Improving Cross-domain Recommendation through Collaborative Rating Network

Zongze Jin
School of Cyber Security,
University of Chinese Academy of Sciences
Institute of Information Engineering,
Chinese Academy of Sciences
Beijing, China
Email: jinzongze@iie.ac.cn

Yun Zhang*
School of Cyber Security,
University of Chinese Academy of Sciences
Institute of Information Engineering,
Chinese Academy of Sciences
Beijing, China
Email: zhangyun@iie.ac.cn

Weimin Mu
School of Cyber Security,
University of Chinese Academy of Sciences
Institute of Information Engineering,
Chinese Academy of Sciences
Beijing, China
Email: muweimin@iie.ac.cn

Weiping Wang
Institute of Information Engineering,
Chinese Academy of Sciences
Beijing, China
Email: wangweiping@iie.ac.cn

Hai Jin
School of Computer Science
and Technology, Huazhong
University of Science and Technology
Wuhan, China
Email: hjin@hust.edu.com

***Abstract*—Recently, cross-domain recommendation is one of the effective solutions to alleviate sparsity problems. Existing cross-domain recommendation algorithms focus on sharing the common latent rating pattern in multiple domains, while ignoring the domain-specific knowledge about the rating patterns. In addition, they only focus on the same users in different domains. In this paper, we present Collaborative Rating Network (CRN) which can not only leverage non-linear approach to capture common rating patterns, but also considering the domain-specific influences about rating patterns. Besides, we create a brigde which uses user information to concatenate different domains. Experimental results conducted on a variety of datasets demonstrate that our model significantly outperforms all baseline recommender systems.**

I. Introduction

Recommender system is an effective technique to help users find what they want [1], [27] in information overload scenarios. Traditional recommender systems based on collaborative filtering (CF) [13], [24] utilize the users' historical behavior to recommend in a single domain(e.g., movie and book). Although the CF techniques have shown good performance, such as Matrix Factorization (MF) [13], [23], they still have limitations. The performance of MF, which factorizes the user-item matrix into a low-rank user matrix and a low-rank item matrix, depends on the numeric ratings. However, many studies [17], [25], [31] have shown that the rating matrix is usually very sparse in real scenarios. The primary reason for this problem is that users can only rate a limited number of items and most of items cannot be rated. In addition, due to some unpopular items and inactive users, data distribution is uneven. Thus, with the explosive growth of user-generated data, many researches focus on how to combat data sparsity [8].

*Corresponding author

Cross-domain recommendation models are highly effective means to combat sparsity [26], which transfer and share knowledge among the different domains can be beneficial. The early researches [15], [16], [22] utilize data in source domain to alleviate data sparsity of the target domain. Unfortunately, these methods, which take the knowledge of the source domain as a prior knowledge of the target domain, do not consider the characteristics of different domains. Subsequent researches [2], [12], [18], [28] have considered different knowledge in multi-domains, ignoring the boundaries between source and target domains, moreover, they combat data sparsity through the sharing the knowledge of source and target domains. These models assume that multiple domains share a common latent rating pattern. Since the knowledge between multi-domains cannot be replaced with each other, the related domains do not need to use the common rating pattern. Thus, the influence of mutual rating pattens among specific-domains should be worthy of attention, and it is obvious that existing approaches are not.

In this paper, we propose a novel method, named Collaborative Rating Network(CRN), to learn the shared knowledge and not-shared effect of each domain simultaneously to improve cross-domain recommendation. For example, movies and books can be both classified by themes, such as science. However, movies can create good visual feelings, neither do books. We consider the common rating patten between movies and books, meanwhile we should capture the specific rating patten about visual feelings in movie domain. In addition, our contributions can be summarized as follows:

– We present a deep model, CRN, which can not only capture the common rating pattern shared across domains, but also capture the domain-specific rating patterns in each domain.

– We adopt user information through our model to concatenate different domains.

– We compare CRN with the state-of-the-art methods on three real world datasets to demonstrate its effectives.

– We demonstrate that CRN can do better on alleviating the sparsity problem.

The rest of the paper is organized as follows. In Section 2, we represent the related work about CRN. Besides, problem definitions about our model are presented in Section 3. Then we describe CRN in detial in Section 4. In Section 5, we present our experiments and analyze the experimental results. Finally, conclusions are presented in Section 6.

II. Related Work

In this section, we give a short overview of two research areas about our approach: cross-domain recommendation and deep neural network.

A. Cross-domain Recommendation

Since the recommender systems always suffer from data sparsity, the cross-domain model has been widely studied by the communities as one of the effective methods to combat the sparsity. Existing methods for cross-domain recommendation fall into two main types, dealing with the data sparsity problem in different scenarios.

The one is the direct transfer approach, which transfers directly knowledge or patterns from the source domain to the target domain. The earliest research is CBT [15], which takes the knowledge of the source domain as a prior knowledge of the target domain. Then the team has proposed a later extension called RMGM [16]. It is a cross-domain model which transfers and shares the common rating pattern by the cluster-level ratings across multiple domains. Transfer Learning for Multiple Domains (TLMD) [20] predicts the missing values in the target domain by extracting knowledge from multiple source domains and transferring it to a single sparse target domain. Coordinate System Transfer (CST) [22] transfers both user and item knowledge from an auxiliary domain.

The other is the common transfer approach, which ignores the boundaries between source and target domains and distinguishes domain-specific factors from domain-sharing factors. Collective Matrix Factorization (CMF) [28] factors each relation matrix with a generalized-linear link function, but whenever an entity type is involved in more than one relationship and ties factors of different models together. In [12], it proposes a context-dependent matrix factorization model, HETEROMF, that extends CMF [28] and considers a general latent factor for every entity type and a context-dependent latent factor for every context in which the entity is involved. Hyper-Structure Transfer (HST) [18], that captures the non-linear correlation of knowledge between domains and allows the non-linearly correlated knowledge between domains to be identified and transferred. Localized Matrix Factorization (LMF) [2] enhances predictions in multi-context recommender problems through a localized latent factor model.

B. Deep Neural Network for Recommendation

With the development of deep learning, deep neural networks (DNN) have gained significant success in many applications. We list the main contributions of DNN in the recommender system. Neural Collaborative Filtering (NCF) [11] is such a framework aiming to capture the non-linear relationship between users and items. Neural Social Collaborative Ranking (NSCR) [29] is developed based on the recent advance of NCF [11], which is further extended to model the cross-domain social relations by combining with the graph regularization technique [10]. Google [6] proposes the Wide and Deep Model, which can solve both regression and classification problems. The wide learning component is a single layer perceptron which can also be viewed as a generalized linear model. And the deep learning component is multilayer perceptron. Wide and Deep Model combines two learning techniques to capture both memorization and generalization. Guo et al [9] proposes an end-to-end model, the Deep Factorization Machine (DeepFM) , which seamlessly integrates factorization machine and MLP. In addition, Chen et al. [5] proposes an attentive collaborative filtering model by leveraging a two-level attention mechanism to find the key factors. In makeup recommendation, Alashkar et al. [3] proposes a DNN model which uses two identical MLPs to model labeled examples and expert rules respectively. In YouTube recommendation, Covington et al. [7] applies DNN model to enhance the performance.

III. Problem Definitions

In this section, we present some definitions and propose definitions of our problems in this paper.

a) ***Definition 1****: (Domain)* A domain is a collection of ratings which are drawn under the same data distribution.

In this paper, we regard ratings collected from one dataset as a domain. This definition is proposed by [16], and [30] has cited it. For example, we can consider that different domains represent ratings collected from different types(e.g. Movies or Books).

b) ***Definition 2****: (Cross-domain)* Cross domain is the shared space of source domain and target domain.

Given a set of ratings collected from A domain and another set of ratings collected from B domain. We regard A as the source domain and B as the target domain. We gain the cross-domain D which is the related domain between A and B.

c) ***Problem Definition****:* In cross-domain D, there is a set of users $U = \{u_1, u_2, \cdots, u_k\}$ to rate a set of items from the source domain A. How to predict the ratings of the target domain through the cross domain D. In addition, how to save the domain-specific rating pattern while capturing the common rating pattern.

TABLE I
IMPORTANT NOTATIONS

Symbols	Definations and Descriptions
1-subnetwork	common rating pattern subnetwork
2-subnetwork	collaborative rating pattern subnetwork
U_s	user index in the source domain
U_t	user index in the target domain
F_U	user information in the 1-subnetwork
F_UI	user-item interaction information in the 1-subnetwork
F_CI	item information in the 1-subnetwork
W_l	weight of the DNN in the 1-subnetwork
b_l	the bias of DNN
W_{out}	the weight of output layer in 1-subnetwork
b_{out}	the bias of the output layer
r^*	the predicted rating in 1-subnetwork
I_t	item index in the target domain
W_{Tl}	weight of DNN in the 2-subnetwork
b_{Tl}	the bias of DNN in the 2-subnetwork
f_t	the output feature in 2-subnetwork
f_c	the output feature in 1-subnetwork
b_t	the bias term
R^*	the predicted rating in CRN

In our proposed model,CRN, we indicate how to predict the missing ratings while considering the interest of multi-domains by transferring correlated knowledge across domains.

IV. ALGORITHM

In real-world scenarios, for items, we observe the domain-specific features and the common features exist simultaneously. Like movies and books, they can be both classified by themes, but books cannot replace the visual feelings of movies. Also themes (the common feature) and the visual feelings of movies(the domain-specific feature) affect the rating results of movies in a certain proportion. Thus we present the architecture of CRN in Figure 1, which is a unified architecture consisting of two subnetworks: *Common Rating Pattern Subnetwork* and *Collaboratives Rating Pattern Subnetwork*.

In this section, we describe the important notations in Table1. Then we present two key parts separately in Section 3.1 and Section 3.2. At last, we describe how to train CRN.

A. Common Rating Pattern Subnetwork

Common Rating Pattern Subnetwork is a architecture, which we leverage the data from the source domain and target domain, for capturing the common rating pattern in multiple domains.

There are two goals in this section. The one is that we need to gain the features of the set of users $U = \{u_1, u_2, \cdots, u_k\}$, and the other is that the common rating pattern feature should be captured by this subnetwork.

*a) **Embedding layers**:* In this layer, we index the set of users from source domain $U_S = \{u_{s1}, u_{s2}, \cdots, u_{sk}\}$ and index the set of users from target domain $U_T = \{u_{t1}, u_{t2}, \cdots, u_{tk}\}$. Then we get the user-pair $(U_S, U_T) = ((u_{s1}, u_{t1}), \cdots, (u_{sk}, u_{tk}))$ from multiple domains. Same as the users, we can get the items pair $(I_S, I_T) = ((i_{s1}, i_{t1}), \cdots, (i_{sk}, ui_{tk}))$.

*b) **Deep Neural Network layers**:* Through the embedding layers, we can get the input vectors. And then we present the key of our subnetwork. In the first deep neural network (DNN) layer, we present three key parts: *User Feature* F_U, *User-item Interaction Feature* F_{UI} and *Item Feature* F_{CI}. *User Feature* is the core part to combine source domain with target domain in CRN and it is the first goal of this subnetwork. In the next layer, we concatenate three features and use Rectified Linear Units(ReLU) [21] as the activation function.

$$x_0 = concatenate(F_U, F_{UI}, F_{CI}) \tag{1}$$

$$x_l = ReLU(W_l x_{l-1} + b_l) \tag{2}$$

where W_l is the weight matrix between the l-th layer and the (l-1)-th layer, and b_l is the bias term. Followed by several layers of fully connected ReLUs, we gain the output of the subnetwork.

*c) **Output layers**:* Through several DNN layers, we gain the output of the subnetwork.

$$r^* = softmax(W_{out} x_{out} + b_{out}) \tag{3}$$

We use $Loss_{com}$ as the loss function.

$$Loss_{com} = \frac{1}{2}[\sum_{ij}^{N}(\frac{r_s + r_t}{2} - r^*)^2] \tag{4}$$

where r_s denotes the rating from the source domain, r_t denotes the rating from the target domain, r^* denotes the prediction of the subnetwork, W_{out} and b_{out} denotes the weight and bias of the output layer. We minimize the loss from the source domain and target domain to capture the common rating pattern. When the subnetwork is successfully trained, we achieve the second goal that the feature of common rating pattern is captured.

When we achieve two goals of this subnetwork, we can use the second subnetwork, Collaborative Rating Pattern Subnetwork, to predict the ratings.

B. Collaborative Rating Pattern Subnetwork

Collaborative Rating Pattern Subnetwork utilize the user information and the common rating feature from the former subnetwork to combine the data of the target domain to predict the missing ratings. In this section, we describe the embedding layers, DNN layers, shared layers and the output layers.

*a) **Embedding layers**:* Different from the former subnetwork, we only index the set of items $I_t = \{i_1, i_2, \cdots, i_k\}$ of the target domains. We adopt the user information F_U of the former subnetwork and I_t as the input of DNN layers. The input vector x_{T0} is as follows.

$$x_{T0} = concatenate(F_U, I_t) \tag{5}$$

*b) **Deep Neural Network layers**:* In these layers, features are concatenated into a wide first layer, followed by some layers of fully connected ReLU. And then we gain the output of this subnetwork.

$$x_{Tl} = ReLU(W_{Tl} x_{Tl-1} + b_{Tl}) \tag{6}$$

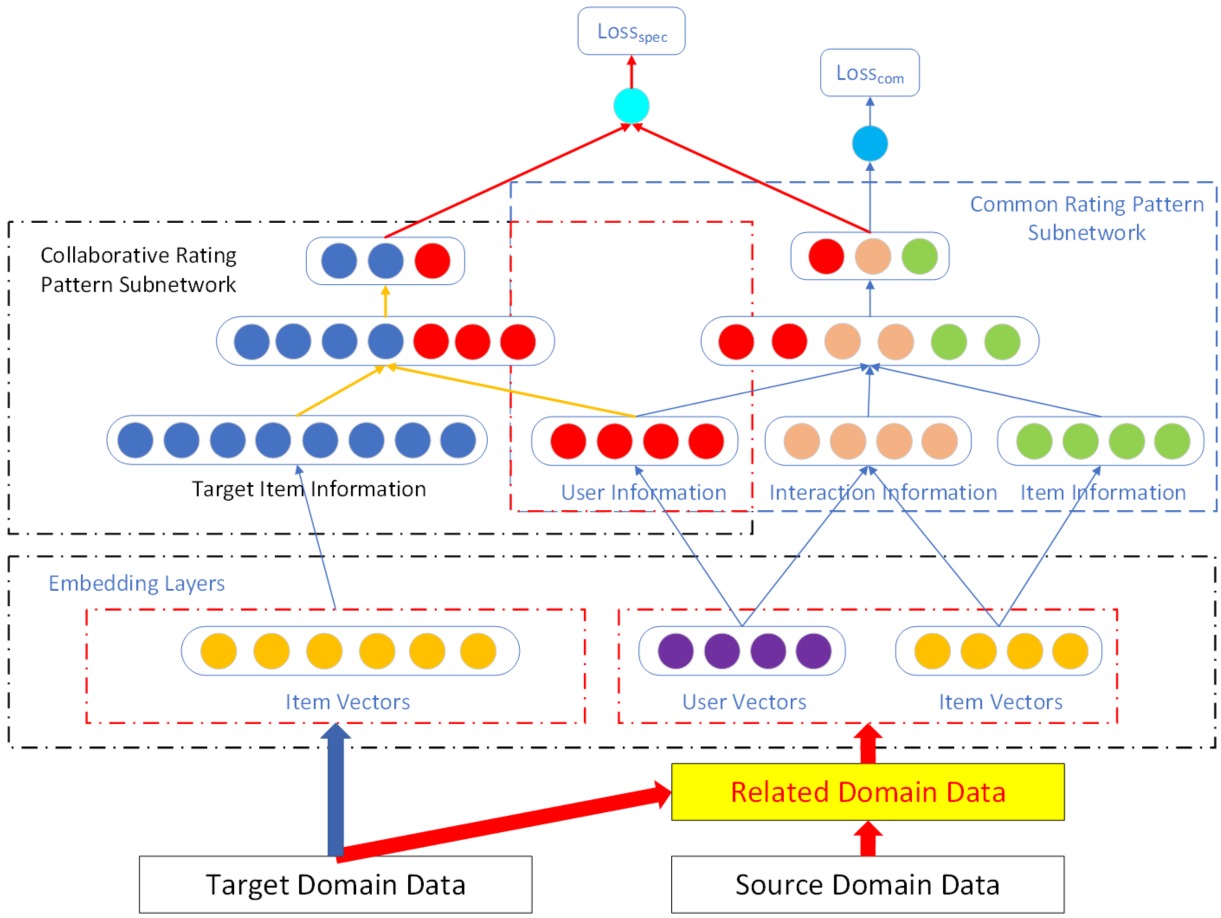

Fig. 1. The Architecture of CRN

*c) **Shared layers**:* We concatenate the common rating feature f_c from the common rating pattern subnetwork and the dense vector of DNN in target domain. In this layer, we do not only capture the specific-domain knowledge, but also capture common knowledge.

$$R^* = Softmax(W_t * f_t + W_c * f_c + b_t) \tag{7}$$

where W_t denotes the weight of the DNN output feature f_t in target domain, W_c represents the weight of the common rating feature f_c.

*d) **Output layers**:* Through the shared layers, we get the output of the subnetwork. Different from the loss function $Loss_{com}$ of the former subnetwork, we use loss function $Loss_{spec}$ in this subnetwork.

$$Loss_{spec} = \frac{1}{2}\sum_{i,j}^{N}(R_{ij} - R^*_{ij})^2 \tag{8}$$

where R_{ij} represents the real ratings and the R^*_{ij} is the prediction. $Loss_{spec}$ follows [19], where $\delta_{ij} = 0$ when $R_{ij} = 0$; otherwise,$\delta_{ij} = 1$.

Through two subnetwork, we achieve our goal to consider the common rating pattern and capture the specific-domain rating pattern.

C. Network Training

In our model, we firstly train the Common Rating Pattern Subnetwork. When Common Rating Pattern Subnetwork has finished training, we begin to train the second subnetwork, Collaborative Rating Pattern Subnetwork. We take the derivative of the loss with respect to the whole set of parameters through back-propagation, and we use stochastic gradient descent (SGD) [4] with mini-batch to update the parameters. We use dropout to avoid the neural network being over-fitting.

V. Experiment

In this section, we first describe three real datasets in our experiments. Then in Section 4.2 and 4.3, we present the baselines and metrics. Experiments settings are illustrated in Section 4.4. Last but not least, we give the experimental results and analyze our experiments.

A. Datasets

For the experiments we have used the following benchmark real-world datasets for performance evaluation:

MovieLens-1M dataset[1]: It contains more than 1,000,000 movie ratings with the scales from 1 to 5 provided by 6040 users on 3952 movies.

EachMovie dataset[2] : It contains 2.8 million movie ratings with the scales from 1 to 6 provided by 72,916 users on 1,628 movies.

Book-Crossing dataset[3] : It contains more than 1.1 million ratings with the scales from 0 to 9 provided by 278,858 users on 271,379 books.

B. Baselines

In this section, we evaluate our model by comparing with several state-of-the-art approaches on cross-domain recommendation:

NMF [14]: The Non-negative Matrix Factorization (NMF) utilizes non-negative matrix factorization approach, which learns the latent factors in each domain, to predict the ratings.

PMF [23]: Probabilistic Matrix Factorization (PMF) presents probabilistic algorithms that scale linearly with the number of observations and perform well on very sparse and imbalanced datasets.

RMGM [16]: Rating-Matrix Generative Model (RMGM) is a cross-domain model which transfers and shares the common rating pattern by the cluster-level ratings across multiple domains.

CST [22]: Coordinate System Transfer (CST) transfers the knowledge from source domain to target domain to combat the sparsity.

LFM [2]: The Localized Factor Models (LFM), that each entity has a global latent factor shared across domains, is a state-of-the-art approach for cross-domain recommendation to capture the common knowledge.

C. Metric

To check the performances of different methods, we use MSE (Mean Squared Error) as the evaluation metrics. The expression are as follows :

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(r_i - r_i^*)^2 \tag{9}$$

where N denotes the number of test ratings, r_i is the true value and r_i^* is the predicted rating. The smaller the value of MSE is, the better the model performs.

D. Settings

Before the experiments, we randomly select 1000 users with more than 20 ratings for each item. In addition, we normalize the rating scales from 1 to 5 for fair comparison. We randomly choose 700 users in three real datasets to train CRN, and utilize 300 users to test our model. And we need to choose the three dataset to be related domains to discover. We regard *MovieLens* vs *Book-Crossing* (MB), *EachMoive* vs *Book-Crossing* (EB) and *MovieLens* vs *EachMovie* (ME) as three related domains. To avoid the cold-start, we set different fractions from 10% to 50%. Through tuning the parameters of different models, we set $K = 100$ and $L_1 = 10$, $L_2 = 15$ and $T = 10$.

In Common Rating Pattern Subnetwork, we set the number of hidden layers is 3. There are 128 neurons in the first layer. The User Feature neurons is 48, the Item Feature is 40 and the interaction feature is 40. And there are 64 neurons in the sencond hidden layer and 32 neurons in the third hidden layer. The output layers is 5 neurons. In Collaborative Rating Pattern Subnetwork, we set the number of hidden layers is also 3. In the first layer, we set the neurons of item features from target domain is 80. And the second layer, the number is 64. There are 32 neurons in the third layer and 5 neurons in the output layer. We adopt Dropout to avoid overfitting, the number is 0.2.

E. Results

In this section, we present the performance evaluation of our experiments and analyze the parameters of our model.

a) ***Performance Evaluation****:* The experimental results are summarized in Table 1. In our experiments, we use the related domains to train Common Rating Pattern Subnetwork and use the one of the related domain datasets to train Collaborative Rating Pattern Subnetwork. For example, we use MB (Movielens and BookCrossing) as the related domain. For the target domains, we select MovieLens and Book-Crossing respectively.

Table 1 lists the MSE preformances of the compared methods. In our experiments, we have carried out the experiments about 20 times in three related-domain datasets. Meanwhile, NMF as the typical traditional MF approach is worst in our baselines. And PMF relying on prior knowledge is better than it in each dataset.

In addition, our model substantially outperforms the baseline approaches. By observing the experimental results of Table 1, we can draw the following conclusions:

(1) From the experimental results, we can find that our proposed method, CRN, is the best performing method among all the models.

(2) The cross-domain recommendation models such as RMGM, LFM and CST are better than the single domain models. It demonstates the latent common rating pattern can gain more useful information than the single-domain models.

(3) CRN is even better than the state-of-the-art methods. It demonstates that our approach by capturing the common rating pattern and the specific-domain rating pattern can enhance the cross-domain recommendation.

b) ***Impact of Parameters****:* Figure 2 shows the impact of parameters in our model. User information F_U is the key factor to combine Common Rating Pattern Subnetwork with Collaborative Rating Pattern Subnetwork. So we firstly describe the dimension of the user information. We fix the item information in Common Rating Pattern Subnetwork, we set it 40. And we can observe the changes of the user information.

[1]http://www.grouplens.org/node/73
[2]http://www.cs.cmu.edu/lebanon/IR-lab.htm
[3]http://www.informatik.uni-freiburg.de/cziegler/BX/

TABLE II
MSE Performances of the compared models

Related Domains	Target Domain	Methods	50%	40%	30%	20%	10%
MB	MovieLens	NMF	0.9011	0.8802	0.8231	0.7731	0.7643
		PMF	0.8912	0.8602	0.7903	0.7721	0.7698
		RMGM	0.8321	0.8147	0.7832	0.7609	0.7496
		CST	0.8207	0.8062	0.7753	0.7607	0.7402
		LFM	0.7993	0.7899	0.7532	0.7421	0.7201
		CRN	**0.7632**	**0.7421**	**0.7317**	**0.7108**	**0.7006**
	Book-Crossing	NMF	0.7891	0.7854	0.7532	0.7321	0.7211
		PMF	0.7799	0.7734	0.7512	0.7337	0.7201
		RMGM	0.7612	0.7597	0.7214	0.7198	0.7023
		CST	0.7632	0.7588	0.7123	0.7077	0.6986
		LFM	0.7326	0.7302	0.7115	0.7030	0.6897
		CRN	**0.7289**	**0.7245**	**0.7012**	**0.6923**	**0.6796**
EB	EachMoive	NMF	0.9765	0.9679	0.9571	0.9132	0.8995
		PMF	0.9123	0.9118	0.8862	0.8691	0.8597
		RMGM	0.9078	0.9002	0.8891	0.8601	0.8573
		CST	0.8974	0.8895	0.8642	0.8543	0.8433
		LFM	0.8901	0.8837	0.8529	0.8421	0.8311
		CRN	**0.8723**	**0.8705**	**0.8421**	**0.8301**	**0.8219**
	Book-Crossing	NMF	0.7762	0.7678	0.7421	0.7219	0.7122
		PMF	0.7657	0.7625	0.7439	0.7201	0.7089
		RMGM	0.7548	0.7501	0.7329	0.7189	0.7011
		CST	0.7532	0.7501	0.7311	0.7190	0.7007
		LFM	0.7433	0.7405	0.7289	0.7162	0.6902
		CRN	**0.7291**	**0.7195**	**0.6997**	**0.6852**	**0.6799**
ME	MovieLens	NMF	0.8518	0.8501	0.8375	0.8123	0.7991
		PMF	0.8502	0.8467	0.8214	0.8167	0.7898
		RMGM	0.8201	0.8165	0.8001	0.7801	0.7704
		CST	0.8118	0.8057	0.7923	0.7767	0.7611
		LFM	0.8009	0.7998	0.7867	0.7697	0.7601
		CRN	**0.7891**	**0.7869**	**0.7645**	**0.7532**	**0.7503**
	EachMovie	NMF	0.6833	0.6801	0.6671	0.6421	0.6349
		PMF	0.6735	0.6709	0.6632	0.6404	0.6331
		RMGM	0.6501	0.6459	0.6302	0.6217	0.6116
		CST	0.6378	0.6333	0.6211	0.6119	0.6093
		LFM	0.6229	0.6215	0.6102	0.6003	0.5912
		CRN	**0.6118**	**0.6077**	**0.5992**	**0.5801**	**0.5774**

Through the experiments, our model can achieve the best performance when we set 48 as the user information size. Same as the size of user information, the dimension of item feature in first subnet and second subnet is the other impact of our model. We fix the user information and set 48 as the size. From Figure 2, you can find that the change tends to be stable when the dimension size is sufficiently large. And this is why we adopt 40 as the dimension size in first subnetwork and 80 as the size in the second subnetwork.

*c) **Impact of Dropout**:* We adopt the dropout in CRN to avoid overfitting instead of regularizing model parameters. Figure 3 presents the performance MSE in first subnetwork (Common Rating Pattern Subnetwork) and the second subnetwork (Collaborative Rating Pattern Subnetwork) by varying the dropout ratio respectively. In the first subnetwork, we compare the related datasets. And in the second subnetwork, we compare the target domain. From Figure 3, we can observe that two subnetworks are overfitting when dropout ratio is 0. Moreover, using a dropout ratio of 0.3 and 0.2 leads to the best performance on our datasets, respectively. In our experiments, we adopt 0.2 as the dropout ratio for achieve the best performance.

*d) **Impact of Related Domain**:* In order to measure the impact of related domain, we adjust the proportion of each part in the relevant domain and show the results in Table 3. For example, $MB(1:9)$ denotes that the ratio of A and B is 1:9 respectively. From the Table 3, we observe that when the ratio is close to 5:5, the model can achieve good performance. Meanwhile, they reason why the EB groups changes smoothly, is that MovieLens and EachMovie are related with movies. They including interactive information are less than the others. When the data between the related domains differs greatly, adjusting the proportion of related domain data will have a great impact on the results.

VI. Conclusions

In this paper, we propose a novel cross-domain recommendation approach, named Collaborative Rating Network(CRN) model. The CRN model contains two subnetworks: Common Rating Pattern Subnetwork and Collaborative Rating Pattern Subnetwork. We utilize Common Rating Pattern Subnetwork to capture the common user information and the common rating pattern feature. And we can get the specific-domain rating pattern through Collaborative Rating Pattern Subnetwork. Meanwhile, we leverage this subnetwork to combine the specific-domain rating pattern with the common rating pattern feature to predict the ratings.

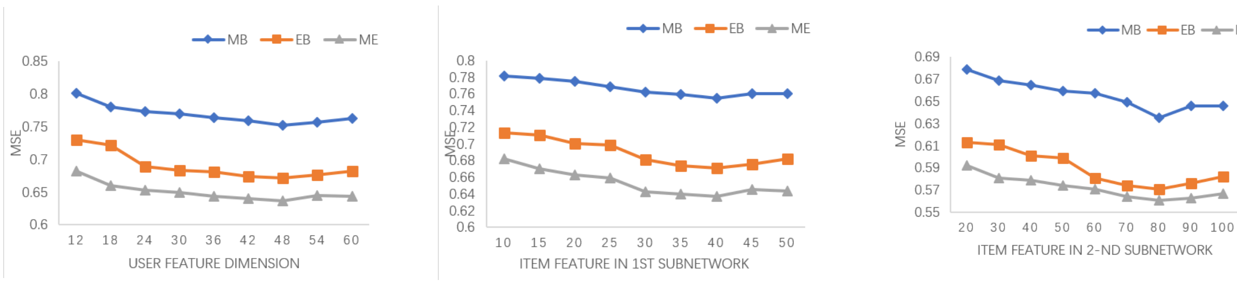

Fig. 2. The impact of common user information, item information from Common Rating Pattern Subnetwork and item information from Collaborative Rating Pattern Subnetwork

Fig. 3. The Impact of Dropout

TABLE III
MSE PERFORMANCES OF THE RELATED DOMAIN

MB(1:9)	MB(2:8)	MB(3:7)	MB(4:6)	MB(5:5)	MB(6:4)	MB(7:3)	MB(8:2)	MB(9:1)
0.892	0.851	0.765	0.741	0.738	0.755	0.779	0.846	0.901
EB(1:9)	EB(2:8)	EB(3:7)	EB(4:6)	EB(5:5)	EB(6:4)	EB(7:3)	EB(8:2)	EB(9:1)
0.911	0.879	0.821	0.754	0.729	0.761	0.792	0.848	0.889
ME(1:9)	ME(2:8)	ME(3:7)	ME(4:6)	ME(5:5)	ME(6:4)	ME(7:3)	ME(8:2)	ME(9:1)
0.802	0.789	0.762	0.736	0.733	0.741	0.773	0.79	0.792

The experimental results show that our proposed CRN odel indeed can benefit from the combination of the common rating pattern and the specific-domain rating pattern. Meanwhile, CRN outperforms the state-of-the-art approaches in cross-domain recommendation.

VII. ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (No.61502478), National High-Tech Research and Development Program of China (2013AA013204) and National HeGaoJi Key Project (2013ZX01039-002-001-001).

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] Deepak Agarwal, Bee Chung Chen, and Bo Long. Localized factor models for multi-context recommendation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 609–617, 2011.

[3] Taleb Alashkar, Songyao Jiang, Shuyang Wang, and Yun Fu. Examples-rules guided deep neural network for makeup recommendation. In *AAAI Conference on Artificial Intelligence*, pages 941–947, 2017.

[4] Shun Ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(45):185–196, 1993.

[5] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–344, 2017.

[6] Heng Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, and Mustafa Ispir. Wide and deep learning for recommender systems. In *The Workshop on Deep Learning for Recommender Systems*, pages 7–10, 2016.

[7] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. pages 191–198, 2016.

[8] Sheng Gao, Hao Luo, Da Chen, Shantao Li, Patrick Gallinari, and Jun Guo. Cross-domain recommendation via cluster-level latent factor model. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 161–176, 2013.

[9] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. pages 1725–1731, 2017.

[10] Xiangnan He, Tao Chen, Min Yen Kan, and Xiao Chen. Trirank:review-aware explainable recommendation by modeling aspects. In *In CIKM*, pages 1661–1670, 2015.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017.

[12] Mohsen Jamali and Laks Lakshmanan. Heteromf: recommendation in heterogeneous information networks using context dependent factor models. In *In WWW 13*, pages 643–654, 2013.

[13] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[14] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative

matrix factorization. In *International Conference on Neural Information Processing Systems*, pages 535–541, 2000.

[15] Bin Li, Qiang Yang, and Xiangyang Xue. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *IJCAI 2009, Proceedings of the International Joint Conference on Artificial Intelligence, Pasadena, California, Usa, July*, pages 2052–2057, 2009.

[16] Bin Li, Qiang Yang, and Xiangyang Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June*, pages 617–624, 2009.

[17] Jovian Lin, Kazunari Sugiyama, Min Yen Kan, and Tat Seng Chua. Addressing cold-start in app recommendation: latent user models constructed from twitter followers. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 283–292, 2013.

[18] Yan Fu Liu, Cheng Yu Hsu, and Shan Hung Wu. Non-linear cross-domain collaborative filtering via hyper-structure transfer. In *International Conference on International Conference on Machine Learning*, pages 1190–1198, 2015.

[19] Andriy Mnih and Ruslan R. Salakhutdinov. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, pages 1257–1264, 2007.

[20] Orly Moreno, Bracha Shapira, Lior Rokach, and Guy Shani. Talmud:transfer learning for multiple domains. pages 425–434, 2012.

[21] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on International Conference on Machine Learning*, pages 807–814, 2010.

[22] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. pages 230–235, 2010.

[23] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *International Conference on Neural Information Processing Systems*, pages 1257–1264, 2007.

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *International Conference on World Wide Web*, pages 285–295, 2001.

[25] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. 39(5):253–260, 2002.

[26] Oren Sar Shalom, Noam Koenigstein, Ulrich Paquet, and Hastagiri P. Vanchinathan. Beyond collaborative filtering: The list recommendation problem. In *International Conference on World Wide Web*, pages 63–72, 2016.

[27] Hua Wei Shen, Dashun Wang, Chaoming Song, and Albertlszl Barabsi. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 291–297, 2014.

[28] Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 650–658, 2008.

[29] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Item silk road: Recommending items from information domains to social users. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 185–194, 2017.

[30] Zihan Zhang, Xiaoming Jin, Lianghao Li, Guiguang Ding, and Qiang Yang. Multi-domain active learning for recommendation. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 2358–2364, 2016.

[31] Ke Zhou, Shuang Hong Yang, and Hongyuan Zha. Functional matrix factorizations for cold-start recommendation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 315–324, 2011.