



# SARFM: A Sentiment-Aware Review Feature Mapping Approach for Cross-Domain Recommendation

Yang Xu, Zhaohui Peng, Yupeng Hu, and Xiaoguang Hong<sup>(✉)</sup>

School of Computer Science and Technology, Shandong University, Jinan,  
People's Republic of China  
{xuyang0211, pzh, huyupeng, hxg}@sdu.edu.cn

**Abstract.** Cross-domain algorithms which aim to transfer knowledge available in the source domains to the target domain are gradually becoming more attractive as an effective approach to help improve quality of recommendations and to alleviate the problems of cold-start and data sparsity in recommendation systems. However, existing works on cross-domain algorithm mostly consider ratings, tags and the text information like reviews, and don't take advantage of the sentiments implicated in the reviews efficiently, especially the negative sentiment information which is easy to be weakened during the process of transferring. In this paper, we propose a sentiment-aware review feature mapping framework for cross-domain recommendation, called SARFM. The proposed SARFM framework applies deep learning algorithm SDAE (Stacked Denoising Autoencoders) to model the Sentiment-Aware Review Feature (SARF) of users, and transfers SARF via a multi-layer perceptron to capture the nonlinear mapping function across domains. We evaluate and compare our framework on a set of Amazon datasets. Extensive experiments on each cross-domain recommendation scenarios are conducted to prove the high accuracy of our proposed SARFM framework.

**Keywords:** Cross-domain recommendation · Sentiment-aware review feature  
Stacked denoising autoencoders

## 1 Introduction

Cross-domain recommendation systems are gradually becoming more attractive as a practical approach to improve quality of recommendations and to alleviate cold-start problem, especially in small and sparse datasets. These algorithms mine knowledge on users and items in a source domain to improve the quality of the recommendations in a target domain. They can also provide joint recommendations for items belonging to different domains by the linking information among these domains [1]. Most existing works about cross domain recommendation tend to aggregate knowledge from different domains from the perspective of explicitly specified common information [2–4] or transferring latent features [5, 8, 9, 13]. However, the aggregated knowledge are merely based on ratings, tags, or the text information like reviews, and don't take advantage of the sentiments implicated in the reviews efficiently. After watching a popular film,

using a novel electronic product or playing a video game, users often rate them and submit reviews to share their feelings, which could convey fairly rich sentiment information.

In this paper, we investigate two key issues of review-based cross-domain recommendation:

(1) *How to effectively transfer user's sentiment information?*

For all we know, existing cross-domain recommendation algorithms which utilize user reviews didn't take full advantage of the sentiment information of these reviews. They implement knowledge transfer by mixing positive and negative reviews together, which will weaken and even lose some sentiment information of the users, especially the negative sentiment. For instance, a user may deeply care about the plot of a novel, and he made positive comments on the plots of some novels in the domain of electronic book (source domain) while made negative comments on the plots of some other novels. If we transfer the knowledge gained from user reviews from the source domain to the target domain without distinguishing the sentiment polarity of these reviews, some latent factors such as "plot", "positive sentiment" and "negative sentiment" of the reviews will be mixed up as "users' feature" to be transferred to the target domain. In the domain of movie (target domain), a movie with poor plots, namely the movie whose latent factors "plot" and "negative sentiment" take higher weight, will produce a match with the users' feature transferred to the target domain. Nevertheless, the user may not be fond of this movie.

(2) *Which model should be leveraged to extract sentiment features of reviews, topic model or deep learning model?*

Digging the latent sentiment feature of user reviews is the key of improving the performance of recommendation algorithm. Review-based recommendation algorithm tend to employ the topic model such as LDA (Latent Dirichlet Allocation) [10] to extract latent features. Each latent feature (topic) extracted by LDA is associated with a set of key words. Thus, we can acquire the interpretable recommendation results through matching topic distributions of users and products. On the other hand, deep learning is adept in learning deep latent features automatically. The deep learning model SDAE (Stacked Denoising Autoencoder) [7] is capable of extracting the deep latent feature of review information and owns lower model complexity compared with the most widely used deep learning algorithms, CNN (Convolutional Neural Network) [22] and RNN (Recurrent Neural Network) [23]. However, the latent features extracted by SDAE are not interpretable, which means SDAE based algorithm is hard to provide a natural interpretation for the recommendation result.

In this paper, we give our answers to the above two questions and thus propose a new cross-domain recommendation framework called SARFM. Under SARFM, we can effectively identify the semantic orientation of user reviews and extract the sentiment-aware review feature (SARF). To achieve the goal of transferring knowledge, we propose a multi-layer perceptron based mapping method to transfer sentiment information of users from source domain to target domain. Through transferring SARF of users, the SARFM method gets a superior performance in cross-domain recommendation.

To summarize, the major contributions of this paper are as follows:

- We consider sentiment information in the cross-domain recommendation problem and transfer positive and negative sentiment-aware review feature from source domain to target domain respectively.
- A novel cross-domain recommendation framework was brought forward in this paper, together with multiple implementations of the sentiment-aware review feature extracting component in the framework.
- We systematically compare our methods with other algorithms on the Amazon dataset. The results confirm that our methods substantially improve the performance of cross-domain recommendation.

The rest of this paper is organized as follows. Section 2 reviews the related works and Sect. 3 presents some notations and the problem formulation. Section 4 introduces the modeling method of SARF. Section 5 details the mapping method of SARF and the cross-domain recommendation approach. Experiments and discussion are given in Sect. 6 and conclusions are drawn in Sect. 7.

## 2 Related Work

Existing works about cross-domain algorithm mostly extract domain-specific information from ratings [5], tags [2] and the text information like reviews [16]. Ren [8] proposed the PCLF model to learn the shared common rating pattern across multiple rating matrices and the domain-specific rating patterns from each domain. Fang [2] exploited the rating patterns across multiple domains by transferring the tag co-occurrence matrix information. Xin [16] exploited review text by learning a non-linear mapping on users' preferences on different topics across domains. However, these approaches don't take advantage of the sentiments implicated in the reviews efficiently.

Sentiment analysis is widely used in recommendation systems. Computing the semantic orientation of a user review has been studied by several researchers. Diao [17] built a language model component in their proposed JMARS model to capture aspects and sentiments hidden in reviews. Zhang [18] extracted explicit product features and user opinions by phrase-level sentiment analysis on user reviews to generate explainable recommendation results. Li [19] proposed a SUIT model to simultaneously utilize the textual topic and user item factors for sentiment analysis. But all these algorithms don't employ sentiment analysis on cross-domain recommendation task.

Deep learning is usually applied on image recognition and natural language processing. However, in recent years, the attempt to apply deep learning on recommendation system has widely emerged. Wang [25] proposed a hierarchical Bayesian model that uses a deep learning model to obtain content features and a traditional CF model to address the rating information. He [12] proposed a deep learning-based recommendation framework in which users and items are represented via the one-hot encoding of their ID. In this paper, we employ deep learning model SDAE to extract sentiment-aware review feature and mapping it to target domain for the sake of making cross-domain recommendations.

### 3 Preliminaries

In this section, we first introduce some frequently used notations. Then, we formulate the cold-start cross-domain recommendation problem and present the SARFM framework to solve the problem.

#### 3.1 Notations

Objects to be recommended in the cross-domain recommendation system are referred to as *items*. Let  $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$  denote the set of common users in both domains and  $\mathcal{I}_S = \{i_1, i_2, \dots, i_{|\mathcal{I}_S|}\}$ ,  $\mathcal{I}_T = \{t_1, t_2, \dots, t_{|\mathcal{I}_T|}\}$  are the sets of items (e.g. movies, books, or electronics) in source domain and in target domain respectively. The user review dataset is represented as  $R_{S,U} = \{r_{u_1}, r_{u_2}, \dots, r_{u_{|\mathcal{U}|}}\}$  in source domain and  $R_{T,U}$  in target domain, where  $r_{u_i}$  is all of the reviews of user  $u_i$  in the corresponding domain. Similarly, we let  $R_{T,I} = \{r_{i_1}, r_{i_2}, \dots, r_{i_{|\mathcal{I}_T|}}\}$  denote the item review dataset in target domain, where  $r_{i_j}$  is all of the reviews which item  $i_j$  acquired in target domain.

In the SARFM, the sentiment analysis algorithm is employed on the review datasets mentioned above to divide them into corresponding positive review datasets (e.g.  $R_{S,U}^{pos}$ ,  $R_{T,U}^{pos}$  and  $R_{T,I}^{pos}$ ) and negative review datasets (e.g.  $R_{S,U}^{neg}$ ,  $R_{T,U}^{neg}$  and  $R_{T,I}^{neg}$ ).  $F_{S,U}^{pos} = (f_{S,u_1}^{pos}, f_{S,u_2}^{pos}, \dots, f_{S,u_{|\mathcal{U}|}}^{pos})^T$  represents the positive review feature matrix of common users in source domain, while  $F_{T,U}^{neg} = (f_{T,u_1}^{neg}, f_{T,u_2}^{neg}, \dots, f_{T,u_{|\mathcal{U}|}}^{neg})^T$  denotes the negative review feature matrix of common users in the target domain, which is similar to  $F_{S,U}^{neg}$  and  $F_{T,U}^{pos}$ . In addition,  $F_I^{pos} = \{f_{t_1}^{pos}, f_{t_2}^{pos}, \dots, f_{t_{|\mathcal{I}_T|}}^{pos}\}$  denotes the positive review feature matrix of all items in target domain, while  $F_I^{neg} = \{f_{t_1}^{neg}, f_{t_2}^{neg}, \dots, f_{t_{|\mathcal{I}_T|}}^{neg}\}$  denotes the negative review feature matrix of all items in target domain. For a cold-start user  $u'$  in target domain,  $f_{S,u'}^{pos}$  ( $f_{S,u'}^{neg}$ ) denotes the positive (negative) review feature of user  $u'$  in source domain, and  $\hat{f}_{T,u'}^{pos}$  ( $\hat{f}_{T,u'}^{neg}$ ) represents the affine positive (negative) review feature of user  $u'$  in target domain.

#### 3.2 Problem Formulation

Given two domains which share the same users  $U$ . Users appearing in only one domain can be regarded as the cold-start users  $U'$  in the other domain. Without loss of generality, one domain is referred to as the source domain and the other as the target domain. The most common cross-domain recommendation approaches focus on transferring information based on ratings, tags and reviews from source domain to target domain, without accounting for sentiment information implicated the reviews sufficiently.

We are tackling cross-domain recommendation task for cold-start users by modeling the Sentiment-Aware Review Feature (SARF) of users and transferring them from source domain to target domain. To achieve this purpose, we propose a cross-domain recommendation framework called SARFM. This framework contains three major steps, i.e., sentiment-aware review feature extracting, cross-domain mapping and cross-domain recommendation, as illustrated in Fig. 1.

In the first step, we apply a SO-CAL [24] based sentiment analysis algorithm to analyze the Semantic Orientation (SO) of each sentence of user reviews in both domains. Then, the original review datasets of both domains are divided into corresponding positive review datasets and negative review datasets respectively. Next, we employ the deep learning model SDAE (Stacked Denoising Autoencoder) on the sentiment tagged datasets to extract the sentiment-aware review feature of users. In addition, we designed an LDA [10] based method to extract SARF of users, which is used to compare the effectiveness with SDAE. In the second step, we model the cross-domain relationships of users through a mapping function based on Multi-Layer Perceptron (MLP) [6]. We assume that there is an underlying mapping relationship between the user’s SARFs of the source and target domains, and further use a mapping function to capture this relationship. Finally, in the third step, we make recommendation for cold-start user in the target domain. We can get an affine SARF for cold-start user in the target domain, with the SARF learned for him/her in the source domain and the MLP-based mapping functions. In the rest of this paper, we will introduce each step of SARFM in details.

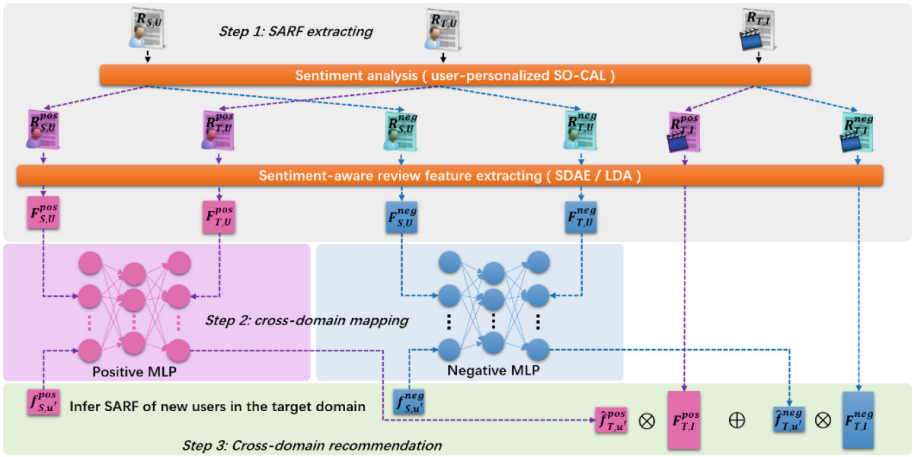


Fig. 1. Overview of the SARFM cross-domain recommendation framework

## 4 Sentiment-Aware Review Feature Extracting

As discussion in the previous section, in order to transfer sentiment-aware review pattern in the source domains to the target domain, the first phase of SARFM is to model the sentiment-aware review feature of common users in both domains. The key challenge is how to extract the user’s focus on the item and the positive or negative emotions expressing in the user reviews. To address this problem, we propose a sentiment-aware review feature extracting method based on sentence-level sentiment analysis approach and SDAE (or LDA).

### 4.1 Sentiment Analysis

The sentiment analysis problem in SARFM can be formulated as follows: Given a set of reviews  $R$ , a sentiment classification algorithm classifies each sentence of review  $r \in R$  into one of the two sets, positive set  $R^{pos}$  and negative set  $R^{neg}$ . Each entry of  $R^{pos}$  and  $R^{neg}$  is an (user/item, {sentence}, SO value) triplet, where {sentence} is a set of sentences with clear semantic orientation (“positive” in  $R^{pos}$  or “negative” in  $R^{neg}$ ) extracted from the raw review data of the user/item, and SO value is the average of semantic orientation values of sentences in {sentence}. For this purpose, we analyze the semantic orientation of each sentence of user reviews based on the state-of-the-art sentiment analysis algorithm SO-CAL described in [24].

The Semantic Orientation CALculator (SO-CAL) is a lexicon-based sentiment analysis approach which mainly consists of three characteristics. First, SO-CAL uses four kinds of manual basic dictionary (adjective, noun, verb, and adverb), which were produced by hand-tagging each of the words on a scale ranging from  $-5$  for extremely negative to  $+5$  for extremely positive. Second, in SO-CAL, intensification is modeled using modifiers, with each intensifying word having a percentage associated with itself, the amplifiers are positive, while the downtoners are negative. Finally, SO-CAL applies a polarity shift method of negation, instead of changing the sign, the SO value is shifted toward the opposite polarity by a fixed amount (such as 4).

In this paper, we improve the SO-CAL to a user-personalized sentiment analysis algorithm via considering a user sentiment bias. Given that a user rating can reflect the overall evaluation and the sentiment of the user on an item, we utilize user’s rating to calculate user sentiment bias to modify the semantic orientation values of dictionary words when calculating the semantic orientation of sentences in the review. Formally, given a piece of review  $r$  of user  $u$ , user-personalized semantic orientation value of dictionary words is defined as follows:

$$SO_u(w) = \begin{cases} (1 + b_u) \cdot SO(w) & \text{if } SO(w) > 0 \\ (1 - b_u) \cdot SO(w) & \text{if } SO(w) < 0 \end{cases}, b_{u,r} = \frac{rat_{u,r} - \overline{rat}_u}{\overline{rat}_u} \quad (1)$$

where  $SO_u(w)$  is user-personalized semantic orientation value of dictionary word  $w$  for user  $u$  and  $SO(w)$  represents the initial semantic orientation value of  $w$ .  $b_{u,r}$  denotes the user sentiment bias for review  $r$ ,  $rat_{u,r}$  is the corresponding rating of review  $r$  and the average rating of user  $u$  is represented as  $\overline{rat}_u$ .

Since the SO calculation procedure in SO-CAL is not the focus of this paper, we refer the readers to the related literature [24] for more details. The effect of sentiment analysis on SARFM will be reported later in the experimental section.

We employ user-personalized SO-CAL method on original review sets  $R_{S,U}, R_{T,U}, R_{T,I}$  to divide them into positive review subsets  $R_{S,U}^{pos}, R_{T,U}^{pos}, R_{T,I}^{pos}$  and negative review subsets  $R_{S,U}^{neg}, R_{T,U}^{neg}, R_{T,I}^{neg}$  on the sentence-level respectively.

## 4.2 Sentiment-Aware Review Feature Extracting

Sentiment-Aware Review Feature (SARF) indicates the user’s focus on the item and the positive or negative emotions expressed in the reviews. In this paper, we apply the deep learning model SDAE to extract SARF. In addition, the LDA based extracting method is also described in this paper which is designed as another implementation of our recommendation framework.

**Stacked Denoising Autoencoder.** DAE, namely denoising autoencoder, which is shown in Fig. 2 (a), consists of an encoder and a decoder. The initial input  $x$  is converted into its corrupted version  $x'$  by means of a stochastic mapping  $x' \sim qD(x|x)$ . The encoder  $e(\cdot)$  takes the given corrupted input  $x'$  and maps it to its hidden representation  $e(x')$ , while the decoder  $d(\cdot)$  maps this hidden representation back to a reconstructed version of  $d(e(x'))$ . Then, a denoising autoencoder is trained to reconstruct the original input  $x$  by minimizing loss  $\mathcal{L}(x, d(e(x')))$ . Nevertheless, existing literatures have shown that multiple layers stacked together can generate rich representations in hidden layers [20, 21]. SDAE [7] is a feedforward neural network which stacks multiple DAEs in order to form a deep learning framework, as shown in Fig. 2 (b). Each layer’s output in SDAE is the input of its next layer. SDAE utilizes greedy layer-wise training strategy to train each layer in the network successively, and then pre-train the whole deep learning network.  $Z_c$  is the clean input matrix and  $Z_o$  represents the noise-corrupted input matrix.  $Z_L$  is the middle layer of the model. Let  $\lambda$  represents regular parameter and  $\|\cdot\|_F$  denotes Frobenius norm. Then the training model of SDAE is formulated as follow:

$$\min_{\{w_l\}, \{b_l\}} \|Z_c - Z_L\|_F^2 + \lambda \sum_l (\|w_l\|_F^2 + \|b_l\|_F^2) \quad (2)$$

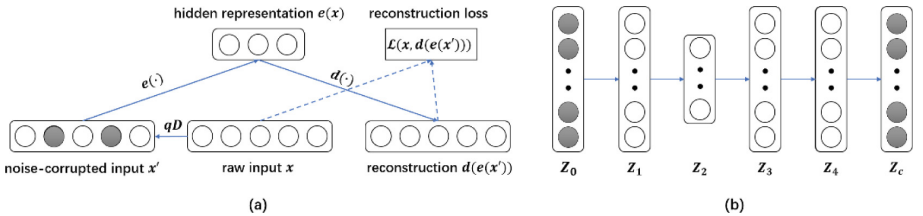


Fig. 2. (a) The architecture of DAE (b) A 2-layer SDAE with  $L = 4$

**SARF Extracting.** In our framework, we need to learn positive (negative) review feature extraction model  $SDAE_S^{pos}$  and  $SDAE_T^{pos}$  ( $SDAE_S^{neg}$  and  $SDAE_T^{neg}$ ) in both domains respectively. Then we will introduce the learning method of  $SDAE_S^{pos}$ , that is similar to  $SDAE_T^{pos}$ ,  $SDAE_S^{neg}$  and  $SDAE_T^{neg}$ .

As described in Sect. 3.1, each entry of  $R_{S,U}^{pos}$  is an (user, {sentence}, SO value) triplet. In our formulation, each {sentence} is considered as a ‘‘document’’ of the user and ‘‘corpus’’ is defined as the collection of {sentence}s in  $R_{S,U}^{pos}$ . We employ TF-IDF method to construct the ‘‘document-word’’ matrix  $X_{S,U}^{pos}$  as the clean input of the  $SDAE_S^{pos}$  model and generate noise-corrupted input  $X_{S,U}^{pos}$  by choosing 5 percent of matrix values randomly and setting them to 0, while the others are left unmodified. In the  $SDAE_S^{pos}$  model, given the total number of layers is  $L$ , the positive review feature vector is generated from the  $L/2$  layer, which is denoted as  $f_{S,u}$  for user  $u$  in source domain.

Similarly,  $SDAE_T^{pos}$ ,  $SDAE_S^{neg}$  and  $SDAE_T^{neg}$  are trained on sentiment classified review datasets  $R_{T,U}^{pos}$ ,  $R_{S,U}^{neg}$  and  $R_{T,U}^{neg}$  respectively. Then we can get the positive and negative review feature matrix in both domains. In this paper, the SARF represents as a set of positive and negative review feature vectors, such as  $SARF_{S,u} = (f_{S,u}^{pos}, f_{S,u}^{neg})$ , which is the sentiment-aware review feature of user  $u$  in source domain.

In addition, LDA can also be used to extract SARF under the above formulation. In the LDA version of our framework, we employ LDA on each sentiment classified review dataset shown in Fig. 1. Then we can get the positive and the negative topic distributions of each ‘‘document’’ which can be represented as SARF in both domains and the per-topic word distributions  $\beta_{T_U}^{pos}$ ,  $\beta_{T_U}^{neg}$ ,  $\beta_{T_I}^{pos}$  and  $\beta_{T_I}^{neg}$  in target domain.

## 5 Sentiment-Aware Review Feature Mapping and Cross-Domain Recommendation

### 5.1 Sentiment-Aware Review Feature Mapping

In this paper, we utilize an MLP-based method to tackle the SARF mapping problem, as shown in Fig. 1. To avoid mutual interference between positive and negative review features during the process of knowledge transfer, two MLP models, the positive MLP model and the negative MLP model, were employed to map the two parts of  $SARF = (f^{pos}, f^{neg})$  from source domain to target domain respectively. Next, we will introduce the proposed mapping algorithm under  $f^{pos}$  mapping scenario, and the mapping algorithm under  $f^{neg}$  mapping scenario is similar.

In our proposed mapping algorithm, only the common users with sufficient review data are used to learn the mapping function in order to guarantee its robustness to noise caused by review data sparsity and imbalance of review distribution in both domains. We use entropy and statistical method to measure the cross-domain degree of common users. Formally, the cross-domain degree is defined as follows:



$$c(u) = (-q_{u,s} \log_2 q_{u,s} - q_{u,t} \log_2 q_{u,t}) \frac{\mathcal{N}(u, s) + \mathcal{N}(u, t)}{\sum_{u_i \in U_c} \mathcal{N}(u_i, s) + \mathcal{N}(u_i, t)}$$

$$\text{where } q_{u,s} = \frac{\mathcal{N}(u, s)}{\mathcal{N}(u, s) + \mathcal{N}(u, t)}, q_{u,t} = \frac{\mathcal{N}(u, t)}{\mathcal{N}(u, s) + \mathcal{N}(u, t)} \quad (3)$$

$\mathcal{N}(u, s)$  is the number of reviews in source domain of common user  $u$ , and  $\mathcal{N}(u, t)$  is that in target domain.  $U_c$  denotes the set of common users between both domains. Intuitively, the previous part of  $c(u)$  is conditional entropy which indicates the distribution of the interactions of the user  $u$  in both domains. The latter part of  $c(u)$  measures the proportion of  $u$ 's reviews in both domain. The common users with  $c(u) > \text{threshold}\gamma$  are selected to learn the mapping function.

Let  $\theta^S = \{\theta_1^S, \theta_2^S, \dots, \theta_N^S\}$  denotes the set of  $f^{pos}$  s in the source domain, and  $\theta^T = \{\theta_1^T, \theta_2^T, \dots, \theta_N^T\}$  represents the set of  $f^{pos}$  s in the target domain.  $N$  is the number of common users in both domains. Under the MLP model setting, we formulate the  $f^{pos}$  mapping problem as: Given  $N$  training instance  $(\theta_i^S, \theta_i^T)$ ,  $\theta_i^S, \theta_i^T \in \mathbb{R}^M$ , ( $i = 1, 2, \dots, N$ ), where  $\theta_i^S = (\theta_{i1}^S, \theta_{i2}^S, \dots, \theta_{iM}^S)$  is the  $f^{pos}$  of common user  $u_i$  in the source domain and  $\theta_i^T = (\theta_{i1}^T, \theta_{i2}^T, \dots, \theta_{iM}^T)$  is that in the target domain, our task is to learn an MLP mapping function to map the  $f^{pos}$  from the source domain to the target domain.

In a feedforward MLP model, the output  $o_{ik}$  is formulated as

$$y_{ik} = \sum_{j=1}^{L'} c_{jk} a_j, o_{ik} = g(y_{ik}) \quad (4)$$

where  $c_{jk}$  represents the weight of the  $j$ 'th input of the output layer neuron  $k$  and  $L'$  is the number of hidden neurons in each hidden layer.  $g(y)$  is the activation function of the output layer, which is set to be the softmax function in this study.  $a_j$  denotes the  $j$ 'th hidden neuron activation of lower hidden layer, which can be defined as

$$y_j = \sum_{p=1}^P w_{pj} a_p, a_j = f(y_j) \quad (5)$$

where  $w_{pj}$  is the weight of the  $p$ 'th input of the hidden layer neuron  $j$  (the hidden bias can be included in the input weights) and  $a_p$  is the input  $\theta_{ip}^S$  or the  $p$ 'th hidden neuron activation of the lower hidden layer.  $P$  represents the number of inputs or neurons in the lower layer.  $f(y)$  is the hidden layer activation function, which is set to be the ReLU function in this study. Considering that input and output of MLP model in this study are feature vectors or topic distributions, the error between  $\theta_i^T$  and  $o_i = (o_{i1}, o_{i2}, \dots, o_{iM})$  is measured by Kullback-Leibler divergence:

$$E = \sum_{i=1}^N \sum_{k=1}^M o_{ik} \log \frac{o_{ik}}{\theta_{ik}^T} \quad (6)$$

To obtain the MLP mapping function, we utilize stochastic gradient descent to learn the parameters. We refresh the parameters of the MLP by looping through the training instances. The back-propagation algorithm is adopted to calculate the gradients of the parameters. The iterative process is stopped until the model converges, thus we can get the positive MLP mapping function  $\mathcal{F}_{pos}(\cdot; \Theta_p)$ , where  $\Theta_p$  is its weights set. Similarly, we can learn the negative MLP mapping function  $\mathcal{F}_{neg}(\cdot; \Theta_n)$ .

## 5.2 Cross-Domain Recommendation

For a cold-start user in target domain, we do not have sufficient information to estimate his/her preference directly in target domain. However, we can get the affine SARF for him/her in target domain, with the SARF learned in source domain and the learned MLPs. In this section, we will introduce how to predict the cold-start user's preference on the items by using SARF in target domain.

Given a cold-start user  $u'$  in the target domain, we can extract user  $u'$ 's  $SARF_{S,u'}$  =  $(f_{S,u'}^{pos}, f_{S,u'}^{neg})$  from  $R_{S,U}^{pos}$  and  $R_{S,U}^{neg}$  in the source domain. Then the affine  $\widehat{SARF}_{T,u'}$  =  $(\hat{f}_{T,u'}^{pos}, \hat{f}_{T,u'}^{neg})$  can be obtained by the following equations:

$$\hat{f}_{T,u'}^{pos} = \mathcal{F}_{pos}(f_{S,u'}^{pos}; \Theta_p), \quad \hat{f}_{T,u'}^{neg} = \mathcal{F}_{neg}(f_{S,u'}^{neg}; \Theta_n) \quad (7)$$

Since  $R_{T,U}^{pos}$  and  $R_{T,I}^{pos}$  share the same review data in target domain, we can extract positive review feature matrix of the items in target domain  $F_I^{pos}$  by applying the learned  $SDAE_T^{pos}$  on  $R_{T,I}^{pos}$ . Similarly, we can get  $F_I^{neg}$  by employing the learned  $SDAE_T^{neg}$  on  $R_{T,I}^{neg}$ . Then the predicted emotional preference of cold-start user  $u'$  to item  $i_j \in \mathcal{I}_T$  in the target domain can be calculated as:

$$e_{u',i_j} = \sigma \left( OS_{S,u'}^{pos} \cdot \hat{f}_{T,u'}^{pos} \cdot f_{i_j}^{posT} + OS_{T,i_j}^{neg} \cdot \hat{f}_{T,u'}^{neg} \cdot f_{i_j}^{negT} \right) \quad (8)$$

where  $OS_{S,u'}^{pos}$  and  $OS_{T,i_j}^{neg}$  are the average OS values of ‘‘document’’ described in Sect. 3.1, and  $\sigma(\cdot)$  is the sigmoid function.

For the LDA version of our framework, we calculate the similarity matrixes of latent topic spaces of user review and item review in the target domain, which are defined as:

$$SIM^{pos} = \left\{ sim_{i,j}^{pos} \right\}, \quad \text{where } sim_{i,j}^{pos} = \cos(\beta_{T_{U,i}}^{pos}, \beta_{T_{I,j}}^{pos}), \quad i, j = 1, 2, \dots, M \quad (9)$$

$$SIM^{neg} = \left\{ sim_{i,j}^{neg} \right\}, \quad \text{where } sim_{i,j}^{neg} = \cos(\beta_{T_{U,i}}^{neg}, \beta_{T_{I,j}}^{neg}), \quad i, j = 1, 2, \dots, M \quad (10)$$

where  $\beta_{T_{u,i}} \left( \beta_{T_{i,i}} \right)$  represents the topic-word distribution of topic  $i$ , which is in the latent topic space of user (item) review in the target domain.  $\cos(\cdot)$  denotes the cosine similarity of two topic-word distributions.

Then the predicted emotional preference of cold-start user  $u'$  to item  $t_j \in \mathcal{I}_T$  in the target domain is calculated as:

$$e_{u',t_j} = \sigma \left( OS_{S,u'}^{pos} \cdot \hat{f}_{T,u'}^{pos} \cdot SIM^{pos} \cdot f_{t_j}^{posT} + OS_{T,t_j}^{neg} \cdot \hat{f}_{T,u'}^{neg} \cdot SIM^{neg} \cdot f_{t_j}^{posT} \right) \quad (11)$$

Finally, the overall preference of cold-start user  $u'$  to item  $t_j$  can be represented as a linear combination of a cross-domain average-value method and an emotional preference component as:

$$S(u', t_j) = b_{\mathcal{I}_T} + b_{u'} + b_{t_j} + e_{u',t_j} \quad (12)$$

where  $b_{\mathcal{I}_T}$  denotes the overall average ratings of all items in the target domain. The parameter  $b_{u'}$  is the user rating bias in the source domain and  $b_{t_j}$  is the item rating bias in the target domain.

## 6 Experiments

### 6.1 Experimental Settings

**Data Description.** We employ Amazon cross-domain dataset [11] in our experiment. This dataset contains product reviews and star ratings with 5-star scale from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. We select the top three domains with the most widely used in previous studies to employ in our cross-domain experiment. The global statistics of these domains used in our experiments are shown in Table 1.

**Table 1.** Characteristics of datasets

Dataset	Books	Electronics	Movies & TV
# of Users	603,668	192,403	123,960
# of Items	367,982	63,001	50,052
# of Reviews	8,898,041	1,689,188	1,697,533
Density	0.004%	0.014%	0.027%

**Experiment Setup.** The domains in the Amazon dataset only have user overlaps. Thus, we evaluate the validity and efficiency of SARFM on the cross-domain recommendation task under the user overlap scenario. We randomly remove all the rating information of a certain proportion of users in the target domain and take them as cross-

domain cold-start users for making recommendation. For the sake of stringency of the experiments, we set different proportions of cold-start users, namely,  $\phi = 30\%$ ,  $50\%$  and  $70\%$ . Moreover, we repeatedly sample users for 10 times to generate different sets to balance the effect of different sets of cold-start users on the final recommendation results and report the average results. Dimension of latent feature used in SDAE and the number of topics in LDA are set as  $M = 20, 50$  and  $100$ . For the mapping function, we set the structure of MLP as three hidden layers with  $2M$  nodes in each hidden layer.

**Compared Methods.** We examine the performance of the proposed SARFM framework by comparing it with the following baseline methods:

- SD\_AVG: SD\_AVG is a single-domain average-value method, which predicts overall preference by global average rating.
- CD\_AVG: CD\_AVG is a cross-domain average-value method described in Sect. 4.2. It predicts overall preference by the following equation:  $r_{ui} = b_T + b_u + b_i$  where  $b_T$  is the overall average ratings of all items in the target domain,  $b_u$  denotes the user rating bias in source domain and  $b_i$  represents item bias in target domain.
- CMF: This is a cross-domain recommendation method proposed in [14]. In CMF, the latent factors of users are shared between source domain and target domain.
- MF-MLP: This is a cross-domain recommendation framework based on MF and MLP, which is proposed by [15]. In our experiments, the structure of the MLP is set as three-hidden layer, and the number of nodes in the hidden layer is set as  $2M$ .
- SARFM\_SDAE, SARFM\_LDA: These are two different implementations of our framework proposed in this paper.
- RFM\_SDAE, RFM\_LDA: These are two cut versions of our framework without sentiment analysis.

### Evaluation Metric

We adopt the metrics of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) to evaluate our method. They are defined as:

$$RMSE = \sqrt{\sum_{r_{ui} \in \mathcal{O}_{test}} \frac{(\hat{r}_{ui} - r_{ui})^2}{|\mathcal{O}_{test}|}}, \quad MAE = \frac{1}{|\mathcal{O}_{test}|} \sum_{r_{ui} \in \mathcal{O}_{test}} |\hat{r}_{ui} - r_{ui}| \quad (13)$$

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

## 6.2 Performance Comparison

**Recommendation Performance.** Experimental results of MAE and RMSE on the two pair of domains “Books-Movies & TV” and “Electronics-Movies & TV” are presented in Tables 2, 3, 4, and 5, respectively. The domain “Books” and “Electronics” are chosen as the target domain because they are much sparser compared with “Movies & TV”.

**Table 2.** Recommendation performance in terms of MAE on the “Books-Movies & TV”

DIM	$\phi$	SD_AVG	CD_AVE	CMF	MF_MLP	SARFM_LDA	SARFM_SDAE
K = 20	30%	0.8767	0.8655	0.8616	0.8446	0.8161	0.8155
	50%	0.8948	0.8663	0.8904	0.8459	0.8382	0.8254
	70%	0.9150	0.8763	0.9101	0.8809	0.8637	0.8485
K = 50	30%	0.8767	0.8655	0.8441	0.8442	0.8338	0.8108
	50%	0.8948	0.8663	0.8843	0.8530	0.8459	0.8252
	70%	0.9150	0.8763	0.8881	0.8790	0.8636	0.8270
K = 100	30%	0.8767	0.8655	0.8254	0.8440	0.8156	0.8111
	50%	0.8948	0.8663	0.8523	0.8687	0.8459	0.8254
	70%	0.9150	0.8763	0.8990	0.8940	0.8635	0.8278

**Table 3.** Recommendation performance in terms of RMSE on the “Books-Movies & TV”

DIM	$\phi$	SD_AVG	CD_AVE	CMF	MF_MLP	SARFM_LDA	SARFM_SDAE
K = 20	30%	1.1440	1.1177	1.0962	1.0885	1.0849	1.0419
	50%	1.1645	1.1400	1.1267	1.1074	1.0894	1.0476
	70%	1.1735	1.1654	1.1361	1.1531	1.1141	1.0585
K = 50	30%	1.1440	1.1177	1.0858	1.0849	1.0849	1.0369
	50%	1.1645	1.1400	1.1275	1.1253	1.0945	1.0474
	70%	1.1735	1.1654	1.1560	1.1621	1.1141	1.0570
K = 100	30%	1.1440	1.1177	1.1296	1.1238	1.0756	1.0420
	50%	1.1645	1.1400	1.1513	1.1506	1.0847	1.0476
	70%	1.1735	1.1654	1.1607	1.1571	1.1076	1.0679

**Table 4.** Recommendation performance in terms of MAE on the “Electronics-Movies & TV”

DIM	$\phi$	SD_AVG	CD_AVE	CMF	MF_MLP	SARFM_LDA	SARFM_SDAE
K = 20	30%	0.9459	0.8626	0.8594	0.8584	0.8509	0.8356
	50%	0.9529	0.9064	0.8960	0.8864	0.8637	0.8619
	70%	0.9733	0.9201	0.9175	0.9023	0.8660	0.8640
K = 50	30%	0.9459	0.8626	0.8599	0.8605	0.8504	0.8346
	50%	0.9529	0.9064	0.8829	0.8820	0.8630	0.8610
	70%	0.9733	0.9201	0.9005	0.8945	0.8658	0.8639
K = 100	30%	0.9459	0.8626	0.8509	0.8625	0.8500	0.8347
	50%	0.9529	0.9064	0.8840	0.9012	0.8623	0.8610
	70%	0.9733	0.9201	0.9060	0.9119	0.8654	0.8642

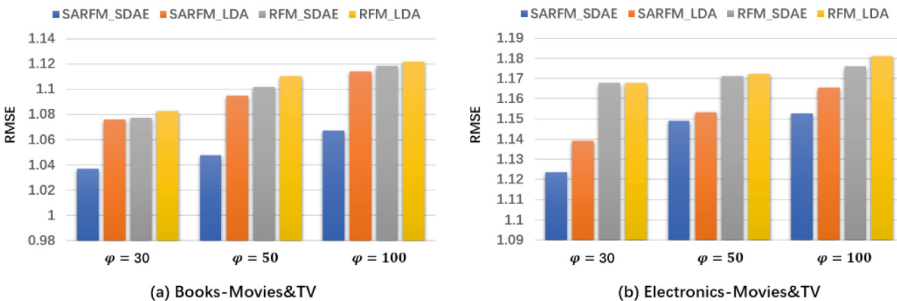
We respectively evaluate all the methods under different  $K$  and  $\phi$  in both cross-domain scenarios. From these tables, we can see that the proposed SARFM\_SDAE outperforms all baseline models in terms of both MAE and RMSE metrics. With the proportion of cold-start users becoming higher, the performance of single domain

**Table 5.** Recommendation performance in terms of RMSE on the “Electronics-Movies & TV”

DIM	$\phi$	SD_AVG	CD_AVE	CMF	MF_MLP	SARFM_LDA	SARFM_SDAE
K = 20	30%	1.2408	1.1603	1.1425	1.1437	1.1391	1.1237
	50%	1.2654	1.2022	1.1794	1.1564	1.1536	1.1495
	70%	1.2948	1.2169	1.1833	1.1781	1.1659	1.1533
K = 50	30%	1.2408	1.1603	1.1601	1.1572	1.1392	1.1234
	50%	1.2654	1.2022	1.1835	1.1807	1.1533	1.1490
	70%	1.2948	1.2169	1.1731	1.2068	1.1656	1.1528
K = 100	30%	1.2408	1.1603	1.1401	1.1592	1.1390	1.1234
	50%	1.2654	1.2022	1.1708	1.1914	1.1532	1.1494
	70%	1.2948	1.2169	1.1838	1.2159	1.1656	1.1529

method SD\_AVG will become progressively worse while the cross-domain methods keep satisfactory results, which shows the effectiveness of knowledge transfer. Compared with CMF and MF\_MLP, our method gets an improvement of 5% to 10% both in RMSE and MAE. These results demonstrate that our framework is more suitable for making recommendations to cold-start users compared to other cross-domain baseline methods, especially in the dataset with high sparsity. Moreover, SARFM\_SDAE performs better than CD\_AVE especially in higher  $\phi$ , which demonstrates that the SARF transferred from the source domain is highly effective. For the proposed SARFM\_SDAE and SARFM\_LDA methods, the optimal values of  $K$  are nearly 50 and 100 respectively. In the following experiments,  $K$  are set to the optimal values by preliminary tests.

**Effect of Semantic Information.** We removed the sentiment analysis component from our framework to get two incomplete versions of RFM\_SDAE and RFM\_LDA, which extract review features from raw review dataset directly. As shown in Fig. 3, both SARFM\_SDAE and SARFM\_LDA perform much better than RFM\_SDAE and RFM\_LDA under different  $\phi$  in both cross-domain scenarios, which verifies the effectiveness of our method in the aspect of utilizing semantic information.

**Fig. 3.** Effect of semantic information on recommendation performance

**Topic Model vs. Deep learning Model.** We designed two implementations of our framework in this paper, SARFM\_SDAE and SARFM\_LDA. For the SARFM\_LDA, the interpretable recommendation results can be gotten by matching topic distributions  $\beta_{T_i}$  and  $\beta_{T_u}$  in the target domain. While the SARFM\_SDAE is hard to provide a natural interpretation for the recommendation result because of the latent feature extraction. However, as shown in Fig. 3, SARFM\_SDAE outperforms SARFM\_LDA under each of configurations, which verifies its effectiveness on extracting the deep latent feature of sentiment information.

## 7 Conclusions

The user reviews contain plenty of semantic information. We proposed a novel framework for cross domain recommendation that establishes linkages between the source and target domains by using sentiment-aware review features of users. In this paper, a user-personalized sentiment analysis algorithm is designed to analyze the Semantic Orientation of each review in both domains. We employed SDAE and MLP based mapping method to model user's SARF and mapped it to the target domain to make recommendations for cold-start users. In different scenarios, experiments convincingly demonstrate that the proposed SARFM framework can significantly improve the quality of cross-domain recommendation and SARFs extracted from reviews are important links between each domain.

**Acknowledgments.** This work is supported by NSF of Shandong, China (Nos. ZR2017MF065, ZR2018MF014), the Science and Technology Development Plan Project of Shandong, China (No. 2016GGX101034).

## References

1. Cremonesi, P., Tripodi, A., Turrin, R.: Cross-domain recommender systems. In: ICDM 2012, pp. 496–503 (2012)
2. Fang, Z., Gao, S., Li, B., Li, J.: Cross-domain recommendation via tag matrix transfer. In: ICDM 2015, pp. 1235–1240 (2015)
3. Chen, W., Hsu, W., Lee, M.L.: Making recommendations from multiple domains. In: KDD 2013, pp. 892–900 (2013)
4. Yang, D., He, J., Qin, H., Xiao, Y., Wang, W.: A graph-based recommendation across heterogeneous domains. In: CIKM 2015, pp. 463–472 (2015)
5. Li, B., Yang, Q., Xue, X.: Transfer learning for collaborative filtering via a rating-matrix generative model. In: ICML 2009, pp. 617–624 (2009)
6. Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., Suter, B.W.: The multilayer perceptron as an approximation to a Bayes optimal discriminant function. *IEEE Trans. Neural Netw.* **1**(4), 296–298 (1990)
7. Vincent, P., Larochelle, H., Lajoie, I., et al.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11** (12), 3371–3408 (2010)

8. Ren, S., Gao, S., Liao, J., Guo, J.: Improving cross-domain recommendation through probabilistic cluster-level latent factor model. In: *AAAI 2015*, pp. 4200–4201 (2015)
9. Gao, S., Luo, H., Chen, D., Li, S., Gallinari, P., Guo, J.: Cross-domain recommendation via cluster-level latent factor model. In: *Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013. LNCS (LNAD)*, vol. 8189, pp. 161–176. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-40991-2\\_11](https://doi.org/10.1007/978-3-642-40991-2_11)
10. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
11. He, R., McAuley, J.: Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering. In: *WWW 2016*, pp. 507–517 (2016)
12. He, X., Liao, et al.: Neural collaborative filtering. In: *WWW 2017*. pp. 173–182 (2017)
13. Pan, W., Liu, N.N., Xiang, E.W., Yang, Q.: Transfer learning to predict missing ratings via heterogeneous user feedbacks. In: *AAAI 2011*, pp. 2318–2323 (2011)
14. Singh, A.P., Kumar, G., Gupta, R.: Relational learning via collective matrix factorization. In: *KDD 2008*, pp. 650–658 (2008)
15. Man, T., Shen, H., Jin, X., Cheng, X.: Cross-domain recommendation: an embedding and mapping approach. In: *IJCAI 2017*, pp. 2464–2470 (2017)
16. Xin, X., Liu, Z., Lin, C., Huang, H., Wei, X., Guo, P.: Cross-domain collaborative filtering with review text. In: *IJCAI 2015*, pp. 1827–1833 (2015)
17. Diao, Q., Qiu, M., Wu, C.Y., Smola, A.J., et al.: Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In: *KDD 2014*, pp. 193–202 (2014)
18. Zhang, Y., Lai, G., Zhang, M., et al.: Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In: *SIGIR 2014*, pp. 83–92 (2014)
19. Li, F., Wang, S., Liu, S., Zhang, M.: SUIIT: a supervised user-item based topic model for sentiment analysis. In: *AAAI 2014*, pp. 1636–1642 (2014)
20. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contractive auto-encoders: explicit invariance during feature extraction. In: *ICML 2011*, pp. 833–840 (2011)
21. Chen, M., Xu, Z., Weinberger, K., Sha, F.: Marginalized denoising autoencoders for domain adaptation. *Computer Science* (2012)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *NIPS 2012*, pp. 1097–1105 (2012)
23. Graves, A., Fernández, S., et al.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: *ICML*, pp. 369–376 (2006)
24. Taboada, M., Brooke, J., Tofiloski, M., et al.: Lexicon-based methods for sentiment analysis. *Comput. Linguist.* **37**(2), 267–307 (2011)
25. Wang, H., Wang, N., Yeung, D. Y.: Collaborative deep learning for recommender systems. In: *SIGKDD 2015*, pp. 1235–1244 (2015)