Probabilistic Matrix Factorization Recommendation of Self-Attention Mechanism Convolutional Neural Networks With Item Auxiliary Information

CHENKUN ZHANG and CHENG WANG
College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China
Corresponding author: Cheng Wang (wangcheng@hqu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71771094 and Grant 71571056, in part by the Guiding Project of Science and Technology Plan of Fujian Province of China under Grant 2017H01010065, in part by the Project of Quanzhou Science and Technology Plan of China under Grant 2018C110R and Grant 2018C114R, and in part by the 2018 Huaqiao University Research and Establishment of Postgraduate Education and Teaching Reform Project under Grant 18YJG28.

ABSTRACT To solve the problem of data sparsity in recommendation systems, this paper proposes a probabilistic matrix factorization recommendation of self-attention mechanism convolutional neural networks with item auxiliary information. First, the self-attention mechanism is added to convolutional matrix factorization and a probabilistic matrix factorization model, based on a convolutional neural networks with self-attention mechanism, is proposed. Second, after integrating auxiliary information, such as item comment, item name, and item category, probabilistic matrix factorization, based on a self-attention mechanism convolutional neural networks, is used for recommendation. Adding the self-attention mechanism allows convolutional matrix factorization to capture the long-distance dependence between different components of the auxiliary information. Integrating the item comment, name, and category information alleviates the data sparsity of recommendation, and improves the accuracy of rating prediction. Experimental results on the MovieLens-1M and MovieLens-10M datasets show that the probabilistic matrix factorization recommendation of self-attention mechanism convolutional neural networks with item comment, name, and category information is superior to existing popular methods, in respect of root mean square error.

INDEX TERMS Recommendation algorithm, self-attention mechanism, convolutional neural networks, probabilistic matrix factorization, item auxiliary information.

I. INTRODUCTION
With the development of society and science, people gradually enter the era of information overloaded from the era of lack of information. How to find the information they are interested in from the massive information and how to make the information they release stand out has become an urgent problem to be solved. Recommender system [1]–[3] is an automatic system to recommend items to users according to their preferences. It greatly saves the time spent by users, and has certain applications for major e-commerce websites, music websites and online video websites.

However, with the rapid growth of the number of users and items, the interactive data between users and items, such as rating data, becomes more and more sparse, which eventually leads to the decline of rating prediction accuracy. Although the existing methods can alleviate the data sparsity by using auxiliary data, they can’t make full use of the auxiliary information, so the rating prediction accuracy needs to be improved.

In this paper we add the self-attention mechanism to the convolutional matrix factorization (ConvMF) model, and propose a probabilistic matrix factorization (PMF) model, based on a convolutional neural networks with self-attention mechanism (SAConvMF). The model uses a convolutional neural networks (CNN) with self-attention mechanism to process auxiliary information. It can capture both the context information and the long-distance dependence between different components of auxiliary information. Finally, to apply the model to recommendation, we expand the auxiliary information.
information to include the item comment information, item name information, and item category information, to further alleviate the data sparsity and improve the accuracy of rating prediction. The main contributions of this paper are as follows:

1. We propose a PMF model based on a CNN with self-attention mechanism, named SAConvMF. The self-attention mechanism CNN (Self-Att-CNN) is used to learn auxiliary data, so that both the context information and the long-distance dependence between different components of auxiliary data can be captured.

2. After integrating the auxiliary information, comprising the item comment, name, and category, SAConvMF is used for recommendation, because this method makes full use of auxiliary information and further alleviates data sparsity, so it improves the accuracy of rating prediction.

3. SAConvMF, biased singular value decomposition (BiasSVD), singular value decomposition considering user implicit feedback (SVD++), PMF, collaborative topic regression (CTR), collaborative deep learning (CDL), and ConvMF are compared and analyzed on the MovieLens-1M and MovieLens-10M public datasets.

II. RESEARCH BACKGROUND
A. LITERATURE REVIEW BACKGROUND

In 2007, Salakhutdinov and Mnih [4] considered the situation where the existing data in a matrix are generated by the interaction of two targets and conform to the Gaussian distribution, for which they proposed the PMF model. Compared with the existing collaborative filtering algorithm, PMF is better suited to large-scale, sparse, and extremely unbalanced datasets. Its main application fields include recommendation, link prediction, biological path analysis, document modeling, and transmission path discovery [5]–[10]. The PMF model is trained according to the existing data in the matrix. In many fields, the prediction accuracy will be affected when the data are extremely sparse. Therefore, in recent years, some researchers have introduced auxiliary data into PMF and processed them with corresponding models, to alleviate the data sparsity in each field. Ryan et al. [8] proposed the DPMF framework, which coupled multiple PMF problems together by Gaussian process priori and combined them with edge information, and successfully used it to predict scores of basketball games. Some researchers have achieved good results by introducing auxiliary data and modeling for documents. Wang et al. [9] proposed the CTR model by combining latent Dirichlet allocation (LDA) and PMF. This used LDA to process a document auxiliary information, and LDA was combined with PMF. Wang et al. [10] proposed the CDL model by combining a stacked denoising autoencoder (SDAE) with PMF. This used SDAE to process a document auxiliary data, and SDAE was then combined with PMF. However, the CTR and CDL models use a bag-of-words model to process a document auxiliary data, so these models cannot capture context information, resulting in poor understanding of a document’s data. In view of the shortcomings of the CTR and CDL models, Kim et al. [11] proposed the ConvMF model by combining a CNN with PMF. The CNN was used to process auxiliary data to capture context information. However, a pure CNN cannot capture the long-distance dependence between different components of the auxiliary information, this is a problem that is addressed in this paper.

As an important part of recommendation systems, collaborative filtering has become one of the mainstream recommendation methods. The singular value decomposition (SVD) recommendation model [12] is often used to analyze user preferences in collaborative filtering, based on potential factors. On the basis of SVD, Paterek [13] proposed the BiasSVD recommendation model, which describes the potential characteristics of users and items in more detail. It is one of the classic SVD recommendation models. Koren [14] use the implicit feedback of user behavior and propose the SVD++ recommendation model. Combining the advantages of the neighborhood model and potential factor model, the SVD++ recommendation model can infer user preferences based on more implicit feedback and analyze their potential opinions through user behavior.

PMF [4] is also one of the classic collaborative filtering algorithms, but suffers from the problem of data sparsity. Therefore, to alleviate data sparsity, Wang et al. [9], Wang et al. [10], and Kim et al. [11] applied the CTR, CDL, and ConvMF models, respectively, to recommendation. However, because of the defects of the models themselves, these models did not learn enough auxiliary data. In addition, many algorithms have been proposed to alleviate data sparsity and improve recommendation accuracy. Al-Bashiri and Salehudin [15] used item attribute data and rating data to calculate the user’s global preference. Based on this, the calculation of similarity and score prediction were improved, so the data sparsity was alleviated. Liji et al. [16] used a clustering algorithm to cluster users, filled the scoring matrix with the scoring time and item type, and finally performed collaborative filtering in each cluster to improve the recommendation accuracy. The effect of this is greatly affected by the filling value, and an inaccurate filling value may become noise data, which worsens the recommendation accuracy. Wu et al. [17] introduced timestamp data and used the long short-term memory (LSTM) to model the users and items, and then expressed and predicted them dynamically. Finally, the user and item vectors were used simultaneously to predict scores, but the process was time-consuming. Suglia et al. [18] used the LSTM to process each item’s time-ordered auxiliary data and obtained the item feature vector, which was combined with the user feature vector to predict the probability of users clicking on the item, but this process was time-consuming. Mao et al. [19] used a CNN that considered time information to extract information from auxiliary data on users and items separately, and then used the fully connected network to predict the score of the two. This model alleviates the data sparsity and can learn the trends in user preferences and item attributes that change over time, to better capture the
interaction between users and items. However, the model cannot capture the long-distance dependence between different components of auxiliary information, and the learning of auxiliary data is not sufficient.

The introduction of auxiliary information can alleviate data sparsity and improve prediction accuracy. However, most of the existing models to process auxiliary information cannot fully learn the information in the auxiliary data. For example, the ConvMF model uses a CNN to process documents’ auxiliary information. Although it can capture the context information in the auxiliary data, it cannot also capture the long-distance dependence between different components of auxiliary data. For example, consider the sentence, “the data are becoming more and more sparse, which leads to a decline in the quality of recommendations.” There is a causal relationship between “data sparsity” and “recommendation quality degradation,” but the ConvMF model can only capture local features. it cannot capture long-distance dependence, particularly when the sentence is very long. Therefore, it cannot capture causality, so its accuracy needs to be improved. Vaswani et al. [20] proposed a transformer model that was completely based on an attention network. Since then, attention networks have been widely used in recommendation, machine translation, natural language processing, and other fields. Wang et al. [21] and others proposed a text classification model based on the self-attention mechanism and the CNN. The model uses the CNN to capture local features and location information and uses the self-attention mechanism to capture long-distance dependence, which can capture the information in the text. Therefore, this paper uses this model to process auxiliary data and combines the model with PMF.

B. RELATED WORK

1) PROBABILISTIC MATRIX FACTORIZATION

The PMF frame diagram is shown in figure 1. Given the matrix \( R \in R^{M \times N} \), there exists a small latent variable \( k \) makes \( R = U^T \times V \), where \( U \in R^{k \times M} \), \( V \in R^{k \times N} \), and \( U \) and \( V \) are unknown low dimensional matrices, and \( R \) is a low rank matrix with rank no more than \( k \). In order to solve the above matrix factorization problem, it is assumed that the conditional probability distribution of the known data in \( R \) is as follows:

\[
p(R | U, V, \sigma^2) = \prod_i^{M} \prod_j^{N} [N(R_{ij}|U_i^T V_j, \sigma^2)]^{l_0} \tag{1}
\]

where \( N(x|\mu,\sigma^2) \) is a Gaussian distribution with mean value \( \mu \) and variance \( \sigma^2 \). \( l_0 \) is an indicative function, when the data \( R_{ij} \) corresponding to row \( i \) and column \( j \) in matrix \( R \) is known, it is 1, otherwise it is 0.

The zero-mean spherical Gaussian priors are applied to \( U \) and \( V \):

\[
p(U | \sigma_u^2) = \prod_i^{M} N(U_i|0, \sigma_u^2 I) \tag{2}
\]

\[
p(V | \sigma_v^2) = \prod_j^{N} N(V_j|0, \sigma_v^2 I) \tag{3}
\]

\[
\sigma_u^2 = \frac{1}{2} \sum_i^{M} \sum_j^{N} \lambda_U \| U_i \|_2^2 + \frac{1}{2} \sigma_u^2 \ln(\sigma_u^2) + K \lambda_U \sigma_u^2 + C \tag{4}
\]

For the convenience of calculation, the natural logarithm are adopted for the posterior distribution of \( U \) and \( V \), and the formula (4) is obtained:

\[
\ln p\left(U, V | R, \sigma^2, \sigma_u^2, \sigma_v^2\right) = \frac{1}{2} \sum_i^{M} \sum_j^{N} \lambda_U \| U_i \|_2^2 + \frac{1}{2} \sigma_u^2 \ln(\sigma_u^2) + K \lambda_U \sigma_u^2 + C \tag{5}
\]

where \( C \) is a constant. When the super parameter (i.e. observation noise variance \( \sigma^2 \) and prior variance \( \sigma_u^2, \sigma_v^2 \)) remains unchanged, maximizing log posteriori are equivalent to minimizing the following objective functions:

\[
E = \frac{1}{2} \sum_i^{M} \sum_j^{N} \lambda_U \| U_i \|_2^2 + \lambda_V \| V_j \|_2^2 \tag{6}
\]

For equation (5), we can use the stochastic gradient descent method for iterative training to establish the model.

2) SELF-ATTENTION MECHANISM

The core of self-attention mechanism is dot-product attention [20]. The calculation process of dot-product attention is shown in the left figure of figure 2, which is defined as follows:

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}
\]

where \( Q, K, V \) denote “query”, “key” and “value” respectively; \( d_k \) is the scaling factor, which represents the dimension of \( K \). For the larger \( d_k \) value, the product of the dot product
is too large, so the softmax function is pushed to the region with minimal gradient [20]. To counter this effect, $\frac{1}{\sqrt{d_k}}$ is used to scale point product [20].

The self-attention mechanism is shown in the right figure of figure 2. When the input data is $Z$, in the self-attention mechanism, $Q, K, V$ are all linear transformations of the same vector. Therefore, self-attention is defined as follows:

$$
\text{Self-Attention} (Z) = \text{Attention}(ZW_Q, ZW_K, ZW_V) = \text{softmax}(\frac{ZW_QW_K^TZV_T}{\sqrt{d_k}})ZW_V
$$

where, $W_Q, W_K, W_V$ are all model parameters, which are obtained during model training.

### III. PROBABILISTIC MATRIX FACTORIZATION BASED ON CONVOLUTIONAL NEURAL NETWORKS WITH SELF-ATTENTION MECHANISM

In PMF, two low-rank matrices $U$ and $V$—obtained by decomposing a given matrix $R$—can represent the row and column implicit vector matrices of $R$. That is, $U_i$ and $V_j$ are the implicit vectors of row $i$ and column $j$, respectively, and $R_{ij} \approx U_i^TV_j$.

When PMF is applied to a certain field, the data in the original matrix $R$ have practical significance, and the rows and columns of $R$ also have practical significance. For example, in [5], PMF was used to predict the association between miRNA and disease. In this case, a row of matrix $R$ represented miRNA and a column represented disease, and the known data in $R$ represented a certain correlation between them. As another example, [9]–[11] used PMF to predict users’ scores on items in the recommendation field. In this case, the original matrix $R$ represents the user–item rating matrix, and the rows and columns of $R$ represent the users and items, respectively.

The reason why we introduce these is because the Self-Att-CNN in the SAConvMF model processes only the auxiliary data attached to the columns of the original matrix $R$. Therefore, as shown in Fig. 3, the auxiliary data $X$ and all parameter matrices $W$ of the Self-Att-CNN both act on the low-rank matrix $V$.

#### A. PROBABILITY MODEL OF SACONVMF

Similarly to PMF, given the matrix $R, R \in \mathbb{R}^{M \times N}$, the goal of the model is first to find two low-rank matrices $U \in \mathbb{R}^{K \times M}$ and $V \in \mathbb{R}^{L \times N}$ such that $R = U^TV$. That is, the original matrix $R$ can be reconstructed according to the matrices $U$ and $V$. From the perspective of probability, the conditional probability distribution of the known data in matrix $R$ is as follows:

$$
\text{p}(R | U, V, \sigma^2) = \prod_i^M \prod_j^N [N(R_{ij}|U_i^TV_j, \sigma^2)]^{I_{ij}}
$$

The sign here is consistent with that in (1). The zero-mean spherical Gaussian prior is applied to $U$, as follows:

$$
\text{p}(U | \sigma^2_U) = \prod_i^M N(U_i|0, \sigma^2_U I)
$$

However, because the SACConvMF model uses a Self-Att-CNN to process auxiliary data attached to the columns of matrix $R$, the conditional distribution of the low-rank matrix $V$ differs from that of PMF:

$$
\text{V}_j = \text{Self-Att-CNN} (W, X_j) + \epsilon_j
$$

For each $w$ in $W$, we apply the zero-mean spherical Gaussian prior:

$$
\text{p}(W | \sigma^2_W) = \prod_k^M N(w_k|0, \sigma^2_W)
$$

In conclusion, the conditional distribution of $V$ is as follows:

$$
\text{p}(V | W, X, \sigma^2_V) = \prod_j^N N(V_j|\text{Self-Att-CNN} (W, X_j), \sigma^2_V I)
$$

Here, $\text{Self-Att-CNN}$ represents the Self-Att-CNN and $X_j$ represents the auxiliary data attached to column $j$. The Gaussian conditional probability distribution with mean $\text{Self-Att-CNN} (W, X_j)$ is a bridge combining the Self-Att-CNN and PMF [11].
B. SELF-ATTENTION MECHANISM CONVOLUTIONAL NEURAL NETWORKS OF SACONVMF

In the SAConVMF model, the Self-Att-CNN framework for processing auxiliary data is shown in Fig. 4. The auxiliary data \( X = \{X_1, X_2, \ldots, X_{\text{num}}\} \), where \( \text{num} \) is the number of types of auxiliary data. The Self-Att-CNN network consists of the following network layers: 1) embedding layer, 2) convolution layer, 3) self-attention layer, 4) pooling layer, and 5) output layer. From the figure, we can observe that each item of auxiliary data corresponds to multiple channels (each channel is composed of the convolution layer, self-attention layer, and pooling layer). This is because each item of auxiliary data may correspond to a variety of convolution kernels, so the final results will be richer. That is, each channel corresponds to a convolution kernel, so the final output of multiple channels should be spliced together to obtain the abstract representation of the auxiliary data. Finally, the abstract representations of each kind auxiliary data are spliced together, and then the abstract feature representation of all auxiliary data is obtained through the output layer.

![Self-attention mechanism CNN](image)

FIGURE 4. Self-attention mechanism CNN.

Next, we take a type of auxiliary data \( D \) (document) passing through a channel as an example to illustrate the main process of the Self-Att-CNN. In the final output layer, we splice together the feature vectors of the auxiliary data \( D \) obtained through multiple channels, and then use them as the input to the fully connected layer to obtain the final feature vector of the auxiliary data \( D \).

1) EMBEDDING LAYER

The embedding layer transforms the input original auxiliary data to a dense digital matrix and serves as the input to the next layer, which is the convolution layer. Supposing that the auxiliary data \( D \) are represented by a document of length \( l \), the document \( D \) is represented as a matrix by connecting the word vectors in the document. Word vectors can be initialized or randomly initialized with a pretrained word embedding model (e.g., Glove [22]). Through the optimization process, the word vector is further trained. The document matrix \( D \in \mathbb{R}^{l \times p} \) then becomes:

\[
D = [\ldots, w_{i-1}, w_i, w_{i+1}, \ldots]
\]

where, \( l \) is the length of document \( D \) and \( p \) is the length of word vector \( w_i \), for the i-th word.

2) CONVOLUTION LAYER

The convolution layer is used to extract the context features. For example, the j-th sharing weight \( W_j^c \in \mathbb{R}^{l \times ws} \) is used to extract the context feature \( c_j^i \in \mathbb{R}^{l \times ws+1} \) of document \( D \) can then be calculated from the sharing matrix \( W_j^c \) by (14):

\[
c_j^i = f(W_j^c \ast D(:, i: (i + ws - 1))) + b_j^i
\]

where, \( \ast \) is the convolution operation, \( b_j^i \in \mathbb{R} \) is the offset corresponding to \( W_j^c \), and \( f \) is the nonlinear activation function (ReLU is used in this paper). The context eigenvector \( c_j^i \in \mathbb{R}^{l \times ws+1} \) of document \( D \) can then be calculated from the sharing matrix \( W_j^c \) by (14):

3) SELF-ATTENTION LAYER

The self-attention layer is used to capture the long-distance dependence between the context feature vectors \( c_j^i \) of document \( D \). The specific operation is as follows.

When the \( nc \) shared weight matrices convolute the document \( D \), a matrix \( Z \in \mathbb{R}^{l \times ws+1 \times nc} \) is obtained. The j-th column of \( Z \) represents the context feature vector \( c_j^i \), whose size is \( l \times ws+1 \), after convoluting the document \( D \) by sharing weight \( W_j^c \). We calculate the matrix \( Z \) as follows:

\[
\text{Self-Attention}(Z) = \text{Attention}(ZW_Q, ZW_K, ZW_V) = \text{softmax}(\frac{ZW_QW_K^TZ^T}{\sqrt{d_k}})ZW_V
\]

Here, \( W_Q \in \mathbb{R}^{nc \times nc} \), \( W_K \in \mathbb{R}^{nc \times nc} \), and \( W_V \in \mathbb{R}^{nc \times nc} \) are the parameter matrices. The value of \( d_k \) is \( nc \). The Self-Attention (Z) is then passed through a residual layer [23] and a normalization layer [24]:

\[
O = \text{LayerNormalization}(\text{Self-Attention}(Z) + Z),
\]

where, \( O=[o^1, o^2, \ldots, o^j, \ldots, o^{nc}] \).

4) POOLING LAYER

The pooling layer is used to extract representative features. By extracting only the largest context feature from each context feature vector, the document representation is simplified to \( nc \) fixed-length vectors, as follows:

\[
d_j = [\max(o^1), \max(o^2), \ldots, \max(o^j), \max(o^{nc})],
\]

where, \( o^j \) is the context feature vector \( c^j \) after the self-attention calculation.
5) OUTPUT LAYER
d_f is the feature vector of a type of auxiliary data D after passing through a channel. However, in the output layer, we need to splice the feature vector generated by the auxiliary data D through different channels to form the feature vector of D, and finally input it to the fully connected layer. We use d_f~pD = [d_f~p_1, d_f~p_2, ..., d_f~p~p_num] to denote the feature vector that D passes through p_num channels, which is then stitched together:

\[
\text{final_feature} = \text{PReLU}(W_f_1 \text{PReLU}(W_f_1 d_f~pD + b_f_1) + b_f_2) \tag{19}
\]

Here, W_f_1 ∈ R^f × p_num × nc and W_f_2 ∈ R^f × f are the weight matrices, b_f_1 ∈ R^f and b_f_2 ∈ R^f are the corresponding offsets, final_feature ∈ R^f, and PReLU is the activation function.

That is, the characteristic representation of the auxiliary data X_j with column j final_feature_j is as follows:

\[
\text{final_feature}_j = \text{Self} - \text{Att} - \text{CNN}(W, X_j) \tag{20}
\]

C. MODEL TRAINING OF SACONVMF
To optimize U, V, and W, we use the maximum a posteriori (MAP) probability estimation:

\[
\begin{align*}
\text{max}_{U, V, W} & \quad p(U, V, W | \text{R}, X, \sigma^2, \sigma^2_U, \sigma^2_V, \sigma^2_W) \\
& = \text{max}_{U, V, W} [p(R(U, V, \sigma^2)p(U|\sigma^2_U)p(V|W, X, \sigma^2_V)p(W|\sigma^2_W)]
\end{align*} \tag{21}
\]

By taking the negative logarithm in (21), it is reformulated as follows:

\[
\Gamma(U, V, W) = \sum_i \sum_j \sum_{iN} \frac{I_{ij}}{2} (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2} \sum_i ||U_i||_2^2
\]

\[
+ \frac{\lambda_V}{2} \sum_j ||V_j - \text{Self} - \text{Att} - \text{CNN}(W, X_j)||_2^2
\]

\[
+ \frac{\lambda_W}{2} \sum_k ||w_k||_2^2, \tag{22}
\]

where, \lambda_U is \sigma^2/\sigma^2_U, \lambda_V is \sigma^2/\sigma^2_V and \lambda_W is \sigma^2/\sigma^2_W.

Equation (22) can be solved by the coordinate descent method. That is, when the other variables are fixed, the remaining variables are optimized iteratively. When W and V (U) are fixed, the objective function \Gamma is a quadratic function of U (V). The optimal solution of U (or V) can be computed analytically in a closed form by simply differentiating the optimization function \Gamma with respect to U_i (or V_j), as follows:

\[
U_i \leftarrow (VI_j^T + \lambda_U I_K)^{-1} V J_i \tag{23}
\]

\[
V_j \leftarrow (U J_i^T + \lambda_V I_K)^{-1} (U J_i U^T + \lambda_Y S_e_l_f - A_t_t - \text{CNN}(W, X_j) \tag{24}
\]

Here, \(I_i \) is a diagonal matrix whose diagonal element is \delta_{ij}, \(i = 1, \ldots, N \), \(R_i = [R_{i1}, R_{i2}, \ldots, R_{iN}] \) is the i-th row vector of matrix R. The definitions of \(I_i \) and \(R_i \) are similar.

Because W is a parameter in the Self-Att-CNN, it cannot be solved analytically, like U and V. However, when U and V are temporarily unchanged, \Gamma can be interpreted as the square error function of the L2 regular term.

\[
e(W) = \frac{\lambda_V}{2} \sum_j ||V_j - \text{Self} - \text{Att} - \text{CNN}(W, X_j)||_2^2
\]

\[
+ \frac{\lambda_W}{2} \sum_k ||w_k||_2 + \text{constant} \tag{25}
\]

We can use the backpropagation algorithm to solve the problem of W.

The whole optimization process (U, V, and W are updated alternately) is repeated until convergence. Finally, through the optimized U, V, and W, we can predict the missing data:

\[
R_{ij} \approx E(R_{ij} | U_i^T V_j, \sigma^2)
\]

\[
= U_i^T V_j = U_i^T (\text{Self} - \text{Att} - \text{CNN} (W, X_j) + \epsilon_j) \tag{26}
\]

D. THEORETICAL ANALYSIS AND COMPARISON OF THE MODELS
In this paper, we compare the SACOnMV model with the ConvMF, CDL, CTR, and PMF models. The PMF model is a classical matrix factorization model, which is widely used in every field, but its prediction accuracy is often low because of the influence of data sparsity in various fields. The CTR and CDL models are the improved models for this short board of PMF. In the CTR model, LDA is used to process auxiliary data and is combined with PMF to alleviate data sparsity and improve prediction accuracy. However, LDA is not sufficiently deep to learn auxiliary data. The CDL model uses SDAE to process auxiliary data and combines it with PMF. Compared with LDA, SDAE can learn auxiliary data in depth. However, the CTR and CDL models both adopt the bag-of-words model, so they cannot capture context information when processing auxiliary data. In view of this shortcoming, the ConvMF model uses a CNN to process auxiliary data to capture context information, but this model cannot capture the long-distance dependence between different components of the auxiliary data. To solve this problem of the ConvMF model, this paper proposes a PMF model based on the Self-Att-CNN (namely SACOnMF). This model can capture both the context information and the long-distance dependence between different components. A theoretical comparison of the models is shown in Table 1.

IV. PROBABILISTIC MATRIX FACTORIZATION RECOMMENDATION OF SELF-ATTENTION MECHANISM
CONVOLUTIONAL NEURAL NETWORKS WITH ITEM AUXILIARY INFORMATION
In the field of recommendation, the problem of data sparsity has always had a strong influence on the accuracy of recommendation, which is an active research topic. When the SACOnMF model is applied to recommendation, its original matrix R ∈ R^{M × N} represents the user-item rating matrix,
such that $R_{ij}$ represents user i’s rating of item j, while the decomposed low-rank matrices $U \in \mathbb{R}^{k \times M}$ and $V \in \mathbb{R}^{k \times N}$ represent the user-feature matrix and item-feature matrix, respectively. Our task is to predict the user’s rating of the item.

### A. INTEGRATING ITEM AUXILIARY INFORMATION INTO PROBABILITY MATRIX FACTORIZATION RECOMMENDATION OF SELF-ATTENTION MECHANISM CONVOLUTIONAL NEURAL NETWORKS

In this paper, we use item auxiliary information, such as comment data, name data, and category data. As described in Section III, our SAConvMF model handles the auxiliary information of items through a Self-Att-CNN. Therefore, when the model is used for recommendation, we also use the Self-Att-CNN to process the auxiliary information of the item. The flowchart is shown in Fig. 5. The flow is the same as that explained in Section III, so it will not be analyzed here.

### B. THE ALGORITHM

The flowchart for the PMF recommendation of the Self-Att-CNN with item auxiliary information is shown in Fig. 6.

### C. THEORETICAL ANALYSIS AND COMPARISON OF METHODS

The BiasSVD and SVD++ algorithms are commonly used in the recommendation field, particularly for collaborative filtering. BiasSVD has the problem of data sparsity, while SVD++ makes use of implicit feedback data from users, but such data are not fully utilized. Similarly, when PMF is used for recommendation, its prediction accuracy is also often low because of the sparsity of rating data. To solve this problem, Wang et al. [9] used the LDA model to process the information comprising the item title and item summary, and then combined it with PMF. Wang et al. [10] used SDAE to process the item title, item summary, and item plot information, and then combined SDAE with PMF. Although both methods alleviate the data sparsity and improve the accuracy of rating prediction, because of the limitations of the bag-of-words model, neither of them can capture the context information in the auxiliary data or the long-distance dependence between different components. Kim et al. [11] processed item comment information using a CNN, and then combined this with PMF. This method can capture the context information in the auxiliary data but cannot capture the long-distance dependence between different components of the auxiliary data. SAConvMF, proposed in this paper, overcomes the shortcomings of the above methods. When SAConvMF is used for recommendation, the item comment data, item name data, and item category data are integrated, which further alleviates the data sparsity and improves the accuracy of rating prediction. A theoretical comparison of the recommendation methods is shown in Table 2.
TABLE 2. Theoretical comparison of recommendation methods.

| Method         | Auxiliary data | Degree of mitigating data sparsity | Whether to capture context information | Whether to capture long-distance dependence | Prediction accuracy |
|----------------|----------------|-----------------------------------|---------------------------------------|---------------------------------------------|---------------------|
| BiasSD         | No             | No                                | No                                    | No                                          | Low                 |
| SVD++          | The hidden data of user behavior | Low                              | No                                    | No                                          | Low                 |
| PMF            | No             | No                                | No                                    | No                                          | Low                 |
| CTR            | The comment of movie | Low                              | No                                    | No                                          | Low                 |
| CDL            | The comment of movie | Low                              | No                                    | No                                          | Middle              |
| ConvMF         | The comment of movie | Low                              | Yes                                   | No                                          | Middle              |
| ConvMF++       | The comment, name and category of movie | High                             | Yes                                   | No                                          | Middle              |
| SAConvMF++     | The comment of movie | Low                              | Yes                                   | Yes                                         | High                |
| SAConvMF       | The comment, name and category of movie | High                             | Yes                                   | Yes                                         | High                |

TABLE 3. Movie comment data.

| Movie ID | Comment                                      | Notes                                      |
|----------|----------------------------------------------|--------------------------------------------|
| 1        | a little...andy davis...                     | A movie may have no comment or multiple comments |
| 10       | when a...1986 soviet...                     |                                           |
| 100      | a young...                                   |                                           |

TABLE 4. Movie name data.

| Movie ID | Name             | Notes                  |
|----------|------------------|------------------------|
| 1        | Toy Story (1995) | A movie has only one name |
| 10       | GoldenEye (1995) |                        |
| 100      | City Hall (1996) |                        |

contains 1,000,209 ratings by 6040 users on 3952 movies, and MovieLens-10M contains 10,000,054 ratings by 71,567 users on 65,133 movies, with the ratings ranging from 1 to 5. In the experiments, we used the auxiliary data of movies, including the movie comment data, movie name data, and movie category data. The formats are shown in Table 3, 4, and 5.

B. DATA PREPROCESSING

We use the data preprocessing method of [11] to preprocess the rating data, comment data, name data, and category data. The preprocessing flowchart is shown in Fig. 7.

After the data preprocessing, we obtain the movie auxiliary data X=[X1, X2, X3], where X1, X2, and X3 represent the movie comment code array, movie name code array, and movie category code array, respectively. Each row represents the comment code, name code, and category code of a movie. The formats are shown in Table 6, 7, and 8.

C. EVALUATION METHOD AND INDEX

1) EVALUATION METHOD

We use the evaluation method described in [11]. The preprocessed MovieLens-1M and MovieLens-10M datasets are divided into a training set, validation set, and test set in the ratio 8:1:1. The training set is used for model training, the validation set is used to select the model, and the test set is used to evaluate the model.
2) EVALUATION INDEX

In this paper, root mean square error (RMSE) is used as the index to evaluate the accuracy of rating prediction:

\[
RMSE = \sqrt{\frac{\sum_{i,j} (R_{ij} - \hat{R}_{ij})^2}{RatingNum_{\text{Test}}}},
\]

(27)

where, \(R_{ij}\) is the true rating of user i for movie j, \(\hat{R}_{ij}\) is user i’s predicted rating for movie j, and \(RatingNum_{\text{Test}}\) is the number of ratings in the test set.

D. COMPARISON ALGORITHMS AND PARAMETER SETTINGS

1) COMPARISON ALGORITHMS

We compare the BiasSVD, SVD++, PMF, CTR, CDL, ConvMF, and SAConvMF algorithms on the MovieLens-1M and MovieLens-10M datasets. These algorithms are as follows:

- BiasSVD [13]: the SVD recommendation algorithm that considers user and item bias.
- SVD++ [14]: the SVD recommendation algorithm that considers users’ implicit feedback.
- PMF [4]: the traditional PMF recommendation algorithm.
- CTR [9]: the collaborative topic regression recommendation algorithm, using only movie comment data.
- CDL [10]: the collaborative deep learning recommendation algorithm, using only movie comment data.
- ConvMF [11]: the convolutional matrix factorization recommendation algorithm, using only movie comment data.
- ConvMF+ [11]: the convolutional matrix factorization recommendation algorithm, using only movie comment data, and using the Glove preprocessing model for word vector assignment.

ConvMF++: the convolutional matrix factorization recommendation algorithm, incorporating the movie comment, movie name, and movie category data, and using Glove for word vector assignment.

SAConvMF-: the PMF recommendation algorithm of Self-Att-CNN, using only movie comment data, and using Glove for word vector assignment.

SAConvMF: the PMF recommendation algorithm of Self-Att-CNN, using movie comment, movie name, and movie category data, and using Glove for word vector assignment.

2) PARAMETER SETTINGS

In the experiments, the hidden vector dimension of U and V was set to 50. The regularization coefficients and learning rate (of both BiasSVD and SVD++) were 0.02 and 0.1, respectively, and the word vector’s dimension p was set to 200. The settings of the hyperparameters \(\lambda_U\) and \(\lambda_V\) of each algorithm are shown in Table 10, and the training parameter settings of the SAConvMF recommended algorithm are shown in Table 11.

E. EXPERIMENTAL RESULTS

Two groups of experiments were conducted. In the first group, SAConvMF and the comparison algorithms were tested on the preprocessed MovieLens-1M and MovieLens-10M datasets. The experimental results are shown in Table 12. In the second group, we divided the training set from the MovieLens-1M dataset in different proportions to study the performance of each algorithm under the conditions of varying data sparsity. The results are shown in Table 13 and Fig. 8.

F. ANALYSIS OF EXPERIMENTAL RESULTS

From Table 12, we can observe that the accuracy of the ConvMF recommendation algorithm is much better than that of the CDL and CTR recommendation algorithms. This shows that the CNN can compensate for the shortcoming of CDL and CTR, namely that the bag-of-words model cannot capture context information. The SAConvMF recommendation algorithm proposed in this paper adds a self-attention mechanism to the CNN, which can capture both the context information and the long-distance dependence between different
TABLE 11. Training parameter settings of SAConvMF.

| Model    | MaxHe r | p  | Count | k  | nc | Types of convolution kernels on comment data | Types of convolution kernels on name data | Types of convolution kernels on category data | Optimization algorithm | Activatio n algorithm | Dropout rate | Batch size | Epoch |
|----------|---------|----|-------|----|----|---------------------------------------------|------------------------------------------|-----------------------------------------|-----------------------|---------------------|-------------|------------|-------|
| SAConvMF | 200     | 200| 10    | 50 | 100| [3,4,5]                                      |                                           | [1,2,3]                                | Adam                  | PReLU, ReLU        | 0.2          | 128        | 5     |
| SAConvMF | 200     | 200| 10    | 50 | 100| [3,4,5]                                      |                                           | [1]                                | Adam                  | PReLU, ReLU        | 0.2          | 128        | 5     |

TABLE 12. Comparison of RMSE of different algorithms on different datasets.

| Model    | MovieLens-1M | MovieLens-10M |
|----------|---------------|---------------|
| BiasSVD  | 0.8968        | 0.8300        |
| SVD++    | 0.8890        | 0.8201        |
| PMF      | 0.8971        | 0.8311        |
| CTR      | 0.8969        | 0.8275        |
| CDL      | 0.8879        | 0.8186        |
| ConvMF   | 0.8531        | 0.7958        |
| ConvMF++ | 0.8549(base)  | 0.7930(base)  |
| ConvMF++ | 0.8473(0.9%)  | 0.7923(0.08%) |
| SAConvMF-| 0.8485(0.75%) | 0.7885(0.57%) |
| SAConvMF | 0.8464(0.99%) | 0.7846(1.06%) |

The proportion of the training set in the whole dataset (data density)

| Model | 20% (0.93%) | 30% (1.39%) | 40% (1.86%) | 50% (2.32%) | 60% (2.78%) | 70% (3.25%) | 80% (3.71%) |
|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| BiasSVD | 1.014       | 0.970       | 0.949       | 0.933       | 0.919       | 0.908       | 0.896       |
| SVD++  | 1.011       | 0.965       | 0.940       | 0.930       | 0.910       | 0.902       | 0.880       |
| PMF    | 1.016       | 0.971       | 0.949       | 0.935       | 0.919       | 0.908       | 0.897       |
| CTR    | 1.012       | 0.968       | 0.948       | 0.933       | 0.919       | 0.908       | 0.896       |
| CDL    | 1.004       | 0.963       | 0.937       | 0.921       | 0.906       | 0.897       | 0.887       |
| ConvMF | 0.974       | 0.933       | 0.906       | 0.889       | 0.872       | 0.867       | 0.853       |
| SAConvMF | 0.912       | 0.987       | 0.988       | 0.886       | 0.862       | 0.858       | 0.846       |

FIGURE 8. Influence of the proportion of the training set, on the MovieLens-1M dataset, for each algorithm.

TABLE 13. Influence of training set with different partition ratios on MovieLens-1M dataset ON each algorithm.

SAConvMF- reduced the RMSE by 0.75% and 0.57%, respectively, compared with ConvMF+, and SAConvMF reduced the RMSE by 0.1% and 1.11%, respectively, compared with ConvMF++.

From Table 12, Table 13, and Fig. 8, we can observe that, for the MovieLens-1M and MovieLens-10M datasets, ConvMF++ reduced the RMSE by 0.9% and 0.08%, respectively, compared with ConvMF+, and SAConvMF reduced the RMSE by 0.25% and 0.49%, respectively, compared with SAConvMF-. This shows that the integration of item name data and category data can further alleviate data sparsity and improve the recommendation accuracy. Moreover, when the proportion of the training set increases from 20% to 80% on the MovieLens-1M dataset, the prediction accuracy of all the recommendation algorithms improves. In general, SAConvMF > ConvMF > CDL > SVD++ > CTR > BiasSVD > PMF. Our recommendation algorithm, SAConvMF, increases the prediction accuracy from 0.79% to 6.37% as the training set’s proportion changes from 80% to 20%, compared with the ConvMF algorithm. This shows that our algorithm can achieve better prediction accuracy even in an environment with sparse data.
VI. CONCLUSION AND FUTURE WORK
In this paper, we propose a probabilistic matrix factorization model based on self-attention mechanism convolutional neural networks (SAConvMF), which overcomes the problem that ConvMF model can’t capture the long-distance dependence between different components of auxiliary data. Therefore, our model can’t only capture the context information of auxiliary data, but also capture the long-distance dependence between different components of auxiliary data. After integrating the auxiliary information of movie comment, movie name and movie category, the SAConvMF is used for recommendation to further alleviate the data sparsity. Finally, the experimental results on MovieLens-1M and MovieLens-10M data sets show that the proposed model has better prediction accuracy.

It is worth noting that our recommendation model SAConvMF doesn’t consider user’s auxiliary data and the time sequence of rating data. Therefore, in the next step, we will improve the model on the basis of these two points, so as to further alleviate the data sparsity and capture the change of user interest.

ACKNOWLEDGMENT
(Chenkun Zhang and Cheng Wang contributed equally to this work.)

REFERENCES
[1] Z. Sun, L. Han, W. Huang, X. Wang, X. Zeng, M. Wang, and H. Yan, “Recommender systems based on social networks,” J. Syst. Softw., vol. 99, pp. 109–119, Jan. 2015.
[2] J. Borràs, A. Moreno, and A. Valls, “Intelligent tourism recommender systems: A survey,” Expert Syst. Appl., vol. 41, no. 16, pp. 7370–7389, Nov. 2014.
[3] Z. D. Champiri, S. R. Shahamiri, and S. B. Salim, “A systematic review of scholar context-aware recommender systems,” Expert Syst. Appl., vol. 42, no. 3, pp. 1743–1758, Feb. 2015.
[4] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in Proc. Adv. Neural Inf. Process. Syst., vol. 20, 2011, pp. 1–8.
[5] J. Ha, C. Park, C. Park, and S. Park, “IMIPMF: Inferring miRNA-disease interactions using probabilistic matrix factorization,” J. Biomed. Inform., vol. 102, Feb. 2020, Art. no. 103358.
[6] Y. Wang, F. Liu, S.-T. Xia, and J. Wu, “Link sign prediction by variational Bayesian probabilistic matrix factorization with student-T prior,” Inf. Sci., vol. 405, pp. 175–189, Sep. 2017.
[7] K. Zhang, X. Yun, J. Liang, X.-Y. Zhang, C. Li, and B. Tian, “Retweeting behavior prediction using probabilistic matrix factorization,” in Proc. IEEE Symp. Comput. Commun. (ISCC), Jun. 2016, pp. 1185–1192.
[8] R. P. Adams, G. E. Dahl, and I. Murray, “Incorporating side information in probabilistic matrix factorization with Gaussian processes,” 2010, arXiv:1003.4944. [Online]. Available: http://arxiv.org/abs/1003.4944
[9] 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), San Diego, CA, USA, 2011, pp. 448–456.
[10] H. Wang, N. Wang, and D.-Y. Yeung, “Collaborative deep learning for recommender systems,” in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Sydney, NSW, Australia, 2015, pp. 1235–1244.
[11] 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., Boston, MA, USA, Sep. 2016, pp. 233–240.
[12] S. Zhang, W. Wang, J. Ford, F. Makedon, and J. Pearlman, “Using singular value decomposition approximation for collaborative filtering,” in Proc. 7th IEEE Int. Conf. E-Commerce Technol. (CEC). New York, NY, USA: IEEE, Jul. 2005, pp. 257–264.
[13] A. Paterek, “Improving regularized singular value decomposition for collaborative filtering,” in Proc. KDD Cup Workshop, 2007, pp. 5–8.
[14] Y. Koren, “Factorization meets the neighborhood: A multifaceted collaborative filtering model,” in Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD). New York, NY, USA: ACM, 2008, pp. 432–434.
[15] H. Al-Bashuir, M. Abdullateef, A. Ronali, and N. Binti, “A developed collaborative filtering similarity method to improve the accuracy of recommendations under data sparsity,” Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 4, pp. 135–142, 2018.
[16] U. Liji, Y. Chai, and J. Chen, “Improved personalized recommendation based on user attributes clustering and score matrix filling,” Comput. Standards Interface, vol. 57, pp. 59–67, Mar. 2018.
[17] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, “Recurrent recommender networks,” in Proc. 10th ACM Int. Conf. Web Search Data Mining (WSDM), Cambridge, U.K., 2017, pp. 495–503.
[18] A. Suglia, C. Greco, C. Musto, M. de Gemmis, P. Lops, and G. Semeraro, “A deep architecture for content-based recommendations exploiting recurrent neural networks,” in Proc. 25th Conf. User Modeling, Adaptation Personalization, Bratislava, Slovakia, Jul. 2017, pp. 202–211.
[19] Y. Xie, M.-S. Shang, and Y. Zhang, “TCR: Temporal-CNN for reviews based recommendation system,” in Proc. Int. Conf. Deep Learn. Technol. (ICDLT), Chongqing, China, 2018, pp. 71–75.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS), Long Beach, CA, USA, 2017, pp. 1–11.
[21] J. Wang, X. Yang, S. Xu, X. Yuan, and Z. Xie, “Text classification model based on convolutional neural network and self-attention mechanism,” J. Sichuan Univ. (Natural Sci. Ed.), vol. 57, no. 3, pp. 469–475, 2020.
[22] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2014, pp. 1532–1543.
[23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778.
[24] J. L. Bu, J. R. Kiros, and G. E. Hinton, “Layer normalization,” Statistics, vol. 1, p. 14, Jul. 2016.
[25] F. M. Harper and J. A. Konstan, “The MovieLens datasets: History and context,” ACM Trans. Interact. Intell. Syst., vol. 5, no. 4, pp. 1–19, Jan. 2016.

Chenkun Zhang was born in 1995. He received the degree in computer science and technology from Huaqiao University, China, in 2019, where he is currently pursuing the degree. His research interest includes recommendation algorithm.

Cheng Wang was born in 1984. He received the Ph.D. degree in mechanics from Xi’an Jiaotong University, China, in 2012. He is currently an Associate Professor with Huaqiao University, China. His research interests include signal processing and recommendation algorithm.