

Article

A Dual-Attention Autoencoder Network for Efficient Recommendation System

Chao Duan ¹, Jianwen Sun ², Kaiqi Li ² and Qing Li ^{2,*}

¹ National Engineering Research Center for E-Learning, Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China; duanchao@mail.ccnu.edu.cn

² National Engineering Laboratory for Educational Big Data, Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China; sunjw@ccnu.edu.cn (J.S.); couragelee@mails.ccnu.edu.cn (K.L.)

* Correspondence: viven_a@mail.ccnu.edu.cn

Abstract: Accelerated development of mobile networks and applications leads to the exponential expansion of resources, which causes problems such as trek and overload of information. One of the practical approaches to ease these problems is recommendation systems (RSs) that can provide individualized service. Video recommendation is one of the most critical recommendation services. However, achieving satisfactory recommendation service on the sparse data is difficult for video recommendation service. Moreover, the cold start problem further exacerbates the research challenge. Recent state-of-the-art works attempted to solve this problem by utilizing the user and item information from some other perspective. However, the significance of user and item information changes under different applications. This paper proposes an autoencoder model to improve recommendation efficiency by utilizing attribute information and implementing the proposed algorithm for video recommendation. In the proposed model, we first extract the user features and the video features by combining the user attribute and the video category information simultaneously. Then, we integrate the attention mechanism into the extracted features to generate the vital features. Finally, we incorporate the user and item potential factor to generate the probability matrix and defines the user-item rating matrix using the factorized probability matrix. Experimental results on two shared datasets demonstrates that the proposed model can effectively ameliorate video recommendation quality compared with the state-of-the-art methods.

Keywords: recommendation system; matrix factorization; attention mechanism; autoencoder; collaborative filtering



Citation: Duan, C.; Sun, J.; Li, K.; Li, Q. A Dual-Attention Autoencoder Network for Efficient Recommendation System. *Electronics* **2021**, *10*, 1581. <https://doi.org/10.3390/electronics10131581>

Academic Editors: Stefano Russo

Received: 28 April 2021

Accepted: 28 June 2021

Published: 30 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recommendation systems (RSs) [1,2], aiming at predicting the scores that users might give the items according to historical data, have been applied in many fields, such as social networking, news, information retrieval, courses [3], movies [4], music [5], and knowledge services [6]. However, with the rapid development of information science, more and more applications are continuously generating large-scale data, leading to an information explosion [7]. All these generated data can entail considerable convenience to our daily lives. Meanwhile, this phenomenon also leads to some problems, such as trek and overload of information. Moreover, how to extract valuable information from large-scale data quickly and efficiently becomes a significant challenge [8]. RSs [9] prove to be effective tools to extract useful information from the chaotic application data, and have attracted extensive attention from academics and the industry. As a result, a set of state-of-the-art research works have been proposed, and many practical applications have been developed. Collaborative filtering is one of the most widely used recommendation algorithms. The algorithm generates user groups close to the current user by analyzing their historical behavior data and then determines recommendation content on the basis of

the preferences of the current user group. Presently, RS has been successfully applied in many areas, including social networking, news, information retrieval, and e-commerce, and so on. With the rapid development of mobile networks and mobile services, the user and item data are split into different blocks and stored in different devices. However, most current studies focus on web-based applications with the data stored in the same database. How to design the recommendation algorithm for the split data scenario is a new challenge.

The shallow learning-based approach cannot catch the deep features of the user and items given that only one hidden level exists and relying on manual design features, traditional recommendation algorithms have limited manageability against the background of big data. With the commercialization of big data, cloud computing, and other technologies, more and more data are being recorded and stored on the Internet. These data include multiple sources of heterogeneous information, such as video [10], audio, text, graphics, and images [11]. They encapsulate rich user and item background information and individualized demands [12–14]. How to effectively alleviate the data sparseness and cold start problems by using the multi-source heterogeneous auxiliary information in the RS has currently become the hottest topic in academic research [15,16]. As a vital branch of artificial intelligence, deep learning technology has outstanding performance in various fields, such as image processing, NLP, and speech recognition, which introduces new commercialization directions for RS [17]. Deep learning can automatically extract features from the massive data of users and items, and obtain deep-level feature representations of users and items. Moreover, deep learning can also integrate multi-source heterogeneous data through feature fusion to obtain a decentralized feature representation [18]. When it is combined with traditional recommendation algorithms, recommendation quality can be ameliorated significantly. Google proposed the Wide & Deep model [19], which combined various linear models and deep neural network models to take the advantages of memory and generalization to make recommendations. Google's model has played a leading role. Recommendation algorithms based on autoencoders have also been extensively studied in the recommendation field. For example, AutoRec [20] is a classic application of early deep learning techniques in the recommendation field, which combines autoencoders and collaborative filtering to make recommendations. ACAE [21] is the latest application that uses an autoencoder fused with an attention mechanism to display recommendation applications, but it does not use auxiliary information. SASRec [22] is a sequence model based on self-attention. It considers the time correlation of the system to try to identify related items from the user's operation history so as to predict the next item. It has accomplished impressive results. In [23], Tal et al. proposed two models—DARIA and SARA— which use the user's personalized historical information and item attribute information to assist recommendation through a dual attention mechanism. Good experimental results were obtained.

In this research, the author puts forward a multi-head attention autoencoder matrix factorization (MAAMF) model and an autoencoder model multi-head attention autoencoder (MAA) fused with a multi-head self-attention mechanism to mine users and items auxiliary information, and it is combined with probability matrix factorization to realize rating prediction. Autoencoders with a dual-multi-head self-attention mechanism effectively extract the hidden factors of user attribute information and video category information and observe the vital information on the user side and the video side, thus improving the interpretability of the recommendation model based on deep learning. It incorporates probability matrix decomposition to achieve rating prediction, effectively alleviating data sparseness and cold start problems. The main contributions of this paper are summarized as follows:

- (1) It utilizes the user's attribute information and the category information of the video to reveal the potential factors of the user and the video successively and combine the probability matrix factorization to realize the rating prediction.
- (2) It explores the role of the self-attention mechanism and multi-head self-attention mechanism in mining hidden factors of side information.

- (3) We have carried out numerous experiments on two public datasets. We extracted features by only using user attribute information, through item category information, through the self-attention mechanism, and through multi-head self-attention mechanism using user and item side information simultaneously to illustrate the effectiveness of the proposed MAAMF model.

The remainder of this article is organized as follows: Section 2 reviews the related works about the RS. Section 3 introduces the proposed MAAMF model in detail. Section 4 presents and analyzes our experimental results. Finally, Section 5 concludes.

2. Related Work

2.1. Matrix Factorization

Matrix factorization [24] is the most frequently used recommendation algorithm. The famous Netflix Prize in the RS uses matrix factorization technology. Matrix factorization has attracted considerable attention due to its high recommendation accuracy, strong model scalability, and easy integration of side information. Common matrix factorizations include SVD [25] and NMF [26]. These methods achieve rating prediction by decomposing a high-dimensional sparse rating matrix into multiple low-dimensional matrices and multiplying multiple low-dimensional matrices to approximate the original scoring matrix. EigenRec [27] realizes top-N recommendations by constructing a low-dimensional representation of similarity between items and combining it with the classic PureSVD algorithm. In [28–30], Liu et al. discussed the process from traditional matrix factorization to modern RS using neural networks for multi-objective learning [31]. The abovementioned recommendation methods are based only on the user's attribute information or the category information of the videos. This paper aims to utilize the user attribute cues and videos category cue to predict the rating accurately, hence the self-attention and multi-head self-attention mechanisms are introduced to mine the hidden factors of side information from users and videos.

2.2. Collaborative Filtering

With the advancement of information technology, side material of users or videos, for instance, user or item attribute information, user portrait information, social network information, comment information, and so on, are recorded and stored. The features of users and videos or user preferences are hidden in these big data. How to ascertain useful information from these big data to ameliorate the accuracy of recommendation and reduce the problems of cold start and data sparsity is a current research hotspot. Numerous studies are applying text comment information to assist recommendation. For instance, ConvMF [32], DeepCoNN [33], D-Attn [34], ANR [35], CARL [36], DAML [37], and other models combine all comments of users or items into a long document and then build a model to assist recommendation. Meanwhile, NARRE [38], TARMF [39], and MPCN [40] build a model on the basis of every single piece of comment of the user or item to obtain the feature of each comment, which is then combined to form the features of the user or item to make recommendations. Moreover, many studies use social network information to assist recommendation, such as [41–44], or using social network and text information to assist recommendation, such as [14,45]. How to extract hidden information from large-scale data is the main challenge faced by RS. In [46], Anastasiu et al. put forward a good research plan on how to find the nearest neighbors from big data and how to expand the potential factors. In [47], Sardinios et al. and in [48], David et al. surveyed various aspects of large-scale social network-based RS, summarizing the challenges and attractive problems, and discussed specific solutions. Current research shows that the advantages of multi-task learning and ensemble learning are very obvious, and they are forceful means to solve multi-source information fusion problem [49,50].

However, in video recommendation, our intuition tells us that people of different genders, age groups, occupations, and regions will have distinct preferences. Similarly, many video websites practice classifying video information by category. One example the

fact that children like to watch cartoons is judged by categories. Everyone prefers one or more categories of video. In [51], Zhao et al. applied the recommendation model on the basis of deep learning to the field of industrial video recommendation. He proposed a large-scale multi-objective ranking system, and applied it to the next video recommendation. The experiment achieved good results. In [52], Chen et al. assisted memory-based collaborative filtering recommendation by fusing video category information, which ameliorates the accuracy, coverage, precision, and recall rate of prediction, but it failed to consider the impact of user side information on video recommendation. Utilizing the side information of users and videos simultaneously for ensemble learning is a vital method to ameliorate the accuracy of video recommendation. Recently, the knowledge graph [53,54] is also introduced as the side information for the recommendation algorithms, which has achieved significant success in the field of natural language processing. Although the existing studies have made some progress in RS, no work has considered the user's attribute information and the category information of the video in an online platform to predict rating.

3. Proposed MAAMF Model

This section first introduces the structure of the proposed deep hybrid rating prediction model MAAMF, which integrates the dual-multi-head self-attention mechanism. By using autoencoder with the multi-head self-attention mechanism, the potential factors of the user side and the project side are determined. The factors are then integrated into the probability matrix decomposition to achieve rating prediction. The principle of the MAAMF model is explained in detail subsequently. Finally, the MAAMF model is optimized through maximum a posteriori estimation. Table 1 summarizes all the symbols used in this article to clarify our method.

Table 1. Summary of symbols description.

| Notation | Meaning | Notation | Meaning |
|-------------|--------------------------------------|-----------------|-------------------------------------------------------|
| U | User side information | ϵ_j | Influence of Gaussian noise |
| V | Item side information | X | Side information |
| U' | approximate representation of U | S' | Unzipped secondary information |
| V' | approximate representation of V | X' | Unzipped rating information |
| W | Weight | h | Hidden layer |
| N | Number of users | L | Number of hidden layers |
| M | Number of items | h_1 | First hidden layer |
| R | User item rating matrix | W_1 | First layer of weight parameters |
| U_i | Side information of the i -th user | m_i | Times |
| V_j | Side information of the j -th item | Q_m | Query |
| α | Compressed representation of U_i | k_m | Key |
| β | Compressed representation of V_j | v_m | Values |
| λ_u | User regularization parameter | $W_{m_i}^{Q_m}$ | parameter matrix of Q_m in the m_i transformation |
| λ_v | Item regularization parameter | | |

3.1. MAAMF Model Structure

Figure 1 shows the structure of the deep hybrid rating prediction model MAAMF fused with the dual-multi-head self-attention mechanism. In the figure, N is the number of users and M is the number of videos, R is the user video rating matrix, and R_{ij} is the rating of the i -th user on the j -th video. U_i is the side information of the i -th user, such as the user's rating information and the user's attribute information. V_j shows the side information of the j -th item, such as the rating information of the video and the category information of the video. α is the compressed representation of U_i , and β is the compressed representation of V_j . λ_u is the user regularization parameter, λ_v is the project regularization parameter, U'_i and V'_j are the new approximate representations obtained by decompressing U_i and V_j through the autoencoder of the fusion multi-head attention mechanism, respectively.

The purpose of MAAMF is to integrate MAA into the PMF framework and achieve rating predictions through PMF and two MAAs.

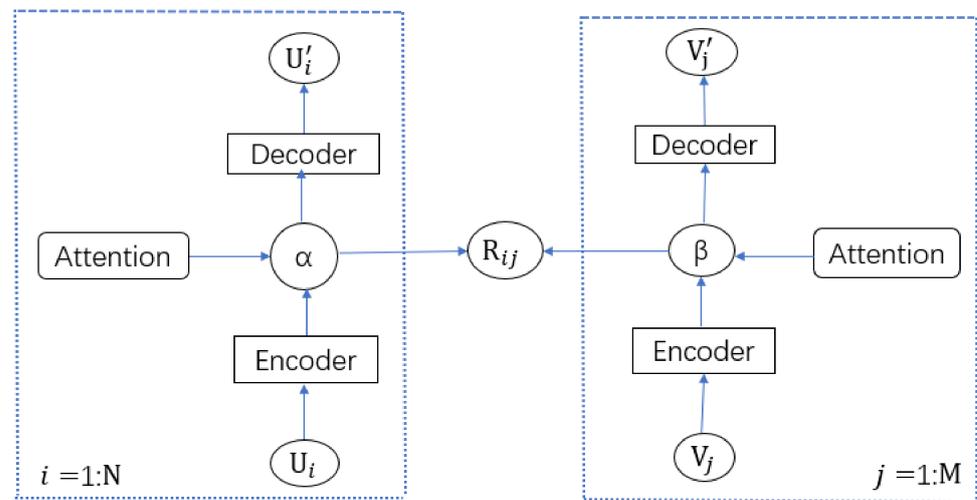


Figure 1. MAAMF model structure.

This study aims to identify the potential matrix of users and videos ($U \in R^{n \times k}$, $V \in R^{m \times k}$) $U^T V$ to rebuild the rating matrix R, and the observed rating matrix is calculated as follows:

$$p(R | U, V, \sigma^2) = \prod_i^n \prod_j^m N(R_{ij} | u_i^T v_j, \sigma^2)^{I_{ij}} \tag{1}$$

where $N(x | \mu, \sigma^2)$ is the probability density function of the Gaussian normal distribution with mean μ and variance σ^2 . I_{ij} demonstrates the indicator function. If user i evaluates the video j , then it is set to 1, otherwise, it is set to 0.

3.2. MAAMF Model Principle

The core of our MAAMF model is the MAA module. The purpose is to use two MAA modules to obtain the potential factors of users and videos from user side information and video side information, respectively. The MAA module is shown in Figure 2. In the figure, S represents side information, specifically user attribute information or video category information in this model. X represents user rating or item rating information. S' represents decompressed auxiliary information, and X' represents the scoring information after decompression. h represents the hidden layer, and L represents the number of hidden layers. h_1 represents the first hidden layer, $h(L/2)$ represents the $L/2$ -th hidden layer. W represents the weight parameter, and W_1 represents the first layer weight parameter, $W(L/2)$ represents the weight parameter of the $L/2$ -th layer.

The input layer uses the preprocessed tensor that is the side information as the input of the neural network. In this study, the user side information is the concatenation of the user rating matrix and the user characteristic information, and the video side information is the concatenation of the video rating matrix and the video classification information.

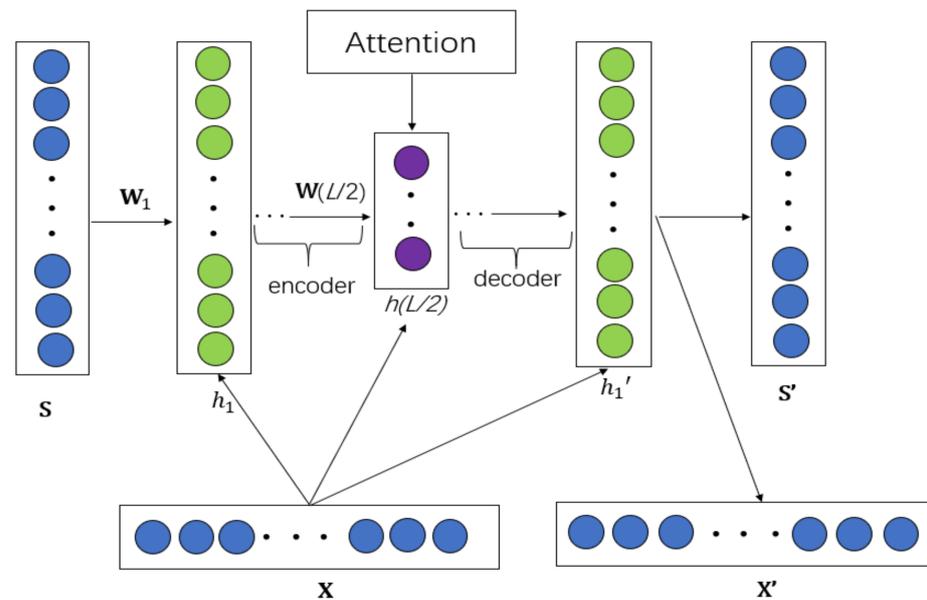


Figure 2. MAA module.

The coding layer contains four layers, which are compressed layer by layer from 400 to 200 to 100 to 50. After each layer passes through the fully connected layer, batch normalization is performed, and then activated by the relu activation function, and finally, dropout is performed to prevent overfitting.

The multi-head attention layer selects vital features. The multi-head attention mechanism [55] first conducts linear transformation, and then the outcome is input to the scaled dot-Product attention, which is done h times in total. One head for each time, so it becomes multi-head. m_i means the m_i time, Q_m means to query, k_m means key, v_m means the values. $W_{m_i}^{Q_m}$, $W_{m_i}^{k_m}$, and $W_{m_i}^{v_m}$ respectively represent the parameter matrix of Q_m , k_m , and v_m in the m th transformation, W_m^0 is the weight initialization parameter, as shown in Equations (2) and (3). The parameters are not shared among the heads, and then the h times of scaled dot-Product attention results are concatenated, and the value is calculated by performing linear transformation, which is used as the result of multi-head attention:

$$hd_{m_i} = \text{Attention}\left(Q_m W_{m_i}^{Q_m}, k_m W_{m_i}^{k_m}, v_m W_{m_i}^{v_m}\right) \tag{2}$$

$$\text{MultiHead}\left(Q_m, k_m, v_m\right) = \text{Concat}\left(hd_1, \dots, hd_h\right) W_m^0 \tag{3}$$

The multi-head attention mechanism is developed on the basis of self-attention. The calculation of self-attention involves the connection between one input and all other inputs, and the calculation process mainly involves matrix multiplication. The self-attention mechanism is to perform three linear transformations on the result matrix of the embedding layer to obtain three new matrices named query matrix Q_m , key matrix k_m , and value matrix v_m . The query matrix Q_m is used as the score corresponding to each key matrix k_m . The score and the key matrix k_m are multiplied to calculate the corresponding weight, which is then multiplied by the value matrix v_m . After performing a weighted average on the value matrix, we can finally obtain the scaled dot-Product attention. In practice, multiple parallel self-attention mechanisms are often used to form a multi-headed attention mechanism, that is, multiple attention matrices and multiple weights are used to weight the input values, and finally the weighted average results are concatenated. The self-attention mechanism can only capture the correlation between adjacent information. If multiple self-attention mechanisms are used, it can capture the correlation between relatively distant information. The final concatenation combines multiple attention mechanisms that can better describe the relationship between information at different distances.

The decoding layer corresponds to the coding layer, decompressing layer by layer from 50 to 100 to 200 to 400. Likewise, after each layer passes through the fully connected layer, batch normalization is performed, and then activated by the relu activation function, and finally, dropout is performed to prevent overfitting.

The probability density function of the zero-mean spherical Gaussian distribution above the user potential model with a difference of σ_U^2 is shown in Equation (4):

$$p(U | \sigma_U^2) = \prod_{i=1}^n N(u_i | 0, \sigma_U^2 I) \quad (4)$$

Different from the traditional PMF model, we assume that the user latent vector is affected by the MAA internal weight w^+ , A_i represents the ancillary information of the i -th user, and ϵ_i is the influence of Gaussian noise, and we obtain Equation (5) as follows:

$$u_i = \text{MAA}(W^+, A_i) + \epsilon_i \quad (5)$$

Using them to optimize the rating of the user's potential model, hence we obtain Equation (6):

$$P(W^+ | \sigma_{W^+}^2) = \prod_k N(w_k^+ | 0, \sigma_{W^+}^2) \quad (6)$$

where $\sigma_{W^+}^2$ is its variance, and k represents the dimension of the latent factor. The user potential model based on conditional distribution is written as,

$$p(U | W^+, A, \sigma_U^2) = \prod_i^n N(u_i | \text{MAA}(W^+, A_i), \sigma_U^2 I) \quad (7)$$

where A represents the side information of the user, the latent vector $\text{MAA}(W^+, A_i)$ obtained by the MAA module is used as the mean of the Gaussian distribution, the variance $\sigma_U^2 I$ is the variance of the Gaussian distribution, I is the identity matrix, and n is the user number.

Similarly, we can obtain the probability density function of the internal weight W of the item as shown in Equation (8). In addition, the probability density function of the item based on conditional distribution is shown in Equation (9), where σ_W^2 is its variance, and k represents the dimension of the potential factor, C represents the ancillary information of the video, the latent vector $\text{MAA}(W, C_j)$ obtained by the MAA module is used as the mean of the Gaussian distribution, the variance $\sigma_V^2 I$ is the variance of the Gaussian distribution, I is the identity matrix, and m is the item number:

$$P(W | \sigma_W^2) = \prod_k N(w_k | 0, \sigma_W^2) \quad (8)$$

$$p(V | W, C, \sigma_V^2) = \prod_j^m N(v_j | \text{MAA}(W, C_j), \sigma_V^2 I) \quad (9)$$

3.3. Model Optimization

The variables U , V , W^+ , and W are optimized by using maximum a posterior estimation. In Equation (10) A represents user side information, C represents project side information, and R represents predicted rating. U represents the user potential factor, V represents the project potential factor, and i and j represent the variables of the user and the project. n is the number of users, m is the number of videos, W is the internal weight of each model, and σ^2 is the variance of the corresponding Gaussian distribution:

$$\begin{aligned} & \max_{U, V, W^+, W} p(U, V, W^+, W | R, A, C, \sigma^2, \sigma_U^2, \sigma_V^2, \sigma_{W^+}^2, \sigma_W^2) \\ & = \max_{U, V, W^+, W} [p(R | U, V, \sigma^2) p(U | W^+, A, \sigma_U^2) p(W^+ | \sigma_{W^+}^2) p(V | W, C, \sigma_V^2) p(W | \sigma_W^2)] \end{aligned} \quad (10)$$

Taking the logarithm of both sides of Equation (10) we obtain Equation (11)

$$\begin{aligned} \ln p(U, V, W^+, W | R, A, C, \sigma^2, \sigma_U^2, \sigma_V^2, \sigma_{W^+}^2, \sigma_W^2) &= \ln p(R | U, V, \sigma^2) + \\ \ln p(U | W^+, A, \sigma_U^2) + \ln p(W^+ | \sigma_{W^+}^2) + \ln p(V | W, C, \sigma_V^2) + \ln p(W | \sigma_W^2) \end{aligned} \quad (11)$$

Substituting Equations (1), (6), (7) (8), (9) into Equation (11), and then making both sides negative we obtain Equation (12):

$$\begin{aligned} \mathcal{L}(U, V, W^+, W) &= \sum_i^n \sum_j^m \frac{I_{ij}}{2} (R_{ij} - u_i^T v_j)^2 + \frac{\lambda_U}{2} \sum_i^n \| u_i - MAA(W^+, A_i) \|_F^2 \\ &+ \frac{\lambda_V}{2} \sum_j^m \| v_j - MAA(W, C_j) \|_F^2 + \frac{\lambda_{W^+}}{2} \sum_k^{|w_k^+|} \| w_k^+ \|_2^2 + \frac{\lambda_W}{2} \sum_k^{|w_k|} \| w_k \|_2^2 \end{aligned} \quad (12)$$

We utilized the coordinate descent method to minimize \mathcal{L} , and iteratively optimize the latent variables while fixing remaining parameters. Equation (12) is the quadratic function of U . Assuming that V and W are constants, the loss function \mathcal{L} can be obtained by U_i differentiation. Take the same operation on V to obtain the following expressions:

$$u_i \leftarrow (VI_iV^T + \lambda_U I_K)^{-1} (VR_i + \lambda_U MAA(W^+, A_i)) \quad (13)$$

$$v_j \leftarrow (UI_jU^T + \lambda_V I_K)^{-1} (UR_j + \lambda_V MAA(W, C_j)) \quad (14)$$

where $I_i, I_j, I_K,$ and I_K are diagonal matrices, and λ_U and λ_V are balance parameters. Equation (13) shows the effect of updating u_i by the user potential vector $MAA(W^+, A_i)$ through λ_U , and Equation (14) shows the effect of updating v_j by the project potential vector $MAA(W, C_j)$ through λ_V . However, we cannot update W^+ and W similar to U and V , because W^+ and W are linked to the nonlinearity in the MAA architecture. When U and V are temporarily constant, the loss function \mathcal{L} can be referred to as the regular term weighted square error function of L_2 . We obtain $\mathcal{E}(W^+)$ and $\mathcal{E}(W)$ as given in Equations (15) and (16).

$$\mathcal{E}(W^+) = \frac{\lambda_U}{2} \sum_i^n \| u_i - MAA(W^+, A_i) \|_F^2 + \frac{\lambda_{W^+}}{2} \sum_k^{|w_k^+|} \| w_k^+ \|_2^2 + \text{constant} \quad (15)$$

$$\mathcal{E}(W) = \frac{\lambda_V}{2} \sum_j^m \| v_j - MAA(W, C_j) \|_F^2 + \frac{\lambda_W}{2} \sum_k^{|w_k|} \| w_k \|_2^2 + \text{constant} \quad (16)$$

By optimizing $U, V, W^+,$ and W , we can finally obtain the predicted user's rating Equation (17):

$$\hat{R}_{ij} = \mathbb{E} [R_{ij} | u_i^T v_j, \sigma^2] = u_i^T v_j = (MAA(W^+, A_i) + \epsilon_i)^T (MAA(W, C_j) + \epsilon_j) \quad (17)$$

where \hat{R}_{ij} is the expected value of rating.

4. Experimental Results and Discussion

In this section, two shared datasets from the video field are used to verify the proposed MAAMF model. First, the datasets and evaluation indicators are described. Then the environment and basic parameter settings of the experiment are introduced, and then the principle of the baseline model is briefly described. Finally, the best result was obtained through experimental adjustment, and the final result was studied and compared to the baseline model.

4.1. Datasets and Evaluation Indicators

To accurately evaluate the performance of the MAAMF model, experiments were conducted on the MovieLens shared dataset. The MovieLens dataset [56] is one of the most widely used open-source datasets in the field of video recommendation. Users' ratings of films range from 1 to 5. This experiment uses two subsets ML-100k and ML-1m, as shown in Table 2. The table presents that ML-100k is rating data in order of 0.1 million, and ML-1m is rating data in order of 1 million.

Table 2. Statistics of the two preprocessed datasets.

| Dataset | User side Information | Video Side Information | User Number | Video Number | Ratings |
|---------|-------------------------------|----------------------------|-------------|--------------|---------|
| ML-100K | ID/gender/age/ job/zipcode | Video category information | 943 | 1546 | 94,808 |
| ML-1m | ID/gender/age/ job/zipcode | Video category information | 6040 | 3544 | 993,482 |

Prediction accuracy measures the ability of a recommendation model to predict user behavior, and is the most vital offline evaluation indicator for recommendation algorithms. This research aims to ameliorate the accuracy of rating prediction. Root mean square error (RMSE) and mean absolute error (MAE) are used to evaluate the performance of the MAAMF model. For user U and video V in the dataset, R_{ij} is the actual rating data of user U_i for video V_j , and \hat{R}_{ij} is the predicted rating data given by the recommendation model, $|R_{ij}|$ is the number of ratings in the test set. The definitions are as (18) and (19), respectively:

$$\text{RMSE} = \sqrt{\frac{\sum_{U,V \in R_{ij}} (R_{ij} - \hat{R}_{ij})^2}{|R_{ij}|}} \quad (18)$$

$$\text{MAE} = \frac{\sum_{U,V \in R_{ij}} |R_{ij} - \hat{R}_{ij}|}{|R_{ij}|} \quad (19)$$

4.2. Experimental Software and Hardware Environment and Parameter Settings

This experiment was carried out on the integrated development platform PyCharm Community Edition 2020.1.1 x64 of Windows 10, using Python 3.7 language development, and the Keras 2.3.1 deep learning library. The equipment environment is an Intel Core i7 4790K 4.00 GHz CPU, equipped with a GeForce GTX 1080 GPU and 32 GB of RAM.

The project side has 18 categories, and an empty category is designed and added to form 19 categories. If a video does not have a category, the 19th column is 1, otherwise, it is marked 1 in the corresponding column. If a video belongs to more than 1 category, then 1 is marked on each of these corresponding columns to obtain the category representation matrix of the video. The auxiliary information on the user side uses the user's ID, gender, age, occupation, and postcode, which are transformed into a vector through one-hot encoding. Training data recorded for 80%, cross-validation data recorded for 10%, and test data recorded for 10%. The dimension of the hidden factor is 50.

4.3. Baseline Models

Six well-known baseline models, including PMF, SVD, aSDAE, R-ConvMF, PHD, and DUPIA are compared to the MAAMF model proposed in this study. They are briefly described as follows:

PMF [24]: Probability matrix decomposition is a widely known collaborative filtering model. It explains the rating prediction task from the perspective of probability. It assumes that the rating obeys a Gaussian distribution.

SVD [25]: Singular value decomposition model is a prominent matrix factorization model, which uses rating information to achieve rating prediction.

aSDAE [57]: aSDAE is a two-tower model, which extracts the hidden factors of the auxiliary information on the user side and the project side through two SDAEs. The item side information uses the word bag model as the input of the deep learning model.

R-ConvMF [58]: Convolutional matrix factorization is a text context-aware hybrid recommendation prototype. It uses item statistics that capture contextual information, integrates CNN into the PMF framework, and considers Gaussian noise in different ways.

PHD [59]: PHD is a new hybrid model that uses an autoencoder to extract user-side hidden factors. Meanwhile, it uses a convolutional neural network to extract item-side text hidden factors, which is then integrated into the PMF framework to achieve rating prediction.

DUPIA [60]: The DUPIA model is based on the PHD model and adds a self-attention to the CNN on the item side.

4.4. Analysis of Experimental Results

In this part, we first explored the influence of different parameters on the experimental results, and then compared the MAAMF model we proposed to the baseline models.

4.4.1. Influence of Balance Parameters on Experimental Results

Tables 3 and 4 test the changes of RMSE and MAE on the two shared datasets under different combinations of λ_u and λ_v , and the best results are shown in bold. The tables exhibit that when $\lambda_u = 8$, $\lambda_v = 50$ on ML-100k and $\lambda_u = 3$, $\lambda_v = 120$ on ML-1m, the MAAMF model achieves the best results. It indicates that the appropriate combination of λ_u and λ_v plays a vital role in balancing user- and project-side information. Adjusting the λ_u and λ_v hyperparameters has a vital influence on RMSE and MAE.

Table 3. Influence of λ_u and λ_v on RMSE and MAE on ML-100k.

| ML-100k | | | | | |
|-------------|--------|--------|--------|--------|--------|
| λ_u | 1 | 5 | 8 | 10 | 10 |
| λ_v | 100 | 50 | 50 | 50 | 60 |
| RMSE | 0.9320 | 0.9050 | 0.8900 | 0.9043 | 0.9051 |
| MAE | 0.7280 | 0.7095 | 0.6985 | 0.7095 | 0.7101 |

Table 4. Influence of λ_u and λ_v on RMSE and MAE on ML-1m.

| ML-1m | | | | | |
|-------------|--------|--------|--------|--------|--------|
| λ_u | 1 | 3 | 3 | 3 | 5 |
| λ_v | 100 | 100 | 150 | 120 | 120 |
| RMSE | 0.8697 | 0.8434 | 0.8416 | 0.8399 | 0.8429 |
| MAE | 0.6787 | 0.6594 | 0.6599 | 0.6574 | 0.6601 |

4.4.2. Influence of Different Dropouts on the Experimental Results

Dropout is one of the most effective and commonly used regularization methods in neural networks. It reduces the risk of overfitting by randomly discarding some layer output features. Figure 3a,b present that when the dropout is 0.2, the ML-100k achieves the optimal MAE value of 0.6985, and when the dropout is 0.4, the ML-1m achieves the optimal MAE value of 0.6574. To reduce overfitting, dropout regularization significantly improved the model's generalization ability.



Figure 3. (a) Description of MAE values under different dropouts on the ML-100k dataset; (b) Description of MAE values under different dropouts on the ML-1m dataset.

4.4.3. Changes of RMSE Value with the Number of Iterations

In the experiment, we assume that when the RMSE value of the test set increases more than five times during the iteration process, the training is terminated. Figures 4 and 5 show that the RMSE values of the cross-validation and the test set are very close. The RMSE value of the training set increased once before decreasing continuously. The RMSE value on the cross-validation set and the test set initially decreased and then slowly increased.

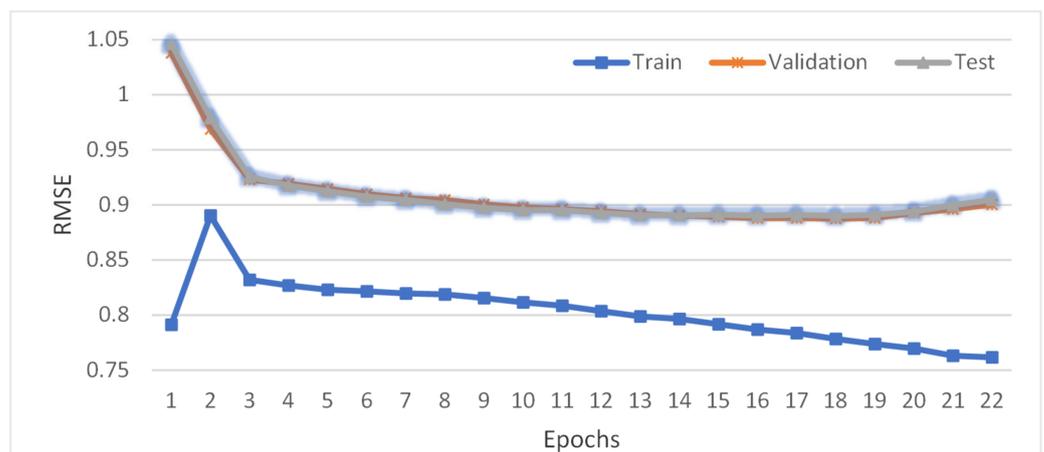


Figure 4. Changes of the RMSE of the training set, cross-validation set, and test set under ML-100k with the number of iterations.

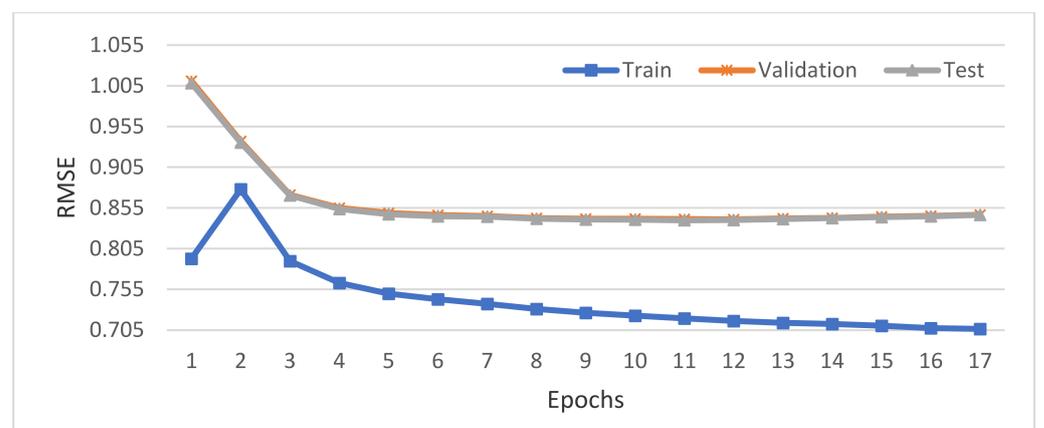


Figure 5. Changes of the RMSE of the training set, cross-validation set, and test set under ML-1m with the number of iterations.

4.4.4. Total Results of the Experiment

Table 5 shows the RMSE results of the proposed model and five well-known RS models on two shared datasets, where MAAMF-U in the table indicates that only user attribute information is used to assist recommendation, and MAAMF-I indicates that only item category information is used to assist recommendation. MAAMF-UI-S represents a user and item information-assisted recommendation through the fusion of self-attention mechanism, and MAAMF-UI-M represents user and item information assisted recommendation through the fusion of multi-head self-attention mechanism. The results of SVD and aSDAE in the table reveal that if there is little hidden information in the auxiliary information, or the training effect of deep learning is not good, the classic traditional recommendation models may outperform the recommendation models based on deep learning. The comparison of the experimental results of MAAMF-UI-S and MAAMF-UI-M with the results of other models shows that the vital information in the auxiliary information can be paid attention to by fusing the attention mechanism.

Table 5. RMSE result of different models in the two shared datasets. Results of the state-of-the-art models are underlined, and the optimal result of our model is marked in bold.

| Models | Datasets | |
|-------------------|---------------|---------------|
| | ML-100k | ML-1m |
| PMF | 0.9412 | 0.8971 |
| SVD | 0.9453 | 0.8684 |
| aSDAE | 0.9441 | 0.8723 |
| R-ConvMF | 0.9447 | 0.8470 |
| PHD | 0.9296 | 0.8465 |
| DUPIA | <u>0.9138</u> | <u>0.8439</u> |
| MAAMF-U | 0.9167 | 0.8544 |
| MAAMF-I | 0.9053 | 0.8509 |
| MAAMF-UI-S | 0.8822 | 0.8412 |
| MAAMF-UI-M | 0.8900 | 0.8399 |

Compared with the most advanced model DUPIA, the RMSE value of MAAMF-UI-S on the ML-100k dataset has improved by 3.46%, whereas the performance of DUPIA on the same dataset has improved by 1.70% compared with PHD, which is lower than the improvement of MAAMF-UI-S. The RMSE value of MAAMF-UI-M on the ML-1m dataset has improved by 0.47% when compared with DUPIA, which has an improvement of 0.31% on the same dataset when compared with PHD, lower than MAAMF-UI-M's improvement. The comparison of experimental results of MAAMF-U and MAAMF-I shows that different auxiliary information affects the experimental results distinctly. In this study, the recommendation assisted with item category information achieves a better result than the recommendation assisted with user attributes. Compared with MAAMF-U, the RMSE value of MAAMF-I on the ML-100k dataset has improved by 1.24%.

Among the well-known recommendation algorithms, the PHD and DUPIA model that uses user and project auxiliary information simultaneously has the same design thought as MAAMF-UI-S and MAAMF-UI-M. It has an outstanding performance on ML-1m with a large amount of data, which shows that integrating multi-source information of users and projects simultaneously and expanding the number of samples in the training dataset are the future research directions. Given that the RMSE value of MAAMF-UI-S on ML-100k is lower than that of MAAMF-UI-M may result from the relatively small scale of the ML-100k dataset, and the deep learning algorithm shows instability on small-scale datasets. When PHD and MAAMF are compared, the optimal deep learning based model and the optimal neural attention mechanism based model respectively, the RMSE values of the latter are respectively 5.10% and 0.78% better than the former.

In addition, we performed a one-tailed paired t-test test on the results of the DUPIA model and the MAAMF-UI-M model under different training data percentages in the ML-

1m dataset. We found that there is a significant difference between the results of the DUPIA model and the MAAMF-UI-M model, with the p-value of the t-test reported in Table 6. The experimental results prove that our model outperforms the state-of-the-art model.

Table 6. The p-value of the one-tailed paired t-test between the DUPIA model and the MAAMF-UI-M model.

| Models | 20% | 30% | 40% | 50% | 60% | 70% | 80% | t | sig |
|------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| DUPIA | 0.9195 | 0.8968 | 0.8829 | 0.8694 | 0.8598 | 0.8470 | 0.8439 | 3.7520 | 0.0047 |
| MAAMF-UI-M | 0.9180 | 0.8848 | 0.8720 | 0.8589 | 0.8519 | 0.8466 | 0.8399 | | |

The overall results of the experiment show that user attribute information and video category information impact the score prediction results. Mining the potential factors on the user side and the potential factors on the item side simultaneously can more effectively assist the recommendation. In addition, importance of the information in the auxiliary information varies, and more attention on more important information will improve recommendation outcome. This provides support for future research on constructing a unified multi-source information fusion framework.

4.4.5. Comparison of Experimental Results under Different Percentages

Finally, the effect of different training data percentages on the RMSE results is also studied. The results are shown in Figure 6. The figure reveals that the proposed MAAMF model obtains the best result except for 20% training data. In the figure, except for the PMF and SVD models, the other models are applications of deep learning in the RS. The results in the figure show that recommendation models combined with deep learning achieve better results than the traditional recommendation models, but the auxiliary information source and the algorithm used to extract potential factors significantly impact the experimental results. The figure also presents that for the same model, as the percentage of training data increases, the result of RMSE gradually decreases, and the accuracy of rating prediction gradually increases. The RMSE curves of the aSDAE and R-ConvMF models cross, and the RMSE curves of the PHD and MAAMF-UI-M models cross, indicating that the robustness of the recommendation model based on deep learning must be ameliorated.

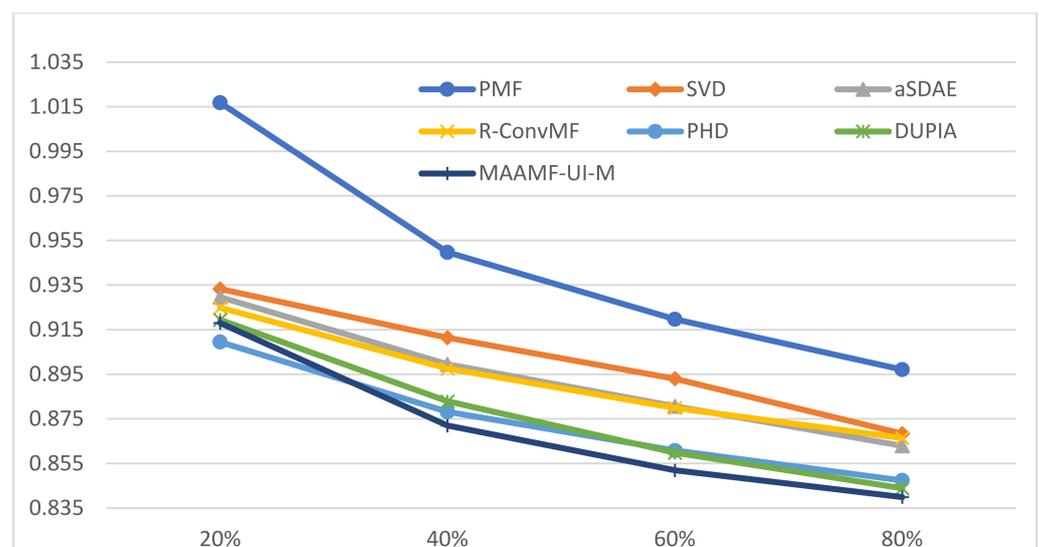


Figure 6. Comparison of the influence of different training data percentages of different models on RMSE results.

In this experiment, we discussed the influence of the regularization parameters λ_u , λ_v , and dropout on the experimental results, and described the changes in the RMSE values of the training set, cross-validation set, and test set under different iteration times. A compar-

ative study between the model proposed in this study and the five well-known baseline models reveals the feasibility of the proposed MAAMF model. The performance of different models is compared through experimental profits under different training data percentages. PMF and SVD are classic collaborative filtering recommendation algorithms. aSDAE, R-ConvMF, and PHD are recommendation algorithms based on deep learning. DUPIA and MAAMF are recommendation algorithms based on neural attention mechanism.

The overall results of the experiments and the comparison results under different percentages reveal that these models are constantly developing and improving, from solely using scoring data to using scoring data and auxiliary information, from using auxiliary information to paying more attention to important information in auxiliary information. The experiment unveils that the use of side details can ameliorate the accuracy of rating prediction, and different side information significantly impacts the prediction results. Making full and effective use of multi-source heterogeneous data can better assist recommendations. In addition, the success of MAAMF demonstrates the importance of the user's attribute information and the category information of the video in the RS, which was ignored by previous research.

5. Conclusions

In this study, we proposed an autoencoder model fused with a multi-head self-attention mechanism to mine user and item auxiliary information combined with probability matrix decomposition to achieve rating prediction. The autoencoder model fused with the dual multi-head self-attention mechanism effectively extracts the hidden factors of the user's basic attribute information and the item category information and pays attention to the vital information on the user side and the item side. It ameliorates the efficiency of the recommendation model on the basis of deep learning. The autoencoder model is combined with probability matrix decomposition to achieve rating prediction. Experimental results on two shared datasets of ML-100k and ML-1m demonstrate the usefulness of the proposed MAAMF model. Exploring more hidden factors in auxiliary information is a vital way to ameliorate the quality of recommendation. Furthermore, the time dimension of auxiliary information is also a vital factor affecting recommendation. For example, most people prefer to pay attention to the latest movies. How to better utilize time details is the focus of future research. We have also noticed that some auxiliary information will also influence one another. Ultimately, future studies must examine that how to effectively integrate the features of each auxiliary information in the RS.

Author Contributions: Conceptualization, Q.L., C.D.; methodology, Q.L.; software, C.D., K.L.; validation, C.D., J.S. and Q.L.; formal analysis, J.S., Q.L.; investigation, C.D.; resources, J.S., K.L.; data curation, J.S., K.L.; writing—original draft preparation, C.D.; writing—review and editing, Q.L.; visualization, C.D.; supervision, J.S.; project administration, Q.L.; funding acquisition, J.S., Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This paper financially is supported by the National Natural Science Foundation of China (61807012, 62077021, 61807013), the Humanity and Social Science Youth Foundation of Ministry of Education of China (20YJC880083), the Strategic research projects of Ministry of Education of China (2020JYKX04), and the Fundamental Research Funds for the Central Universities (CCNU20QN027, CCNU20ZN007).

Institutional Review Board Statement: The presented research was conducted according to the code of ethics of University of Minnesota.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data Availability Statement: Data of our study is available upon request.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Ban, Y.; Lee, K. How the Multiplicity of Suggested Information Affects the Behavior of a User in a Recommender System. *Electronics* **2021**, *10*, 741. [\[CrossRef\]](#)
2. Krichene, W.; Rendle, S. On Sampled Metrics for Item Recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA, 6–10 July 2020*; ACM: New York, NY, USA, 2020; pp. 1748–1757. [\[CrossRef\]](#)
3. Lin, Y.; Feng, S.; Zeng, W.; Lin, F.; Liu, Y.; Wu, P. Adaptive course recommendation in MOOCs. *Knowl. Based Syst.* **2021**, *224*, 107085. [\[CrossRef\]](#)
4. Jin, J.; Qin, J.; Fang, Y.; Du, K.; Zhang, W.; Yu, Y.; Zhang, Z.; Smola, A.J. An Efficient Neighborhood-based Interaction Model for Recommendation on Heterogeneous Graph. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, CA, USA, 6–10 July 2020*; ACM: New York, NY, USA, 2020; pp. 75–84. [\[CrossRef\]](#)
5. Paul, D.J.; Kundu, S. A survey of music recommendation systems with a proposed music recommendation system. In *Emerging Technology in Modelling and Graphics. Advances in Intelligent Systems and Computing*; Springer: New York, NY, USA, 2020; Volume 937. [\[CrossRef\]](#)
6. Li, Z.; Liu, H.; Zhang, Z.; Liu, T.; Xiong, N.N. Learning Knowledge Graph Embedding with Heterogeneous Relation Attention Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *10*, 1–13.
7. Warren, J. *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*; Manning Publications: Shelter Island, NY, USA, 2015.
8. Adomavicius, G.; Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 734–749. [\[CrossRef\]](#)
9. Li, D.; Liu, H.; Zhang, Z.; Lin, K.; Fang, S.; Li, Z.; Xiong, N.N. CARM: Confidence-aware recommender model via review representation learning and historical rating behavior in the online platforms. *Neurocomputing* **2021**, *455*, 283–296. [\[CrossRef\]](#)
10. Liu, T.; Li, Y.; Liu, H.; Zhang, Z.; Liu, S. RISIR: Rapid Infrared Spectral Imaging Restoration Model for Industrial Material Detection in Intelligent Video Systems. *IEEE Trans. Ind. Inf.* **2019**, *1*. [\[CrossRef\]](#)
11. Liu, T.; Liu, H.; Li, Y.; Zhang, Z.; Liu, S. Efficient Blind Signal Reconstruction with Wavelet Transforms Regularization for Educational Robot Infrared Vision Sensing. *IEEE/ASME Trans. Mechatron* **2019**, *24*, 384–394. [\[CrossRef\]](#)
12. Sun, Z.; Guo, Q.; Yang, J.; Fang, H.; Guo, G.; Zhang, J.; Burke, R. Research commentary on recommendations with side information: A survey and research directions. *Electron. Commer. Res. Appl.* **2019**, *37*, 100879. [\[CrossRef\]](#)
13. Ravanifard, R.; Buntine, W.; Mirzaei, A. Recommending content using side information. *Appl. Intell.* **2021**, *51*, 3353–3374. [\[CrossRef\]](#)
14. Liu, T.; Wang, Z.; Tang, J.; Yang, S.; Huang, G.Y.; Liu, Z. Recommender systems with heterogeneous side information. In *Proceedings of the World Wide Web Conference (WWW '19), San Francisco, CA, USA, 13–17 May 2019*; pp. 3027–3033.
15. Zhang, S.; Yao, L.; Sun, A.; Tay, Y. Deep Learning Based recommender system: A survey and new perspectives. *ACM Comput. Surv.* **2019**, *52*, 1–38. [\[CrossRef\]](#)
16. Batmaz, Z.; Yurekli, A.; Bilge, A.; Kaleli, C. A review on deep learning for recommender systems: Challenges and remedies. *Artif. Intell. Rev.* **2019**, *52*, 1–37. [\[CrossRef\]](#)
17. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
18. Zhang, G.; Liu, Y.; Jin, X. A survey of autoencoder-based recommender systems. *Front. Comput. Sci.* **2020**, *14*, 430–450. [\[CrossRef\]](#)
19. Cheng, H.T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispit, M.; et al. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016*; ACM: New York, NY, USA, 2016; pp. 7–10.
20. Sedhain, S.; Menon, A.K.; Sanner, S.; Xie, L. AutoRec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web, Florence, Italy, 18–22 May 2015*; pp. 111–112.
21. Chen, S.; Wu, M. Attention collaborative autoencoder for explicit recommender systems. *Electronics* **2020**, *9*, 1716. [\[CrossRef\]](#)
22. Kang, W.C.; McAuley, J. Self-attentive sequential recommendation. In *Proceedings of the 2018 IEEE Conference on Data Mining (ICDM), Singapore, 17–20 November 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 197–206.
23. Tal, O.; Liu, Y.; Huang, J.; Yu, X.; Aljbawi, B. Neural Attention Frameworks for Explainable Recommendation. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 2137–2150. [\[CrossRef\]](#)
24. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [\[CrossRef\]](#)
25. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. Application of dimensionality reduction in recommender system—A case study. In *Proceedings of the WebKDD-2000 Workshop, Boston, MA, USA, 20 August 2000*.
26. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [\[CrossRef\]](#)
27. Nikolakopoulos, A.N.; Kalantzis, V.; Gallopoulos, E.; Garofalakis, J.D. Eigenrec: Generalizing puresvd for effective and efficient top-n recommendations. *Knowl. Inf. Syst.* **2018**, *58*, 59–81. [\[CrossRef\]](#)
28. Koutrika, G. Modern recommender systems: From computing matrices to thinking with neurons. In *Proceedings of the International Conference on Management of Data, Houston, TX, USA, 10–15 June 2018*; pp. 1651–1654.
29. Shen, X.; Yi, B.; Liu, H.; Zhang, W.; Zhang, Z.; Liu, S.; Xiong, N. Deep variational matrix factorization with knowledge embedding for recommendation system. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 1906–1918. [\[CrossRef\]](#)

30. Yi, B.; Shen, X.; Liu, H.; Zhang, Z.; Zhang, W.; Liu, S.; Xiong, N. Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Trans. Knowl. Data Eng.* **2019**, *15*, 4591–4601. [[CrossRef](#)]
31. Liu, T.; Liu, H.; Li, Y.; Chen, Z.; Zhang, Z.; Liu, S. Flexible FTIR Spectral Imaging Enhancement for Industrial Robot Infrared Vision Sensing. *IEEE Trans. Ind. Inf.* **2020**, *16*, 544–554. [[CrossRef](#)]
32. Kim, D.; Park, C.; Oh, J.; Lee, S.; Yu, H. Convolutional matrix factorization for document context-aware recommendation. In Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, 15–19 September 2016; ACM: New York, NY, USA, 2016; pp. 233–240.
33. Zheng, L.; Noroozi, V.; Yu, P.S. Joint deep modeling of users and items using reviews for recommendation. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, Cambridge, UK, 6–10 February 2017; ACM: New York, NY, USA, 2017; pp. 425–434.
34. Seo, S.; Huang, J.; Yang, H.; Liu, Y. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In Proceedings of the Eleventh ACM Conference, Como, Italy, 27–31 August 2017; ACM: New York, NY, USA, 2017; pp. 297–305.
35. Wu, L.; Quan, C.; Li, C.; Wang, Q.; Zheng, B. A context-aware user-item representation learning for item recommendation. *IEEE Trans. Knowl. Data Eng.* **2017**, *37*, 1–29. [[CrossRef](#)]
36. Chin, J.Y.; Zhao, K.; Joty, S.; Cong, G. ANR: Aspect-based neural recommender. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM, Torino, Italy, 22–26 October 2018*; ACM: New York, NY, USA, 2018; pp. 147–156.
37. Liu, D.; Li, J.; Du, B.; Chang, J.; Gao, R. DAML: Dual Attention Mutual Learning between Ratings and Reviews for Item Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019*; ACM: New York, NY, USA, 2019; pp. 344–352.
38. Chong, C.; Min, Z.; Liu, Y.; Ma, S. Neural attentional rating regression with review-level explanations. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1583–1592.
39. Lu, Y.; Dong, R.; Smyth, B. Coevolutionary recommendation model: Mutual learning between ratings and reviews. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 773–782.
40. Yi, T.; Luu, A.T.; Hui, S.C. Multi-pointer co-attention networks for recommendation. In Proceedings of the 24th ACM SIGKDD International Conference, London, UK, 19–23 August 2018; ACM: New York, NY, USA, 2018; pp. 2309–2318.
41. Huang, X.; Liao, G.; Xiong, N.; Vasilakos, A.; Lan, T. A survey of context-aware recommendation schemes in event-based social networks. *Electronics* **2020**, *9*, 1583. [[CrossRef](#)]
42. Guo, G.; Zhang, J.; Yorke-Smith, N. TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 123–129.
43. Zhu, Z.; Wang, J. James Caverlee. In Improving top-K recommendation via joint collaborative autoencoders. In Proceedings of the 30th International Conference on World Wide Web, San Francisco, CA, USA, 13–17 May 2019; pp. 3482–3483.
44. Park, C.; Kim, D.H.; Oh, J.; Yu, H. TRecSo: Enhancing top-k recommendation with social information. In Proceedings of the WWW (Companion Volume), Montréal, QC, Canada, 11–15 April 2016; pp. 89–90.
45. Chen, C.; Zheng, X.; Yan, W.; Hong, F.; Zhen, L. Context-ware collaborative topic regression with social matrix factorization for recommender systems. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14), Québec City, QC, Canada, 27–31 July 2014*; AAAI Press: Palo Alto, CA, USA, 2014; pp. 9–15.
46. Anastasiu, D.C.; Christakopoulou, E.; Smith, S.; Sharma, M.; Karypis, G. Big Data and Recommender Systems. 2016. Available online: <https://conservancy.umn.edu/handle/11299/215998> (accessed on 2 June 2021).
47. Sardianos, C.; Tzirakis, N.; Varlamis, I. *A Survey on the Scalability of Recommender Systems for Social Networks*; Springer: Cham, Switzerland, 2018. [[CrossRef](#)]
48. Eirinaki, M.; Gao, J.; Varlamis, I.; Tserpes, K. Recommender Systems for Large-Scale Social Networks: A review of challenges and solutions. *Future Gener. Comput. Syst. FGCS* **2018**, *78*, 413–418. [[CrossRef](#)]
49. Zhu, N.; Cao, J.; Lu, X.; Gu, Q. Leveraging pointwise prediction with learning to rank for top-N recommendation. *World Wide Web* **2021**, *24*, 375–396. [[CrossRef](#)]
50. Liu, H.; Fang, S.; Zhang, Z.; Li, D.; Lin, K.; Wang, J. MFDNet: Collaborative Poses Perception and Matrix Fisher Distribution for Head Pose Estimation. *IEEE Trans. Multimed.* **2021**, *1*. [[CrossRef](#)]
51. Zhao, Z.; Chi, E.; Hong, L.; Chen, J.; Nath, A.; Andrews, S.; Kumthekar, A.; Sathiamoorthy, M.; Yi, X.; Chi, E. Recommending what video to watch next: A multitask ranking system. In Proceedings of the 13th ACM Conference, Copenhagen, Denmark, 16–20 September 2019; ACM: New York, NY, USA, 2019; pp. 43–51.
52. Chen, L.; Yuan, Y.; Yang, J.; Zahir, A. Improving the prediction quality in memory-based collaborative filtering using categorical features. *Electronics* **2021**, *10*, 214. [[CrossRef](#)]
53. Zhang, Z.; Li, Z.; Liu, H.; Xiong, N.N. Multi-scale Dynamic Convolutional Network for Knowledge Graph Embedding. *IEEE Trans. Knowl. Data Eng.* **2021**, 1–10. [[CrossRef](#)]
54. Li, Z.; Liu, H.; Zhang, Z.; Liu, T.; Shu, J. Recalibration Convolutional Networks for Learning Interaction Knowledge Graph Embedding. *Neurocomputing* **2021**, *427*, 118–130. [[CrossRef](#)]

55. Vaswani, A.; Shazeer, N.; Parmar, N. Attention is all you need. In Proceedings of the Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
56. Harper, F.M.; Konstan, J.A. The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* **2016**, *5*, 1–19. [[CrossRef](#)]
57. Dong, X.; Yu, L.; Wu, Z. A hybrid collaborative filtering model with deep structure for recommender systems. In *AAAI*; AAAI Press: San Francisco, CA, USA, 2017; pp. 1309–1315.
58. Kim, D.; Park, C.; Oh, J.; Yu, H. Deep hybrid recommender systems via exploiting document context and statistics of items. *Inf. Sci.* **2017**, *417*, 72–87. [[CrossRef](#)]
59. Liu, J.; Wang, D.; Ding, Y. PHD: A probabilistic model of hybrid deep collaborative filtering for recommender systems. In Proceedings of the Asian Conference on Machine Learning, Seoul, Korea, 15–17 November 2017; PMLR: Seoul, Korea; Volume 77, pp. 224–239.
60. Xz, A.; Hl, A.; Xc, B.; Zhong, J.; Wang, D. A novel hybrid deep recommendation system to differentiate user’s preference and item’s attractiveness. *Inf. Sci.* **2020**, *519*, 306–316.