DeepAPF: Deep Attentive Probabilistic Factorization for Multi-site Video Recommendation

Huan Yan, **Xiangning Chen**, **Chen Gao**, **Yong Li** and **Depeng Jin**

Beijing National Research Center for Information Science and Technology (BNRist),
Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China
liyong07@tsinghua.edu.cn

Abstract

Existing web video systems recommend videos according to users' viewing history from its own website. However, since many users watch videos in multiple websites, this approach fails to capture these users' interests across sites. In this paper, we investigate the user viewing behavior in multiple sites based on a large scale real dataset. We find that user interests are comprised of cross-site consistent part and site-specific part with different degrees of the importance. Existing linear matrix factorization recommendation model has limitation in modeling such complicated interactions. Thus, we propose a model of Deep Attentive Probabilistic Factorization (DeepAPF) to exploit deep learning method to approximate such complex user-video interaction. DeepAPF captures both cross-site common interests and site-specific interests with non-uniform importance weights learned by the attentional network. Extensive experiments show that our proposed model outperforms by 17.62%, 7.9% and 8.1% with the comparison of three state-of-the-art baselines. Our study provides insight to integrate user viewing records from multiple sites via the trusted third party, which gains mutual benefits in video recommendation.

1 Introduction

In recent years, watching online videos has become increasingly popular in our daily activities [Ding *et al.*, 2018]. This creates a large ecosystem with different content providers (CPs). In this ecosystem, CPs offer a variety of video contents for users. Since users freely watch their liked videos from various video sites, the competition for users is critically fierce among CPs. To achieve success, each CP takes efforts to provide the best video service. One of the most important strategies is designing smart personalized recommender systems to help users explore the videos of interest.

Most of recommender systems use one-site viewing data to predict user preferences on videos [Zhou *et al.*, 2010; Qian *et al.*, 2014; Bu *et al.*, 2016]. In particular, some recent works apply Deep Neural Networks (DNNs) into the recommendation methods [Zhang *et al.*, 2016; Zheng *et al.*, 2017; Zhang *et al.*, 2017; Chen *et al.*, 2017; Wang *et al.*, 2017; Zhu *et al.*, 2018; Zhang *et al.*, 2018; Zhou *et al.*, 2018; Gao *et al.*, 2019; Fang *et al.*, 2019]. For example, Covington *et al.* [Covington *et al.*, 2016] propose a deep neural network for YouTube video recommendation. Song *et al.* [Song *et al.*, 2016] model user temporal behavior by recurrent neural network (RNN) in recommender systems. Similarly, authors in [Gao *et al.*, 2017] design a dynamic RNN to capture user temporal preferences in the video recommendation. Although most of them take efforts to improve recommendation performance, two important factors are neglected. First, many users visit multiple video sites to watch their favorite videos. Thus, it is insufficient to model user viewing behaviors in a single site. Second, some videos are distributed among multiple sites. It is common that one site may push videos that users have viewed in another site. By collaborating with the Internet Service Provider (ISP) in China, we obtain over 205 million user-video viewing records from six popular video sites. This provides us an opportunity to explore how user preferences distribute over multiple sites, and how multi-site data can be beneficial to the performance improvement of video recommendation. Yang *et al.* [Yang *et al.*, 2017] give a preliminary analysis on multi-site user viewing behaviors. However, it only linearly learns the user-video interactions by inner product used in Matrix Factorization (MF) model, which cannot capture the complex interactions [He *et al.*, 2017].

In this paper, we aim to accurately capture user preferences as well as model the complex user-video interactions based on the multi-site viewing records, which is very challenging. First, since MF has its limitation to model the complicated user-video interaction, how can we properly learn them? Second, users are free to watch videos in different sites, thus what are the features of user interests over multiple sites? How can we model them? Third, the importances of different features of user interests may be not equal with each other, and thus how to accurately learn them is the final challenge.

Inspired by [He *et al.*, 2017], we use deep learning method to model the complex user-video interaction. Based on our analysis, we find that the cross-site commonality and site peculiarity are two main features of user interests. Specifically, in our dataset, users who have visited multiple sites (named *multi-homed users*) do not have consistent or independent user interests across sites. Videos that appear in multiple sites (named *multi-homed vides*) imply the consistence of user in-

terests; while exclusive videos in each site contribute to the disparity of user interests. To learn the importances of these two features, we introduce the attention mechanism that is initially proposed to solve static control problem in neural machine translation [Bahdanau *et al.*, 2014]. Finally, we propose a model of Deep Attentive Probabilistic Factorization (DeepAPF). It has ability to accurately capture user interests by learning the importances between cross-site commonality and site peculiarity, and approximate the complex interaction between users and videos to improve the recommendation performance.

Moreover, we conduct extensive experiments to evaluate the performance of our proposed model by comparing with different baseline models with that of our model based on the dataset from all six sites as well as from any pairs of two sites. The results show that DeepAPF achieves the best performance of the recommendation. Meanwhile, we also evaluate the impact of using attention mechanism. Compared with DeepAPF without attention, it can further improve the performance. Since using multi-site data is helpful for improving the recommendation performance, it can create the win-win chances for CPs to share viewing data via the trusted third party.

We summarize our main contributions as the following three aspects:

- We analyze the features of user preferences on videos over multiple sites. From the observations, we find that user-video interaction is complex. For multi-homed users, user preferences contain both common part and site-specific part with different importance for different users.
- We design a model of DeepAPF to accurately capture user preferences (including cross-site commonality and site peculiarity) as well as learn the complex user-video interactions in multiple sites.
- We conduct extensive experiments to evaluate the performance of our proposed DeepAPF. The results show that it achieves the best performance with the comparison of baselines.

2 Data Collection and Motivation

2.1 Dataset

We obtain the video viewing dataset via the collaboration with a major Internet Service Provider (ISP) in China. The dataset is collected at gateways deployed in the fixed networks of a large metropolis in China. Then, it is parsed by deep package inspection (DPI) appliances that have ability to parse the application layer protocol of data packets like Hyper Text Transfer Protocol (HTTP). Considering the protection of user privacy, ISP has anonymized the user information in the viewing logs before handing to us.

Overall, our dataset contains over 8 million users, 7.5 million videos and 205 million viewing logs between Nov. 1 and Dec. 31, 2014. Each log is comprised of the anonymized user identity (ID), access time and request Uniform Resource Locator (URL). By crawling the video URLs, we collect the basic information of viewed videos including the title, the type and the website where a video is requested. Since the access to large international video websites like YouTube is blocked by Great Firewall in China, we focus on 6 major domestic content providers (CPs) including Youku (YK), IQiyi (IQI), Sohu (SH), Kankan (KK), LeTV (LE), and Tencent Video (TC).

	# Users (10^3)	# Videos (10^3)	# Views (10^3)
YK	4,479	1,936	90,647
SH	3,156	174	39,927
IQI	3,156	169	24,721
LE	2,798	111	24,091
TC	2,781	130	17,883
KK	1,351	59	8,455

Table 1: General statistics of users, videos and views in 6 sites

In addition, the videos are classified into 6 major types, which includes TV series, movie, news, cartoon, user generated content (UGC) and show. This classification is based on the video types labeled by each site. To distinguish the multi-homed videos, we match their titles according to the specific naming rules. Specifically, based on our observation, we find that the CP's name is usually located at the beginning of the video titles. With this rule, we preprocess the video titles, and then check whether any two titles are the same. In this way, we accurately and effectively identify the multi-home videos.

Table 1 shows the number of users, videos and views in the 6 sites from our dataset. We observe that YK attracts most users to watch over 1 million videos; while KK has lowest user population in video viewing. This shows the differences exists between CPs.

Since users are free to watch videos in any sites, we can classify the users into two types of exclusive and multi-homed. For an exclusive user, she/he only visits one site for video viewing. When a user visits more than one site, he/she is multi-homed. Similarly, the same videos may be uploaded in multiple sites, thus we would identify the exclusive and multi-homed videos. Overall, we obtain about 5 million multi-homed users (over half of total users), which indicates that it is common that users visit more than one site for video viewing. For multi-homed videos, although its number is small (3% of total videos) but they can draw 25% of views.

2.2 Motivation

Our goal is to effectively use multiple-site data to improve the performance of video recommendation, which has three challenges needed to address.

Challenge 1: Characterizing the complicated interaction between users and videos

Matrix Factorization (MF) has become popular in many practical recommender systems [Hu *et al.*, 2018; He *et al.*, 2016]. It maps users and videos into a common latent space, where a latent feature vector can be used to represent a user or a video. Then, the interaction $\widehat{s}_{ij}$ between user i and video j can be formulated as

$$\widehat{s}_{ij} = \mathbf{u}_i^T \mathbf{v}_j = \sum_{l=1}^{L} u_{il} v_{lj}, \tag{1}$$

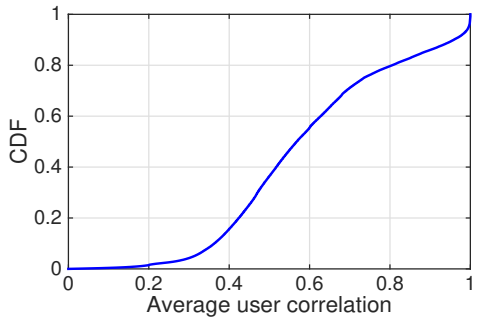

Figure 1: Distribution of average user correlation.

Pearson Correlation	YK	SH	IQI	LE	TC	KK
KK	0.6013	0.5858	0.5854	0.6310	0.4996	1
TC	0.5226	0.5422	0.5398	0.5409	1	
LE	0.7187	0.7513	0.6344	1		
IQI	0.5969	0.6064	1			
SH	0.6767	1				
YK	1					

Table 2: Average Site-Site Correlation between Two Sites.

where $\mathbf{u}_i$ and $\mathbf{v}_j$ represent the L-dimensional latent feature vector of user i and video j, respectively. It can be seen that MF linearly model such interaction by calculating inner product of associated latent vectors. However, as analyzed in [He *et al.*, 2017], it cannot sufficiently explore the complicated interactions between users and videos. Thus, how to characterize the complex interactions is a challenging problem.

Challenge 2: Finding the features of user interests across multiple sites

Most recommender systems use one-site viewing data to make video recommendation, which lacks the global information of user interests across different sites. One way to resolve this problem is to merge the multiple-site viewing data, which models user interests as one set of latent feature vectors that are site-agnostic. However, it is known that different sites have their own unique features. For example, TC, as a large social media company, actively pushes news videos at the news portal though the social network. To explore whether such differences exist, we perform the following empirical analysis.

We define a feature vector $\mathbf{p}_u(i)$ to represent user interests of user u in site i, where each element is the number of views from a specified video type in this site. Since the videos are classified into six types in our dataset, the dimension of the feature vector is set to 6. For multi-homed users, we can obtain their feature vectors in different sites. Next, we compute Pearson product-moment correlation between $\mathbf{p}_u(i)$ and $\mathbf{p}_u(j)$ of multi-homed users.

In our experiment, we consider the multi-homed users who have more than 100 views, because inactive users are insufficient to reflect user interests. For a multi-homed user u, we average his/her correlation coefficients of different two-site pairs, and plot the results in Figure 1. We find that most of multi-homed users do not have consistent or independent preferences across sites. Further, we calculate the average site-site correlation, and list the results in Table 2. We observe that in each site-site pair, the correlation is neither 0 or 1, which indicates that user interests have the features of the cross-site commonality and site peculiarity. On one hand, different CPs have their specialized features (*i.e.,* P2P downloading, social network construction) in popularizing their video services. On the other hand, multi-homed videos make it possible to share user interests across sites. Thus, the cross-site commonality and site peculiarity of user interests co-exist, and they are critically important for the design of our model. However, it is challenging to model them.

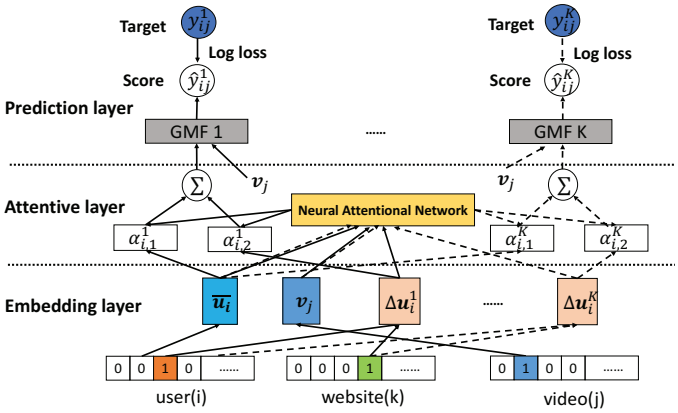

Figure 2: Deep attentive probabilistic factorization model.

Challenge 3: Estimating the feature importances of user interests

As discussed above, the features of user interests contain two parts: the cross-site commonality and site peculiarity. Intuitively, they contribute differently for user interests. For example, as TC that pushes the news through social network, when predicting a user's interest on a news video in this site, it is reasonable that the importance of site peculiarity should be higher than that of cross-site commonality. Thus, the importances between the cross-site commonality and site peculiarity are non-uniform for different users, and needs to be dynamically adjusted. However, how to accurately learn the importance weights is another challenging problem.

3 Method

To solve the three challenges discussed in Section 2, we propose a model of DeepAPF for video recommender systems using the multi-site data. Figure 2 illustrates its architecture which includes the following three layers.

- **Embedding layer**. The sparse representations of users and videos are projected to the dense vectors. The user embedding is decomposed into two parts: common part and site-specific part, which addresses the second challenge.
- **Attentive layer**. To overcome the third challenge, we make use of an neural attentional network to estimate the varying importances of the two features of user interest across sites.

- **Predict layer**. We use a generalized matrix factorization (GMF) model [He *et al.*, 2017] to learn the complex user-video interaction, which solves the first challenge.

3.1 Embedding Layer

First, we convert the identity of a user (i) and a video (j) and a site (k) to a sparse vector using one-hot encoding, which are formulated as follows,

$$\mathbf{c}_i^{\mathcal{U}} = \text{one}-\text{hot}(i), \mathbf{c}_j^{\mathcal{V}} = \text{one}-\text{hot}(j), \mathbf{c}_k^{\mathcal{K}} = \text{one}-\text{hot}(k), \tag{2}$$

where function $\text{one}-\text{hot}(i)$ can generate a vector of all zero values except i-th element with value 1. $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{K}$ are the sets of users, videos and sites, respectively.

Then, the sparse vectors are projected to the dense ones. Since the cross-site commonality and site peculiarity co-exist for multi-homed users, the embedding vector $\mathbf{u}_i^k$ of user i at site k is decomposed into the common part $\bar{\mathbf{u}}_i \in \mathbb{R}^L$ and the site-specific part $\Delta\mathbf{u}_i^k \in \mathbb{R}^L$, which are computed as follows,

$$\bar{\mathbf{u}}_i = \mathbf{P}^T \mathbf{c}_i^{\mathcal{U}}, \Delta\mathbf{u}_i^k = \Delta\mathbf{u}_i \mathbf{c}_k^{\mathcal{K}} = \mathbf{X}_k^T \mathbf{c}_i^{\mathcal{U}}, \tag{3}$$

where $\mathbf{P} \in \mathbb{R}^{|\mathcal{U}|\times L}$ and $\mathbf{X}_k \in \mathbb{R}^{|\mathcal{U}|\times L}$ are learnable parameters. We denote the total number of users and sites as $|\mathcal{U}|$ and $|\mathcal{K}|$, respectively. $\Delta\mathbf{u}_i \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{K}|}$ is combined by $|\mathcal{K}|$ site-specific embeddings of user i.

Meanwhile, we use $\mathbf{v}_j$ to denote the L-dimensional video latent feature vector, which is computed as follows,

$$\mathbf{v}_j = \mathbf{Q}^T \mathbf{c}_j^{\mathcal{V}}, \tag{4}$$

where $\mathbf{Q} \in \mathbb{R}^{|\mathcal{V}|\times L}$ are parameters to be learned. $|\mathcal{V}|$ is the total number of videos.

3.2 Attentive Layer

Since the cross-site commonality and site peculiarity differently contribute to user preference, we use a neural attention network that is widely used in natural language processing [Bahdanau *et al.*, 2014] to estimate their influences. Specifically, we use the attention parameter $\alpha_{i,m}^k (m = 1, 2)$ to represent the importance of the common and site-specific part. This attention network uses $\bar{\mathbf{u}}_i$, $\Delta\mathbf{u}_i^k$ and $\mathbf{v}_j$ as input, and exploits a multi-layer perceptron (MLP) to estimate the attention function f, which is formulated as

$$f(\mathbf{u}_i, \mathbf{v}_j) = \mathbf{h}^{\mathbf{T}} ReLU(\mathbf{W}(\mathbf{u}_i \odot \mathbf{v}_j) + \mathbf{b}), \tag{5}$$

$$\alpha_{i,1}^k = \frac{\exp(f(\bar{\mathbf{u}}_i, \mathbf{v}_j))}{\exp(f(\bar{\mathbf{u}}_i, \mathbf{v}_j)) + \exp(f(\Delta\mathbf{u}_i^k, \mathbf{v}_j))}, \tag{6}$$

$$\alpha_{i,2}^k = 1 - \alpha_{i,1}^k, \tag{7}$$

where $\odot$ is the element-wise product of vectors. $\mathbf{W}$ denotes the weight matrix in the attention network. $\mathbf{b}$ and $\mathbf{h}$ represents the bias vector and a weight vector, respectively. For simplicity, we use $\mathbf{u}_i$ to denote $\bar{\mathbf{u}}_i$ or $\Delta\mathbf{u}_i^k$. $ReLU(\cdot)$ is used as the activation function of Rectifier Linear Unit (ReLU) [Cao *et al.*, 2018] for the hidden layer in the attention network. Inspired by [He *et al.*, 2018], to avoid the large variance on attention weights for users, we smooth the softmax function used in (6), which is as follows

$$\alpha_{i,1}^k = \frac{\exp(f(\bar{\mathbf{u}}_i, \mathbf{v}_j))}{[\exp(f(\bar{\mathbf{u}}_i, \mathbf{v}_j)) + \exp(f(\Delta\mathbf{u}_i^k, \mathbf{v}_j))]^\beta}, \tag{8}$$

where β is the smoothing parameter, which ranges from 0 to 1. Especially, $\beta = 1$ represents the standard softmax function, and $\beta < 1$ would alleviate the punishment on the attention weights of active users. According to Equation (6) and (8), we obtain the user embedding at site k

$$\mathbf{u}_i^k = \alpha_{i,1}^k \times \bar{\mathbf{u}}_i + \alpha_{i,2}^k \times \Delta\mathbf{u}_i^k. \tag{9}$$

3.3 Prediction Layer

At prediction layer, we use the user embedding $\mathbf{u}_i^k$ and video embedding $\mathbf{v}_j$ as input, and define a mapping function as:

$$\phi_k^{GMF} = \mathbf{u}_i^k \odot \mathbf{v}_j. \tag{10}$$

Then, we estimate the prediction score $\widehat{y}_{i,j}^k$, which represents how likely user i prefers video j. We formulate it as:

$$\widehat{y}_{i,j}^k = \sigma(\mathbf{h'}^T \phi_k^{GMF}), \tag{11}$$

where $\sigma(\cdot)$ and $\mathbf{h'}^T$ denote the sigmoid function and the weight vector of the prediction layer, respectively.

3.4 Training

To learn the model parameters, we employ the log loss, an objective function, to minimize the distance between predicted score and target one, which is defined as:

$$L = -\sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{Z}} y_{i,j}^k \log \widehat{y}_{i,j}^k + (1 - y_{i,j}^k) \log(1 - \widehat{y}_{i,j}^k), \tag{12}$$

where $\mathcal{Z}$ denotes the training set that contains the positive and negative samples generated from the viewing records. The log loss of Equation (12) is optimized by mini-batch Adam optimizer [Kingma and Ba, 2014] to learn the model parameters. To construct a mini-batch, we sample the positive samples from the viewing records, and negative samples from unobserved videos for the corresponding users. With the constructed mini-batch, we perform gradient descent approach to minimize the objective function.

3.5 Discussion

MPF is demonstrated to achieve better performance with comparison of existing popular approaches such as Merged Matrix Factorization (MMF) [Yang *et al.*, 2017]. Now we prove how MPF can be regarded as a special case. According to the one-hot encoding of user or video identity, we can obtain the embedding vector that can be regarded as the latent feature vector. At the attentive layer, we enforce α_1^k be 0.5, thus $\mathbf{u}_i^k = \bar{\mathbf{u}}_i + \Delta\mathbf{u}_i^k$. Then, in Equation (11), we redefine $\sigma(\cdot)$ as the identify function, and use a full-one vector for $\mathbf{h}'$. Under these steps, the MPF model can be exactly recovered.

4 Experimental Design and Results

We now conduct several experiments to compare the recommendation performance of our DeepAPF model with the state-of-the-art recommendation models.

4.1 Experimental Design and Setting

In the experiments, we mainly focus on multi-homed users to evaluate the DeepAPF model, and videos with less than 100 views are removed. As our dataset only contains implicit feedback of video viewing, we use the top-N recommendation to measure the performance of our proposed model. Specifically, in the test set, the score $\widehat{y}_{ij}^{s}$ that user i prefers video j in site s can be predicted. We sort $\widehat{y}_{ij}^{s}$ by the descending order to obtain the top-N recommendation list, and choose the first N videos to recommend.

Then we describe how to generate the training set and test set. In each set, there are both positive samples and negative samples. We treat the viewing records as the positive samples, and users not viewing the videos as the negative samples. In the training set, the negative samples are 1.5 times as many as the positive samples per user. In the test set, the ratio of the negative samples over the positive samples per user is set to 10. To make the prediction rational, users and videos in the test set also appear in the training set.

Our methods are implemented based on Tensorflow. We use the Gaussian distribution with a mean of 0 and a standard deviation of 0.01 to initialize the embedding and hidden parameters. We set the dimension size L of the latent feature vectors to 64, and set the batch size to 256. The learning rate is initially set to 0.01. We tune the hyper-parameter β in the attention network to achieve the near optimal performance. Note that in this work, we omit testing different dimension sizes since the previous work [He *et al.*, 2017] has discussed it.

Baselines. We compare our proposed DeepAPF model with the following methods:

- **GMF**. [He *et al.*, 2017] This approach has ability to learn the complex user-item interaction of each site based on DNNs.
- **MPF**. [Yang *et al.*, 2017] This is the state-of-the-art matrix factorization approach that captures both cross-site commonality and site peculiarity over multi-site data.
- **DeepMMF**. This approach merges the user viewing records from all sites, and uses the GMF method to learn one set of the site-agnostic user latent features and video latent features.

Evaluation Metric. Following [Yang *et al.*, 2017], we adopt F-measure as the evaluation metric in our experiments. Thus, F-measure value can be formulated as

$$\text{F-measure} = \frac{\sum_{i \in \mathcal{U}^{\text{test}}} |\mathcal{V}_i^{\text{test}} \cap \mathcal{V}_i^{\text{rec}}|}{\sum_{i \in \mathcal{U}^{\text{test}}} |\mathcal{V}_i^{\text{test}}|}, \tag{13}$$

where $\mathcal{U}^{\text{test}}$ denotes the user set in the test set. $\mathcal{V}_i^{\text{test}}$ and $\mathcal{V}_i^{\text{rec}}$ are the set of positive test samples and the set of top N recommended videos for user i, respectively. Because we draw N as the same as the number of positive test sample per user, the F-measure value is equal to the precision value and recall value. Therefore, it is effective to evaluate the model performance. As future work, we plan to use more metrics (*e.g.*, Normalized Discounted Cumulative Gain (NDCG)) in our experiments.

F-measure	**MPF**	**DeepMMF**	**GMF**	**DeepAPF**
YK	0.8805	0.9277	0.9075	**0.9600**
SH	0.7688	0.8124	0.7979	**0.8871**
IQI	0.7134	0.8100	0.8453	**0.8866**
LE	0.8054	0.8539	0.8133	**0.9081**
TC	0.7177	0.8084	0.8204	**0.8855**
KK	0.6907	0.7769	0.7950	**0.8672**
Overall	0.7628	0.8315	0.8299	**0.8972**

Table 3: F-measure of 6 Sites by DeepAPF and other baselines.

Improved Rate	YK	SH	IQI	LE	TC	KK
YK	\	2.92%	3.90%	3.52%	5.99%	4.71%
SH	3.46%	\	2.35%	2.44%	7.15%	2.00%
IQI	5.46%	3.21%	\	3.89%	7.68%	8.69%
LE	3.66%	2.80%	2.93%	\	5.33%	4.15%
TC	5.97%	7.60%	8.35%	6.66%	\	5.65%
KK	0.68%	2.43%	10.82%	2.48%	6.73%	\

Table 4: Site-site Improved Rate of F-measure by DeepAPF over MPF.

Improved Rate	YK	SH	IQI	LE	TC	KK
YK	\	3.64%	0.49%	1.19%	1.97%	7.54%
SH	5.49%	\	0.39%	0.03%	1.11%	2.89%
IQI	4.84%	0.69%	\	1.85%	2.69%	4.33%
LE	1.97%	0.59%	0.18%	\	1.80%	1.31%
TC	4.94%	6.26%	7.25%	7.41%	\	3.30%
KK	2.49%	2.21%	8.32%	4.53%	1.04%	\

Table 5: Site-site Improved Rate of F-measure by DeepAPF over GMF.

Improved Rate	YK	SH	IQI	LE	TC	KK
YK	\	2.28%	3.80%	3.50%	6.31%	4.71%
SH	2.96%	\	1.99%	2.07%	6.74%	7.89%
IQI	4.87%	2.43%	\	3.19%	6.23%	6.07%
LE	3.54%	2.21%	2.42%	\	5.30%	1.93%
TC	6.44%	5.69%	6.66%	5.91%	\	5.92%
KK	3.17%	8.32%	7.76%	1.14%	6.86%	\

Table 6: Site-site Improved Rate of F-measure by DeepAPF over DeepMMF.

Improved Rate	YK	SH	IQI	LE	TC	KK
YK	\	1.66%	0.11%	0.17%	1.37%	4.71%
SH	1.89%	\	0.44%	0.08%	0.30%	4.00%
IQI	0.33%	0.36%	\	0.33%	0.47%	1.92%
LE	0.10%	0.04%	0.19%	\	0.22%	2.48%
TC	1.46%	0.9%	1.32%	0.78%	\	0.96%
KK	2.26%	3.1%	3.80%	4.15%	1.37%	\

Table 7: Site-site Improved Rate of F-measure by DeepAPF with Attention over DeepAPF without Attention.

4.2 Experimental Results

We first measure how the overall recommendation performance of DeepAPF perform with the comparison of the baselines based on the multi-site data.

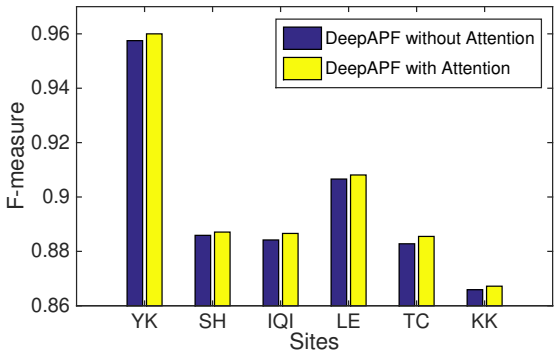

Figure 3: The F-measure of DeepAPF with attention and without attention mechanism for each site by using other 5 sites' data.

Table 3 shows the performance of our DeepAPF model and other three baselines for each site based on the data from other 5 sites. We observe that the average F-measure of our DeepAPF model is 17.62%, 7.9% and 8.1% higher than MPF, DeepMMF and GMF, respectively. The results show that our model can more accurately capture the user preferences over multi-site data, and thus achieve the significant improvements. In addition, although DeepMMF merges the data from other sites, it does not always perform well for each site when compared with GMF. This shows that blindly merging the data cannot effectively achieve the performance gain. In contrast, it enlarges the approximate error due to the restriction of the model space of learning various patterns of each site.

Site-site Performance

To investigate how one site's recommendation performs with the help of the data from another site, we conduct the experiments with different pairs of sites.

First, we compute the improved rate of F-measure by DeepAPF over MPF, and show the results in Table 4. We observe that DeepAPF outperforms the MPF methods for all the site pairs. For example, the F-measure in KK improves 10.82% when collaborating with the data from IQI. This is because our proposed methods can learn the complex user-video interactions. Further, the improved rate exhibits the differences among different pairs of sites. This indicates that the features (*e.g.,* site peculiarity) in different sites have impact on the information transferring.

Next, we compare the F-measure achieved by DeepAPF with that by GMF, as shown in Table 5. It can be observed that DeepAPF achieves the improvement of F-measure over GMF, which indicates that using another site's data is beneficial for predicting user interests.

Finally, we make a comparison between the F-measure of DeepAPF and that of DeepMMF for each site by virtue of another site's data. From Table 6, we can see that the F-measure of DeepAPF is higher than that of DeepMMF for each site. This is because that DeepAPF simultaneously captures both the common part and site-specific part of user preferences.

The Impact of Attention Mechanism

To explore the impact of attention mechanism in video recommendation, we conduct two experiments to compare the F-measure of DeepAPF with attention with that of DeepAPF without attention.

Table 7 shows improved rate of F-measure by DeepAPF with attention over DeepAPF without attention with the help of another site's data. From the results, we observe that for some site pairs (*e.g.*, SH-LE pair) using attention mechanism achieves a limited improvement of F-measure; while for KK, our model with attention has a great improvement by virtue of another site's data. This indicates that KK that attracts least views is highly sensitive to the importance between the site-peculiarity and cross-site commonality.

Figure 3 depicts the F-measure of DeepAPF with attention and without attention mechanism for each site by virtue of the data from other 5 sites. From the results, we find that using attention mechanism can improve the performance of recommendation.

Summary and Implications

Based on the above experiments, we observe 1) the F-measure of DeepAPF outperforms other three factorization models by 17.62%, 7.9% and 8.1%; 2) by virtue of only one-site data, our proposed model achieves the best performance; 3) the DeepAPF with attention outperforms that without attention. These give us a valuable and insightful guideline that integrating the data from multiple sites is win-win for CPs. However, due to the data privacy of different CPs, it is difficult to directly share the one CP's data with another. A potential solution is to introduce a trusted third party in charge of data collection as well as model training. When the trained model is ready, the third part distributes it to each site to perform the video recommendation.

5 Conclusion

In this paper, we propose a model of DeepAPF that not only accurately captures both the cross-site and site-specific user interests, but also learns the complex user-video interactions. Extensive experimental results demonstrate that our model outperforms three state-of-art baseline methods. In future, we will focus on the privacy issue of data sharing between sites. We plan to study how to improve our model without directly using other sites' data, which can better achieve win-win situation for different sites.

Acknowledgements

This work was supported in part by the National Nature Science Foundation of China under 61861136003, 61621091 and 61673237, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.

References

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Computer Science*, 2014.

[Bu *et al.*, 2016] Jiajun Bu, Xin Shen, Bin Xu, Chun Chen, Xiaofei He, and Deng Cai. Improving collaborative recommendation via user-item subgroups. In *IEEE Trans. on Knowl. and Data Eng.*, volume 28, pages 2363–2375, 2016.

[Cao *et al.*, 2018] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. Attentive group recommendation. In *Proceedings of ACM SIGIR*, pages 645–654, 2018.

[Chen *et al.*, 2017] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *Proceedings of ACM SIGIR*, pages 335–344, 2017.

[Covington *et al.*, 2016] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of ACM Conference on Recommender Systems*, pages 191–198, 2016.

[Ding *et al.*, 2018] Jingtao Ding, Yanghao Li, Yong Li, and Depeng Jin. Click versus share: A feature-driven study of micro-video popularity and virality in social media. In *SDM*, pages 198–206, 2018.

[Fang *et al.*, 2019] Hui Fang, Guibing Guo, Danning Zhang, and Yiheng Shu. Deep learning-based sequential recommender systems: Concepts, algorithms, and evaluations. In *International Conference on Web Engineering*, pages 574–577. Springer, 2019.

[Gao *et al.*, 2017] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. A unified personalized video recommendation via dynamic recurrent neural networks. In *Proceedings of ACM International Conference on Multimedia*, pages 127–135, 2017.

[Gao *et al.*, 2019] Chen Gao, Xiangning Chen, Fuli Feng, Kai Zhao, Xiangnan He, Yong Li, and Depeng Jin. Cross-domain recommendation without sharing user-relevant data. In *The World Wide Web Conference*, pages 491–502. ACM, 2019.

[He *et al.*, 2016] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of ACM SIGIR*, pages 549–558, 2016.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of WWW*, pages 173–182, 2017.

[He *et al.*, 2018] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. Nais: Neural attentive item similarity model for recommendation. In *IEEE Transactions on Knowledge and Data Engineering*, volume 30, pages 2354–2366, 2018.

[Hu *et al.*, 2018] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of ICDM*, pages 263–272, 2018.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*, 2014.

[Qian *et al.*, 2014] Xueming Qian, He Feng, Guoshuai Zhao, and Tao Mei. Personalized recommendation combining user interest and social circle. In *IEEE Transactions on Knowledge and Data Engineering*, volume 26, pages 1763–1777, 2014.

[Song *et al.*, 2016] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. Multi-rate deep learning for temporal recommendation. In *Proceedings of ACM SIGIR*, pages 909–912, 2016.

[Wang *et al.*, 2017] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Item silk road: Recommending items from information domains to social users. In *Proceedings of ACM SIGIR*, pages 185–194, 2017.

[Yang *et al.*, 2017] Chunfeng Yang, Huan Yan, Donghan Yu, Yong Li, and Dah Ming Chiu. Multi-site user behavior modeling and its application in video recommendation. In *Proceedings of ACM SIGIR*, pages 175–184, 2017.

[Zhang *et al.*, 2016] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of ACM SIGKDD*, pages 353–362, 2016.

[Zhang *et al.*, 2017] Shuai Zhang, Lina Yao, and Aixin Sun. Deep learning based recommender system: A survey and new perspectives. In *arXiv preprint arXiv:1707.07435*, 2017.

[Zhang *et al.*, 2018] Yan Zhang, Hongzhi Yin, Zi Huang, Xingzhong Du, Guowu Yang, and Defu Lian. Discrete deep learning for fast content-aware recommendation. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 717–726. ACM, 2018.

[Zheng *et al.*, 2017] Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 425–434. ACM, 2017.

[Zhou *et al.*, 2010] Renjie Zhou, Samamon Khemmarat, and Lixin Gao. The impact of youtube recommendation system on video views. In *Proceedings of ACM SIGCOMM*, pages 404–410, 2010.

[Zhou *et al.*, 2018] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1059–1068. ACM, 2018.

[Zhu *et al.*, 2018] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1079–1088. ACM, 2018.