electronics

IMPACT FACTOR 2.6

CITESCORE 5.3

Article

Improving the Performance of Cold-Start Recommendation by Fusion of Attention Network and Meta-Learning

Shilong Liu, Yang Liu, Xiaotong Zhang, Cheng Xu, Jie He and Yue Qi

Special Issue
Advances in Spatiotemporal Data Management and Analytics

Edited by
Prof. Dr. Yanwei Yu, Dr. Zhu Xiao and Dr. Ziqiang Yu

MDPI

https://doi.org/10.3390/electronics12020376 electronics

MDPI

Article

Improving the Performance of Cold-Start Recommendation by Fusion of Attention Network and Meta-Learning

Shilong Liu [1,2,3], Yang Liu [1], Xiaotong Zhang [1,2], Cheng Xu [1,2], Jie He [1,2,*] and Yue Qi [1,2,*]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2] Beijing Advanced Innovation Center for Materials Genome Engineering, University of Science and Technology Beijing, Beijing 100083, China
[3] Beijing Key Laboratory of Knowledge Engineering for Materials Science, Beijing 100083, China
* Correspondence: hejie@ustb.edu.cn (J.H.); qiyuee@ustb.edu.cn (Y.Q.)

Abstract: The cold-start problem has always been a key challenge in the recommendation research field. As a popular method to learn a learner that can rapidly adapt to a new task through a small number of updates, meta-learning is considered to be a feasible algorithm to reduce the error of cold-start recommendation. However, meta-learning does not take the diverse interests of users into account, which limits the performance improvement in cold-start scenarios. In this paper, we proposed a new model for a cold-start recommendation, which combines the attention mechanism and meta learning. This method enhances the ability of modeling the personalized user interest by learning the weights between users and items based on the attention mechanism and then improves the performance of the cold-start recommendation. We validated the model with two publicly available datasets in the recommendation field. Compared with the three benchmark methods, the proposed model reduces the mean absolute error by at least 2.3% and the root mean square error of 2.5%.

Keywords: recommendation system; meta-learning; cold-start problem; user preference estimation; attention network

check for updates

Citation: Liu, S.; Liu, Y.; Zhang, X.; Xu, C.; He, J.; Qi, Y. Improving the Performance of Cold-Start Recommendation by Fusion of Attention Network and Meta-Learning. *Electronics* **2023**, *12*, 376. https://doi.org/10.3390/electronics12020376

Academic Editors: Yanwei Yu, Zhu Xiao and Ziqiang Yu

Received: 22 November 2022
Revised: 26 December 2022
Accepted: 9 January 2023
Published: 11 January 2023

1. Introduction

Recommendation systems are usually divided into collaborative filtering-based, content-based, or hybrid recommendation systems based on the different data used by the recommendation algorithm. The collaborative filter-based system estimates user interests by collecting preference information of different users. However, this type of method cannot handle the cold-start problem due to the lack of user–item interaction information [1]. Recently, there are some new approaches based on collaborative filtering that are used in cloud based services to tackle the cold-start problem [2,3]. Content-based recommendation [4] uses the attribute characteristics of the item itself to make recommendations. This method does not require a scoring matrix for user–item interaction records, so it can ease the cold-start to a certain extent. However, it raises a new problem that there are no user evaluations of the items taken into account. Therefore, it can only recommend the same results to users with similar backgrounds rather than personalized recommendation results. The hybrid recommendation, which combines collaborative filtering and content information [5], cannot handle the cold-start scenarios where user–item interaction data are sparse. Therefore, based on the above analysis, these traditional methods are not suitable for solving the cold-start problem.

In recent years, some scholars have tried to solve the cold-start problem from the perspective of meta-learning [6–8]. The goal of cold-start is to learn a model from the interaction history of marked users, which can quickly adapt to new users or new items [9]. Meta-learning, or “learning to learn”, aims to learn a learner that can rapidly adapt to a new task with a few examples [10]. Therefore, the target of meta-learning is similar to the cold-start problem. Since the knowledge learned in all tasks can help the model quickly converge when facing a new task, meta-learning is now a new popular method to improve the performance of cold-start recommendation [11,12].

The traditional meta-learning recommendation method mainly focus on modeling the users or the items [13]. The representative research of the former is the meta-learning user preference estimator (MeLU) [14], which is based on model-agnostic-meta-learning (MAML) to estimate the preferences of new users [15]. The latter is the meta-embedded generator algorithm (ML2E) to embed new items in the model [16]. However, the traditional meta-learning recommendation method does not take the correlation between the users and the items into account. It only embeds the user and item information respectively and then directly submits the embedding data to the multi-layer perceptron (MLP) for training [17].

Attention mechanism originates from the Neural Machine Translation (NMT) field [18]. NMT adopted an approach of taking a weighted sum of all the annotations of input words to compute an expected annotation, which lets the model focus only on information relevant to the generation of the next target word. In the research area of the deep learning recommendation system, the attention model has been widely used due to its excellent weight learning [19]. By drawing on the characteristics of human visual mechanisms, the attention mechanism allows the model to dynamically assign a personalized weight to each part of the network. The attentional factorization machine model (AFM) [20] improves the traditional FM [21] model by distinguishing the importance of different feature combinations. It employs the attention mechanism on feature interactions by performing a weighted sum on the interacted vectors. There are also a series of recommendation models with the attention mechanism proposed by the Ali algorithm team, such as a deep interest network (DIN) [22], deep interest evolution network models [23], etc. These studies show that the use of attention mechanism can help capture user interest.

Inspired by the previous work, this study proposed a recommendation model that combines attention network and meta-learning. Our study provides two main contributions. First, we integrate the Model-Agnostic Meta-Learning (MAML) algorithm and attention network into our model that can not only improve performance by learning with only a small amount of training data, but also capture the user's diversified interests. In each meta-learning training task, the attention network is introduced to weight the interaction between the user and the item. The design of the attention network mainly includes the user fully connected layer and the item fully connected layer. The interaction weights of the user embedding vector and the item embedding vector are used to measure the user's attention to different candidate items, and then reflect the user's diversified interests. Second, the performance of our model is validated by experiments with two benchmark datasets and one application dataset.

The remainder of this study is organized as follows: Section 2 describes related research on recommendation models designed to reduce the error of cold-start recommendation. Section 3 presents our proposed recommendation approach. Then, Sections 4 and 5 describe the experimental design along with the results for two benchmark datasets and one application dataset, respectively. Finally, Section 6 summarizes our conclusions and future research.

2. Related Work

While collaborative filtering-based recommendation systems [24–27] has achieved considerable success in a recommendation field, there still remains a key challenge called cold-start recommendation that deals with new users or items with sparse user–item interactions. Some researchers tried to solve a cold-start problem in the perspective of bandits. Stephane et al. utilized the multi-armed bandits to embed the preference of new users in a social network [28]. On the other hand, some researchers addressed this problem using the Matrix Factorization (MF) technique. It is used to find out user item latent factors based on the available users' preferences, and predict user's preference for new items. Pujahari et al. proposed a probabilistic matrix factorization method to integrate user and side information about items into the model to enable it to capture all the desired user–item interaction information [24]. Feng et al. proposed a hybrid CF ranking model that combines rating-oriented probabilistic matrix factorization (PMF) and a pairwise ranking-oriented Bayesian personalized ranking (BPR) together to address the cold start scenarios [26]. Panagiotakis et al. proposed a Dual Training Error based Correction approach (DTEC) that improves the prediction of recommendations in two stages. DTEC introduced a second stage after initial execution of the recommendation system to improve prediction accuracy by correcting the error in the training set [27]. These methods can improve the recommendation performance as they utilize the contextual information. Nevertheless, most of the hybrid methods make the learning procedure quite complex because of introducing an objective term.

Meta-learning, also known as learning to learn, intends to learn the general knowledge across similar learning tasks, which can rapidly adapt to new tasks based on a few examples [13]. Meta-learning can be classified into three types: metric-based, memory-based, and optimization-based meta-learning. Metric-based methods [29] learn a metric or distance function over tasks, Model based methods [30] aim to design a training process capable of rapid generalization across different tasks. Optimization based methods [15,31] adjust the optimization algorithm directly to enable quick adaptation with a few examples.

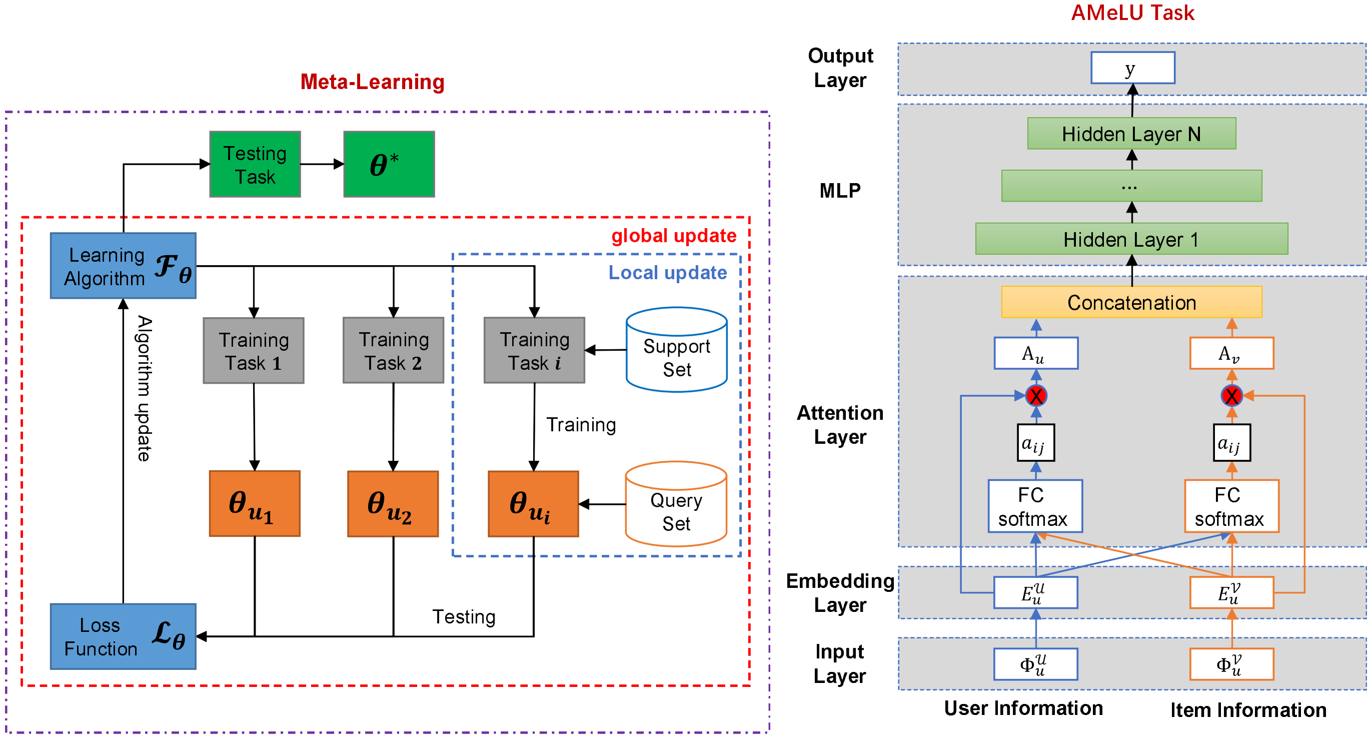
Recently, meta-learning has brought great interest in many research fields, such as computer vision, recommendation system, etc. The success of meta-learning in few-shot settings has shed light on the problem of cold-start recommendation. Specifically, Vartak et al. [32] present a metric-based approach to predict whether a user consumes an item or not to address the item cold-start problem. Lee et al. [14] applies the MAML framework [15] to an MeLU model, which can rapidly adapt to new users or items based on sparse interaction history. Meta-learning with MAML has also been applied to scenario-based cold-start problems, which formulates the scenario-aware recommendation and proposes a novel sequential scenario-specific framework to solve the cold-start problem in some recommendation scenarios [33]. Unfortunately, these methods focus on the algorithm selection of the meta-learning part, but ignores the feature processing of the recommendation module, which often use the same interest vector to describe diverse interests of users. Therefore, in this study, we enhance the ability of modeling the personalized user interest by learning the weights between users and items based on the attention mechanism, which then improves the performance of cold-start recommendation on different cold-start scenarios. The attention mechanism introduced in the model allows different parts of the network to have different contributions when compressed into a single representation, that is, it will enable the model to complete the user interest by calculating the local activation for the user with different items. Therefore, our model can use the correlation of the user's interaction history with candidate items to calculate the interest vector adaptively, rather than using the same vector to describe diverse interests of every user. The attention representation of the interest vector of user and item is relearned by weighted summation.

3. Attention Meta-Learning Recommendation Network

In this section, we describe the details of an Attentional Meta-Learned User preference estimator (AMeLU). The overall framework of the proposed user preference estimation model for the recommendation system is shown in Figure 1. The model takes user preference estimation as a meta-learning training task. For each training task, the attention network is introduced to learn the interaction between users and items, by learning the weight between the user and the candidate item, and then using the weighted sum method to realize the adaptive calculation of the user's interest expression vector.

Inspired by MeLU, our model is based on MAML training. As shown in Figure 1a, the model can quickly adapt to the training of new tasks. However, unlike the MeLU task, we introduce the attention network into the user preference estimator, as shown in Figure 1b.

First, the model takes user content and item content as input, and then uses embedding technology to process the input vector. Second, the attention network uses the user and item content vector after embedding processing as input, and then calculates the weight value between different users and different items. Finally, the user and item feature vectors after the sum of the weights are output.



(**a**) (**b**)

Figure 1. The framework of the user preference estimation model: (**a**) the meta-learning framework; (**b**) the model structure of AMeLU task used in meta-learning.

The meta-learning framework of this model is shown in Figure 1a. The goal of this module is to learn a model parameter on a set of tasks so that, when faced with a new task, it can quickly adapt to the new task. The meta-learning research method used in this paper is Model-Agnostic Meta-Learning. In addition, we will predict each user's preference for different candidate items as a task of learning and training. For each task, prepare a support set $D_{u_i}^S$ and an algorithm $\mathcal{F}$ parameterized by φ for user u_i. The support set contains the user's historical behavior records. The algorithm $\mathcal{F}$ is used to train a recommendation model based on the support set, and then a recommendation model for the user θ_{u_i} is generated. The specific definition of θ_{u_i} is

$$\theta_{u_i} = \mathcal{F}(D_{u_i}^S, \varphi) \tag{1}$$

In the meta-training process, the algorithm $\mathcal{F}$ is trained with multiple training tasks. For a user u_i, we randomly select a support set $D_{u_i}^S$ and a query set $D_{u_i}^Q$ from the user's historical behavior data. The support set and query set here do not have overlapping data, and both contain labeled data. For each user u_i, we use $D_{u_i}^S$ to train and generate a model θ_{u_i} and then use θ_{u_i} to test on $D_{u_i}^Q$, and calculate the gradient of the test loss $\mathcal{L}_{D_{u_i}^Q}(\theta_{u_i})$. The parameter φ of the algorithm is updated and optimized by using the test loss gradient of the sampling task, and the objective function is defined as

$$\min_{\varphi} \sum_{u_i} \mathcal{L}_{D_{u_i}^Q}(\theta_{u_i}) = \sum_{u_i} \mathcal{L}_{D_{u_i}^Q}(\mathcal{F}(D_{u_i}^S, \varphi)) \tag{2}$$

Algorithm 1 shows the detailed training process of user preference estimation. Inspired by the MAML algorithm, we use two training stages: local update and global update. In the local update phase, we initialize the parameters φ randomly, and then train the model for each batch of users. The training process of the local update is completed based on the user's support set, and the process of local update can be regarded as a user-personalized training process. In the global update phase, each user has its own specific parameters θ_{u_i}. We traverse all users, sum the loss gradients calculated on the user query set, and update the original parameter space φ through back-propagation.

Algorithm 1 MAML for User Preference Estimator

```
Input:
    $\alpha$: local update hyperparameters
    $\beta$: global update hyperparameters
    $D_u^S$: meta-training set
    $D_u^Q$: meta-testing set
Output:
    $\varphi$: model's trained parameter space
randomly initialize $\varphi$
while not converge do
    sample batch of users $B \sim p(\mathcal{U})$
    for user $u_i$ in $B$ do
        $\theta_{u_i} = \varphi$;
        local update $\theta_{u_i} \leftarrow \theta_{u_i} - \alpha \nabla_{\theta_{u_i}} \mathcal{L}_{D_{u_i}^S}(\theta_{u_i})$
    end for
    global update $\varphi \leftarrow \varphi - \beta \nabla_{\varphi} \sum_{i \in B} \mathcal{L}_{D_{u_i}^Q}(\theta_{u_i})$
end while
```

In short, we train each user-specific recommendation model through local updates and find an ideal initial recommendation model parameter space for all users through global updates. When faced with a new task, it can use the learned parameters to quickly converge.

AMeLU Task

We then introduce the model structure of AMeLU task used in meta-learning. We begin with input representation. Our aim is to develop a model that is able to handle cold-start scenarios. Consequently, input to the model needs to contain supplementary preference information in addition to content information. Let $\mathcal{U} = \{u_1, \cdots, u_N\}$ and $\mathcal{V} = \{v_1, \cdots, v_M\}$ denote the sets of users and items in the system, respectively, where N is the number of users, and M is the number of items. The user–item interactions can be represented by an $N \times M$ preference matrix $\boldsymbol{R}$, where $\boldsymbol{R}_{uv}$ is the preference for item v by user u. We use $V_v = \mathcal{U}(v) = \{u \in \mathcal{U} \,|\, \boldsymbol{R}_{uv} \neq 0\}$ to denote the set of users that expressed preference for v, and $U_u = \mathcal{V}(u) = \{v \in \mathcal{V} \,|\, \boldsymbol{R}_{uv} \neq 0\}$ to denote the set of items that u expressed preference for. We use $\boldsymbol{\Phi}^{\mathcal{U}}$ and $\boldsymbol{\Phi}^{\mathcal{V}}$ to denote the content features for users and items, respectively, where $\boldsymbol{\Phi}_u^{\mathcal{U}}$ ($\boldsymbol{\Phi}_v^{\mathcal{V}}$) is the content feature vector for user u (item v).

The embedding layer uses embedding technology to extract useful parts from the input features of the model for further training. Studies have shown that embedding technology can convert high-dimensional sparse input vectors into low-dimensional dense representations, which is helpful for the subsequent learning of deep neural networks [34].

Similar to the previous work, the module consists of two embedding matrices E_u and E_v, corresponding to the input users and items, respectively. When the number of user content is P, we define the embedding of user u as

$$E_u^{\mathcal{U}} = [e_{u1}c_{u1}; \cdots; e_{uP}c_{uP}]^T \tag{3}$$

where c_{up} represents a d_p-dimensional one-hot vector for categorical content $p \in \{1, \cdots, P\}$ of user u, and e_{up} represents the d_e-by-d_p embedding matrix for the corresponding categorical content of the user. d_e and d_p are the embedding dimension and the number of categories for content p, respectively. Similarly, when the number of user content is Q, we can define the embedding of item v as

$$E_v^{\mathcal{V}} = [e_{v1}c_{v1}; \cdots ; e_{vQ}c_{vQ}]^T \tag{4}$$

where c_{vq} represents a d_q-dimensional one-hot vector for categorical content $q \in \{1, \cdots, Q\}$ of item v, and e_{vq} represents the d_e-by-d_q embedding matrix for the corresponding categorical content of the item. d_e and d_q are the embedding dimension and the number of categories for content q, respectively.

The subsequent layer is the attention layer. Since the attention mechanism was introduced into the modeling learning of deep neural networks, it has been widely used in many different scenarios, such as information retrieval field, computer vision field, and recommendation field. We know that the traditional meta-learning recommendation network research often only focuses on the algorithm selection of the meta-learning part, but ignores the feature processing of the recommendation module. Specifically, the traditional meta-learning recommendation model performs compression mapping on the input features of the user's items and then directly aggregates and splices them to form a fixed-length vector that represents the user's interest, as shown in Equation (5):

$$E_{\text{pool}} = \text{pooling}(U_i; I_j) \tag{5}$$

The final vector remains unchanged for a given user whatever the candidate item is, which cannot reflect the diverse interests of the user. In order to solve this problem, some researchers have proposed to expand the dimension of the embedding vector to improve its representation ability, but this method will heavily increase the difficulty of learning parameters, causing model overfitting and additional computational overhead.

Inspired by the model mentioned in the paper DIN, we introduced an attention mechanism to allow different parts of the network to have different contributions when compressed into a single representation, that is, to complete the user interest by calculating the local activation for the same user with different candidate items. The introduction of the attention mechanism makes it no longer use the same vector to express the different interests of all users. Instead, the user's interest vector is calculated adaptively by considering the correlation between the user's historical behavior and the candidate items. The attention representation of user and item vector is relearned by weighted summation. The framework of the attention network is shown in Figure 2.

The internal structure of the network is a neural network composed of a fully connected layer (FC) and a softmax output layer. Its input is the interaction vector of two features. These two features have undergone preliminary encoding in the embedding space. Formally, the definition of the attention network is as

$$q_i = w^q \cdot e_i \tag{6}$$

$$k_i = w^k \cdot e_i \tag{7}$$

$$v_i = w^v \cdot e_i \tag{8}$$

$$a'_{ij} = q_i \cdot k_j \tag{9}$$

$$a_{ij} = \frac{\exp(a'_{ij})}{\sum_{(i,j)} \exp(a'_{ij})} \tag{10}$$

$$A_u = [\textstyle\sum_{j=1}^{n} v_i^1 \cdot a_{ij}; \cdots ; \sum_{j=1}^{n} v_i^P \cdot a_{ij}] \tag{11}$$

$$A_v = [\textstyle\sum_{j=1}^{n} v_i^1 \cdot a_{ij}; \cdots ; \sum_{j=1}^{n} v_i^Q \cdot a_{ij}] \tag{12}$$

$$L_0 = [A_u; A_v] \tag{13}$$

Among them, w^q, w^k and w^v correspond to the parameters of model training. The attention weight score a_{ij} is standardized by the softmax function. Equations (11) and (12) apply attention to the user embedding vector $E_u^{\mathcal{U}}$ and item embedding vector $E_v^{\mathcal{V}}$ vectors, respectively, to distinguish the importance of different features and obtain the output A_u and A_v after the final weight distribution of the layer. Finally, we concatenate the output of the upper network in the form of a full connection to obtain the vector L_0, as shown in Equation (13).

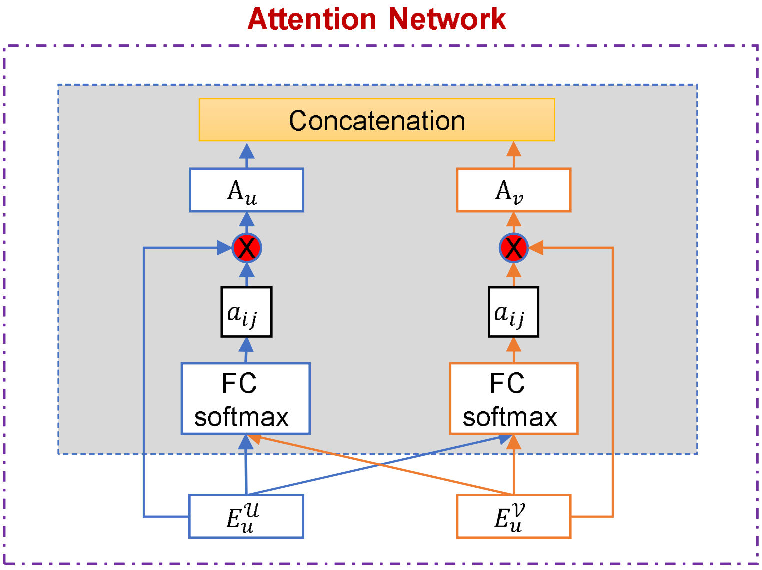


Figure 2. The framework of the attention network.

Let us take a brief look at the added complexity of introducing attention layer. Let $\boldsymbol{Q}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ be the matrix form of q_i, k_i and v_i, respectively. Let $\boldsymbol{X}$ be the input to the attention layer. Then, $\boldsymbol{X}$ will have shape (n, d), where n is the sequence length, and d is the representation dimension. In this attention network, linearly transform the rows of $\boldsymbol{X}$ to compute the query $\boldsymbol{Q}$, key $\boldsymbol{K}$, and value $\boldsymbol{V}$ matrices, each of which has shape (n, d). This is accomplished by post-multiplying $\boldsymbol{X}$ with three learned matrices of shape (d, d), amounting to a computational complexity of $\mathcal{O}(n^2 \cdot d)$. Then, we apply a softmax which is n*n operations, and then multiply the resulting (n, n) matrix by the (n, d) matrix $\boldsymbol{V}$, again resulting in $\mathcal{O}(n^2 \cdot d)$, which is the total complexity of attention layer. In experiments, n is set to be small according to user contents and item contents, and d is set to be 32. Therefore, the attention layer adds relatively little computation.

As for the MLP, we use a N-layer fully-connected neural network to learn high-level combinations of features. This structure can improve the representation ability of the model by fully intersecting the feature information of different dimensions. The formal definition of each layer is shown in Equation (14):

$$\begin{aligned} L_1 &= \mathrm{ReLU}(W_1^{\mathrm{T}} L_0 + b_1) \\ &\cdots \\ L_N &= \mathrm{ReLU}(W_N^{\mathrm{T}} L_{N-1} + b_N) \end{aligned} \tag{14}$$

where W_n and b_n are the weight matrix and bias vector for the n-th layer.

The final output layer uses the activation function sigmoid to obtain the final predicted value and can be formulated as

$$\hat{y}_{uv} = \delta(W_o^{\mathrm{T}} L_N + b_o) \tag{15}$$

where $\hat{y}_{uv}$ is user u's estimated preference for item v, and W_o^{T} and b_o represent the weight matrix and bias vector of the output layer.

4. Experiments

4.1. Dataset

We use two public datasets in the research field of recommendation systems to evaluate the performance of our model in different cold-start environments. The specific datasets used are MovieLens-1M (ML-1M) [35] and Book-Crossing (BC) [36] with basic user item information. The ML-1M dataset is a dataset of an active community named GroupLens. BC is a special recommendation dataset in the book field, which is used to recommend favorite books to users. Both are the most commonly used public datasets in the recommendation field. Table 1 summarizes the characteristic information of these two datasets.

Table 1. Basic statistics of the MovieLens and Bookcrossing dataset.

Characteristics	MovieLens	BookCrossing
Number of users	6040	278,858
Number of items	3706	271,379
Number of ratings	1,000,209	1,149,780
Sparsity	95.5316%	99.9985%
User contents	Gender, Age, Occupation, Zip code	Age, Location
Item contents	Publication year, Rate, Genre, Director, Actor	Publication year, Author, Publisher
Range of ratings	1∼5	1∼10

We divide items and users into two groups (existing/new) to evaluate the performance of the model in user cold-start and item cold-start scenarios. Specifically, the cold-start scenario is divided into four parts: (1) recommendation of existing items for existing users, (2) recommendation of existing items for new users, (3) recommendation of new items for existing users, and (4) recommendation of new users for new items.

4.2. Dataset Pre-Processing

MovieLens-1M: The movies in the movie dataset are divided into existing movies released before 1997 and new movies released after 1998 (approximately 8:2). Similarly, we randomly select 80% of users from all users as existing users in the system, and the rest as new users in the cold-start scenario.

Book-Crossing: The processing method is similar to the movie dataset. We divide the books into existing books released before 1997 and new books released after 1998 (approximately 5:5). At the same time, half of all users are randomly selected as existing users, and the rest are new users in the cold-start scenario.

4.3. Experimental Settings

Meta-learning network settings. In each cold-start scenario, we select users with a length of user history interaction records between 13 and 100. For each user, we will randomly select users from the historical records. Ten items are used as Query Sets for testing in the global update stage, and the remaining 3–90 items are used as Support Sets for training in the local update stage.

Attention recommendation network settings. The dimension of the embedding layer is set to 32, and the dimension of the attention network is set to 32. MLP adopts a two-layer hidden layer and one-layer output layer structure, with 64 nodes in each layer. Its dimensions are 256 × 64, 64 × 64, and 64 × 1, respectively. We set the learning rates α and β in meta-learning to 5×10^{-5} and 5×10^{-4}, respectively. The number of local updates varies from 1 to 5. The epoch and batch size in the algorithm are set to 35 and 32, respectively.

Experiment environment settings. All experiments are conducted on a Linux server with one GPU (GeForce RTX) and CPU (Intel Xeon Platinum 8260), and its operating system is Ubuntu 18.04. We implement the proposed AMeLU with the deep learning library PyTorch. The Python and PyTorch versions are 3.8 and 1.12.0, respectively.

Evaluation indicators. The experiments use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to evaluate the final model performance.

4.4. Experimental Results

We conducted experiments on one non-cold-start scenario, three cold-start scenarios and simulated different levels of cold-start according to different numbers of users or items. Specifically, we defined a proportional variable cold-rate, which is used to describe the level of cold-start scenario (the larger the value, the higher the degree of cold-start and the harsher the experimental conditions):

$$\text{cold-rate} = \frac{N_{new}}{N_{total}} \tag{16}$$

Among them, N_{new} represents the number of new users or new items trained as a cold-start scenario, and N_{total} represents the number of users or items in the dataset. Therefore, we not only divided the experimental scenarios into a conventional non-cold-start and three cold-start scenarios, but also tested the performance of our model in the cold-start scenarios by simulating different cold-start levels.

We compared the proposed method with three benchmark models: the paired preference regression model PPR [37], the Wide and Deep model [38], and the meta-learning user preference prediction model MeLU. Table 2 shows the best performance of these four methods in four recommendation scenarios on two public datasets.

Table 2. Experimental results on MovieLens and Bookcrossing datasets.

Type	Method	MovieLens		BookCrossing	
		MAE	RMSE	MAE	RMSE
Recommendation of existing items for existing users	PPR	**0.1820**	**0.4756**	3.8092	5.2367
	Wide and Deep	0.9047	1.1033	1.6206	4.0802
	MeLU	0.7206	0.8763	1.3003	1.5604
	AMeLU	0.7277	0.8822	**1.2737**	**1.5308**
Recommendation of existing items for new users	PPR	1.0748	1.3421	3.8430	2.5780
	Wide and Deep	1.0694	1.1084	2.0457	2.6475
	MeLU	**0.7446**	**0.9044**	1.4621	1.7344
	AMeLU	0.7466	0.9039	**1.2987**	**1.5521**
Recommendation of new items for existing users	PPR	1.2441	1.5600	3.6821	6.7846
	Wide and Deep	1.2655	1.6453	2.2648	3.8564
	MeLU	0.9077	1.0877	1.7214	1.9811
	AMeLU	**0.8836**	**1.0595**	**1.4155**	**1.6594**
Recommendation of new items for new users	PPR	1.2596	1.7779	3.7046	9.8854
	Wide and Deep	1.3114	1.9012	2.3088	7.3998
	MeLU	0.8951	1.0732	1.7049	1.9696
	AMeLU	**0.8742**	**1.0459**	**1.3959**	**1.6493**

The experimental results in Table 2 show that, in the conventional non-cold-start scenario, the benchmark model PPR performs best on the MovieLens dataset. However, for the three cold-start scenarios, the performance of the PPR model and the Wide and Deep model is relatively poor. The performance degradation of PPR may come from the overfitting when sparsity is low in the MovieLens dataset. Meanwhile, because the Bookcrossing dataset has large sparsity, these models were not well fitted compared to the MovieLens dataset, and we could infer that overfitting did not occur. The MeLU model based on meta-learning has a certain improvement in the performance of the cold-start scenario, but due to the simple design of the recommendation network, the overall performance is worse than the AMeLU model proposed in this paper. Moreover, we could also infer that our proposed model performs well when only a small amount of information about users is available, as for the Bookcrossing dataset. Therefore, it also verifies the necessity for us to increase the attention network. Overall, the performance of our proposed model is the best in the three cold-start scenarios.

In addition, we conduct experiments on different cold-start levels of users, to evaluate the performance of our model at different cold-start levels. The results show that, with the increasing of the cold-start level, the recommendation performance shows a trend of first rising and then falling. As shown in Figures 3 and 4, it can be seen that the performance of our proposed model is better than the other three benchmark models (PPR, Wide and Deep, and MeLU), and the model achieves the best performance when the cold rate is equal to 0.5 to 0.6. Compared with other methods, our model always maintains the best cold-start recommendation performance.

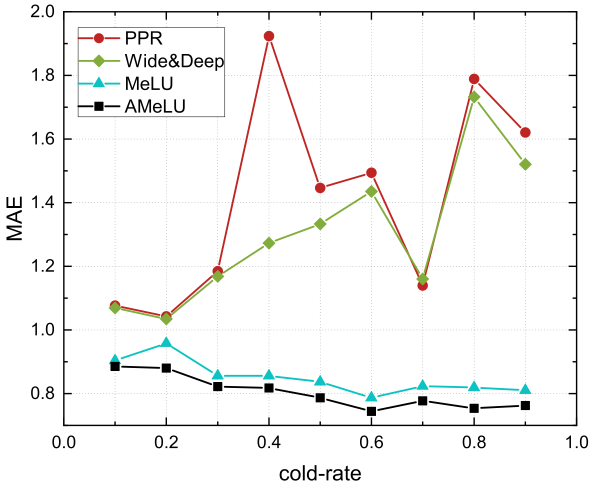


Figure 3. The MAE of four models over different cold-start levels.

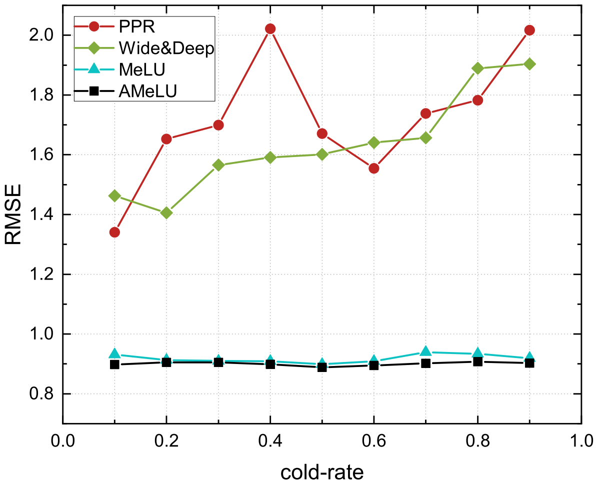


Figure 4. The RMSE of four models over different cold-start levels.

5. Application

The National Materials Data Management and Service platform focuses on the exchange and management of multi-source heterogeneous data in the field of materials research [39]. At present, the platform has a total of 79,588 registered users, 65,621,600 pieces of materials data, 122 kinds of materials fields, 8804 groups of materials datasets and 1920 materials data templates. It is challenging for researchers to quickly find out the datasets they are interested in from such a huge amount of data in the platform. Meanwhile, the numbers of views and downloads of the data in the platform were 134,314 and 1,975,896 times, respectively, and the number of visits of the platform also has reached 799,890 times. The interaction records between users and the platform are relatively few, and the overall data sparsity is high. As a result, whether we make recommendations for users or materials data itself, we will face serious cold-start recommendation performance problems. Therefore, we apply the algorithm proposed in this paper to the materials data recommendation scenario to evaluate the effect of the algorithm in practical application.

For the convenience of description, we denoted the specific application dataset of the platform as MGE-DATA. The dataset contains basic information about materials users, materials metadata, and their interaction records. We conducted a series of preprocessing operations on the original data, and the feature information of the processed MGE-DATA dataset is shown in Table 3. The sparsity of the whole dataset is over 99.92%.

Table 3. Basic statistics and feature information of the MGE-DATA dataset.

Characteristics	MGE-DATA
Number of users	1452
Number of items	12,216,221
Number of interactions	104
Number of search	164,822
Number of uploads	12,691,965
Sparsity	99.9275%
User contents	UserID, Institution,Views, MaterialProjectID, MaterialSubjectID
Item contents	ItemID, Downloads, Views, Project, Subject, CategoryID, TemplateID
User-Item contents	Visit_user, Visit_data_meta_id, Search_params, Views, Upload_records

In practical engineering applications, in order to avoid the occasional error of algorithm training and ensure the fairness of algorithm comparison, it is necessary to perform model training several times and select the average of results. Therefore, in this chapter, each algorithm was trained 10 times independently, and then we take the average of all training results for each algorithm to compare performance.

Table 4 shows the average performance of the four method for the four types of recommendation on the MGE-DATA dataset. The proposed model AMeLU outperforms the other three comparative method in three types of cold-start scenarios in the MGE-DATA dataset. For the recommendation performance of non-cold start scenario, PPR shows the best MAE and RMSE. Specific analysis shows that the performance gaps of PPR comes from the low data sparsity of non-cold-start scenario, which is more suitable for PPR.

Table 4. Experimental results on the MGE-DATA dataset.

Type	**Method**	**MGE-DATA**	
		MAE	**RMSE**
Recommendation of existing items for existing users	PPR	**0.7497**	**1.3721**
	Wide and Deep	0.8034	1.4896
	MeLU	0.7572	1.4904
	AMeLU	0.7549	1.4850
Recommendation of existing items for new users	PPR	1.4847	1.6030
	Wide and Deep	1.4616	1.5926
	MeLU	1.4195	1.5298
	AMeLU	**1.4108**	**1.5106**
Recommendation of new items for existing users	PPR	1.6603	1.7170
	Wide and Deep	1.6162	1.6986
	MeLU	1.5491	1.6196
	AMeLU	**1.5269**	**1.6118**
Recommendation of new items for new users	PPR	1.9263	2.2658
	Wide and Deep	1.8990	2.0936
	MeLU	1.7742	2.0830
	AMeLU	**1.7739**	**2.0783**

Generally, our model showed good performance in cold-start scenarios and non-cold-start scenarios for data recommendation in the materials field. Therefore, our model can alleviate the poor recommendation performance of the platform in the cold-start scenario to a certain extent, and improve the probability and user experience for which material researchers find preferenced data quickly and accurately from massive data in the platform.

6. Conclusions

In this paper, we proposed a recommendation model AMeLU that combines attention network and meta-learning to mitigate the cold-start problem. It was able to perform fine-grained interactive processing on the features by using the small sample learning ability of meta-learning. Moreover, the introduction of the attention mechanism greatly improves the model's ability to capture the diverse interests of users. We have validated AMeLU with two benchmark datasets, and it outperforms other three comparative benchmark models on the datasets. The proposed model also shows good performance in the application area of data recommendation for material data platforms. Therefore, our model can further improve the recommendation performance in different cold-start scenarios. There still remains some promising topics for future research. For example, there are various meta-learning based recommendation systems that can be studied and used to further improve the performance of our model. We plan to introduce input dropout during training to condition for missing preference information, which may lead to excellent generalization on both warm and cold start scenarios.

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

Funding: This work was supported by the National Key Research and Development Program of China (Grant No. 2021YFB3702400), National Natural Science Foundation of China (Grant No. 61971031), Foshan Science and Technology Innovation Special Foundation (Grant No. BK22BF001).

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mooney, R.J.; Roy, L. Content-based book recommending using learning for text categorization. In Proceedings of the Fifth ACM Conference on Digital Libraries, San Antonio, TX, USA, 2–7 June 2000; pp. 195–204. [CrossRef]
2. Venkatesan, T.; Saravanan, K.; Ramkumar, T. A Big Data Recommendation Engine Framework Based on Local Pattern Analytics Strategy for Mining Multi-Sourced Big Data. *J. Inf. Knowl. Manag.* **2019**, *18*, 1950009.
3. Nagarajan, R.; Thirunavukarasu, R. A Service Context-Aware QoS Prediction and Recommendation of Cloud Infrastructure Services. *Arab. J. Sci. Eng.* **2020**, *45*, 2929–2943.
4. Narducci, F.; Basile, P.; Musto, C.; Lops, P.; Caputo, A.; de Gemmis, M.; Iaquinta, L.; Semeraro, G. Concept-based item representations for a cross-lingual content-based recommendation process. *Inf. Sci.* **2016**, *374*, 15–31. [CrossRef]
5. Kouki, P.; Fakhraei, S.; Foulds, J.; Eirinaki, M.; Getoor, L. Hyper: A flexible and extensible probabilistic framework for hybrid recommender systems. In Proceedings of the 9th ACM Conference on Recommender Systems, Vienna, Austria, 16–20 September 2015; pp. 99–106. [CrossRef]
6. Bharadhwaj, H. Meta-Learning for User Cold-Start Recommendation. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8. [CrossRef]
7. Mantovani, R.G.; Rossi, A.L.D.; Alcobaca, E.; Vanschoren, J.; de Carvalho, A.C. A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers. *Inf. Sci.* **2019**, *501*, 193–221. [CrossRef]
8. Collins, A.; Tkaczyk, D.; Beel, J. One-at-a-time: A Meta-Learning Recommender-System for Recommendation-Algorithm Selection on Micro Level. *arXiv* **2018**, arXiv:1805.12118. [CrossRef]
9. Huisman, M.; van Rijn, J.N.; Plaat, A. A survey of deep meta-learning. *Artif. Intell. Rev.* **2021**, *54*, 4483–4541. [CrossRef]
10. Cunha, T.; Soares, C.; de Carvalho, A.C. Metalearning and Recommender Systems: A literature review and empirical study on the algorithm selection problem for Collaborative Filtering. *Inf. Sci.* **2018**, *423*, 128–144. [CrossRef]
11. Zhang, F.; Zhou, Q. A Meta-learning-based Approach for Detecting Profile Injection Attacks in Collaborative Recommender Systems. *JCP* **2012**, *7*, 226–234. [CrossRef]
12. Chen, F.; Luo, M.; Dong, Z.; Li, Z.; He, X. Federated Meta-Learning with Fast Convergence and Efficient Communication. *arXiv* **2019**, arXiv:1802.07876. [CrossRef]
13. Vilalta, R.; Drissi, Y. A perspective view and survey of meta-learning. *Artif. Intell. Rev.* **2002**, *18*, 77–95. [CrossRef]
14. Lee, H.; Im, J.; Jang, S.; Cho, H.; Chung, S. Melu: Meta-learned user preference estimator for cold-start recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 1073–1082. [CrossRef]
15. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
16. Wang, H.; Zhao, Y. ML2E: Meta-learning embedding ensemble for cold-start recommendation. *IEEE Access* **2020**, *8*, 165757–165768. [CrossRef]
17. Gardner, M.W.; Dorling, S.R. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmos. Environ.* **1998**, *32*, 2627–2636. [CrossRef]
18. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2016**, arXiv:1409.0473. [CrossRef]
19. Rush, A.M.; Chopra, S.; Weston, J. A Neural Attention Model for Abstractive Sentence Summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 379–389. [CrossRef]
20. Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; Chua, T.S. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17), Melbourne, Australia, 19–25 August 2017; pp. 3119–3125.
21. Rendle, S. Factorization Machines. In Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, Australia, 13–17 December 2010; pp. 995–1000. [CrossRef]
22. Zhou, G.; Zhu, X.; Song, C.; Fan, Y.; Zhu, H.; Ma, X.; Yan, Y.; Jin, J.; Li, H.; Gai, K. Deep interest network for click-through rate prediction. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 1059–1068. [CrossRef]
23. Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; Gai, K. Deep Interest Evolution Network for Click-Through Rate Prediction. *AAAI* **2019**, *33*, 5941–5948. [CrossRef]

24. Pujahari, A.; Sisodia, D.S. Pair-wise Preference Relation based Probabilistic Matrix Factorization for Collaborative Filtering in Recommender System. *Knowl.-Based Syst.* **2020**, *196*, 105798. [CrossRef]
25. Natarajan, S.; Vairavasundaram, S.; Natarajan, S.; Gandomi, A.H. Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data. *Expert Syst. Appl.* **2020**, *149*, 113248. [CrossRef]
26. Feng, J.; Xia, Z.; Feng, X.; Peng, J. RBPR: A hybrid model for the new user cold start problem in recommender systems. *Knowl.-Based Syst.* **2021**, *214*, 106732. [CrossRef]
27. Panagiotakis, C.; Papadakis, H.; Papagrigoriou, A.; Fragopoulou, P. Improving recommender systems via a Dual Training Error based Correction approach. *Expert Syst. Appl.* **2021**, *183*, 115386. [CrossRef]
28. Caron, S.; Bhagat, S. Mixing bandits: A recipe for improved cold-start recommendations in a social network. In Proceedings of the 7th Workshop on Social Network Mining and Analysis (SNAKDD '13), Chicago, IL, USA, 11–14 August 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 1–9. [CrossRef]
29. Snell, J.; Swersky, K.; Zemel, R. Prototypical Networks for Few-shot Learning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2019; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
30. Santoro, A.; Bartunov, S.; Botvinick, M.; Wierstra, D.; Lillicrap, T. Meta-Learning with Memory-Augmented Neural Networks. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1842–1850.
31. Li, Z.; Zhou, F.; Chen, F.; Li, H. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. *arXiv* **2017**, arXiv:1707.09835.
32. Vartak, M.; Thiagarajan, A.; Miranda, C.; Bratman, J.; Larochelle, H. A Meta-Learning Perspective on Cold-Start Recommendations for Items. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
33. Du, Z.; Wang, X.; Yang, H.; Zhou, J.; Tang, J. Sequential Scenario-Specific Meta Learner for Online Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19), Anchorage, AK, USA, 4–8 August 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 2895–2904. [CrossRef]
34. Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1225–1234. [CrossRef]
35. Harper, F.M.; Konstan, J.A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst. (TIIS)* **2015**, *5*, 1–19. [CrossRef]
36. Ziegler, C.N.; McNee, S.M.; Konstan, J.A.; Lausen, G. Improving recommendation lists through topic diversification. In Proceedings of the 14th International Conference on World Wide Web, Sydney, Australia, 23–24 December 2005; pp. 22–32. [CrossRef]
37. Park, S.T.; Chu, W. Pairwise preference regression for cold-start recommendation. In Proceedings of the Third ACM Conference on Recommender Systems, New York, NY, USA, 22–25 October 2009; pp. 21–28. [CrossRef]
38. Cheng, H.T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M. Wide & deep learning for recommender systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 7–10. [CrossRef]
39. Liu, S.; Su, Y.; Yin, H.; Zhang, D.; He, J.; Huang, H.; Jiang, X.; Wang, X.; Gong, H.; Li, Z.; et al. An infrastructure with user-centered presentation data model for integrated management of materials data and services. *npj Comput. Mater.* **2021**, *7*, 88. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.