. MovieLens-1M contains more than 1 million rating data on 3706 items from 6040 users. Movielens-10M contains more than 9 million scoring data on 10,073 items from 69,878 users. User auxiliary side information includes attributes such as ID, gender, age, and occupation, and item auxiliary side information includes information such as movie type and movie description. In this experiment, the entire dataset is divided into training data, validation data and test data at the ratio of 80%, 10%, 10%.

2) EVALUATION METRICS

In order to evaluate the performance of the comparison algorithm in rating prediction, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Recall are used as how accurately the recommender system is in predicting rating

values, which are defined as:

$$RMSE = \sqrt{\frac{\sum i,j \in T(R_{ij} - R_{ij})^2}{T}} \tag{28}$$

$$MAE = \frac{\sum i,j \in T|R_{ij} - R_{ij}|}{T} \tag{29}$$

T represents the total number of ratings, R_{ij} represents the observed rating, and $\hat{R}_{ij}$ represents the predicted rating.

$$recall@K = \frac{\sum\limits_{u \in U} |R(u) \cap T(u)|@K}{\sum\limits_{u \in U} |T(u)|} \tag{30}$$

u is the user set, $R(u)$ is the list of items recommended to the user, $T(u)$ is the list of items actually watched by the user, and K is the top K items recommended to the users.

3) BASELINES

To verify the performance of our model, we compare with the following methods:

a. Probabilistic Matrix Factorization (PMF) [9]: it is a probabilistic method for matrix factorization, which assigns a D-dimensional latent feature vector (following Gaussian distributions) for each user and item. The ratings are derived from the inner-product of corresponding latent features.

b. Additional Stacked Denoising Autoencoder (aSDAE) [10]: it fuses aSDAE with the matrix factorization (MF) model to construct a hybrid collaborative filtering model, which can extract effective potential features from auxiliary information at the same time, and obtain the implicit relationship between users and items.

c. Convolutional Matrix Factorization (ConvMF) [12]: it captures subtle contextual differences of a word in a document and further enhances the rating prediction accuracy when the rating data is extremely sparse.

d. A Probabilistic Model of Hybrid Deep Collaborative Filtering (PHD) [17]: it proposes a probabilistic model that combines a stacked denoising autoencoder and a convolutional neural network together with auxiliary side information (e.g., both from users and items) to extract users and items' latent factors.

e. GRU-Attention Matrix Factorization (GAMF): we integrate GRU-Attention and CNN into PMF with users and items' auxiliary side information.

f. Adversarial GRU Matrix Factorization (AGAMF-N): we integrate GRU and CNN into PMF with users and items' auxiliary side information, and the perturbations are enforced on each embedding vector of user and item.

4) PARAMETER SETTINGS

Several comparative experimental method parameters are shown in Table 1:

Considering that the parameters λ_U and λ_V will affect the performance of the AGAMF model, where λ_U and λ_V are balancing parameters [26], the experimental results on the two datasets are shown in Table 2.

TABLE 1. Parameter settings of different methods.

Methods	Movielens-1M	Movielens-10M
PMF	λ_u= 0.01, λ_v= 0.01	λ_u= 0.01, λ_v= 0.01
aSDAE	λ_u=10, λ_v= 100	λ_u= 10, λ_v= 100
ConvMF	λ_u= 200, λ_v= 5	λ_u= 100, λ_v= 1
PHD	λ_u= 3, λ_v= 250	λ_u= 15, λ_v= 80
GAMF	λ_u= 1, λ_v= 500	λ_u= 20, λ_v= 100
AGAMF-N	λ_u= 1, λ_v= 500	λ_u= 20, λ_v= 100
AGAMF	λ_u= 1, λ_v= 500	λ_u= 20, λ_v= 100

TABLE 2. Impacts of λ_u and λ_v on the AGAMF model.

	Density 1.413%		MovieLens-1M		
λ_u	1	1	3	3	3.5
λ_v	500	700	150	900	300
RMSE	0.8362	0.8404	0.8483	0.8561	0.8488
MAE	0.6536	0.6644	0.6661	0.6787	0.6654

	Density 4.641%		MovieLens-10M		
λ_u	0.1	1	20	20	30
λ_v	100	100	100	500	100
RMSE	0.9005	0.8063	0.7518	0.7586	0.7604
MAE	0.7174	0.6328	0.5784	0.5885	0.5943

TABLE 3. The RMSE Performance comparison of different methods in ML-1M and ML-10M.

Datasets	MovieLens-1M	MovieLens-10M
PMF	0.8885	0.8305
aSDAE	0.8725	0.7921
ConvMF	0.8654	0.7883
PHD	0.8578	0.7796
GAMF	0.8472	0.7598
AGAMF-N	0.8429	0.7518
AGAMF	0.8362	0.7452

Table 2 shows the impacts of λ_U and λ_V on two datasets. We observe that on a dataset with sparse rating data, better results can be obtained by decreasing λ_U and increasing λ_V. Setting proper λ_U and λ_V values can map the auxiliary side information of users and items to the appropriate potential space, better balance the auxiliary side information of users and items, and improve the rating prediction accuracy of the AGAMF model.

B. EXPERIMENTAL ANALYSIS

1) RMSE COMPARISON

Discuss the performance of different methods in the same environment. Table 3 shows the rating prediction performance on two different sparsity datasets.

From Table 3 we can observe that AGAMF model achieve better RMSE performance than other methods on the two datasets. On ML-1M, compared with PMF, aSDAE, ConvMF, and PHD models, the RMSE value of the AGAMF model increased by 5.23%, 3.63%, 2.92%, and 2.16% respectively. It shows that the use of GRU based on the attention mechanism and adversarial learning in the framework of matrix factorization have effectively improved the performance of the model. In addition, on ML-10M, the RMSE of the AGAMF model compared with PMF, aSDAE, ConvMF, and PHD models has increased by 9.53%, 4.69%, 4.31%, and 3.44% respectively, indicating the effectiveness of combining users and item's auxiliary side information. It also shows that the AGAMF model has a strong ability to extract auxiliary side information. The performance of AGAMF, AGAMF-N and GAMF is better than PHD model, the first three models all use the GRU network to extract the deep features of the contextual information, and emphasize the long-term dependence between words in the document. Since we consider improving a recommender model by making it resistant to adversarial perturbations on its parameters. We can get a more robust and stable predictive function, and in turn improving its generalization performance.

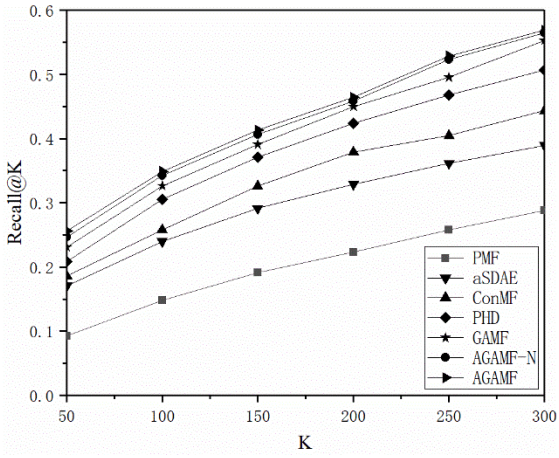
Both AGAMF and AGAMF-N models have better performance than GAMF model. The former two models are enforced on adversarial perturbations to the potential vectors of users and items, it is crucial to increase a model's robustness by learning with adversarial perturbations, which in turn can increase its generalization performance. We believe that this insight is particularly useful for the recommendation. On ML-1M, the performance of AGAMF and AGAMF-N has improved compared with GAMF model. When on ML-10M, the RMSE of AGAMF and AGAMF-N is increased by 1.46% and 0.8% compared with the GAMF model, it shows that adversarial learning can effectively reduce the interference of model training by model parameters, thereby improving model performance.

The performance of the AGAMF is better than the AGAMF-N model. The former model uses the attention mechanism to express the characteristic information of important words, assigns corresponding weights to each word, and highlights the key information in the context. Both on ML-1M and ML-10M, the RMSE of the AGAMF model is 0.67% higher than that of the AGAMF-N model on average, which verifies the effectiveness of the attention mechanism.

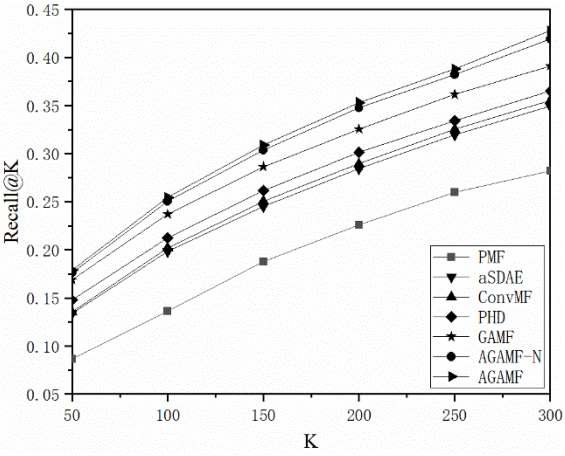
2) RECALL COMPARISON

Discuss the recall of top-K value on different methods. Experiments are performed on two datasets with different sparsity, as shown in Figure 4.

From Figure 4 we can observe that several algorithms are on the rise with the increase of K value on the two datasets. Among them, the traditional method PMF has the lowest performance because PMF ignores the auxiliary side information of users and items, which makes the recommendation result poor. The performance of the PHD model



(a) MovieLens-1M



(b) MovieLens-10M

FIGURE 4. The impact of top-K value on recall.

is better than that of aSDAE and ConvMF, which shows that the combination of traditional matrix factorization and deep learning models can learn effective latent factors and better extract auxiliary side information. The performance of AGAMF, AGAMF-N and AGMF models is significantly better than the PHD model, which shows that GRU can establish long-term dependence between words can make up for the shortcomings of aSDAE to extract context information, better representation and modeling of the context, thereby improving recommended performance. Models other than PMF perform better In ML-1M, indicating that neural network-based models are more suitable for sparse relational data. The AGAMF model is superior to the AGAMF-N model because the attention mechanism can adaptively combine context information to achieve different levels of attention to context information, effectively improve the accuracy of model classification, and thereby improve model performance. The AGAMF model is better than the GAMF model because the introduction of adversarial learning makes the model fitting process stable and the model robustness is enhanced. The AGAMF model still has robust and good performance when

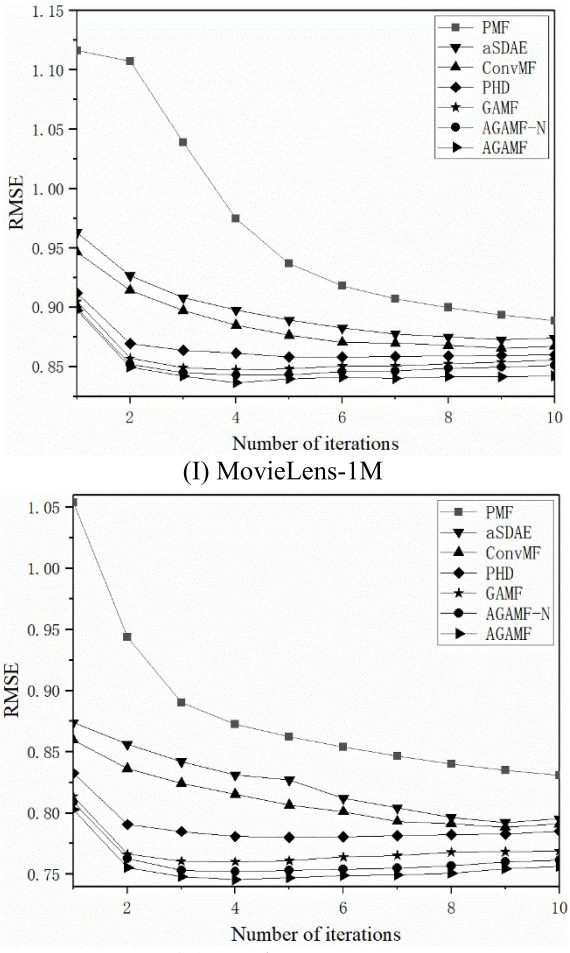


FIGURE 5. The effect of iterations on models' RMSE.

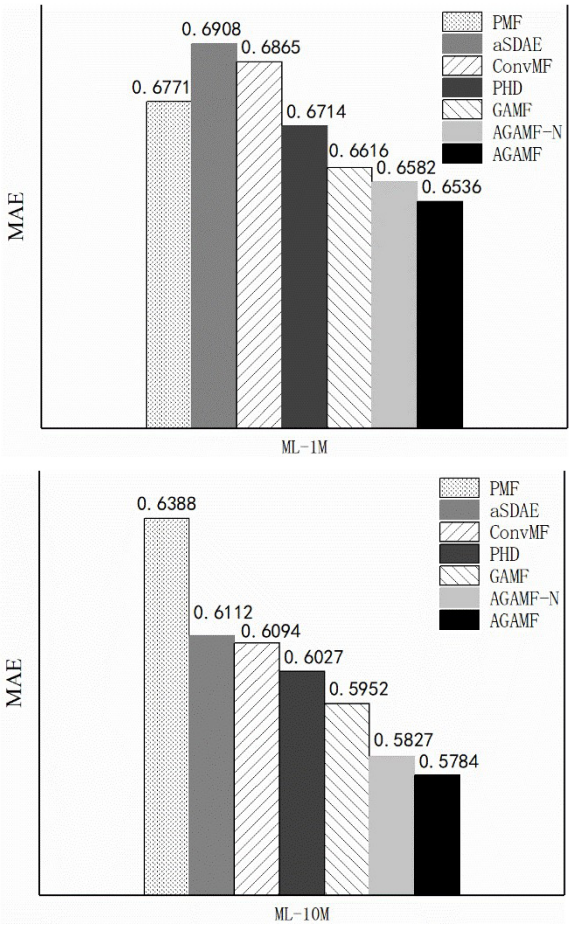


FIGURE 6. Comparison of MAE of different models on two datasets.

compared with traditional approaches and other deep learning models.

3) THE IMPACT OF ITERATIONS ON PERFORMANCE

Discuss different methods' RMSE on two datasets with different sparsity under different iteration. As shown in Figure 5.

From Figure 5 it can be seen that several models' RMSE gradually decreases as the number of iterations increases, and eventually stabilizes. However, too many iterations will result in lower model performance, because too many iterations will cause the model to overfit, resulting in poor performance. The AGAMF model is better than the PHD model, indicating that GRU and attention layer can quickly extract the deep features of the context, highlight the key information of the context, and converge faster than the coding layer of the additional stacked denoising autoencoder. The adversarial learning makes the model fitting process stable, so that the recommendation performance is better. The AGAMF, AGAMF-N, and GAMF models converge faster, and the RMSE is better than the other 4 models during initial training. High performance can be achieved with a small number of iterations, which can achieve a high number of cost-effective iterations performance, that is, the fewer iterations, the more effective the training process.

4) MAE COMPARISON

Discuss the performance of different methods in the same environment. Experiment on two datasets with different sparsity, as shown in Figure 6.

Figure 6 shows that AGAMF model's MAE is better than other models on the ML-1M and ML-10M dataset. On the sparse ML-1M dataset, PMF's performance is better than aSDAE and ConvMF, indicating that when the data matrix is too sparse, aSDAE and ConvMF can't effectively extract the latent factors. Even the data is too sparse, compared with traditional methods and other deep learning models, the AGAMF model still has good performance. This shows that deep learning structure can create better quality of auxiliary information features, especially when GRU-Attention and CNN are combined to extract latent factors that is more effective. In addition, after adversarial perturbations are added, the larger backward error caused by the interference of the linear model parameters is reduced, so that the model is more robust.

V. CONCLUSION

Due to the data sparsity problem in traditional recommendation systems, the recommendation model is extremely vulnerable to the interference of its parameters, and the additional

stacked denoising autoencoder lacks the ability to extract deep features and key information of the context. This paper adopts a matrix factorization recommendation model combining adversarial learning and GRU-Attention to improve recommendation performance. In addition, compared with several methods that combine deep learning, the experimental results show that the AGAMF model shows good results on two datasets. This shows that the AGAMF model proposed in this paper improves the recommendation performance by modeling auxiliary side information, fully learning the contextual semantic relationship, and adding adversarial perturbations to stabilize the fitting process.

Although the accuracy of the AGAMF model has been improved to some extent, due to the sparsity rating information and auxiliary side information of users and items. The model framework combined with deep learning is complex, the training time is long, and the experimental results are not greatly improved. Shainoor J etc. [27] gives a good research idea, and follow their framework we will consider how to deal with sparsity data more effectively and build a simplified and reasonable recommendation framework in the future.

REFERENCES

[1] G. Tian and L. Jing, "Recommending scientific articles using bi-relational graph-based iterative RWR," in *Proc. 7th ACM Conf. Recommender Syst. - RecSys*, 2013, pp. 399–402.

[2] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–45, Jul. 2014.

[3] X. Li and D. Li, "An improved collaborative filtering recommendation algorithm and recommendation strategy," *Mobile Inf. Syst.*, vol. 2019, pp. 1–11, May 2019.

[4] T. Trinh, D. Wu, R. Wang, and J. Z. Huang, "An effective content-based event recommendation model," *Multimedia Tools Appl.*, pp. 1–20, Apr. 2020.

[5] Y. Cao, W. Li, and D. Zheng, "A hybrid recommendation approach using LDA and probabilistic matrix factorization," *Cluster Comput.*, vol. 22, no. S4, pp. 8811–8821, Jul. 2019.

[6] J. Shu, X. Shen, H. Liu, B. Yi, and Z. Zhang, "A content-based recommendation algorithm for learning resources," *Multimedia Syst.*, vol. 24, no. 2, pp. 163–173, Mar. 2018.

[7] J. Zhao, Z. Liu, H. Chen, J. Zhang, and Q. Wen, "Hybrid recommendation algorithms based on ConvMF deep learning model," in *Proc. Int. Conf. Wireless Commun., Netw. Multimedia Eng. (WCNME)*, 2019, pp. 159–162.

[8] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 2009, vol. 42, no. 8, pp. 30–38.

[9] A. Mnih, R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, vol. 2008, pp. 1257–1264.

[10] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, "A hybrid collaborative filtering model with deep structure for recommender systems," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1309–1315.

[11] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 3, pp. 3371–3408, 2010.

[12] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 233–240.

[13] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 355–364.

[14] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Short Papers)*, vol. 2, 2016, pp. 207–212.

[15] Q. Zhang *et al.*, "Standardized matrix factorization recommendation algorithm based on attention mechanism," *J. Softw.*, to be published.

[16] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 950–958.

[17] J. Liu, D. Wang, and Y. Ding, "PHD: A probabilistic model of hybrid deep collaborative filtering for recommender systems," in *Proc. Asian Conf. Mach. Learn.*, 2017, pp. 224–239.

[18] Y. Wu, D. Bamman, and S. Russell, "Adversarial training for relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1778–1783.

[19] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semisupervised text classification," in *Proc. ICLR*, 2017, pp. 1–11.

[20] S. Park, J. Park, J. Shin, and I. C. Moon, "Adversarial dropout for supervised and semi-supervised learning," in *Proc. AAAI*, 2018, pp. 3917–3924.

[21] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. ICLR*, 2015, pp. 1–11.

[22] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1765–1773.

[23] C. Deng, W. W. Che, and P. Shi, "Cooperative fault-tolerant output regulation for multiagent systems by distributed learning control approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4831–4841, Nov. 2020.

[24] Y. Wei, X. Wang, W. Guan, L. Nie, Z. Lin, and B. Chen, "Neural multimodal cooperative learning toward micro-video understanding," *IEEE Trans. Image Process.*, vol. 29, pp. 1–14, 2020.

[25] C. Cui, J. Shen, L. Nie, R. Hong, and J. Ma, "Augmented collaborative filtering for sparseness reduction in personalized POI recommendation," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 5, pp. 1–23, Sep. 2017.

[26] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining - KDD*, 2011, pp. 448–456.

[27] S. J. Ismail, K. Hardy, M. C. Tunis, K. Young, N. Sicard, and C. Quach, "A framework for the systematic consideration of ethics, equity, feasibility, and acceptability in vaccine program recommendations," *Vaccine*, pp. 1–17, Jun. 2020.

HONGBIN XIA received the Ph.D. degree from Jiangnan University, Wuxi, China.

He is currently an Associate Professor and the Vice Dean of the Department of Artificial Intelligence and Computer Science, Jiangnan University. He received the grand prize of the Science and Technology Award of the China Federation of Commerce. His research interests include computer network optimization, natural language processing, and computational intelligence.

JING JING LI is currently pursuing the M.S. degree with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. Her research interests include deep learning and recommendation systems.

YUAN LIU is currently a Professor with the School of Artificial Intelligence and Computer Science, Jiangnan University. He engaged in the research and development of network information systems and network security, and digital media related software. His current research interests include network traffic measurement, social networking, and digital media.