IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 23, 2019, accepted August 20, 2019, date of publication August 27, 2019, date of current version September 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937841

A Generic Framework for Learning Explicit and Implicit User-Item Couplings in Recommendation

QUANGUI ZHANG, LI WANG, XIANGFU MENG, KEDA XU, AND JIAYAN HU
School of Electronic and Information Engineering, Liaoning Technical University, Huludao 125105, China

Corresponding author: Quangui Zhang (zhqgui@126.com)

This work was supported in part by the Liaoning Province Science Foundation under Grant 20180550995, and in part by the National Science Foundation (NSF), China, under Grant 61772249.

ABSTRACT The nature of recommendation is Non-IID, which has potential in improving recommendation quality and addressing issues such as sparsity and cold start. However, existing many state-of-the-art methods assume users and items are independent and same distributed while ignoring complex coupling relationships within and between users and items, resulting in limited performance improvement. To solve this issue, this paper proposes a novel neural user-item coupling learning model, short for CoupledCF, based on non-IID learning for collaborative filtering. CoupledCF joint learns explicit coupling with CNN and implicit coupling with deepCF within/between users and items accompanying user/item side information for recommendation tasks. User/item side information contains of attribute-based and feature-based. For different user/item side information, we use different embedding methods to learn embedding representation. We conduct comparative experiments on (1) two datasets from MovieLens1M and Tafeng with attribute-based user/item information for Top-K recommendation. (2) two datasets from MovieLens1M and BookCrossing with attribute-based user/item information for rating prediction. (3) two datasets from Amazon Movies and TV (AMT) and Yelp for feature-based user/item information for Top-K item recommendation and rating prediction tasks. Empirical results on five available real-world large datasets prove our proposed CoupledCF model is able to obtain better recommendation accuracy compared with several mainstream approaches for recommendation: BMF, neural matrix factorization, Google's Wide&Deep network, DeepFM, convMF, and A^3NCF model.

INDEX TERMS Coupling Learning, convolutional neural network (CNN), collaborative filtering, deep learning, user-item couplings.

I. INTRODUCTION

In reality, coupling learning has great potential for building a deep understanding of the essence of business problems and handling challenges that have not been addressed well by existing learning theories and tools [1]. Any recommended items and users are non-IID, which may essentially disclose why a user likes (or dislikes) an item [2]. There exists essential connection between users and items on attributes or features. Accordingly, it is important to learn complex user-item coupling relationships in deep models based on non-IID learning.

In recommendation systems (RS), collaborative filtering (CF) is one of the most popular approaches to predict a new user whether to interact with an item (e.g., ratings) and recommend the top items which user may like by analyzing the relationships between users (or items) according to the past user behavior [3], such as ratings, reviews, clicking or purchasing behaviors on items. The user-item rating matrix shows the user's overall preference on items, in which each entry denotes the preference of a user on an item. The rating matrix is widely used as the main data source for recommendation study.

In practice, often the rating matrix is very sparse, i.e., most of its entries are absent. Therefore, it will encounter the common cold-start and low recommendation accuracy problem. The method of dimensionality reduction has been proposed, however, it does not fundamentally change the nature of the problem, more and more studies have combined auxiliary information with ratings for improving recommendation performance, such as in [4], the author proposes a joint prediction model that exploits both the user-item rating matrix

The associate editor coordinating the review of this article and approving it for publication was Huiyu Zhou.

and the item-based side information to build top-N recommendation. In [5], a novel generic coupled matrix was proposed which integrates the intra-coupled interactions within an attribute and inter-coupled interactions among different attributes.

The same as rating information, textual reviews contain a large number of information written by users such as user preferences and item characteristics. Several studies [6], [7] have shown the quality of recommendation can be improved by combining ratings and review texts, especially for the users and items with few ratings. Researchers have paid extensive attentions to learn user/item features from textual reviews, e.g., Latent semantic models such as the SVD-based latent semantic analysis (LSA) [8] and the probabilistically motivated latent dirichlet allocation (LDA) [9]. However, these methods ignore word order existing in reviews and require prior knowledge of the number of topics in the corpus [10]. Word2vec method was proposed by Google in 2013, a shallow neural network learns word embedding vector. Word2vec considers order and semantic information between words. We can directly average the vector of all words when learning representative vector from sentence or document. However, it ignores the influence of the order between words on sentence or text information. Doc2vec, which proposed by Le and Mikolov (2014), is an extension of Word2vec to extend the learning of embeddings from words to sentence, paragraph or document. It is applied to a document as a whole rather than individual words. Therefore, Doc2vec may be an appropriate method for embedding learning of sentence or document, i.e., user features and item features learned from review texts. In recent years, this method have extensive applied on Natural Language Processing (NLP) tasks, e.g., sentiment detection on the sentence-level or document-level.

Incorporating user/item auxiliary information into recommendation model has been extensive concerned in recent years.For example, a novel matrix factorization method [11], incorporating both rich bag-of-words type meta-data on items and user ratings simultaneously to enhance predictions and handle cold start problems. In [12], a factor analysis approach based on probabilistic matrix factorization integrates social contextual information and user-item rating matrix to alleviate the data sparsity and poor prediction accuracy problems. Li *et al.* [13] exploits the rich user information including a user's query history, purchasing and browsing activities to improve OCCF accuracy. In [14], the author leverages social relationships to model user preferences for recommendation. In [6], the Hidden Factors as Topics (HFT) model combines latent factors with latent review topics for rating prediction. Compared to several models which only use ratings or reviews, this method achieves significant improvements. The above work mainly involves specific user/item auxiliary information, such as contexts, user historical activities, user demographics or reviews, or item description into recommender system for addressing the problem of rating shortage and the challenges such as sparsity and cold start.

However, some of the above works simply integrate user or item side information into a recommendation model but ignore the various coupling relationships [15] within and between users and items and the non-IID nature of recommendation [1], [2] which may essentially disclose why a user likes (or dislikes) an item [2]. Existing many relevant approaches can only lead to limited improvement as they assume users and items are independent and identically distributed (IID). For example, most of the previous works regarding user/item information as IID cannot make the best use of user/item information to improve the recommendation accuracy when users/items are actually non-IID. In [2], Non-IID learning were introduced to content with the non-IID nature of users/items, i.e., learning couplings and heterogeneities within/between users and items, and some obviously achievements have been made in creating non-IID recommendation systems, e.g., coupled user/item similarity-based matrix factorization [5], [16] and in many other learning tasks [17]–[23].

While the above works consider the user-user and/or item-item couplings but do not jointly model explicit and implicit user-item couplings with their features and relationships for CF. It is very difficult to learn explicit and implicit user-item couplings in recommender system, as this involves high dimensional and diverse interactions between observable and latent user/item attributes [1]. In addition, deep neural networks such as convolutional neural network (CNN) has great potential in representing abstract features especially in image processing and natural language processing. In [24], a DeepCoNN model jointly learns item properties and user behaviors from review texts based on CNN. ConvMF [25] utilizes CNN to learn user and item embedding and incorporates it into probabilistic matrix factorization for rating prediction.

This work models explicit and implicit user-item couplings in recommendation for collaborative filtering, which reflects the various relationships between users and items. This paper proposes a coupled CF model, CoupledCF, which jointly learns and combines both explicit and implicit user-item couplings according to both deep local features learned by CNN and explicit global features describing users and items. The main contributions of this work are as follows:

- CoupledCF first learns the explicit user-item couplings w.r.t. user attributes/features and item attributes/features by a CNN-based user-item coupling learning network in which user features and item features are learned from review texts with Doc2vec model, then builds a deep CF (DeepCF) model to learn the implicit user-item couplings instead of traditional matrix factorization with dot-product w.r.t user latent features and item latent features, and finally integrates the learned explicit user-item couplings with DeepCF to systematically represent user, item and user-item couplings. To the best of our knowledge, CoupledCF is the first model to joint learn both explicit and implicit user-item couplings by CNN-based network and DeepCF.

- The CNN-based user-item coupling learning model consists of two components: a local CoupledCF which models the explicit user-item couplings by a convolution filter-based neural network (CNN) to capture local user-item interactions, and a global CoupledCF which combines local CoupledCF output with the user/item embedding product-based representation to capture the global user-item interactions.
- We co-train two neural networks: the local/global CoupledCF and DeepCF, to embed both explicit and implicit user/item attributes/features and relations into CF to jointly learn both explicit and implicit user-item couplings towards a comprehensive representation of user-item couplings.
- CoupledCF not only solves the cold start problems that are common in rating information by integrating user/item side information but also significantly improves the overall recommendation performance.

Empirical evaluation of various CoupledCF models: local CoupledCF, global CoupledCF, DeepCF, and their combination CoupledCF are conducted on three real-life large datasets with certain attribute-based user/item information and two large-scale available datasets with certain feature-besed user/item information. The results show all CoupledCF models outperform the baselines in evaluation metrics HR@K and NDCG@K for Top-K recommendation and RMSE and MAE for rating prediction; in particular, (1) CoupledCF with the attribute-based user/item information significantly beats neural MF [26] (by over 9.6% on NDCG@10), Google's Wide&Deep [27] network (by over 9.7% on NDCG@10), and DeepFM [28] model (by over 18% on HR@10) on MovieLens 1M and Tafeng data and (by over 25.15% on RMSE) on MovieLens1M and BookCrossing data. (2) CoupledCF with the feature-based user/item information significantly outperforms neural MF [26] (by over 34.68% on HR@10), convMF [25] (by over 31.32% on HR@10 and 5.97% on MAE), and A^3NCF [29] network (by over 36.71% on HR@10) on Amazon Movies and TV (AMT) and Yelp data.

The rest of this paper are organized as follows. Section 2 provides the related works. Section 3 describes the CoupledCF model in detail. Experiments and Evaluation are introduced in Section 4 to train the CoupledCF model and demonstrate improvements compared with the state-of-the-art methods. The conclusion and future work are given in Section 5.

II. RELATED WORKS

In academia, deep learning has shown great success in recommender systems, such as [17], [30]. Convolutional neural network (CNN), a widely-used deep neural network in computer vision [31], natural language processing [32], and abstract feature representation [25]. CNN demonstrates high potential in effectively representing local and abstract features in image or documents. Such as in [24] and [25], CNN is used to learn user and item feature representation from documents.

It is critical for learning feature interactions in recommendation systems. In recent years, deep neural networks have been widely used to learn feature interactions, such as DeepFM [28] and NCF [26], DeepFM combines the power of factorization machines for recommendation and deep learning for feature learning in a new neural network; In [26], a NeuMF model including GMF which models high-dimensional feature interaction and MLP that learns low-dimension feature interaction was proposed for collaborative filtering with implicit feedback. In our CoupledCF model, we construct and integrate a CNN-based user-item coupling learning network (local CoupledCF) and a deep CF model (deepCF) to co-learn both explicit and implicit user-item couplings, to the best of our knowledge, this cannot be done by existing work.

In order to obtain better recommendation accuracy, some methods have been involved user/item information, such as user/item attributes or featues, into CF. In [27], Wide&Deep learning jointly trains wide linear models which get the benefit of memorization by cross-product feature transformations and deep neural networks (DNN) for generalization of recommendation. In [33], a model based on matrix factorization integrates hierarchical information for pages and ads for response prediction. In [29], an A^3NCF model develops a new topic model to extract both user and item features from reviews to guide the aspect-aware representation learning and introduces an attention network to capture the varying attention vectors of each specific user-item pair. In this paper, we use user and item side information as the input of the CNN-based learning framework to model explicit user-item couplings.

There are various representative learning methods for categorical feature, e.g., user demographics and item genres. One-hot encoding is a popular method that is used to transform the categorical data to numerical representation. It encodes each categorical feature with a one-zero vector where per vector has value '1' corresponding to one value, and all the rest of the entries are '0'. However, one-hot encoding is high dimensional when there exists more categorical features and leads to data sparsity issue. Several embedding methods have been applied to represent the categorical data, such as [34], the author not only uses an effective embedding approach available for categorical data, but also pays attention to value coupling relationship. In this paper, we learn dense embedding vector for each categorical feature through neural network embedding layer if user/item side information is attribute-based.

Several effective embedding methods are suitable for textual data to express user/item features, such as latent semantic indexing (LSI) [8], latent Dirichlet allocation (LDA) [9], skip-gram [35], and their variants [36], [37]. However, LSI and LDA don't take the order between words into account. Skip-gram method learns word-level embedding vector. Doc2vec is a unsupervised learning algorithm that learns vector representation of sentence or docunment, there are two embedding learning ways: PV-DM and PV-DBOW.

PV-DM method is similar to the continuous bag of words method, however, in addtion to multiple target words, PV-DM approach introduces an supernumerary document token as input (such as the document token d represent document vector). The hidden layer concatenates or averages, which depends on the specific implementation, the document token and several input word vectors. The output is a prediction of specific word; PV-DBOW is simpler and trains faster, similiar to the skip-gram model [35], the input is a special token representing the document without context words. The output is target context with softmax function. In this paper, we use the PV-DM method to learn user/item features from review texts. By training the Doc2vec model, the document embedding that represents user/item features will be learned.

Non-independent and identical distribution (non-IID) essentially explains the reason that user preference on items. Some pioneering works have been introduced about learning the non-IIDness in recommendation [38], such as in CF [5], [16], [20], [39], and by statistical learning [22]. This paper integrates explicit coupling with CNN and implicit coupling with deepCF to propose user-item coupling model which builds on non-IID.

III. THE COUPLEDANCF MODEL

A. PRELIMINARIES

Suppose there are M users and N items in systems. Let $U = \{u^1, u^2, \ldots, u^M\}$ and $V = \{v^1, v^2, \ldots, v^N\}$ denote the user set and item set respectively. We construct user-item rating matrix $R^{M\times N}$ in which each entry $r_{ij} \in R^{M\times N}$ reflects the user i specific preferences on item j. In this paper, we conduct Top-K recommendation and rating prediction based on the rating matrix. For Top-K recommendation, we transform the explicit rating $r_{ij} = \{1, 2, \ldots, 5\}$ into interaction score $r_{ij} = 1$ and add negative samples where $r_{ij} = 0$. It is formulated as:

$$r_{ij} = \begin{cases} 1, & \text{if user i interacts with item j;} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The value of r_{ij} is 1 shows that user i interacts with item j, vice versa. In addition to above rating matrix $R^{M\times N}$, we also introduce the user/item side information, w.r.t (1) user/item attributes such as user demographics and item description. (2) user/item features learning from review texts, in the CoupledCF model.

Below, we demonstrate the explicit user-item coupling learning network with CNN in Section 3.2. Section 3.3 presents the deepCF model which models the implicit user-item couplings. Section 3.4 describes the framework of CoupledCF in detail. Section 3.5 shows the embedding learning method for attribute-based user/item information. Section 3.6 introduces the embedding learning method for feature-based user/item information. The user/item features are learned from review texts. The mathematical notations used in this paper are summarized in Table 1.

TABLE 1. Mathematical notations.

Symbols	Definitions and descriptions
R	rating matrix
U	user set
V	item set
u	an example of user set U
v	an example of item set V
p	latent vector of user u
q	latent vector of item v
u_c	the embedding vector of user information
v_c	the embedding vector of item information
W	the weight matrix of fully connected layer
b	the bias vector of fully connected layer
$y^{(i)}$	the ground truth of the i^{th} example of the training/test dataset
$\hat{y}^{(i)}$	the predicted value of the i^{th} example of the test dataset
f_p	the pooling function
g	non-linear activation function
X_c	the coupling matrix of user-item
D	document set which each document contains of several reviews.
d_{ui}	the i^{th} document included all the reviews written by u.
d_{vj}	the j^{th} document included all the reviews of item v written by all users.

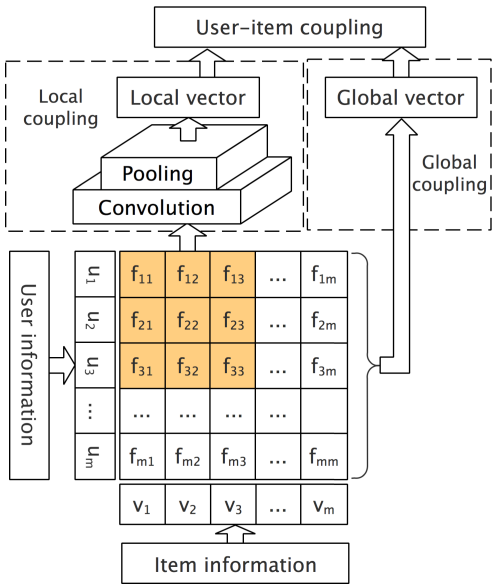

FIGURE 1. CNN-based local and global explicit user-item coupling learning by introducing user/item information.

B. CNN-BASED EXPLICIT USER-ITEM COUPLING LEARNING

Fig. 1 shows the user-item explicit coupling learning by introducing user auxiliary information containing user demographics and features which learns from review texts and item side information including item description and item features learning from review texts. User demographics and item description information are embed as dense vector by neural network embedding layer (as explained in section 3.5) and user/item features are embed as dense vector by Doc2vec model (as demonstrated in section 3.6). We uniformly represent the user/item dense vector as $u_c = \{u_{c1}, u_{c2}, \ldots, u_{cm}\}$ and $v_c = \{v_{c1}, v_{c2}, \ldots, v_{cm}\}$ respectively, in which $u_{ci}(v_{ci})$

represents the i^{th} element. The m is the dimension of u_c and v_c. For dense vector u_c and v_c, we construct the user-item coupling matrix X_c by the coupling function, e.g., $f_\theta(u_c\ v_c)$ that learns the coupling relationships between u_c and v_c. Each $x_{ij} \in X_c$ reflects the couplings between element u_{ci} of vector u_c and element v_{cj} of vector v_c. For coupling matrix X_c, we first feed it into convolutional neural network (CNN) to learn local user-item explicit couplings, forming the local CoupledCF model as shown in left box in Fig. 1, and output a local vector e_l and then flatten the coupling matrix X_c to global vector e_g, representing the global user-item couplings as shown in right box in Fig. 1. CNN includes convolution layer and pooling layer. The convolution layer learns abstract feature representation and it is formulated as:

$$a_c = g(W * X_c + b) \tag{2}$$

where W and b are convolution filters and corresponding bias vector. g is the non-linear activation function, in our CoupledCF model, we use Rectified Linear Units (ReLU) [40] as activation function, as it's calculation simple and convergence speed significantly outperforms other activation functions, such as sigmoid and tanh.

The pooling layer reduces the feature dimensionality, compressing the number of the parameters, reducing over-fitting, and improves the robustness of the model. It contains max-pooling and average-pooling. In this work, we adopt max-pooling, which extracts the maximum value as the feature corresponding to this particular convolutional filter. The max pooling operator is a non-linear subsampling function that returns the maximum of a set of values [41]. It reduces the dimension of the features and learns more abstract coupling feature vectors. The max pooling is performed as:

$$a_p = f_p(a_c) \tag{3}$$

where f_p is the max-pooling function and a_c is the output of convolution layer.

C. DEEPCF FOR IMPLICIT USER-ITEM COUPLING LEARNING

In this section, we model the user-item implicit couplings by constructing a deep collaborative filtering (deepCF) model, shown in Fig. 2. First, this model maps the latent user and item factors in the same embedding space according to user past rating behaviours [3]. The latent item factors may explain the explicit characteristics such as a movie's genre and/or the hidden features of items. The latent user factors reflect the degree that a user likes an item in terms of the corresponding latent factors. For example, a user likes a movie containing latent factors such as comedy and love, if a movie has these latent factors then the model will recommend this movie to the user. We encode the user and item identifies into one-hot vector o_u and o_v respectively. For example, suppose we have the user identifies: $\{0, 1, 2, \ldots, 9\}$, we transform these into one-hot matrix $O^{10\times10}$, each vector $o_i \in O^{10}$ denotes the i^{th} user with length 10. For user 0, the corresponding one-hot encoding vector o_0 can be represented as [10000000000],

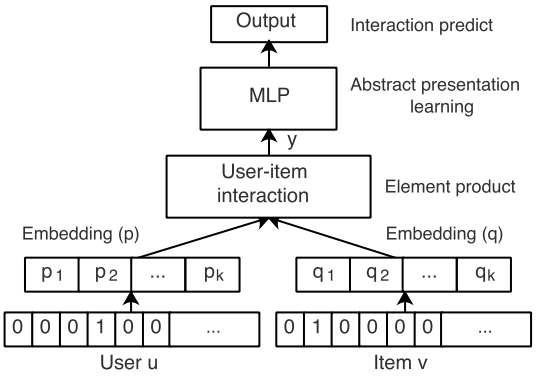

FIGURE 2. DeepCF: Learning implicit user-item couplings.

in which the number of the valid location is 1, others are 0. Similar to Skip-Gram [35] model, we map the one-hot vectors of both users and items o_u and o_v into lower-dimension dense vectors by using a neural fully-connected layer as the embedding layer, denoted as p and q respectively. The process of embedding learning is performed as:

$$\begin{aligned} p &= W_u^T o_u \\ q &= W_v^T o_v \end{aligned} \tag{4}$$

where $W_u \in R^{k\times|U|}$ and $W_v \in R^{k\times|V|}$ are weight matrices between the input fully connected layer and embedding layer.

Then, DeepCF maps both users and items to a common latent factor space with the same dimensionality k. Further, the embedding vectors p and q are fed into a multiplication fully-connected layer which conducts the element-wise product of p and q. It then outputs a linear interaction vector y which represents the linear user-item interactions. We formulate it as:

$$y = p \otimes q = (p_1q_1, p_2q_2, \ldots, p_kq_k) \tag{5}$$

We use multi-layer fully-connected neural network to replace traditional inner-products used in Matrix Factorization methods. The capacity and nonlinearity of deep neural network can learn better complex mapping relationships between users and items. Then, the user-item interaction vector y is fed into a multi-layer fully-connected neural network to deeply learn the high-level abstract user-item interactions. After training DeepCF by stochastic gradient descent algorithm, matrices W_u and W_v represent the latent factors for all users and items. With the one-hot encoded representation of users and items, each column of W_u and W_v represents a certain user or item latent factors p and q respectively. For a given item v, each dimension of v measures the extent to which the item has these factors. For a given user u, each dimension of u measures the extent of interest the user has in the corresponding factors of the item. Accordingly, the output vector y of the element product layer captures the linear interactions between users and items. The fully-connected layers further transform the output to represent the non-linear interactions between

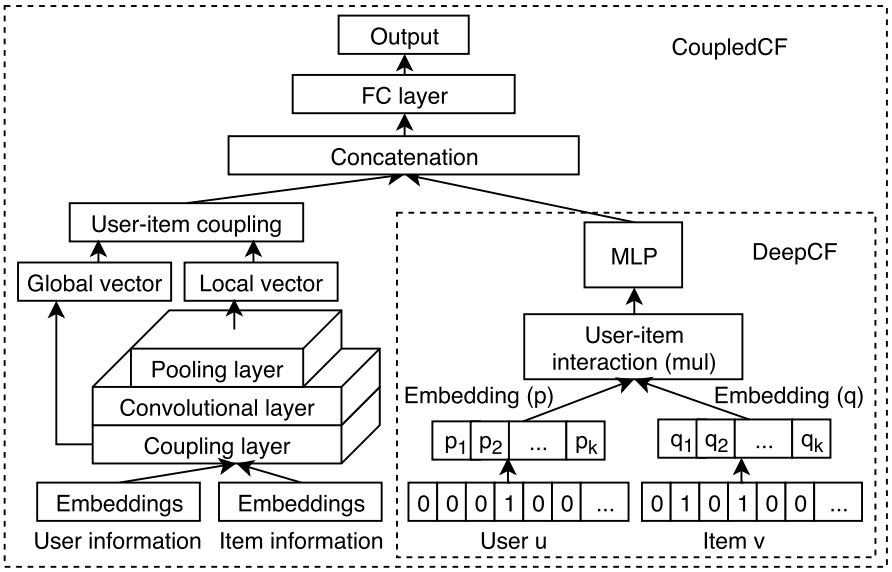

FIGURE 3. CoupledCF: Jointly learning explicit and implicit user-item couplings.

users and items. We formulate it as:

$$\begin{aligned} a_1 &= ReLU(W_1^T y + b_1) \\ a_2 &= ReLU(W_2^T a_1 + b_2) \\ &\cdots \\ a_L &= ReLU(W_L^T a_{L-1} + b_L) \end{aligned} \tag{6}$$

where L is the number of layers, $W_1, W_2, \ldots, W_L$ and $b_1, b_2, \ldots, b_L$ denote the weight matrices and bias vectors of each layer, and $a_1, a_2, \ldots, a_L$ denote the output of each layer activated by the ReLU function.

For Top-K recommendation, as CF models do not involve negative examples, we sample some negative examples by a negative sampling strategy, which is inspired by NCF [26], to make DeepCF discriminative. In our experiments, we sample negative examples from the unobserved interactions in the rating matrix $R^{M\times N}$ by a uniform negative sampling strategy. DeepCF further predicts the probability of user-item interactions (e.g., rating or not) by a logistic Sigmoid function to squash the model output into the interval [0, 1], where 1 indicates a user favors an item, otherwise 0, by converting the multi-scale ratings to binary. It interprets a user-item interaction prediction w.r.t. a probability:

$$P_\Theta(y = 1|u, v) \tag{7}$$

where Θ is the neural network weights. We use $\hat{y}$ as the output of model.

$$\hat{y} = g(W_0^T * a_L + b_0) \tag{8}$$

where g is the non-linear activation function w.r.t Sigmoid function for Top-K recommendation, W_0 and b_0 indicate the weight matrix and bias of the last layer.

For rating prediction problem, the model output a score $\hat{y} \in [0, 5]$ which shows the user's overall preference on item.

For an example in the training dataset D = $\{(u^{(i)}, v^{(i)}), y^{(i)}\}$ and the corresponding predicted output $\hat{y}^{(i)}$ (here i denotes the i^{th} example and $i \in \{1, 2, 3, \ldots, |D|\}$. For Top-K recommendation, the loss function is:

$$\zeta(\hat{y}^{(i)}, y^{(i)}) = -y^{(i)} \log(\hat{y}^{(i)}) - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \tag{9}$$

For rating prediction, the loss function is:

$$\zeta(\hat{y}^{(i)}, y^{(i)}) = \sum (\hat{y}^{(i)} - y^{(i)})^2 \tag{10}$$

We then learn the network parameters Θ per the following cost function by performing stochastic gradient descent algorithm with back propagation:

$$J = \frac{1}{m} \sum_{i=1}^{m} \zeta(\hat{y}^{(i)}, y^{(i)}) \tag{11}$$

D. COUPLEDCF: INTEGRATING EXPLICIT AND IMPLICIT USER-ITEM COUPLING LEARNING

In this section, we combine the CNN-based local/global explicit user-item coupling learning with the implicit user-item coupling learner DeepCF to build a comprehensive coupled CF model: CoupledCF. The left network in Fig. 3 implements the CNN-based user-item coupling learning, the user/item information dense vectors u_c and v_c are fed into the coupling layer. In our experiments, to simplify the learning, we define a user-item coupling calculation function below:

$$f_{ij} = u_{ci} * v_{cj} \tag{12}$$

Accordingly, the user-item coupling matrix X_c can be viewed as the cross-product of u_c and v_c. We execute two processes on the user-item coupling matrix X_c. First, X_c is fed into the CNN components to learn the local user-item coupling vector. Second, X_c is flattened as a vector to learn

the global user-item couplings. The local and global vectors are concatenated and then fed into a multilayer perceptron (MLP) network to learn highly abstract representation.

We further integrate the CNN-based local/global explicit user-item coupling learning with the DeepCF-based implicit user-item coupling learning by concatenating the output of these two networks. The concatenated vector (denoted as r) is processed by a fully-connected layer to generate the final user-item coupling vector. The integration of two neural networks generates the CoupledCF model. For Top-K recommendation, CoupledCF model finally outputs a user-item interaction score $\hat{y} \in [0, 1]$ by a Sigmoid non-linear activation function. The Sigmoid function is usually used as binary classification problem for logistic regression model. It squashes the output vector of the last neural layer to the range [0, 1]. The Sigmoid function is formulated as:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

For rating prediction, CoupledCF model finally outputs a user-item interaction score $\hat{y} \in [0, 5]$.

Hence, the training dataset of CoupledCF can be represented as $D = \{(u^{(i)}, v^{(i)}, u_c^{(i)}, v_c^{(i)}), y_i\}, i \in \{1, 2, \ldots, |D|\}$, the $|D|$ denotes the number of examples of training datasets, the objective of CoupledCF for Top-K recommendation is implemented by predicting the probability P_Θ:

$$P_\Theta(y = 1|u, v, u_c, v_c) \tag{14}$$

The final output $\hat{y}$ of coupledCF is formulated as:

$$\hat{y} = Sigmoid(W^T ReLU(W_0^T * r + b_0) + b) \tag{15}$$

For rating prediction the final output is performed as:

$$\hat{y} = W^T ReLU(W_0^T * r + b_0) + b \tag{16}$$

where W and b are the weight matrix and bias vector of the last layer of CoupledCF. The cost function in Eq.9 and Eq.10 are used to train CoupledCF.

E. THE EMBEDDING LEARNING METHOD FOR ATTRIBUTE-BASED USER/ITEM INFORMATION

For the attribute-based user/item information, such as user demographic information and item attributes, we use different methods to learn the embedding features for the categorical features and numerical features. For one categorical feature x_i which represented as an one-hot vector o_i, as shown in Fig. 4, we use a neural network fully-connected layer as the embedding layer to learn the dense real-valued embedding vector. The embedding vectors are initialized randomly and then the values are updated by training the CoupledCF model.

The embedding learning process is formulated as:

$$c_i = W_i^T * o_i \tag{17}$$

where $W_i \in R^{M \times N}$ is the weight matrix between input and embedding layer where M and N represent dimensions of embedding vector and one-hot vector respectively. For numerical features, We normalize the features then concatenate them with the categorical features' embeddings.

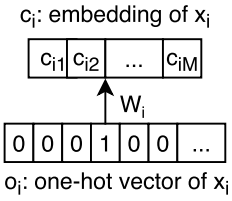

FIGURE 4. The embedding learning of one categrical feature x_i.

F. THE EMBEDDING LEARNING METHOD FOR REVIEW-BASED USER/ITEM INFORMATION

For review-based user/item information, we learn user/item embedding features using PV-DM method of Doc2vec model. Let $D = \{d_{u_1}, d_{u_2}, \ldots, d_{u_m}, d_{v_1}, d_{v_2}, \ldots, d_{v_n}\}$ be the input document set, where d_{u_i} is the document included the reviews to all items written by user u_i, and d_{v_j} is the document included all the reviews to item v_j written by all users.

1) Word2vec

Word2vec is an unsupervised learning method, which maps words into distributed vectors by a mapping function f. It formulates as:

$$R^m = f(w) \tag{18}$$

where w represents the word of dictionary, and R^m represents the m-dimensional distributed vector. The objective function of Word2vec is to maximise the log probability of context word (w_O) given its input word (w_I), i.e., $\log P(w_O|w_I)$ [42]. The parameters are constantly updated by training model. W matrix learned from Word2vec represents word embedding matrix, each column of W shows a word embedding.

2) Doc2vec

Slightly different from Word2vec, Doc2vec is a document-level embedding method, it adds a token besides word vectors, which is a unique, randomly initialized numerical vector, representing the document. The token is updated by training Doc2vec model. Ultimately, it can describe the topic of the document.

We use Doc2vec algorithm to learn user/item review embeddings (vector for representation of document) containing two stages: (1) Add unique token d_i, i represent the i^{th} document in the corpus, for each document d_{ui} or d_{vj}, because the document token is shared between all the words/contexts sampled in the document and the word vectors are shared among all the documents these words appear on, which can be considered as a memory function [10]. (2) We get matrix D by training the Doc2vec model, each column of D denotes a document vector representing user or item features. The matrix D is updated by adopting stochastic gradient descent algorithm with back propagation. The implementation of the Doc2vec model uses the Gensim Python package [43].

TABLE 2. Statistics of the datasets on MovieLens1M, Tafeng and BookCrossing.

Datasets	Users	Items	Ratings	Sparsity
MovieLens1M	6,040	3,952	1,000,209	95.81%
Tafeng	32,266	23,812	817,741	99.89%
BookCrossing	3,704	10,000	133,699	99.64%

IV. EXPERIMENTS AND EVALUATION

Here, we perform Top-K item recommendation and rating prediction on five available datasets and conduct a series of experiments to evaluate our proposed CoupledCF model and it's variations.

A. EXPERIMENTAL SETTINGS

1) DATASETS WITH ATTRIBUTE-BASED USER/ITEM INFORMATION

Three publicly large-scale datasets MovieLens1M[1], Tafeng[2] and BookCrossing[3] with consistent ratings and attribute-based user/item information.

2) MovieLens1M

It consists of 1M transactions from 6,040 users and 3,952 items, where each user has at least 20 interactions with items. User demographics contain Gender, Age, Occupation, and Zip code. Item attributes include movie's Genres.

3) TAFENG

It includes 817,741 ratings (from 1-5) with 32,266 users and 23,812 items. User demographics include Customer ID, Age, and Region. Item attributes contain Original ID, Sub class, Amount, Asset, and Price.

4) BOOKCROSSING

It contains 1,149,780 integer ratings (from 0-10) with 271,379 books and 278,858 users. User demographics include location and age. Item attributes contain book title, book author, the year of publication, and publisher.

For BookCrossing dataset, we first clear the user-item pairs where the rating is equal to 0 and then filter the dataset in the same way with MovieLens1M which has at least 20 user-item interactions. Finally, the subset remains 133,699 ratings with 3,704 users and 10,000 books.

The basic statistics of these datasets are shown in Table 2.

5) DATASETS WITH REVIEW-BASED USER/ITEM INFORMATION

Two available real-world datasets Amazon[4] and Yelp[5] review datasets, which include user reviews and ratings.

[1]https://grouplens.org/datasets/movielens/
[2]http://www.bigdatalab.ac.cn/benchmark/bm/dd?data=Ta-Feng
[3]https://grouplens.org/datasets/book-crossing/
[4]http://jmcauley.ucsd.edu/data/amazon/
[5]https://www.kaggle.com/yelp-dataset/

TABLE 3. The information of datasets on Amazon Movies and TV (AMT) and Yelp.

Datasets	Users	Items	Ratings	Sparsity
AMT	23,982	98,593	217,568	99.99%
Yelp	192,056	139,188	509,218	99.99%

6) AMAZON MOVIES AND TV (AMT)

The dataset has been widely used in content-based or hybrid recommendation. It includes 4,607,047 product reviews and ratings with 61,743 users and 155,459 items from Amazon website which spanning May 1996 - July 2014. The dataset has been removed the duplicate user-item pairs by the provider.

7) YELP

Yelp review dataset contains 4,700,000 reviews and ratings with 200,000 users and 156,000 items on kaggle website. The data spans 11 metropolitan areas.

For AMT and Yelp datasets, we extracted "userID", "itemID", "rating (1 to 5 rating stars)", and "review texts" for experiments. We merge all the reviews written by a user u to all items as a document which represents the user review and merge the reviews to item v written by all users as a document which represents the item review. We remove punctuation, stopwords, and infrequent times appearing less than 10 for each user/item review. Besides, we preprocess the datasets by keeping every user/item review with at least 50 words. We make per user has at least 5 ratings. Finally, the AMT dataset remains 217,568 reviews with 23,982 users and 98,593 items and Yelp dataset keeps 509,218 reviews with 192,056 users and 139,188 items. The basic statistics of these datasets are summarized in Table 3.

For Top-K recommendation, similar to [26], [44], we binarize the ratings in four datasets (MovieLens1M, Tafeng, AMT, and Yelp) to create implicit feedback for evaluation. Accordingly, we transform the original rating matrix scaled from $\tilde{R} \in \{1, 2, \ldots, 5\}$ into a binarized preference matrix $R \in \{0, 1\}$, in which each rating element is expressed as either 0 or 1, where 1 indicates an interaction between a user and an item; otherwise 0. After transforming the datasets to the implicit version, we uniform-randomly sample 4 negative instances for each positive instance.

8) BASELINE METHODS

Our CoupledCF model is customized to four variants below to learn various types of user-item couplings:

- DeepCF: only learns the user-item interactions based on latent user/item factors by deep neural network (Fig. 2);
- lCoupledCF: the lCoupledCF learns the local explicit user-item couplings by CNN-based neural framework and outputs a local representation vector. (the model with the local vector locates in left box in Fig. 1);
- gCoupledCF: the gCoupledCF model includes the global user-item coupling learning component without

TABLE 4. The baseline methods for Top-K and rating prediction.

		Baseline methods	Datasets
Top-K	Attribute-based	NeuMF;Wide&Deep;DeepFM	MovieLens1M;Tafeng
	Review-based	NeuMF;convMF;A^3NCF	Amazon Movies and TV(AMT);Yelp
Rating prediction	Attribute-based	BMF;DeepFM	MovieLens1M;BookCrossing
	Review-based	BMF;convMF	Amazon Movies and TV(AMT);Yelp

the CNN component (the model with the global vector lies in the right box in Fig. 1);
- CoupledCF: the CoupledCF consists of the local vector, global vector, and DeepCF vector.

The following relevant and representative state-of-the-art methods (shown as Table 4) are used as the baselines to evaluate our methods. For Top-K recommendation, NeuMF, Wide&Deep, and DeepFM are used to attribute-based user/item information. NeuMF, convMF, and A^3NCF are used to review-based user/item information. For rating prediction, BMF, DeepFM are used to attribute-based user/item information. BMF, convMF are used to feature-based user/item information.

- BMF [3]: a basic matrix factorization model, which maps both users and items to a joint latent factor space with the same dimensionality, the user-item interactions are modeled as inner products.
- NeuMF [26]: a CF method with implicit feedback embedded into a neural network. It confirms our method incorporating user/item side information performs better than the basic neural CF without user/item side information.
- Wide&Deep [27]: a benchmark Google's wide&deep neural network co-training wide linear models and deep neural networks, and combining the advantages of memorization and generalization for recommendation. The deep neural network can generalize to previously unseen feature interactions through low-dimensional embeddings while wide network can memorize sparse feature interactions using cross-product feature transformations. To be fair, we convert the relevant user/item side information such as user/item attributes into cross-product features, which are then entered together with the raw features into the Wide&Deep model. We compare the performance of our model embedded with explicit/implicit user/item information to this Wide&Deep model that uses refined cross-product features.
- DeepFM [28]: an extension of Wide&Deep, which combines the power of factorization machines for recommendation and deep learning for feature learning in a new neural network architecture. It emphasizes both low and high-order feature interactions and has a shared input to its ''wide'' and ''deep'' parts, with no need of feature engineering besides raw features compared to Wide&Deep. We compare the performance of CoupledCF which jointly learns implicit and explicit feature interactions to DeepFM model that learns low-order feature interactions like FM and models high-order feature interactions like DNN.
- convMF [25]: a novel context-aware recommendation model, convolutional matrix factorization (convMF) that integrates convolutional neural network (CNN) into probabilistic matrix factorization (PMF). It indicates our model which learns feature interactions by deep neural network (CNN and DNN) outperforms the convMF model which learns feature interaction by inner products.
- A^3NCF [29]: a benchmark deep neural network combining ratings and reviews for rating prediction, which extracts user preferences and item characteristics from review texts with topic model. It shows how well our CoupledCF model embedded with user/item features learning from review texts by training Doc2vec performs compared to A^3NCF model embedded with user/item features extracted from topic model.

For convMF and A^3NCF model in Top-K item recommendation, existing several methods rarely use both ratings and reviews to predict user-item interactions, therefore we change the explicit rating prediction to a binary classification task to predict the probability of user-item interaction.

For DeepFM model in rating prediction, we transform the binary classification problem with sigmoid activation function to regression problem without activation function.

B. MODELING SETTINGS

We implemented CoupledCF model using Python based on the Keras framework. All the baseline methods used in our paper are implemented following their Github experiment configuration. All the experiments are performed in a 3.5GHz NVidia Geforce 1080Ti GPU with 32GB memory. We randomly divide each dataset in Table 2 and Table 3 into training, validation, and testing sets and tune hyper-parameters of CoupledCF and baselines on validation data. By optimizing the loss of Eqs. 9 and 10, we get the optimal hyper-parameter settings of CoupledCF model. The hyper-parameters of CoupledCF on Top-K recommendation mainly include:

- The dimensionality of the embedding layers of the DeepCF model: We evaluate the number of embedding layers w.r.t {8, 16, 32, 64}, and obtain the best results when the number is 32 for four datasets (MovieLens1M, Tafeng, AMT, Yelp).
- The number of the embedding layers of the CNN-based user-item explicit coupling learning network: We evaluate it w.r.t. {8, 16, 32, 64, 128, 256}, and get the best performance on the model with 8 embedding layers for

TABLE 5. The interpretation of hyper-parameter in Doc2vec.

Hyper-parameter	Interpretation
Vector_size	Dimension of document vector
Window_size	Left/right context window size
Epochs	The iterative times of training
Min_count	Minimum frequency of words

MovieLens1M and Tafeng datasets and 64 embedding layers for Amazon Movies and TV (AMT) and Yelp review datasets.

- We construct two CNN layers, set the filter as (3, 3) and the channel as 8 for MovieLens1M and Tafeng datasets and set the filter as (3, 3) and the channel as 64 for Amazon Movies and TV (AMT) and Yelp review datasets.
- The dimensionality of the hidden layers before the last output layer of CoupledCF model: We evaluate it w.r.t. {8, 16, 32, 64, 128, 256}, and get the best performance on the model with 64 fully-connected layers for MovieLens1M and Tafeng datasets and 32 fully-connected layers for Amazon Movies and TV (AMT) and Yelp review datasets.
- The learning rate: We set it as 0.001 for MovieLens1M, 0.005 for Tafeng, 0.00001 for Amazon Movies and TV (AMT) and Yelp review datasets.
- The activation function of fully-connected layer: we use ReLU as the activation function for each fully-connected layer, The activation function is Sigmoid for the last layer of DeepCF and CoupledCF.
- The batch-size: We assess batch-size w.r.t {32, 64, 128, 256}, and get the best performance on the model with 256 for four datasets.
- The epochs: We set 100 for four datasets, however, model performance gets the best result in 30^{th} epoch approximately for MovieLens1M and Tafeng and around 10^{th} epoch for Amazon Movies and TV (AMT) and Yelp datasets. Although the training loss keeps degrading, model performance has been bad. At this time, model may result in overfitting, we ultilize earlystopping or dropout strategy to prevent it.
- We adopt batchnormalization and dropout strategy to prevent overfitting with the dropout ratio is 0.5.
- The hyper-parameter settings of Doc2vec: we set the vector_size is 100, the epochs is 100, window_size is 5, min_count is 10. The interpretation of these hyper-parameter is shown in Table 5.

Similar to Top-K recommendation, we test various hyper-parameters and show the best performance of CoupledCF model in Table 6 for rating prediction.

We use Adam as the optimizer for our model. Adam optimizer is an adaptive learning rate approach, which makes the parameters of model stable, and implements simply, calculates efficiently and memory requirements is low. For parameter initialization, we initialize the embedding matrix of user and item identifies with a random normal distribution with the mean and standard deviation are 0 and 0.01 respectively, and use glorot-uniform as the initializer for the fully-connected layers. All biases in this model are initialized with zero. In our proposed model, we use Batch Normalization after the Dense layers as it possesses following advantages: (1) it can improve the speed of model and effectively avoid the gradient disappearance and explosion by adopting high learning rate. (2) it is equivalent to dropout strategy to prevent overfitting.

1) TOP-K RECOMMENDATION EVALUATION METRICS

We use the widely-used leave-one-out performance validation to evaluate all the comparison methods for implicit feedback-based recommendation as in [45]. Similar to [46], we randomly sample one item with user-item interactions as the test item for each user and the remaining items with interactions as the training data. We randomly sample another 99 items which are not in the user's interacted item set to form the user's test data together with the above selected test item. We let each model to rank these 100 items for each user, and then evaluate the performance. We take the top-K Hit Ratio (HR@K), which measures whether the test item appears on Top-K list, and Normalized Discounted Cumulative Gain (NDCG) [26], which takes up the position of the hit by allocating higher scores to hits with top ranks, as evaluation metrics. We calculate evaluation metrics for each test user and inform the average score.

- HR@K: a recall-based measure, as,

$$HR@K = \frac{\#hits@K}{|GT|} \tag{19}$$

- NDCG: is a ranking-based measure, as,

$$NDCG@K = Z_k \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{20}$$

where GT denotes the test list set, rel_i is the graded relevance value of the item at position i and Z_K is the normalization. In our experiments, we set $rel_i \in \{0, 1\}$, which depends on whether i is in the test dataset.

2) RATING PREDICTION EVALUATION METRICS

Similar to most collaborative prediction algorithms [25], [39], we use widely-used for rating prediction in recommendation systems root mean squared error (RMSE) and mean absolute error (MAE) to evaluate the performance of CoupledCF model. The slower RMSE and MAE represent better performance of model.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y^{(i)}} - y^{(i)})^2} \tag{21}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y^{(i)}} - y^{(i)}| \tag{22}$$

where N denotes the number of test examples.

TABLE 6. The optimal settings of hyper-parameter in rating prediction.

Hyper-parameter \ Datasets	MovieLens1M	BookCrossing	AMT	Yelp
The dimensionality of the embedding layers of the DeepCF model	32	32	32	32
The number of embedding layers of the CNN-based network	16	8	32	64
Convolutional filters and channels	(3, 3), 16	(3, 3), 8	(3, 3), 32	(3, 3), 64
The dimensionality of the hidden layers before the last layer	32	64	16	32
The learning rate	0.0001	0.0001	0.00001	0.00001
The epochs	30	50	20	20
The batch_size	128	128	128	64

TABLE 7. HR@10 and NDCG@10 for Top-10 item recommendation for MovieLens1M and Tafeng.

	MovieLens1M		Tafeng	
	HR@10	NDCG@10	HR@10	NDCG@10
NeuMF	0.731	0.448	0.6519	0.4329
Wide&Deep	0.73	0.447	0.642	0.4522
DeepFM	0.6452	0.3703	0.6821	0.4522
deepCF	0.7147	0.4312	0.6506	0.4322
lCoupledCF	0.8212	0.5408	0.6798	**0.47**
gCoupledCF	0.7826	0.5252	0.6643	0.4205
CoupledCF	**0.8252**	**0.544**	**0.6953**	0.4623

TABLE 8. HR@10 and NDCG@10 for Top-10 item recommendation for Amazon Movies and TV (AMT) and Yelp.

	AMT		Yelp	
	HR@10	NDCG@10	HR@10	NDCG@10
NeuMF	0.4579	0.2554	0.4420	0.2635
A^3NCF	0.4609	0.2563	0.4217	0.2581
ConvMF	0.4848	0.2689	0.4756	0.2793
deepCF	0.4308	0.2356	0.4545	0.2512
lCoupledCF	0.5987	0.3423	0.7564	0.5421
gCoupledCF	0.4876	0.2975	0.4680	0.3429
CoupledCF	**0.6017**	**0.3567**	**0.7888**	**0.5629**

C. RESULTS AND ANALYSIS FOR TOP-K RECOMMENDATION

1) TOP-K ITEM RECOMMENDATION RESULTS

The evaluation results with HR@10 and NDCG@10 of our model and other baselines are summarized in Table 7 and Table 8. We also test the Top-K (K = 1, 2, …, 10) item recommendations in Fig. 5, Fig. 6, Fig. 7, and Fig. 8. CoupledCF is compared with it's variants containing local CoupledCF (lCoupledCF for short), global CoupledCF (gCoupledCF), and DeepCF as well as all the baseline methods. According to these figures, we can know the models get the best performance when the K is 10. The experimental results show that CoupledCF significantly improves recommendation performance, e.g., up to 12.68% improvement over NeuMF (HR@2), up to 15.27% improvement over Wide&Deep (HR@4), and up to 27.74% improvement over DeepFM (HR@6), and averaged 6.91% improvement over NeuMF, averaged 7.94% over Wide&Deep, and 11.78% over DeepFM on MovieLens1M and Tafeng datasets; up to 40.25% improvement over NeuMF (HR@2), up to 32.16% improvement over convMF (HR@7), and 35.03% improvement over A^3NCF (HR@3), and averaged 22.84% improvement over NeuMF, averaged 18.21% over convMF, and averaged 17.74% over A^3NCF on Amazon Movies and TV (AMT) and Yelp review datasets.

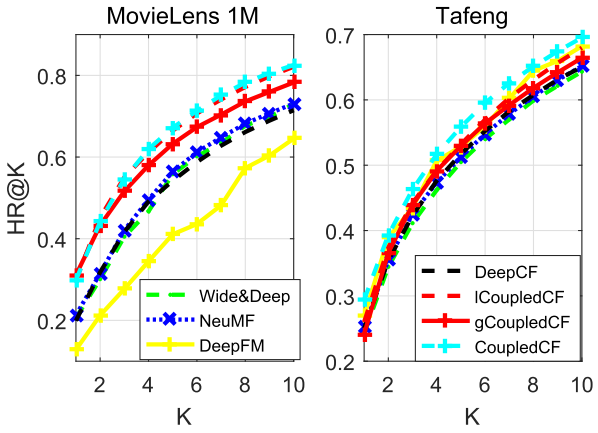

FIGURE 5. HR@K results of Top-K item recommendation on MovieLens1M and Tafeng.

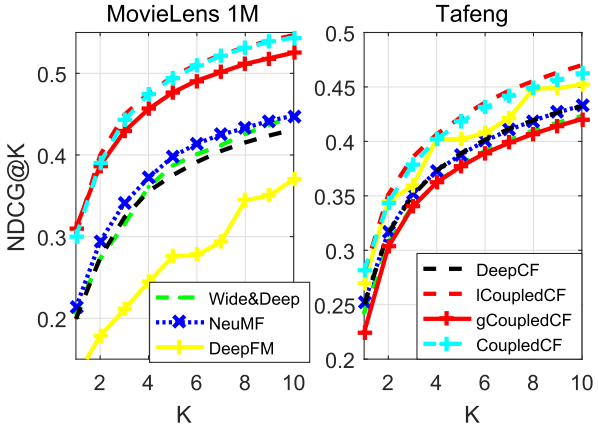

FIGURE 6. NDCG@K results of Top-K item recommendation on MovieLens1M and Tafeng.

2) COMPARISON WITH BASELINES

First, compared to neural MF model NeuMF, as shown in Table 7 and Table 8, CoupledCF beats NeuMF by 9.42% on MovieLens1M, 4.34% on Tafeng, 14.38% on Amazon Movies and TV (AMT) as well as 34.68% on Yelp w.r.t. on HR@10; and 9.60% on MovieLens1M, 2.94% on Tafeng, 10.13% on Amazon Movies and TV (AMT) and 29.94% on Yelp w.r.t. NDCG@10. It demonstrates our proposed CoupledCF model introducing user/item auxiliary information

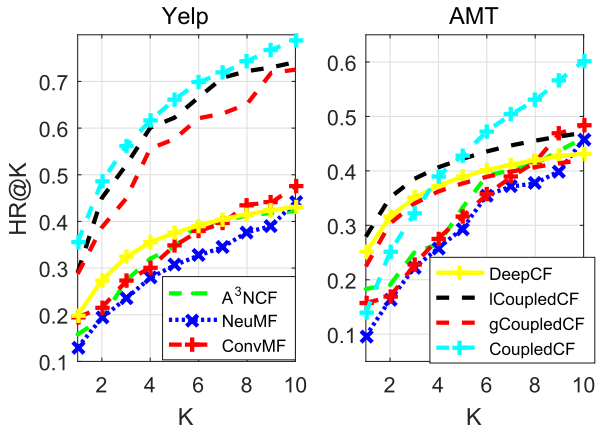

FIGURE 7. HR@K results of Top-K item recommendation on Yelp and Amazon Movies and TV (AMT).

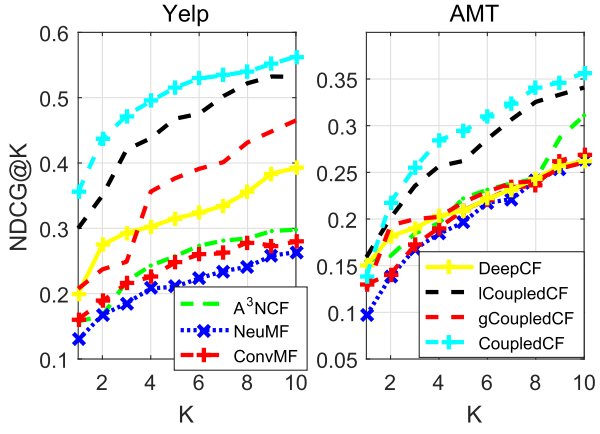

FIGURE 8. NDCG@K results of Top-K item recommendation on Yelp and Amazon Movies and TV (AMT).

significantly beats the NeuMF model without considering side information.

Second, compared to the Google benchmark Wide&Deep model, CoupledCF beats it by 9.52% on MovieLens1M and 5.33% on Tafeng w.r.t. on HR@10; and 9.70% on MovieLens1M and 1.01% on Tafeng w.r.t. NDCG@10. This shows the learning way of CoupledCF and incorporating explicit/implicit user-item coupling relationships performs well than the feature engineering-based Wide&Deep's.

Third, compared to the DeepFM model, CoupledCF beats it by 20.28% on MovieLens1M and 1.32% on Tafeng w.r.t HR@10; and 19.26% on MovieLens1M and 1.01% on Tafeng w.r.t NDCG@10. It shows the way of CoupledCF feature interaction (coupling learning) with CNN surpasses the way of DeepFM low and high-order feature interactions by a large margin.

Fourth, compared to the A^3NCF model, CoupledCF beats it by 14.08% on Amazon Movies and TV (AMT) and 36.71% on Yelp w.r.t. on HR@10; and 10.04% on Amazon Movies and TV (AMT) and 30.48% on Yelp w.r.t. NDCG@10. It indicates the way of CoupledCF learning and integrating explicit/implicit user-item interactions outperforms the feature Topic-based A^3NCF's.

3) TESTING THE CoupledCF EFFECTIVENESS

We further evaluate the working mechanism of CoupledCF in terms of different components embedded in the model. DeepCF, lCoupledCF, gCoupledCF are the variants of CoupledCF. As shown in Table 7 and Table 8 and Fig. 5, Fig. 6, Fig. 7, and Fig. 8, CoupledCF generally beats other variants for Top-K recommendations on four large available datasets.

By comparison, local CoupledCF beats DeepCF and global CoupledCF in all cases. For example, for Top-10 item recommendations, local CoupledCF beats DeepCF up to 10.65% improvement and surpasses global CoupledCF by 3.86% on Movielens1M, 2.92% and 1.55% on Tafeng w.r.t HR@10; local CoupledCF outperforms DeepCF by 10.96% and beats global CoupledCF by 1.56% on MovieLens1M, 3.78% and 4.95% on Tafeng w.r.t NDCG@10; local CoupledCF outperforms DeepCF by 16.79% and beats global CoupledCF by 11.11% on Amazon Movies and TV (AMT), 30.19% and 28.84% on Yelp w.r.t HR@10; local CoupledCF outperforms DeepCF by 10.67% and beats global CoupledCF by 4.48% on Amazon Movies and TV (AMT), 29.09% and 19.92% on Yelp w.r.t NDCG@10. This shows local CoupledCF which captures CNN-based explicit user-item coupling learning outperforms the implicit user-item coupling neural network (DeepCF) and indicates local CoupledCF learns feature interaction by CNN-based learning network outperforms the global CoupledCF that flattens the coupling matrix as global vector.

From the Fig. 5, Fig. 6, Fig. 7, and Fig. 8 which show the Top-K recommendation results on four datasets where $K = \{1, 2, 3, \ldots, 10\}$, we can observe that both the local CNN-based component and the global coupling learning component contribute to improve the recommendation performance, while the comprehensive CoupledCF model integrating local/global attribute-based/review-based user-item couplings and implicit user-item couplings generally gains the best performance. It once again proved the deep neural network CNN could model complex interactions between users and items to effectively improve the recommendation accuracy.

As shown in Fig. 9 and Fig. 10, we observe that the best performance on HR@10 and NDCG@10 in around 30^{th} epoch on MovieLens1M and Tafeng datasets, and in 10^{th} iteration roughly on Amazon Movies and TV (AMT) and Yelp datasets. More training times may cause model overfitting, we can take earlystopping method to prevent it when validation loss starts to rise.

D. RESULTS AND ANALYSIS FOR RATING PREDICTION

The rating prediction results of our CoupledCF model and other baseline methods on four datasets are given in Table 9 and Table 10.

From the above results we can observe that:

First, compared to BMF model which only uses the user-item rating matrix as input without considering side information, our CoupledCF model integrating review texts performs better than it by 16.8% on AMT and 28.54% on Yelp w.r.t

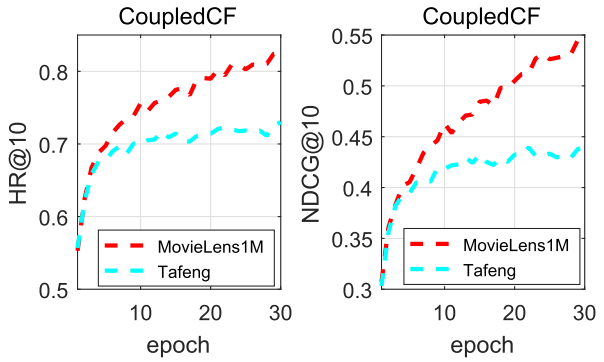

FIGURE 9. HR@10 and NDCG@10 performance w.r.t. the number of epoch on MovieLens1M and Tafeng datasets.

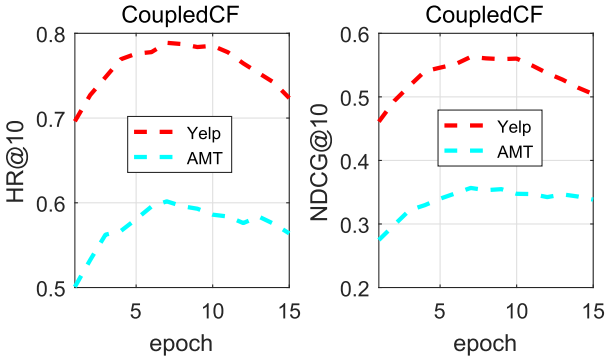

FIGURE 10. HR@10 and NDCG@10 performance w.r.t. the number of epoch on Yelp and Amazon Movies and TV(AMT) datasets.

TABLE 9. RMSE and MAE for rating prediction on MovieLens1M and BookCrossing datasets.

	MovieLens1M		BookCrossing	
	RMSE	MAE	RMSE	MAE
BMF	1.1334	0.7929	0.9200	0.5215
DeepFM	0.9504	0.7452	0.8437	0.4104
deepCF	0.9304	**0.7264**	0.8013	0.2705
lCoupledCF	1.1484	0.9462	0.9450	0.5752
gCoupledCF	0.9406	0.7467	0.8064	0.4074
CoupledCF	**0.9295**	0.7285	**0.5922**	**0.2198**

TABLE 10. RMSE and MAE for rating prediction on Amazon Movies and TV (AMT) and Yelp datasets.

	AMT		Yelp	
	RMSE	MAE	RMSE	MAE
BMF	1.1290	0.7929	1.5452	1.3580
convMF	0.9879	0.7233	1.3191	1.0237
deepCF	1.1045	0.7632	1.2963	1.0095
lCoupledCF	1.0057	0.7324	1.2630	0.9943
gCoupledCF	1.0109	0.7425	**1.2479**	0.9897
CoupledCF	**0.9610**	**0.6636**	1.2598	**0.9768**

RMSE, and 12.93% on AMT and 38.12% on Yelp w.r.t MAE, and integrating attribute information beats it by 20.39% on MovieLens1M and 32.78% on BookCrossing w.r.t RMSE, and 6.44% on MovieLens1M and 30.17% on BookCrossing w.r.t MAE. This is not surprising, as side information (review texts and attribute information) are able to complement the missing values in the rating information. It can solve cold-start and data sparsity issue and learn better user and item features to improve prediction accuracy.

TABLE 11. Case study for recommendation list given a user on MovieLens1M.

User favorite item genres	The Top-3 recommendation item genres of our model
Animation, Children's, Musical	Crime, Thriller
Animation, Children's, Musical	Action, Animition, Sci-Fi, Thriller
Adventure, Children's, Drama	Animition, Musical

TABLE 12. Case study for recommendation list given a user on Yelp.

The key words in review on user favorite items	The key words in the review on Top-3 recommendation items of our model
coffee, tea, drink, sweet, great, design	dessert, drink, tea, dish, place, service
cake, place, taste, tea, sweet, dessert	rice, dish, service, chicken
delicious, dish, service, brunch;	drink, definitely, service, chicken, burger

Second, compared to BMF model which utilizes traditional matrix factorization techniques, CoupledCF and DeepFM that use deep learning collaborative filtering (CF) perform better than it. It shows the popular deep learning CF algorithm has been far beyond the traditional MF methods. As deep learning could model users and items in a no-linear way and deep neural network, e.g., CNN has shown huge potential in representing abstract features [24], and deep neural network can automatically learn the parameters of the network through the random gradient descent algorithm.

Third, compared to DeepFM model which models low and high-dimensional feature interactions, our model that joint learns explicit-implicit feature interactions outperforms it by 2.09% on MovieLens1M and 25.15% on BookCrossing w.r.t RMSE, and 1.67% on MovieLens1M and 19.06% on BookCrossing w.r.t MAE. This shows our model learning explicit-implicit feature interactions outperforms the DeepFM that only learns the explicit feature interactions.

Fourth, compared to convMF which utilizes CNN to extract item features from review texts, our CoupledCF that learns user and item features with Doc2vec model exceeds it by 2.69% on AMT and 5.93% on Yelp w.r.t RMSE, and 5.97% on AMT and 4.69% on Yelp w.r.t MAE, although they all use review texts, the performance of model mainly depends on the embedding learning way.

E. ILLUSTRATIVE EXAMPLES

Here, we demonstrate the illustrative examples of our recommendation model. The attribute-based user/item information is shown in Table 11 and review-based user/item information is shown in Table 12.

To be simply, we only show the first three items of the recommendation list. For each item, we demonstrate the attribute information such as item genre and item description and show textual reviews that is the most frequent words. We can see that the attributes and features learning from reviews can better describe the characteristics of the items. For example, for most of movies, we can get many available information from their genre. By reading the movie genre, we know that

TABLE 13. The comparison of conference and journal.

	Conference	Journal
Side information	Attribute-based	Attribute-based
		Review-based
Solve problem	Top-K	Top-K
		Rating prediction
Datasets	MovieLens1M	MovieLens1M
		Tafeng
	Tafeng	**BookCrossing**
		Amazon Movies and TV (AMT)
		Yelp
Baselines	NeuMF	NeuMF
		Wide&Deep
		DeepFM
	Wide&Deep	**ConvMF**
		A^3NCF
		BMF

some movies are interesting Animation and some are relaxing music.

Through above two cases, we can see that our CoupledCF model successfully captures the user preferences and provides reasonable recommendations. Specifically, from the three movies that user likes, we speculate the user likes animation and romantic music, our model recommend three related movies which accords with user preferences. The same reason, for yelp restaurant, we can know the user likes drink and sweets from the three examples that user interacts and recommendations reflect the user's taste.

Finally, the illustrative examples confirm the effectiveness of our proposed model which joint learns the user-item explicit and implicit couplings via incorporating the user/item side information into CF.

V. CONCLUSION AND FUTURE WORK

The nature of recommendation is non-IID. This work joint learns explicit and implicit user-item interactions for recommendation: user/item attribute-based or feature-based user-item interactions by CNN, implicit user-item interactions by MLP, and their integration. We propose a coupled deep collaborative filter: CoupledCF to learn and combine the above user-item interactions. The experimental results indicate the effectiveness of CoupledCF model compared to several state-of-the-art neural baselines on five available real-world datasets. In this work, we only use the user information included user demographic information and user reviews and item information contained limited item attributes and item reviews, while the real-life data may obtain all user/item attributes. We are working on finding real-life business data with rich user/item attributes to test the CoupledCF model and exploring other deep architectures for representing hierarchical and heterogeneous user-item coupling relationships.

In this paper, to simply the coupling learning, we define a user-item coupling function as shown in Eq. 12. In future work, first, we will investigate the more effective coupling interaction way to learn user-item coupling relationships and we will introduce social relationships between users as user information, and we will study the better embedding learning method for extracting user and item features. In addition, we are interested in involving multimodal information as user/item information, such as images and videos, include more abundant visual information [47], that can better express user interest and item characteristics.

VI. EXPLANATION

This paper is an extension of the conference version appeared in IJCAI 2018 (Quangui Zhang, Longbing Cao, Chengzhang Zhu, Zhiqiang Li, Jinguang Sun. CoupledCF: Learning Explicit and Implicit User-item Couplings in Recommendation for Deep Collaborative Filtering. IJCAI. 2018: 3662-3668.). This manuscript contains, among others, the following new materials: (1) Add review texts as the side information and use Doc2vec model to learn user and item features from review texts (Section 3.6), which can improve model's recommendation accuracy. (2) Add rating prediction experiments on four available datasets including MovieLens1M, BookCrossing, Amazon Movies and TV (AMT), and Yelp review datasets (Section 3.4, Section 4). (3) Add several comparison methods including attribute-based and review-based which show the effectiveness of our model (Section 4). (4) Add more detail descriptions and analysis in experiment part (Section 4). The detailed descriptions are shown in Table 13.

REFERENCES

[1] L. Cao, "Coupling learning of complex interactions," *J. Inf. Process. Manage.*, vol. 51, no. 2, pp. 167–186, 2015.

[2] L. Cao, "Non-iid recommender systems: A review and framework of recommendation paradigm shifting," *Engineering*, vol. 2, no. 2, pp. 212–224, 2016.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[4] F. Zhao and Y. Guo, "Learning discriminative recommendation systems with side information," in *Proc. IJCAI*, 2017, pp. 3469–3475.

[5] F. Li, G. Xu, and L. Cao, "Coupled matrix factorization within non-IID context," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Springer, 2015, pp. 707–719.

[6] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: Understanding rating dimensions with review text," in *Proc. 7th ACM Conf. Recommender Syst.*, 2013, pp. 165–172.

[7] R. Catherine and W. Cohen, "Transnets: Learning to transform for recommendation," in *Proc. 11th ACM Conf. Recommender Syst.*, 2017, pp. 288–296.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[10] S. Stiebellehner, J. Wang, and S. Yuan, "Learning continuous user representations through hybrid filtering with doc2vec," 2017, *arXiv:1801.00215*. [Online]. Available: https://arxiv.org/abs/1801.00215

[11] D. Agarwal and B.-C. Chen, "fLDA: Matrix factorization through latent Dirichlet allocation," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 91–100.

[12] H. Ma, T. C. Zhou, M. R. Lyu, and I. King, "Improving recommender systems by incorporating social contextual information," *ACM Trans. Inf. Syst.*, vol. 29, no. 2, p. 9, Apr. 2011.

[13] Y. Li, C. Zhai, and Y. Chen, "Exploiting rich user information for one-class collaborative filtering," *Knowl. Inf. Syst.*, vol. 38, no. 2, pp. 277–301, 2014.

[14] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proc. 23rd ACM Int. Conf. Inf. Knowl. Manage.*, 2014, pp. 261–270.

[15] L. Cao, Y. Ou, and P. S. Yu, "Coupled behavior analysis with applications," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1378–1392, Aug. 2012.

[16] T. Li, J. Lu, and L. M. López, "Preface: Intelligent techniques for data science," *Int. J. Intell. Syst.*, to be published.

[17] C. Wang, L. Cao, and C. H. Chi, "Formalization and verification of group behavior interactions," *IEEE Trans. Syst., Man, Cybernetics, Syst.*, vol. 45, no. 8, pp. 1109–1124, Aug. 2015.
[18] Z. Xu, Y. Zhang, and L. Cao, "Social image analysis from a non-IID perspective," *IEEE Trans. Multimedia*, vol. 16, no. 7, pp. 1986–1998, Nov. 2014.
[19] G. Pang, L. Cao, and L. Chen, "Outlier detection in complex categorical data by modelling the feature value couplings," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1–7.
[20] K. Georgiev and P. Nakov, "A non-IID framework for collaborative filtering with restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1148–1156.
[21] S. Jian, L. Hu, L. Cao, and K. Lu, "Metric-based auto-instructor for learning mixed data representation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.
[22] T. D. T. Do and L. Cao, "Coupled Poisson factorization integrated with user/item metadata for modeling popular and sparse ratings in scalable recommendation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.
[23] C. Zhu, L. Cao, Q. Liu, J. Yin, and V. Kumar, "Heterogeneous metric learning of categorical data with hierarchical couplings," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1254–1267, Jul. 2018.
[24] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 425–434.
[25] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 233–240.
[26] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
[27] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, and M. Ispir, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.
[28] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," 2017, *arXiv:1703.04247*. [Online]. Available: https://arxiv.org/abs/1703.04247
[29] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. S. Kankanhalli, "A^3NCF: An adaptive aspect attention model for rating prediction," in *Proc. IJCAI*, 2018, pp. 3748–3754.
[30] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 627–636.
[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[32] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2042–2050.
[33] A. K. Menon, K.-P. Chitrapura, S. Garg, D. Agarwal, and N. Kota, "Response prediction using collaborative filtering with hierarchies and side-information," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 141–149.
[34] S. Jian, G. Pang, L. Cao, K. Lu, and H. Gao, "CURE: Flexible categorical data representation by hierarchical coupling learning," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 853–866, Jun. 2018.
[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: https://arxiv.org/abs/1301.3781
[36] A. T. Wilson and P. A. Chew, "Term weighting schemes for latent Dirichlet allocation," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2010, pp. 465–473.
[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[38] L. Cao, "Non-IIDness learning in behavioral and social data," *Comput. J.*, vol. 57, no. 9, pp. 1358–1370, Sep. 2014.
[39] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1235–1244.
[40] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[42] J. H. Lau and T. Baldwin, "An empirical evaluation of doc2vec with practical insights into document embedding generation," 2016, *arXiv:1607.05368*. [Online]. Available: https://arxiv.org/abs/1607.05368
[43] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*, 2010, pp. 45–50.
[44] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. UAI*, 2009, pp. 452–461.
[45] I. Bayer, X. He, B. Kanagal, and S. Rendle, "A generic coordinate descent framework for learning from implicit feedback," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 1341–1350.
[46] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 426–434.
[47] R. Hong, Y. Yang, M. Wang, and X.-S. Hua, "Learning visual semantic relationships for efficient visual retrieval," *IEEE Trans. Big Data*, vol. 1, no. 4, pp. 152–161, Dec. 2015.

QUANGUI ZHANG received the Ph.D. degree from the Beijing University of Technology. He is currently an Associate Professor with the School of Electronic and Information Engineering, Liaoning Technical University, Huludao, China. His current research interests include deep learning and recommended systems.

LI WANG received the master's degree from the School of Electronic and Information Engineering, Liaoning Technical University, Huludao, China. Her research interest includes recommendation systems.

XIANGFU MENG received the Ph.D. degree from Northeastern University. He is currently a Professor with the School of Electronic and Information Engineering, Liaoning Technical University, Huludao, China. His current research interests include spatial data management, recommendation systems, and Web database query.

KEDA XU received the master's degree from the School of Electronic and Information Engineering, Liaoning Technical University, Huludao, China. His research interest includes recommendation systems.

JIAYAN HU received the master's degree from the School of Electronic and Information Engineering, Liaoning Technical University, Huludao, China. Her research interest includes recommendation systems.