CoupledCF: Learning Explicit and Implicit User-item Couplings in Recommendation for Deep Collaborative Filtering

Quangui Zhang[1], Longbing Cao[2], Chengzhang Zhu[2], Zhiqiang Li[1], Jinguang Sun[1]
[1] Liaoning Technical University
[2] University of Technology, Sydney
zhqgui@126.com, longbing.cao@uts.edu.au, kevin.zhu.china@gmail.com
lizhiq1007@126.com, sunjinguang@lntu.edu.cn

Abstract

Non-IID recommender system discloses the nature of recommendation and has shown its potential in improving recommendation quality and addressing issues such as sparsity and cold start. It leverages existing work that usually treats users/items as independent while ignoring the rich couplings within and between users and items, leading to limited performance improvement. In reality, users/items are related with various couplings existing within and between users and items, which may better explain how and why a user has personalized preference on an item. This work builds on non-IID learning to propose a neural user-item coupling learning for collaborative filtering, called CoupledCF. CoupledCF jointly learns explicit and implicit couplings within/between users and items w.r.t. user/item attributes and deep features for deep CF recommendation. Empirical results on two real-world large datasets show that CoupledCF significantly outperforms two latest neural recommenders: neural matrix factorization and Google's Wide&Deep network.

1 Introduction

Collaborative filtering (CF) is dominant for recommender systems (RS) to predict new user-item interactions (e.g., ratings) by analyzing the relationships between users (or items) in terms of the past user behavior [Koren *et al.*, 2009], such as ratings, viewing, clicking or purchasing behaviors on items. The user-item rating matrix shows the user 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. It would be very difficult to gain reasonable recommendation accuracy when just based on this sparse data. In addition to approaches such as dimensionality reduction, which do not fundamentally change the problem nature, increasing efforts are made on incorporating side information into rating estimation, especially for issues as sparsity and cold start. For this, user and/or item attributes [Shi *et al.*, 2014; Li *et al.*, 2015] are increasingly used to leverage the rating data for better recommendation quality.

Incorporating and learning user/item attributes and contextual information have been a recent focus in recommender systems. For example, Bayesian matrix factorization with side information [Park *et al.*, 2013] handles cold start problems. In [Yao *et al.*, 2015], contextual information is incorporated into CF and Context-aware Personalized Random Walk and Semantic Path-based Random Walk methods are proposed. Li et al. [Li *et al.*, 2014] involved user information such as a user's query history, purchasing and browsing activities. In [Pan and Chen, 2016], a group Bayesian personalized ranking introduces group preference instead of individual preference to relax the individual and independence assumptions. To various extents, the above work aims to involve specific user/item properties and characteristics, such as contexts, user historical activities, demographics and reviews, and item description into recommenders for addressing the shortage of ratings alone and respective challenges such as sparsity and cold start.

However, existing relevant approaches can only lead to limited improvement as they assume users and items are independent and identically distributed (IID). For example, some of the above work simply incorporates user or item properties into a recommendation model but overlooks the various coupling relationships [Cao *et al.*, 2012] within and between users and items and the non-IID nature of recommendation [Cao, 2015; 2016], which may essentially disclose why a user likes (or dislikes) an item [Cao, 2016]. Methods treating user/item information as IID cannot take the full advantage of user/item information to improve the recommendation performance when users/items are actually non-IID. Non-IID recommender systems were introduced to cater for the non-IID nature of users/items in [Cao, 2016], i.e., learning couplings and heterogeneities within/between users and items, and some impressive achievements have been made in creating non-IID RS, e.g., coupled user/item similarity-based matrix factorization [Li *et al.*, 2013; 2015] and in many other learning tasks [Wang *et al.*, 2015a; Chen *et al.*, 2015; Pang *et al.*, 2016; Georgiev and Nakov, 2013; Jian *et al.*, 2018; Do and Cao, 2018; Zhu *et al.*, 2018].

While the above work considers the user-user and/or item-item couplings but does not jointly learns explicit and implicit user-item couplings in deep networks for CF. Modeling explicit and implicit user-item couplings in deep models for CF is actually highly challenging, as this involves high-dimensional and diverse interactions between observable and latent user/item attributes [Cao, 2015]. In addition, deep neural networks such as CNN have demonstrated their power in representing abstract features and made significant achievements in image processing and natural language processing. However, few deep models incorporate explicit attributes and their relations, resulting in gaps in sufficiently representing both implicit and explicit attributes and relations.

In this paper, we propose a coupled CF model, CoupledCF, which jointly learns and integrates both explicit and implicit user-item couplings in terms of both deep features learned by CNN and explicit attributes describing users and items.

- CoupledCF first learns the explicit user-item couplings w.r.t. user attributes and item attributes by a CNN-based user-item coupling learning network, then builds a deep CF (DeepCF) model to learn the implicit user-item couplings, and finally integrates the learned explicit user-item couplings with DeepCF to systematically represent user, item and user-item couplings.
- 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 implicit and explicit user/item features and relations into CF to jointly learn both explicit and implicit user-item couplings towards a comprehensive representation of user-item interactions.

Empirical evaluation of various CoupledCF models: local CoupledCF, global CoupledCF, DeepCF, and their combination CoupledCF are conducted on two real-life large datasets with certain user/item information. The results show all CoupledCF models outperform the baselines; in particular, CoupledCF significantly beats neural MF [He *et al.*, 2017] (by over 40%) and Google's Wide&Deep network (by over 15%) on MovieLens 1M and Tafeng data.

2 Related work

Deep learning is promising for effective and abstract representations in computer vision, natural language processing and speech recognition and deep networks-based representations [Cheng *et al.*, 2016; Guo *et al.*, 2017; He *et al.*, 2017]. Convolutional neural network (CNN) is a widely-used deep learning model in computer vision and natural language processing. CNN shows high potential in effectively representing local and abstract features in image or documents. In recent years, deep learning has been used in recommender systems, such as [Wang *et al.*, 2015b; 2018]. In [Kim *et al.*, 2016], CNN is integrated into probabilistic MF for document context-aware recommendation. In [He *et al.*, 2017], a neural CF combines a shallow MF-based neural network with a multi-layer perceptron, NCF, to learn a user-item interaction function with implicit feedback. In this model, we build and combine a CNN-based user-item coupling learning network with a deep CF model to co-train both explicit and implicit user-item couplings, which cannot be done by existing work.

User/item information has been increasingly involved into CF. In [Cheng *et al.*, 2016], Wide&Deep learning jointly trains wide linear models and deep neural networks. In the wide model, they use cross-product feature transformations to get the benefit of memorization; and in the deep model, they use a three-ReLU-layer neural network for generalization of recommendation. A DeepFM was proposed in [Guo *et al.*, 2017]. Unlike Google's Wide&Deep model, shared raw features are treated as input to DeepFM without feature engineering. In [Zhao *et al.*, 2016], a linear regression model is built on the item information and incorporated into the low-rank CF. In [Park *et al.*, 2013], a Bayesian MF with side information handles the cold start problems. In this paper, we build the embeddings of the user/item information as input into a CNN-based network to learn a user-item coupling representation.

The nature of recommendation is non-IID [Cao, 2016]. Some pioneerning works have been introduced about learning the non-IIDness in recommendation [Cao, 2014], such as in CF [Yu *et al.*, 2013; Li *et al.*, 2013; 2015; **?**],[Georgiev and Nakov, 2013; Wang *et al.*, 2015b], and by statistical learning [Do and Cao, 2018]. This paper builds on the framework of non-IID recommendation, and integrates non-IID explicit coupling learning with CNN to propose a deep CF model.

3 The CoupledCF Model

3.1 Preliminaries

Given a recommendation problem, assume a user set $U = \{u^1, u^2, \ldots, u^m\}$ and an item set $V = \{v^1, v^2, \ldots, v^n\}$ contains m users and n items respectively, and the rating behaviors reflect the interactions between users and items, forming the rating matrix R. Each element y_{ij} in R reflects the interactions between user i and item j, which represents the user's explicit (such as 5-star ratings) or implicit preference on an item. In this work, we focus on user's implicit feedback on items, which can be formulated as a logistic regression problem to predict whether there will be interactions (such as clicking, viewing or purchasing behaviors) between users and items. Besides the above rating matrix, we involve the user and item attributes such as user demographics and item description in the interaction prediction.

Below, we present the CNN-based user-item coupling learning framework in Section 3.2, then introduce the DeepCF model in Section 3.3 and the corresponding CoupledCF model in Section 3.4.

3.2 CNN-based Explicit User-item Coupling Learning

Figure 1 shows the CNN-based user-item coupling learning by involving user/item attributes. To learn the attributes-based user-item couplings, we embed the user and item information as dense vectors $u_c = \{u_{c1}, u_{c2}, \ldots, u_{cm}\}$ and $v_c = \{v_{c1}, v_{c2}, \ldots, v_{cm}\}$ respectively in the same vector space. u_{ci} (v_{ci}) denotes the i^{th} element in vector u_c (v_c). In practice, the user information may be user demographics, user review information, or social relationships between users; The item information may include item descriptions. Accordingly, we take different embedding methods for different types of user/item information. For the embedding vectors u_c and v_c, we construct a function to calculate the coupling relationships, i.e., $f_\Theta(u_c, v_c)$, which measures the interactions between elements of u_c and v_c. Θ denotes the parameters of f, which in practice can be fixed or learned. We then obtain a user-item coupling matrix X_c, in which each element is denoted as $f_{i,j}$, to represent the couplings between the element u_{ci} of vector u_c and element v_{cj} of vector v_c.

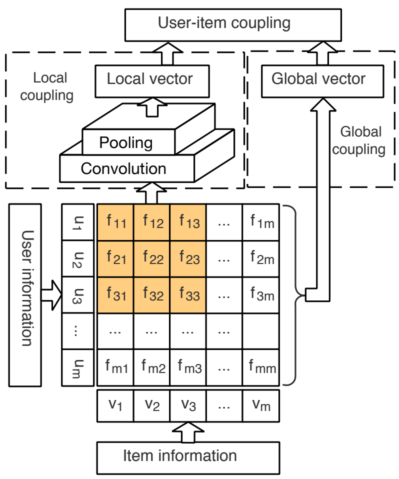


Figure 1: CNN-based learning of local and global explicit user-item couplings by involving user/item attributes.

For the coupling matrix X_c, we utilize CNN to learn the local user-item couplings, forming the local CoupledCF model, as shown in the left box in Figure 1. The local CoupledCF model consists of convolutional and pooling layers. The convolution is performed as:

$$a_c = g(W * X_c + b) \tag{1}$$

$*$ is the convolution operator, W are the filters, b is a bias for W, and g is a non-linear activation function such as ReLU. On top of the convolutional layer, the pooling layers reduce the representation size of the outputs of the convolution to speed up the computation and refine the features from convolution for better discrimination/robustness. The pooling is performed as:

$$a_p = pooling(a_c) \tag{2}$$

A_p is the pooling function such as max pooling or average pooling. The CNN represents the attributes-based local user-item couplings w.r.t. a local convolution filter to capture the local user-item interactions, represented as a local vector. To learn the global user-item couplings, we flatten the coupling matrix X_c as a vector and then concatenate it with the local user-item coupling representation to obtain the global user-item coupling representation, denoted as a global vector e_c.

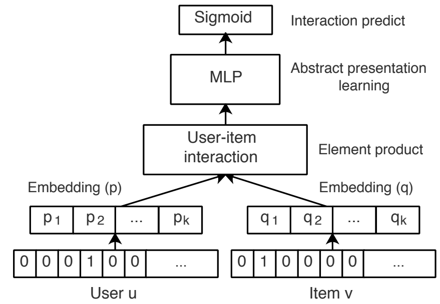


Figure 2: DeepCF: learning implicit user-item couplings.

3.3 DeepCF for Implicit User-item Coupling Learning

We construct a deep CF model, DeepCF, to predict the probability of a user-item interaction, shown in Figure 2. This model captures the implicit user-item couplings w.r.t. latent user and item factors inferred from the rating behaviors [Koren *et al.*, 2009], explaining the rating formation. The latent item factors may explain the explicit characteristics such as a movie's genre and/or the hidden features of items. The user factors measure how much a user likes a movie in terms of the corresponding latent factors. With similar purpose as in NCF [He *et al.*, 2017], DeepCF concatenates the user and item latent factors into a multi-layer fully-connected neural network to learn the implicit user-item interactions. For this, DeepCF represents the user identities and item identities as one-hot vectors u and v respectively. Inspired by the Skip-Gram model [Mikolov *et al.*, 2013], a fully-connected layer is used as the embedding layer to learn the lower-dimension dense vectors, denoted as p and q for u and v respectively. The embedding process is formulated as:

$$\begin{aligned} p &= W_u^T u \\ q &= W_v^T v \end{aligned} \tag{3}$$

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

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

$$r = p \otimes q = (p_1 q_1, p_2 q_2, \ldots, p_k q_k) \tag{4}$$

Unlike Matrix Factorization models that model user-item ratings by inner-products, we feed vector r into a multi-layer fully-connected neural network to deeply learn the high-level abstract user-item interactions. After training DeepCF, 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's 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 v measures the extent of interest the user has in the corresponding factors of the item. Accordingly, the output vector

r 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 users and items. We formulate it as

$$\begin{aligned} a_1 &= ReLU(W_1^T r + b_1) \\ a_2 &= ReLU(W_2^T a_1 + b_2) \\ &\cdots \\ a_L &= ReLU(W_L^T a_{L-1} + b_L) \end{aligned} \tag{5}$$

$W_1, W_2, \ldots, W_L$ and $b_1, b_2, \ldots, b_L$ denote the weight matrices and biases of each layer, and $a_1, a_2, \ldots, a_L$ denote the output of each layer activated by the ReLU function.

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:

$$P_\Theta(y = 1|u, v) \tag{6}$$

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

$$\hat{y} = Sigmoid(w_o^T a_L + b_o) \tag{7}$$

As CF models do not involve negative examples, we sample some negative examples by a negative sampling strategy to make DeepCF discriminative. In our experiments, we sample negative examples from the unobserved interactions in the matrix R by a uniform negative sampling strategy as in [He *et al.*, 2017].

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 $i \in \{1, 2, 3, \ldots, |D|\}$) the loss is:

$$\mathcal{L}(\hat{y}^{(i)}, y^{(i)}) = -y^{(i)} \log(\hat{y}^{(i)}) - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \tag{8}$$

We then learn the network parameters Θ per the following cost function:

$$J = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) \tag{9}$$

3.4 CoupledCF: Integrating Explicit and Implicit User-item Coupling Learning

Here we integrate the CNN-based local/global explicit user-item coupling learning with the implicit user-item coupling learner DeepCF to construct a comprehensive coupled CF model: CoupledCF.

The left network in Figure 3 implements the CNN-based user-item coupling learning, where user demographic information and item attributes are used as the user and item information. For user information, a dense embedding vector is learned for each categorical feature. We concatenate all the user embeddings with the numerical features which are normalized. The item information is processed in the same way as for the user information. Accordingly, two vectors u_c

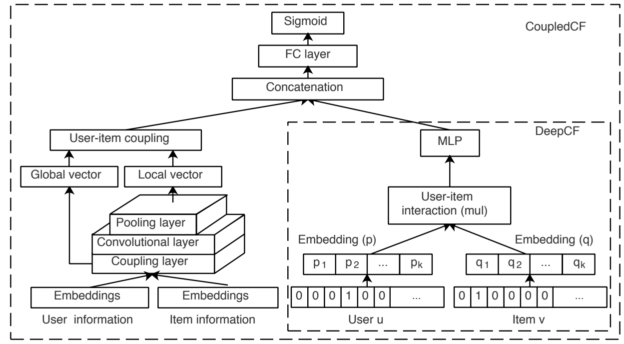


Figure 3: CoupledCF: Jointly learning explicit and implicit user-item couplings, the left box for CNN-based explicit user-item coupling learning, and the right for implicit user-item coupling learning, which are fused to learn the overall user-item couplings.

and v_c are generated and fed into the coupling layer. In our experiments, to simplify the learning, we define a user-item coupling calculation function below:

$$f_{ij} = u_{ci} \times v_{cj} \tag{10}$$

Accordingly, the user-item matrix X_c can be viewed as the cross-product of u_c and v_c. We execute two processes on the user-item 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 forms the CoupledCF model. The overall output of CoupledCF is activated by a Sigmoid function and squashed to the range [0,1].

Hence, the training dataset of CoupledCF can be represented as $D = \{< (u^{(i)}, v^{(i)}, u_c^{(i)}, v_c^{(i)}), y^i >\}$ $i \in \{1, 2, \ldots, |D|\}$, and the objective of CoupledCF is implemented by predicting the probability P_Θ:

$$P_\Theta(y = 1|u, v, u_c, v_c) \tag{11}$$

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

$$\hat{y} = Sigmoid(ReLU(W_o^T r + b_o) + b) \tag{12}$$

The cost function in Eqn. (9) is used to train CoupledCF.

4 Experiments and Evaluation

Similar to most CF methods [He *et al.*, 2017; Rendle *et al.*, 2009], we test our methods against the baselines for recommendations items to users who may likely be interested in based on implicit feedback reflecting user-item interactions.

4.1 Experimental Settings

Datasets. As few large-scale datasets are available with consistent ratings and user/item information, MovieLens 1M

[Harper and Konstan, 2016] and Tafeng dataset are used in our experiments. MovieLens1M consists of 1M transactions from 6,040 users and 3,952 items, Tafeng contains 817,741 transactions with 32,266 users and 23,812 items. Both datasets have user demographics (Gender, Age, Occupation and Zip code in MovieLens 1M and Customer ID, Age, and Region in Tafeng) and some item attributes (Genres in MovieLens 1M and Original ID, Sub class, Amount, Asset and Price in Tafeng). To create implicit feedbacks in these datasets, similar to the above methods, we binarize the ratings in two datasets 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 attributed datasets to the implicit version, we uniform-randomly sample 4 negative instances for each positive instance.

Baseline Methods. Our CoupledCF is customized to four versions below to learn various types of user-item couplings:

- DeepCF: only learns the user-item interactions based on latent user/item factors (Figure 2);
- lCoupledCF: the CoupledCF integrating the the CNN-based local user-item coupling learning and DeepCF (the model with the local vector in Figure 1);
- gCoupledCF: the CoupledCF including the global coupling learning component without the CNN component (the model with the global vector in Figure 1);
- CoupledCF: the CoupledCF consisting of the local vector, global vector and DeepCF.

The following relevant and representative state-of-the-art methods are used as the baselines to evaluate our methods.

- NeuMF [He *et al.*, 2017]: a CF method with implicit feedback embedded into a neural network. It verifies the contributions made by learning user/item information and how well our method with user/item information performs compared to the basic neural CF without side information.
- Wide&Deep [Cheng *et al.*, 2016]: a benchmark Google's wide&deep neural network combining memorization and generalization for recommendation, which involves feature engineering (such as cross-product features) of the input to the wide network. To be fair, we transform the relevant user/item attributes to 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 attributes to this Wide&Deep model that uses refined cross-product features.

Modeling Settings. The CoupledCF model is implemented in Python based on the Keras framework. All the baseline methods are implemented per their Github settings. All the experiments are performed in a 3.4GHz Titan Cluster with 96GB memory. The hyper-parameters of CoupledCF mainly include:

- The dimensionality of the embedding layers of the DeepCF model: After testing a various number of embedding layers, we empirically choose the best number 32 corresponding to Tafeng for both datasets.
- The number of the embedding layers of the CNN-based user-item coupling learning network: We evaluated it w.r.t. $\{8, 16, 32, 64, 128, 256\}$, and got the best performance on the model with 8 embedding layers for both datasets.
- We construct two CNN layers, set the filter as (8,8) and the channel as 8 for both datasets.
- The number of the fully-connected layers before the final output layer: We evaluated it w.r.t. $\{8, 16, 32, 64, 128, 256\}$, and got the best performance on the model with 64 fully-connected layers for the two datasets.
- The learning rate: We set it as 0.001 for MovieLens1M and 0.005 for Tafeng.

We use Adam as the optimizer for our model. For parameter initialization, we initialize the embedding matrix with a random normal distribution (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.

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 [Bayer *et al.*, 2017]. Similar to [Koren, 2008], 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) and Normalized Discounted Cumulative Gain (NDCG) [He *et al.*, 2017] as evaluation metrics.

- HR@K: a recall-based measure, i.e., $HR@K = \frac{\#hits@K}{|GT|}$, to indicate whether the test item is in the top-K recommended item list.
- NDCG: is a ranking-based measure, $NDCG@K = Z_k \sum_{i=1}^{K} \frac{2^{rel_i}-1}{\log_2(i+1)}$, which assigns higher scores to hits with top ranks.

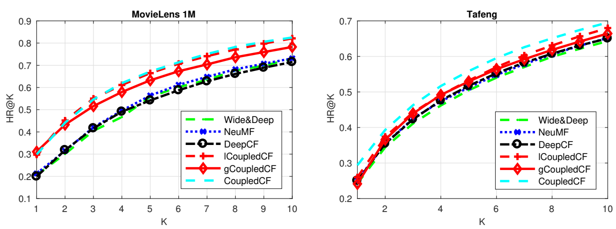
where GT denotes the test list set, rel_i is the graded relevance value of the item at position i and Z_K is the normalization. In our experiments, we set $rel_i \in \{0, 1\}$, which depends on whether i is in the test dataset.

4.2 Results and Analysis

Top-K item recommendation results. The performance of the CoupledCF models against the baselines is reported in terms of HR@10 and NDCG@10 for top-10 items in Table 1. We also test the top-K (K=1 to 10) item recommendations in Figure 4 and Figure 5. CoupledCF is compared with local CoupledCF (lCoupledCF for short in Table 1), global CoupledCF (gCoupledCF), and DeepCF as well as all the baselines. The results show that CoupledCF significantly improves recommendation performance, e.g., up to 40.25% improvement over NeuMF (HR@2) and up to 15.27% improvement over Wide&Deep (NDCG@1), and averaged 22.84% improvement over NeuMF and averaged 11.78% over Wide&Deep.

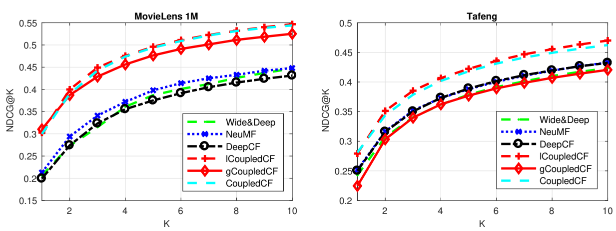
Table 1: HR@10 and NDCG@10 for Top-10 item recommendation

	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	0.4233
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



(a) HR@K on MovieLens (b) HR@K on Tafeng

Figure 4: HR@K results of Top-K item recommendation.



(a) NDCG@K on MovieLens (b) NDCG@K on Tafeng

Figure 5: NDCG@K results of Top-K item recommendation.

Comparison with baselines. First, compared to neural MF model NeuMF, as shown in Table 1, CoupledCF beats NeuMF by 12.89% on MovieLens1M and 6.66% on Tafeng w.r.t. on HR@10; and 21.70% MovieLens1M and 6.79% on Tafeng w.r.t. NDCG@10. It shows our model involving user/item information significantly outperforms the neural MF without side information.

Second, compared to the Google benchmark Wide&Deep model, CoupledCF beats it by 9.52% MovieLens 1M and 5.21% on Tafeng w.r.t. on HR@10; and 9.60% MovieLens1M and 3.89% on Tafeng w.r.t. NDCG@10. This shows the way of CoupledCF learning and integrating explicit/implicit user-item interactions outperforms the feature engineering-based Wide&Deep's.

Testing the CoupledCF effectiveness. We further evaluate the working mechanism of CoupledCF in terms of different components embedded in the model. As introduced earlier, CoupledCF is customized to four versions: DeepCF, local CoupledCF, global CoupledCF, and the comprehensive CoupledCF. As shown in Table 1 and Figures 4 and 5, CoupledCF generally outperforms other versions for top-K item recommendations on both datasets. By comparison, local CoupledCF beats DeepCF and global CoupledCF in all cases. For example, for top-10 item recommendations, local CoupledCF beats DeepCF by 14.90% improvement and beats global CoupledCF by 4.93% w.r.t. HR@10 on Movielens1M, showing local CoupledCF captures CNN-based explicit user-item coupling learning outperforms the implicit user-item coupling neural network.

As shown in Figures 4 and 5, the above observations remain for the top-K item recommendation where K ranges from 1 to 10. The results show 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 user-item coupling and implicit user-item couplings generally gains the best performance.

5 Conclusions

The nature of recommendation is non-IID. This work learns explicit and latent user-item interactions for recommendation: user/item attributes-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 show significant improvement over state-of-the-art baselines on large datasets. In this work, we only use the user demographic information and limited item attributes, 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.

References

[Bayer *et al.*, 2017] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. A generic coordinate descent framework for learning from implicit feedback. In *WWW*, pages 1341–1350, 2017.

[Cao *et al.*, 2012] Longbing Cao, Yuming Ou, and Philip S Yu. Coupled behavior analysis with applications. *IEEE Trans. on Knowledge and Data Engineering*, 24(8):1378–1392, 2012.

[Cao, 2014] Longbing Cao. Non-iidness learning in behavioral and social data. *The Computer Journal*, 57(9):1358–1370, 2014.

[Cao, 2015] Longbing Cao. Coupling learning of complex interactions. *J. Information Processing and Management*, 51(2):167–186, 2015.

[Cao, 2016] Longbing Cao. Non-iid recommender systems: A review and framework of recommendation paradigm shifting. *Engineering*, 2(2):212–224, 2016.

[Chen *et al.*, 2015] Qianqian Chen, Liang Hu, Jia Xu, Wei Liu, and Longbing Cao. Document similarity analysis via involving both explicit and implicit semantic couplings. In *DSAA*, pages 1–10, 2015.

[Cheng *et al.*, 2016] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 7–10, 2016.

[Do and Cao, 2018] Trong Dinh Thac Do and Longbing Cao. Coupled poisson factorization integrated with user/item metadata for modeling popular and sparse ratings in scalable recommendation. In *AAAI2018*, pages 1–7, 2018.

[Georgiev and Nakov, 2013] Kostadin Georgiev and Preslav Nakov. A non-iid framework for collaborative filtering with restricted boltzmann machines. In *ICML*, pages 1148–1156, 2013.

[Guo *et al.*, 2017] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[Harper and Konstan, 2016] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19, 2016.

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

[Jian *et al.*, 2018] Songlei Jian, Liang Hu, Longbing Cao, and Kai Lu. Metric-based auto-instructor for learning mixed data representation. In *AAAI'2018*, pages 1–7, 2018.

[Kim *et al.*, 2016] Donghyun Kim, Chanyoung Park, Jinoh Oh, Sungyoung Lee, and Hwanjo Yu. Convolutional matrix factorization for document context-aware recommendation. In *RecSys*, pages 233–240, 2016.

[Koren *et al.*, 2009] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *KDD*, pages 426–434, 2008.

[Li *et al.*, 2013] Fangfang Li, Guandong Xu, Longbing Cao, Xiaozhong Fan, and Zhendong Niu. Cgmf: Coupled group-based matrix factorization for recommender system. *WISE (1)*, 8180:189–198, 2013.

[Li *et al.*, 2014] Yanen Li, ChengXiang Zhai, and Ye Chen. Exploiting rich user information for one-class collaborative filtering. *Knowledge and information systems*, 38(2):277–301, 2014.

[Li *et al.*, 2015] Fangfang Li, Guandong Xu, and Longbing Cao. Coupled matrix factorization within non-iid context. In *PAKDD (2)*, pages 707–719, 2015.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Pan and Chen, 2016] Weike Pan and Li Chen. Group bayesian personalized ranking with rich interactions for one-class collaborative filtering. *Neurocomputing*, 207:501–510, 2016.

[Pang *et al.*, 2016] Guansong Pang, Longbing Cao, and Ling Chen. Outlier detection in complex categorical data by modelling the feature value couplings. In *IJCAI2016*, pages 1–7, 2016.

[Park *et al.*, 2013] Sunho Park, Yong-Deok Kim, and Seungjin Choi. Hierarchical bayesian matrix factorization with side information. In *IJCAI*, pages 1593–1599, 2013.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[Shi *et al.*, 2014] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR)*, 47(1):3, 2014.

[Wang *et al.*, 2015a] Can Wang, Longbing Cao, and Chi-Hung Chi. Formalization and verification of group behavior interactions. *IEEE Trans. Systems, Man, and Cybernetics: Systems*, 45(8):1109–1124, 2015.

[Wang *et al.*, 2015b] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *KDD*, pages 1235–1244, 2015.

[Wang *et al.*, 2018] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. Attention-based transactional context embedding for next-item recommendation. In *AAAI'2018*, pages 1–7, 2018.

[Yao *et al.*, 2015] Weilong Yao, Jing He, Guangyan Huang, Jie Cao, and Yanchun Zhang. A graph-based model for context-aware recommendation using implicit feedback data. *World wide web*, 18(5):1351–1371, 2015.

[Yu *et al.*, 2013] Yonghong Yu, Can Wang, Yang Gao, Longbing Cao, and Xixi Chen. A coupled clustering approach for items recommendation. In *PAKDD*, pages 365–376, 2013.

[Zhao *et al.*, 2016] Feipeng Zhao, Min Xiao, and Yuhong Guo. Predictive collaborative filtering with side information. In *IJCAI*, pages 2385–2391, 2016.

[Zhu *et al.*, 2018] Chengzhang Zhu, Longbing Cao, Qiang Liu, Jianpin Yin, and Vipin Kumar. Heterogeneous metric learning of categorical data with hierarchical couplings. 2018.