A Novel Cross-Domain Recommendation with Evolution Learning

YI-CHENG CHEN, National Central University, Taiwan
WANG-CHIEN LEE, The Pennsylvania State University, USA

In this "info-plosion" era, recommendation systems (or recommenders) play a significant role in finding interesting items in the surge of online digital activities and e-commerce. Several techniques have been widely applied for recommendation systems, but the cold-start and sparsity problems remain a major challenge. The cold-start problem occurs when generating recommendations for new users and items without sufficient information. Sparsity refers to the problem of having a large amount of users and items but with few transactions or interactions. In this article, a novel cross-domain recommendation model, Cross-Domain Evolution Learning Recommendation (abbreviated as CD-ELR), is developed to communicate the information from different domains in order to tackle the cold-start and sparsity issues by integrating matrix factorization and recurrent neural network. We introduce an evolutionary concept to describe the preference variation of users over time. Furthermore, several optimization methods are developed for combining the domain features for precision recommendation. Experimental results show that CD-ELR outperforms existing state-of-the-art recommendation baselines. Finally, we conduct experiments on several real-world datasets to demonstrate the practicability of the proposed CD-ELR.

CCS Concepts: • **Information systems** → **Recommender systems** • **Computing methodologies** → **Neural networks**;

Additional Key Words and Phrases: Cross-domain recommendation, deep learning, matrix factorization, recommendation system, recurrent neural network

ACM Reference Format:
Yi-Cheng Chen and Wang-Chien Lee. 2024. A Novel Cross-Domain Recommendation with Evolution Learning. *ACM Trans. Internet Technol.* 24, 1, Article 6 (February 2024), 23 pages. https://doi.org/10.1145/3639567

1 INTRODUCTION

In the last decade, partly due to the exponential growth of data and information on the **World Wide Web (WWW)**, we have witnessed the birth of an information-explosion (info-plosion) era. Without any doubt, it is critical to extract/mine useful information relevant to users' interests efficiently and effectively, which is often achieved by recommendation systems (also called recommenders). This is particularly obvious for online e-commerce on the WWW. By capturing potential preferences of users to find interesting items for them, recommenders have emerged as an essential part of e-commerce over the years. Many studies [26, 46, 49, 54] have discussed and proven the importance of recommenders in satisfying users' needs and increasing revenue in e-commerce.

Authors' addresses: Y.-C. Chen, Department of Information Management, National Central University, Taoyuan City, Zhongli District, Zhongda RD, No. 300, Taiwan; e-mail: ycchen@mgt.ncu.edu.tw; W.-C. Lee, Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA; e-mail: wlee@cse.psu.edu.

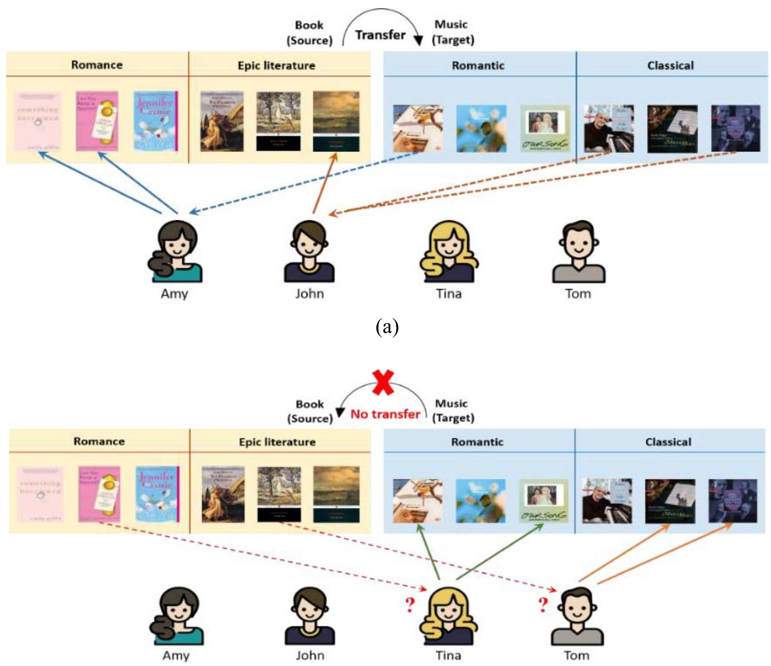
ACM 1533-5399/2024/02-ART6
https://doi.org/10.1145/3639567

We use the successful story of Amazon as an example to show the crucial role played by recommendation systems [48]. A strategy that drives the success of Amazon is to attract customers to click on recommended items by providing valuable information of items that match well with the needs and preferences of customers. According to a study [48], up to 35% of Amazon's sales are coming from the proprietary product recommendation models, which have played a significant role in Amazon's developed strategy to recommend a set of product items based on a particular item that a customer is looking for. The similar-context strategy persuades the customer that the promoted product items in the recommended lists are worth looking into. However, generally speaking, recommendation systems may suffer from the data sparsity problem and cold-start issues. The data sparsity problem refers to the situation where users' interests are skewed, i.e., most clicks (feedbacks) are concentrated in a few items while most items only get a few feedbacks. The cold-start problem refers to the situation where the recommendations systems could not make effective personalized recommendations for new customers, because there is not sufficient user history data to derive personal preferences.

Several prior studies explore ***cross-domain recommendation*** **(CDR)** to address data sparsity and cold-start problems, aiming to transfer the information from a data-rich domain to another data-sparse domain. The idea is to use the ratings or feedbacks from the data-rich *source* domain to improve the recommendation effectiveness in the data-sparse *target* domain. Nevertheless, existing CDR techniques face two critical issues. First, prior CDRs mostly utilize only the information in the source domain to assist the recommendations in the target domain, i.e., the recommendation focuses on unidirectional transfer. We argue that as the source and target domains may both have useful information to each other, the transfer could be bidirectional. Not only the source may assist the target, but also the information from the target may improve the recommendations in the source. We use a case scenario as an example to explain our argument in detail.

Case Scenario 1 (Importance of bidirectional CDR): Suppose we have user feedbacks from the book and music domains, where the book domain contains much richer rating information than the music domain. Typically, CDR systems transfer the source domain knowledge (i.e., book) to the target domain (i.e., music) to improve the recommendations in the target domain (i.e., music), but not vice versa. As shown in Figure 1(a), in the book domain, Amy and John prefer the romance and epic literatures, respectively. Hence, although user feedbacks are sparse in the music domain, the traditional CDR techniques would increase the probability for recommending romantic music to Amy and classical music to John based on the enhancement from the book domain. On the contrary, as in the example shown in Figure 1(b), in the music domain, while Tina likes romantic music and Tom likes electric music, the information about their preference in music is not transferred to books, since the music domain is sparser than the book domain. CDR only learns the knowledge from the source for use in the target domain, but not vice versa. Intuitively, as the patterns in the music domain could influence the recommendations of books, romantic and epic novels could be recommended to Tina and Tom, respectively, as well. This example shows the importance of the bidirectional transfer in cross-domain recommendations.

In general, user preferences change over time in many real-world applications. The latent factors discovered from traditional CDR techniques may not properly depict the evolution of user preferences due to the lack of the concept of time progress. Moreover, the evolving relationships among users and items are pivotal for making recommendations. Such a relationship could indicate the potential interests of a user to an item over time. Basically, prior CDR methods only discover the static features of users and items without considering the evolving relationship among them. Consider the scenario as follows.



(a)

(b)

Fig. 1. Example of unidirectional cross-domain recommendation.

Name	Category									
	Romance book			Science fiction			Graphic Novels		Horror book	
	Twilight	Bet Me	The Unhoneymooners	Dune	A Wrinkle in Time	Elevation	Adulthood Is a Myth	Black Panther	The Hollow Places	Imaginary Friend
Maria	2010/06 5★	2011/10 4.5★	2020/03 2★	2020/12 5★	2020/03 4★					
Eric							2011/07 4.5★	2011/09 5★	2020/12 5★	2020/01 4★
Ray				2014/08 4.5★	2010/05 5★	2020/01 5★				

Fig. 2. Example of preference evolution in recommendation.

***Case Scenario 2 (Preference evolution in recommendation)*:** As illustrated by the example in Figure 2, Maria liked to read romance books 10 years ago, such as "Twilight" and "Bet Me." However, her reading preference changes as her age increases. Currently, Maria likes to read science fiction such as "Dune" and "A Wrinkle in Time" instead of "The Unhoneymooners," as shown in the rating matrix in Figure 2. If we do not take into account the factor of preference evolution, the system may recommend Maria both science fiction and romance books, while the latter is no longer a good recommendation. On the contrary, Ray always likes science fiction literatures and thus science fiction books should be recommended to him directly, due to Ray's steady interest (which is also a kind of preference evolution). The example depicts an important role evolving preference may play in recommendations and thus should be considered in the CDR techniques.

In this article, by extending **matrix factorization (MF)** and **recurrent neural network (RNN)** techniques, a novel recommendation system, called ***Cross-Domain Evolution-Learning Recommendation*** **(CD-ELR)**, is developed to make personalized recommendations of items to users based on the ideas of preference evolution with cross-domain knowledge transfer. We introduce an evolving MF method in CD-ELR to extract user preference and item characteristics with efficient computation resource utilization. Then, CD-ELR integrates context awareness into RNN to efficiently capture the evolution patterns to predict the user preferred items in the future. The main contributions of this study are as follows:

- We propose an effective approach to transform a large-scale user rating matrix into a series of smaller feedback matrices based on users' rating time. Without information loss, the series of feedback matrices properly express the users' accumulated rating behaviors. Clearly, since every feedback matrix is much smaller than the whole rating matrix, less computation time and memory resource are required for processing.
- A novel **evolving matrix factorization (EMF)** method is proposed for matrix decomposition. Latent user preferences and item characteristics therefore are derived more efficiently. Since there are many customers and product items in real-world applications, the feedback rating matrix is extremely large, requiring significant computational resources (i.e., memory usage and execution time, to name a few). Compared with traditional MF-based methods, the proposed EMF method reduces more than 20% of the computational resources.
- By incorporating EMF to extract user preferences and item characteristics, a novel RNN-based learning model, called ***Fusion Long Short-Term Memory*** **(F-LSTM)**, is developed to dynamically capture the evolution patterns of user interests with the discovered characteristics in each domain. By considering the discovered latent factors of users and items in an evolutionary manner, the proposed CD-ELR model can recommend the "right" items to users at the "right" time.
- The proposed F-LSTM integrates knowledge extracted from multiple domains. Moreover, we could effectively consider the knowledge extracted from each domain to find out the relation between users and items. Via the well-trained F-LSTM model, CD-ELR predicts future user preferences to precisely recommend related items.
- Different from prior studies, CD-ELR achieves multidirectional transfer for CDRs. Not only does the source domain transfer knowledge to the target domain, but the target domain also enhances the recommendations in the source domain. With the proposed fusion operations, multiple transfers could be naturally implemented.
- Extensive experiments are performed on real datasets. Experimental results indicate that CD-ELR significantly outperforms existing cross-domain recommendation models. Furthermore, the proposed CD-ELR exhibits great generalization and robustness on real datasets with all considered metrics.

The rest of the article is organized as follows. Section 2 reviews the Related Work and Section 3 introduces the proposed CD-ELR, respectively. Section 4 details the experiments for performance evaluation. Finally, Section 5 concludes the work.

2 RELATED WORK

2.1 Traditional Recommendation

In this info-explosion era, recommendation systems are an important tool to help users efficiently find what they are personally interested in. Due to the ability for tackling the data sparsity and cold-start problems, model-based collaborative filtering techniques, e.g., MF, are widely employed to recommend relevant items for users. MF, by decomposing a matrix into the production of several matrices (or the sum of some matrices) based on certain constraints, aims to discover some latent characteristics embedded in the original matrix. Many existing studies focus on effectively factorizing a given matrix into multiple components of interested entities. Du et al. [11] integrate users' feedback comments for the learning process to make more precise recommendations. Huang et al. [14] propose to jointly perform active feature querying and supervised matrix completion in order to train an effective classification model with least acquisition. ML-JMF [47] utilizes a multi-label aggregation method to effectively factorize the associated matrices. Wang et al. [53] propose a confidence-aware matrix factorization framework that introduces variance parameters for users and terms in the MF process. He et al. [15] propose an MF-based algorithm based on the **element-wise alternating least squares (eALS)** technique to seamlessly update the model while new entries are given. PMF [43] includes the adaptive prior on the model parameters for probabilistic MF on a large and imbalanced database. LPMF [35] utilizes local optimal points to estimate the parameters and downgrade the degree of over-fitting in local models. Wang and Ma [51] propose an adversarial MF-based method to tackle the missing-at-random and ineffective utilization issues in large datasets. Peng et al. [40] propose a novel non-negative matrix factorization method to learn local similarity and clusters in a mutually enhanced way. Yeh [55] introduce a model to increase the performance and reliability of a neural network which could directly enhance the recommendation.

Some prior MF-based studies constrain the decomposition results to non-negative value to address the issue of extreme data sparsity in a given rating matrix. RSNMF [31] regularizes a single element in the non-negative update process which depends on individual feature vectors instead of the whole matrix. Given a target dataset, semi-non-negative MF [45] aims to derive more effective low-dimensional representation than traditional clustering methods. Deep Semi-NMF learns latent features and interprets results based on the property of given dataset. DMF [6] derives different time factorization models by extending the idea of **non-negative MF (NMF)** [23] and **Linear Dynamics System (LDS)**. DynamicMF [42] automatically captures low-dimensional features for several applications. SDMF [50] first learns real-valued latent features by MF and then derives binary codes in the DMF framework to preserve the geometrical structures collectively hidden in users and items learned in a vector space. Koren et al. [21] describe the characteristics of the users and items with two decomposed low-dimensional matrices, and then inner product the two derived matrices to reconstruct and predict the user ratings. HSBMF [37] integrates multiple confounding factors for factorizing the matrix to predict the missing salary information in the salary matrix. Thai-Nghe et al. [38] make effective recommendations by implicitly including the latent factors and proposing a tensor factorization method for temporal effects. Abdi et al. [1] incorporate contextual information into MF approaches to improve the quality and accuracy for recommendation with large-scale datasets. Bin and Sun [4] propose a new algorithm based on matrix decomposition, which takes the reviews among users as auxiliary information. Park et al. [39] incorporate additional bias in MF and discuss the cold-start problem raised in the context of precise recommendations. Kawale et al. [20] apply **particle Thompson sampling (PTS)** in MF recommendation to automatically find the most relevant items and less recommended items. Wu et al. [54] analyze the deviation degree of ratings of users and items, and propose the concept of user and item centrality. Yu et al. [56] generalizes MF by incorporating the side information from user or item features and derive an efficient alternating minimization procedure for optimization. Jamali and Ester [19] explore a model-based approach for recommendation in social networks and incorporate MF techniques with the mechanism of trust propagation into the model. CoFactor [24] jointly decomposes the user rating and the item co-occurrence matrices jointly with shared latent features. FeatureMF [58] incorporates item features into the MF framework by projecting available attributes in item features into the same latent space of users and items. Tran et al. [46] propose a **regularized multi-embedding method (RME)** to simultaneously encapsulate several important pieces of information while deriving decomposition. RI-SGD [15] factorizes the implicit matrix with alternating least squares and weight regularization. **Stratified SGD (SSGD)** [13], a variant of SGD, considers more sufficient conditions to converge by stochastic approximation and regenerative process. Rendle et al. [41] apply the multilayer perceptron on collaborative filtering and MF, which revisit the issues of proper hyperparameter selection, simple dot product, and the proposed learned similarities.

Finally, we give a summarization of the utilized context information to facilitate the factorization process for recommendation. The social context [4, 37, 38, 51] emerges the relationship among users which could enhance the recommendation by potential similar preference. Several prior studies [9, 10, 14, 20, 31, 39, 40, 43, 45, 50] discuss the influence on the recommendation results with the similarity context which implied the probability of the identical interest. In addition, many works [6, 11, 15, 24, 46, 47, 54, 56, 58] discuss the correlation among feedback context and the purchases for rating prediction. The integration of location information [35, 53] also has been introduced to recommend the spatial-awareness relative items. Since the preference usually evolves, several existing models [6, 42] utilize the time factor to capture the dynamic variation for more precise recommendations.

2.2 CDR

CDRaims to enhance the recommendation quality by transferring knowledge from one (source) domain to another (target) domain. With the information from source domain, CDR improves the recommendations in the target domain. According to different transferring techniques, CDR methods can be divided into two categories: *content-based* and *transfer-based*. The content-based CDR methods extract common attributes of users and items and integrate user preferences from different domains. Berkovsky et al. [2] utilize cross-domain mediation of collaborative models to address the data sparsity problem. Chung et al. [3, 7] introduce an aggregation approach to solve the collaborative filtering issues and improve the recommendation accuracy. Winoto et al. [52] uncover the dependences of items and the relationships of user preferences across domains. Fernández-Tobias et al. [12] point out several difficulties in content-based CDR to apply in real scenario and propose to utilize social tags to bridge the relationships between two domains. Tan et al. [44] combine multi-type media information, such as media descriptions, text data, and ratings, to transfer user interest cross domains and make recommendations.

Transfer-based CDR methods address the data sparsity problem in recommendation by transferring knowledge from an auxiliary source domain to the target domain. Different from the content-based models, transfer-based approaches integrate user and item knowledge by transferring learning the neural networks. Hu et al. [17] utilize multilayer feedforward networks with dual connections and joint loss functions to enable dual knowledge transfer across domains by cross connections from one base network to another. Li et al. [30] develop a novel cross-domain collaborative filtering via **codebook-based knowledge transfer (CBT)**, which transfers a user-item rating matrix from an auxiliary domain to the target domain. Chen and Chen [10] propose a transfer learning algorithm, which employs the common users and items as a bridge to link different domains for knowledge transfer. Lu et al. [25] develop a CDR model to handle the problem that the data from the source domain are not consistent with the observations in the target domain. Li and Tuzhilin [28] construct a novel approach based on dual learning with the bidirectional latent relations between users and items. Liu et al. [26] utilize a graph collaborative filtering network to achieve bidirectional transfer learning for recommendations. ATLRec [29] adopts adversarial

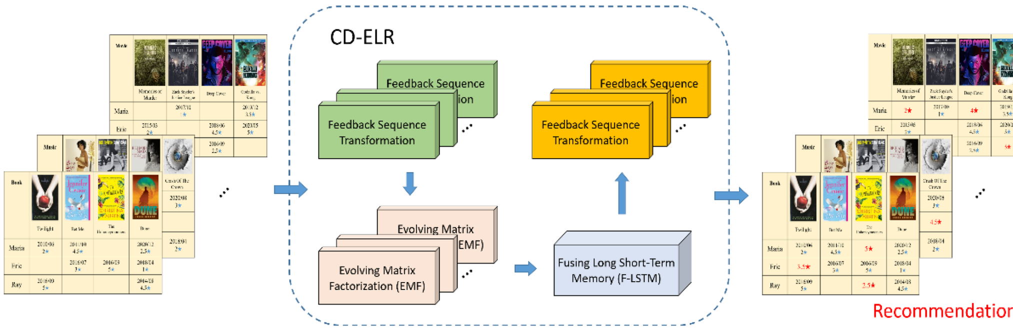


Fig. 3. The framework of the CD-ELR recommendation system.

learning to obtain domain-sharable features for CDR. DMF-CDR [22] combines the collaborative approach and multilayer perceptron structures to learn the representation and offer more precise cross-domain recommendations. Hong et al. [16] introduce a **cross-domain deep neural network (CD-DNN)** to jointly learn the features of users and items from both domains.

The cold-start issue is a critical problem for traditional recommendation systems. CATN [59] utilizes an attention mechanism to learn the aspect correlations extracted from review documents for transfer to the target domain. A **review-aware cross-domain recommendation algorithm (RACRec)** [18] solves the fully-cold-start problem by combining a user's preference vector with a product's feature vector to make the prediction. DCDIR [5] develops a meta-path method over insurance product knowledge graph and uses a feature mapping function to recommend products to cold-start users in the target domain. ACDN [32] extracts latent features by an aesthetic network and integrates them across domains to transfer users' aesthetic preferences.

3 CD-ELR RECOMMENDATION SYSTEM

Given l domains $D^1, \ldots, D^l$, let $U = \{u_1, u_2, \ldots, u_m\}$ be a set of users and $V^k = \{v_1, v_2, \ldots, v_n\}$ be a set of items in domain D^k. A rating record is a pair $(r_{ij}, t_{ij})^k$ where r_{ij} and t_{ij} denote the rating value and the rating time generated by user u_i for item v_j in domain D^k, respectively. A rating matrix $R^k(m \times n)$ consists of all rating records $(r_{ij}, t_{ij})^k$ being generated by $u_i \in U$ in domain D^k. Given a rating matrix R^k, the task of the recommendation system is to predict the rating value $r_{ij} \notin R^k$, i.e., the items that have not yet been rated by users.

Given rating matrices $R^1, \cdots, R^k, \cdots, R^l$ in multiple domains, the objective of CD-ELR is to enhance the recommendation quality in the domain D^k by leveraging the richer information from multiple domains $D^1, \ldots, D^k, \ldots, D^l$. We could tackle the cold-start and sparsity issues with the sharing knowledge from discovered characteristics of users in multiple domains effectively. The system framework of the proposed CD-ELR recommendation system for recommending users a set of items is shown in Figure 3. There are four components in the CD-ELR recommendation system: (1) feedback sequence transformation, (2) EMF, (3) fusing evolution learning, and (4) prediction and recommendation.

3.1 Feedback Sequence Transformation and EMF

Definition 1 (Feedback Matrix and Sequence). Suppose a matrix R^k including all rating records $(r_{ij}, t_{ij})^k$ in a domain D^k, a *feedback matrix* M_s^k of timestamp s (supposed $1 \leq s \leq \tau$) is a matrix consisting of all the rating value r_{ij}, where $t_0 \leq t_{ij} < t_0 + (s \times \Delta t), 1 \leq s \leq \tau$. t_0 is the start time and time interval Δt is the user-specified time interval. The sequence $M_1^k, \ldots, M_s^k, \ldots, M_\tau^k$ is called *feedback sequence* of domain D^k. An updated matrix $\Delta M_s^k = M_s^k - M_{s-1}^k$ contains all rating values r_{ij} that $t_0 + ((s-1) \times \Delta t) \leq t_{ij} < t_0 + (s \times \Delta t)$. The number of ratings in any M is denoted as $|M|$.

For preprocessing, each rating matrix R^k is transformed into a feedback sequence consisting of several smaller feedback matrices based on the rating time t_{ij} of rating records. The feedback sequence could effectively express the evolution of rating behaviors and relations of users and items for further learning processes to construct the recommendation system. The pseudo code is extended from [9, 10] and given in Algorithm 1. According to the user-specified time interval length, we decompose R^k into several smaller feedback matrices (lines 4–12, Algorithm 1). Notice that the feedback matrix is generated in an accumulated manner (lines 6 and 7, Algorithm 1). The feedback matrix M_{s+1}^k is derived by the union of the feedback matrix M_s^k and all the rating values of rating records with the current time interval. Finally, we concatenate all feedback matrices to derive a feedback sequence (line 9, Algorithm 1). After feedback sequence generation, CD-ELR factorizes the feedback matrix to discover the latent preferences of users and the characteristics of items with a novel MF.

Algorithm 1: Feedback Sequence Transformation

Input: R^k: a user rating matrix of domain D^k;
cur_time: current time;
Δt: user-specified interval length;
Output: M^k_Seq: feedback sequence in domain D^k

01: $M_0^k \leftarrow \emptyset$; $M^k\_Seq \leftarrow \emptyset$;
02: $t\_min = 0$; $t\_max = \Delta t$; // initialization
03: $s = 1$; // feedback matrix index
04: **while** ($t\_max \leq cur\_time$)
05: **for each** $(r_{ij}, t_{ij}) \in R^k$ **do**
06: **if** $t\_min \leq t_{ij} \leq t\_max$
07: $M_s^k \leftarrow M_{s-1}^k \cup r_{ij}$;
08: **end**
09: $M^k\_Seq \leftarrow \langle M^k\_Seq, M_s^k \rangle$;
10: $t\_min = t\_min + \Delta t$; $t\_max = t\_max + \Delta t$; // next
11: $s = s + 1$;
12: **end**
13: output M^k_Seq;

In this article, we extend the method [9, 10] and propose an effective method, called ***evolving matrix factorization (EMF)***, to efficiently decompose the latent features. As the feedback matrices in a feedback sequence are generated in a cumulative manner, EMF reuses the result obtained on a matrix to facilitate the computation on the next matrix.

Definition 2 (Preference and Characteristic Matrix). Let d-dimensional vectors p_i and q_j denote the vector of latent preference for user u_i and the vector of item characteristics for item v_j, respectively, in domain D^k. $P^k(m \times d)$ represents the preference matrix of all user latent vectors p_i, $1 \leq p_i \leq m$; $Q^k(d \times n)$ represents the characteristic matrix of all item latent vectors q_j, $1 \leq q_j \leq n$.

The main idea is extended from [9, 10] for factoring a matrix more efficiently and effectively. For M_s^k, if r_{ij} is not in ΔM_s^k, obviously, we could know that it appears in M_{s-1}^k. Thus, we directly use the previously calculated results in M_{s-1}^k. Since user u_i does not rate item v_j in time interval

$t_0 + ((s-1) \times \Delta t) \le t_{ij} < t_0 + (s \times \Delta t)$, the latent user preferences and item characteristics in M_s^k are the same as in M_{s-1}^k as defined in Equation (1).

$$\text{If } r_{ij} \notin \Delta M_s^k, \quad p_i^s, q_j^s = p_i^{s-1}, q_j^{s-1}. \tag{1}$$

However, if r_{ij} is in ΔM_s^k, EMF needs to derive a new latent user preference p_i^s and item characteristic q_j^s by minimizing the loss function in Equation (2).

$$\text{If } r_{ij} \in \Delta M_s^k,$$
$$p_i^s,\ q_j^s = \operatorname{argmin}\left(\frac{1}{\left|\Delta M_s^k\right|} \sum_{r_{ij} \in \Delta M_s^k} (r_{ij} - p_i^T q_j)^2 + \lambda_p \sum_{i=1}^{m} ||p_i||^2 + \lambda_q \sum_{j=1}^{n} ||q_j||^2 \right), \tag{2}$$

where λ_p and λ_q are two user-specified parameters. In this article, to achieve efficient derivation, we utilize the gradient decent for optimization in (2). We iteratively adapt p_i and q_j by Equation (3) and Equation (4):

$$p_i^\ell = p_i^{\ell-1} - \eta \cdot \left(-\frac{2}{\left|\Delta M_s^k\right|} \sum_{r_{ij} \in \Delta M_s^k} q_j (r_{ij} - p_i^T q_j) + 2\lambda_p p_i^{\ell-1} \right), \tag{3}$$

$$q_j^\ell = q_j^{\ell-1} - \eta \cdot \left(-\frac{2}{\left|\Delta M_s^k\right|} \sum_{r_{ij} \in \Delta M_s^k} p_i (r_{ij} - p_i^T q_j) + 2\lambda_q q_j^{\ell-1} \right), \tag{4}$$

where ℓ is the learning iteration index and η is the learning rate. Obviously, in Equations (3) and (4), η could control how fast that p_i and q_j change to reduce the mean squared error in Equation (2).

3.2 Fusing Evolution Learning

In the next step of CD-ELR, the discovered latent features (i.e., feedback matrix sequence) in all domains are fed into the learning module to extract representative evolution behaviors, also called *patterns*. To discover the latent user preferences, the series of factorized latent matrices is adopted for modeling short- and long-term evolutions simultaneously. We borrow and extend the LSTM model for evolution learning. Intuitively, the patterns of variation in both user preferences and item characteristics could be captured by the recurrent component in LSTM architecture.

We extend LSTM to propose a novel evolution learning model, called *fusing evolution learning*, to capture the complex evolving patterns of user interests and item characteristics. In each domain, we feed the sequence of latent vectors (including user preferences and item characteristics) factorized by EMF into the recurrent layer and output the hidden state step by step. The idea of fusing evolution learning is illustrated in Figure 4. The main learning component is the F-LSTM in each domain. Different from the traditional LSTM, the memory cell in the proposed F-LSTM, also called long-term memory, contains the historical interactions of user preferences and item characteristics that reflect the long-term evolution. The hidden states are also referred to as the short-term memory of the latent vectors. During the processing, memory cell (long-term memory) captures the long-term evolution by memorizing the order of variations and the relationships among users and items. We decompose the memory cell into short- and long-term interests and then decay the long-term interest by the effect of current variation with an interval-aware weight utility function which converts the evolution lapse into an appropriate weight.

Obviously, how much the short- and long-term interests would contribute to recommendations heavily depends on the degree of variations within the evolution. If a lot of new ratings appear in

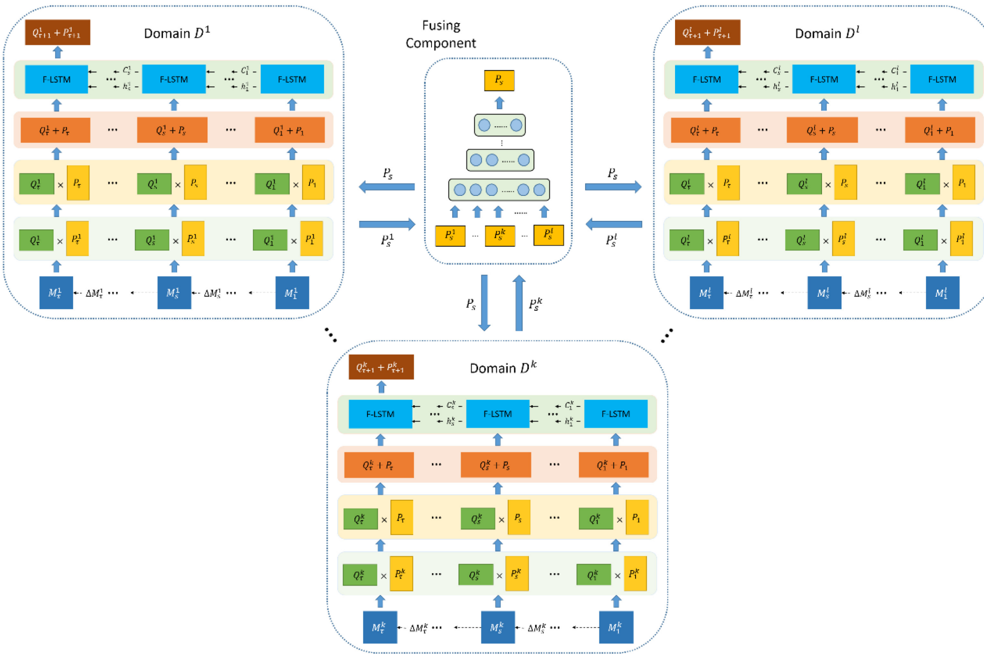


Fig. 4. An example of evolving matrix factorization.

an update matrix (e.g., ΔM_s^k), the user preferences and item characteristics may also have changed a lot. In this case, the short-term interest should contribute more in the next item recommendation. On the contrary, if no new rating occurs in the update matrix, the user preferences and item characteristics would not change. Therefore, the recommendation should follow more based on long-term interest. As mentioned above, the relationships among users and items are a critical issue for recommending the "right" items to the "right" users. The proposed fusing evolution aims to capture the evolution patterns in the user preference matrix and the item characteristic matrix in all domains separately, but still incorporate the cross-relation among users and items which needs to be effectively learned and estimated.

Consider a scenario of l domains $D^1, \ldots, D^l$. As shown in Figure 4, each domain D^k has a F-LSTM model to capture the evolutions of user preference and item characteristics. For each F-LSTM, at timestamp s ($1 \leq s \leq \tau$), the inputs consist of P_s^k (the preference matrix), Q_s^k (the characteristics matrix), $|\Delta M_s^k|$ (the number of rating in the updated matrix), C_{s-1}^k (memory cell for the interactions of preference and characteristic history), and h_{s-1}^k (previous output result). The objective functions of F-LSTM in domain D^k are as follows.

$$P_s = W(P_s^1 + \cdots + P_s^k + \cdots + P_s^l) + b \tag{5}$$

$$f_s^k = \sigma\left(W_f^k\left(P_s + Q_s^k\right) + U_f^k h_{s-1}^k + b_f^k\right) \tag{6}$$

$$i_s^k = \sigma\left(W_i^k\left(P_s + Q_s^k\right) + U_i^k h_{s-1}^k + b_i^k\right) \tag{7}$$

$$o_s^k = \sigma\left(W_o^k\left(P_s + Q_s^k\right) + U_o^k h_{s-1}^k + b_o^k\right) \tag{8}$$

$$Z_s^k = \sigma\left(W_Z^k\left(P_s + Q_s^k\right) + U_Z^k h_{s-1}^k + b_Z^k\right) \tag{9}$$

$$C_{s-1}^{k^l} = tanh\left(W_l^k C_{s-1}^k + b_l^k\right) \tag{10}$$

$$C_{s-1}^{k^s} = C_{s-1}^k - C_{s-1}^{k^l} \tag{11}$$

$$\tilde{C}_{s-1}^{k^l} = \gamma(|\Delta M_s^k|) \circ C_{s-1}^{k^l} \tag{12}$$

$$\tilde{C}_{s-1}^k = C_{s-1}^{k^s} + \tilde{C}_{s-1}^{k^l} \tag{13}$$

$$C_s^k = f_s^k \circ \tilde{C}_{s-1}^k + i_s^k \circ Z_s^k \tag{14}$$

$$h_s^k = o_s^k \circ tanh(C_s^k) \tag{15}$$

We first concatenate and feed all user preferences $P_s^1, \ldots P_s^k, \ldots P_s^l$ extracted from each domain into an MLP network, named as *fusion network*, to derive a fused user preference P_s in Equation (5). This step is called the *fusing step*. Notice that the dimensionality of the final fused vector P_s is identical to those in the preference matrix P_s^k. Then, in each domain D^k, we derive the necessary components of the proposed F-LSTM. The forget gates f_s^k, input gate i_s^k, and output gate o_s^k, decide how much information to be forgotten, input, and output in F-LSTM by Equations (6)–(8), respectively. Z_s^k is learned from the inputs P_s and Q_s^k, and the previous result h_{t-1}^k in Equation (9).

In this study, the memory cell in the proposed F-LSTM is decomposed into long- and short-term interests, $C_{s-1}^{k^l}$ and $C_{s-1}^{k^s}$ by Equations (10) and (11). This subspace decomposition plays a significant role in the model of fusing evolution learning. We segment the memory of the previous timestep into short- and long-term interests. Obviously, the short-term interest fits more for the current user preference; therefore, we keep the entire short-term interest $C_{s-1}^{k^s}$ for prediction. However, the long-term interest may not thoroughly express the preference trend of users; we calibrate properly based on the amount of pdated records at timestep s. Intuitively, the more updated records, the more variations occur during the user preference evolution. Hence, the dependence on the long-term interest does not play a significant role in the prediction on the current output. The decay gate γ in Equation (12) converts the number of updated entries in the updated user preference and item characteristics matrices into an appropriate weight in F-LSTM. The decay gate function $\gamma(x) = 1/\log(e + x)$ is a heuristic decaying function such that the larger the number of $|\Delta M_s^k|$, the lesser the effect of the long-term memory. Hence, we derive the decayed long-term interests $C_{s-1}^{k^l}$ by $\gamma(|\Delta M_s^k|)$ in Equation (12). The short-term and decayed long-term interests are combined to compose the adjusted long-term memory as $\tilde{C}_{s-1}^k$ in Equation (13). Finally, in Equations (14) and (15), we derive the new memory cell C_s^k and output h_s^k.

When training the F-LSTM learning model in the proposed CD-ELR system, we first construct the input instance for training, which is a set of vector sequences $\langle X^1, \ldots, X^s, \ldots, X^\tau$. When an input X^s is fed into the model, the output is h_s, where $1 \leq s \leq \tau$. h_s also could be considered as the prediction vector for the next timestamp. Hence, the error function is formulated by root mean square error (RMSE) between the predicted embedding vector h_s and the actual embedding vector X^{s+1} as shown below.

$$E = \sum_{s=1}^{\tau} \left(h_s - X^{s+1}\right)^2 \tag{16}$$

For the rating matrix $R^1, \ldots, R^k, \ldots R^l$ of the training dataset, we transform each R^k into a feedback sequence $M_1^k, \ldots, M_s^k, \ldots, M_\tau^k$. Note that the feedback sequence is generated in a cumulative manner. Then, we utilize the proposed EMF method to decompose each feedback matrix M_s^k into the user preference matrix P_s^k and the item characteristics matrix Q_s^k. In each timestamp s, preference matrices $P_s^1, \ldots, P_s^k, \ldots P_s^l$ from all domains are fused into one representative preference

Table 1. AMAZON [60] and FOXCONN Datasets

Dataset	#users	#items	#ratings	Rating range
AMAZON_Book	28,423	36,435	244,640	1–5
AMAZON_Music	16,872	16,386	95,890	1–5
AMAZON_Video	13,374	11,816	78,282	1–5
FOXCONN_Movie	8,746	16,508	543,403	1–5
FOXCONN_Drama	6,393	9,425	363,391	1–5

matrix P_s. Clearly, there is one F-LSTM in each domain D^k for evolution learning. The training instance is a vector sequence $\langle P_1 + Q_1^k, \ldots, P_s + Q_s^k, \ldots, P_\tau + Q_\tau^k$ for F-LSTM. Then, we leverage the mini-batch learning method to train the model on input instance until convergence. We take a gradient step to minimize the error based on the h_s and X^{s+1}. Notice that Adam optimizer is adopted to tune the learning rate for discovering the optimal parameter settings in CD-ELR.

3.3 Recommendation

After successfully training the F-LSTM model, we derive the final output result for predicting the user preferences and item characteristic in the next timestamp $\tau + 1$. For fusing evolution learning, the output is $P_{\tau+1}^k + Q_{\tau+1}^k$ in each domain D^k. CD-ELR decomposes $P_{\tau+1}^k + Q_{\tau+1}^k$ into two matrices: user preference matrix $P_{\tau+1}^k(m \times d)$ and item characteristic matrix $Q_{\tau+1}^k(d \times n)$. As shown in Equation (17), by multiplying $P_{\tau+1}^k$ and $Q_{\tau+1}^k$ together, we construct the prediction matrix $R^{k'}$ for domain D^k as follows.

$$R^{k'} = P_{\tau+1}^k \times Q_{\tau+1}^k. \tag{17}$$

The rating value $r'_{ij} \in R^k$ but $\notin R^{k'}$ is referred to as the prediction value of user u_i for item v_j. In domain D^k, with predefined number N, for the query about user u_i, CD-ELR lists the top-N prediction values in the ith row of $R^{k'}$ to make the item recommendation for u_i.

Finally, we discuss the constraints of the proposed method. First, the transformed feedback sequence should be in a cumulative manner. Since LSTMs are designed to process sequential data, the sequential nature introduces dependencies between the elements in the sequence. The cumulative feedback matrices in sequence could effectively facilitate the learning process of LSTM construction. Then, the rating scope should be fixed to a limited range. The rating scope is essential in MF since it affects the modeling and optimization process. Before applying MF, the input is necessary to normalize the ratings to a common scale. The fixed rating scope ensures that the optimization algorithm operates effectively and that the ratings are comparable across users and items.

4 PERFORMANCE EVALUATION

To validate the proposed CD-ELR in real-world environments, in this section, we conduct experiments on two real datasets. The AMAZON dataset is collected from Amazon [60], the world's largest online marketplace that provides products and services. In this article, as summarized in Table 1, we consider three domains: Book, Music, and Video in AMAZON, in experiments. AMAZON_Book, AMAZON_Music, and AMAZON_Video contain customer id, product id, rating, and review date, where the period of data collection is 12 months (i.e., from Jan. 2014 to Dec. 2014). In addition, we perform intensive experiments on a dataset from Foxconn Corporation (https://www.foxconn.com/), which is the world's largest electronics manufacturer. The FOXCONN dataset is collected from a TV streaming service.

The dataset, including two domains FOXCONN_Movie and FOXCONN_Drama, consists of user id, video id, rating, genre, title, and watching time. The period of data collection is 12 months (i.e.,

Table 2. Three Tasks for CDR from AMAZON [60] and FOXCONN Datasets

TASK	#common users	#items in source domain	#ratings in source domain	#items in target domain	#ratings in target domain
Task 1 (B2M)	7,643	7,289	75,492	4,147	28,547
Task 2 (B2V)	7,643	7,289	75,492	4,991	33,703
Task 3 (M2D)	4,606	6488	110,567	4,621	86,786

from Jan. 2018 to Dec. 2018). Due to the sparsity and the limitation of computational resource (memory limitation), by preprocessing, we filter out the users who interact with system less than threshold. Also, all items watched with less than threshold times are excluded. The summarization of numbers of users, items, and ratings of all datasets are given in Table 1.

Then, we use three tasks to evaluate CDRs. Task 1 and Task 2 are set as Book → Music (B2M) and Book → Video (B2V) on the AMAZON dataset, respectively. We adopt common users in three different domains in the AMAZON dataset, i.e., users have rating records in AMAZON_Book, AMAZON_Music, and AMAZON_Video. In Task 1 (B2M), we take AMAZON_Book as the source domain and AMAZON_Music as the target domain. In task 2 (B2V), we take AMAZON_Book as the source domain and AMAZON_Video as the target domain. There are 7,643 common users rating items in both domains in Tasks 1 and 2. Task 3 focuses on Movie → Drama (M2D) in the FOXCONN dataset. In Task 3 (M2D), we take FOXCONN_Movie as the source domain and FOXCONN_Drama as the target domain. There are 4,606 common users rating items in both domains. The details of the three tasks are shown in Table 2.

To evaluate the effectiveness of the proposed method, we use two well-known metrics: *root mean square error at d* (RMSE@d) and *mean absolute error at d* (MAE@d), respectively, where d is the size of latent vector d. Notice that we set $d = 30$, 50, and 100 to compare the different results of metrics at d. RMSE@d and MAE@d are used to evaluate the quality of the recommendation list. RMSE@d is the average root square difference between the predicted values and the actual values. Likewise, MAE@d is the average absolute difference between the predicted ratings and the actual ratings. Both metrics are widely utilized to evaluate the quality of predicted results by a model or an estimator.

$$RMSE@d = \sqrt{\frac{\sum_{i=1}^{n} (r_i - \hat{r}_i)^2}{n}} \tag{18}$$

$$MAE@d = \frac{\sum_{i=1}^{n} |r_i - \hat{r}_i|}{n} \tag{19}$$

In Equations (18) and (19), n is the number of total predicted ratings, r_i presents the predicted rating of the user for the ith item, and $\hat{r}_i$ is the actual rating of the user for the ith item.

We demonstrate the performance of the proposed CD-ELR model in comparison with several baseline methods, including the traditional **collaborative filtering (CF)**, MF, NMF [23], **robust matrix factorization (RMF)** [24], EMCDR [33], and CoNet [17] as follows.

- **CF:** The collaborative filtering method uses the derived similarity between users or items to recommend related items for potential users. However, both user- and item-based CF methods may suffer from sparsity and cold-start problems.
- **MF:** A traditional MF method that predicts user ratings by learning the latent factors via regressing over existing user-item ratings. Typically, the derivation of rating prediction is SGD-based or ALS-based methods.
- **NMF [23]:** A MF method focuses on the non-negative updated process depending on each involved feature and the whole feature matrices.

Table 3. Recommendation Performance of RMSE on Task 1 (B2M) of AMAZON Dataset

Task 1 (B2M)	Source domain (Book)			Target domain (Music)		
	RMSE@100	RMSE@200	RMSE@300	RMSE@100	RMSE@200	RMSE@300
CF	2.3189	2.3189	2.3189	2.6627	2.6627	2.6627
MF	0.8889	0.8686	0.8453	1.0078	0.8829	0.8569
PMF	0.8788	0.8248	0.8200	1.0229	0.9263	0.9244
RMF	0.8954	0.8825	0.8738	0.9885	0.9329	0.9156
EMCDR-LM	0.8511	0.8510	0.8172	0.7425	0.7410	0.7302
EMCDR-MLP	0.8583	0.8564	0.8522	0.7416	0.7411	0.7402
CoNet	0.8628	0.8628	0.8628	0.6392	0.6392	0.6392
CD-ELR	**0.6514**	**0.5563**	**0.5564**	**0.5428**	**0.5224**	**0.4969**

Table 4. Recommendation Performance with MAE on Task 1 (B2M) of AMAZON Dataset

Task 1 (B2M)	Source domain (Book)			Target domain (Music)		
	MAE@100	MAE@200	MAE@300	MAE@100	MAE@200	MAE@300
CF	1.7178	1.7178	1.7178	1.9200	1.9200	1.9200
MF	0.7123	0.6836	06721	0.8440	0.7541	0.7355
PMF	0.7829	0.6933	0.6756	0.8535	0.8129	0.8129
RMF	0.7165	0.6968	0.6913	0.8192	0.7955	0.7531
EMCDR-LM	0.7689	0.7594	0.7602	0.6548	0.6548	0.6548
EMCDR-MLP	0.7689	0.7681	0.7612	0.6546	0.5546	0.5546
CoNet	0.7894	0.7894	0.7894	0.6001	0.6001	0.6001
CD-ELR	**0.5977**	**0.5239**	**0.5339**	**0.5144**	**0.4907**	**0.4580**

— **RMF [24]:** A regularizing MF jointly decomposes the user-item interaction matrix and the item-item co-occurrence matrix with shared item latent factors.
— **EMCDR [33]:** An embedding framework adopts liner and multilayer perceptron as the mapping function to capture the relation between the source and the target domain. Both EMCDR-LM and EMCDR-MLP use MF and BPR as latent factor models applied with different mapping functions.
— **CoNet [17]:** CoNet uses multilayer feedforward networks with dual connections and joint loss functions to enable dual knowledge transfer across domains by cross connections from one base network to another.

All models are implemented in Python and Tensorflow, running on a workstation with i7-9700 CPU, 64GB RAM, and two Nvidia GeForce RTX3090 GPUs. The evaluation results of three tasks aim to demonstrate the superiority and generalization of the examined recommendation systems. For evaluating the recommendation results, we discuss several experiments and provide a real case study in the following sections.

4.1 Analysis on Overall Performance

In this section, we analyze the result in terms of RMSE@d and MAE@d as shown in Tables 3–8. To evaluate the performance, for traditional recommendation systems, including CF, MF, NMF, and RMF, we train them in each domain and derive the prediction results. Differently, for CDR systems, including EMCDR-LM, EMCDR-MLP, and CoNet, we train them in both of the source and target domains and derive the prediction results in both domains. Notice that, according to the metrics

Table 5. Recommendation Performance of RMSE on Task 2 (B2V) of AMAZON Dataset

Task 2 (B2V)	Source domain (Book)			Target domain (Video)		
	RMSE@100	RMSE@200	RMSE@300	RMSE@100	RMSE@200	RMSE@300
CF	2.3189	2.3189	2.3189	2.2473	2.2473	2.2473
MF	0.9889	0.9686	0.9453	0.9700	0.8777	0.8617
PMF	0.8788	0.8248	0.8200	0.9369	0.9318	0.9319
RMF	0.8954	0.8825	0.8738	0.9381	0.9352	0.9345
EMCDR-LM	0.8511	0.8510	0.8172	0.7355	0.7321	0.7359
EMCDR-MLP	0.8583	0.8564	0.8522	0.7402	0.7315	0.7337
CoNet	0.8628	0.8628	0.8628	0.6222	0.6222	0.6222
CD-ELR	**0.6308**	**0.5570**	**0.5531**	**0.5849**	**0.5748**	**0.5588**

Table 6. Recommendation Performance of MAE on Task 2 (B2V) of AMAZON Dataset

Task 2 (B2V)	Source domain (Book)			Target domain (Video)		
	MAE@100	MAE@200	MAE@300	MAE@100	MAE@200	MAE@300
CF	1.7178	1.7178	1.7178	1.6157	1.6157	1.6157
MF	0.7123	0.6836	06721	0.8255	0.7639	0.7585
PMF	0.7829	0.6933	0.6756	0.7956	0.7961	0.7929
RMF	0.7165	0.6968	0.6913	0.7942	0.7937	0.7931
EMCDR-LM	0.7689	0.7594	0.7602	0.6448	0.6410	0.6430
EMCDR-MLP	0.7689	0.7681	0.7612	0.6305	0.6329	0.6336
CoNet	0.7594	0.7594	0.7594	0.5864	0.5864	0.5864
CD-ELR	**0.5977**	**0.5139**	**0.5116**	**0.5144**	**0.4907**	**0.4780**

Table 7. Recommendation Performance of RMSE on Task 3 (M2D) of FOXCONN Dataset

Task 3 (M2D)	Source domain (Movie)			Target domain (Drama)		
	RMSE@100	RMSE@200	RMSE@300	RMSE@100	RMSE@200	RMSE@300
CF	2.5330	2.5330	2.5330	2.7542	2.7542	2.7542
MF	1.1763	1.1541	1.1539	1.3266	1.3025	1.3024
PMF	1.1376	1.1355	1.1354	1.6445	1.6420	1.6420
RMF	1.2541	1.2289	1.2273	1.7100	1.7023	1.7015
EMCDR-LM	1.3267	1.3235	1.3196	1.4865	1.4732	1.4706
EMCDR-MLP	1.2896	1.2788	1.2761	1.3323	1.3294	1.3294
CoNet	1.0742	1.0742	1.0742	1.2670	1.2670	1.2670
CD-ELR	**0.8362**	**0.8279**	**0.8270**	**0.9391**	**0.9266**	**0.9266**

in Equations (18) and (19), the MAE value is the lower bound of RMSE value, i.e., the RMSE value is always larger and equal than the MAE value.

Comparing with all models, including traditional and CDR systems, we observe that the proposed CD-ELR performs significantly better than all baselines due to its powerful cross-domain transferring ability. As shown in Tables 3 and 4, in Task 1 (B2M), CD-ELR significantly outperforms the traditional baselines of CF and MF models, and also performs better than the single-domain NMF and PMF models. Compared with CDR models, in the AMAZON_Book domain (the source domain in Tasks 1 and Task 2), the CD-ELR models outperform the EMCDR-LM, EMCDR-MLP, and CoNet models by 32%, 35%, and 36% on RMSE, respectively. Similarly, in the FOXCONN_Movie

Table 8. Recommendation Performance of MAE on Task 3 (M2D) of FOXCONN Dataset

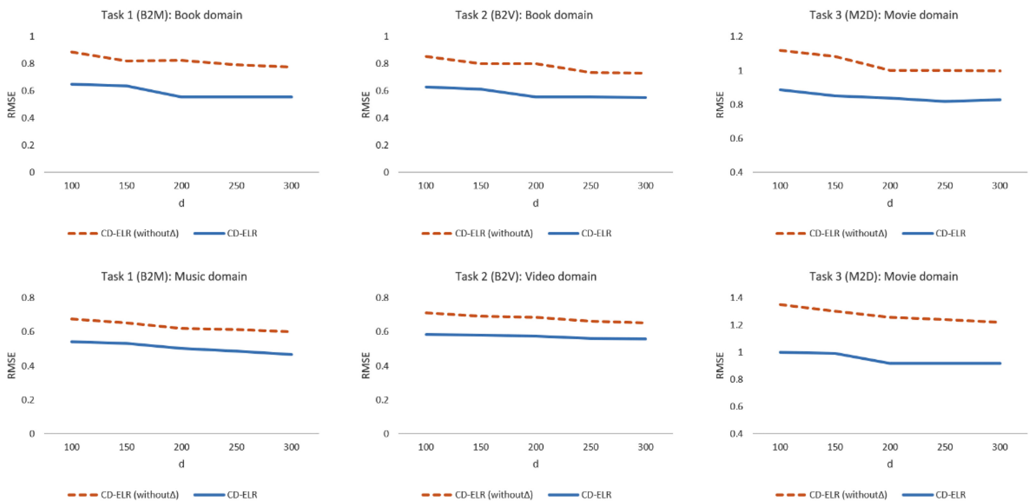
Task 3 (M2D)	Source domain (Movie)			Target domain (Drama)		
	MAE@100	MAE@200	MAE@300	MAE@100	MAE@200	MAE@300
CF	1.8474	1.8474	1.8474	1.9252	1.9252	1.9252
MF	0.8466	0.8251	0.8235	0.9893	0.9866	0.9866
PMF	0.8863	0.8801	0.8796	1.1286	1.1263	1.1258
RMF	0.9036	0.9021	0.9018	1.1327	1.1300	1.1300
EMCDR-LM	1.0173	1.0146	1.0132	0.9368	0.9271	0.9270
EMCDR-MLP	0.9983	0.9972	0.9953	0.9251	0.9230	0.9230
CoNet	0.8165	0.8165	0.8165	0.8857	0.8857	0.8857
CD-ELR	**0.7274**	**0.7260**	**0.7261**	**0.7998**	**0.7962**	**0.7958**

domain (the source domain in Task 3), CD-ELR models also outperform the EMCDR-LM, EMCDR-MLP, and CoNet models by 38%, 36%, and 24% on RMSE, respectively. The result shows that CD-ELR uses the information from the target domain to improve the recommendation quality in the source domain due to the proposed idea of multidirectional transferring.

It is not surprising that there is a performance improvement of CD-ELR over existing single-domain models since CD-ELR has more capability in transferring knowledge learned in all different domains. In each domain, CD-ELR learns and captures the user preference and item characteristic patterns simultaneously. In addition, we propose a fusing component to integrate user preference vectors in all domains. Hence, the results in Tables 3, 5, and 7 depict the significant improvement of recommendation quality. Specifically, in Task 3 (M2D), CD-ELR outperforms the single-domain recommendation models CF, MF, PMF, and RMF by 78%, 29%, 28%, and 19%, respectively. Also, we observe that the proposed CD-ELR leverages the irregularity of preference and characteristic of the transition context, which is more effective than the baselines in capturing the variation effects between consecutive feedback matrices.

Next, we discuss the impact of domain transferring. As shown in Tables 3 and 4, in the AMAZON_Music domain (the target domain) of Task 1 (B2M), the proposed CD-ELR has improved the performance more than 30% as compared to other CDR models in general. Similarly, as shown in Tables 5 and 6, in the AMAZON_Video domain (the target domain of Task 2 B2V), CD-ELR outperforms the EMCDR-LM, EMCDR-MLP, and CoNet models by 25%, 24%, and 11%, respectively. These results validate the effectiveness of the proposed cross-domain method CD-ELR which transfers the knowledge from data-rich domains to improve the performance of recommendations in the data-sparse domain. The same observations appear in the FOXCONN dataset. As shown in Tables 7 and 8, for both Movie and Drama domains in Task 3 (M2D), CD-ELR improves the recommendation quality over all baselines. For the Drama domain (the target domain), the recommendation results show the importance of preference evolution and domain transferring which influence the accuracy of recommendation.

In particular, CD-ELR decays the content of long-term interest by employing the variation between sequential elements. Different from previous studies, we improve the performance without feeding extra information or learning additional neural networks to affect the output prediction. Due to the sparse nature of the rating system, the same contexts (time difference) may have different impacts on different datasets. Instead of using the static weight decay function, in CD-ELR, we integrate the dynamic learning context weight for variation contexts. As shown in Tables 3–8, we find that with dynamic learning context weight by CD-ELR, the results in terms of accuracy are improved over existing CDR models. This indicates that the proposed CD-ELR is able to leverage



(a) Task 1 (B2M) (b) Task 2 (B2V) (c) Task 3 (M2D)

Fig. 5. The effectiveness of variation transition context in CD-ELR.

the dynamic learning context weight of variation transition contexts to enhance the accuracy. Finally, we observe that CD-ELR improves over 15% comparing with both EMCDR methods and CoNet, which indicates the integration of the user preference from domains and the incorporation of dynamic evolution significantly enhance the learning process for recommendation.

4.2 Analysis on Overall Performance

As shown in the above experimental results, CD-ELR outperforms all baseline models. In this section, we analyze the improvement in incorporating rating variation differences between successive feedback matrices. In CD-ELR, different from traditional RNN learning, there are two decompositions of memory cell to represent long-and short-term interests. We capture user long-term interest by memorizing not only the order of user's historical ratings, but also the difference in the number of ratings between two consecutive feedback matrices. Meanwhile, we dynamically learn to give a proper decay weight to each context in the long-term interests. Normally, the long-term interest affects the determination of a user's choices which heavily depend on the difference in the number of ratings between the current and the next feedback matrices. Intuitively, the more ratings, the more significant role the short-term interest plays. Hence, we use the variation of ratings to control the contribution of long-term interests in memory cell by learned decay gate.

The effectiveness of integrating the transition context of the number of rating variation is also investigated in this section. In order to explore the improvement of variation context on discounted long-term memory, we compare CD-ELR equipped with the proposed delta differences decay and without delta differences decay. As shown in Figures 5(a), (b), and (c), CD-ELR has significant improvement in RMSE with different d on average by 10%–30% on the three tasks as compared with CD-ELR (without Δ), which means that the context is critical for improving the recommendation performances.

4.3 The Ability and Influence of Cross-Domain Transfer

In this section, we evaluate the effectiveness of cross-domain communication. In this article, we introduce a novel concept of multidirectional transfer. In the proposed CD-ELR, we could

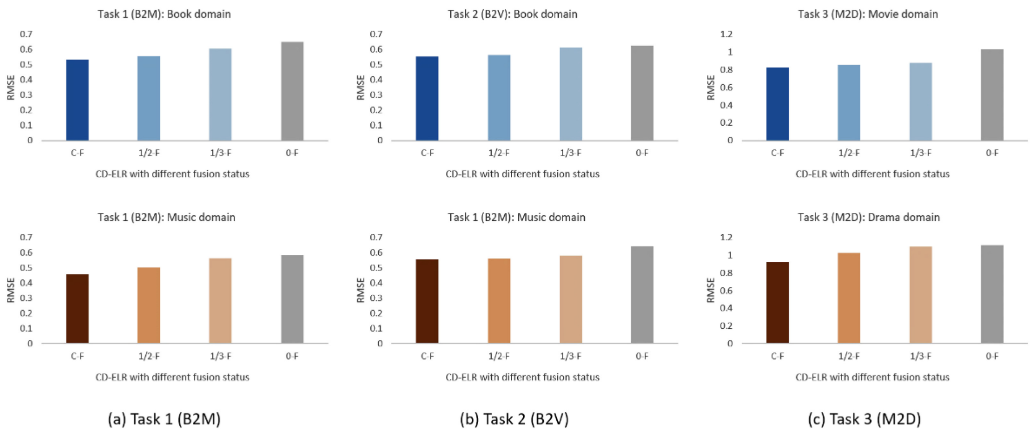


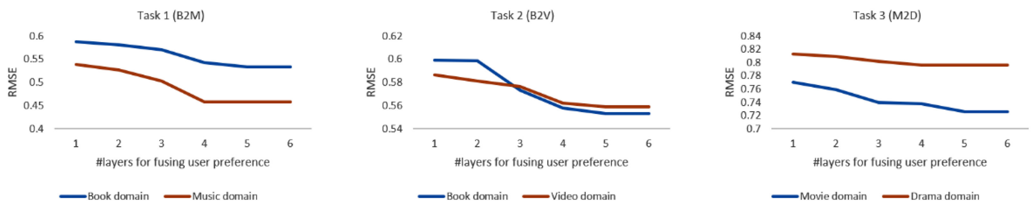
Fig. 6. Performance comparison with difference fusion status.

incorporate all user preferences in all domains. To analyze the ability of cross-domain transfer, we evaluate the CD-ELR by varying the number of cross-domain transfers in the fusing step. To observe the influence and ability of multidirectional transfer, CD-ELR is designed to equip in four cases of fusing steps.

- Complete-Fusion (C-F): CD-ELR combines all preference vectors in all domains of all timestamps. For each timestamp, we feed the preference vector in each domain into the MLP network in Equation (5) to derive the fused preference vector, i.e., $P_s = W(P_s^1 + \cdots + P_s^k + \cdots + P_s^l) + b, 1 \le s \le \tau$.
- Half-Fusion (1/2-F): For half of the timestamps, CD-ELR combines preference vectors in all domains, i.e., $s = \{1,\ 3,\ 5,\ 7, \ldots\}$.
- One-Third-Fusion (1/3-F): We only combine preference vectors from all domains in one-third of the timestamps, i.e., $s = \{1,\ 4,\ 7,\ 10, \ldots\}$.
- No-Fusion (0-F): CD-ELR does not combine any preference vector in each domain for each timestamp.

As shown in Figure 6, for Tasks 1–3, we observe that the recommendation quality is significantly improved. In both the source and the target domains, the RMSE metric increases when we decrease the number of preference vector fusions. CD-ELR achieves the best recommendation accuracy under the setting of complete fusion. Clearly, the cross-domain transfer affects the recommendation in the target domain more than in the source domain. Especially in Tasks 1 and 2 (shown in Figures 6(a) and (b)), we find that complete-fusion improves more than 30% in terms of accuracy than no-fusion. The result is reasonable since the information and knowledge extracted from the source domain usually are much richer than when extracted from the target domain. From the experiment, we demonstrate the ability of cross-domain transfer in the proposed CD-ELR model.

The influence of the fusion network for combining the preference vector in each domain is also discussed to conduct the ability of cross-domain transfer. We concatenate and feed all user preferences $P_s^1, \ldots P_s^k, \ldots P_s^l$ extracted from each domain into a fusion network to derive a fusing user preference P_s in Equation (5). Obviously, the MLP network dominates the fusion performance. In Figure 7, we show how the number of layers in the fusion network affects the performance of CD-ELR models in terms of the RMSE metric. Notice that the lower the RMSE value the better the accuracy performance. Definitely, the number of layers determines the model complexity. The



(a) Task 1 (B2M) (b) Task 2 (B2V) (c) Task 3 (M2D)

Fig. 7. Performance comparison with different number of layers in fusion network.

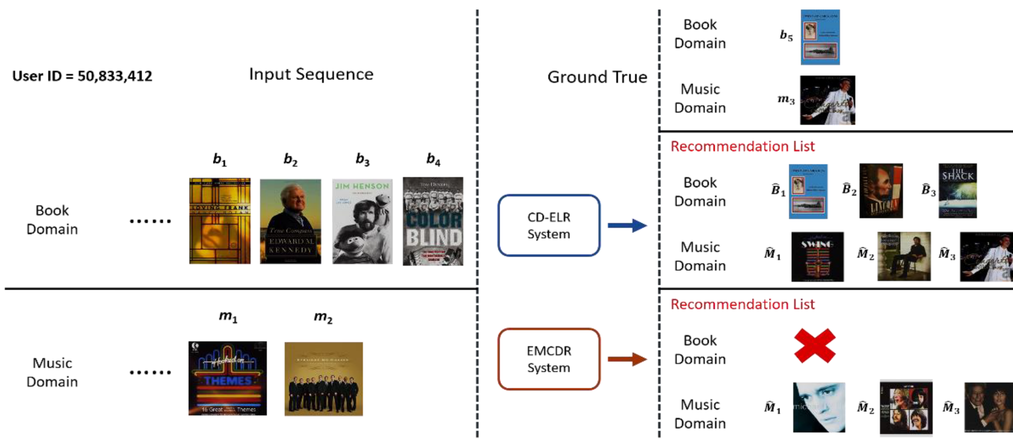


Fig. 8. Case study on effectiveness of cross-domain transfer.

deeper network structure fits the data better. One interesting point is that when we increase more than eight layers, the performance does not improve any further. Definitely, the number of neurons also dominates the impact of the final fusion results. We adopt more neurons in the fusion network when the network includes more layers. It is clearly found that CD-ELR has the better result when the fusion network has more neurons. The result is straightforward and makes sense, since the fusion network with few layers and neurons may not have the ability to capture the fusing pattern for precise recommendation.

4.4 Case Study

To demonstrate the practicability and performance of the proposed recommendation system, CD-ELR is applied on the real dataset AMAZON [60] to show the recommended results. We take several users' recommended lists in Tasks 1 and 2 to discuss the highlights of CD-ELR. Figures 8 and 9 show the top-N ranked books recommended by CD-ELR after the model is well trained for a specific user. Note that the particular user sequence is randomly chosen from the testing dataset.

First, we show the effectiveness of cross-domain transfer in CD-ELR, and we utilize buying behaviors in two different domains as an example. As shown in Figure 8, two rating sequences of the user (id = 50,833,412) in the book domain and the music domain are b_1 (Loving Frank), b_2 (True Compass: A Memoir), b_3 (Jim Henson: The Biography), b_4 (Color Blind) and m_1 (Hooked on Themes), m_2 (Holiday Sprits), respectively. Without any doubt, CD-ELR utilizes and communicates the knowledge between both domains. In the book domain, the top-three books we recommend are $\{\hat{B}_1$ (Twenty-Five Milk Runs), $\hat{B}_2$ (Lincoln), $\hat{B}_3$ (The Shack)$\}$. We could observe that the

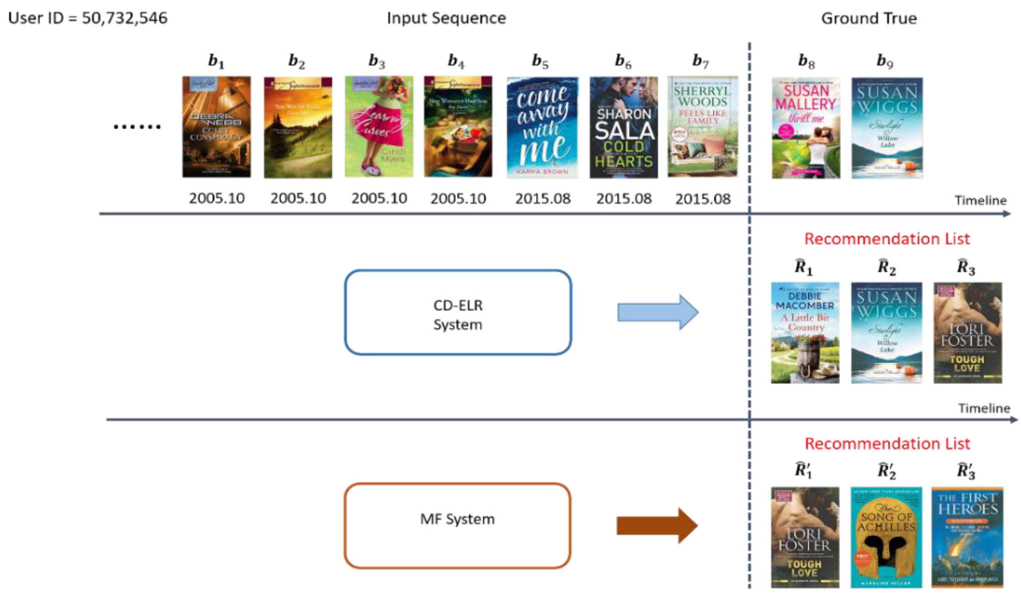


Fig. 9. Case study on influence of preference evolution.

information from the music domain enhances the rating prediction; CD-ELR hit the ground truth b_5 (Twenty-Five Milk Runs: The Biography). In addition, in the music domain, the top-three music recommendations from CD-ELR to user are $\{\hat{M}_1$ (Hooked on Swing), M_2 (Tuskegee), $\hat{M}_3$ (One Night in Central Park)$\}$. CD-ELR also hit the ground truth m_3 (One Night in Central Park) due to the assistance from the knowledge in the book domain. Nevertheless, as in the aforementioned discussion, traditional cross recommendation systems (i.e., EMCDR and Conet) only consider the knowledge transfer from the source domain to the target domain, but not vice versa. The information in the book domain could help the system recommend potential items more precisely in the music domain but could not improve the recommendation quality in the book domain. Hence, the recommendation results for books do not have any improvement from knowledge in the music domain as shown in Figure 8.

Then, we discuss the influence of preference evolution in the proposed CD-ELR. As shown in Figure 9, for a user (id = 50,732,546) in the Amazon database, the rating sequence consists of seven books: b_6 (Colby Conspiracy), b_7 (The Winter Road), b_8 (Learning Curves), b_9 (Not Without Her Son) which are rated in 2005, and b_{10} (Come Away with Me), b_{11} (Cold Hearts), b_{12} (Feels Like Family) which are read in 2015. When feeding the rating sequence, for our CD-ELR, the top-three recommendation results ranks as $\{\hat{R}_1$ (A Little Bit Country), $\hat{R}_1$ (Starlight on Willow Lake), $\hat{R}_3$ (Tough Love: An Anthology)$\}$. We find that CD-ELR could hit the ground truth; the actual succeeding books are $\{b_8$ (Thrill Me), b_9 (Starlight on Willow Lake)$\}$. Obviously, all books in the recommendation list of CD-ELR are very similar; $\hat{R}_1$, $\hat{R}_2$, and $\hat{R}_3$ belong to the romance genre. However, with the same rating sequence, traditional MF and CF recommendation systems will recommend $\{\hat{R}'_1$ (Tough Love: An Anthology), $\hat{R}'_2$ (The Song of Achilles), $\hat{R}'_3$ (The First Heroes)$\}$ which belong to the romance and literature genres as the top-three ranking. Clearly, the result could not match the ground truth. From the example, we could find the influence of preference evolution for recommendation. The recommended result shows the ability and performance of the proposed CD-ELR on the sequential recommendation with preference evolution.

5 CONCLUSION

Owing to the popularity of e-commerce and online activities, a recommendation system plays a more and more important role for users quickly finding out potentially interesting items from massive amounts of merchandise. In this study, to tackle the cold-start and sparsity issues, a novel cross-domain recommendation model, CD-ELR, is developed to communicate the information from different domains by integrating the MF and RNN. We introduce an EMF to efficiently decompose the rating matrix in a dynamic manner. Furthermore, a novel F-LSTM model is developed to fuse the discovered evolution patterns of user interests in multiple source domains to enhance the recommendation result of a target domain with several optimization techniques. The experimental results show that CD-ELR outperforms prior state-of-the-art recommendation baselines on several evaluation metrics. Finally, we conduct some case studies on a real-world dataset to demonstrate the practicability of the proposed CD-ELR.

REFERENCES

[1] M. Abdi, G. Okeyo, and R. Mwangi. 2018. Matrix factorization techniques for context-aware collaborative filtering recommender systems: A survey. *Computer and Information Science* 11, 2 (2018).

[2] S. Berkovsky, T. Kuflik, and F. Ricci. 2007. Cross-domain mediation in collaborative filtering. In *Proceedings of the 11th International Conference on User Modeling*. 355–359.

[3] S. Berkovsky, T. Kuflik, and F. Ricci. 2007. Distributed collaborative filtering with domain specialization. In *Proceedings of the 1st ACM Conference on Recommender Systems*. 33–40.

[4] S. Bin and G. Sun. 2021. Matrix factorization recommendation algorithm based on multiple social relationships. *Hindawi Mathematical Problems in Engineering* 2021 (2021), Article ID 6610645, 1–8.

[5] Y. Bi, L. Song, M. Yao, Z. Wu, J. Wang, and J. Xiao. 2020. DCDIR: A deep cross-domain recommendation system for cold start users in insurance domain. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1661–1664.

[6] F. Chua, R. Oentaryo, and E. Lim. 2013. Modeling temporal adoptions using dynamic matrix factorization. In *Proceedings of the IEEE 13th International Conference on Data Mining (ICDM'13)*. 91–100.

[7] R. Chung, D. Sundaram, and A. Srinivasan. 2007. Integrated personal recommender systems. In *Proceedings of the 9th International Conference on Electronic Commerce*. 65–74.

[8] L. Chen, J. Zheng, M. Gao, A. Zhou, W. Zeng, and H. Chen. 2017. TLRec:Transfer learning for cross-domain recommendation. In *Proceedings of the International Conference on Big Knowledge (ICBK'17)*. 167–172.

[9] Y. Chen and W. Lee. 2024. An evolving preference-based recommendation system. *IEEE Transactions on Emerging Topics in Computational Intelligence (early access)*, 2024. DOI: https://doi.org/10.1109/TETCI.2023.3343998

[10] Y. Chen and Y. Chen. 2024. A novel virtual-communicated evolution learning recommendation. *Industrial Management & Data Systems* 124, 1 (2024), 416–441.

[11] Y. Du, C. Xu, and D. Tao. 2017. Privileged matrix factorization for collaborative filtering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*. 1610–1616.

[12] I. Fernández-Tobías and I. Cantador. 2014. Exploiting social tags in matrix factorization models for cross-domain collaborative filtering. In *Proceedings of the 1st Workshop on New Trends in Content-based Recommender Systems*. 34–41.

[13] R. Gemulla, E. Nijkamp, P. Haas, and Y. Sismanis. 2011. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. 69–77.

[14] S. Huang, M. Xu, M. Xie, M. Sugiyama, G. Niu, and S. Chen. 2018. Active feature acquisition with supervised matrix completion. In *Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'18)*. 1571–1579.

[15] X. He, H. Zhang, M. Kan, and T. Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM Conference on Research and Development in Information Retrieval (SIGIR'16)*. 549–558.

[16] W. Hong, N. Zheng, Z. Xiong, and Z. Hu. 2020. A parallel deep neural network using reviews and item metadata for cross-domain recommendation. *IEEE Access* 8 (2020), 41774–41783.

[17] G. Hu, Y. Zhang, and Q. Yang. 2018. CoNet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 667–676.

[18] Y. Jin, S. Dong, Y. Cai, and J. Hu. 2020. RACRec: Review aware cross-domain recommendation for fully-cold-start user. *IEEE Access* 8 (2020), 55032–55041.

[19] M. Jamali and M. Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the 4th ACM Conference on Recommender Systems (RecSys'10).* 135–142.

[20] J. Kawale, H. Bui, B. Kveton, L. Thanh, and S. Chawla. 2015. Efficient Thompson sampling for online matrix-factorization recommendation. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS'15).*

[21] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* 42 (2009), 30–37.

[22] H. Kuang, W. Xia, X. Ma, and X. Liu. 2021. Deep matrix factorization for cross-domain recommendation. In *Proceedings of the IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC'21).* 2171–2175.

[23] D. Lee and H. Seung. 2000. Algorithms for non-negative matrix factorization. In *Proceedings of the 13th Conference on Advances in Neural Information Processing (NIPS 2000).*

[24] D. Liang, J. Altosaar, L. Charlin, and D. Blei. 2016. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys'16).* 59–66.

[25] Z. Lu, Z. Dou, J. Lian, X. Xie, and Q. Yang. 2015. Content-based collaborative filtering for news topic recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 217–223.

[26] M. Liu, J. Li, G. Li, and P. Pan. 2020. Cross domain recommendation via bi-directional transfer graph collaborative filtering networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 885–894.

[27] D. Lee and H. Seung. 2000. Algorithms for non-negative matrix factorization. In *Proceedings of the 13th Conference on Advances in Neural Information Processing (NIPS 2000).*

[28] P. Li and A. Tuzhilin. 2020. DDTCDR: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining.* 331–339.

[29] Y. Li, J. J. Xu, P. P. Zhao, J. H. Fang, W. Chen, and L. Zhao. 2020. ATLRec: An attentional adversarial transfer learning network for cross-domain recommendation. *Journal of Computer Science and Technology* 35 (2020), 794–808.

[30] B. Li, Q. Yang, and X. Xue. 2009. Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence.* 2052–2057.

[31] X. Luo, M. Zhou, Y. Xia, and Q. Zhu. 2014. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics* 10, 2 (2014), 1273–1284.

[32] J. Liu, P. Zhao, F. Zhuang, Y. Liu, V. S. Sheng, and J. Xu. 2020. Exploiting aesthetic preference in deep cross networks for cross-domain recommendation. In *Proceedings of The Web Conference.* 2768–2774.

[33] R. Mehta and K. Rana. 2017. A review on matrix factorization techniques in recommender systems. In *Proceedings of the 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA'17).* 269–274.

[34] T. Man, H. Shen, X. Jin, and X. Cheng. 2017. Cross-domain recommendation: An embedding and mapping approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence.* 2464–2470

[35] W. Ma, Y. Wu, M. Gong, C. Qin, and S. Wang. 2017. Local probabilistic matrix factorization for personal recommendation. In *Proceedings of the 13th International Conference on Computational Intelligence and Security (CIS'17).* 97–101.

[36] H. Ma, H. Yang, M. Lyu, and I. King. 2008. SoRec: Social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management (CIKM'08).* 931–940.

[37] Q. Meng, H. Zhu, K. Xiao, and H. Xiong. 2018. Intelligent salary benchmarking for talent recruitment: A holistic matrix factorization approach. In *Proceedings of the IEEE 18th International Conference on Data Mining (ICDM'18).* 337–346.

[38] N. Thai-Nghe, L. Drumond, T. Horváth, A. Nanopoulos, and L. Thieme. 2011. Matrix and tensor factorization for predicting student performance. In *Proceedings of the 3rd International Conference on Computer Supported Education (CSEDU'11).*

[39] H. Park, J. Jung, and U. Kang. 2017. A comparative study of matrix factorization and random walk with restart in recommender systems. In *Proceedings of the IEEE 2017 International Conference on Big Data (BigData'17).* 756-765.

[40] C. Peng, Z. Zhang, Z. Kang, C. Chen, and Q. Cheng. 2021. Nonnegative matrix factorization with local similarity learning. *Information Sciences* 562 (2021), 325–346.

[41] S. Rendle, W. Krichene, L. Zhang, and J. Anderson. 2020. Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys'20).* 240–248.

[42] N. Sorkunlu, D. Luong, and V. Chandola. 2018. Dynamicmf: A matrix factorization approach to monitor resource usage in high performance computing systems. In *Proceedings of the IEEE 2018 International Conference on Big Data (BigData'18).*

[43] R. Salakhutdinov and A. Mnih. 2007. Probabilistic matrix factorization. In *Proceedings of the ACM 20th International Conference on Neural Information Processing Systems (NIPS'07).* 1257–1264.

[44] S. Tan, J. Bu, X. Qin, C. Chen, and D. Cai. 2014. Cross domain recommendation based on multi-type media fusion. *Neurocomputing* 127 (2014), 124–134.

[45] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller. 2017. A deep matrix factorization method for learning attribute representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 3 (2017), 417–429.

[46] T. Tran, K. Lee, Y. Liao, and D. Lee. 2018. Regularizing matrix factorization with user and item embeddings for recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM'18)*. 687–696.

[47] J. Tu, G. Yu, C. Domeniconi, J. Wang, G. Xiao, and M. Guo. 2018. Multi-label answer aggregation based on joint matrix factorization. In *Proceedings of the IEEE 18th International Conference on Data Mining (ICDM'18)*. 517–526.

[48] T. Wallace, C. Godwin, J. Thomson, and A. Tjernlund. 2019. The definitive guide to selling on amazon. *BigCommerce* (2019), 1–229.

[49] C. Wang, Q. Liu, R. Wu, E. Chen, C. Liu, X. Huang, and Z. Huang. 2018. Confidence-aware matrix factorization for recommender systems. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*. 434–442.

[50] J. Wu, F. Luo, Y. Zhang, and H. Wang. 2020. Semi-discrete matrix factorization. *IEEE Intelligent Systems* 35, 5 (2020), 73–83.

[51] B. Wang and W. Ma. 2020. Matrix factorization based on batchnorm and preference bias. In *Proceedings of the IEEE 13th International Conference on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI'20)*. 942–946.

[52] P. Winoto and T. Tang. 2008. If you like the Devil Wears Prada the book, will you also enjoy the Devil Wears Prada the movie? A study of cross-domain. *New Generation Computing* 26 (2008), 209–225.

[53] Q. Wang, P. Tan, and J. Zhou. 2018. Imputing structured missing values in spatial data with clustered adversarial matrix factorization. In *Proceedings of the IEEE 18th International Conference on Data Mining (ICDM'18)*. 1284–1289.

[54] Z. Wu, H. Tian, X. Zhu, and S. Wang. 2018. Optimization matrix factorization recommendation algorithm based on rating centrality. In *Proceedings of the 3rd International Conference on Data Mining and Big Data (DMBD'18)*, Lecture Notes in Computer Science Vol. 10943. 114–125.

[55] W. Yeh. 2017. A squeezed artificial neural network for the symbolic network reliability functions of binary-state networks. *IEEE Transactions on Neural Networks and Learning Systems* 28, 11 (2017), 2822–2825.

[56] H. Yu, H. Huang, I. Dihillon, and C. Lin. 2017. A unified algorithm for one-cass structured matrix factorization with side information. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*. 2845–2851.

[57] J. Zhang and C. Chow. 2016. CRATS: An LDA-based model for jointly mining latent communities, regions, activities, topics, and sentiments from geosocial network data. *IEEE Transactions on Knowledge and Data Engineering* 28, 11 (2016), 2895–2909.

[58] H. Zhang, I. Ganchev, N. S. Nikolov, Z. Ji, and M. O'Droma. 2021. FeatureMF: An item feature enriched matrix factorization model for item recommendation. *IEEE Access* 9 (2021), 65266–65276.

[59] C. Zhao, C. Li, R. Xiao, H. Deng, and A. Sun. 2020. CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 229–238.

[60] Amazon review datasets. Retrieved from https://jmcauley.ucsd.edu/data/amazon/

Received 23 July 2022; revised 8 June 2023; accepted 18 June 2023