CPIN: Comprehensive present-interest network for CTR prediction

Wenxing Hong[a,*], Ziang Xiong[a], Jinjie You[a] and Xiaolin Wu[a]

[a]*School of Aerospace Engineering, Xiamen University, Xiamen 361005, Fujian, China*

ARTICLE INFO

Keywords:
Recommender system
Deep neural network
Interest representation.

ABSTRACT

Personalized recommendation is a popular direction of both corporate and academic researches. Many researches on recommender systems utilize the user's interaction history on items to represent users' interests, and achieved remarkable success. Users' interests in the real world are dynamically changing and have a strong correlation with the interaction sequence. However, sometimes users' interests are less relevant to the order of the current interaction sequence, but are more relevant to some of the items existing in the user interaction history. In this paper, we proposed a novel deep neural network model to handle this situation. Our model consists of two parts: the present interest relevant to the order of the interaction sequence and the comprehensive interest relevant to some items in the interaction sequence. And we constructed an ancillary multi-layer perceptron (MLP) to improve the training of our model. We conducted experiments on public and industrial datasets. The results revealed that our proposed model outperforms the state-of-the-art models, and verified the effectiveness of the ancillary MLP.

1. Introduction

With the rapid development of the Internet, the way people obtaining information has changed greatly, which could be roughly divided into two categories. The first is retrieving information initiatively using search engines, such as Baidu, Google, Bing, etc. The other is passive information recommendation, which we call it recommender systems officially.

The recommender system is one of the research fields of machine learning, and it is also an important part of business application(Covington et al., 2016; Lei et al., 2020; He et al., 2014; Richardson et al., 2007). The development of recommender system started in 1990s(Resnick et al., 1994; Resnick & Varian, 1997; Linden et al., 2003). The task of recommender system could be divided into 3 categories roughly: Click-Through Rate (CTR) prediction(Cheng et al., 2016; Qu et al., 2016; Juan et al., 2016; Guo et al., 2017), Rating prediction(Tang et al., 2015; Symeonidis et al., 2011; Hong et al., 2020), and Top-N recommendation(Xue et al., 2019; Deshpande & Karypis, 2004; Wu et al., 2016). CTR prediction is usually treated as the binary classification problem. The classic method to predict click-through rate is logistic regression (LR). LR model is fast and explainable, the weight of each feature could be explained as the importance to the output, and the output represents the probability directly. However, in most recommending scenarios, the features are no-linear. LR method could not handle no-linear interaction of features unless we manually build the no-linear feature interactions.

There are several methods to extract no-linear interactions among features. The most used method is deep neural networks (DNN) which could passibly fit any no-linear functions with its hidden layers and activation functions. Besides, there are hundreds of thousands of items in many fields respectively, such as electronic commerce, movies and job-search websites. While the items that the user interacted is far less than the whole item set, which leads to the problem of data sparsity. A widely used method to solve the problem of data sparsity is converting the high-dimensional and sparse feature into the low-dimensional and dense vector. The DNN model then learns the no-linear and high-order interactions among features. Besides of the no-linear interactions, Cheng proposed Wide&Deep(Cheng et al., 2016) model to jointly train the DNN part and the linear part, alleviated the over-generalize effect of the DNN. Qu proposed the Product-based Neural networks (PNN)(Qu et al., 2016) with a product layer ahead of the DNN to capture interactive patterns of features, given a novel solution of feature interaction extraction. Factorization machine (FM)[(Rendle, 2010) is proposed to solve the problem of data sparsity, which is another method to extract interactions of features. Guo proposed the DeepFM(Guo et al., 2017) based on Cheng's work to jointly train the DNN part and the FM part, these two parts are regarded as the high-order and the low-order interactions respectively, and achieved considerable performance.

Most existing recommender system models ultilize user-item interaction history to capture user's interest, such as purchasing, rating, browsing, etc. These models utilize the items existing in the interaction history of the user without extracting the sequence information to predict the user's preference on a certain item. However, user's interest is changing dynamically and has a strong correlation with the order of the interaction sequence. For instance, the user bought a computer motherboard, we could assume that the motherboard is for changing. And then he bought CPU, GPU and hard drive, then we could assume that he is assembling a new computer, so the memory chips, mouse and keyboard should be recommended. We call this kind of interest as present interest. However, the recommendation would be biased to those items which are highly relevant to the sequence if the recommender system only considered the order of the

*Corresponding author

hwx@xmu.edu.cn (W. Hong); xiongza@stu.xmu.edu.cn (Z. Xiong); 23220191151262@stu.xmu.edu.cn (J. You); wuxiaolin@stu.xmu.edu.cn (X. Wu)

ORCID(s): 0000-0003-3687-1812 (W. Hong); 0000-0002-3783-9462 (Z. Xiong)

purchasing sequence. In other words, the user may have the other needs not relate to computer after his computer is built completely. We call this kind of interest as comprehensive interest.

We use the gated recurrent unit (GRU)(Cho et al., 2014) which is well performed in sequential information extraction to extract user's present interest. The comprehensive interest is composed of those items in the interaction history, which are highly relevant to the target item. The relevance is captured by an attentive network taking each item in the interaction history and the target item as input. The present interest, comprehensive interest and the target item are input into the multi-layer perceptron (MLP) to explore high-order and no-linear feature interactions. We name our proposed model as Comprehensive Present-Interest Network, short as CPIN. The contributions of this paper are summarized as follows:

- We propose a model which is closer to the real-world recommending scenario by jointly learning both present and comprehensive interest of the user, and our model achieved better performance than the state-of-the-art models.
- We hypothesize that user's comprehensive interest remains stable for some time, and we take the sub-sequence and its following item in the interaction history as a pair of positive example. We construct another MLP (Ancillary MLP) with similar structure to train the correlationship between the sub-sequence and its following item. In this way, we could increase the training data indirectly and increase the generalization capability of our model.
- We conducted our experiments on both public and industrial data, and the results show that our proposed model performs better than the state-of-the-art models on multiple evaluation metrics. And we verified the effectiveness of the ancillary MLP.

The rest of this paper is organized as follows. We first review those works that related to our proposed model in Section 2. Then we split out proposed model CPIN into several parts and illuminate each part in details in Section 3. Afterwards we conduct experiments on both public benchmark and industrial datasets in Section 4. Finally in Section 5, we give a short conclusion to our work in this paper and point the future work.

2. Related work

Many DNN based recommender system researches show the fact that DNN based models perform better than traditional recommendation models, like collaborative filtering(Sarwar et al., 2001), SVD(Golub & Reinsch, 1971), SVD++(Koren, 2008), on handling sparse features and extracting nonlinear and high-order interactions. Following this way, our proposed model was developed from the state-of-the-art DNN based recommendation models. The rest of this section reviews those models which inspired us to build our model.

In the real world, features are not independent of each other, thus linear models have limitations in extracting nonlinear interactions(Bishop, 2006). It is necessary to introduce the nonlinear interactions among the features. However, constructing these feature interactions is a complex task with artificial feature engineering and requires a large amount of expert knowledge. With the help of deep neural network, we could handle those sparse and discrete features through compressing the high-dimensional and sparse features into low-dimensional and dense vectors. In this way, we can greatly reduce the manual feature engineering and find some unseen feature combination patterns through the training of the deep neural networks. However, the deep neural networks could generally over-generalize.

Wide&Deep(Cheng et al., 2016) model jointly trains the linear model and the deep neural network to combine the strengths of both memorization and generalization. Their work revealed that the deep neural network has the capability of combining the advantages of submodules.

DeepFM(Guo et al., 2017) model is developed from Wide&Deep, their work point out that how to construct the interactions effectively among features is a major challenge in recommender system research. Some feature interactions which could be easily understood could be constructed artificially, while the other hidden in the data should be extracted automatically through machine learning. Besides of the high-order interactions, the low-order interactions should not be neglected either. They replaced the wide part of Wide&Deep with factorization machine (FM)(Rendle, 2010) to construct the low-order interactions which contains both the 1-order linear model and the 2-order interactions. The DeepFM model could construct feature interactions automatically by jointly training the deep neural network and the factorization machine.

Just like the latent factor model(LFM) taking the inner product of the user feature vector and the item feature vector to predict the rating, He proposed the NAIS(He et al., 2018) model in 2018. They took the items which exist in the user's interaction history to predict the user's preference to the target item. Their work reveals that the item existing in the interaction history does not contribute equally in estimating the similarity between the user interest and the target item. However, the NAIS is some kind of a linear model. Xue proposed DeepICF(Xue et al., 2019) based on NAIS later to extract the high-order and nonlinear interactions in the embedding vectors of the items and achieved better performance than NAIS. Plenty of works(Wu et al., 2019; Li et al., 2018; Sachdeva et al., 2018; Chen et al., 2017) have verified that learning the dynamic weights of features or items through methods like attention mechanism could improve the feature representation and then improve the performance of the model.

Those works above proposed different feature constructing and interaction extracting methods and made such great contributions. However, those works only considered whether the items existed in the user interaction history, and the order of these items does not make any contribution to

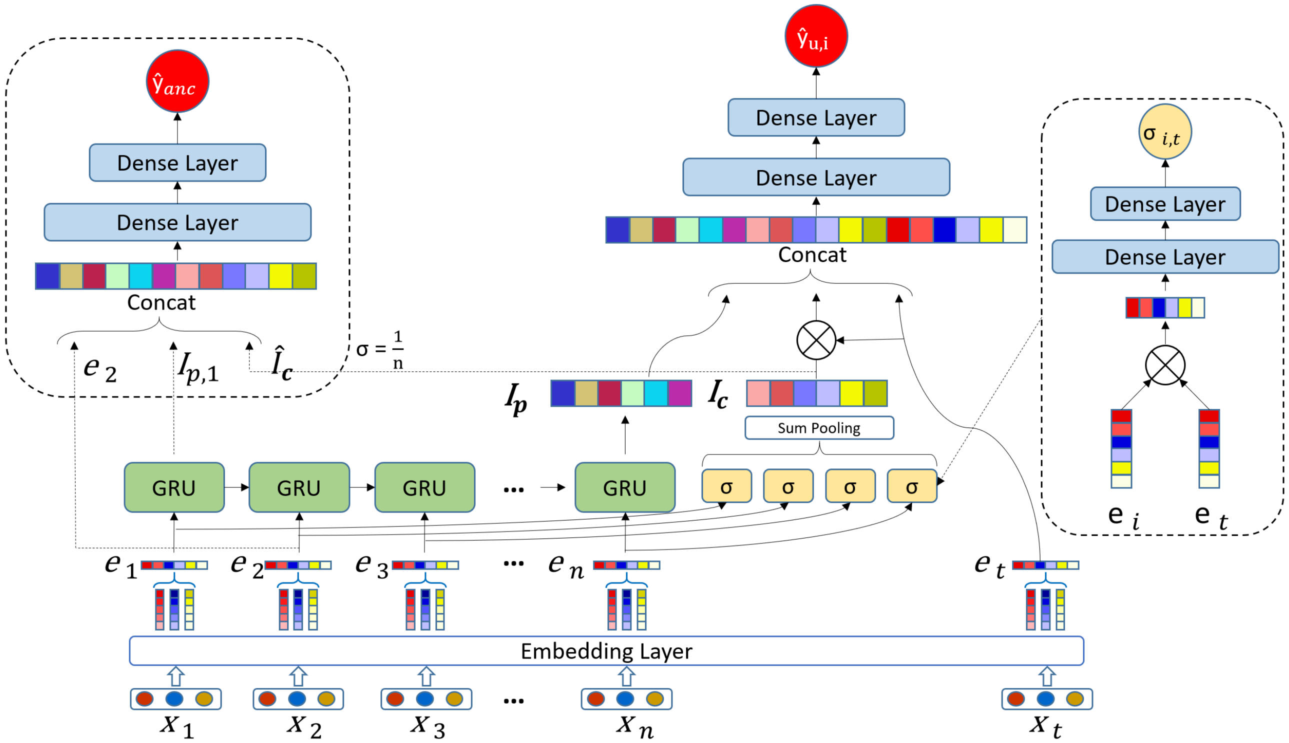

Figure 1: The structure of our proposed model CPIN which captures both the present interest I_p and the comprehensive interest I_c. The attentive weight σ is estimated from the attention network taking the individual item in the interaction history and the target item as input.

the recommending result. As a matter of fact, in the real-world recommending scenario, the order of the user's interacted items does contribute to the next interacted item generally. We name this kind of interest as present interest in this paper. Besides of the present interest, there also exists the basic interest which is greater than the present interest sometimes, just as the example we mensioned in Section 1. We name this kind of interest as comprehensive interest in this paper. Then we proposed the comprehensive present-interest network (CPIN) to jointly train the relationship of the target item to both the comprehensive and present interest.

3. Comprehensive present-interest network

In this section we will introduce our proposed model CPIN which combines both the present interest and the comprehensive interest to estimate the preference of the user to the target item. Figure 1 illustrated the structure of CPIN, the overall structure is a deep neural network which takes the interaction history and the target item as input and estimates the CTR score $\hat{y}_{u,i}$. Each part of CPIN will be illustrated in details in the following subsections.

3.1. Item embedding layer

As a matter of fact, the items that a user interacted account for little among the whole item set in the real-world scenario, which leads to the problem of sparsity. If the feature is represented by the items directly, the parameters will be too many to train and lead to poor performance. While in the DNN based model we use low-dimensional and dense vectors to represent each item. Items are represented by $e_i \in \mathbb{R}^{N\times d}$ respectively, where i is the index of the item in the whole item set, N is the size of the item set and d is the size of the embedding vector. The embedding vectors of the items in interaction history and the target item share the same feature space, each dimension of the embedding vector represents a feature of the item, such as brand and category in commercial recommending scenario. Typically, we do not care about the real meaning of the embedding vectors and then we do not set these vectors artificially, these embedding vectors are constructed automatically without any artificial feature engineering but depend on large number of data.

3.2. User's present interest representation

In real life, users' interests are often continuous and with some causal relations. For shopping, people's interest on a certain category of products often increases after purchasing a series of goods. So the next product to be recommended should be highly relevant to the previous purchasing sequence. We name this kind of interest as user's present interest.

Recurrent Neural Network (RNN) is usually used to handle the problems with sequential input. However, if the sequence is too long, the gradient would decline to near 0, which leads to a slow and under fitting training process. Besides, similar with the next word prediction problem in natural language processing (NLP), the words (items) in the sequence (user's interaction history) do not contribute equally to the prediction of the next word (target item). So we use the gated recurrent unit (GRU) to extract user's present interest. The GRU is formulated as follows:

$$\begin{cases} z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \\ r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \\ \tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\ h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \end{cases} \tag{1}$$

where x_t is the embedding vector of the t-th interacted item, h_t is the state of the t-th unit, $W_z, W_r, W \in R^{n_h \times n_x}$.

As shown in Figure 1, the last state contains the sequential interest information extracted from the user's interaction history. Then we use the last state of GRU to represent user's present interest. The user's present interest I_p is represented as follows:

$$I_p = h_n \tag{2}$$

3.3. User's comprehensive interest representation

The GRU declines the contributions of some items which are less relevant to the interaction history sequence. However, different from the NLP problem which is highly relevant to the order of the sequence, those items should not be abandoned directly. There are also many items that the user is interested in which is less relevant to the sequence of user's interaction history. We not only assemble computers, but also buy clothes and food. If we take only the user's present interest into consider, the recommended items would be biased to the user's interaction history sequence. Once the user's sequential interest ended, such as the computer is assembled completely, the recommended items may not be the most relevant items to the user.

Then we introduce our point in this paper. Some items in the interaction history may contribute little to the sequential feature extraction, but there are also the other interest of the user besides the present interest which may be relevant to those items. We name this kind of interest as user's comprehensive interest. Similar with the content-based recommendation, the user's interest is constructed by the features of the items in the user's interaction history. While the items in the user's interaction history do not contribute equally to the user interest. The items which are highly relevant to the target item should contribute more to construct the user's interest, reducing the interference of the irrelevant items. And then we use the attention network to generate the weight of each item in the user's interaction history. The attention network functions are formulated as follows:

$$\begin{cases} v_i = e_i \cdot e_t \\ v = [v_1, v_2 \ldots, v_n] \\ \sigma(v) = softmax(h^T ReLU(W \cdot v + b)) \end{cases} \tag{3}$$

where v_i is the point-wise product of the embedding vectors of the i-th interacted item and the target item, $W \in R^{n_h \times n_i}$, $h \in R^{n_h \times 1}$, and $b \in R^{n_h \times 1}$, n_i is the interaction history size and n_h is the hidden size. The user's comprehensive interest is generated as:

$$I_c = \sum_{i=1}^{N} \sigma(v)[i] \cdot e_i \tag{4}$$

The user's comprehensive interest I_c is then represented by the weighted sum of the embedding vectors of the items in the interaction history.

3.4. Nonlinear and high-order interactions

The multi-layer perceptron (MLP) is generally used to capture the nonlinear and high-order feature interactions. We concatenate the user's present interest I_p, the user's comprehensive interest I_c and the embedding vector of the target item e_t as the input of the MLP. We name this MLP as the main MLP. The loss function is formulated as follows:

$$L_{main} = -\frac{1}{N} \sum_{i=1}^{N} (y_i \log \tilde{y}_i + (1 - y_i) \log(1 - \tilde{y}_i)) \tag{5}$$

where N is the number of the samples, $y_i \in \{0,1\}$ represents whether the target item is interacted by the user, and $\tilde{y}_i$ is the final output of the main MLP which represents the preference score of the user on the item.

Generally, a sample is generated by a interaction list by a certain user. However, the number of users is generally far small than items, which leads to the problem that many items will not be trained as the target item. To enlarge the dataset, we split the original interaction sequence into several subsequences. The subsequence with its following item in the original interaction list composes a positive sample, while the subsequence with a randomly selected item composes the corresponding negative sample. In this way, more sequence-item relationship will be covered.

We proposed the assumption that the user's comprehensive interest remains stable for some time. This kind of interest is represented the same with the equation (4), where the σ is manually set to $\frac{1}{n}$, where n is the size of the interaction sequence. Meanwhile, the state of each subsequence is generated in the processing of the final state, we do not have to calculate them repeatedly. With the new $I_{p,i}$, $\hat{I}_c$ and e_{i+1} split from the interaction history, we constructed another MLP similar with the main MLP to help training the GRU weights and the embedding matrix of items. As the training progressed, we introduced additional information from the subsequences to improve the presentation of the item and update the GRU weights. We name the MLP of this part as the ancillary MLP.

The loss function of the ancillary MLP is the same with the main MLP, and the final loss function is formulated as follows:

$$L = L_{main} + L_{anc} \tag{6}$$

3.5. Different representation of interest

In this subsection we will introduce several different ways to combine features to build the user's interest representation DNN_{input}, which is the input of the deep neural network. We have tried 3 different feature combination methods and described them in detail respectively in the following content.

CPIN-C The comprehensive interest I_c, the present interest I_p and the embedding vector of the target item e_t are concatenated into one vector. In this way, the deep neural network could combine the advantages of different parts of interest extracted from the user's interaction history. The concatenating process is shown in Figure 2.

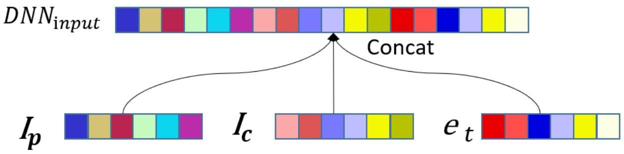

Figure 2: The feature combination of CPIN-C.

CPIN-D Many works(Qu et al., 2016; Guo et al., 2017; Xue et al., 2019) demonstrated that the low-order interaction is also important besides the non-linear and high-order interactions extracted by the deep neural networks. We introduced the point-wise product operation in the comprehensive interest representation to capture low-order interactive patterns. The concatenating process is shown in Figure 3.

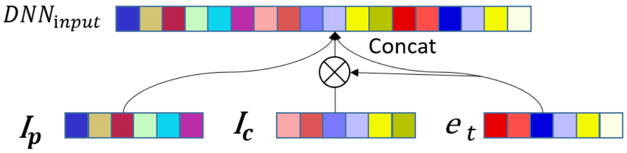

Figure 3: The feature combination of CPIN-D.

CPIN-M We have tried introducing the low-order interaction to every single step of the GRU network, which means the target item affects the order of the items that the user has interacted and finally leads to the target item. The embedding vectors of the items in the sequence multiply the target item vector before inputing into the GRU to capture some latent interactive patterns. The concatenating process is shown in Figure 4.

4. Experiments

In this section, we will elaborate our experiments that we conducted on three datasets, which contains two public datasets and one industrial dataset, and compare the results of different models. We will also analyze the experimental results in detail and verify the effectiveness of our proposed model.

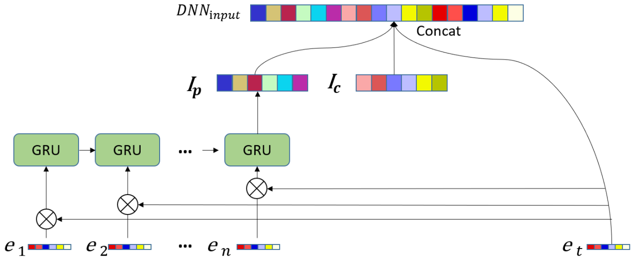

Figure 4: The feature combination of CPIN-M.

4.1. Datasets and Experimental settings

In most recommending scenario, no matter watching movies or seeking for jobs, the user's next choice of the items is bound to be related to the items that the user has intereacted before. Besides, some of them is highly relevant to the order of the interaction sequence. We evaluated our proposed model CPIN on three datasets.These three datasets are described in detail respectively in the following content.

Public dataset

1. We chose the Movielens-20M(Harper & Konstan, 2015) provided by GroupLens as one of the public datasets. The Movielens-20M dataset includes 20,000,263 ratings applied to 27278 movies by 138,493 users. The ratings range from 1 to 5. Each user rated at least 20 movies.
2. Amazon product data(He & McAuley, 2016; McAuley et al., 2015) contains the users' ratings on different categories of products spanning May 1996 - July 2014. We chose the books ratings which contains the most ratings samples among all the other categories as another public dataset to conduct our experiments. This dataset includes 22 million ratings and we reduced them to about 8 million by removing those users whose purchasing number of books are less than 5.

Industrial dataset

The industrial dataset is provided by a job seeker website. The dataset includes 3 million job application records applied to 166,876 jobs by 97,804 users from April to October in 2019. Each user applied at least 5 jobs.

The statistic details of the three datasets are presented in Table 1.

Table 1
Statistics of public and industrial datasets

Dataset	Users	Items	Interactions	Interacted items
Movielens-20M	138,493	27,278	20,000,263	20~100
Amazon-books	1,210,644	371,167	8,917,149	5~100
Job Seeker	97,804	166,876	2,862,584	5~100

We artificially selected the movie ID and the corresponding category as the feature of movies in the Movielens-20M dataset, while we use only the book ID to represent each book. The feature of jobs in the industrial dataset is formed by job ID, company ID, the nature of company, etc. There are 9 features of jobs selected artificially in the industrial dataset. The longest interaction sequence is limited to 100, and the shortest sequence is limited to 20 in Movielens-20M dataset and 5 in Amazon-books and Job Seeker datasets. We take the last item as the positive target item and randomly select an item that the user has not interacted as the negative target item. Besides, for training the ancillary MLP, the number of negtive target items is equal to the number of the subsequences.

The embedding size is set to 8, and the network structure of the main MLP and the ancillary MLP are both set to [32, 16, 1]. We also use the batch normalization layer(Ioffe & Szegedy, 2015) to speed up training and avoid over fitting. We use AUC metric to evaluate the CTR performance and use NDCG@K and HR@K to eavaluate the Top-N performance of different models, K is set to 10.

The full name of AUC is Area Under Curve, it is the area between the ROC curve and the axes, plotted by False Positive Rate(FPR) on the horizontal axis and True Positive Rate(TPR) on the vertical axis. It evaluates the ranking performance of the model.

NDCG@K is Normalized Discounted Cumulative Gain, it evaluates the performance of the rank of the recomended items. We select the last item in the interaction list as the item that the user is interested in, we name it the positive item, and 99 negative items which are randomly selected from the item set. The higher rank the positive item gets, the better performance the model performs.

HR@K is the Hit Ratio of the positive item in the 100 items mentioned above. It evaluates the ratio that the positive item ranks top K.

4.2. Model comparison

We compared and verified the effectiveness of our proposed model by comparing the experimental results of different models on three datasets. We briefly introduce these recommender system methods as follows:

Wide&Deep It is composed of the wide part and the deep part. The wide part is the 1-order linear model and the deep part is the deep neural network. The DNN automatically captures nonlinear and high order feature interactions.

PNN This model constructs a product layer before DNN to capture interactive patterns between inter-field categories, taking the result of the product layer which contains the low-order interactive patterns as the input of the fully connected layers.

DeepFM This model is developed from Wide&Deep. The low-order interactions are not captured enough by only the 1-order linear model, so the wide part is replaced by the factorization machine to capture both 1-order and 2-order interactions.

DeepICF This model generates the user's interest based on the relevance of the item in the interaction history to the target item, and also introduces a product layer before the deep neural networks.

4.3. Performance evaluation

In this subsection, we will evaluate the performance of different models on three datasets.

4.3.1. Effectiveness comparison of different models

The performance of CTR and Top-N prediction of different models on three datasets is shown in Table 2, Table 3 and Table 4 respectively. We observed that learning the 2-order feature interactions improves the performance of the models from the fact that the DeepFM and PNN outperform Wide&Deep on all three datasets. The performance of DeepICF model revealed that the items in the interaction history do not contribute equally in representing user's interest. The attention network evaluated the corresponding weight of each item and improved the representation of user interest. Our proposed model CPIN got the best performance which revealed that learning user's present interest and comprehensive interest jointly improves the performance of the recommender system, and it is more closer to the real recommending scenario that taking both comprehensive and present interest into consider. Among the 3 representation of interest, the CPIN-D performed the best. The point-wise product operation captures low-order interactive patterns in features and then gets some improvement based on CPIN-C. While in CPIN-M, the product layers become noise information in user's present interest representation. We suppose it is because that the target item is not always highly relevant to every item in the interaction sequence, and it introduces extra noise information into the representation of the items.

Table 2
Performance evaluation of different models on Movielens-20M dataset.

Models	AUC	HR@10	NDCG@10
Wide&Deep	0.9494	0.8250	0.5403
PNN	0.9516	0.8273	0.5421
DeepFM	0.9532	0.8271	0.5404
DeepICF	0.9456	0.8322	0.5489
ComprehensivePart[a]	0.9467	0.8326	0.5472
PresentPart[b]	0.9740	0.8412	0.5458
CPIN-C	0.9855	0.8427	0.5517
CPIN-D	**0.9878**	**0.8444**	**0.5551**
CPIN-M	0.9556	0.8365	0.5521
CPIN-D\ancMLP[c]	0.9786	0.8412	0.5503

[a] CPIN-D without present interest.
[b] CPIN-D without comprehensive interest.
[c] CPIN-D without ancillary MLP.

Figure 5 and Figure 6, which are more representative, demonstrated the training process of 5 different models with the industrial dataset. These figures show that our proposed model CPIN exceeds all the other models in the first epoch and reduces the fitting epoches. With the same datasets and learning rate, our model converges to the optimal performance earliest. Besides, there is more obvious improvements of CPIN on industrial dataset, which reveals that our

Table 3
Performance evaluation of different models on Amazon-books dataset.

Model	AUC	HR@10	NDCG@10
Wide&Deep	0.7776	0.5086	0.3369
PNN	0.7773	0.5078	0.3411
DeepFM	0.7782	0.5116	0.3423
DeepICF	0.6909	0.3933	0.2754
ComprehensivePart	0.7254	0.4333	0.3054
PresentPart	0.8022	0.5372	0.3542
CPIN-C	0.8166	0.5440	0.3606
CPIN-D	**0.8174**	**0.5480**	**0.3624**
CPIN-M	0.8124	0.5412	0.3606
CPIN-D\ancMLP	0.7815	0.5077	0.3413

Table 4
Performance evaluation of different models on industrial dataset.

Model	AUC	HR@10	NDCG@10
Wide&Deep	0.8046	0.5297	0.3108
PNN	0.8577	0.6250	0.3868
DeepFM	0.8316	0.5475	0.3367
DeepICF	0.9031	0.7037	0.4922
ComprehensivePart	0.9071	0.7089	0.4934
PresentPart	0.9020	0.6281	0.3640
CPIN-C	0.9088	0.7098	0.4982
CPIN-D	**0.9322**	**0.7263**	**0.5113**
CPIN-M	0.9253	0.7164	0.5086
CPIN-D\ancMLP	0.9096	0.6928	0.4982

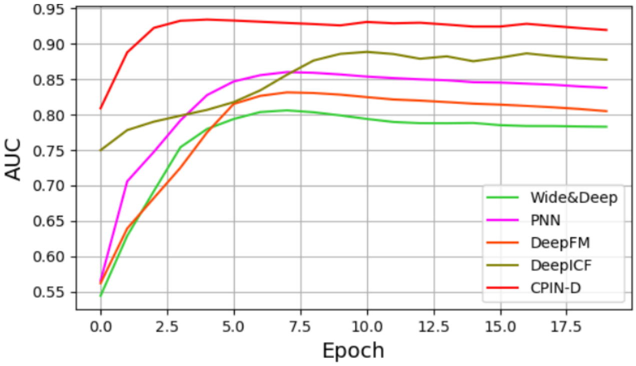

Figure 5: The CTR prediction performance of different models on inductrial dataset

proposed model CPIN handles sparse dataset better than state-of-the-art methods.

4.3.2. *Effectiveness verification of different part of interests.*

We split our model into 2 single interest based models to evaluate whether jointly learning comprehensive and present interest could improve the performance. The results shown in Table 2 and Table 3 show that present interest based model performs better than comprehensive interest based model on public dataset. It reveals that in the movie watching and book purchasing scenarios the target item prediction is more dependent on the sequential information in the interaction history. The model pays more attention to the order of the interacted items. While in the job recommending scenario, the comprehensive interest contributes more in job recommendation. It is because that the user's interest of seeking jobs does not change strictly following the sequence of the jobs that the user sought. Most people do the similar jobs in their whole life, and the order of jobs affects little in their interest. We could infer that the contributions of the comprehensive interest and the present interest differ in different scenarios. However, the CPIN model jointly learning the comprehensive interest and the present interest performs better than the models with any single part of it, which means the situation that the user's sequential interest could be decreased exists in the datasets, and CPIN could handle this situation well.

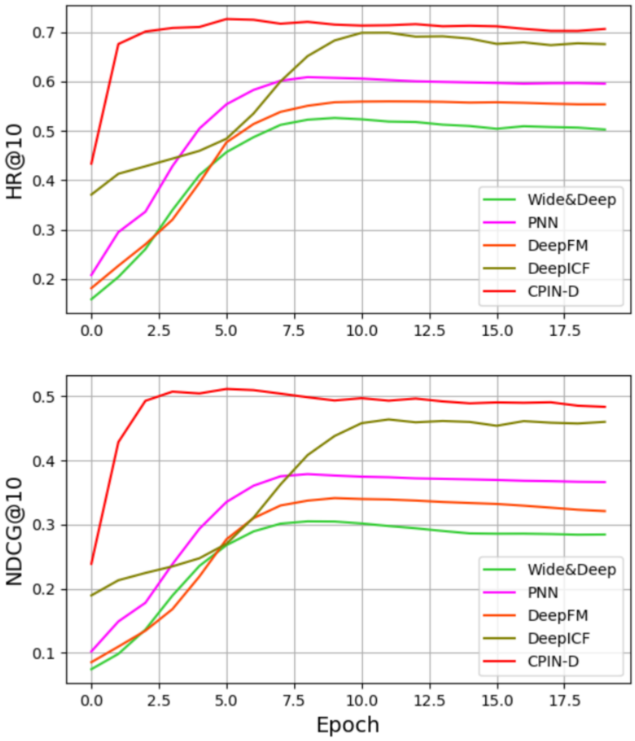

Figure 6: The Top-N recommendation performance of different models on industrial dataset.

4.3.3. *Effectiveness verification of the ancillary MLP*

In this subsection, we will verify the effectiveness of the ancillary MLP. The ancillary MLP enlarged the dataset indirectly by splitting the interaction sequence into several subsequences. The subsequence and its following item form a pair of positive training sample. By this way, the ancillary MLP helps training the parameters of the GRU and the embedding matrix. Table 2 ∼ 4 show that the ancillary MLP improves the performance of CPIN. The improvement on Amazon-books and industrial datasets is more obvious comparing with the result on Movielens-20M dataset, which reveals that ancillary MLP helps training the model with sparse data. The results verified our point that the ancillary MLP improves the training of the GRU parameters and the embedding matrix, leading to the more correct representation of the user's interest and the feature of the items.

5. Conclusion

In this paper, we put forward the idea that the user's interest contains both present interest and comprehensive interest, and we proposed a novel recommending model CPIN to jointly train the user's present and comprehensive interest. This model is proposed to balance the dependency of the target item on these two kinds of interests. We hypothesize that the user's comprehensive remains stable for some time, then we can split the interaction sequence into several subsequences to enlarge the dataset. And we built an ancillary MLP like the main MLP to help training parameters of the GRU and the embedding matrix using these subsequences. The user's comprehensive interest in the training of the ancillary MLP also contains some future information and helps the training. We conducted experiments on three public and industrial datasets to evaluate the performance of state-of-the-art models and our proposed model CPIN, and our model achieved the best performance on all three public and industrial datasets. The CPIN model with the ancillary MLP got improvements of 0.94%, 4.6%, 2.5% on AUC, 0.38%, 7.9%, 4.8% on HR@10 and 0.87%, 6.2%, 2.6% on NDCG@10 on three datasets respectively. The experiment results show that CPIN with both comprehensive interest and present interest performs better than the models with any single part of these two interests, which reveals that the situation that the user's sequential interest could be decreased exists in the datasets, and CPIN could handle it well. In the future, we plan to design a module to weight the effect of present interest and comprehensive interest on the recommending which could be regarded as a kind of explanation of the results, and introduce the profile of the users and the items to solve the cold start problem.

Acknowledgements

This work is partly supported by National Key R&D Program of China (Grant No. 2018YFC0830300), Science and Technology Program of Fujian, China (Grant No. 2018H0035), and Science and Technology Program of Xiamen, China (3502Z20183011).

References

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Chen, J., Zhang, H., He, X., Nie, L., Liu, W., & Chua, T.-S. (2017). Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 335–344).

Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M. et al. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 7–10).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, .

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 191–198).

Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, *22*, 143–177.

Golub, G. H., & Reinsch, C. (1971). Singular value decomposition and least squares solutions. In *Linear Algebra* (pp. 134–151). Springer.

Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, .

Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, *5*, 1–19.

He, R., & McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web* (pp. 507–517).

He, X., He, Z., Song, J., Liu, Z., Jiang, Y.-G., & Chua, T.-S. (2018). Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *30*, 2354–2366.

He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S. et al. (2014). Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising* (pp. 1–9).

Hong, W., Zheng, N., Xiong, Z., & Hu, Z. (2020). A parallel deep neural network using reviews and item metadata for cross-domain recommendation. *IEEE Access*, *8*, 41774–41783.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, .

Juan, Y., Zhuang, Y., Chin, W.-S., & Lin, C.-J. (2016). Field-aware factorization machines for ctr prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 43–50).

Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 426–434).

Lei, W., He, X., Miao, Y., Wu, Q., Hong, R., Kan, M.-Y., & Chua, T.-S. (2020). Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining* (pp. 304–312).

Li, H., Zhu, J., Liu, T., Zhang, J., Zong, C. et al. (2018). Multi-modal sentence summarization with modality attention and image filtering, .

Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, *7*, 76–80.

McAuley, J., Targett, C., Shi, Q., & Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval* (pp. 43–52).

Qu, Y., Cai, H., Ren, K., Zhang, W., Yu, Y., Wen, Y., & Wang, J. (2016). Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)* (pp. 1149–1154). IEEE.

Rendle, S. (2010). Factorization machines. In *2010 IEEE International Conference on Data Mining* (pp. 995–1000). IEEE.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175–186).

Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, *40*, 56–58.

Richardson, M., Dominowska, E., & Ragno, R. (2007). Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web* (pp. 521–530).

Sachdeva, N., Gupta, K., & Pudi, V. (2018). Attentive neural architecture incorporating song features for music recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems* (pp. 417–421).

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285–295).

Symeonidis, P., Tiakas, E., & Manolopoulos, Y. (2011). Product recommendation and rating prediction based on multi-modal social networks. In *Proceedings of the fifth ACM conference on Recommender systems* (pp. 61–68).

Tang, D., Qin, B., Liu, T., & Yang, Y. (2015). User modeling with neural network for review rating prediction. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Wu, C., Wu, F., An, M., Huang, J., Huang, Y., & Xie, X. (2019). Neural news recommendation with attentive multi-view learning. *arXiv preprint arXiv:1907.05576*, .

Wu, Y., DuBois, C., Zheng, A. X., & Ester, M. (2016). Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (pp. 153–162).

Xue, F., He, X., Wang, X., Xu, J., Liu, K., & Hong, R. (2019). Deep item-based collaborative filtering for top-n recommendation. *ACM Transactions on Information Systems (TOIS)*, *37*, 1–25.

Wenxing Hong received Ph.D. degree in System Engineering from Xiamen University, China, in 2010. He is the member of IEEE and CCF. Since 2010, he has been with the Department of Automation, Xiamen University, China, where he is currently an Associate Professor. He is the author or co-author of more than 25 papers. He has led and participated in more than 15 research projects and funds, including National Natural Science Foundation of China. His current research interests are in the area of data mining, big data, artificial intelligence, recommendation system, and FinTech. He is the dean of Research Center for Systems and Control, Xiamen University. He has been served as the general secretary of the International Conference on Computer Science and Education (ICCSE) and Fujian Systems Engineering Society, from 2006 and 2010, respectively.

Ziang Xiong is a graduate student at the Department of Automation, Xiamen University, China. He obtained his bachelor of engineering from Xiamen University, China, in 2018. His current research interest is deep learning and recommender system and he is working at the System and Control Center Laboratory, Xiamen University.

Jinjie You is a grauate student at the Department of Automation,Xiamen University, China. He obtained his bachelor of engineering from Fuzhou University, China, in 2019. His current research interest is recommender system and blockchain and he is working at the System and Control Center Laboratory, Xiamen University.

Xiaolin Wu is a graduate student at the Department of Automation, Xiamen University, China. She obtained her bachelor of engineering from Xiamen University, China, in 2018. Her current research interest is data analysis and recommender system and she is working at the System and Control Center Laboratory, Xiamen University.