Knowledge-Based Systems 220 (2021) 106948

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Deep sparse autoencoder prediction model based on adversarial learning for cross-domain recommendations

Yakun Li [a,b], Jiadong Ren [a,b,*], Jiaomin Liu [a], Yixin Chang [a]

[a] College of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, China
[b] The Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province, Qinhuangdao, Hebei, China

ARTICLE INFO

Article history:
Received 7 July 2020
Received in revised form 3 February 2021
Accepted 9 March 2021
Available online 13 March 2021

Keywords:
Data sparsity
Sparse autoencoder
Adversarial learning
Recommender systems
Matrix factorization

ABSTRACT

Online recommender systems generally suffer from severe data sparsity problems, and this are particularly prevalent in newly launched systems that do not have sufficient amounts of data. Cross-domain recommendations can provide us with some new ideas for assisting with user recommendations in sparse target domains by transferring knowledge from a source domain with rich data. In this paper, a deep sparse autoencoder prediction model based on adversarial learning for cross-domain recommendations (DSAP-AL) is proposed to improve the accuracy of rating predictions in similar cross-domain recommender systems. Specifically, joint matrix factorization and adversarial network learning models are adopted to integrate and align user and item latent factor spaces in a unified pattern. Then, a deep sparse autoencoder is represented and modeled by transferring the latent factors and interlayer weights. Furthermore, a domain factor adaptation algorithm is proposed to capture robust user and item factors, and the learned regularization constraints are added to the objective function, thereby alleviating the data sparsity issue. Experimental results on four real-world datasets demonstrate that, even without overlapping entities (users or items) in the source and target domains, the proposed DSAP-AL method achieves competitive performance relative to other state-of-the-art individual and cross domain approaches. Moreover, the DSAP-AL model is not only effective for scenarios with sparse data but also robust for noise-containing recommendations.

1. Introduction

Recommender systems are rooted in our daily lives and provide us with a variety of services in many scenarios, such as product recommendations in Taobao, book recommendations in Dangdang, music recommendations in Last•fm, friend recommendations in Twitter, etc. These services meet the needs of users by providing personalized recommendations [1–3]. Matrix factorization-based collaborative filtering (MFCF) [4] and content-based recommendations [5] are very popular techniques applied in real recommender systems. The former generates recommendations by extracting the latent features of users and items after performing matrix decomposition, and the latter analyzes the attributes of users or items to model user preferences for the purpose of performing recommendations. However, data sparsity seriously hinders the development of recommender systems [6], since most users only rate a small number of items they prefer. To solve this stubborn problem, cross-domain recommendation (CDR) systems have shown great learning potential, and some related studies have been proposed [7]. CDR mainly uses the rich knowledge present in the source domain to improve the accuracy of the item rating predictions in the target domain.

Many existing CDR systems assume that users or products partially or completely overlap. For example, P. Cremonesi and M. Quadrana [8] divided cross-domain scenarios into four categories, as shown in Fig. 1. A. Stewart et al. [9] proposed building social links from cross-domain tagging systems based on partially overlapping users. Overlapping attributes among the cross-domain products were calculated to predict missing ratings [10]. The literature [11] has explored the transfer of rating knowledge by using overlapping users and items in the auxiliary and target domains. However, in real industrial applications, these situations are very rare, and in most recommender systems, neither users nor products overlap (see Fig. 1(a)). How to establish interdomain associations without overlapping entities is the key to cross-domain knowledge transfer for providing recommendations. In addition, to the best of our knowledge, the existing works on cross-domain recommender systems rarely explore the latent features of users or items implicit in rating patterns.

To solve these problems, transfer learning based on matrix factorization has attracted increasing attention. Transfer learning uses rich auxiliary domain data to relieve data sparsity and

* Corresponding author at: College of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei, China.
E-mail address: jdren@ysu.edu.cn (J. Ren).

https://doi.org/10.1016/j.knosys.2021.106948
0950-7051/

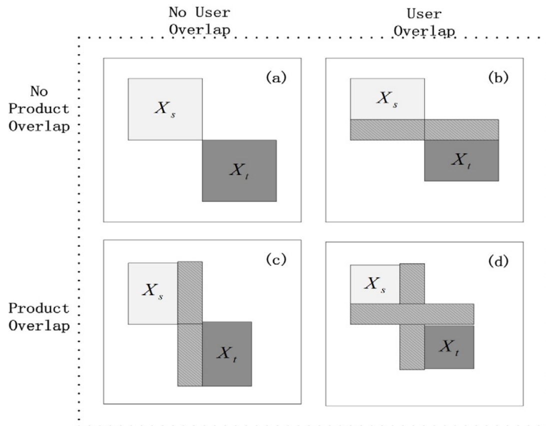


Fig. 1. Various scenarios of overlapping entities. (a) No overlap. (b) User overlapping. (c) Product overlapping. (d) User and product overlapping.

improve the accuracy of rating predictions in the target domain. Although the transfer of user–item interaction ratings or item semantic attributes has solved the cross-domain heterogeneity problem to some extent [12,13], the premise of explicit entity connections has restricted the learning efficiency and prediction performances of such recommendation systems. In view of the above issues, latent factor models that map users and items into the same latent feature spaces were adopted and utilized to train factor features on observed ratings in [14]. Especially in newly launched systems, latent factor techniques based on transfer learning can significantly improve recommendation performance [15]. In addition, in deep learning models, how to use knowledge and achieve knowledge transfer is crucial for generating cross-domain recommendations.

Deep learning is widely used in the field of feature learning and prediction [16]. In particular, some nonlinear learning models can efficiently predict the missing ratings of entities. Currently, cross-domain recommendations face many challenges. They not only need to transfer cross-domain rating data but also often need to deal with a large amount of heterogeneous auxiliary information. Traditional methods have difficulty solving these problems effectively. Therefore, deep learning can be applied to cross-domain recommendations to improve the quality of the corresponding rating predictions.

In this paper, we present a deep sparse autoencoder learning model based on adversarial learning for cross-domain recommender systems. First, we jointly decompose the rating matrices in the auxiliary and target domains to obtain the latent factors of users and items respectively, and then they can be trained by a generative adversarial network model to acquire aligned user and item factor spaces in an integrated pattern. This step is the prerequisite for solving our problem and generating predictions. Second, a sparse autoencoder is extended to a deep neural network model, and then a sparse regularization constraint is added to the network training process. As a result, a deep sparse autoencoder model is achieved by transferring the interlayer weights and entity factors. Then, to reduce the influence of abnormal factors on the recommendation results, a domain factor adaptation algorithm is proposed to filter out factor noise and obtain robust user and item factor spaces. Finally, factor regularization constraints are added to the cross-domain model training process. The main contributions of our paper are as follows.

(1) A generative adversarial network approach that aligns user and item latent factors after joint matrix factorization. This method trains and unifies the factor features of each entity acquired from the auxiliary and target domains to ensure the consistency of entity representation across domains. The feature divergence caused by domain heterogeneity is eliminated.

(2) A deep sparse autoencoder learning method for establishing a neural network training model in heterogeneous scenarios. To the best of our knowledge, this is the first work to explore a deep sparse autoencoder model for improving rating prediction accuracy in cross-domain scenarios.

(3) A domain factor adaptation algorithm for eliminating abnormal factors that deviate from the normal range. This filtering algorithm ensures that highly useful knowledge is used for network training. Extensive experiments are performed on four datasets, and the results also show that the DSAP-AL can alleviate the data sparsity issue and improve the prediction accuracy of cross-domain recommender systems.

The remainder of this paper is organized as follows. The next section discusses the related work. The problem definition and model background are presented in Section 3. The details of our proposed DSAP-AL model are introduced in Section 4. Finally, a series of experiments are conducted in Section 5, and Section 6 outlines the conclusions and future research directions.

2. Related work

In this section, we first introduce deep learning recommendations, and then outline the existing cross-domain recommender systems, which are further divided into cross-domain recommendation systems based on knowledge aggregation and those based on knowledge transfer.

2.1. Deep learning recommendations

Most current deep learning recommendations focus on collaborative filtering models based on matrix factorization to improve the efficiency of recommender systems. Manotumruksa [17] proposed a deep collaborative filtering approach to alleviate data sparsity and improve rating prediction accuracy. W. Zhang et al. [18] adopted a convolutional neural network model based on "label weight nearest neighbors" for item recommendations. H. Guo et al. [19] proposed the DeepFM algorithm, which incorporated both the ratings and content features of users and items into recommender systems to alleviate the data sparsity problem. Furthermore, some deep recurrent network models [20,21] have also been exploited for collaborative recommendations.

An autoencoder is an unsupervised deep learning network that reconstructs original samples or extracts features by constantly adjusting its parameters. Many researchers have integrated deep autoencoder models into other application methods. J. Yiet al. [22] presented an adversarial learning autoencoder to improve the quality of rating predictions. Deep autoencoders were proposed to overcome the natural sparseness of collaborative filtering [23]. L. Sascha et al. [24] constructed a deep autoencoder-based neural network model to learn the feature spaces of entities. The transferability of autoencoder-based learning with respect to entity detection and prediction was also studied [25,26]. However, these works only focused on single-domain recommendations, and the network training structures were too simple to adjust the weights and input data effectively.

Recently, F. Zhuang et al. [27] proposed a novel representation learning model via dual-autoencoder recommendations. In this model, the hidden representations of users and items are learned by using autoencoders and gradient descent training. Recommendation systems based on stacked denoising autoencoders have also been studied to learn increasingly effective rating representations for tag recommendations [28]. With the development of deep autoencoder models, more complex studies can also be found [29–31]. Instead of simply establishing deep training frameworks to improve the prediction performance of a single-domain system, our DSAP-AL learning model can work on cross-domain scenarios by using aligned user and item latent factors as bridges for knowledge transfer.

2.2. Cross-domain recommender systems

2.2.1. Cross-domain recommender systems with knowledge aggregation

In general, cross-domain recommendations based on knowledge aggregation gather information from different source domains and then generate the final recommendations in the target domain. This type of recommendation method can be built at the group-level or cluster-level, both of which are related to sharing latent entity features or rating patterns.

Y. Chen et al. [32] presented an effective heterogeneous recommendation framework, where user preferences from different platform domains were aggregated to learn the potential interaction features of user and item entities. A. M. Elkahky et al. [33] exploited cross-domain user modeling data (user similarity, user neighbors) to jointly learn item features and extended the system to analyze user behavior patterns for the purpose of improving its recommendation quality. The results also proved that combining features derived from multiple domains can yield much better performance than building individual models for each domain. Furthermore, another way to aggregate knowledge is to summarize the recommendation results for each domain when both users and items overlap across domains. For example, the observed user rating vectors in [34] were weighted to calculate the predicted ratings of items in the source domain separately, and then aggregated for user recommendations in the target domain. However, the aforementioned prerequisites for overlapping user or item entities are not common due to cross-domain heterogeneity. Therefore, F. Zhuang et al. [35] proposed a cross-domain consistent regularized learning method, whose main contribution was to model the rating data in the source domain as multiple classifiers and then use the knowledge learned in the target domain to coordinate the differences between these classifiers. The advantage of this recommendation approach is that the users or items in multiple domains do not need to overlap.

In addition, some other knowledge aggregation methods [36–40] have been proposed to improve the prediction accuracy of cross-domain recommender systems, and these are closely related to our research. However, these approaches still cannot solve the data sparsity issue effectively, and some models need to optimize too many parameters, resulting in low-efficiency recommendation.

2.2.2. Cross-domain recommender systems with knowledge transfer

Transfer-based cross-domain recommendations make full use of the explicit or implicit associations between cross-domain data to transfer the available knowledge in the auxiliary domain to the target domain for easing cold conditions and improving the accuracy of recommendation systems. Compared to knowledge aggregation techniques, this type of recommendation has attracted much attention and made great progress in terms of missing rating predictions.

With regard to the topic modeling of semantically clustered vocabularies in different domains, A. Kumar et al. [41] designed a latent semantic space, where vocabularies with similar semantic relationships across domains were used and transferred to capture user preferences and interests in the target domain. The popular WordNet model was utilized to evaluate the semantic associations among textual words. The Bayesian hierarchical method based on latent Dirichlet allocation was proposed by S. Tan et al. [42] to transfer user preferences from the auxiliary domain to the target domain. In the proposed approach, cross-domain multitype media information (textual data, media descriptions and observed ratings) was learned to profile topic distributions for documents and user interests for predictive ratings. In addition to transferring the latent features of users or items, a multiple incomplete domain transfer learning model [43] was presented to compress user–item rating information into multiple compact cluster-level matrices. The target matrix was then reconstructed to effectively learn the interactive rating knowledge from multiple incomplete domains, and the final recommendation was generated. In addition, some recent studies [44–46] have integrated both ratings and side information into transfer learning across domains to achieve rating predictions.

With recent advances in transfer techniques, some emerging approaches use rating patterns as bridges for connecting different domains and then achieve knowledge transfer and cross-domain recommendations in an autonomous or common way. F. Yuan et al. [47] proposed a deep domain adaptation model, where rating patterns from the source domain were extracted and transferred to the target domain for top-N recommendation tasks. Wang et al. [48] transferred the rating representations from the auxiliary domain to the target domain in an adversarial way. With a generative adversarial network model, a domain discriminator is generated by learning a rating mapping function for cross-domain recommendations.

B. Li et al. [49] took the average rating matrix in the source domain as a codebook for applying matrix tri-factorization and realized rating transfer in an autonomous way. In a more recent study, Liu and Wu et al. [50] transferred interdomain rating knowledge nonlinearly by adopting the super-structure sharing technique. However, few existing transfer approaches can leverage deep encoding models to transfer the latent factor features of user or item entities for cross-domain recommender systems. In our paper, we propose the DSAP-AL learning model to fill in the gaps in cross-domain recommendation research. In addition, the knowledge transfer process in our paper mainly means that the latent feature factors in the source and target domains are transferred to integrated factor spaces through transfer matrix factorization across domains. The integrated latent factor spaces that eliminate domain heterogeneity provide the foundation and premise for building deep sparse autoencoder models and training neural networks.

3. Problem definition and model background

In a sparse target domain, assume that there are M user entities and N item entities, and the user interaction ratings for items are denoted as $\mathrm{X}_t \in R^{M\times N}$. The observed rating knowledge reflects user preferences for items, and the rating range in a general recommendation system is a set of positive integers from 1–5. Similarly, in a dense source domain, the user rating matrix $\mathrm{X}_s \in R^{M\times N}$ has an analogous entity representation. In real industrial applications, single-domain recommender systems are insufficient for meeting the needs of user recommendations, especially when the rating knowledge in the system is very sparse. This motivated us to study how to use dense data in the auxiliary domain to assist with recommending products to users in the target domain. Thus, our task is to predict the missing rating values in the target domain from the observed rating knowledge in X_t and X_s. In other words, the problem to be solved in this paper is to predict the missing rating values in the rating matrix of the target domain based on the known rating matrices in the source and target domains.

In a single domain, user preferences are modeled by probabilistic matrix factorization (PMF) through two latent feature factors [4].

$$X = UV^T \tag{1}$$

where the rating matrix X is decomposed into two low-rank feature matrices, namely a user latent feature matrix $U \in R^{M\times D}$ and an item latent feature matrix $V \in R^{N\times D}$. Accordingly, a given user u is associated with a d-dimensional vector U, which represents user u's preferences for all items. Each item i is associated with a d-dimensional vector V, which represents the degree to which the item possesses those preference features. Regularization terms for users and items are often added to the model optimization process to avoid overfitting [14].

$$J(U,V) = \frac{1}{2}\|I \circ (X - UV^T)\|_F + \frac{\delta}{2}\|U\|_F + \frac{\delta}{2}\|V\|_F \tag{2}$$

where J is the objective function of the predicted ratings, I is the identity matrix indicating whether the rating is known, $\circ$ is the Hadamard product between matrices, and δ is the model learning parameter. However, the model is also deeply trapped in the inherent defects of recommender systems and cannot directly solve the cross-domain recommendation problem. In our proposed model, the abovementioned single-domain optimization model is improved and extended to enhance the quality of cross-domain rating predictions.

4. The proposed learning model

In this section, our proposed DSAP-AL learning model is presented in detail. In cross-domain recommendations, our framework goes beyond the limitation that there must be overlapping entities or some explicit knowledge connections. This section begins with an overview of the overall DSAP-AL approach, followed by a detailed explanation of the four model steps.

4.1. Overview of the proposed model

To achieve cross-domain knowledge sharing and build deep loop encoders, sparse constraints are added to the joint matrix factorization process to obtain integrated user and item factor spaces. In the integrated factor patterns, regardless of whether there are overlapping entities (users or items) across domains, knowledge transfer can be achieved in our proposed deep learning model. Abnormal factors are eliminated for improved neural network training through the proposed robust domain factor adaptation algorithm. Furthermore, regularization constraints on the integrated user and item factors can benefit the model to obtain improved rating prediction results.

Specifically, the DSAP-AL method is composed of the following four major steps, as shown in Fig. 2.

Step 1: Joint matrix factorization, integrate and align the latent factor spaces through the proposed generative adversarial network model.

Step 2: A deep sparse autoencoder model is established to transfer the factor knowledge and weights using cross-domain representations.

Step 3: Abnormal noise factors are detected and eliminated from the factor spaces through the proposed domain factor adaptation algorithm, and robust user and item factors are output.

Step 4: Regularization constraints are added to the optimization model for algorithm learning, and the final recommendation results are generated.

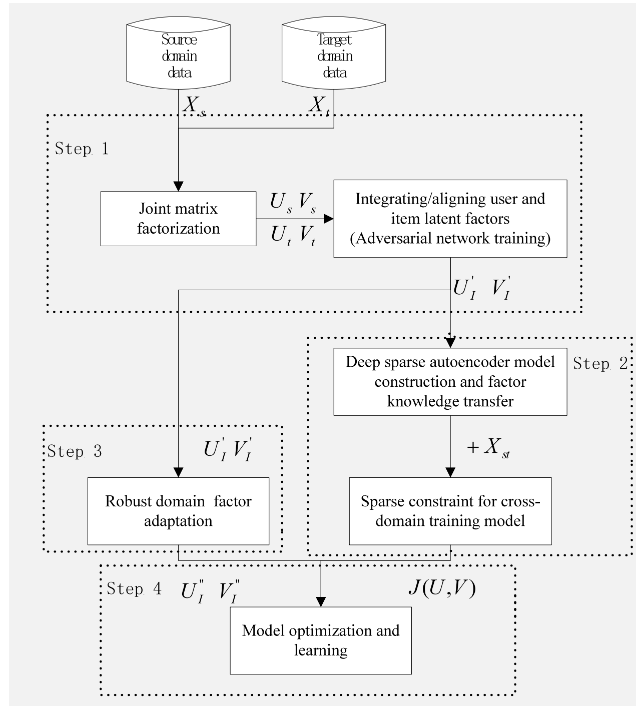


Fig. 2. The proposed DSAP-AL learning framework.

4.2. Proposed cross-domain recommendation learning model

4.2.1. Joint matrix factorization and latent factor space alignment

Sparse target domain recommender systems usually have only a small amount of available data, and this leads to inefficient recommendation results. Based on this situation, similar source domain systems with rich data can be utilized for knowledge transfer to assist the system with providing user recommendations in the target domain. In addition, due to the domain heterogeneity among different recommender systems, a transfer matrix factorization across domains [11] can be exploited to build interdomain "bridges" and transfer connections in our proposed model. Therefore, the source rating matrix $X_s \in R^{A\times B}$ and the target rating matrix $X_t \in R^{C\times D}$ are decomposed jointly and aligned into two integrated user and item latent factor spaces, namely $U_I \in R^{P\times K}(P \geq A + C)$ and $V_I \in R^{Q\times K}(Q \geq B + D)$.

Specifically, to align latent factors across domains, a generative adversarial network [51] is utilized to adjust the interdomain factors. In our paper, the generators G_i^j are defined as different types of generators with $j \in \{s, t\}$ and $i \in \{u, v\}$.

$$G_i^j(U, V; \Theta_\tau) \tag{3}$$

where i is the user or item latent factor after matrix factorization, j represents different domains (source or target), and Θ_τ is a learnable parameter.

If the joint matrix factorization across domains satisfies the following equations, the user and item latent factors are mapped into the corresponding integrated factor spaces.

$$P(U_I) = P(U_s) \cdot P(U_t) \tag{4}$$

$$P(V_I) = P(V_s) \cdot P(V_t) \tag{5}$$

where $P(U_s), P(U_t), P(U_I)$ represent the marginal probability distributions of U_s, U_t, U_I. The transfer representation of the item latent factors is similar to that of the user factors above. Then, the individual user and item latent factors in the source and target domains are aligned in the integrated factor spaces. However, it is very difficult to solve the abovementioned probabilistic optimization problem. To this end, an adversarial network model is proposed to build a mapping function.

$$P(G(U_I)) = P(G_u^s) \cdot P(G_u^t) \tag{6}$$

$$P(G(V_I)) = P(G_v^s) \cdot P(G_v^t) \tag{7}$$

To learn adversarial representations, long short term memory (LSTM) is used to train a recurrent neural network.

$$f_t = \sigma(W_f \cdot (h_{t-1}, x_t) + b_f)$$

$$i_t = \sigma(W_i \cdot (h_{t-1}, x_t) + b_i)$$

$$\bar{c}_t = \tanh(W_c \cdot (h_{t-1}, x_t) + b_c)$$

$$c_t = f_t * c_{t-1} + i_t * \bar{c}_t$$

$$o_t = \sigma(W_o \cdot (h_{t-1}, x_t) + b_o)$$

$$h_t = o_t * \tanh(c_t) \tag{8}$$

where c_t is the memory cell and f_t, i_t and o_t are the forget, input and output gates, respectively.

According to the standard GAN model [48], U is the real user latent factor after matrix decomposition, and $G(U)$ is the generated user latent factor. Therefore, the loss function of the adversarial network model is defined to eliminate interdomain heterogeneity.

$$\begin{array}{l} \max l_a(U_s, U_t, G_u^s, G_u^t) \\ \min l_i(U, G(U)) \end{array} \tag{9}$$

For the alignment of the item latent factors, the training process is the same as that of the user. Fig. 3 shows the proposed end-to-end adversarial network model for aligning user and item latent factors and the specific network structure of the adversarial training process. Different from the scenarios in the literature [48], Fig. 3(a) mainly consists of three stages. First, the rating matrices in the source and target domains are decomposed into latent feature factors. Then, the factor generator and discriminator are defined based on the obtained latent factors. Finally, the aligned user and item latent factors are obtained through the game function of the adversarial network. In Fig. 3(b), the proposed adversarial network is mainly divided into factor embedding layer and adversarial layer. The embedding layer is composed of the latent feature factors obtained after joint matrix decomposition and factor samples. The adversarial layer mainly includes a factor generator and discriminator.

Finally, the user and item latent factors in individual domains are aligned into integrated user latent factors U_I' and item latent factors V_I', respectively.

The algorithm process of aligning latent factor spaces (ALFS) is as follows.

4.2.2. Domain factor transfer learning via sparse autoencoder

The main application of a recommender system is to predict missing rating values in the target rating matrix. An autoencoder is a kind of neural network that can reconstruct the input sample and express its features [52]. Therefore, latent factor features and network weights are transferred and learned through the cross-domain representations of sparse autoencoder.

For a latent factor x in the user or item factor space, the activation h for the hidden layer nodes is calculated as follows.

$$h = f(Wx + b) \tag{10}$$

where W is the connection weight between layers, b is the interlayer deviation, and f is the activation function. To effectively reconstruct the training data, the PReLU function is adopted as our activation function for training the neural network [53].

$$PReLU(x_i) = \begin{cases} x_i & if\ (x_i > 0) \\ a_i x_i & if\ (x_i \le 0) \end{cases} \tag{11}$$

In the proposed model, the PReLU activation function is an improvement over the commonly used sigmoid function. This is because the former can save considerable computations when calculating the error gradient via back propagation. Moreover, for deep training networks, the PReLU function can effectively solve the problem that the gradient easily disappears, and it additionally alleviates the overfitting phenomenon.

Deep learning and knowledge transfer have demonstrated their efficiency and effectiveness in recommendation applications [54,55]. The single-layer SAE network cannot improve the prediction accuracy of cross-domain recommendations. Hence, we expand and deepen the structure of the SAE, take the weights and activation functions of the first layer as the inputs of the next layer, and continuously transfer the interlayer weights and deviations. Then, the representation of the weight transfer process is as follows.

$$\begin{cases} W_{i+1} = W_i \\ b_{i+1} = b_i \end{cases} \quad i = 3, 4, \ldots, n; \tag{12}$$

where n is the number of middle layers of the deep neural network.

Generally, too few cells in the middle layer of the neural network leads to difficulty in reconstructing the input sample, while too many cells leads to unit redundancy and reduced efficiency during data compression. To solve this problem, a sparse regularization constraint is introduced into the training model of deep autoencoders.

$$J_{sparse}(\tilde{x}_n; W, b) = \sum_{n=1}^{N} \|x_n - \tilde{x}_n\|^2 + \beta \sum KL(\rho \parallel \hat{\rho}_j) \tag{13}$$

where $\hat{\rho}_j$ is the average activation of the jth node in a hidden layer, β is a sparse parameter that can adjust and control the sparse penalty and $KL(\rho \parallel \hat{\rho}_j)$ is the Kullback–Leibler divergence [56], which represents the difference between the calculated average activation and the target value.

$$\hat{\rho}_j = \frac{1}{N} \sum_{n=1}^{N} f(W_j x_n + b_j) \tag{14}$$

$$KL(\rho \parallel \hat{\rho}_j) = \rho \log(\frac{\rho}{\hat{\rho}_j}) + (1-\rho) \log(\frac{1-\rho}{1-\hat{\rho}_j}) \tag{15}$$

To learn the joint features of cross-domain networks,we modularize the source rating matrix $X_s \in R^{A \times B}$ and the target rating matrix $X_t \in R^{C \times D}$ into a new true rating matrix $X_{st} = \begin{bmatrix} X_s & 0 \\ 0 & X_t \end{bmatrix} \in R^{(A+C)\times(B+D)}$. Finally, the cost function for deep autoencoders with sparse constraints is rewritten as shown below.

$$J(U_I, V_I; \delta, \beta) = \frac{1}{2} \|I \circ (X_{st} - U_I V_I^T)\|_F^2 + \frac{\delta}{2} \|U_I\|_F^2 + \frac{\delta}{2} \|V_I\|_F^2 + \beta KL(\rho \parallel \hat{\rho}_j)\ s.t. U_I^T V_I = I \tag{16}$$

4.2.3. Robust domain factor adaptation

Local abnormal factors are abnormal data that deviate from most other data. If the local abnormal coefficients of the latent factors in the sample are greater than a certain threshold, then they are defined as abnormal factors. During the cross-domain recommendation process, it is possible that a few outliers might have a sufficiently large and negative impact on the recommendation results. To avoid allowing abnormal factor information

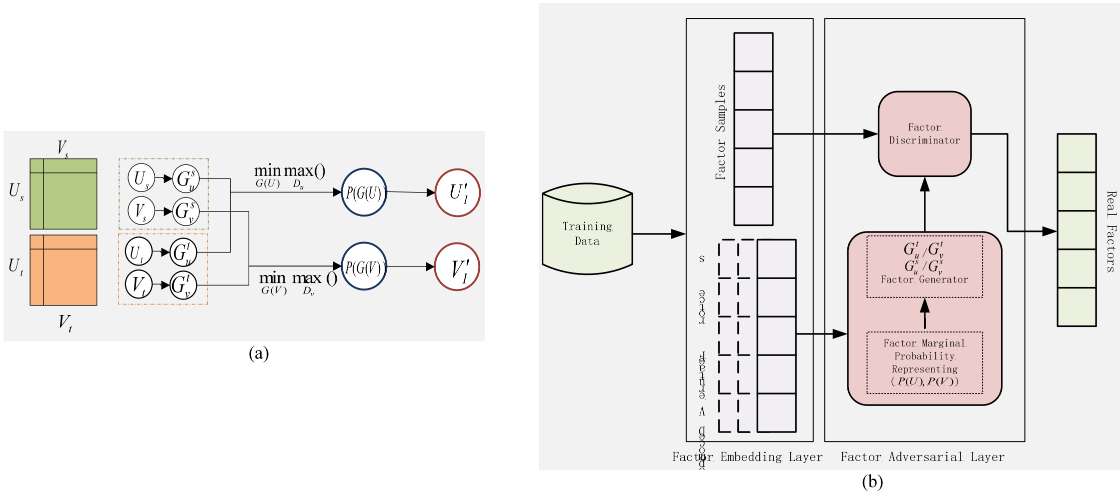


Fig. 3. (a) The proposed end-to-end adversarial network model for aligning user and item latent factors. (b) The specific network structure of the adversarial training.

Algorithm 1 ALFS

Inputs: Source rating matrix X_s, Target rating matrix X_t

Output: Integrated user latent factor U_I' and itemlatent factor V_I'

1: Initialize integrated user latent factor U_I' and itemlatent factor V_I'

2: Decompose X_s and obtain individual user latent factor U_s and item latent factor V_s in the source domain.

3: Decompose X_t and obtain individual user latent factor U_t and item latent factor V_t in the target domain.

4: Represent adversarial generator $G_i^j(U, V; \Theta_\tau)$

5: Construct an adversarial representation mapping function by using marginal probability distribution of latent factors

$$P(G(U_I)) = P(G_u^s) \cdot P(G_u^t)$$

$$P(G(V_I)) = P(G_v^s) \cdot P(G_v^t)$$

6: Train adversarial networks by extracting factor adversarial representation

7: Align user latent factor U_I' and itemlatent factor V_I' by Game function $\min\max()$

8: **Return** U_I' and V_I'

---

to control the recommendations, robust domain factor adaptation is proposed to further improve the prediction accuracy of recommendations. In other words, to accurately represent user preferences, noise or abnormal factors in the sample set are detected and eliminated in this subsection. To this end, a robust domain factor adaptation (RDFA) algorithm is proposed to obtain robust user or item latent factor spaces.

Specifically, the reachability distance (RD) from a user or item latent factor variable x to x' is first defined in the cross-domain representation as follows.

$$RD_k(x, x') = \max(\|x - x^{(k)}\|, \|x - x'\|) \tag{17}$$

where $x^{(k)}$ is the kth closest sample to x in the training factorset $\{x_i\}_{i=1}^n$.

According to the reachability distances among the latent factors, local reachability density (LRD) of x is defined by the following formula.

$$LRD_k(x) = (\frac{1}{k}\sum_{i=1}^{k} RD_k(x^{(i)}, x))^{-1} \tag{18}$$

The local reachability density of x is the inverse of the average reachability distance from $x^{(i)}$ to x. When the density of the factor training sample of x is high, the value of its local reachability density is also large.

With the local reachability density, the local abnormal coefficient (LAC) of x is defined in a cross-domain recommender system as shown below.

$$LAC_k(x) = \frac{\frac{1}{k}\sum_{i=1}^{k} LRD_k(x^{(i)})}{LRD_k(x)} \tag{19}$$

where $LAC_k(x)$ is the ratio of the average local reachability density of $x^{(i)}$ to the local reachability density of x. The larger the value of $LAC_k(x)$ is, the larger the abnormality of x. When the density around $x^{(i)}$ is high and the density around x is low, the value of $LAC_k(x)$ is relatively large, and x is regarded as an abnormal factor. In contrast, x is considered a normal value for the next cross-domain training iterations.

Then, a robust candidate set of latent factors RCS_ε for cross-domain network training is defined in our proposed model.

$$RCS_\varepsilon = \{x | LAC_k(x) < \varepsilon, x \in X\} \tag{20}$$

where ε is an abnormal factor threshold and X represents the user or item latent factor space.

Robust Domain Factor Adaptation (RDFA) is summarized in Algorithm 2.

Algorithm 2 RDFA

Inputs: User or item latent factors X and factor abnormal threshold ε
Output: Robust latent factors X'
1: **Initialize** X';
2: computeReachabilityDistance($RD_k(x,x')$)
3: computeLocalReachabilityDensity(x)
4: Obtain local outlier factor $LAC_k(x)$
5: **for** a latent factor x in the factor space X
6: **if** $(LAC_k(x) < \varepsilon)$
7: $X' \leftarrow x$
8: **end if**
9: **end for**
10: **Return** X'

4.2.4. Model optimization and algorithm learning

After the abnormal factors greater than a threshold ε are eliminated, the robust user latent factor U_I'' and item latent factor V_I'' are obtained and used for the training of the objective function. In addition, in the scenario of our model, one of the most attractive advantages is that users or items in different domains do not need to overlap. This provides a highly feasible and valuable solution for the rating predictions of cross-domain recommender systems. Specifically, we decompose the rating matrices in the source and target domains into user and item latent factor spaces through joint matrix factorization. Then, by introducing the probability distribution of latent factors and an adversarial learning model, the user or item latent factors in different domains are transferred and aligned to the integrated user or item factor spaces, and interdomain knowledge transfer is also achieved by establishing a deep sparse autoencoder model. Therefore, our proposed recommendation model does not rely on the premise that there must be overlap between entities.

In addition, based on the assumption that "similar latent factors will also be more likely to have similar entities", the regularization terms of the user and item latent factors are added to the constraints of the cross-domain network training process.

$$\Re(U_I'') = tr((U_I'')^T L_u U_I'') \tag{21}$$

$$\Re(V_I'') = tr((V_I'')^T L_v V_I'') \tag{22}$$

where L_u denotes a Laplacian matrix, $L_u = D - W_u$. W_u is the user similarity matrix, and D is a diagonal matrix calculated by W_u. U_I'' and V_I'' are robust user and item latent factors, respectively. It should be noted that in our proposed model, the integrated and aligned user latent factors are used by the cosine method [57] to calculate and obtain the user similarity in W_u. The representations of items are similar to those of users. Laplacian regularization constraints consider the feature spaces of the user and item latent factors, punish large errors, and make model training smoother and more flexible.

Finally, the loss function of the deep autoencoder with sparse constraints is further optimized as follows.

$$J(U_I'', V_I''; \delta, \beta, \lambda) = \frac{1}{2}\|I \circ (X_{st} - U_I''(V_I'')^T)\|_F^2 + \frac{\delta}{2}\|U_I''\|_F^2 + \frac{\delta}{2}\|V_I''\|_F^2$$
$$+\frac{\lambda}{2}[tr((U_I'')^T L_u U_I'') + tr((V_I'')^T L_v V_I'')] + \beta KL(\rho \parallel \hat{\rho}_j)$$
$$s.t.\,(U_I'')^T V_I'' = I \tag{23}$$

Since there are at least two variables that need to be optimized, the neural network training process has difficulty updating them simultaneously. Therefore, we can optimize them one by one, that is, optimize one parameter by fixing the other. As a result, the objective function is trained with the following gradient descent update rule.

$$U_I'' \leftarrow U_I'' - \eta_u[(U_I''(V_I'')^T - X_{st})V_I'' + \delta U_I'' + \lambda L_u U_I'' + \Delta\rho_u] \tag{24}$$

$$V_I'' \leftarrow V_I'' - \eta_u[(V_I''(U_I'')^T - X_{st}^T)U_I'' + \delta V_I'' + \lambda L_v V_I'' + \Delta\rho_v] \tag{25}$$

By iteratively updating U_I'' and V_I'', the final optimized approximation of $X_{st} = U_I''(V_I'')^T$ is obtained. The recommendation results are generated based on the rating prediction task in the target domain.

The overall proposed learning model (DSAP-AL) is outlined in Algorithm 3.

The DSAP-AL learning algorithm is mainly composed of three parts. First, DSAP-AL initializes the connection weights and network training layers and calls the ALFS algorithm to obtain the integrated user latent factor U_I' and item latent factor V_I'. Then, the loss function is built by transferring the connection weights, and the RDFA algorithm is called to obtain the robust user and item latent factors. Finally, during the iteration phase, we randomly select a small portion of the items that have not yet been observed to train the neural network until the objective function converges. Ultimately, the network error is minimized, and the recommendation results are output.

5. Experiment and analysis

This section first presents the utilized datasets, evaluation metrics, and comparison methods. Immediately afterwards, we evaluate the impacts of the model parameters on the cross-domain recommendation results. Then, the evaluation results in terms of the prediction accuracy and top-N recommendations are presented with some discussion on the related results to conclude this section.

5.1. Datasets and settings

To fairly evaluate the proposed DSAP-AL method, four public stable benchmark datasets collected from the real-world are selected and applied to our cross-domain recommendation experiments: (1) AmazonBooks, (2) Epinions, (3) the MovieLens 1M Dataset and (4) the MovieLens 25M dataset. Detailed descriptions of these datasets are shown below. Since there are no existing datasets suitable for our cross-domain scenario, previous works [58,59] are followed with regard to the experimental setup. Then, we describe in detail how to generate dense source domain data and sparse target domain data for AmazonBooks and briefly describe the other three datasets, which are generated via a similar process.

Amazon is one of the largest online book e-commerce sites in the United States, where users are free to rate products from 1 to 5. The AmazonBooks rating dataset[1] collected by R. He and J. McAuley [60] contains 22,507,155 ratings given by 8,026,324 users on approximately 2,330,066 items, for which there are five grade ratings. We first randomly split the AmazonBooks rating dataset into training and testing sets, with 60% and 40% of the ratings, respectively. Then, we randomly extract the ratings with greater than average sparsity in the training set as the dense source domain matrix, and the rest are used as the sparse target domain matrix. Finally, the obtained source and target domain matrices are utilized for the subsequent experimental training process.

Epinions is a social review website where users can browse reviews of various products to help them make purchase decisions, and they can also post reviews about products. The public Epinions dataset[2] adopted by our experiment was collected

1 http://jmcauley.ucsd.edu/data/amazon/.
2 http://alchemy.cs.washington.edu/data/epinions/.

Algorithm 3 DSAP-AL

Inputs: Source rating matrix X_s, Target rating matrix X_t, Factor abnormal threshold ε, Regularization term δ, Sparse parameter β
Output: Final predicted rating X_F

```
1: Initialize Connection weight W, Deep autoencoder layers L
2: ALFS(X_s, X_t)   // Integrated user latent factor U_I' and item latent factor V_I' are obtained
3: for layers=1,..., L-1 do
4:     transferWeight(W_i, W_{i+1})
5:     x_{i-1} <- x_i    // (x_i ∈ U_I' or V_I')
6: end for
7: Establish objective function J(U_I, V_I; δ, β)
8: RDFA(U_I', V_I', ε)    // Robust latent factors U_I'', V_I'' are obtained
9: for count=1,..., T do // T is the number of iterations
10:   for u=1,...,n do
11:     min J(U_I, V_I; δ, β)   //Compute predictionerror
12:     while not converged do
13:        Compute gradient ∇U_I'' // Update parameters according to gradients
14:        Compute gradient ∇V_I''
15:        Update parameter U_I''
16:        Update parameter V_I''
17:     end while
18:   end for
19: end for
20: return X_F
```

by Richardson et al. [61] from Epinions.com. After redundant rating records are filtered out, the dataset consists of 55,848 users, 481,452 items and 900,608 ratings. The Epinions dataset is extracted and set up in the same way as the AmazonBooks dataset.

MovieLens is a noncommercial, research-oriented online movie recommendation community that was created by the GroupLens team of the Department of Computer Science and Engineering at the University of Minnesota [62]. The MovieLens 1M dataset[3] and MovieLens 25M dataset[4] are adopted as our datasets for evaluating the experimental performance of our proposed method. Specially, the MovieLens 1M dataset contains over 1 million ratings from 6000 users on 4000 movies, and the MovieLens 25M dataset contains 20 million ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users. We preprocess the two datasets in a similar way as that done for the AmazonBooks dataset to generate source and target training data.

5.2. Evaluation metrics

To measure the performance of the proposed DSAP-AL model, the root mean square error (RMSE) and mean absolute error (MAE) are adopted as our evaluation metrics, both of which are currently the most popular in rating predictions. Formally, the RMSE between the predicted rating and the true rating is given by

$$RMSE = \sqrt{\frac{\sum_{u,i\in\zeta}(\hat{r}_{u,i} - r_{u,i})^2}{|\zeta|}} \tag{26}$$

where $\hat{r}_{u,i}$ is the predicted rating, $r_{u,i}$ denotes the true rating, and ζ is the validation dataset. Similarly, the other commonly used evaluation metric MAE is given by

$$MAE = \frac{\sum_{u,i\in\zeta}|\hat{r}_{u,i} - r_{u,i}|}{|\zeta|} \tag{27}$$

Obviously, the lower the RMSE and MAE values are, the smaller the prediction error of the learning model, and the better the performance in terms of cross-domain recommendations.

In our experiments, precision, recall and F-measure, as three commonly used evaluation metrics, are also adopted to further evaluate the top-N performance of the proposed learning model.

$$Precision = \frac{\sum|\mathrm{N}\bigcap \mathrm{T}|}{\sum|\mathrm{N}|} \tag{28}$$

$$Recall = \frac{\sum|\mathrm{N}\bigcap \mathrm{T}|}{\sum|T|} \tag{29}$$

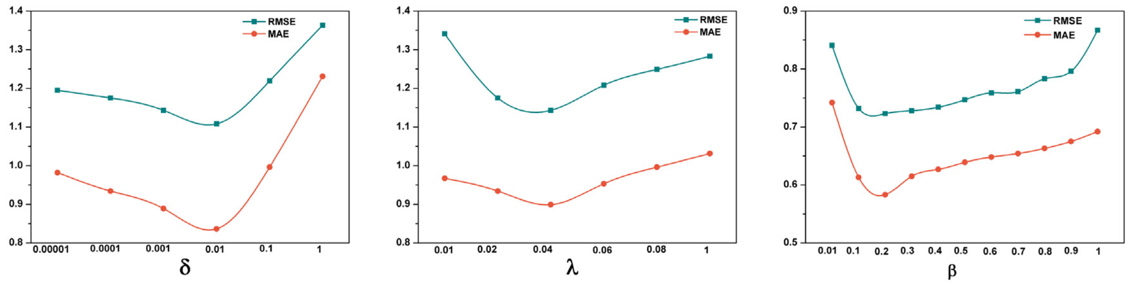
$$F\text{-}measure = \frac{2\times Precision\times Recall}{Precision + Recall} \tag{30}$$

where T represents the number of ratings in the testing set and N denotes the number of ratings in the recommendation set. The F-measure is a balance indicator between precision and recall. Obviously, larger values of these metrics reveal better recommendations.

5.3. Parameter sensitivity tests

As shown in Section 4, the training process of the proposed DSAP-AL model depends on all the parameters of the final objective function, including the model learning parameter δ, regularization parameter λ, and sparse parameter β. Therefore, the implementation of parameter sensitivity tests is necessary to investigate their impacts on the performance of the proposed model. In repeated parameter sensitivity experiments, we observe the specific changes of one variable by fixing the other parameters. For the sake of similarity, Fig. 4 only presents the experimental results for the AmazonBooks rating dataset, and the results measured for the other datasets are very similar.

3 https://grouplens.org/datasets/movielens/1m/.
4 https://grouplens.org/datasets/movielens/25m/.



(a) (b) (c)

Fig. 4. Parameter sensitivity analysis of DSAP-AL.

In Fig. 4(a), we can observe that when the model learning parameter δ is 0.001, the RMSE and MAE are at their lowest values, and the proposed learning model has the best performance. When δ is far from 0.001, both the RMSE and MAE increase gradually. In particular, the sharp changes of the performance during the early stage indicate that the δ parameter is very sensitive in the initial stage of latent factor training. In addition, this parameter controls the complexity of the algorithm to avoid overfitting.

As shown in Fig. 4(b), we can see that the RMSE and MAE values of the proposed DSAP-AL model decrease gradually and then increase slowly. The learning model achieves the lowest MAE=0.899 and RMSE=1.143 when the regularization parameter $\lambda = 0.04$. When $\lambda = 0$, meaning the $tr((U_l'')^T L_u U_l'')+tr((V_l'')^T L_v V_l'')$ term loses its influence, the MAE is 1.031 and the RMSE is 1.283.

In Fig. 4(c), we can notice that the RMSE and MAE change with different sparse settings for β, and this reflects the impact of sparse encoder learning on the proposed DSAP-AL model. The algorithm achieves the best performance when $\beta = 0.2$, where the lowest RMSE=0.723 and the lowest MAE=0.583. Similarly, when the sparse parameter β is far from 0.2, both the RMSE and MAE increase gradually.

5.4. Comparison methods

In this section, we compare the performance of the proposed DSAP-AL model with those of some state-of-the-art single-domain and cross-domain recommendation methods that are listed below.

(1) An alternating direction-based nonnegative latent factor model (ANLF) [63], as an efficient sparse learning method in single domain recommendation systems, was proposed to implement alternating direction-based latent factor optimizations with regard to each single feature and obtain high prediction accuracy and low complexity.

(2) The deep learning-based collaborative filtering model (DLCF) [64] is a promising single-domain recommendation approach that learns low-rank vector spaces for users and items by embedding semantic information into a collaborative filtering process. Furthermore, a deep feed-forward neural network is utilized to represent and train the interaction ratings among users and items.

(3) A deep autoencoder model based on convolutional text networks (DA-CTN) [65] is a single-domain recommendation method designed to address data sparsity and cold-start recommendations. Then, the latent information and the correlations between user and item features are deeply mined from user browsing behavior records, and the final rating predictions are generated by combining the rating vector spaces and multilayer perceptions.

(4) Complete tag-induced cross-domain recommendation (CTCR) [66] learns the intradomain and interdomain associations of users or items from their tag profiles and then adds structured user and item constraints to the collective matrix factorization. This model can make full use of the knowledge encoded in shared and individual domain tags to improve the rating accuracy. The CTCR and DSAP-AL models do not require entities to overlap, and their performance are compared under the same experimental conditions.

(5) Deep transfer collaborative filtering (DTCF) [44] was proposed in the cross-domain recommendation model, where nonnegative matrix tri-factorization and stacked denoising autoencoders are adopted to capture ratings' statistical characteristics and the side information. Deep transfer learning can generate effective latent representations.

(6) Kernel-induced knowledge transfer (KerKT) [67] presents a cross-domain recommendation framework, where the feature spaces of overlapping users or items are adjusted by a domain adaptation technique, and the nonoverlapping users or items between domains are aligned by a diffusion kernel filling algorithm. Finally, knowledge can be effectively transferred through overlapping entities to alleviate the data sparsity problem. The impact of the cross-domain transfer learning performance on the rating prediction accuracy is compared among the DTCF, the KerKT and the proposed DSAP-AL models.

5.5. Results and analysis

In the experiment, the parameters of the proposed DSAP-AL model and other comparison methods are set to their optimal values through cross-validation. For example, for the Amazon-Books rating dataset, the proposed learning method has a learning parameter of 0.001, a regularization parameter of 0.04, and a sparse parameter of 0.2, as shown in the parameter sensitivity test section.

5.5.1. Rating prediction accuracy

We summarize the results of the comparison experiments on four datasets with data sparsities of 0.05%, 0.5% and 5%, and the number of nearest neighbors of 25, 50, and 75. The specific results are shown in Tables 1–4, and the analysis of the results is summarized as follows.

(1) Compared with the single-domain recommendation methods, including ANLF, DLCF and DA-CTN, the proposed DSAP-AL model achieves the lowest error and the best performances under different sparse conditions. We can also find that, in general, the deep DA-CTN and DSAP-AL learning models perform better than the ANLF and DLCF models without encoding representation techniques. However, in all experiments, the DSAP-AL significantly outperforms all single-domain models.

(2) In contrast with the cross-domain recommendation approaches, including the CTCR, DTCF and KerKT methods, our proposed DSAP-AL model achieves the lowest error in terms of the RMSE and MAE metrics. Similarly, the experimental results show that our model not only has impressive rating prediction

Table 1
The comparative performance on the AmazonBooks dataset.

		0.05%			0.50%			5%		
		25	50	75	25	50	75	25	50	75
RMSE	ANLF	0.9034	0.9006	0.8997	0.8875	0.8890	0.8883	0.8845	0.8853	0.8814
	DLCF	0.8847	0.8814	0.8808	0.8647	0.8607	0.8623	0.8569	0.8572	0.8588
	DA-CTN	0.8660	0.8649	0.8626	0.8603	0.8579	0.8592	0.8567	0.8534	0.8515
	CTCR	0.8399	0.8405	0.8411	0.8372	0.8359	0.8334	0.8236	0.8249	0.8222
	DTCF	0.8249	0.8273	0.8219	0.8193	0.8178	0.8187	0.8162	0.8133	0.8141
	KerKT	0.7774	0.7739	0.7802	0.7718	0.7699	0.7704	0.7689	0.7654	0.7687
	DSAP-AL	**0.7499**	**0.7533**	**0.7512**	**0.7482**	**0.7459**	**0.7466**	**0.7418**	**0.7433**	**0.7425**
MAE	ANLF	0.6628	0.6607	0.6661	0.6577	0.6538	0.6554	0.6452	0.6419	0.6404
	DLCF	0.6389	0.6444	0.6405	0.6317	0.6300	0.6274	0.6207	0.6135	0.6184
	DA-CTN	0.6330	0.6252	0.6277	0.6234	0.6215	0.6207	0.6170	0.6138	0.6122
	CTCR	0.6193	0.6187	0.6155	0.6075	0.6088	0.6118	0.6004	0.6025	0.6033
	DTCF	0.6165	0.6143	0.6109	0.6018	0.6026	0.6065	0.5931	0.5935	0.5966
	KerKT	0.5846	0.5858	0.5809	0.5797	0.5743	0.5770	0.5708	0.5665	0.5691
	DSAP-AL	**0.5711**	**0.5680**	**0.5642**	**0.5566**	**0.5608**	**0.5575**	**0.5549**	**0.5526**	**0.5534**

Table 2
The comparative performance on the Epinions dataset.

		0.05%			0.50%			5%		
		25	50	75	25	50	75	25	50	75
RMSE	ANLF	0.9577	0.9557	0.9561	0.9455	0.9427	0.9380	0.9288	0.9172	0.9254
	DLCF	0.9393	0.9314	0.9322	0.9134	0.9257	0.9201	0.9117	0.9009	0.9104
	DA-CTN	0.9028	0.8991	0.9004	0.8921	0.8993	0.8905	0.8799	0.8852	0.8817
	CTCR	0.8889	0.8837	0.8876	0.8766	0.8724	0.8735	0.8661	0.8630	0.8594
	DTCF	0.8812	0.8749	0.8723	0.8692	0.8640	0.8565	0.8586	0.8507	0.8523
	KerKT	0.8416	**0.8311**	0.8377	0.8301	0.8309	0.8206	0.8235	0.8244	0.8257
	DSAP-AL	**0.8258**	0.8313	**0.8217**	**0.8155**	**0.8173**	**0.8200**	**0.8126**	**0.8084**	**0.8101**
MAE	ANLF	0.7719	0.7701	0.7625	0.7561	0.7473	0.7524	0.7416	0.7441	0.7368
	DLCF	0.7342	0.7297	0.7325	0.7261	0.7206	0.7288	0.7162	0.7133	0.7084
	DA-CTN	0.7258	0.7230	0.7166	0.7114	0.7099	0.7107	0.6975	0.7034	0.7011
	CTCR	0.7062	0.6983	0.7016	0.6921	0.6880	0.6868	0.6772	0.6722	0.6803
	DTCF	0.7006	0.6992	0.6934	0.6854	0.6742	0.6784	0.6702	0.6651	0.6690
	KerKT	0.6702	0.6665	0.6637	**0.6419**	0.6540	0.6522	0.6351	0.6414	0.6408
	DSAP-AL	**0.6515**	**0.6527**	**0.6488**	0.6423	**0.6366**	**0.6407**	**0.6278**	**0.6314**	**0.6334**

Table 3
The comparative performance on the MovieLens 1M dataset.

		0.05%			0.50%			5%		
		25	50	75	25	50	75	25	50	75
RMSE	ANLF	1.0662	1.0625	1.0607	1.0564	1.0481	1.0526	1.0453	1.0429	1.0400
	DLCF	1.0375	1.0308	1.0327	1.0256	1.0223	1.0190	1.0162	1.0144	1.0121
	DA-CTN	1.0137	1.0112	1.0105	1.0068	0.9974	1.0029	0.9946	0.9928	0.9907
	CTCR	1.0018	0.9923	0.9966	0.9890	0.9819	0.9847	0.9817	0.9774	0.9725
	DTCF	0.9886	0.9733	0.9764	0.9714	0.9625	0.9648	0.9604	0.9581	0.9532
	KerKT	0.9616	0.9630	**0.9555**	0.9538	0.9489	0.9514	0.9427	0.9435	0.9472
	DSAP-AL	**0.9468**	**0.9417**	0.9562	**0.9331**	**0.9267**	**0.9307**	**0.9233**	**0.9215**	**0.9166**
MAE	ANLF	0.9510	0.9487	0.9436	0.9355	0.9394	0.9428	0.9322	0.9239	0.9201
	DLCF	0.9054	0.9033	0.9017	0.9009	0.8926	0.8947	0.8865	0.8812	0.8824
	DA-CTN	0.8903	0.8856	0.8817	0.8734	0.8706	0.8633	0.8609	0.8558	0.8527
	CTCR	0.8714	0.8700	0.8623	0.8601	0.8591	0.8538	0.8475	0.8516	0.8404
	DTCF	0.8534	0.8488	0.8516	0.8442	0.8407	0.8381	0.8320	0.8257	0.8213
	KerKT	0.8461	0.8330	0.8326	0.8269	0.8222	0.8165	0.8184	0.8137	0.8108
	DSAP-AL	**0.8170**	**0.8144**	**0.8106**	**0.8011**	**0.7985**	**0.8037**	**0.7881**	**0.7961**	**0.7907**

accuracy, but is also more robust than other models to different types of sparse data.

(3) The KerKT and our proposed DSAP-AL cross-domain transferring methods achieves the best performances among all the comparison methods. These two methods achieve the lowest error with regard to the RMSE and MAE. In addition, in most cases, the performance of our proposed approach is significantly better than that of the KerKT model. Although the KerKT sometimes shows the best improvement (Tables 2 and 3), the proposed DSAP-AL model is also very close to the optimal values in these cases.

(4) As the results show, the pure latent factor model ANLF performs the worst among all the comparison methods in terms of rating predictions. However, the proposed DSAP-AL approach, which combines the deep sparse encoding and knowledge transfer techniques, demonstrates impressive performances in terms of cross-domain rating predictions. Thus, we naturally conclude that our proposed model outperforms all advanced contrast approaches in terms of prediction accuracy.

Table 4
The comparative performance on the MovieLens 25M dataset.

		0.05%			0.50%			5%		
		25	50	75	25	50	75	25	50	75
RMSE	ANLF	1.2086	1.8367	1.1790	1.1582	1.1608	1.1538	1.1473	1.1364	1.1305
	DLCF	1.1221	1.1184	1.1137	1.1075	1.1007	1.0954	1.0920	1.0899	1.0826
	DA-CTN	1.0834	1.0722	1.0719	1.0607	1.0594	1.0505	1.0466	1.0373	1.0321
	CTCR	1.0409	1.0286	1.0200	1.0122	1.0078	1.0060	0.9997	1.0024	0.9931
	DTCF	1.0147	1.0112	1.0068	1.0011	0.9956	0.9900	0.9881	0.9825	0.9844
	KerKT	0.9889	0.9860	0.9805	0.9774	0.9718	0.9633	0.9607	0.9562	0.9539
	DSAP-AL	**0.9756**	**0.9728**	**0.9692**	**0.9640**	**0.9584**	**0.9611**	**0.9523**	**0.9495**	**0.9457**
MAE	ANLF	0.9925	0.9794	0.9810	0.9751	0.9748	0.9633	0.9582	0.9608	0.9566
	DLCF	0.9439	0.9527	0.9411	0.9390	0.9324	0.9286	0.9178	0.9233	0.9105
	DA-CTN	0.9360	0.9295	0.9221	0.9107	0.9062	0.9016	0.8908	0.8846	0.8813
	CTCR	0.9200	0.9153	0.9106	0.9014	0.8968	0.8925	0.8777	0.8812	0.8709
	DTCF	0.8975	0.8940	0.8854	0.8805	0.8663	0.8768	0.8609	0.8532	0.8501
	KerKT	0.8757	0.8609	0.8571	0.8488	0.8522	0.8436	0.8404	0.8347	0.8324
	DSAP-AL	**0.8616**	**0.8578**	**0.8537**	**0.8520**	**0.8469**	**0.8411**	**0.8309**	**0.8266**	**0.8215**

5.5.2. Performance on Top-N recommendation

To better measure Top-N performance of the proposed learning model and other comparison approaches, we conducted another set of experiments on the AmazonBooks rating datasets by changing N from 10 to 100. Fig. 5 shows the specific comparison results in terms of precision, recall and F-measure metrics.

Fig. 5(a) depicts the comparison results based on precision. Obviously, the performance of the proposed algorithm is significantly better than those of other benchmark models throughout the test. In particular, our method achieves the best RMSE 0.84 when N is close to 40. Therefore, these results indicate that our model has impressive performance in terms of precision.

Figs. 5(b) and(c) show the recall and F-measure results, respectively. Our method performs better as N increases, especially when the number of nearest neighbors exceeds 50. Even compared with the efficient KerKT algorithm, our model has an obvious accuracy advantage in the top-N recommendations tests. This is because the cross-domain recommendations in our model can provide more available rating data for neural network training. Thus, these results demonstrate that the proposed DSAP-AL model outstandingly outperforms the other advanced comparison methods according to the precision, recall and F-measure metrics.

5.5.3. Complexity and convergence analysis

The complexity analysis of the proposed DSAP-AL model contains all four steps in Section 4. Assuming that the rating matrices in both the source and target domains are $n*n$ matrices, the time complexity of each step is analyzed as follows.

*Step 1:*The time iteration cost of joint matrix factorization is $O(2n^2)$. The time consumption for integrating and aligning latent factors is $O((n+n)^2)$.

*Step 2:*The time complexity of weight and factor transfer in deep sparse autoencoder is $O(kn)$. k is the dimension of the neural network trainings.

*Step 3:*The time consumption of the domain factor adaptation is $O(4n^2)$.

*Step 4:*The time consumption of parameter and variable gradient update in the proposed learning model is $O(4n^2)$.

Therefore, the overall complexity of the proposed DSAP-AL model is $O(kn^2)$. In addition, the total time consumption of each comparison method on the AmazonBooks dataset is calculated and summarized in Table 5. The comparative experiments were performed on a computer with Intel Core i5 and 8 GB memory.

As shown in Table 5, we can see that since single-domain recommendation models do not need to deal with data transfer across domains, their algorithms generally consume less time. Alternatively, compared with other cross-domain models, the time consumption of our model is the shortest, which can provide a reliable premise for application to online recommender systems.

Table 5
Time consumption comparison on the AmazonBooks dataset.

Methods	Single domain			Cross-domain			
	ANLF	DLCF	DA-CTN	CTCR	DTCF	KerKT	DSAP-AL
Time (s)	32.76	56.41	22.94	215.38	324.79	189.55	157.43

In addition, the convergence trends of the proposed DSAP-AL method and three benchmark cross-domain models on the AmazonBooks and Epinions datasets are shown in Fig. 6.

As shown in Fig. 6, all cross-domain methods converge very quickly in the early stages of the iteration trainings, and our proposed DSAP-AL model always converges faster than other models throughout the training process. Furthermore, all models tend to converge and stabilize after a certain number of iterations, such as 10 and 20 iterations on the AmazonBooks and Epinions datasets, respectively. In conclusion, the proposed model not only shows better convergence but also has higher efficiency.

5.5.4. Cluster visualization of item prediction ratings

To show the transfer efficiency of the latent factors more clearly, the prediction ratings of the proposed DSAP-AL and the ANLF model are visualized in a two-dimensional view, where three types of books (Management, Science and Technology, and Biography) randomly selected from the AmazonBooks dataset are exploited for visualization. Different categories are denoted by different colors. The clustering visualization results of these item categories are exhibited in Fig. 7(a) and (b).

In Fig. 7(a), the different book categories in the ANLF model do not have obvious boundaries and clusters with respect to positions of books. However, as can be seen in Fig. 7(b), the different books can be well classified and clustered, and the positions of books can be predicted regularly. As a result, the proposed DSAP-AL learning model has the impressive accuracy of rating predictions from the perspective of visualization.

6. Conclusion and future work

In this paper, we present a novel cross-domain learning recommendation model, named the deep sparse autoencoder learning model based on adversarial learning (DSAP-AL), to solve the data sparsity problem and improve the accuracy of rating predictions. Unlike in previous studies, our model, which can utilize aligned latent factors as a bridge between the source and target domains, does not require the premise that any of the user or item entities overlap. Specifically, with joint matrix factorization, the latent factors of the users and items in the source and target domains are integrated and aligned in the unified

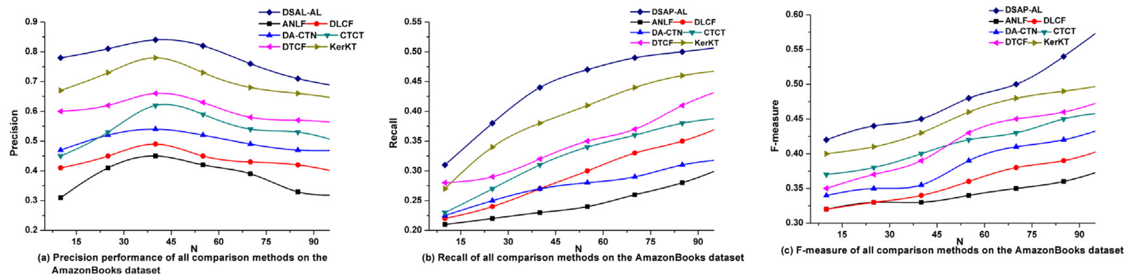


Fig. 5. Top-N recommendation results of all comparison methods on the AmazonBooks datasets.

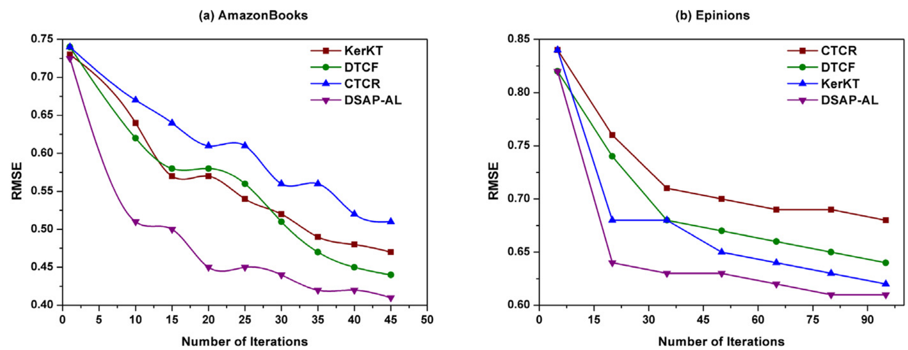


Fig. 6. Convergence trends of all cross-domain models on two datasets.

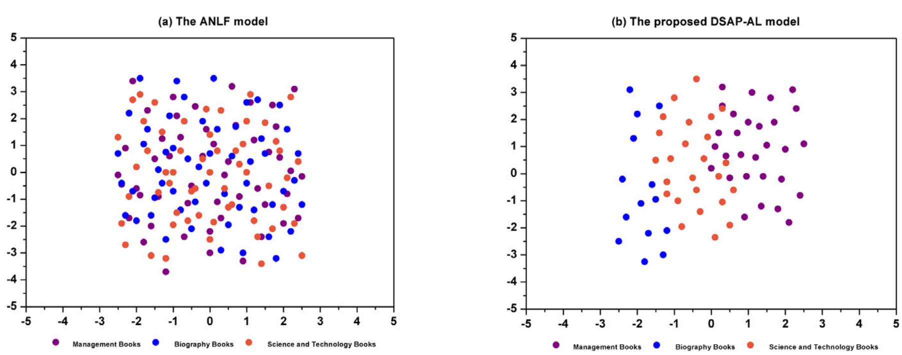


Fig. 7. Clustering visualization of prediction rating for different item categories.

factor pattern through a generative adversarial network model. Moreover, a deep sparse autoencoder model is built to transfer the aligned factors and weights. In addition, robust user and item factors are obtained by domain factor adaptation to train the proposed DSAP-AL learning model. Experiments on four datasets demonstrate the efficiency of the proposed learning approach. In particular, compared to the prediction accuracy of the competing transfer learning models applied in cross domain recommender systems [43,66], our proposed method has higher knowledge transfer efficiency and the best prediction results.

However, static rating profiles are mainly used for our cross-domain recommendations, and these cannot handle the dynamic requirements of recommendation systems well. Therefore, our future research will explore the influences of temporal and contextual information on cross-domain recommendation models.

CRediT authorship contribution statement

Yakun Li: Conceptualization, Methodology, Software, Data curation, Writing - original draft. Writing - review & editing. **Jiadong Ren:** Supervision, Software, Validation. **Jiaomin Liu:** Supervision. **Yixin Chang:** Visualization, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J. Xu, T. Xing, M.V.D. Schaar, Personalized course sequence recommendations, IEEE Trans. Signal Process. 64 (20) (2016) 5340–5352.

[2] S. Huang, J. Zhang, L. Wang, X.S. Hua, Social friend recommendation based on multiple network correlation, IEEE Trans. Multimedia 18 (2) (2015) 1-1.

[3] S.K. Lee, Y.H. Cho, S.H. Kim, Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations, Inform. Sci. 180 (11) (2010) 2142–2155.

[4] L. Yu, C. Liu, Z.K. Zhang, Multi-linear interactive matrix factorization, Knowl.-Based Syst. 85 (2015) 307–315.

[5] L.M.D. Campos, M.F.L. Juan, J.F. Huete, M.A. Rueda-Morales, Combining content-based and collaborative recommendations: a hybrid approach based on bayesian networks, Int. J. Approx. Reason. 51 (7) (2010) 785–799.

[6] A.Y. Xue, J. Qi, X. Xie, R. Zhang, J. Huang, Y. Li, Solving the data sparsity problem in destination prediction, VLDB J. 24 (2015) 219–243.

[7] M.M. Khan, R. Ibrahim, I. Ghani, Cross domain recommender systems, ACM Comput. Surv. 50 (3) (2017) 1–34.

[8] P. Cremonesi, M. Quadrana, Cross-domain recommendations without overlapping data: myth or reality? in: Proc. 8th ACM Conf. Recommender systems, Oct. 2014, pp. 297-300.
[9] A. Stewart, E. Diazaviles, W. Nejdl, L.B. Marinho, L. Schmidtthieme, Cross-tagging for personalized open social networking, in: Proc. 20th ACM Conf. Hypertext Hypermedia, 2009, pp. 271-278.
[10] R. Chung, D. Sundaram, A. Srinivasan, Integrated personal recommender systems, in: Proc. 9th Int. Conf. Electronic Commerce, Aug. 2007, pp. 19-22.
[11] W. Pan, N.N. Liu, E.W. Xiang, Q. Yang, Transfer learning to predict missing ratings via heterogeneous user feedbacks, in: Proc. 22nd Int. Joint Conf. Artificial Intelligence, Jul. 2011, pp. 16-22.
[12] Q. Zhang, D. Wu, J. Lu, F. Liu, G. Zhang, A cross-domain recommender system with consistent information transfer, Decis. Support Syst. 104 (2017) 49–63.
[13] H. Zhan, B. Shi, A.C. Kot, Cross-domain shoe retrieval with a semantic hierarchy of attribute classification network, IEEE Trans. Image Process. (2017) 1-1.
[14] C.L. Liu, X.W. Wu, Large-scale recommender system with compact latent factor model, Expert Syst. Appl. 64 (2016) 467–475.
[15] X. Luo, H. Wu, H. Yuan, M.C. Zhou, Temporal pattern-aware qos prediction via biased non-negative latent factorization of tensors, IEEE Trans. Cybern. PP (99) (2019) 1–12.
[16] Y. Xu, D. Li, Z. Wang, Q. Guo, W. Xiang, A deep learning method based on convolutional neural network for automatic modulation classification of wireless signals, Wirel. Netw. 25 (7) (2019) 3735–3746.
[17] J. Manotumruksa, Deep collaborative filtering approaches for context-aware venue recommendation, in: Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. 2017, pp. 1383-1383.
[18] W. Zhang, F. Liu, L. Jiang, D. Xu, Recommendation based on collaborative filtering by convolution deep learning model based on label weight nearest neighbor, in: Proc. 10th Int. Symp. Comput. Intell. Design, 2017, pp. 504-507.
[19] H. Guo, R. Tang, Y. Ye, Z. Li, X. He, Deepfm: a factorization-machine based neural network for CTR prediction, 2017, arXiv preprint arXiv:1703.04247.
[20] C. Wu, J. Wang, J. Liu, W. Liu, Recurrent neural network based recommendation for time heterogeneous feedback, Knowl.-Based Syst. 109 (2016) 90–103.
[21] Y. Zuo, J. Zeng, M. Gong, L. Jiao, Tag-aware recommender systems based on deep neural networks, Neurocomputing (2016).
[22] J. Yi, J. Huang, J. Qin, Rating prediction in review-based recommendations via adversarial auto-encoder, in: Proc. IEEE/WIC/ACM Int. Conf. Web Intell. 2018, pp. 144-149.
[23] O. Kuchaiev, B. Ginsburg, Training deep autoencoders for collaborative filtering, 2017.
[24] S. Lange, M. Riedmiller, Deep auto-encoder neural networks in reinforcement learning, in: Proc. Int. Joint Conf. Neural Netw. Jul. 2010, pp. 18-23.
[25] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670.
[26] J. Deng, Z. Zhang, F. Eyben, B. Schuller, Autoencoder-based unsupervised domain adaptation for speech emotion recognition, IEEE Signal Process. Lett. 21 (9) (2014) 1068–1072.
[27] F. Zhuang, Z. Zhang, M. Qian, C. Shi, X. Xie, Q. He, Representation learning via dual-autoencoder for recommendation, Neural Netw. 90 (2017) 83–89.
[28] H. Wang, X. Shi, D.Y. Yeung, Relational stacked denoising autoencoder for tag recommendation, in: Proc. 29th AAAI Conf. Artif. Intell. 2015.
[29] C. Zhang, T. Li, Z. Ren, Z. Hu, Y. Ji, Taxonomy-aware collaborative denoising autoencoder for personalized recommendation, Appl. Intell. (2019).
[30] J. Zhao, X. Geng, J. Zhou, Q. Sun, Y. Xiao, Z. Zhang, Attribute mapping and autoencoder neural network based matrix factorization initialization for recommendation systems, Knowl.-Based Syst. 166 (2019) 132–139.
[31] B. Bai, Y. Fan, W. Tan, J. Zhang, Dltsr: a deep learning framework for recommendation of long-tail web services, IEEE Trans. Serv. Comput. (2017) 1-1.
[32] Y. Chen, J. Zhang, M. Guo, J. Cao, Learning user preference from heterogeneous information for store-type recommendation, IEEE Trans. Serv. Comput. (2017) 1-1.
[33] A.M. Elkahky, Y. Song, X. He, A multi-view deep learning approach for cross domain user modeling in recommendation systems, in: Proc. 24th Int. Conf. Int. World Wide Web, 2015.
[34] S. Berkovsky, T. Kuflik, F. Ricci, Cross-domain mediation in collaborative filtering, 2007.
[35] F. Zhuang, P. Luo, H. Xiong, Y. Xiong, Q. He, Z. Shi, Cross-domain learning from multiple sources: A consensus regularization perspective, IEEE Trans. Knowl. Data Eng. 22 (12) (2010) 1664–1678.
[36] A. Farseev, I. Samborskii, A. Filchenkov, T.S. Chua, Cross-domain recommendation via clustering on multi-layer graphs, in: Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. 2017, pp. 195-204.
[37] G. Sheng, L. Hao, C. Da, S. Li, P. Gallinari, J. Guo, Cross-domain recommendation via cluster-level latent factor model, in: Mach. Learn. Knowl. Discovery Databases, 2013, pp. 161-176.
[38] N. Mirbakhsh, C.X. Ling, Improving top-n recommendation for cold-start users via cross-domain information, ACM Trans. Knowl. Discov. Data 9 (4) (2015) 1–19.
[39] H. Hwangbo, Y. Kim, An empirical study on the effect of data sparsity and data overlap on cross domain collaborative filtering performance, Expert Syst. Appl. 89 (2017) 254–265.
[40] H. Wu, Z. Zhang, K. Yue, B. Zhang, J. He, L. Sun, Dual-regularized matrix factorization with deep neural networks for recommender systems, Knowl.-Based Syst. 145 (2018) 46–58.
[41] A. Kumar, N. Kumar, M. Hussain, S. Chaudhury, S. Agarwal, Semantic clustering-based cross-domain recommendation, in: Proc. IEEE Symp. Comput. Intell. Data Mining, 2014, pp. 137-141.
[42] S. Tan, J. Bu, X. Qin, C. Chen, D. Cai, Cross domain recommendation based on multi-type media fusion, Neurocomputing 127 (2014) 124–134.
[43] M. He, J. Zhang, J. Zhang, MINDTL: Multiple incomplete domains transfer learning for information recommendation, China Commun. 14 (11) (2017) 218–236.
[44] S. Gai, F. Zhao, Y. Kang, Z. Chen, D. Wang, A. Tang, Deep transfer collaborative filtering for recommender systems, in: 16th Pacific Rim Int. Conf. Artif. Intell. 2019, pp. 515-528.
[45] J. Lian, F. Zhang, X. Xie, G. Sun, CCCFNet: a content-boosted collaborative filtering neural network for cross domain recommender systems, in: Proc. 26th int. conf. World Wide Web companion, 2017, pp. 817-818.
[46] G. Hu, Y. Zhang, Q. Yang, Transfer meets hybrid: a synthetic approach for cross-domain collaborative filtering with text, in: The World Wide Web Conf. 2019, pp. 2822-2829.
[47] F. Yuan, L. Yao, B. Benatallah, Darec: deep domain adaptation for cross-domain recommendation via transferring rating patterns, 2019.
[48] C. Wang, M. Niepert, H. Li, Recsys-dan: discriminative adversarial networks for cross-domain recommender systems, IEEE Trans. Neural Netw. Learn. Syst. (2019) 1–10.
[49] B. Li, Q. Yang, X. Xue, Can movies and books collaborate? Cross-domain collaborative filtering, for sparsity reduction:, in Proc. Int. Joint Conf. Artif. Intell. 2009, pp. 11-17.
[50] S.H. Wu, Non-linear cross-domain collaborative filtering via hyper-structure transfer, in: Proc. 32nd Int. Conf. Mach. Learn. 2015, pp. 1190–1198.
[51] Y. Zhong, L. Liu, D. Zhao, H. Li, A generative adversarial network for image denoising, Multimedia Tools Appl. (2) (2019).
[52] S. Yan, X. Yan, Using labeled autoencoder to supervise neural network combined with k-nearest neighbor for visual industrial process monitoring, Ind. Eng. Chem. Res. 58 (23) (2019).
[53] W. QingJie, W. WenBin, Research on image retrieval using deep convolutional neural network combining l1 regularization and prelu activation function, IOP Conf. Ser. Earth Environ. Sci. (2017).
[54] J. Lu, J. Xuan, G. Zhang, X. Luo, Structural property-aware multilayer network embedding for latent factor analysis, Pattern Recognit. 76 (2018) 228–241.
[55] J. Lu, H. Zuo, G. G. Zhang, Fuzzy multiple-source transfer learning, IEEE Trans. Fuzzy Syst. (2019).
[56] T. Van Erven, P. Harremos, Rényi divergence and kullback-leibler divergence, IEEE Trans. Inform. Theory 60 (7) (2014) 3797–3820.
[57] M. Wang, S. Sun, Y. Li, A compressive tracking method based on gaussian differential graph and weighted cosine similarity metric, IEEE Signal Process. Lett. 25 (4) (2016) 501–505.
[58] W. Pan, Z. Ming, Interaction-rich transfer learning for collaborative filtering with heterogeneous user feedback, IEEE Intell. Syst. 29 (6) (2014) 48–54.
[59] W. Pan, Q. Yang, Transfer learning in heterogeneous collaborative filtering domains, Artif. Intell. 197 (2013) 39–55.
[60] R. He, J. McAuley, Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering, in Proc. 25th Int. Conf. World Wide Web, 2016, pp. 507–517.
[61] M. Richardson, P. Domingos, Mining knowledge-sharing sites for viral marketing, in: Proc. 8th ACM SIGKDD Int. Conf.Knowl. discovery data mining, 2002, pp. 61-70.
[62] Y. Li, J. Liu, J. Ren, A novel learning model based on trust diffusion and global item for recommender systems, IEEE Access 7 (2019) 170270-170281.
[63] X. Luo, M.C. Zhou, S. Li, Z. You, Y. Xia, Q. Zhu, A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method, IEEE Trans. Neural Netw. Learn. Syst. (2015) 1-1.

[64] M. Fu, H. Qu, Z. Yi, L. Lu, Y. Liu, A novel deep learning-based collaborative filtering model for recommendation system, IEEE Trans. Cybern. (2018) 1–13.

[65] W. Yan, D. Wang, M. Cao, J. Liu, Deep auto encoder model with convolutional text networks for video recommendation, IEEE Access 7 (2019) 40333–40346.

[66] P. Hao, G. Zhang, L. Martinez, J. Lu, Regularizing knowledge transfer in recommendation with tag-inferred correlation, IEEE Trans. Cybern. PP (99) (2017) 1–14.

[67] Q. Zhang, J. Lu, D. Wu, G. Zhang, A cross-domain recommender system with kernel-induced knowledge transfer for overlapping entities, IEEE Trans. Neural Netw. Learn. Syst. 30 (7) (2018).