Deeply Fusing Reviews and Contents for Cold Start Users in Cross-Domain Recommendation Systems

Wenjing Fu,[1] Zhaohui Peng,[1*] Senzhang Wang,[2] Yang Xu,[1] Jin Li[1]

[1] School of Computer Science and Technology, Shandong University, Qingdao, China
{fuwenjing, ljin}@mail.sdu.edu.cn, {pzh, xuyang0211}@sdu.edu.cn
[2] College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics, Nanjing, China
szwang@nuaa.edu.cn

Abstract

As one promising way to solve the challenging issues of data sparsity and cold start in recommender systems, cross-domain recommendation has gained increasing research interest recently. Cross-domain recommendation aims to improve the recommendation performance by means of transferring explicit or implicit feedback from the auxiliary domain to the target domain. Although the side information of review texts and item contents has been proven to be useful in recommendation, most existing works only use one kind of side information and cannot deeply fuse this side information with ratings. In this paper, we propose a Review and Content based Deep Fusion Model named RC-DFM for cross-domain recommendation. We first extend Stacked Denoising Autoencoders (SDAE) to effectively fuse review texts and item contents with the rating matrix in both auxiliary and target domains. Through this way, the learned latent factors of users and items in both domains preserve more semantic information for recommendation. Then we utilize a multi-layer perceptron to transfer user latent factors between the two domains to address the data sparsity and cold start issues. Experimental results on real datasets demonstrate the superior performance of RC-DFM compared with state-of-the-art recommendation methods.

Introduction

With the explosively growing amount of online information, recommender systems (Han et al. 2018; Zhang et al. 2014) have become an essential tool to help consumers find the commodities that suit their tastes. A preeminent recommender system can assist in retaining users and increasing sales. Unfortunately, recommendation algorithms are generally faced with data sparsity and cold start problems in that a user's feedback usually merely involves an extremely tiny part of the commodities on a website. As a promising solution to address these issues, cross-domain recommendation algorithms (Song et al. 2017; Wang et al. 2018) have gained increasing attention in recent years. This kind of algorithm tries to utilize explicit or implicit feedbacks from multiple auxiliary domains to improve the recommendation performance in the target domain.

*Corresponding author.

However, existing works related to cross-domain recommendation still have rough edges. Some single domain recommendation works (Guan et al. 2016; Park, Kim, and Choi 2013) have shown that side information such as review texts and basic descriptions could contribute to generating better latent features for the users and items so as to improve the recommendation performance. Nevertheless, most existing cross-domain recommendation works still do not take full consideration of different kinds of valuable side information. Although several recent works (Hao, Zhang, and Lu 2016; Xin et al. 2015) tried to utilize one kind of side information, they cannot fully and deeply fuse the side information with the rating matrix. In this paper, we take both the review texts and item contents into consideration and make them cooperate with ratings in a deep fusion way.

After watching a movie or purchasing a product, except for the explicit ratings, some users may also write reviews to express their feelings on the products. These reviews and the corresponding ratings complement each other, and the combination of them can more comprehensively reflect user behaviors and item properties. Compared with ratings, users' reviews can express more sentiment information in detail which could explain why one user gave this product such a rating. Recently, some studies in single domain recommendation (Zheng, Noroozi, and Yu 2017) have shown that the utilization of review texts can improve the performance of recommendation, especially for the items and users that have relatively fewer ratings.

Apart from user reviews, item contents can also contribute to describing user behaviors and item properties. Taking movies as example again, if a user gives a movie a high rating, we can naturally infer that the user thinks highly of the plot and other aspects of the movie, which can reflect the user's personal preference on movies. Even if the user does not give a high rating, we can still hold the view that he/she may be interested in this movie at the very beginning. Also, the plot of a movie, namely the movie's item content, can be the reflection of its property. Compared with the basic description such as movies' genres, plots contain richer semantic information. Item content has also been proven to be an aid to improve the performance of recommender systems (Wang, Wang, and Yeung 2015).

In recommender systems, both the rating matrices and the different kinds of side information we mentioned above can be too sparse to be effectively utilized. Luckily, deep learning methods have remarkable capacity to handle sparse data, and different kinds of models are designed for different scenarios. For example, Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) is powerful to handle temporal information while Convolutional Neural Network (CNN) (Lawrence et al. 1997) has remarkable ability to handle pictorial information. Stacked Denoising Autoencoders (SDAE) (Vincent et al. 2010) is suitable to be used for learning efficient representations from sparse data by reconstructing its inputs.

In this paper, we propose a reviews and contents based deep fusion model for cross-domain recommendation namely RC-DFM. Specifically, we first propose to extend the basic SDAE model to make it fit the feature learning scenario in cross-domain recommendation. Then several variants of SDAE are integrated to learn more effective user and item latent factors in both auxiliary and target domains. After obtaining the latent factors of the common users in both domains, a feature mapping between the two domains is learned by a mapping function. According to (Man et al. 2017), the MLP-based nonlinear mapping performs better than the linear mapping. Hence, we choose MLP to more effectively transfer user latent factors from the auxiliary domain to the target domain.

Our major contributions are summarized as follows:

- We integrate several extended SDAEs to handle rating matrices and the two kinds of side information in a deep fusion way, so that more effective latent factors of users and items with richer semantic information are learned.
- We propose a novel cross-domain recommendation framework called RC-DFM, which can make more satisfying recommendations for cold start users in cross-domain recommender systems.
- We systematically evaluate our proposal through comparing it with the state-of-the-art algorithms on the Amazon dataset. The results confirm that our method substantially improves the recommendation performance.

Preliminaries

Notations

$U = \{u_1, u_2, ..., u_m\}$ denotes the common users between the auxiliary domain $D^{(A)}$ and the target domain $D^{(T)}$. If a user only appears in one domain, we regard him/her as a **cold start user** in the other domain. Each domain has a set of items, denoted as $I^{(A)} = \{i_1^{(A)}, i_2^{(A)}, ..., i_a^{(A)}\}$ and $I^{(T)} = \{i_1^{(T)}, i_2^{(T)}, ..., i_t^{(T)}\}$, respectively. Each item in $I^{(A)}$ is associated with a piece of text to describe its main content, and the set of these item contents can be denoted as $P^{(A)} = \{p_1, p_2, ..., p_a\}$. Moreover, users can express their feelings about the items they ever bought via ratings and corresponding reviews. Based on this, we build the rating matrices of the two domains and represent them as R^{aux} and R^{tar}, respectively.

Problem Formulation and Framework

Given the rating matrices, review texts and item contents, our goal is to deeply fuse them to learn more effective latent factors for users and items, and then effective transfer these latent factors from the auxiliary domain to the target domain.

To achieve this goal, we propose a review and content based deep fusion model RC-DFM. As shown in Figure 1, RC-DFM contains four major steps: vectorization of reviews and item contents, generation of latent factors, mapping of user latent factors and cross-domain recommendation.

The first step is illustrated as the leftmost part of Figure 1. The raw texts include the review texts and item contents. We first separate all the reviews in each domain from the aspects of users and items respectively. That means, all the reviews a user ever wrote constitute the user's side information and all the reviews an item ever received constitute the item's side information. After text preprocessing, we adopt word-embedding and max-pooling to generate the corresponding structured vector representation of each group of reviews. The review vectors for each user are denoted as AUR_u and TUR_u in the auxiliary domain and the target domain respectively, where A represents auxiliary domain, T represents target domain, R denotes reviews and U denotes users. Similarly, the review vectors for each item in the two domains are represented as $AIR\_i^{(A)}$ and $TIR\_i^{(T)}$, where I represents items. We generate the vector representation for each item content in auxiliary domain in the same way, denoted as $AIP\_i^{(A)}$, where P denotes plots.

In the second step, we first denote the rating vectors of each user and item in the two domains as $AIS\_i^{(A)}$, $TIS\_i^{(T)}$, AUS_u and TUS_u respectively, where S denotes scores. Then we utilize four SDAEs, i.e. AI_SDAE, AU_SDAE, TI_SDAE and TU_SDAE, to learn latent factors of users and items in both domains. Specifically, we take the rating vector of each user/item as the input of an SDAE, and inject the corresponding review vector to each layer. The output of the middle layer is regarded as the latent factor of a user or an item. Meanwhile, in auxiliary domain, we employ an extra AP_SDAE to handle the content vectors and construct a mapping relationship between the output of its middle layer $h_i^{(A)}$ and the corresponding item latent factor $V_i^{(A)}$. In this way, we can deeply fuse item contents, reviews and ratings. Thus, the learned latent factors not only contain semantic information but also reflect user preference and item property. In the third step, an MLP network is used to learn the cross-domain feature mapping between the common users in the two domains. Finally, we can make recommendations for cold start users based on the user latent factors and the learned mapping function.

Deep Fusion Model for Cold Start Users in Cross-Domain Recommendation

Vectorization of Reviews and Contents

Before using reviews and contents, we first need to transform them into structured vector representations. In general, there are two kinds of methods to represent text. The first

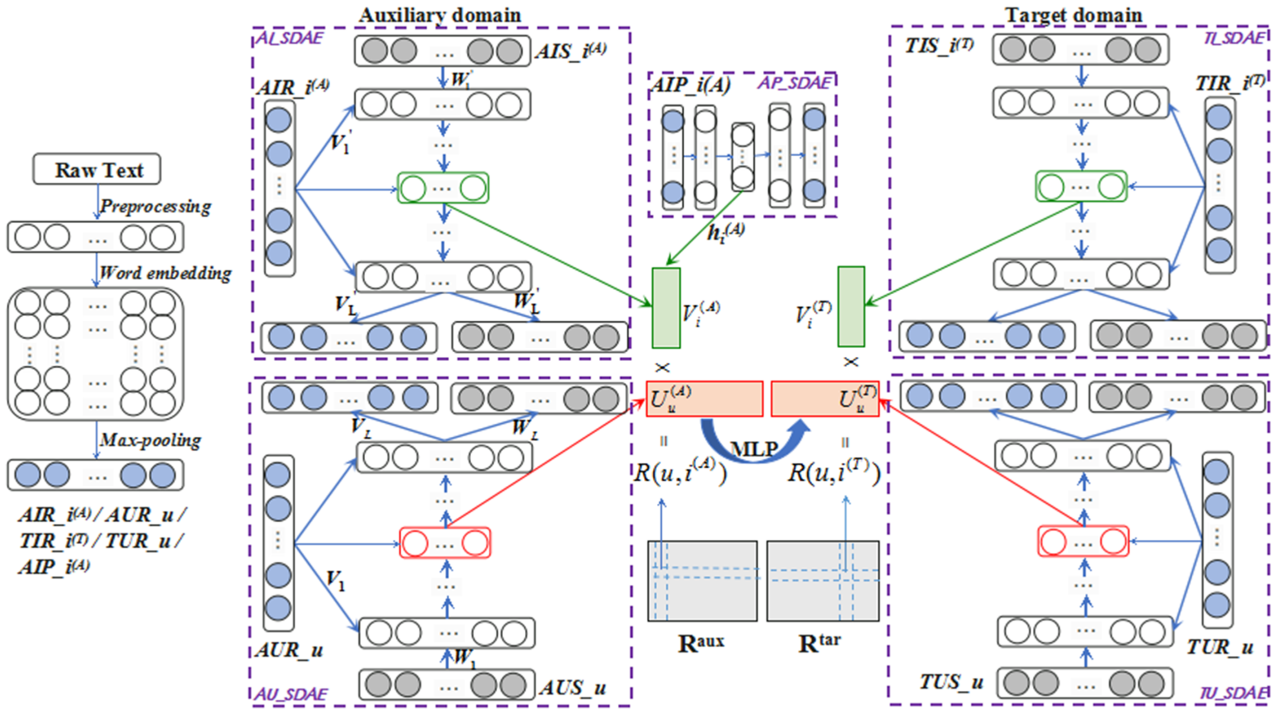

Figure 1: Framework of the proposed RC-DFM model for cross-domain recommendation.

method is one-hot representation. It tries to utilize an extremely sparse and possibly long vector to represent a word, which contains a 1 and many 0s. The second method is distributed representation. Compared with one-hot representation, this kind of method can keep the semantic information implicit in text (Zheng, Noroozi, and Yu 2017). In this paper, we employ word embedding, which is the representative method of distributed representation, to represent the text of reviews and item contents. A word embedding f tries to map a word into a n-dimensional distributed vector.

Taking all the reviews written by user u in auxiliary domain as an example, similar to study (Zheng, Noroozi, and Yu 2017), we first merge these reviews to a single document $d_{1:n}^{au}$ with n words. Then we utilize word embeddings that are pre-trained on more than 100 billion words from Google News[1] (Mikolov et al. 2013) to obtain the corresponding 300-dimensional word vector for each word $w_k^{au}, k \in \{1, 2, ..., n\}$ in this document. Then we concatenate all the word vectors into a matrix $M_{1:n}^{au}$:

$$M_{1:n}^{au} = w_1^{au} * w_2^{au} * ... * w_n^{au}$$

where $*$ denotes the concatenation operator. Next, we apply a max pooling operation over the matrix and take the maximum value of each column as the feature value of this dimension. Through this way, we can get a 300-dimensional vector for the group of reviews of user u in auxiliary domain, denoted as AUR_u. It can be regarded as the feature vector extracted from the group of reviews written by user u.

Similarly, we can obtain the feature vectors from the corresponding reviews for each item in auxiliary domain, each user in target domain and each item in target domain, denoted as $AIR\_i^{(A)}$, TUR_u and $TIR\_i^{(T)}$ respectively. For the contents of each item in auxiliary domain, we can get its corresponding feature vector in the same way, denoted as $AIP\_i^{(A)}$. All the representation vectors and rating matrices of the two domains constitute the inputs of RC-DFM.

[1]https://code.google.com/archive/p/word2vec/.

Generation of Latent Factors

Based on the extracted feature vectors discussed above, we can learn latent factors for users and items by integrating several extended SDAEs. Next we will introduce the basic SDAE model first and then introduce the extended model.

DAE is a feedforward neural network for reconstructing the input from its corrupted version with the motivation of learning a more robust mapping function. If we stack several DAEs together, we can get a deep network, namely SDAE, which is derived from the idea that multiple layers lead to higher level representations (Vincent et al. 2010). SDAE tries to make the output of the middle layer close to the latent factor as much as possible. It has been proven to be efficient in specific scenario (Huo et al. 2017).

Dong (Dong et al. 2017) proposed a variant of SDAE called aSDAE, which can integrate the one-hot representation of the basic description such as genre into the learned latent factors efficiently. Figure 2 shows the general architecture of aSDAE. There are L layers. The former $L/2$ layers act as an encoder and the latter $L/2$ layers act as a decoder. The original inputs of aSDAE are the rating vector of a user or an item and the one-hot representation of the basic description, denoted as x and s respectively. $h_0 = \tilde{x}$ and $\tilde{s}$

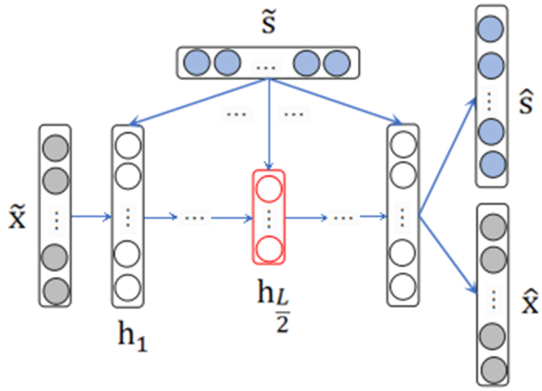

Figure 2: The architecture of aSDAE.

denote their corrupted versions. The final outputs of aSDAE are the reconstructions of x and s, denoted as $\hat{x}$ and $\hat{s}$.

For each hidden layer $l \in \{1, 2, ..., L-1\}$, the hidden representation h_l can be computed by:

$$h_l = g\left(W_l h_{l-1} + V_l \tilde{s} + b_l\right),$$

where W and V denote the weight matrices and b represents the bias vector. $g(\cdot)$ is the activation function. For the output layer L, the outputs can be calculated by:

$$\hat{x} = f\left(W_L h_L + b_{\hat{x}}\right)$$
$$\hat{s} = f\left(V_L h_L + b_{\hat{s}}\right)$$

where $f(\cdot)$ is the activation function.

Inspired by the way aSDAE handling side information, we come up with a method to extend and integrate several SDAEs to effectively fuse review texts and item contents with rating matrix. The details are as follows.

In auxiliary domain, given the rating matrix R^{aux}, we first take each column of the matrix as the rating vector of each item, denoted as $AIS\_i^{(A)}$. Similarly, we take each row of the matrix as the rating vector of each user u, denoted as AUS_u. These rating vectors as well as the feature vectors $AIR\_i^{(A)}$ and AUR_u extracted in last section constitute the inputs of the auxiliary domain. We simultaneously train three different kinds of SDAEs, i.e. AI_SDAE, AU_SDAE and AP_SDAE illustrated in Figure 1, to learn the latent factors of users and items ($U_u^{(A)}$ and $V_i^{(A)}$). In AI_SDAE, we take $AIS\_i^{(A)}$ of item $i^{(A)}$ as input and inject its feature vector $AIR\_i^{(A)}$ to each layer of the network. Then we try to reconstruct the input vectors in the last layer. The output of the middle layer is regarded as the item's latent factor. Similarly, in AU_SDAE, we take AUS_u as input and inject its feature vector AUR_u to each layer. The output of the middle layer is regarded as the user's latent factor. In AP_SDAE, we try to reconstruct $AIP\_i^{(A)}$ in the last layer and build a mapping relationship between the output of its middle layer $h_i^{(A)}$ and the corresponding item latent factor $V_i^{(A)}$. In this way, the learned item latent factors will reflect not only the inherent features of items (extracted from item contents) but also the item features the user prefer (extracted from the reviews that these items ever received).

The objective function in auxiliary domain is:

$$\begin{aligned} L_a = & \sum_{u,i^{(A)}} I_{u,i^{(A)}} \left(R_{u,i^{(A)}} - U_u^{(A)} V_i^{(A)}\right) \\ & + \alpha \sum_u \left(AUS\_u - \widehat{AUS\_u}\right)^2 \\ & + (1-\alpha) \sum_u \left(AUR\_u - \widehat{AUR\_u}\right)^2 \\ & + \beta \sum_{i^{(A)}} \left(AIS\_i^{(A)} - \widehat{AIS\_i^{(A)}}\right)^2 \\ & + (1-\beta) \sum_{i^{(A)}} \left(AIR\_i^{(A)} - \widehat{AIR\_i^{(A)}}\right)^2 \\ & + \gamma \sum_p \left(AIP\_i^{(A)} - \widehat{AIP\_i^{(A)}}\right)^2 \\ & + \mu \sum_p \left(h_i^{(A)} - V_i^{(A)}\right)^2 + \lambda \cdot f_{reg}, \end{aligned} \tag{1}$$

where I is the indicator matrix. α and β are weight parameters used to balance the weight between the reconstruction errors of review vectors and rating vectors. γ is a weight parameter and μ is the mapping coefficient between latent feature of item content and the corresponding item latent factor. λ is the regularization parameter and f_{reg} is the regularization term, which can be calculated as:

$$\begin{aligned} & f_{reg} = \\ & \sum_u ||U_u^{(A)}||_F^2 + \sum_i ||V_i^{(A)}||_F^2 + \sum_l \left(||W_l^P||_F^2 + ||b_l^P||_F^2\right) + \\ & \sum_l \left(||W_l||_F^2 + ||V_l||_F^2 + ||b_l||_F^2 + ||W_l'||_F^2 + ||V_l'||_F^2 + ||b_l'||_F^2\right) \end{aligned} \tag{2}$$

where W_l, V_l, W_l', V_l' and W_l^P are the weight matrices for AI_SDAE, AU_SDAE and AP_SDAE at layer l. b_l, b_l' and b_l^P are the corresponding bias vectors.

In target domain, we first get $TIS\_i^{(T)}$ and TUS_u for each item $i^{(T)}$ and each user u from R^{tar} in a similar way. Then, based on the rating vectors as well as the extracted feature vectors $TIR\_i^{(T)}$ and TUR_u, we simultaneously train two different SDAEs, i.e. TI_SDAE and TU_SDAE, to learn user and item latent factors $U_u^{(T)}$ and $V_i^{(T)}$. Unlike auxiliary domain, we do not utilize item contents in target domain to reflect real application scenario. Thus we assume the item content information in target domain is much sparser than that in auxiliary domain. The objective function in target domain is:

$$\begin{aligned} L_t = & \sum_{u,i^{(T)}} I_{u,i^{(T)}} \left(R_{u,i^{(T)}} - U_u^{(T)} V_i^{(T)}\right) \\ & + \alpha \sum_u \left(TUS\_u - \widehat{TUS\_u}\right)^2 \\ & + (1-\alpha) \sum_u \left(TUR\_u - \widehat{TUR\_u}\right)^2 \\ & + \beta \sum_{i^{(T)}} \left(TIS\_i^{(T)} - \widehat{TIS\_i^{(T)}}\right)^2 \\ & + (1-\beta) \sum_{i^{(T)}} \left(TIR\_i^{(T)} - \widehat{TIR\_i^{(T)}}\right)^2 + \lambda \cdot f'_{reg}. \end{aligned} \tag{3}$$

Note that the meanings of parameters in formula (3) are the same as that in formula (2). f'_{reg} is similar to f_{reg}, only without the third term related to AP_SDAE.

Nonlinear Mapping Based on MLP

Similar to study (Man et al. 2017), we employ MLP to perform latent space matching from auxiliary domain to target domain. We take the user latent factors in auxiliary domain as input and the user latent factors in target domain as output to learn a nonlinear mapping function. Back propagation is applied to optimize the parameters of MLP, and the optimization problem can be formalized as:

$$\min_{\theta} \sum_{u \in U} L\left(f_{mlp}\left(U_u^{(A)}; \theta\right), U_u^{(T)}\right), \tag{4}$$

where $f_{mlp}(\cdot)$ is the MLP mapping function and θ denotes the parameter set, including the weight matrices and biases.

Cross-Domain Recommendation

In this paper, we assume that the cold start users have sparse ratings and reviews in auxiliary domain, but no feedback at all in target domain. After learning the latent factors in auxiliary domain $U^{(A)}$, we can get the corresponding affine latent factors $\widehat{U^{(T)}}$ based on the nonlinear MLP mapping function:

$$\widehat{U^{(T)}} = f_{mlp}\left(U^{(A)}; \theta\right). \tag{5}$$

Based on the learned $\widehat{U^{(T)}}$ and $V^{(T)}$, we can make predictions for cold start users.

Experiments

In this section, we will perform extensive experiments to demonstrate the effectiveness of RC-DFM by comparing it with state-of-the-art methods.

Experimental Settings

Datasets. We use the Amazon dataset[2] (McAuley, Pandey, and Leskovec 2015) to evaluate the performance of our model and baselines. This dataset have ratings and metadata (including titles of movies) from the Amazon website[3] and there are 142.8 million reviews in total spanning from May 1996 to July 2014. It has 21 categories of items and we choose the three most widely used categories in cross-domain recommendation to perform the experiment. The details of our dataset are illustrated in Table 1. We first select the items that have more than 120 reviews in "Books" or more than 30 reviews in "Music CDs". Then, in each pair of domains, we select the users who give more than 10 feedbacks in both domains. Since there is no item contents in this dataset, we utilize the titles of movies in metadata to match the corresponding plots from IMDb[4]. In reality, we cannot match all plots successfully. Thus, for those movies, we apply word-embedding on their titles directly rather than plots.

[2]http://jmcauley.ucsd.edu/data/amazon/.

[3]https://www.amazon.com/.

[4]https://www.imdb.com/.

Table 1: The Statistics of the datasets

	dataset 1		dataset 2	
Domain	Movies	Books	Movies	Music CDs
Users	792		1,464	
Items	3,550	2,271	3,605	4,611
Ratings	51,422	20,365	78,431	59,299
Density	1.83%	1.13%	1.49%	0.88%

Experiment Setup. According to (Cantador et al. 2015), there are different scenarios of data overlap between user and item sets in two domains. In this paper, we evaluate the performance of RC-DFM in the scenario that there are user overlaps between two domains. We randomly select a certain proportion of users and remove all their feedbacks in target domain so that we can regard them as cold start users. In our experiments, we set the proportions of cold start users as 80%, 50% and 20% of the initial users respectively. The proportion is denoted as ϕ. The dimensions of latent factors of users, denoted as K, are set as 10, 20, and 30 respectively. The number of layers for each neural network is set to 4. We use grid search to find the best settings of parameters α, β, γ and μ as 0.5, 0.8, 0.7 and 0.2 respectively. The regularization parameter λ is set to 0.01. The learning rate is set to 0.0001. We utilize 5-fold cross validation and report the average results.

Evaluation Metric. In our experiments, we adopt Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) defined as follows as the evaluation metrics.

$$RMSE = \sqrt{\sum_{r_{ui} \in \mathcal{O}_{test}} \frac{(\hat{r}_{ui} - r_{ui})^2}{|\mathcal{O}_{test}|}}$$

$$MAE = \frac{1}{|\mathcal{O}_{test}|} \sum_{r_{ui} \in \mathcal{O}_{test}} |\hat{r}_{ui} - r_{ui}|,$$

where $\mathcal{O}_{test}$ is the set of test ratings. r_{ui} denotes an observed rating in $\mathcal{O}_{test}$. $\hat{r}_{ui}$ represents the predictive value of r_{ui}. $|\mathcal{O}_{test}|$ is the number of test ratings.

Baselines. We choose the following methods as baselines for comparison.

- **PMF**: Probabilistic Matrix Factorization (PMF) is introduced in (Salakhutdinov and Mnih 2007), which models latent factors of users and items by Gaussian distributions.
- **CMF**: Collective Matrix Factorization (CMF) (Singh and Gordon 2008) tends to incorporate different sources of information by simultaneously factorizing multiple matrices. The latent factors of entities are shared between the auxiliary domain and the target domain.
- **EMCDR**: This model is proposed in (Man et al. 2017). It adopts matrix factorization to learn latent factors first and then utilize an MLP network to map the user latent factors from the auxiliary domain to the target domain.
- **DFM**: This is the cut version of RC-DFM, which merely takes ratings into consideration and does not utilize side information.

- **R-DFM**: This is also the cut version of RC-DFM. It takes review texts as side information and neglects item contents.

Performance Evaluation

Recommendation Performance. The experimental results of RMSE and MAE on "Movies & Books" are shown in Table 2, and the results on "Movies & Music CDs" are presented in Table 3. The best performance of these models is shown in boldface. In each pair of domains, we regard "Movies" as the auxiliary domain and the other domain as the target domain.

We evaluate the performance of different models under different values of ϕ and K in terms of RMSE and MAE respectively. From Tables 2 and 3, one can draw the conclusion that RC-DFM is superior to all the state-of-the-art methods in cross-domain recommendation for cold start users. Compared with EMCDR, when ϕ is 20%, RC-DFM improves the performance by 3% to 6% both in terms of RMSE and MAE. When ϕ gets larger, the improvement of performance gets greater too. As for PMF, the performance drops remarkably as ϕ gets larger, while all the other cross-domain methods still provide satisfying results to some extent. This is because we try to make more accurate recommendations for cold start users who have no feedbacks at all in target domain, while PMF is a single domain recommendation algorithm and its performance is totally dependent on the richness of users' history information. One can also observe that CMF needs to select a suitable K (20) to get a desirable result while other models can get similar results under different situations, especially RC-DFM. It demonstrates the robustness of RC-DFM. In addition, DFM and EMCDR all perform better than CMF which proves the effectiveness of deep learning methods. Besides, DFM achieves more superior results than EMCDR in the vast majority of cases, which demonstrates that the way we handle rating vectors in cross-domain recommendation is more suitable than factorizing rating matrices directly. In addition, R-DFM performs better than DFM, and RC-DFM performs better than R-DFM, which prove the effectiveness of utilizing review texts and item contents. We successfully fuse these side information with rating matrices. One can also observe that the results of all these models in "Movies & Music CDs" are generally worse than that in "Movies & Books" due to that the density of data in "Movies & Music CDs" is much lower.

Impact of Side Information. Apart from the overall performance compared with the baselines, we also study the sensitivity of the algorithm on the richness of side information. Since we cannot match all the plots of movies from IMDB successfully, we need to verify whether the performance will be better if richer item contents are available. Therefore, we select the movies which have complete plots (item contents), and denote our framework in this scenario as RC-DFM-Full. We compare the performance of RC-DFM-Full with DFM, R-DFM, and RC-DFM, and the results are illustrated in Figures 3 and 4. One can observe that RC-DFM-Full performs better than the three incomplete versions of our model. Along with the side information

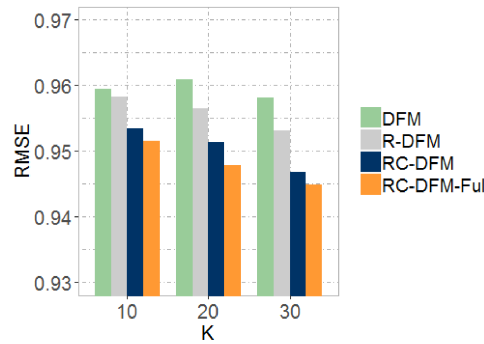

Figure 3: The performance on "Movies & Books"

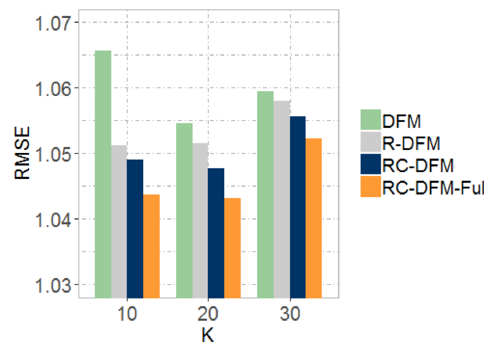

Figure 4: The performance on "Movies & Music CDs"

getting richer, that means, from DFM to RC-DFM-Full, the model performance gets better. It demonstrates the effectiveness of our method in fusing review texts and item contents. It demonstrates that richer side information leads to better performance.

Related Work

Existing works about cross-domain recommendation mostly extracted valuable information from rating matrices. Zhu (Zhu et al. 2018) proposed a Deep framework called DCDCSR for both Cross-Domain and Cross-System Recommendations. It took into account the rating sparsity degrees of individual users and items in different domains or systems. Ren (Ren et al. 2015) proposed a novel Probabilistic Cluster-level Latent Factor (PCLF) model to learn the common rating pattern shared across domains, as well as capture the domain-specific rating patterns of users and clustering of items in each domain. The performance of this kind of works is largely limited by the inherent sparsity of rating matrices.

Valuable side information has been proven to be very useful in cross-domain recommendation tasks. Song (Song et al. 2017) proposed a joint tensor factorization model to fully exploit the aspect factors extracted from reviews. Fang (Fang et al. 2015) proposed a novel way to exploit rating patterns across multiple domains by transferring the tag co-occurrence matrix information, which could be used for revealing common user pattern. All these works merely consider one kind of side information and cannot effectively and

Table 2: Recommendation performance on “Movies & Books”

		RMSE						MAE					
K	ϕ	PMF	CMF	EMCDR	DFM	R-DFM	RC-DFM	PMF	CMF	EMCDR	DFM	R-DFM	RC-DFM
10	80%	4.0892	1.5381	1.5276	0.9987	0.9893	**0.9770**	3.8438	1.2809	1.2246	0.7997	0.7958	**0.7902**
	50%	1.6204	1.5005	1.1174	0.9820	0.9719	**0.9648**	1.2326	1.2282	0.8538	0.7939	0.7865	**0.7808**
	20%	1.3839	1.4653	1.0482	0.9594	0.9583	**0.9535**	1.0611	1.1914	0.8246	0.7756	0.7719	**0.7694**
20	80%	4.1113	1.2624	1.5500	0.9990	0.9790	**0.9744**	3.8986	0.9884	1.2777	0.7996	0.7898	**0.7886**
	50%	1.7428	1.2470	1.1648	0.9721	0.9700	**0.9632**	1.3770	0.9625	0.8984	0.7859	0.7844	**0.7774**
	20%	1.4198	1.2146	1.0305	0.9609	0.9565	**0.9514**	1.1098	0.9432	0.7824	0.7758	0.7704	**0.7667**
30	80%	4.1103	1.5437	1.6676	0.9838	0.9807	**0.9721**	3.9034	1.2050	1.3633	0.7938	0.7904	**0.7858**
	50%	1.7933	1.3455	1.1823	0.9682	0.9646	**0.9591**	1.4537	1.0441	0.9413	0.7816	0.7752	**0.7671**
	20%	1.4835	1.2760	1.0548	0.9581	0.9532	**0.9468**	1.1892	0.9917	0.8176	0.7706	0.7645	**0.7590**

Table 3: Recommendation performance on “Movies & Music CDs”

		RMSE						MAE					
K	ϕ	PMF	CMF	EMCDR	DFM	R-DFM	RC-DFM	PMF	CMF	EMCDR	DFM	R-DFM	RC-DFM
10	80%	4.1154	1.6483	1.6757	1.0976	1.0726	**1.0706**	3.8374	1.3804	1.3660	0.8824	0.8611	**0.8589**
	50%	1.6585	1.6456	1.1887	1.0741	1.0637	**1.0578**	1.2419	1.3658	0.9289	0.8666	0.8564	**0.8461**
	20%	1.4015	1.6318	1.0857	1.0657	1.0513	**1.0490**	1.0466	1.3512	0.8464	0.8483	0.8360	**0.8005**
20	80%	4.1353	1.3287	1.7407	1.0917	1.0724	**1.0698**	3.8760	1.0275	1.4479	0.8773	0.8603	**0.8573**
	50%	1.6942	1.3269	1.2273	1.0719	1.0553	**1.0507**	1.3122	1.0081	0.9593	0.8653	0.8431	**0.8410**
	20%	1.4624	1.3039	1.0802	1.0546	1.0516	**1.0477**	1.1460	0.9898	0.8308	0.8267	0.8218	**0.7782**
30	80%	4.1894	1.5249	1.6540	1.0880	1.0720	**1.0661**	3.9655	1.1787	1.3451	0.8739	0.8591	**0.8522**
	50%	1.7951	1.3944	1.2093	1.0713	1.0637	**1.0586**	1.4315	1.0749	0.9456	0.8669	0.8532	**0.8483**
	20%	1.5190	1.3505	1.0852	1.0594	1.0580	**1.0557**	1.2106	1.0323	0.8190	0.8328	0.7836	**0.7778**

deeply fuse different side information with ratings.

Deep learning is a powerful way to learn effective representations from sparse data. Thus recently researchers try to apply it on recommender systems to handle sparse rating matrices and side information to extract better latent factors. Li (Li, Kawale, and Fu 2015) proposed a general deep architecture named DCF to combine probabilistic matrix factorization with marginalized denoising stacked auto-encoders, which is applied on rating matrix and basic description of users and items. Wang (Wang, Wang, and Yeung 2015) proposed a hierarchical Bayesian model called CDL to jointly perform deep representation learning for content information and collaborative filtering for rating matrix. However, all these works focused on single domain recommendation.

Conclusion

In this paper, we proposed a review and content based deep fusion model named RC-DFM to make recommendations for cold start users in cross-domain recommendation scenario. In this model, we took review text and item content as side information and integrated several extended versions of SDAE to effectively fuse side information with rating matrix in both auxiliary and target domains. Through this way, the learned user and item latent factors can preserve more semantic information. Finally, we employed MLP to cope with the latent space matching problem so that we could make predictions for cold start users in the target domain. The experimental results showed the superiority of our model.

Acknowledgments

This work was supported by 973 Program (No. 2015CB352502), the Key Project of NSF of China (No. 61332015), NSF of China (No. 61602237, No. 61802231), NSF of Shandong Province, China (No. ZR2017MF065) and NSF of Jiangsu Province, China (No. BK20171420).

References

Cantador, I.; Fernández-Tobías, I.; Berkovsky, S.; and Cremonesi, P. 2015. Cross-domain recommender systems. In Ricci, F.; Rokach, L.; and Shapira, B., eds., *Recommender Systems Handbook*. Springer. 919–959.

Dong, X.; Yu, L.; Wu, Z.; Sun, Y.; Yuan, L.; and Zhang, F. 2017. A hybrid collaborative filtering model with deep structure for recommender systems. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 1309–1315.

Fang, Z.; Gao, S.; Li, B.; Li, J.; and Liao, J. 2015. Cross-domain recommendation via tag matrix transfer. In *IEEE International Conference on Data Mining Workshop, ICDMW 2015, Atlantic City, NJ, USA, November 14-17, 2015*, 1235–1240.

Guan, L.; Alam, M. H.; Ryu, W.; and Lee, S. 2016. A phrase-based model to discover hidden factors and hidden topics in recommender systems. In *2016 International Conference on Big Data and Smart Computing, BigComp 2016, Hong Kong, China, January 18-20, 2016*, 337–340.

Han, X.; Shi, C.; Wang, S.; Yu, P. S.; and Song, L. 2018. Aspect-level deep collaborative filtering via heterogeneous information networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, 3393–3399.

Hao, P.; Zhang, G.; and Lu, J. 2016. Enhancing cross domain recommendation with domain dependent tags. In *2016 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2016, Vancouver, BC, Canada, July 24-29, 2016*, 1266–1273.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Huo, H.; Liu, X.; Zheng, D.; Wu, Z.; Yu, S.; and Liu, L. 2017. Collaborative filtering fusing label features based on SDAE. In *Advances in Data Mining. Applications and Theoretical Aspects - 17th Industrial Conference, ICDM 2017, New York, NY, USA, July 12-13, 2017, Proceedings*, 223–236.

Lawrence, S.; Giles, C. L.; Tsoi, A. C.; and Back, A. D. 1997. Face recognition: a convolutional neural-network approach. *IEEE Trans. Neural Networks* 8(1):98–113.

Li, S.; Kawale, J.; and Fu, Y. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, 811–820.

Man, T.; Shen, H.; Jin, X.; and Cheng, X. 2017. Cross-domain recommendation: An embedding and mapping approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 2464–2470.

McAuley, J. J.; Pandey, R.; and Leskovec, J. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, 785–794.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, 3111–3119.

Park, S.; Kim, Y.; and Choi, S. 2013. Hierarchical bayesian matrix factorization with side information. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 1593–1599.

Ren, S.; Gao, S.; Liao, J.; and Guo, J. 2015. Improving cross-domain recommendation through probabilistic cluster-level latent factor model. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 4200–4201.

Salakhutdinov, R., and Mnih, A. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, 1257–1264.

Singh, A. P., and Gordon, G. J. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, 650–658.

Song, T.; Peng, Z.; Wang, S.; Fu, W.; Hong, X.; and Yu, P. S. 2017. Review-based cross-domain recommendation through joint tensor factorization. In *Database Systems for Advanced Applications - 22nd International Conference, DASFAA 2017, Suzhou, China, March 27-30, 2017, Proceedings, Part I*, 525–540.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11:3371–3408.

Wang, X.; Peng, Z.; Wang, S.; Yu, P. S.; Fu, W.; and Hong, X. 2018. Cross-domain recommendation for cold-start users via neighborhood based feature mapping. In *Database Systems for Advanced Applications - 23rd International Conference, DASFAA 2018, Gold Coast, QLD, Australia, May 21-24, 2018, Proceedings, Part I*, 158–165.

Wang, H.; Wang, N.; and Yeung, D. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, 1235–1244.

Xin, X.; Liu, Z.; Lin, C.; Huang, H.; Wei, X.; and Guo, P. 2015. Cross-domain collaborative filtering with review text. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, 1827–1834.

Zhang, H.; Li, Z.; Chen, Y.; Zhang, X.; and Wang, S. 2014. Exploit latent dirichlet allocation for one-class collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, 1991–1994.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, 425–434.

Zhu, F.; Wang, Y.; Chen, C.; Liu, G.; Orgun, M. A.; and Wu, J. 2018. A deep framework for cross-domain and cross-system recommendations. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, 3711–3717.