Review-Based Cross-Domain Collaborative Filtering: A Neural Framework

Thanh-Nam Doan, Shaghayegh Sahebi
University at Albany - SUNY
{tdoan,ssahebi}@albany.edu

ABSTRACT

Cross-domain collaborative filtering recommenders exploit data from other domains (e.g., movie ratings) to predict users' interests in a different target domain (e.g., suggest music). Most current cross-domain recommenders focus on modeling user ratings but pay limited attention to user reviews. Additionally, due to the complexity of these recommender systems, they cannot provide any information to users to support user decisions. To address these challenges, we propose Deep Hybrid Cross Domain *(DHCD)* model, a cross-domain neural framework, that can simultaneously predict user ratings, and provide useful information to strengthen the suggestions and support user decision across multiple domains. Specifically, DHCD enhances the predicted ratings by jointly modeling two crucial facets of users' product assessment: ratings and reviews. To support decisions, it models and provides natural review-like sentences across domains according to user interests and item features. This model is robust in integrating user rating and review information from more than two domains. Our extensive experiments show that *DHCD* can significantly outperform advanced baselines in rating predictions and review generation tasks. For rating prediction tasks, it outperforms cross-domain and single-domain collaborative filtering as well as hybrid recommender systems. Furthermore, our review generation experiments suggest an improved perplexity score and transfer of review information in *DHCD*.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Cross-domain Collaborative filtering, neural network, hybrid collaborative filtering

1 INTRODUCTION

Nowadays, users are overwhelmed by the number of choices online. Recommender systems are increasingly used as an essential tool, to alleviate this problem. Despite improvements in recommender systems, many of them still suffer from problems, including cold-start [21] and difficulty in explaining their suggestions [26]. Moreover, collaborative filtering recommenders [11] cannot use obvious feature-based relations between users and items. Content-based approaches cannot capture deeper social or semantic similarities between users and items, nor they can suggest novel items (outside the scope of user profile features) to users [17].

Two major approaches to address some of these problems are hybrid [18] and cross-domain [4] recommender systems. Hybrid recommender systems merge content-based and collaborative filtering approaches to provide higher-quality recommendations. Some hybrid recommender systems jointly model user ratings and reviews to introduce a more sophisticated view to user interests and item features, that leads to improved recommendation results [18].

The idea behind cross-domain recommendation systems is to share useful information across two or more domains to improve recommendation results [4]. They work by transferring information from one or more *source* or *auxiliary* domains to suggest useful items in a *target* domain. Especially, when user history in the target domain (e.g., books) does not provide enough information about user interests, user preferences in another source domain (e.g., movies) can provide useful insights that can lead to more accurate or novel recommendations[1]. In addition to improving recommendation results, cross-domain recommender algorithms provide a solution to problems, such as cold-start or user profiling, in single-domain recommenders.

Both hybrid and cross-domain recommender systems have shown to be successful in the current literature. However, a combination of two has been rarely studied. Additionally, the problem of providing more information to users to support their decisions in cross-domain recommender systems, has not been studied. Most of the current research in cross-domain recommenders focus on collaborative filtering cross-domain approaches [19]. These approaches incorporate users' explicit (e.g., rating) or implicit (e.g., purchase) feedback in the auxiliary domain to recommend items in the target domain. Many of these algorithms jointly model multiple domains by sharing common user's latent representations across them. Collaborative filtering cross-domain recommenders, similar to their single-domain counterparts, suffer from ignoring content information. Having advanced models, which are built on users' rating or binary feedback, complicates the reasoning of why a specific user may be interested in an item. Moreover, these recommender algorithms lose the explicit user-item similarities by ignoring an important source of information: user reviews.

To further enhance the performance and transparency of cross-domain recommendation systems, we propose to combine hybrid and cross-domain approaches together. With this fusion, we can benefit from the strength of both hybrid and cross-domain recommender systems: cross-domain modeling will enhance user latent features by providing extra information from other domains (especially in sparser ones), reviews will bring another dimension for enriching user and item latent features and offer insights to increase the recommendation transparency. Therefore, merging

ComplexRec 2019, 20 September 2019, Copenhagen, Denmark

[1]While other definitions of domain exist in the literature, e.g., time-based domains, in this paper, we focus on item domains (e.g., item type or category).

the two will enrich content features by using review information across domains as well as enhance prediction performance.

Accordingly, we propose Deep Hybrid Cross Domain (*DHCD*) recommender, which models various types of user feedback (both ratings and reviews) across multiple domains under neural network framework. We use neural network as a natural choice to model reviews due to its success in natural language processing and generating natural language sentences [5, 26]. In addition to using reviews for producing better-quality suggestions, DHCD can generate natural and useful reviews to support user decisions for suggested cross-domain items. By generating a review that is based on the specific user's interests across domains and other reviews, we can help clarify why a specific item is recommended to user. Our model shares information across domains in two levels by sharing users' latent representations, and cascading it into reviews' latent representations. It can capture non-linear user-item relationships by having a neural network framework [5]. Our results and findings of this research are summarized as follows:

- We propose a neural network framework named Deep Hybrid Cross Domain (DHCD) model which unifies ratings and reviews of users and items across multiple domains.
- To the best of our knowledge, DHCD is the first framework which is able to automatically generate cross-domain reviews that in turn can provide decision support for cross domain recommendations.
- We design and implement multiple experiments to evaluate DHCD's performance in three real-world datasets. Our evaluation is performed via two main tasks: rating prediction and review generation tasks, to answer four research questions.

2 RELATED WORKS

Here, we briefly review the literature on cross-domain recommendations and neural network-based collaborating filtering.

Cross-Domain Recommendation focuses on learning user preferences from data across multiple domains [4]. There are two focuses on cross domain recommendation: collaborative filtering [3] and content-based methods [20]. In this work, we focus on collaborative filtering cross domain recommendations. Similar to single-domain collaborative filtering, research work on cross domain recommendation usually use matrix factorization. For example, Pan et al. [16] propose a cross domain recommendation system based on matrix factorization by using a coordinate system transfer method. Elkahky et al. [3] use deep learning framework to improve the performance of cross domain recommendation and also provide a scalable method to handle large datasets. However, not considering the reviews of items is the main limitation of these methods.

Xin et al. [24] proposed the first review-based cross-domain recommender model. They proposed a graphical model to capture the user ratings and item reviews across domains but reviews are not used to model user latent features. Later, Song et al. proposed a joint tensor factorization model to capture both user reviews and implicit feedback on items to provide cross-domain recommendations [22]. However, it does not capture non-linearities across domains, nor it models reviews as natural sequences of words. None of the above works generate reviews.

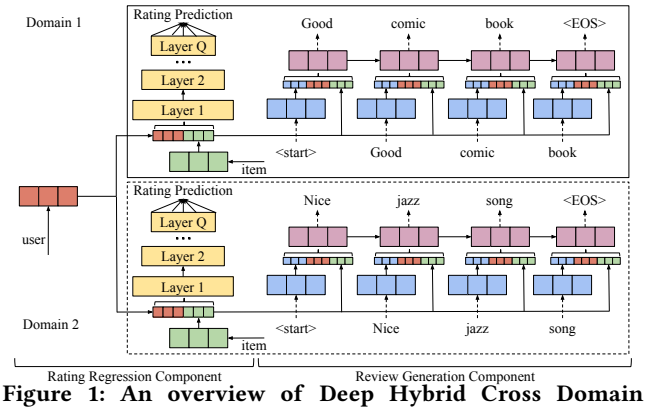

Figure 1: An overview of Deep Hybrid Cross Domain (DHCD) recommendation system.

Neural Frameworks for Collaborative Filtering. Due to its ability to approximate non-linear relation of users and items, neural network is rapidly growing in recommendation systems [25].

He et al. [7] propose a fusion model that combines matrix factorization and multi-layer perceptron. Despite the efficiency of their proposed model, it does not consider reviews and is not extended to cross-domain recommendation system. Collaborative Deep Learning (CDL) [23] overcome the sparsity of ratings by using auxiliary information such as reviews. Using reviews as a set of words, their model outperforms baselines but not considering the sequential nature of words in reviews is a limitation.

Review Generation. Ni et al. [14] presented one of the first works that focuses on generating reviews along with preference prediction. Ni and McAuley [15] propose a neural network based upon attention model to assist users writing reviews of items. However, these works and others [1, 8] do not model the preference between users and items, nor they are extendable to cross-domain recommenders.

3 PROPOSED FRAMEWORK

In this section, we describe the architecture of **D**eep **H**ybrid **C**ross **D**omain (DHCD) recommendation system in detail.

3.1 Architecture

DHCD predicts user ratings on items and generates user reviews on them using two main components: the *rating regression component* and the *review generation component*. In the rating regression component (RRC), user ratings on items of each domain are modeled as a function of user and item latent representations. For each user, this component learns a shared latent representation across all domains. Moreover, the shared representations of users has a role as a gate to transfer information across domains. The shared user latent representations in combination with domain-specific latent item representations predict user ratings on items. The review generation component (RGC) generates user reviews on items according to user, item, and word latent representations. In this component, the user and item representations from rating regression component work as a guide to learn review word embeddings per user-item review. This guidance helps sharing word embedding information across domains. Figure 1 illustrates the architecture of our model. In the following, we present our model in more details.

Notations We model the system to include a set of users $\mathcal{U}$, and a set of item domains $\mathcal{D}$. Each of these domains include a set of items $\mathcal{I}^d$, $d \in \mathcal{D}$. For a user $u \in \mathcal{U}$ and each item $i \in \mathcal{I}^d$, the training data may include user's rating on that item (r^d_{ui}) and user's review on that item (s^d_{ui}). Accordingly, we model training data in domain d as a set of tuples $\mathcal{T}^d = \{(u, i, s, r) | u \in \mathcal{U}, i \in \mathcal{I}^d, s \in \mathcal{S}^d, r \in \mathcal{R}^d\}$. Given training data in all domains $\mathcal{T}$, our goal is to simultaneously estimate user u's missing rating on item i in domain d ($\hat{r}^d_{u,i}$) and generate user u's missing textual review on that item ($\hat{s}^d_{u,i}$).

3.2 Rating Regression Component (RRC)

The main purpose of this component is to form a structure to infer user and item representations using observed user feedback on items across all domains. To do this, we model each user u's interests as latent factors v_u and item i's representation (in domain d) as latent factors v^d_i. Then, user u's predicted rating on item i, $\hat{r}^d_{ui}$, is calculated as a function $g_r(\cdot)$ of v_u and v^d_i. Formally, we have:

$$\hat{r}^d_{ui} = g_r(v_u, v^d_i) \tag{1}$$

In many single-domain factorization-based recommender systems, g_r is modeled as the vector dot product of these latent factors plus some bias b [11]. Namely, $\hat{r}_{ui} = v^T_u v^d_i + b$. This specification has some limitations that makes it inappropriate for our cross-domain problem. First, the simple factorization formulation is not fit for a cross-domain problem, as it does not transfer information across domains. Also, the predicted ratings in this model are assumed to be a linear combination of user and item latent factors. However, recent work suggests that using a non-linear model can enhance the representation ability of user and items, and lead to more accurate results [5, 12]. More specifically, in cross-domain recommenders, Xin et al. have shown that user ratings across different domains can have non-linear relationships with each other [24]. Finally, the above formulation requires a shared latent space between users and items. This assumption can restrict the expressiveness capacity of the model since it (i) limits the user and item latent vectors to have equal sizes, and (ii) assumes the k^{th} element of user latent vector must only interact with the corresponding element of item latent factor. For further information, He et al. [7] provide an example which illustrates these above restrictions.

To tackle the non-linearity problem, we model g_r using deep neural networks. Neural networks have been successfully used in collaborative filtering problems [2, 7, 25] and can inherently model non-linear relationships [9]. To have a cross-domain solution, we extend the collaborative neural network to include multiple item domains. As shown in left side of Figure 1, for each domain d, we construct a multi-layer perceptron network (H^d) with Q layers. To share and transfer information across different domains, we model the user latent factors v_u to be shared across all domains. Additionally, to avoid having a shared latent space between users and items, we use concatenation instead of dot product in g_r. Consequently, the input x^d_{ui} to our multi-layer perceptron's first layer is a concatenation of embedding latent vector v_u of user u and embedding latent vector v^d_i of item $i \in \mathcal{I}_d$. Formally, $x^d_{ui} = [v_u; v^d_i]$.

Layer H^d maps this input x^d_{ui} to rating $\hat{r}^d_{ui}$. We denote the q-th hidden layer of H^d as h^d_q, which includes a non-linear function projecting from the output of h^d_{q-1}; i.e. $h^d_q = \text{ReLU}(W^d_q h^{q-1}_d + b^d_q)$ where W^d_q and b^d_q are parameters of H^d's q^{th} layer for domain d and $\text{ReLU}(x) = max(0, x)$. For the first layer, $h^d_0 = x^d_{ui}$ is the input. We ensure the full connectivity between each two adjacent hidden layers h^d_q and h^d_{q-1}. We use regression to map the output vector $\hat{y}^d_Q$ of final layer to the prediction value $\hat{r}^d_{ui}$ i.e. $\hat{r}^d_{ui} = w^d_y \hat{y}^d_Q + b^d_y$ where $\hat{r}^d_{ui}$ is the predicted rating value of user u and item i in domain d. $w^d_y \in \mathbb{R}^r$ and $b^d_y \in \mathbb{R}$ are regression parameters.

To learn the parameters of RRC, we optimize the following regression loss function:

$$\mathcal{L}^r = \sum_{d \in \mathcal{D}} \sum_{u \in \mathcal{U}, i \in \mathcal{I}^d} (r^d_{ui} - \hat{r}^d_{ui})^2 \tag{2}$$

where r^d_{ui} is the observed rating of user u and i in domain d.

3.3 Review Generation Component (RGC)

This component is to model and generate reviews for user-item pairs in cross-domain setting. Here, we model user, item, and review word latent factors to generate natural language sentences.

Recently, recurrent neural networks with components such as long-short term memory (LSTM) and gated recurrent units (GRU) have showed high performance in natural language processing-related tasks such as image captioning, Q&A system [5]. Inspired by their success, we adapt LSTM as a component for our review generation process.

As shown in Figure 1, for each domain d, we construct a separated LSTM model $\bar{H}^d$, that can connect to the rating regression component. Assume s^d_{ui}, user u's review on item i in domain d, as a sequence of words t_j where $j \leq J_{ui}$ (J_{ui} is the number of words in this review). Given a text sequence $t_1, t_2, ..., t_{J_{ui}}$, the LSTM network will update its hidden state parameters ($\bar{h}^d_j$), in step j, according to t_j and previous step's hidden state ($\bar{h}^d_{j-1}$). Subsequently, the network will predict t_{j+1}, step $(j+1)$'s word, using all of its previous words ($t_{<j+1}$). The output layer is connected to a softmax layer. The general idea of review modeling is expressed by $p(t_j | t_{<j}, \Phi^d) = \delta(\bar{h}^d_j)$ where Φ^d represents neural network's hyperparameters for domain d, and $\delta(\cdot)$ is the softmax function. Each hidden state $\bar{h}^d_j$ is modeled as a function of word t_j and previous state $\bar{h}^d_{j-1}$

The above "vanilla" LSTM can only model sentences from a corpus and is unable to embed user and item latent features. For reviews to represent user tastes, we have to make sure to include user and item features. To enhance modeling power, we first apply word2vec [13] to the corpus of review texts to learn the embedding vector of each word. Then, for each word in the review, we concatenate this embedding vector with user and item latent vectors to create a latent vector for word t_j i.e. $[\text{word2vec}(t_j); v_u; v^d_i]$. There are three main advantages for this representation mechanism: (i) concatenation with user and item latent vectors ensures that the user and item information will not vanish over steps. Consequently, it can enhance the sequence generation; (ii) since user latent vectors are shared across domains concatenation with user latent vectors work as a mean to transfer information across domains for reviews; and (iii) word2vec is able to learn some hidden word characteristics,

within corpus, which cannot be inferred from one-hot encoding vector or tf-idf [5].

To learn the parameters of LSTM network, we optimize the Negative Log-likelihood of review data:

$$\mathcal{L}^s = -\sum_{d\in\mathcal{D}}\sum_{u\in\mathcal{U}, i\in\mathcal{I}^d}\sum_{j=1}^{J_{ui}}\log p(t_j|t_{<j}, \Phi^d) \tag{3}$$

3.4 Joint Model Learning

To simultaneously model ratings and reviews across domains, we jointly learn the unified loss function:

$$\mathcal{L} = \lambda_r\mathcal{L}^r + \lambda_s\mathcal{L}^s + \lambda(\|V_u\|_2^2 + \|V_i\|_2^2 + \|\Phi\|_2^2) \tag{4}$$

where hyperparameters λ_r and λ_s control the trade off between rating regression and review generation tasks. λ is the regularization term for avoiding overfitting. V_u and V_i are matrices that stack all latent factors of users/items in all domains. Φ represents all parameters of DHCD. The above loss function is efficient to be optimized in end-to-end manner using back-propagation [5].

4 EXPERIMENTS

In this section, we evaluate our proposed model against several baselines to demonstrate the robustness of DHCD.

4.1 Datasets

We consider three category combinations of Amazon datasets [6]: Book and Digital Music; Book and Office Products; Digital Music and Office Products. For each cross domain dataset, we select users who make purchase and write reviews on both categories. In each review, we filter out words whose frequency is less than 50. Table 1 describes some statistic of the datasets.

Training/Test Data: For each dataset and each user, we chronologically split their first 80% of ratings and reviews as training and the remaining 20% as testing data.

Dataset	#users	#items	#ratings	Avg. Review Len.
Book + Digital Music	3,054	100,055	186,160	163.87
Book + Office Products	4,187	172,484	413,463	146.65
Digital Music + Office Products	186	2,375	4,739	143.53

Table 1: Dataset Statistic

4.2 Baselines and Parameter Settings

For *rating prediction* experiment, we compare our proposed DHCD model with these below baselines:

- Matrix factorization (MF) [11]: It uses user and item ratings as an input. The predicted value is a linear combination of interaction between user and item latent features as well as the user/item/global bias.
- Neural Collaborative Filtering (NCF) [7]: With ratings as its input, this single-domain model combines neural network and matrix factorization to capture the non-linear interaction between users' and items' latent factors.
- Collaborative Deep Learning (CDL) [23]: Using ratings and bag of words of reviews, this single-domain model fuses neural network and topic modeling.
- Collaborative Filtering with Generative Concatenative Networks (CF-GCN) [14]: This single-domain hybrid model unifies both ratings and reviews under a neural framework.
- Cross-domain neural network (CDN) [3]: This model utilizes neural networks for cross domain recommendation system. However, it does not consider reviews along with ratings of users and items.

We design the experiments in two settings: regular single-domain, and cross-domain. In the regular single-domain setting, we model user feedback of one domain to recommend items in the same domain. In the cross-domain setting, although the baseline may have been designed for one domain, we use user feedback on both domains to recommend useful items. For the single domain models, we add a prefix "cd" to their name to indicate when they are provided data from both domains. Specifically, both domain datasets are unified and used for training cdMF and cdNCF.

For *rating prediction* task, we compare DHCD against all the baselines. For *review generation* task, we compare DHCD with CF-GCN since it is the only baseline with review generation capacity. Moreover, we also use word-based LSTM (W-LSTM) and character-based LSTM (C-LSTM) [14] as baselines for performance comparison.

Significance Testing: Hypothesis testing is used to ensure if prediction performance of our model is significantly different from the baselines. In this test, for each metric, we select the method whose performance is nearest to our DHCD for comparison.

Default Parameter Setting: Number of latent factors for users and items is set to 20 for all models. For the models using multilayer perceptron (i.e. all models except MF), the number of layers Q is equal in all domains. The capacity of layers of multi-layer perceptron are set to 64, 32, 16, and 8. Embedding size of each word is 50. For models using LSTM (i.e. DHCD, CF-GCN, C-LSTM, W-LSTM), numbers of layers in LSTM is set to 2 and its hidden size is set to 128. We assume that rating and review contributions are equal. So, we set $\lambda_r = \lambda_g = 1$ and the regularization term $\lambda = 0.01$ (see Equation 4). To learn model parameters, we use ADAM [5] with learning rate 10^{-4}.

4.3 Rating Prediction Performance

Performance Measures: We use *recall at K (r@K)*, *mean absolute error (MAE)*, and *root mean square error (RMSE)* to measure the performance of our proposed model and the baselines.

Cross-Domain Results: Table 2 shows our model's and baselines' performance in cross domain setting. For *r@K*, we assume ratings greater than 3 as positive. We apply paired t-test [10] for significance testing. From the table, we observe that our model significantly outperforms all baselines in all metrics. For example, the performance of our model is 7% better than the one of CDL model in term of MAE. Performance of CDL is generally better than cdNCF which indirectly infers that using reviews can help us to enhance the prediction performance. Similarly, CDN outperforms cdNCF which emphasize the importance of modeling items into domains in a cross-domain design, instead of simply merging different domains' data. Our model i.e. DHCD unifies review and rating in a cross-domain design. Hence, it achieves higher performance than other baselines.

	Book + Digital Music					Book + Office Products					Digital Music + Office Products				
	r@10	r@50	r@100	MAE	RMSE	r@10	r@50	r@100	MAE	RMSE	r@10	r@50	r@100	MAE	RMSE
cdMF	0.028	0.116	0.158	0.91	1.15	0.046	0.146	0.183	0.895	1.21	0.021	0.083	0.214	1.031	1.327
cdNCF	0.05	0.215	0.301	0.812	0.96	0.056	0.203	0.297	0.824	0.982	0.033	0.102	0.258	0.825	1.211
cdCDL	0.052	0.235	0.318	0.796	0.98	0.061	0.241	0.315	0.771	0.969	0.037	0.11	0.274	0.78	1.19
cdCF-GCN	0.062	0.243	0.325	0.765	0.977	0.07	0.256	0.324	0.772	0.921	0.046	0.135	0.288	0.705	1.107
CDN	0.047	0.226	0.328	0.762	0.974	**0.08**	0.204	0.296	0.773	0.976	0.041	0.144	0.302	0.754	1.084
DHCD	**0.073****	**0.252***	**0.335***	**0.741***	**0.914***	**0.08***	**0.279***	**0.356***	**0.745***	**0.891***	**0.05***	**0.157***	**0.324***	**0.698***	**1.013***

Table 2: Prediction Performance. For $r@K$, the larger the value, the better the model. For MSE and RMSE, the lower the better. Notation * and ** denote $p < 0.05$ and $p < 0.01$ respectively in significance test.

	Book(B) + Digital Music(DM)									
	r@10		r@50		r@100		MAE		RMSE	
	B	DM	B	DM	B	DM	B	DM	B	DM
MF	0.03	0.031	0.110	0.122	0.159	0.159	0.95	0.98	1.18	1.16
NCF	0.057	0.059	0.233	0.206	0.30	0.292	0.766	0.732	0.98	0.95
CDL	0.055	0.061	0.244	0.242	0.318	0.322	0.771	0.781	0.97	0.92
CF-GCN	0.070	0.068	0.245	0.248	0.301	0.35	0.752	0.745	0.94	0.911
CDN	0.043	0.049	0.219	0.234	**0.312**	0.339	0.772	0.751	0.991	0.933
DHCD	**0.071***	**0.077****	**0.25***	**0.257***	0.31	**0.342***	**0.749***	**0.735***	**0.93***	**0.902***

Table 3: Prediction Performance of DHCD and baselines in Book + Digital Music dataset. Notation * and ** denote $p < 0.05$ and $p < 0.01$ respectively in significance test.

	Book + Digital Music		Book + Office Products		Digital Music + Office Products	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
CDN	0.767	0.97	0.767	0.986	0.743	1.052
DHCD	**0.751***	**0.94***	**0.75***	**0.922***	**0.725***	**1.031***

Table 4: Prediction Performance in Cold-start setting. Notation * denotes $p < 0.05$ in significance test.

Single-Domain Results: Table 3 shows rating prediction performance of DHCD and baselines in single domain set of experiments. We only show the result for Book + Digital Music dataset to save space since the experiments on the other two show the similar result. For the baselines like MF, NCF, CDL, and CF-GCN, the training data and test data are from the same domain. For DHCD and CDN, we use the training data from both domains and report the performance in each domain separately. The first observation that we observe from the table is that for baseline models that are single-domain by design, their performance in single domain is better than in cross domain. For example, the $r@10$ of MF is 0.03 in Book domain. However, this method only achieves 0.028 in domain Book + Digital Music domains (Table 2). It indirectly implies that using heterogeneous data without proper integration can harm the performance of models. Secondly, in general, our DHCD model outperforms the baselines in each domain's performance. It implies that DHCD has the proper manner to fuse the ratings and reviews of users and items from different domains under one framework. Thirdly, almost in all models, but more specifically in DHCD, performance in smaller domains is better than the one of larger domains. For example, Book domain is larger than Digital Music domain and the performance of CF-GCN in Book domain is worse than the one in Digital Music. It suggests that larger domain contains more noise than the smaller ones, and that the smaller datasets may benefit more from information transfer.

In general, the performance of DHCD is significantly better than the best baseline in significance test for both cross-domain and single domain settings.

Rating Prediction in Cold-start: To conduct this experiment, we keep the same test set and remove users with more than 5 ratings in both domains from training set and use default parameter settings.

	Book + Digital Music	Book + Office Products	Digital Music + Office Products
C-LSTM	3.10	3.09	3.05
W-LSTM	3.12	3.06	3.02
CF-GCN	3.02	2.99	2.95
DHCD	**2.93**	**2.95**	**2.88**

Table 5: Perplexity comparison between our model and baselines (Lower is better).

Table 4 shows the result of our model in cold-start setting. Due to the space limitation, we only provide the performance of CDN for comparison on MAE and RMSE. As shown in the table, DHCD significantly outperforms CDN in all three datasets. This shows that using reviews in our model adds extra valuable information for predicting user ratings, compared to the CDN model that only uses rating information across the two domains.

4.4 Review Generation Analysis

In this section, we use perplexity as a measurement for review generation. The lower the perplexity, the better the model.

$$ppx = \exp\left(-\frac{1}{N}\sum_{(u,i)}\frac{1}{J_{ui}}\sum_{c=1}^{J_{ui}}\log p(t_c|t_{<c},\Phi)\right) \tag{5}$$

where the pair (u, i) is a pair of user and item from test set and N is the number of reviews in test set and J_{ui} is the number of words of each review between u and i. Φ denotes the parameters.

Result: Table 5 shows the perplexity of DHCD and the baselines C-LSTM, W-LSTM and CF-GCN. As shown in the table, perplexity of our model is lower than the one of baselines in the three datasets. For example, the perplexity of DHCD is 6% better than W-LSTM. Therefore, it suggests the latent representations of users and items learned from multi-layer perceptron are able to encode the review generation process. Moreover, the performance of DHCD is better than the one of CF-GCN which implies the domain consideration is useful to generate reviews.

4.5 The Effect of Reviews in Training

In this section, we investigate the impact of reviews in training our model. To do so we compare the rating regression training loss of CDN and our model through epochs (Equation 2). CDN can be considered as a simplified version of our model without using reviews. The faster the convergence of the training loss, the better the method. The parameters are kept as default values.

Experimental Results: Figure 2 shows the training loss of rating regression of DHCD and CDN through 50 epochs on the three datasets. From the figure, our first observation is that the training loss decreases when the number of epoch increases and it reaches

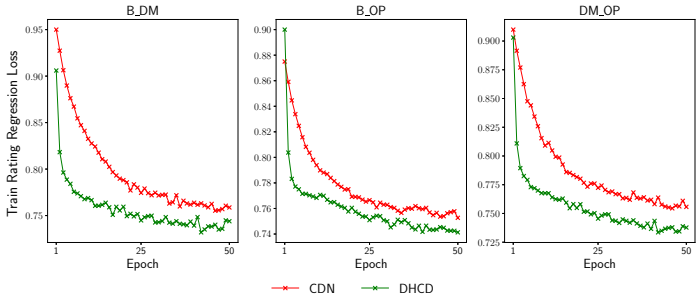

Figure 2: Training loss of rating regression of CDN and our model (DHCD) in the three datasets: Book + Digital Music (B_DM), Book + Office Products (B_OP) and Digital Music + Office Products (DM_OP) through epochs.

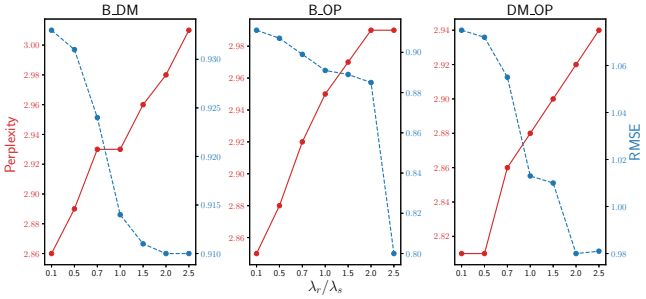

Figure 3: Perplexity and RMSE of DHCD with different ratio λ_r/λ_s in the three datasets: Book + Digital Music (B_DM), Book + Office Products (B_OP) and Digital Music + Office Products (DM_OP). For both metrics, the lower the better.

a stable value after some certain numbers of epoch. Secondly, the two methods seem to converge to a fixed point in the three datasets. However, DHCD converges faster than CDN method. For instance, after 10 epochs, our method seems to be close to the convergence but CDN needs 25 epochs to have the same behavior on the dataset of Book + Digital Music. From the result, we can conclude that reviews are actually helpful for the learning of DHCD.

4.6 The Balance between Rating Prediction and Review Generation

In DHCD, λ_r and λ_s are used to control the trade-off between rating prediction and review generation tasks. To study their effects, we keep $\lambda_s = 1$ and use various values of λ_r for training, then, we measure the performance of DHCD on test set. The two metrics perplexity and RMSE are selected for evaluation.

Experimental Results: In this experiment, the values of ratio λ_r/λ_s are $\{0.1, 0.5, 0.7, 1.0, 1.5, 2.0, 2.5\}$. The result is plotted on Figure 3. From the figure, we observe that increasing the ratio λ_r/λ_s leads to the better RMSE and worse perplexity since the larger value of the ratio means that the more effort is used for rating prediction. Moreover, the phenomenon is similar for the three datasets.

5 CONCLUSION

In this paper, we have proposed Deep Hybrid Cross Domain (DHCD) recommendation system which captures the reviews and ratings of users and items across different domains. Through our extensive experiments, DHCD outperforms the baselines on rating prediction and review generation tasks.

There are several directions to extend our work further. DHCD has not considered the sequence of users' decision and the social effect of users' friends on their decision. Thus, these interesting directions should be studied in the near future especially to address the data sparsity issues.

REFERENCES

[1] Shuo Chang, F. Maxwell Harper, and Loren Gilbert Terveen. 2016. Crowd-Based Personalized Natural Language Explanations for Recommendations. In *RecSys*.
[2] Thanh-Nam Doan and Ee-Peng Lim. 2018. PACELA: A Neural Framework for User Visitation in Location-based Social Networks. In *UMAP*.
[3] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *WWW*.
[4] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish Conference on Information Retrieval*.
[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.
[6] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *WWW*.
[7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*.
[8] Reinhard Heckel, Michail Vlachos, Thomas Parnell, and Celestine Dünner. 2017. Scalable and interpretable product recommendations via overlapping co-clustering. In *ICDE*.
[9] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2, 5 (1989).
[10] Henry Hsu and Peter A Lachenbruch. 2014. Paired t test. *Wiley StatsRef: Statistics Reference Online* (2014).
[11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* (2009).
[12] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*.
[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[14] Jianmo Ni, Zachary C Lipton, Sharad Vikram, and Julian McAuley. 2017. Estimating reactions and recommending products with generative models of reviews. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 783–791.
[15] Jianmo Ni and Julian McAuley. 2018. Personalized review generation by expanding phrases and attending on aspect-aware representations. In *ACL*.
[16] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction.. In *AAAI*.
[17] Denis Parra and Shaghayegh Sahebi. 2013. Recommender systems: Sources of knowledge and evaluation metrics. In *Advanced Techniques in Web Intelligence-2*.
[18] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. Recommender systems: introduction and challenges. In *Recommender systems handbook*. Springer, 1–34.
[19] Shaghayegh Sahebi and Peter Brusilovsky. 2015. It Takes Two to Tango: An Exploration of Domain Pairs for Cross-Domain Collaborative Filtering. In *RecSys*.
[20] S. Sahebi and T. Walker. 2014. Content-Based Cross-Domain Recommendations Using Segmented Models. In *Workshop on New Trends in Content-based Recommender Systems (CBRecsys)*. ACM, 57–63.
[21] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *SIGIR*.
[22] Tianhang Song, Zhaohui Peng, Senzhang Wang, Wenjing Fu, Xiaoguang Hong, and S Yu Philip. 2017. Review-Based Cross-Domain Recommendation Through Joint Tensor Factorization. In *DASFAA*.
[23] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *SIGKDD*.
[24] Xin Xin, Zhirun Liu, Chin-Yew Lin, Heyan Huang, Xiaochi Wei, and Ping Guo. 2015. Cross-Domain Collaborative Filtering with Review Text.. In *IJCAI*.
[25] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* (2019).
[26] Yongfeng Zhang and Xu Chen. 2018. Explainable Recommendation: A Survey and New Perspectives. *CoRR* abs/1804.11192 (2018). arXiv:1804.11192