Cyclic Transfer Learning for Recommender Systems with Heterogeneous Feedbacks

Cyclic Transfer Learning for Recommender Systems with Heterogeneous Feedbacks

Xuelian Ni[†] Fei Xiong [*†] Yutian Hu[†] Shirui Pan[‡] Hongshu Chen[§] Liang Wang[¶]

Abstract

Transfer learning uses auxiliary domains to help complete learning tasks of the target domain. However, the combination of recommendation and transfer learning often has two problems. One is that it's difficult to find an auxiliary domain which is highly related to the target domain. The other is that useful information in auxiliary domains cannot be fully utilized. To make use of the knowledge in auxiliary domains as much as possible, this paper proposes a cyclic transfer learning method which can transfer the shared knowledge in the auxiliary domain and target domain multiple times. Combining this method with recommendation, this paper presents a recommendation framework based on heterogeneous feedbacks and cyclic transfer learning (HCTL-Rec). By studying the relationship between different behaviors of users, this paper proposes two specific recommendation algorithms which combine the novel framework with two auxiliary domains. One is to use users' binary attitude information as an auxiliary domain to better represent users' ratings. The other is to use users' trust relationship as an auxiliary domain and make social recommendation. Experiments are carried out on two real-world datasets with trust relationship. The results show that recommendation quality of the two specific algorithms can achieve significant improvement compared with other state-of-the-art algorithms and can effectively relieve the cold-start problem.

1 Introduction

With the rapid development of Internet technology, the amount of data in the network is increasing exponentially. In the case of information overload [1], it is difficult for users to find favorite items in a timely and accurate manner. Recommender systems can filter a large amount of irrelevant information for users by analyzing the relevant characteristics of users and items and recommend items that meet users' preferences [2], [3]. The accuracy of recommender systems is often limited by two major problems. One is data sparsity since users often rate a small number of items [4], [5]. The other one is the cold-start problem caused by new items or new users [6], [7].

As one of the most classic recommendation algorithms, collaborative filtering (CF) has a prominent advantage that it is easy to implement in engineering. CF can be divided into three categories, that is, user-based CF [8], item-based CF [9] and model-based CF [10],[11]. The user/item-based CF analyzes existing ratings to find users with similar interests or items with similar characteristics. Then, it makes recommendation based on the weighted sum of ratings of the nearest neighbors. The model-based CF uses ratings to train the model and then uses the model to predict missing ratings. The most classic model of model-based CF is matrix factorization (MF), which converts a sparse high-dimensional matrix into two low-dimensional matrices and multiplies them to predict missing ratings.

Although MF achieves good prediction results, its performance will be limited in the case of data sparsity and cold start. Many researchers try to combine different types of data with ratings to design recommendation algorithms [12]-[16]. Transfer learning can extract valuable knowledge from different information domains to help improve the learning performance in the target domain. It has been widely used in many fields such as image processing and autonomous driving [17]. In addition, transfer learning has also been proven that it can solve the problem of data sparsity and cold start effectively [18]. Therefore, the field of recommender systems has also begun to combine transfer learning with traditional recommendation algorithms. Usually, in recommender systems, the rating domain is called the target domain, and other information domains are called auxiliary domains. Recommendation algorithms based on transfer learning usually use auxiliary domains related to users or items and extract shared information between auxiliary domains and the target domain as a bridge. Finally, the transfer of knowledge is carried out

[*]Corresponding Author.

[†]These authors are co first authors. Key Laboratory of Communication and Information Systems, Beijing Jiaotong University, Beijing 100044, China. {xuelian.ni, xiongf}@bjtu.edu.cn.

[‡]Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia. shirui.pan@monash.edu.

[§]School of Management and Economics, Beijing Institute of Technology, Beijing 100081, China. hongshu.chen@bit.edu.cn.

[¶]School of Computer Science, Northwestern Polytechnical University, Xi'an 10072, China. liangwang0123@gmail.com.

through the bridge. The transferred knowledge can help rating prediction of the target domain.

Compared with traditional algorithms that only use a single domain, transfer learning involve multiple data domains. Therefore, more attention should be paid in the process of knowledge transfer and rating prediction to avoid introducing noise and degrading the recommendation performance. The problems that need to be studied mainly contain two aspects: the data domain and the method of transfer learning [19]. Usually, users' ratings and attitudes towards the same item tend to be consistent. Users tend to be affected by their trusted users. Therefore, we use attitude information and trust relationship as auxiliary domains. Our work has the following contributions:

(1) We analyze the relationship between the different behaviors of users towards items, and study the construction method of implicit features of auxiliary domains and shared features between the target and auxiliary domains.

(2) We combine transfer learning and MF to propose a recommendation framework with heterogeneous feedbacks and cyclic transfer learning (HCTLRec). In particular, to make full use of auxiliary domains, we design a cyclic transfer learning method to complete the two-way knowledge transfer.

(3) We use attitude information and trust relationship as two auxiliary domains combined with the HCTLRec framework to implement two specific recommendation algorithms, that is, HCTLRec-B and HCTLRec-T.

(4) Using two real-world datasets, we conduct experiments to test the performance of two specific algorithms in general environment and cold-start environment. The results show that compared with other transfer learning algorithms, HCTLRec-B and HCTLRec-T can both achieve better results.

2 RELATED WORK

In traditional machine learning, multiple tasks are independently conducted and do not have correlation [8]-[11]. Each learning task needs to train an applicable model according to its own task. Transfer learning is suitable for the environment of data sparsity or cold start [17]-[18],[20]. The training samples in the target task are regarded as the target domain and related data that contain additional training samples are regarded as auxiliary domains [19]. Learning valuable information from an auxiliary domain and transferring it to the target domain can help the target task to obtain better results. The process of transfer learning is shown in Fig. 1.

At present, recommender systems based on transfer learning have gotten some achievements [12],[13],[22].

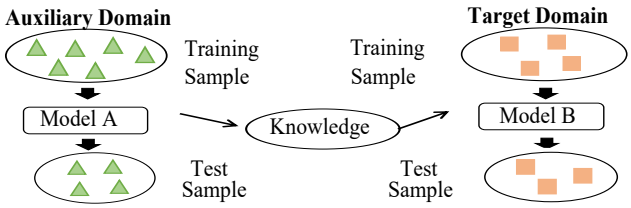


Figure 1: Workflow of Transfer Learning

Jiang et al. [21] designed a recommendation algorithm with knowledge transfer in a random walk manner by studying the relationship between different information in the social network platform. The algorithm makes the prediction with only one auxiliary domain, and it can also expand auxiliary domains according to the actual situation. Pan et al. proposed a recommendation algorithm, i.e., transfer by mixed factorization (TMF) [14] that uses two kinds of heterogeneous data to conduct transfer learning. TMF uses rating data as the target domain and users' attitude as an auxiliary domain. Then, it uses the features of items as a bridge to transfer knowledge.

Deep learning is used to better exploit the features of users and items in recommender systems with transfer learning. Yuan et al. [12] developed a deep domain adaptation model which can extract and transfer patterns from the rating matrix without considering any auxiliary information. Jiang et al. [22] factorized the sparse high-dimensional rating matrix multiple times to obtain a shared feature matrix and multiple feature matrices. The auxiliary information can be obtained only by deep factorization of the sparse rating matrix.

3 HCTLREC FRAMEWORK

3.1 Problem definition This paper studies the recommendation with transfer learning, so it is necessary to consider two data domains, that is, the target domain and auxiliary domain. We assume that there are m users and n items in the system. The rating domain in the system is the target domain, which is defined as

$$R = \{(u, i, r_{ui}) | u \in \{1, 2 \dots m\}, i \in \{1, 2 \dots n\} \\ r_{ui} \in 1, 2, 3, 4, 5\} \tag{3.1}$$

where u represents the user ID, i represents the item ID, and r_{ui} represents the rating that user u has given to item i. In addition, an auxiliary domain in the recommender system needs to be defined. The content of the auxiliary domain varies according to the selected data in actual.

3.2 Construction of auxiliary domains HCTLRec first extracts the valuable feedback in an auxiliary domain, allowing the information in the auxiliary do-

main to complete two tasks. The features in an auxiliary domain which are associated with the target domain can directly help the construction of the features in the target domain. Therefore, the first task is to supplement the implicit feedback information in the auxiliary domain and directly integrate it into the prediction model of the target domain. The second task is to design a predictive model for the auxiliary domain and extract the shared features of the two domains as a bridge for knowledge transfer. Specifically, HCTLRec uses MF in the target domain to calculate the missing value $\hat{r}_{ui}$, as shown in (3.2).

$$\hat{r}_{ui} = \mu_r + b_{ur} + b_{ir} + s_u \cdot a_i \tag{3.2}$$

where $\hat{r}_{ui}$ represents the predicted value of the target domain, μ_r represents the global average rating in the target domain, b_{ir} represents the rating bias of item i in the target domain and b_{ur} represents the rating bias of user u. s_u represents the shared characteristics of the target and auxiliary domain, mainly including the feature vector of user u. a_i represents the unique characteristics of item i in the target domain, including the inherent features in the target domain and the implicit feedback in the auxiliary domain which is obtained from the aforementioned first task of the auxiliary domain. MF is also used as the prediction model of the auxiliary domain, as shown in (3.3).

$$\hat{y}_{ui} = s_u \cdot b_i \tag{3.3}$$

where $\hat{y}_{ui}$ represents the predicted value of the auxiliary domain. s_u here is completely consistent with that in (3.2). It promotes the two-way knowledge transfer between the two domains. b_i represents the unique feature vector of the auxiliary domain.

3.3 Rating prediction with cyclic transfer learning According to (3.2) and (3.3), to optimize the prediction, it is necessary to minimize the prediction errors in the target and auxiliary domain. Therefore, the objective function is shown in (3.4) and (3.5).

$$L_1 = \min_{\theta_r} \sum_u \sum_i [\frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \frac{\lambda_r}{2}||\theta_r||^2], \leftrightarrow s_u \tag{3.4}$$

$$L_2 = \min_{\theta_a} \sum_u \sum_i [\frac{1}{2}(y_{ui} - \hat{y}_{ui})^2 + \frac{\lambda_a}{2}||\theta_a||^2], \leftrightarrow s_u \tag{3.5}$$

where L_1 and L_2 represent the formal representation of the objective function. θ_r and θ_a represent the parameters that need to be fitted, respectively. λ_r and λ_a represent the regularization parameters, respectively. $||\cdot||^2$ represents the regularization term of the related parameters, and it can prevent overfitting. The notation $\leftrightarrow s_u$ indicates that s_u is used as a bridge to promote knowledge transfer. Both objective functions adopt the stochastic gradient descent (SGD) method in the parameter fitting process. Different auxiliary domain embedding models can be selected according to the actual situation, and the specific calculation algorithm needs to be analyzed according to the auxiliary domain. Rating prediction of HCTLRec is shown in Fig. 2.

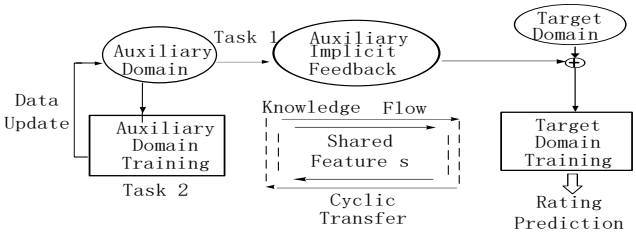


Figure 2: Overall Framework Diagram of HCTLRec

4 RECOMMENDATION ALGORITHMS BASED ON HCTLREC

4.1 HCTLRec-Binary attitude User's two-level attitude information can be used to construct the user's preference profile, with 1 for likes and -1 for dislikes. We use the attitude as the auxiliary domain, combined with the HCTLRec framework, to realize the transfer learning algorithm HCTLRec-Binary attitude (HCTLRec-B).

Because the attitude information still comes from the same system, the attitude matrix A, like the rating matrix, has also $m \times n$ dimensions, which contains the relevant information of m users and n items, namely:

$$A = \{(u, i, y_{ui}) | u \in \{1, 2 \dots m\}, i \in \{1, 2 \dots n\} , y_{ui} \in \{-1, 1\}\} \tag{4.6}$$

where y_{ui} represents the attitude of user u towards item i. If the user marks the item as $y_{ui} = 1$, it means item i is liked by user u; otherwise, $y_{ui} = -1$ means dislike. Therefore, we count the set of users who like item i, denoted by Pos_i. The auxiliary feature vector $\bar{v}_i$ of the item can be constructed, as shown in (4.7).

$$\bar{v}_i = |Pos_i|^{-\frac{1}{2}} \sum_{j \in Pos_i} v_j \tag{4.7}$$

where $\bar{v}_i$ is composed of user's attitude implicit vector v_j, and it is the sum of the implicit feedbacks of item i from the set of users who like item i. Therefore, $\bar{v}_i$ can also be used as a part of the item characteristics of the target domain, and then, we complete task 1 of the auxiliary domain in the HCTLRec framework. Combined with the implicit feedback $\bar{v}_i$ of the attitude auxiliary domain, $\hat{r}_{ui}$ of the target domain in (3.2) is revised as (4.8).

$$\hat{r}_{ui} = \mu_r + b_{ur} + b_{ir} + p_u \cdot (\rho * q_i + (1 - \rho) * \bar{v}_i)^T \tag{4.8}$$

where q_i represents the feature vector of the item, and p_u represents user u's feature vector and here is the specific expression of the shared feature vector s_u. ρ represents the weight parameter, which controls the

influence of q_i and $\bar{v}_i$.

Considering that users' ratings depend on the unique nature of users and items [23], according to (3.3) in the second task of the auxiliary domain, the prediction of the attitude auxiliary domain is shown in (4.9).

$$\hat{y}_{ui} = \mu_a + b_{ua} + b_{ia} + p_u w_i^T \tag{4.9}$$

where μ_a represents the global average attitude, b_{ua} and b_{ia} represent the user's bias and the item's bias in the auxiliary domain, respectively. w_i is the feature vector of the item in the auxiliary domain. p_u is the shared feature vector of the two domains and will serve as a bridge for transfer learning.

Then, we realize the specific expression of HCTLRec. According to (4.8) and (4.9), the objective functions in the target and auxiliary domain of HCTLRec-B are shown in (4.10) and (4.11), respectively.

$$\begin{aligned} L_1 = \sum_u \sum_{i \in I(u)} [\frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \frac{\lambda_r}{2}(||b_{ur}||^2 + ||b_{ir}||^2 \\ + ||p_u||^2 + ||q_i||^2 + \sum_{j \in Pos_i} ||v_j||^2]), \leftrightarrow p_u \end{aligned} \tag{4.10}$$

$$\begin{aligned} L_2 = \sum_u \sum_{i \in A(u)} [\frac{1}{2}(y_{ui} - \hat{y}_{ui})^2 + \frac{\lambda_a}{2}(||b_{ua}||^2 + ||b_{ia}||^2 \\ + ||p_u||^2 + ||w_i||^2), \leftrightarrow p_u \end{aligned} \tag{4.11}$$

where $\leftrightarrow p_u$ means that p_u is used as a bridge for knowledge transfer. $I(u)$ represents the set of items rated by user u in the target domain, and $A(u)$ represents the set of items that has been marked by u in the auxiliary domain.

The gradient of each parameter in the objective function L_1 is shown in (4.12).

$$\begin{aligned} \frac{\partial L_1}{\partial b_{ur}} &= -\sum_{i \in I(u)} e_{ui} + \lambda_r b_{ur} \\ \frac{\partial L_1}{\partial b_{ir}} &= -\sum_{u \in U(i)} e_{ui} + \lambda_r b_{ir} \\ \frac{\partial L_1}{\partial p_u} &= -\sum_{i \in I(u)} e_{ui}(\rho \cdot q_i + (1-\rho) \cdot \bar{v}_i) + \lambda_r p_u \\ \frac{\partial L_1}{\partial q_i} &= -\sum_{u \in U(i)} e_{ui} \rho p_u + \lambda_r q_i \\ \frac{\partial L_1}{\partial v_j} &= -\sum_{u \in U(i)} e_{ui} \frac{1}{\sqrt{|Pos_i|}}(1-\rho)\, p_u + \lambda_r v_j, j \in Pos_i \end{aligned} \tag{4.12}$$

The gradient of each parameter in the objective function L_2 is shown in (4.13).

$$\begin{aligned} \frac{\partial L_2}{\partial b_{ua}} &= -\sum_{i \in A(u)} \tilde{e}_{ui} + \lambda_r b_{ur} \\ \frac{\partial L_2}{\partial b_{ia}} &= -\sum_{u \in Y(i)} \tilde{e}_{ui} + \lambda_r b_{ir} \\ \frac{\partial L_2}{\partial p_u} &= -\sum_{i \in A(u)} \tilde{e}_{ui} w_i + \lambda_r p_u \\ \frac{\partial L_2}{\partial w_i} &= -\sum_{u \in Y(i)} \tilde{e}_{ui} p_u + \lambda_r w_i \end{aligned} \tag{4.13}$$

where $e_{ui} = r_{ui} - \hat{r}_{ui}$ and $\tilde{e}_{ui} = y_{ui} - \hat{y}_{ui}$ respectively represent the rating and attitude value prediction error, $U(i)$ represents the set of users who have rated item i in the target domain, $Y(i)$ is the set of users who have marked attitude of item i in the auxiliary domain.

According to SGD, the iterative equation in the fitting process is shown in (4.14).

$$\theta_{(n+1)} = \theta_n - \alpha \frac{\partial L}{\partial \theta_n} \tag{4.14}$$

where α represents the learning rate, and θ represents the parameters that need to be fitted.

The training of the HCTLRec-B is carried out according to the framework of HCTLRec. The detailed calculation process is shown in Algorithm 1.

Algorithm 1 HCTLRec-B

Input: Number of outer iteration N_{out}, weight parameter ρ
Target Domain: Rating triples (user ID, item ID, rating), regularization parameter λ_r, learning parameter α_r, iteration numbers N_r
Auxiliary Domain: Attitude triples (user ID, item ID, attitude), regularization parameter λ_a, learning parameter α_a, iteration numbers N_a
Output: Target domain parameter θ_r:b_{ur},b_{ir},p_u,q_i,v_j, auxiliary domain parameter θ_a:b_{ua},b_{ia},p_u,w_i

```
Generating target matrix R, auxiliary matrix A
Initialization parameters
for n_out = 1, 2, ... N_out do://outer iteration
    for i = 1, 2, ... n do
        Select the user set Pos_i that prefer item i
    for n_r = 1, 2, ... N_r do://inner iteration
        compute gradients of θ_r according to (4.12)
        update θ_r according to (4.14)
    for n_a = 1, 2, ... N_a do:// inner iteration
        compute gradients of θ_a according to (4.13)
        update θ_a according to (4.14)
return θ_r:b_ur,b_ir,p_u,q_i,v_j;θ_a:b_ua,b_ia,p_u,w_i
```

4.2 HCTLRec-Trust relationship In social networks, trusters are willing to follow their friends' choices [24], [25]. The trust relationship between users in the system can be used as the auxiliary domain to provide transferable knowledge for rating prediction of the target domain. Combined with the HCTLRec framework, we propose the HCTLRec-Trust relationship (HCTLRec-T) algorithm, and the implicit feedback of the trust relationship reflects user' characteristics.

The dimensions of the trust matrix should be $m \times m$, which indicates the trust relationship between m users in the system, shown in (4.15).

$$\begin{aligned} T = &\{(u, j, z_{uj}) | u \in \{1, 2 \dots m\}, \\ &j \in \{1, 2 \dots m\}, z_{uj} \in \{0,\ 1\}\} \end{aligned} \tag{4.15}$$

where u and j both represent user ID. z_{uj} represents the trust statement of user u to user j. z_{uj}=1 indicates

that user u trusts user j; otherwise, z_{uj}=0.

Firstly, according to the trust auxiliary matrix, we obtain the trust user set Tru_u of each user. Then, we construct the auxiliary feature vector $\bar{t}_u$ of trust relationship in the auxiliary domain, as shown in (4.16).

$$\bar{t}_u = |Tru_u|^{-\frac{1}{2}} \sum\nolimits_{k \in Tru_u} t_k \tag{4.16}$$

where user's trust feature vector $\bar{t}_u$ is composed of the implicit vector t_k of the trust relationship, and its physical meaning is the sum of the implicit influence of the users trusted by the user u. Therefore, $\bar{t}_u$ will participate in the prediction as a part of user's characteristics of the target domain and complete the first task of the auxiliary domain in the HCTLRec framework. Combining (4.16) and (3.2), the specific equation for calculating the predicted rating $\hat{r}_{ui}$ is shown in (4.17).

$$\hat{r}_{ui} = \mu_r + b_{ur} + b_{ir} + (\rho \cdot p_u + (1-\rho) \cdot \bar{t}_u) \cdot {q_i}^T \tag{4.17}$$

where p_u and q_i respectively represent the feature vectors of the user and the item. ρ still represents the weight parameter. It adjusts the contribution of p_u and $\bar{t}_u$ in the user's characteristics.

To complete the second task of the auxiliary domain in HCTLRec-T, the prediction formula of the auxiliary domain is constructed, as shown in (4.18).

$$\hat{z}_{uj} = p_u g_j^T \tag{4.18}$$

where $\hat{z}_{uj}$ represents the predicted trust relationship and g_j represents the trustee feature vector of the user in the auxiliary domain. In addition, to ensure the accurate update of the trust auxiliary domain, the objective function of the auxiliary domain will also use social regularization terms to constrain the relationship between users, as shown in (4.19).

$$\frac{\lambda_t}{2} \sum\nolimits_u \sum\nolimits_{k \in Tru_u} ||p_u - p_k||^2 \tag{4.19}$$

where λ_t represents the regularization parameter of social regularization.

According to (4.17) and (4.18), combined with the social regularization (4.19), the objective function of HCTLRec-T is obtained as (4.20) and (4.21).

$$\begin{aligned} L_1 =& \sum\nolimits_u \sum\nolimits_{i \in I(u)} [\frac{1}{2}(r_{ui} - \hat{r}_{ui})^2 + \frac{\lambda_r}{2}(||b_{ur}||^2 + ||b_{ir}||^2 \\ &+ ||p_u||^2 + ||q_i||^2 + \sum\nolimits_{k \in Tru_u} ||t_k||^2]), \leftrightarrow p_u \end{aligned} \tag{4.20}$$

$$\begin{aligned} L_2 =& \sum\nolimits_u \sum\nolimits_{j \in Tru_u} [\frac{1}{2}(z_{uj} - \hat{z}_{uj})^2 + \frac{\lambda_a}{2}(||p_u||^2 + ||g_j||]^2 \\ &+ \frac{\lambda_t}{2} \sum\nolimits_u \sum\nolimits_{k \in Tru_u} ||p_u - p_k||^2]), \leftrightarrow p_u \end{aligned} \tag{4.21}$$

where $\leftrightarrow p_u$ means that p_u is used as a bridge for knowledge transfer. The gradient of each parameter in the objective function L_1 is shown in (4.22).

$$\begin{aligned} \frac{\partial L_1}{\partial b_{ur}} =& -\sum\nolimits_{i \in I(u)} e_{ui} + \lambda_r b_{ur} \\ \frac{\partial L_1}{\partial b_{ir}} =& -\sum\nolimits_{u \in U(i)} e_{ui} + \lambda_r b_{ir} \\ \frac{\partial L_1}{\partial p_u} =& -\sum\nolimits_{i \in I(u)} e_{ui} \rho q_i + \lambda_r p_u \\ \frac{\partial L_1}{\partial q_i} =& -\sum\nolimits_{u \in U(i)} e_{ui}(\rho \cdot p_u + (1-\rho) \cdot \bar{t}_u) + \lambda_r q_i \\ \frac{\partial L_1}{\partial t_k} =& -\sum\nolimits_{i \in I(u)} e_{ui}(1-\rho) \frac{1}{\sqrt{|Tru_u|}} q_i + \lambda_r t_k, k \in Tru_u \end{aligned} \tag{4.22}$$

The gradient of each parameter in the objective function L_2 is shown in (4.23).

$$\begin{aligned} \frac{\partial L_2}{\partial p_u} =& -\sum\nolimits_{j \in Tru_u} \tilde{e}_{uj} g_j + \lambda_a p_u + \lambda_t \sum\nolimits_{k \in Tru_u} (p_u - p_k) \\ \frac{\partial L_2}{\partial g_j} =& -\sum\nolimits_{u \in T(j)} \tilde{e}_{uj} p_u + \lambda_a g_j \end{aligned} \tag{4.23}$$

where $\tilde{e}_{uj} = z_{uj} - \hat{z}_{uj}$ represents the prediction error of the trust value in the auxiliary domain. $T(j)$ represents the set of users who trust user j.

The training of HCTLRec-T is carried out in the HCTLRec framework. The detailed calculation process is shown in Algorithm 2.

Algorithm 2 HCTLRec-T

Input: Number of outer iteration N_{out}, weight parameter ρ
Target Domain: Rating triples (user ID, item ID, rating), regularization parameter λ_r, learning parameter α_r, iteration numbers N_r
Auxiliary Domain: Trust relationship (trustee ID,trusted ID), regularization parameter λ_a, social regularization parameter λ_t, learning parameter α_a, iteration numbers N_a
Output: Target domain parameter θ_r: b_{ur},b_{ir},p_u,q_i,t_k, auxiliary domain parameter θ_a: p_u,g_j

```
    generating target matrix R, auxiliary matrix T
 2: initialization parameters
    for n_out = 1, 2, ... N_out do://outer iteration
 4:     for u = 1, 2, ... m do
            select user u's trust user set Tru_u
 6:     for n_r = 1, 2, ... N_r do://inner iteration
            compute gradients of θ_r according to (4.22)
 8:         update θ_r according to (4.14)
        for n_a = 1, 2, ... N_a do:// inner iteration
10:         compute gradients of θ_a according to (4.23)
            update θ_a according to (4.14)
12: return θ_r:b_ur,b_ir,p_u,q_i,t_k;θ_a:p_u,g_j
```

4.3 Complexity analysis In each iteration, the calculation cost is divided into the calculation of the target and auxiliary domain. The computational time of each domain contains the calculation of the loss function and

the parameter gradients. f represents the dimension of the feature vectors. m and n represent the number of users and items. k_1 represents the average number of items that users like. k_2 represents the average number of trusted users. $|R|$, $|T|$ and $|A|$ represent the numbers of observed entries in the target domain, trust auxiliary domain, and attitude auxiliary domain, respectively.

For HCTLRec-B, in the target domain, the complexity of the objective function is $O(f \cdot |R|)$. The gradient calculation complexity of the parameters b_{ur},b_{ir},p_u,q_i,v_j are $O(|R|)$, $O(|R|)$, $O(f \cdot |R|)$, $O(f \cdot |R|)$ and $O(f \cdot |R| \cdot k1)$. Therefore, the computational complexity of each iteration in the target domain is $O(f \cdot |R| \cdot (3 + k_1))$. In the auxiliary domain, the complexity of the objective function is $O(f \cdot |A|)$. The gradient calculation complexity of the parameters b_{ua},b_{ia},p_u,w_i are $O(|A|)$, $O(|A|)$, $O(f \cdot |A|)$, $O(f \cdot |A|)$, respectively. Therefore, the calculation cost of each iteration in the auxiliary domain is $O(3 \cdot f \cdot |A|)$. In HCTLRec-B, attitude data are obtained from ratings, so $|R| = |A|$. In summary, the time complexity in an iteration is $O(f \cdot |R| \cdot (6 + k_1))$. Because $k_1 \ll |R|$, the complexity of HCTLRec-B is linearly related to the number of ratings.

For HCTLRec-T, the complexity of the objective function in the target domain is $O(f \cdot |R|)$. The gradient calculation complexity of the parameters $b_{ur}, b_{ir}, p_u, q_i, t_k$ are $O(|R|), O(|R|), O(f \cdot |R|), O(f \cdot |R|)$, and $O(f \cdot |R| \cdot k_2)$, respectively. Therefore, the computational cost of each iteration in the target domain is $O(f \cdot |R| \cdot (3 + k_2))$. In the auxiliary domain, the complexity of the objective function is $O(f \cdot |T|)$. The gradient calculation complexity of the parameters p_u,g_j are $O(f \cdot |T|)$ and $O(f \cdot |T|)$, respectively. Therefore, the calculation cost of each iteration in the auxiliary domain is $O(3 \cdot f \cdot |T|)$. In summary, the time complexity in an iteration is $O(f \cdot (|R| \cdot (3 + k_2) + 3 \cdot |T|))$. Because $k_2 \ll min(|R|, |T|)$, the complexity of HCTLRec-T is linearly related to the number of ratings and trust links.

5 Experiments

5.1 Datasets and evaluations To verify the effectiveness of HCTLRec, we conduct extensive experiments by using two real-world datasets, namely, Ciao (www.ciao.co.uk) and Epinions (www.epinions.com). These two datasets which contain both ratings and trust links originate from users' real behaviors on these two online shopping websites. Each dataset contains ratings on items, which are recorded in the form of (UserID,ItemID,Rating). Each rating in the datasets adopts a five-level integer, namely $r \in \{1, 2, 3, 4, 5\}$. The specific characteristics of each dataset are shown in Table 1.

Table 1: STATISTICS OF DATASETS

Dataset	Ciao	Epinions
Users	7267	7411
Items	11211	8728
Ratings	149,147	276,116
Trust Statements	110755	52982

The auxiliary domains in the experiments include the attitude domain and trust domain. The attitude auxiliary domain is constructed based on the rating information. According to the statistics, the average rating is 4.140 in Ciao and 3.898 in Epinions. Since ratings are all recorded as integer values, the value 4 is chosen as the threshold for attitude division. If a rating exceeds 4 points, the attitude value is set to 1. If it is less than 4 points, it is set to -1. If a user's rating on an item is missing, the attitude value is set to 0. The trust auxiliary domain is directly obtained from user trust relationship in the dataset.

We conduct reasonable experiments based on five-fold cross validation. We divide the data into five parts randomly. One part (20%) is used as the test set and the remaining four parts (80%) are used as the training set each time. We do the test five times in total, and report the average result of five tests. We choose two most common evaluation metrics to measure the performance, i.e., the mean absolute error (MAE) and root mean square error (RMSE). Note that a smaller MAE and RMSE value indicate better performance.

5.2 Baseline To demonstrate the performance, we select some representative state-of-the-art models to conduct comparison experiments with HCTLRec-B and HCTLRec-T.

BiasedMF[26]: It is a kind of matrix factorization which factorizes the high-dimensional rating matrix into low-dimensional user and item feature matrices. It also adds the bias of users and items in rating prediction.

SVD++[23]: It uses implicit feedbacks to obtain implicit feature matrix and takes it as a part of users' features to constitute rating prediction based on MF.

TrustMF[27]: It constructs a user-user trust matrix based on trust information and factorizes the high-dimensional trust matrix into low-dimensional trust feature matrices. It utilizes two feature matrices about trust to further make user-item rating prediction.

TrustSVD[28]: It uses trust relationship to construct users' implicit features of trust. It adds the trust implicit features to users' features to do prediction based on SVD++.

GraphRec[29]: It is a social-based recommendation method that captures the interactions and opinions in the user-item graph and the user-user trust network.

TMF[14]: It is a transfer learning recommendation algorithm that uses users' two-level attitude information as the auxiliary domain. It uses the auxiliary information to construct users' positive and negative feature vectors for items, and uses items' feature vectors as a bridge to complete knowledge transfer.

TLRec[15]: It selects different rating domains as the auxiliary domain and target domain, and uses overlapping users and items in the two domain as bridges for knowledge transfer. It constrains the features of different domains in the form of regularization, empirical error and smoothness.

RMGM[16]: It divides the high-dimensional rating matrix into multiple sub-matrices and treats each submatrix as different domains. It utilizes the common rating features of each sub-domain to complete knowledge transfer and makes rating prediction within the domain.

DARec[12]: It is capable of extracting and transferring patterns from the rating matrix. It utilizes a domain classifier to learn shared rating patterns of the same user in different domains and transfer knowledge via adversarial training.

NeuCDCF[13]: It is a novel end-to-end neural network model based on a wide and deep framework. It extends matrix factorization ideas to the cross-domain recommendation and combines the domain-specific and domain independent embeddings to get user embeddings.

We select the optimal parameters in both datasets through cross validation. For HCTLRec-B, $\lambda_r = \lambda_a = 0.01$ and $\alpha_r = \alpha_a = 0.01$ in both datasets. We set $\rho = 0.8$ in Ciao and $\rho = 0.6$ in Epinions. For HCTLRec-T, the parameters in Ciao and Epinions are set as $\lambda_r = \lambda_a = \lambda_t = 0.01$, $\alpha_r = \alpha_a = 0.01$, and $\rho = 0.6$. For baselines, we determine parameters by consulting references and conducting cross-validation experiments.

5.3 Results (1) Effect of parameters. In the process of parameter setting, there is a very important parameter in the HCTLRec framework, i.e., the weight parameter ρ. ρ controls the weight of the two feature vectors in the target and auxiliary domain on rating prediction. A larger ρ means a greater weight of the features in the target domain.

Figure 3 and 4 show the effect of ρ on performance of HCTLRec-B and HCTLRec-T, respectively. $\rho = 0$ means that the characteristics of the target domain have no effect. The results show that errors of both algorithms become the largest at this condition. Therefore, even if the algorithms with transfer learning combine the auxiliary domain, the information of the target domain is still the most important for prediction. In particular, the recommendation performance of HCTLRec-T decreases more evidently when $\rho = 0$. The reason is

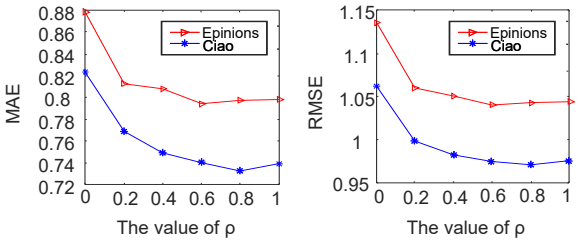


Figure 3: Impact of ρ for HCTLRec-B

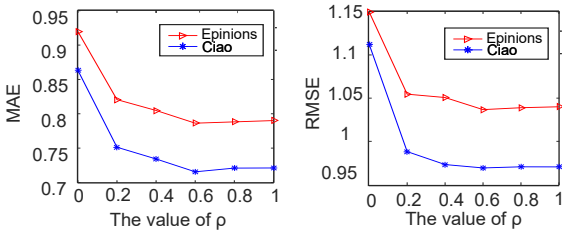


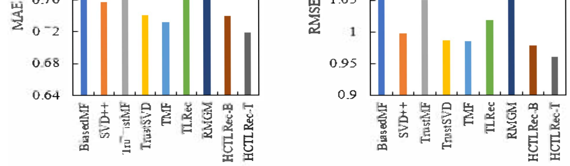
Figure 4: Impact of ρ for HCTLRec-T

that the auxiliary domain of HCTLRec-T is trust relationship, while the auxiliary domain of HCTLRec-B is attitude information. In the absence of the rating, the attitude auxiliary domain can provide more information related to users' preferences. As ρ increases, the errors of both algorithms decrease. HCTLRec-B achieves the smallest error at $\rho = 0.8$ in Ciao and $\rho = 0.6$ in Epinions, respectively. HCTLRec-T achieves the smallest error at $\rho = 0.6$ in both datasets. When ρ becomes greater than the optimal value, errors of both algorithms increase slightly, because the auxiliary domain does not exert enough influence on recommendation. This phenomenon demonstrates that only the proper combination of the target domain and auxiliary domain can obtain the best performance.

(2) Comparison with baselines. Table 2 shows the performance of HCTLRec-B and HCTLRec-T, compared with baselines in Ciao and Epinions. Observing experiment results, we draw the following conclusions:

1) All algorithms can improve recommendation performance. In particular, HCTLRec-T achieves the best performance. Compared with the classic BiasedMF, the accuracy of HCTLRec-T has a 7.9% increase in Ciao and a 9.14% increase in Epinions in terms of MAE. The result shows that using trust relationship as an auxiliary domain and performing transfer learning in the cyclic method can improve the accuracy. Compared with BaisedMF, MAE for HCTLRec-B in Ciao and Epinions decrease by 5.71% and 8.21%, respectively. Therefore, the attitude auxiliary domain is also helpful for recommendation.

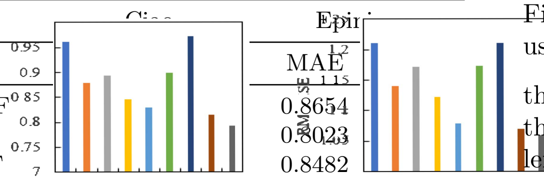
2) TMF, TLRec, RMGM, DArec, NeuCDCF, HCTLRec-B and HCTLRec-T all are algorithms with transfer learning. Except RMGM, these algorithms can



achieve good results, demonstrating the effectiveness of transfer learning in recommendation. Although RMGM uses the transfer learning, it divides ratings into the target and auxiliary domain without introducing other auxiliary information. It reduces the data density of the target domain instead, so it does not achieve better results.

3) TrustSVD and GraphRec have good recommendation performance. Therefore, trust relationship can help analyze users' preferences on items. Similarly, in the two specific algorithms of HCTLRec combined with different auxiliary domains, although both algorithms perform well, HCTLRec-T performs better than HCTLRec-B. The reason is that compared with the attitude auxiliary domain, the trust auxiliary domain can provide more user-related knowledge.

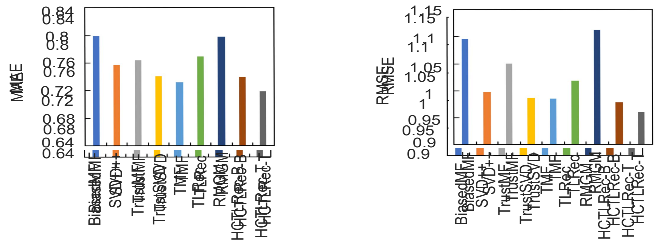
Table 2: PERFORMANCE EVALUATION OF DIFFERENT RECOMMENDATION ALGORITHMS



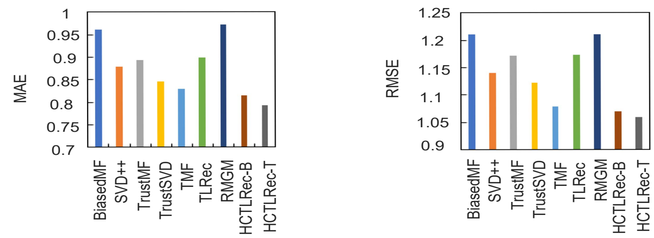
	Ciao		Epin[illegible]	
	[illegible]	[illegible]	MAE	[illegible]
BiasedMF	[illegible]	[illegible]	0.8654	[illegible]
SVD++	[illegible]	[illegible]	0.8023	[illegible]
TrustMF	[illegible]	[illegible]	0.8482	[illegible]
TrustSVD	0.7381	0.9704	0.8089	1.0428
GraphRec	0.7394	0.9851	0.8070	1.0497
TMF	0.7372	0.9727	0.7981	1.0433
TLRec	0.7550	0.9938	0.8307	1.0875
RMGM	0.7743	1.1103	0.8506	1.1217
DArec	0.7496	0.9908	0.8210	1.0912
NeuCDCF	0.7498	1.0070	0.8389	1.0724
HCTLRec-B	0.7325	0.9709	0.7943	1.0401
HCTLRec-T	**0.7155**	**0.9692**	**0.7863**	**1.0367**

(3) Cold-start experiment. One of the most serious problems affecting performance is the cold-start problem. We treat users who have rated less than 10 ratings in the datasets as cold-start users, and select several representative comparison algorithms to compare. The results are shown in Fig. 5.

For cold-start users, performance of all algorithms decreases, but it can be seen from the results that TrustSVD, TMF, HCTLRec-B and HCTLRec-T can still achieve good results. TrustSVD supplements the lack of users' information for cold-start users by incorporating trust features into users' features, while TMF, HCTLRec-B and HCTLRec-T use auxiliary domain information to supplement the knowledge of the target domain. HCTLRec-B and HCTLRec-T algorithms perform well, revealing the effectiveness of cyclic transfer learning for obtaining more knowledge from the auxiliary domain. HCTLRec-T uses trust relationship as the auxiliary domain, which can provide more adequate information related to cold-start users, so it has achieved



(a) Result in Ciao



(b) Result in Epinions

Figure 5: Results of different algorithms for cold-start users in different datasets

the best result for cold-start users. The result shows that the HCLTRec framework can alleviate the problem of cold-start users to a certain extent.

6 CONCLUSION AND FUTURE WORK

Traditional recommendation algorithms use user-item rating information to analyze users' preferences and complete the recommendation task. Combing recommendation algorithms with transfer learning is a novel method of improving the recommendation accuracy. However, some actual problems always limit the performance of transfer learning. One is that auxiliary domains and the target domain are not highly correlated. The other is that useful information in auxiliary domains cannot be fully utilized. To address these problems, this paper proposes a recommendation framework with cyclic transfer learning based on heterogeneous implicit feedbacks. Combining the framework with different auxiliary domains, two specific recommendation algorithms are proposed, that is, HCTLRec-B with the attitude auxiliary domain and HCTLRec-T with the trust auxiliary domain. We implement a two-way cyclic transfer learning method which can promote more knowledge exchange between the auxiliary domain and target domain. Finally, the two algorithms are verified using two real-world datasets with trust relationship. The results show that the use of cyclic transfer learning can improve the recommendation performance and effectively exploit the information from auxiliary domains. In future, we will focus on analyzing the relationship between different auxiliary domains and trying to combine two or more auxiliary domains in the same

algorithm, to further solve the problems of data sparsity and cold start.

7 ACKNOWLEDGMENT

This work has been supported by the National Natural Science Foundation of China under Grant 61872033, the Fundamental Research Funds for the Central Universities (2021YJS015), the Beijing Nova Program from Beijing Municipal Science Technology Commission under Grant Z201100006820015, the Humanity and Social Science Youth Foundation of Ministry of Education of China under Grant 18YJCZH204.

References

[1] M. J. Eppler and J. Mengis, *The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines*, Inf. Soc., 5 (2004), pp. 325–344.

[2] P. M. Alamdari, N. J. Navimipour, M. Hosseinzadeh, A. A. Safaei and A. Darwesh, *A systematic study on the recommender systems in the E-commerce*, IEEE Access, 8 (2020), pp. 115694-115716.

[3] H. Zhang, I. Ganchev, N. S. Nikolov and M. Stevenson, *UserReg: A Simple but Strong Model for Rating Prediction*, In ICASSP, 2021, pp. 3595–3599.

[4] H. Wang, Z. Wang and W. Zhang, *A node addition model for symbolic factorization*, In ICCCBDA, 2018, pp. 78-82.

[5] G. Guo, H. Qiu, Z. Tan, Y. Liu, J. Ma and X. Wang, *Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems*, Knowledge-Based Syst., 138 (2017), pp. 202-207.

[6] Z. Zhang, M. Dong, K. Ota and Y. Kudo, *Alleviating new user cold-start in user-based collaborative filtering via bipartite network*, IEEE Trans. Comput. Soc. Syst., 3 (2020), pp. 672-685.

[7] L. H. Son, *Dealing with the new user cold-start problem in recommender systems: A comparative review*, Inf. Syst., 58 (2016), pp. 87-104.

[8] Z. D. Zhao and M. S. Shang, *User-based collaborative-filtering recommendation algorithms on hadoop*, in KDD, 2010, pp. 478-481.

[9] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, *Item-based collaborative filtering recommendation algorithms*, in WWW, 2001, pp. 285-295.

[10] T. Do Noia, R. Mirizzi, V. C. Ostuni and D. Romito, *Exploiting the web of data in model-based recommender systems*, in RecSys, 2012, pp. 253-256.

[11] S. Pyo and E. Kim, *LDA-Based Unified Topic Modeling for Similar TV User Grouping and TV Program Recommendation*, IEEE Trans. Cybern., 8 (2014), pp. 1476-1490.

[12] F. Yuan, L. Yao and B. Benatallah, *DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patternso*, in IJCAI, 2019, pp. 1476-1490.

[13] M. Vijaikumar, S. Shevade and M. N. Murty, *Neural Cross-Domain Collaborative Filtering with Shared Entities*, in ECML PKDD, 2020, pp. 729-745.

[14] W. Pan, S. Xia, Z. Liu, X. Peng and Z. Ming, *Mixed factorization for collaborative recommendation with heterogeneous explicit feedbacks*, Inf. Sci., 332 (2016), pp. 84-93.

[15] L. Chen, J. Zheng, M. Gao, A. Zhou, W. Zeng and H. Chen, *TLRec: transfer learning for cross-domain recommendation*, In ICBK, 2017, pp. 167-172.

[16] M. Li and L. Wen, *An Improved Matrix Approximation for Recommender Systems Based on Context-Information and Transfer Learning*, IEEE Access, 7 (2019), pp. 124256-124267.

[17] H. Liang, W. Fu and F. Yi, *A survey of recent advances in transfer learning*, in ICCT, 8 (2019), pp. 1516-1523.

[18] W. Pan, Q. Yang ,Y. Duan and Z. Ming, *Transfer learning for semisupervised collaborative recommendation*, in TiiS, 2 (2016), pp. 1-21.

[19] W. Pan, *A survey of transfer learning for collaborative recommendation with auxiliary data*, Neurocomputing, 177 (2016), pp. 447-453.

[20] H. Xie, B. Liu and Y. Xiao, *Transfer Learning-Based One-Class Dictionary Learning for Recommendation Data Stream*, Inf. Sci., 547 (2021), pp. 526-538.

[21] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu and S. Yang, *Social recommendation with cross-domain transferable knowledge*, IEEE Trans. Cybern., 11 (2015), pp. 3084-3097.

[22] S. Jiang, Z. Ding and Y. Fu, *Heterogeneous Recommendation via Deep Low-rank Sparse Collective Factorization*, IEEE Trans. Pattern Anal. Mach. Intell., 5 (2019), pp. 1097-1111.

[23] Y. Koren, *Factorization meets the neighborhood: a multifaceted collaborative filtering model*, In SIGKDD, 8 (2014), pp. 426-434.

[24] C. Park, D. Kim, J. Oh and H. Yu , *Improving top-K recommendation with truster and trustee relationship in user trust network*, Inf. Sci., 2016, pp. 100-114.

[25] A. Javari and M. Jalili, *A probabilistic model to resolve diversity–accuracy challenge of recommendation systems*, IEEE Trans. Cybern., 3 (2015), pp. 609-627.

[26] Y. Koren, R. Bell and C. Volinsky, *Matrix factorization techniques for recommender systems*, Computer, 8 (2009), pp. 30-37.

[27] B. Yang, Y. Lei, J. Liu and W. Li, *Social collaborative filtering by trust*, IEEE Trans. Pattern Anal. Mach. Intell., 8 (2016), pp. 633-1647.

[28] G. Guo, J. Zhang and N. Yorke-Smith, *A novel recommendation model regularized with user trust and item ratings*, IEEE Trans. Knowl. Data Eng., 7 (2016), pp. 1607-1620.

[29] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang and D. Yin, *Graph Neural Networks for Social Recommendation*, in WWW, 2019, pp. 417-426.