Computer Assisted Methods in Engineering and Science
31(3): 389–415, 2024, doi: 10.24423/cames.2024.900

This article belongs to the *Special Issue on AI-Based Future Intelligent Networks and Communications Security* edited by Dr. S. Kumar, Dr. G. Mapp, Dr. A. Bansal and Dr. K. Cengiz

STN-CDRS: Sentiment Transfer Network for Cross-Domain Recommendation Systems

Nikita TANEJA[1)*], Hardeo K. THAKUR[2)]

[1)] *Computer Science and Technology, Manav Rachna University, Faridabad, India*

[2)] *School of Computer Science Engineering and Technology, Bennett University, Greater Noida, India; e-mail: hardeokumar@gmail.com*

[*] *Corresponding Author e-mail: nikita.taneja@gmail.com*

In enterprise environments, the products may come from a variety of categories or domains. Users may engage with entities in one domain, but not in the others when they are presented with multiple domains. Such users are referred to as "cold-starters" in other domains. The primary difficulty in cross-domain recommendation systems is to efficiently transfer user's latent information based on their engagements in one domain into the other domains. The advancements in recommendation systems have inspired us to develop review-driven recommendation models that utilize cross-domain knowledge transfer and deep learning models. This work proposes a sentiment transfer network specifically designed for providing recommendation in cross-domain (STN-CDRS). The novelty of the work lies in the user rating enrichment mechanism, which is done by extracting latent information from user review data to fill sparse rating matrix. This enrichment uses previously developed RNN-Core method for efficiently learning user reviews. The reviews provided by the users are used to enrich sparse data across domains. This enrichment allows two things: alleviates the cold start problem and allows more intersecting users across domains to bridge the gap while learning. This work empirically demonstrates its efficiency by iteratively updating over the baseline recommendation models in terms of MAE (mean absolute error), RMSD (root mean squared deviation), precision and recall measures with other state-of-the-art-review-aided cross-domain recommendation systems.

Keywords: cross-domain recommendations, sentiment transfer network, user reviews, deep learning, knowledge transfer.

1. Introduction

In real-world contexts, recommender systems [1, 2] are increasingly confronted with data sparsity [3]. Because of insufficient knowledge, recommender systems might struggle to provide recommendations for new goods or users. This is called the cold start problem [4]. Without enough data, the underlying recommender model cannot be accurately calculated, and users' preferences cannot be accurately predicted. Transferring information from other domains or activities [5] and incorporating heterogeneous external knowledge [6] have been found to reduce the data sparsity problem in recommender systems.

To deal with sparsity, several collaborative filtering-based recommender systems have been developed, including transfer learning. Transfer learning extracts common information from a domain with sparse data [7] and applies it to the target domain to enhance recommendations. This strategy can greatly improve the performance of the underlying recommender system [8]. Cross-domain recommender systems are those that employ transfer learning techniques. These systems are created with the goal of making suggestions in the target domain based on data from the source domain. However, with cross-domain recommender systems, the most pressing challenge is how to identify the common information shared across the two different domains. The following are some of the issues that cross-domain recommendation systems face:

- **Data sparsity caused by inconsistent domain features**: There are usually no explicit features, just extracted latent characteristics. Furthermore, because the observed sparse ratings may not completely reflect a user's preferences, characteristics retrieved from the same person in two different domains could be inconsistent. As a result, creating a suitable feature space is quite difficult.
- **Data sparsity caused by domain heterogeneity**: Domain adaptation strategies can align extracted hidden domain features from overlapping entities. However, there is no direct association between recovered latent characteristics from non-overlapping entities, and their features are diverse.
- **Data sparsity caused by partially overlapping entities**: Various partially overlapping entities can constitute a relatively small percentage of the overall entities under the target domain.

To illustrate these problems, as shown in Fig. 1, consider two domains: domain A and domain B, each containing user and item features along with user–item interactions. In real business practices, common users in domain A very seldom overlap with domain B, as shown in Fig. 1. This means that there are very few users who engage with items shared across the domains, as there is little or even zero overlap between the entities of the two domains, i.e., each entity belongs to one domain only.

It becomes very difficult for a recommender system to predict user preferences from domain A in domain B due to the lack of connected users or items across the domain, which a recommender system may use to transfer knowledge regarding the user into the other domain. There are various methods for transferring

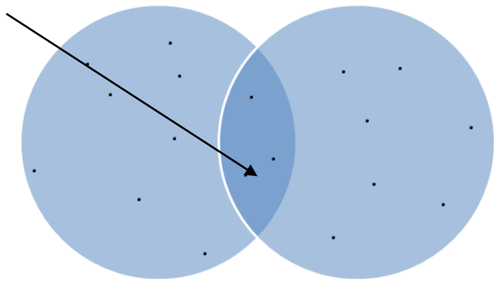

Fig. 1. Illustration of challenges faced by cross-domain recommendation systems due to sparsity, inconsistency and partially overlapping entities.

knowledge across the domains as indicated by our literature review presented in the next sections. Nevertheless, based on whether and how the entities for every domain overlap, the strategies for extracting knowledge and transfer it differ significantly. Prior cross-domain recommender systems often presume that no elements are common to both domains and that all elements are one-to-one mapped. Non-overlapping approaches are more likely to derive common information from group-level user behavior. Even though many of these strategies were created to address specific scenarios, they are unable to integrate knowledge from several entities when new information becomes available. The source domain and the target rating matrices are often factorized together in completely overlapping techniques, and then the entities' characteristics are retrieved. Constraints placed on each entity guarantee that these traits are identical in the target domain, thus allowing them to serve as a bridge for knowledge transfer.

In realistic scenarios, on the other hand, these entities seldom conform to the “full” or “nothing” overlap assumption; instead, many lie somewhere in the middle, as illustrated in Fig. 1. Also, in fully overlapping methods, large-scale factorization is computationally expensive, potentially increasing exponentially with the size of the data [33], as discussed in earlier works.

Constraints across the source and target domains are established using information about overlapping items. These constraints are frequently related to the characteristics of the items in the domain. Domain divergence occurs when there are slight changes in rating patterns across the two domains, even when the user is still the same. Knowledge transfer will struggle to yield good results and prediction accuracy will suffer, if item restrictions or constraints are not carefully managed. Thus, to this end, this work tries to solve this issue by utilizing external knowledge from reviews given by users to reduce data sparsity across the domain. Reviews given by users in domain A can also be used to identify user preferences in domain B.

This paper proposes a sentiment transfer network specifically designed for providing recommendation in a cross-domain (STN-CDRS). This model enriches sparse recommendations by utilizing user review data across domains from reviews. This enrichment uses previously developed RNN-Core method, which efficiently learns user reviews. The reviews provided by users are used to enrich sparse data across domains. This enrichment addresses two challenges: it alleviates the cold start problem and allows the bridging of gaps between intersecting users across domains during the learning process. Furthermore, a novel transfer learning-based approach for generating cross-domain recommendations through an improved deep neural network architecture is developed. In contrast to matrix factorization, which is quite slow [33] and cannot learn complex relationships between users, items and ratings, deep learning combined with transfer learning methods can achieve this. STN-CDRS enables smooth cross-domain knowledge transfer by establishing cross connections between base networks. STN-CDRS has a cross-connected multi-layer deep neural network and a loss function shared across domains. The STN-CDRS model is trained using the M-SGD optimizer. The approach improves recommendation performance in both domains simultaneously and outperforms all baseline models. The suggested model provides the following advantages over earlier approaches:

1. It enriches the data from word embeddings of user reviews instead of relying on explicit information from user–item interactions present in the dataset. This allows to capture complex user–item relationships more effectively.
2. The enrichment helps in solving cold start problems and data sparsity issues. With more data available for learning, the model can improve its performance, providing better learning.
3. It iteratively updates on the recommendation model using deep transfer learning model, facilitating the transfer of user interests across domains. This knowledge transfer method can enhance performance in both domains simultaneously if trained properly.
4. The method uses an implicit preference transfer function for transferring user preferences across domains, and this transfer function can be applied across any domain for transferring knowledge across different domains.

Our contributions in this paper are as follows:

1. This work proposes the application of data enrichment methods to resolve sparsity and cold start problems. This includes efficient deep learning architecture and knowledge transfer function to improve recommendation performance across domains.
2. This work also provides a novel knowledge transfer method that iteratively updates the recommendation model to facilitate the cross-domain transfer of user preferences using user review embedding.

3. This work also develops a novel deep neural network architecture capable of learning from review embedding and providing recommendations in a cross-domain context.
4. We demonstrate that the proposed STN-CDRS can outperform existing baseline techniques and improves recommendation accuracy across a variety of domains and experimental scenarios.

The rest of the paper is organized as follows: Sec. 2 provides a review of existing methods in recommender systems, covering knowledge transfer methods, deep learning-based recommender systems, and approaches using review/text data for deep learning and knowledge transfer, and Sec. 3 provides the methodology used to build the STN-transfer network. In Sec. 4, the paper presents experiments and their results to demonstrate the benefits of knowledge transfer-based recommender systems. Section 5 concludes the paper and discusses interesting research directions for the future.

2. Related works

This section primarily summarizes three types of research related to our work. These cross-domain recommendation systems employ knowledge transfer techniques, deep learning-based algorithms and deep learning-based recommender systems using review data.

2.1. Knowledge transfer methods

The advantages of information transfer-based strategies are twofold: 1) to capture user interests, knowledge transfer tasks uses user–item interaction data, and 2) knowledge transfer can assist in the integration of knowledge from various tasks and domains into a global user–item repository, which can then be tailored to different scenarios in recommendation system. This includes mitigating the cold start problem and performing cross-domain knowledge transfer.

Recently, methods employing knowledge transfer models have achieved remarkable success across a wide range of works in the field of natural language processing (NLP) [9]. Such algorithms are often developed on unstructured massive data to acquire universal language understandings. Next, these representations are fine-tuned on downstream tasks to enable knowledge transfer. Knowledge transfer models may be trained to acquire deep context-aware language representations using word-vectors [10]. The resulting language models have shown to be quite effective in a variety of applications, including but not limited to natural language inference, question answering, as well as recommendations [11] and context adaptation [12].

One may divide studies that use knowledge transfer methods to improve recommendation accuracy into two groups in the domain of recommender systems:

1) feature-driven models and 2) fine-tunable models. Knowledge transfer models often employ feature-driven models to acquire features for users and entities from the meta information (for example, item content and knowledge sources) [13]. The fine-tunable methods, on the other hand, employ data from user–item interactions to train a deep transferable neural model, which is then further fine-tuned for subsequent recommendation tasks [14]. For instance, in [15], an attentional adversarial transfer learning network for cross-domain recommendation (ATLRec) used adversarial learning (AL) models to gain an understanding of user–item engagements in both the source and the much-needed target domain. This technique used attention mechanism to learn user–item engagements by leveraging shared users with an engagement history, with the goal of better connecting item entities in other domains and capturing cross-domain item–item relationship to enable domain-shared knowledge learning.

2.2. Deep learning methods

In past few years, with the resurgence of deep learning methods, many deep learning-based systems have been introduced to further improve knowledge transfer in recommender systems.

Numerous studies [16–18] used deep machine learning algorithms to create user–item profiles using a rating matrix, such as de-noising auto-encoders or Boltzmann machines. In [19], the authors introduced a novel word embedding approach based on neural networks. They started by creating a rating matrix using explicit ratings and latent quasi-responses. This matrix was then used to train a deep neural network (DNN) to learn a low-dimensional space for incorporating user–item entities. Furthermore, the authors introduced a new cost function based on binary cross-entropy for effective training. In another study, [20], by incorporating the latest items among temporal and semantic spaces, the authors proposed a deep neural word-embedding mechanism providing a top-N cross-domain recommendation system.

Autoencoder with attention mechanism (AAM) [21] provides a recommendation model that uses an autoencoder with an attention layer to transmit and merge knowledge between multiple domains for learning the cross-domain ratings. Embedding and mapping cross domain recommendation (EMCDR) [22] clearly maps user representations using linear mapping and multi-layer perceptron to map the source to the target domain. Deep cross-domain cross-system recommendation (DCDCSR) [23] further expands EMCDR [22] using a multi-layer fully connected neural network to directly transfer user interpretations from multiple domains. Different from the existing CDR and CSR approaches, a novel DCDCSR [23] framework generates benchmark factors that address challenges spanning multiple domains and systems. Using knowledge solely from user–item matrices, domain adaptation recommendation (DARec) [24] can discover and transmit conceptual rating patterns of user's representations in multiple domains using a shared domain classifier. The user–item engagement matrix is used in preference propagation graph network PPGN [25] to record the mechanism of user interest propagation. Employing an attention-based method, many features for every user and each item from the user reviews are collected, enabling understanding of aspect connections across domains. Furthermore, cross-domain recommendation for cold-start users via aspect transfer network (CATN) [26] uses supplementary opinions from similar users to improve a user's aspect latent information. A parallel information-sharing network for shared-account cross-domain sequential recommendations π Net [27] is a model developed to provide mutual cross-domain consecutive recommendations.

In [44], a method is proposed to generate a generalized user representation incorporating user information across domains in sequential recommendation, even with few to no common users across domains. For each domain, an autoencoder was used to predict the origin domain of a generated user representation. Authors in [45] created a recommendation method for giving suggestions designed for flash-sales, which can accommodate user predilections for specific time periods based on item variations in the availability for sale in e-commerce. However, where consumer preferences are relatively consistent, this technique loses its advantages. In [46], authors evaluate the weaknesses and merits of graph-embedding models compared with conventional models for various recommendation tasks, this study concludes conventional methods outperform existing graph-embedding method for predicting user recommendations, they further suggested considering the trade-off between the two before final deployment to users.

Since most of these approaches solely employ rating information or user–item interactions, they fall under the umbrella of collaborative filtering. Moreover, deep learning algorithms can also be used to extract text data to provide recommendations across domains.

2.3. Methods using text/review information

As this work considers using review information to improve the cross-domain recommender systems, there are several articles that have focused on this area. Let us briefly review these works.

Sentiment review pattern mapping (SRPM) [28] uses a multilayer perceptron (MLP) to capture the nonlinear mapping function across domains to describe the user sentiment pattern. Furthermore, the authors employ smoothed latent Dirichlet allocation (SLDA) on these sentiment-tagged datasets to model the sentiment review patterns of users. SLDA and MLP based mapping method are used to model user's SRP and map it to the target domain to make recommendations for cold-start users. Sentiment analysis-based review feature mapping (SARFM) [29] uses the semantic orientation CALculator (SO-CAL) – a lexicon-based sentiment analysis approach to establish relationships across domains using an MLP-based mapping approach. To provide suggestions for cold-start users, this technique analyzes user's lexical alignments and translates them to the target domain.

By merging the sentiment information inherent in user evaluations in diverse areas, CDR-SAFM [30] is founded on sentiment classification and implicit feature mapping. Using sentiment analysis on customer review data, the method was able to generate cross-domain suggestions. To create the implicit opinion review characteristics, latent Dirichlet allocation (LDA) is applied to characterize the user's lexical information. The cross-domain quasi translation function is obtained using a perceptron that can transmit the user's opinion attributes. It is also probable that rating errors will persist in CDR-SAFM, and it does not give an effective way to pre-process rating data. CD-DNN [31] creates a single low-dimensional projection for representing user features, allowing the user model to be optimized alongside item characteristics from other domains. It also proposes a recommendation model that incorporates opinions from textual reviews and a rating matrix to improve prediction accuracy by learning features of users and items from multiple domains simultaneously.

DeepCGSR [32] proposes a matrix decomposition-based method for handling user–item ratings and enabling cross-domain sentiment of user reviews. However, as noted in previous research [33], matrix factorization-based methods are quite slow, which is indicated by much lower computational time taken by neighborhood-based methods compared to matrix factorization methods. On average, neighborhood-based methods are about six times faster, which is significantly high. This speed difference, even when considering precision and recall, makes it challenging to use matrix factorization methods instead of neighborhood-based methods, as it is more difficult to scale matrix factorization methods than neighborhood-based methods. Future works can focus on developing more efficient matrix factorization methods that can match the time efficiency of neighborhood-based methods. Moreover, most of the algorithms used are based on MLP learners; however, the deep learning has progressed a lot making MLP a bit outdated. In addition, review-based transfer solutions have outperformed traditional interaction-based procedures. Nevertheless, researches on the latter ones have a number of flaws that must be addressed.

In [34], collaborative cross network (CoNet) by establishing cross connections between base networks, promotes knowledge transfer across fields. CoNet solves the data sparsity problem by using a sparse target user–item interaction matrix that can be rebuilt using knowledge direction from a source domain. DDTCDR [35] through a dual learning method efficiently transfers user latent information across domain pairings. The inner product is replaced by a deep network to learn any function from input in NCF [36]. In addition, this approach generalizes matrix factorization and replaces the inner product with a deep neural network to capture complex user–item interactions. It uses an MLP to model 9 the user–item engagement function.

3. Proposed work

This work assumes that the same user exist across both the source and target domains. Users who are present in one domain may also exist in the other domain. Thus, domains share the same users. One domain is distinguished as the source domain (D_A) and the other the target domain (D_A), without loss of generality. Let $U = \left\{User_1, User_2, ..., User_{|U|}\right\}$ represent the shared set of users between the source (D_A) and target (D_B) domains, i.e., overlapping users. Also, $D_A = \left\{item_1, item_2, ..., item_{|D_B|}\right\}$ and $D_B = \left\{item_1, item_2, ..., item_{|D_B|}\right\}$ are the sets of items available in the source and target domains, respectively. Reviews given by users in the source domain are $\mathit{\Gamma}_A = \{r_{U1}, r_{U2}, ..., r_{D_A}\}$ from the source domain, while $\mathit{\Gamma}_B = \{r_{U1}, r_{U2}, ..., r_{D_B}\}$ are the reviews given by users from D_B, where r_{Ui} is a set of reviews given by $User_i$ from the respective domain. Likewise, it is assumed that $T_I = \left\{r_{I1}, r_{I2}, ..., r_{I|D_A|}\right\}$ is the dataset of items reviewed by users in the target domain, where each item I_j in the dataset has a corresponding set of reviews r_{Ij} associated with it. Also, $\mathit{\Gamma}_s$ and $\mathit{\Gamma}_t$ are matrices of user ratings for the source (D_A) and target (D_B) domains, respectively.

3.1. Problem formulation

This paper aims to analyze the sentiment information derived from users domains (D_A) and (D_B). This is done using common users (|U|) as a bridge to map underlying latent knowledge, which can then be transferred from D_A to D_B. This knowledge transfer is done to resolve the estimation of cold-start users in D_B (target domain), a challenge arising from data sparsity. For this purpose, this work introduces STN-CDRS, a cross-domain recommendation algorithm. STN-CDRS uses sentiment information-based dataset enrichment and latent knowledge transfer from D_A to D_B. The algorithm proceeds in following steps: 1) extraction of user sentiment information in both the domains about all the items, and 2) using these reviews to enrich the dataset using a knowledge transfer method to share latent information from domain A to domain B, and lastly 3) using a deep learning model to provide cross-domain recommendations.

In the first step, an improved RNNCore method, developed in our earlier work [37], is combined with the word embedding model from CoreNLP-based method [38], to identify user review embedding, and this is explained in

Subsec. 3.2. User opinions as ratings in domain D_A are denoted as $\mathit{\Gamma}(A)_U^{\mathit{\Psi}}$, while user opinions in domain D_B are divided into $\mathit{\Gamma}(B)_U^{\mathit{\Psi}}$, where $\mathit{\Psi}$ is the sentiment score obtained from the RNNCore-based sentiment analysis. Then, we aim to find the sentiment embeddings of reviews using a mapping function ($\mathit{\Omega}$), as explained in Subsec. 3.3. This function, mapping the user opinions, assumes that opinions on any item are linked via shared users, and the reviews given by the user on such an item are the outcome of a features mix of the user's latent preferences and sentiment score. In other words, the comments usually serve as sentiment scores as a feature. The mapping function $\mathit{\Omega}$ can be used to classify user review information given by the user's rating of the item. Moreover, to improve the impact of sentiment classification and further improve the overall algorithm's accuracy, this work employs an efficient sentiment classification algorithm, namely RNNCore. When a user prefers an item, their ultimate opinions become more favorable, resulting in a high score provided by the user for that particular item.

In the next step, this mapping function is created to model the cross-domain review feature embedding. It assumes an implicit link between user review characteristics in D_A and D_B, and uses the knowledge transfer function in order to capture this link. STN-CDRS employs this latent knowledge transfer function to distinguish the relationships among distinct sentiments to reduce mutual interference problem among user reviews throughout the knowledge transfer process. The knowledge transfer function is trained across distinct sentiments through data processing and the employment of mutual users in the two domains whose review features are available. This is done to prevent the absence of mutual latent user features during the mapping process.

Lastly, the model is used to recommended new users (cold start) in D_B. By using STN-CDRS, we can obtain the target domain's matching implicit features and utilize them to influence the final recommendation outcomes. Changing the mapping properties and features of these cold-start users can yield varying effects on their ratings in the target domain.

3.2. RNNCore sentiment analysis

RNNCore model, shown in Fig. 2 and developed earlier in [37], is used to provide sentiment scores for user reviews about items in any given domain, and this model is trained over word embeddings, which are numerical representations of the text. This model can identify its own set of features from the word embedding and hence does not require a manual selection of features. Moreover, it can utilize pre-trained word embeddings made available by CoreNLP [38]. As a result, it can use input corpus and pre-trained word vectors to generate a rating, as depicted in the image below. Unlike polarity (-1–1), the RNNCore is pro-

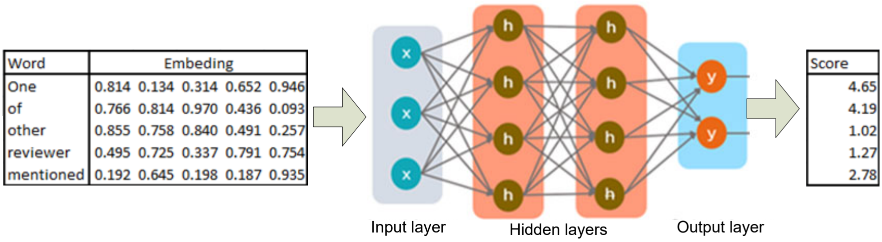

FIG. 2. Sentiment analysis in RNNCore using word embedding [37].

grammed to provide a sentiment score in the range of 1 to 5. This broader range allows the RNN output to be employed for cross-domain recommendation tasks.

The RNNCore extracts "ratings" from the "user reviews" within the review dataset. It preprocesses the reviews using de-punctuation, lowercase, and eliminating stop words. The trained RNNCore model is directly used to map a user's item review to a rating in the range of 1 to 5.

3.3. Domain enrichment and knowledge transfer

Domain enrichment is a process to reduce the sparsity problem, as shown in Fig. 3. The domain enrichment involves adding more data (enriching), specifically word embeddings of user reviews instead of explicit information present in the dataset between user–item interactions. By enriching the dataset, complex user–item relationships can be captured, thereby addressing cold start problems and data sparsity. The availability of more data for learning enhances the overall learning process. An essential topic to be tackled in this research is how to properly transfer latent features across domains. Previous recommendation algorithms relied on user opinions and did not fully use the sentiment data in user reviews. STN-CDRS presumes that it can map latent information between domains using a mapping function in order to bridge D_A and the D_B. Factorization approaches can be used to connect user–item pairings within a mutual space and employ implicit features to represent users and entities, which are crucial tools for

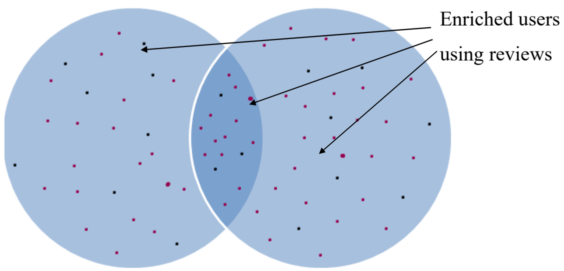

FIG. 3. The dataset enrichment process using sentiment-derived reviews.

providing recommendations. While it is reasonable to extend matrix factorization methods utilizing implicit user rating feature embedding approaches to simulate implicit engagements between users and items, this can also be done for other feature information as well, this approach may become suboptimal during the cross-domain recommendation process due to performance issues of the matrix factorization methods, as discussed in [33]. Applying simple neural network models across all domains sparsely is not sufficient enough to obtain optimal results.

One key assumption is that users interested in domain D_A should also have comparable interests in domain D_B. This implies that it is possible to learn user interests from both domains D_A and D_B simultaneously. Therefore, to gain a better knowledge of user preferences from D_B, we may merge user sentiments with target domain user interest information from D_B. Moreover, if user interest information is employed from D_A into D_B simultaneously, we may be able to improve recommendation performance.

To effectively extract user preferences from reviews, RNNCore is utilized within the knowledge transfer framework, mapping user reviews to user ratings using feature embedding.

STN-CDRS introduces the analysis of user interest through two aspects: current-domain preferences r, which record user engagements and predict user ratings in domain D_A itself, and cross-domain preferences $\grave{r}$, which consider user activities in the source as well as target domain D_B. A new parameter called transfer rate ω is introduced, denoting the relative relevance of the two components in predicting user preferences. We suggest the following method for estimating user ratings in domain pairings $(D_A,\, D_B)$:

$$\grave{r}_A = \omega \cdot \mathit{\Pi}_B\left(\mathit{\Omega} \times \mathit{\Theta}_{u_A}, \mathit{\Theta}_{i_A}\right) + (1-\omega) \cdot \mathit{\Pi}_A\left(\mathit{\Theta}_{u_A}, \mathit{\Theta}_{i_A}\right), \tag{1}$$

$$\grave{r}_B = \omega \cdot \mathit{\Pi}_A\left(\mathit{\Omega}^T \times \mathit{\Theta}_{u_B}, \mathit{\Theta}_{i_B}\right) + (1-\omega) \cdot \mathit{\Pi}_B\left(\mathit{\Theta}_{i_B}, \mathit{\Theta}_{i_B}\right), \tag{2}$$

where $\mathit{\Theta}_{u_A}$, $\mathit{\Theta}_{i_A}$, $\mathit{\Theta}_{u_A}$, $\mathit{\Theta}_{i_A}$ are the latent information about users $U \in |U|$ and items $\left\{i_1, i_2, ..., i_{|D_{A,B}|}\right\}$ and $\mathit{\Pi}_A$, $\mathit{\Pi}_B$ are recommendations generated for D_A and D_B, respectively.

Each equation's first part $\omega \cdot \mathit{\Pi}_B\left(\mathit{\Omega} \times \mathit{\Theta}_{u_A}, \mathit{\Theta}_{i_A}\right)$ and $\omega \mathit{\Pi}_A\left(\mathit{\Omega}^T \times \mathit{\Theta}_{u_B}, \mathit{\Theta}_{i_B}\right)$ quantifies single-domain interests based on user and item attributes, whereas the second part expresses cross-domain user interests based on the implicit mapping function $\mathit{\Omega}$, which captures domain diversity. This adaptive learning model can be divided into two independent recommendation models if $\omega = 0$, i.e., if there no mutual users between D_A and D_B; however, if $\omega = 0.5$, the model converts into a knowledge transfer model and merges the two domains into a single global recommendation model that can provide recommendation across both domains D_A and D_B. Depending on the quantity of user reviews, typically, ω should take a positive value between 0.1 and 0.25 for individuals who appear in both domains, indicating that self-domain preferences play a considerable job in understanding user patterns.

3.4. DNN architecture

The overall architecture of STN-CDRS network is presented in Fig. 4. It is made up of an RNNCore-based review encoder for embedding user reviews

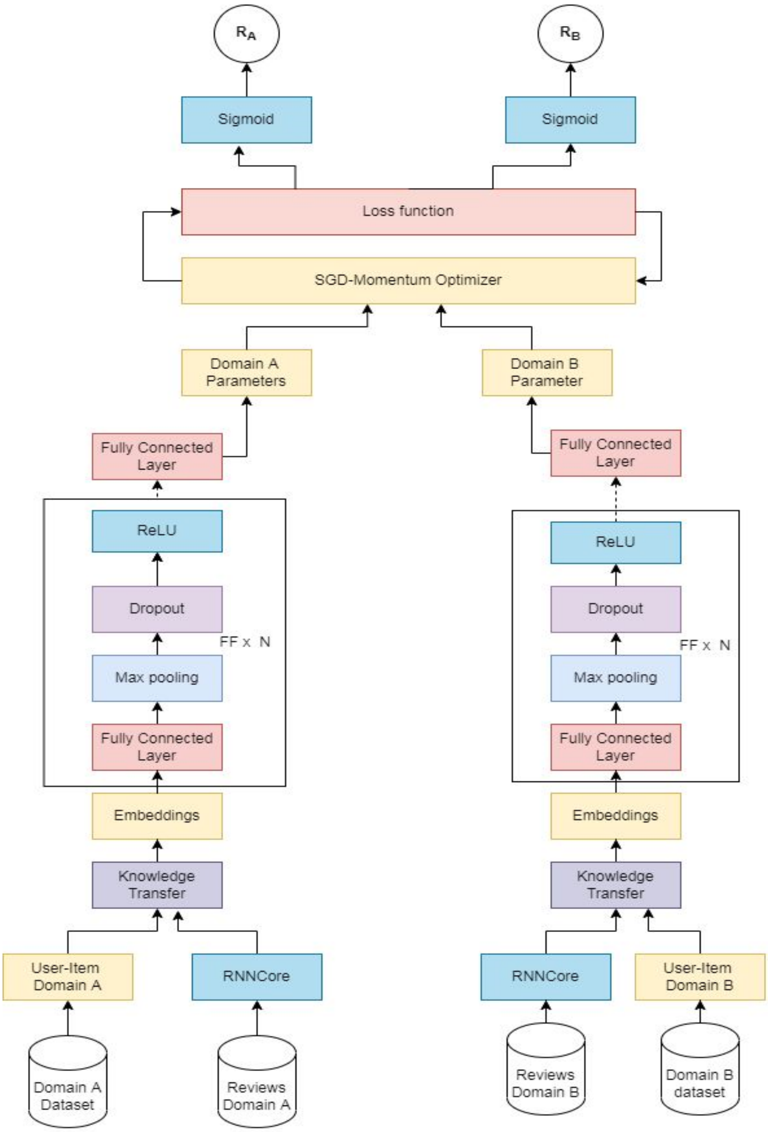

Fig. 4. Architecture of the STN-CDRS.

into embedding vectors with sentiment information. While D_A (source) and D_B (target) share the same set of users, as shown in Fig. 3, the items cannot be shared because of the different domains. Thus, STN-CDRS uses the RNNCore sentiment to transform users' latent review features, which are then utilized upstream.

The knowledge transfer-based method mentioned in Subsec. 3.3 is then utilized to enrich the missing values in the rating matrices of both domains. STN-CDRS takes the observed word embedding for every user and item as its input, maps each vector into a low-dimensional representation, and then outputs a reconstitution layer to recover the embedding, allowing the unavailable or missed ratings to be computed for recommendation purposes. The RNNCore and the knowledge transferring method have been already explained earlier in Subsecs. 3.2 and 3.3, respectively. Further layers of the STN-CDRS network are explained as follows.

3.4.1. User sentiment embedding layer. The user sentiment embedding layer, also known as text convolution layer, is the first to generate user review embeddings. The embedding layer's job is to create a low-dimensional latent information embedding space. Each word present in a user review corresponds to $\Sigma(w) \rightarrow |W|$, where the word dictionary $|W|$ contains every word present in the review corpus. Every word w can be mapped into a $\forall$-dimensional dense vector using an embedding function $\Sigma(w)$. This word embedding approach is used by STN-CDRS to incorporate the interpretations of both user and item reviews. Opinions and item information are stored as word embedding matrix in the embedding layer, each of which carries its own semantics. Moreover, when placing the embedding layer, all of a user's reviews are linked together in a document called d_u. An embedding matrix $\Pi^u_{i:l}$ is generated for user u using this embedding layer represented as:

$$\Pi^u_{i:l} = \left[\Omega\left(\Gamma^U_1\right), \Pi\Omega\left(\Gamma^U_2\right), ..., \Pi\Omega\left(\Gamma^U_l\right)\right], \tag{3}$$

where Γ^U_l is j-th review word written by user $User_i$ review Γ_u and latent mapping function $\Omega\left(\Gamma^U_j\right)$ results the matching $\forall$-dimensional embedding vector for the review word and Π is a join operation.

3.4.2. Feed forward layers. To approximate the learning of user–item preferences, feedforward layers are employed in the STN-CDRS model. These layers are built using a mapping function $\Upsilon = F(X, H)$ where the parameters H values are trained to optimize the function. As the knowledge is transferred using the mapping function, this function uses the features X to estimate the intermediate blocks required to achieve the function F. These models are known as feed-forward models because the information flows from the input layers to the output without any feedback links, meaning that the model's outputs do not flow back into it. Given that the application involves textual data rather than images, deep feedforward layers were chosen over convolutional or other types of layers. Deep feedforward layers aid in the identification of hidden user preferences in the input data.

3.4.3. Max pooling. By conducting nonlinear down sampling, pooling simplifies the result, and the number of parameters that the network has to learn decreases. This technique is beneficial in feedforward networks as the outbound connections frequently receive similar input. Max pooling selects the maximum element from the input feature matrix covered by the filter. As a consequence, the max pooling layer's result is a feature map that contains the most important features from the previous layers. In the case of average pooling, this pooling produces down-sampled or pooled feature maps that highlight the most present feature in the patch rather than the average presence of the feature. For NLP applications, max pooling has been proven to perform better in practice than average pooling.

3.4.4. Dropout layer. To train the fully connected layers and reduce the overfitting problem, dropout layers have been added. Their fundamental job is to stop some of the updating of neuron weights in the hidden layer. This simplifies the complex co-adaption among the neurons.

3.4.5. Fully connected layers. Along with convolution layers, fully connected (FC) layers are the building blocks of most neural networks. They are the units (layers) from which most neural networks are built. Fully connected layers are multiplication parameters that connect one layer of a neural network to successive layers, resulting in each layer's weights being a linear combination of the weights of the preceding layer. However, they differ from convolutional layers in how they link two layers of a neural network.

Fully connected layers, as the name implies, connect every neuron in the output layer to every neuron in the input layer. This work uses FC layers instead of convolution layers because of two reasons. Firstly, the data type is textual, not images, for which convolution operation is most suited, and secondly, in the input layer, FC layers can be used to describe any generic pattern. FC layers excel at identifying global patterns in a neural network layer because of this. As a result, they are ideal for wrapping up all of the patterns identified by the prior layers. Furthermore, because the layers at the final levels of a neural network are generally tiny, the large number of parameters inside a fully connected layer is less of an issue when it comes to learning. However, because of the large number of weights and associated issues such as overfitting, FC layers may not always be the most favored choice in neural network systems. Convolution layers [39], on the other hand, link the output layer to a previous layer using a universal "filter", resulting in a significantly lower number of parameters to learn compared to a fully-connected layer. Consequently, highly adapted convolution layers can be used for detecting local features that may exist anywhere in the input data. Instead of training each layer one by one to identify a similar set of features, the network learns a single layer that is shared by all nodes. FC layers, on the other hand, are employed to identify certain global configurations of characteristics identified by the net's lower layers. Thus, convolution layers are more suitable for image-related task, where they break the input image into features, whereas the FC layers work together to combine these features, such as word embedding in our case of textual data, into ∀-dimensional entities the network can recognize.

3.4.6. SGD momentum optimizer. In addition to the existing stochastic gradient descent (SGD) method, a momentum-based SGD technique may be utilized to improve the learning algorithm, which nearly always performs better and quicker than stochastic gradient descent. Momentum SGD is a method for speeding up convergence by accelerating gradient vectors in the proper directions. It is one of the most widely used optimization techniques and is used to train many cutting-edge models. SGD momentum makes use of exponentially weighted averages to cope with numerical sequences. When we use momentum, such as when pushing a ball down a hill, we are essentially employing momentum. The ball gains momentum as it goes downhill, accelerating faster and faster, as it does so. The loss function $\mathrm{Loss}(H)$ used by the SGD momentum is formulated as:

$$\mathrm{Loss}(H) = -\sum_{(u,i)}^{S} r_{B(u,i)} \log \grave{r}_{B(u,i)} + \left(1 - r_{B(u,i)}\right) \log\left(1 - \grave{r}_{B(u,i)}\right), \tag{4}$$

where S is the union of observed interactions, H is the model parameters $= \{L, B, I, h\}$, where L is the learning rate, B is the batch size, I is the number of iterations, and h is the number of latent dimensions. The STN-CDRS uses the word embedding matrix $\mathbf{W}$ to reduce model complexity by utilizing Eq. (4) as the loss function. The $\mathrm{Loss}(H)$ mentioned in Eq. (4) is optimized by the momentum-based stochastic gradient descent (M-SGD) algorithm. M-SGD is an upgraded variant of gradient descent that is used to better optimize the issue of the loss function being too large in the update and to expedite convergence.

3.4.7. Sigmoid activation function. The logistic function and the sigmoid activation function are two terms for the same operation. As the

STN-CDRS is converted into a regression learning approach, the logistic function is employed. The logistic function accepts any number as input and returns a value between [0, −1] as output. The greater the input, the closer the output of the logistic function to 1, and the lower the input, the closer the function output to 0. As the cross-domain recommendation problem is constructed as a linear regression problem, the sigmoid activation function for regression problem can be summarized as:

$$\text{logistic}(z) = \frac{1}{1 + e^{-(\alpha+\beta z)}}, \tag{5}$$

where $\alpha + \beta z$ is similar to the linear model $\gamma = \alpha z + \beta$. Usually, recommendation systems generally work in the range of (1–5); therefore, the logistic function can be updated for values in any rage, for example (1–5), using:

$$\text{logistic}(z) = \frac{T}{1 + e^{-(\alpha+\beta z)}}, \tag{6}$$

where T is set to the maximum value, which in our case is $T = 5$.

4. Experiments

The experiments are conducted using the ratings from the Amazon books and movies dataset [40]. In the research, we examined several baseline methodologies and analyzed the efficacy of rating and ranking prediction using a set of measures. A 5-fold cross-validation is used to conduct tests and assessments of the proposed model, which analyzes cross-domain recommendation performance using RMSE, MAE, precision, and recall metrics [41]. In our experiments, a group of state-of-the-art methods including Nearest Neighbor (KNN) [42], SVD++ [43], NCF [36], and DDTCDR [35] are evaluated for comparison, alongside our model, where:

1. **KNN** [42] is a supervised learning technique used for classification and regression. To classify a new data point, the algorithm saves all the data points and classifies the new point based on similarity index by putting it into the most similar saved category. It is also known as lazy learner algorithm because instead of learning immediately from training data, it learns at the time of classification.
2. **SVD++** [43] uses implicit feedback information for matrix factorization model, Although, SVD++, as discussed in earlier works, provides the best recommendations in traditional methods, itis very slow compared to other methods. The prediction $\grave{r}_{B(u,i)}$ with SVD++ is given by:

$$\grave{r}_{B(u,i)} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u}^{U} y_j \right), \tag{7}$$

where y_j refers to a new set of item factors that represent implicit ratings, implicit ratings occur when a user u rates an item j, independent of the rated value. The factors p_u and bias b_u, are assumed to be zero, if the user u is unknown, and the same is applicable to item i with b_i, q_i and y_i.

3. **NCF** [36] replaces matrix factorization using a deep architecture that learns any function from the data, allowing it to express and generalize matrix factorization. It also uses a MLP to learn the user–item relation function.
4. **DDTCDR** [35] efficiently transfers user preferences across domain pairs through a dual learning mechanism. It operates by embedding latent information from one domain into the other and iteratively loops through the transfer learning loop until both domain models settle.

During the evaluation, for each baseline method under comparison, all hyper-parameters settings during training and testing are kept fixed. Also, both book and movies domains are used as the target domain for evaluating the performance of STN-CDRS. Tables 1 and 2 show the statistics of the data for evaluation for KNN, SVD++, NCF, DDTCDR and STN-CDRS.

4.1. MSE and MAE convergence

Before comparing the proposed model to other models, the mean spared error (MSE) and mean absolute error (MAE) metrics are evaluated to assess the performance of the network during training. The MSE metric is computed as:

$$\mathrm{MSE} = \frac{\sum \left|\grave{r}_{B(u,i)} - r_{B(u,i)}\right|^2}{N}, \tag{8}$$

and MAE, a measure of errors between the predicted and actual observations [15], is expressed as:

$$\mathrm{MAE} = \frac{\sum \left|\grave{r}_{B(u,i)} - r_{B\,(u,i)}\right|}{N}, \tag{9}$$

where N represents the total number of expected outcome, $\grave{r}_{B(u,i)}$ is the predicted value for user u on item i and $r_{B(u,i)}$ gives the true rating. During training, MSE and MAE metrics are calculated at each epoch to validate the performance of the STN-CDRS, Figs. 5 and 6 show the convergence curve of the STN-CDRS in terms of MSE and MAE, respectively.

It is evident from Figs. 5 and 6 that the STN-CDRS network converges very fast, i.e., in less than 10 epochs the network achieves minima in terms of both MSE and MAE, for both movie and book domain. Compared to existing approaches, STN-CDRS takes less time to achieve lower MSE and MAE values.

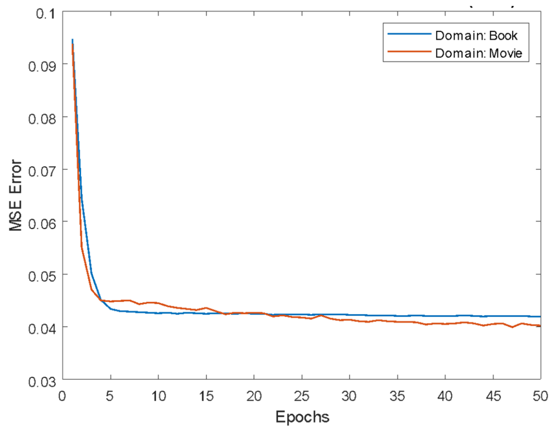

Fig. 5. MSE error of the STN-CDRS during epochs.

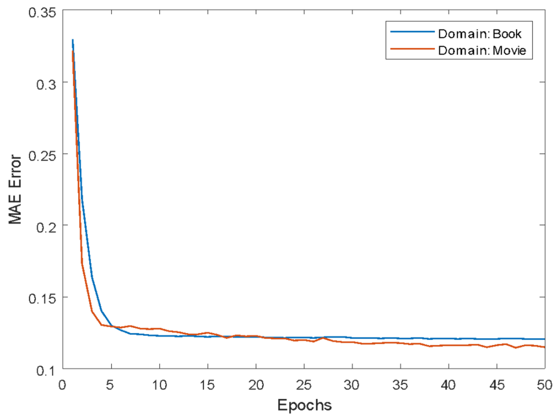

Fig. 6. MAE error of the STN-CDRS during epochs.

The training loss for the book and movie domains is compared in Tables 1 and 2, respectively, which also evaluates various methods in terms of RMSE, MAE, precision and recall.

Table 1. Comparison of existing methods with STN-CDRS for the target domainbook.

Domain: Book				
Algorithm	RMSE	MAE	Precision	Recall
STN-CDRS	0.1937	0.1247	0.8937	0.981
DDTCDR	0.2213	0.1708	0.8595	0.9594
NCF	0.2315	0.1887	0.8357	0.8924
SVD++	0.9101	0.7216	0.585	0.331
KNN	0.9761	0.7785	0.563	0.315

It is evident from Table 1 that, in terms of RMSE, the STN-CDRS outperformed all other methods: DDTCDR, NCF, SVD++ and KNN significantly. For the book domain, the RMSE values obtained using STN-CDRS, DDTCDR, NCF, SVD++ and KNN are 0.1937, 0.2213, 0.2315, 0.9101 and 0.9761, respectively. STN-CDRS outperformed the existing methods by 0.0276, 0.0378, 0.7164 and 0.7824 in absolute RMSE terms, respectively. Overall, STN-CDRS shows improvements of 12.47%, 16.33%, 78.72% and 80.16% over DDTCDR, NCF, SVD++ and KNN methods, respectively.

For the book domain, the MAE values obtained using STN-CDRS, DDTCDR, NCF, SVD++ and KNN are 0.1247, 0.1708, 0.1887, 0.7216 and 0.7785, respectively. STN-CDRS outperformed the existing methods: DDTCDR, NCF, SVD++ and KNN by 26.99%, 33.92%, 82.72% and 83.99%, respectively. Precision, which is the count of correct positive outcomes (recommended items liked by the user) divided by the classifier's predicted count of positive results (total recommended items), is calculated as:

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}, \tag{10}$$

whereas, recall is calculated by dividing the count of truly positive results (recommended items liked) by the total number of targeted samples (recommended liked items and also liked items that are not recommended):

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}. \tag{11}$$

A plot illustrating the distinctions between precision and recall is shown below in Fig. 7. STN-CDRS shows good improvement in precision compared to the baseline methods, with a 3.98 % improvement over DDTCDR, a 7.32%

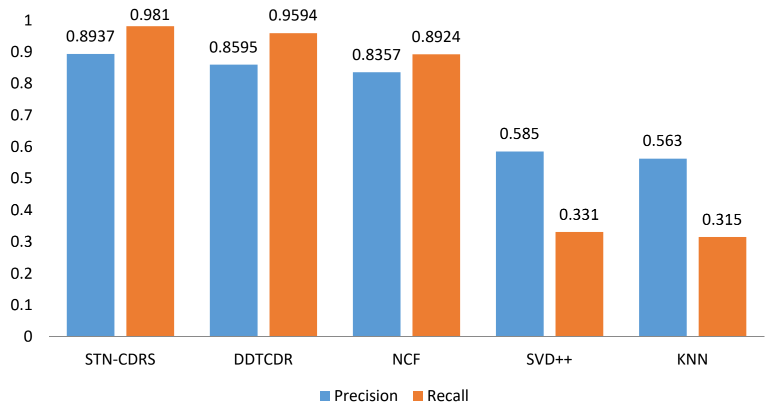

Fig. 7. Precision and recall value comparisons of STN-CDRS, DDTCDR, NCF, SVD++ and KNN.

improvement over CoNet, a 52.78% increase in precision compared to SVD++, and a 58.75% improvement over KNN in percentage terms. In addition, STN-CDRS shows improvement in recall compared to the baseline methods: a 2.26% better recall than DDTCDR, a 9.13% better recall than CoNet, a 196.39% better recall than SVD++, and a 211.44% better recall than KNN.

As STN-CDRS is capable of producing recommendation across domains, the book domain can be used as a source domain to recommend into the movie domain as the target. Table 2 shows the comparison of the proposed method with DDTCDR, NCF, SVD++ and KNN methods with STN-CDRS for target Domain Movie in terms of RMSE, MAE, precision and recall.

TABLE 2. Comparison of existing methods with STN-CDRS for the target movie domain.

Domain: Movie				
Algorithm	RMSE	MAE	Precision	Recall
STN-CDRS	0.1870	0.1193	0.9145	0.9718
DDTCDR	0.2213	0.1714	0.8925	0.9871
NCF	0.2276	0.1903	0.8644	0.9589
SVD++	0.9246	0.7340	0.5966	0.3334
KNN	0.9856	0.7800	0.5775	0.3241

It is clear from Fig. 8 that in terms of RMSE, STN-CDRS outperformed all other methods significantly: DDTCDR, NCF, SVD++ and KNN. The RMSE measures for the target domain movie obtained using the STN-CDRS, DDTCDR, NCF, SVD++ and KNN are 0.1870, 0.2213, 0.2276, 0.9246 and 0.9856, respectively.

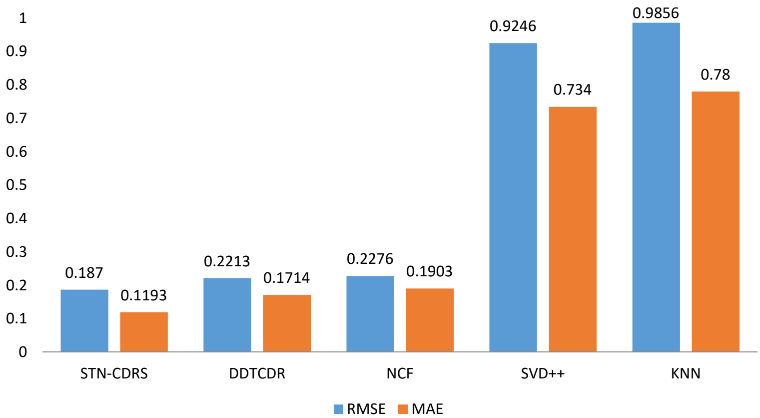

FIG. 8. RMSE and MAE values compared for the movie domain.

STN-CDRS outperformed the existing methods by 0.0343, 0.0406, 0.7376 and 0.7986 in absolute RMSE terms for the target domain movie. Also, in terms of RMSE, STN-CDRS shows improvements, which are 15.50% better than DDTCDR, 17.84% better than NCF, 79.78% better than SVD++, and 81.03% better than KNN. Likewise, for MAE, STN-CDRS shows improvements of 30.40% better than DDTCDR, 37.31% better than CoNet, 83.75% better than SVD++, and 84.71% better than KNN.

For precision, STN-CDRS outperformed the baseline methods by 0.022, 0.050, 0.317 and 0.337 in absolute terms, which is 2.46% better than DDTCDR, 5.80% better than NCF, 53.29% better than SVD++, and 58.35% better than KNN. In terms of recall, STN-CDRS shows improvements of 1.22% better than CoNet, 191.49% better than SVD++, 199.86% better than KNN. However, only for recall, the proposed method shows lower gain, which is -1.55% lower than DDTCDR.

4.2. Training loss vs. epochs

To further show the improvements made by the proposed algorithm, the training loss vs. epoch is plotted for both domains. It should be evident from the figures that STN-CDRS converges much faster. Figures 9 and 10 show the convergence curve of the proposed algorithm compared with the existing method of NCF and DDTCDR taken from [35].

The convergence curves in Figs. 9 and 10 illustrate the number of epochs required for each method to reach a stable minimum. It is evident from both figures that STN-CDRS only requires about 20 epochs to reach a stable minimum, whereas other algorithms took considerably longer and were still unable to identify a proper minimum. This efficiency of the STN-CDRS can be attributed to the utilization of the momentum-based SGD algorithm. Furthermore, it should be noted that the size of the dataset greatly impacts the performance of the

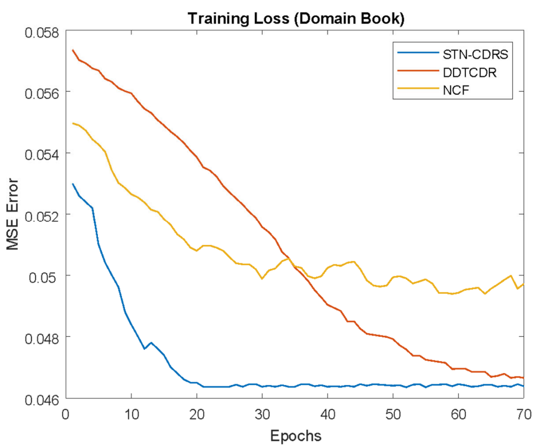

FIG. 9. Convergence curve of STN-CDRS for the domain book.

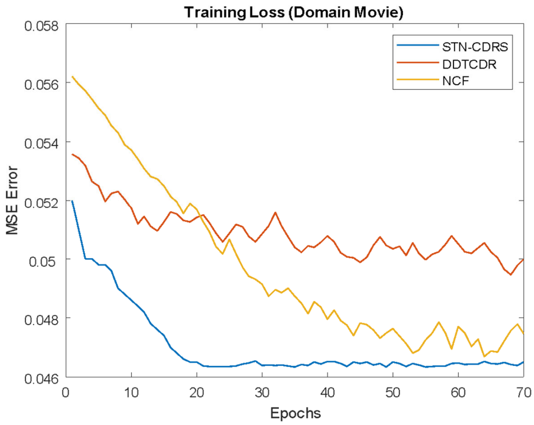

FIG. 10. Convergence curve of STN-CDRS for the domain movie.

network. If more data is available in one source domain and less data is available in the target domain, the performance will likely be impacted.

5. Conclusion and future scope

This work proposes Sentiment Transfer Network especially designed for providing recommendation in cross-domain scenarios (STN-CDRS). This model improves sparse recommendations utilizing user review data across domains from provided user reviews. This enrichment process uses the previously developed RNNCore method [37] for efficient learning from user reviews. This concept of review-aided cross-domain recommendation systems is a direct advancement over various existing recommendation systems inspiring the development of this model.

For this purpose, this work introduces STN-CDRS, a cross-domain recommendation algorithm. STN-CDRS uses sentiment information-based dataset enrichment and latent knowledge transfer from domain D_A to domain D_B. It utilizes cross-domain knowledge transfer and deep learning models to achieve its goals. STN-CDRS enriches the data from word embedding of user reviews instead of relying on explicit information present in the dataset regarding user–item interactions to capture complex user–item relationships. This enrichment helps in solving cold start problems and addresses data sparsity. More data available for learning results in better learning outcomes.

The results clearly demonstrate the supremacy of the STN-CDRS method, as it outperforms existing baseline techniques and improves recommendation accuracy across a variety of domains and experimental situations. STN-CDRS requires fewer epochs to reach a stable minimum compared to other algorithms. This property makes it scalable enough to be used in enterprise settings. It is observed that the size of the dataset highly impacts the performance of the network, if more data is available in one source domain and less data is available in the target domain, the performance will likely be impacted. How to effectively produce recommendations for smaller domains can be explored in the future. Also, it can be explored how the model can be directly adapted to support multi-domain recommendation scenarios.

References

1. S.S. Khanal, P.W.C. Prasad, A. Alsadoon, A. Maag, A systematic review: machine learning based recommendation systems for e-learning, *Education and Information Technologies*, **25**(4): 2635–2664, 2020, doi: 10.1007/s10639-019-10063-9.

2. T. Zang, Y. Zhu, H. Liu, R. Zhang, J. Yu, A survey on cross-domain recommendation: Taxonomies, methods, and future directions, *ACM Transactions on Information Systems*, **41**(2): 1–39, 2022, doi: 10.1145/3548455.

3. G. Guo, H. Qiu, Z. Tan, Y. Liu, J. Ma, X. Wang, Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems, *Knowledge-Based Systems*, **138**: 202–207, 2017, doi: 10.1016/j.knosys.2017.10.005.

4. B. Lika, K. Kolomvatsos, S. Hadjiefthymiades, Facing the cold start problem in recommender systems, *Expert Systems with Applications*, **41**(4, Part 2): 2065–2073, 2014, doi: 10.1016/j.eswa.2013.09.005.

5. I. Cantador, I. Fernández-Tobías, S. Berkovsky, P. Cremonesi, Cross-domain recommender systems, [in:] *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira [Eds.], pp. 919–959, Springer, Boston, MA, 2015, doi: 10.1007/978-1-4899-7637-6_27.

6. F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, G. Liu, Cross-domain recommendation: challenges, progress, and prospects, *arXiv*, 2021, arXiv: 2103.01696.

7. W. Pan, E. Xiang, N. Liu, Q. Yang, Transfer learning in collaborative filtering for sparsity reduction, [in:]*Proceedings of the AAAI Conference on Artificial Intelligence*, **24**(1): 230–235, 2010, doi: 10.1609/aaai.v24i1.7578.

8. X. Fang, Making recommendations using transfer learning, *Neural Computing and Applications*, **33**(15): 9663–9676, 2021, doi: 10.1007/s00521-021-05730-3.

9. X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, *Science China Technological Sciences*, **63**: 1872–1897, 2020, doi: 10.1007/s11431-020-1647-3.

10. T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, [in:] *Proceedings of the NIPS*, Red Hook, NY, USA, 5–8 December, pp. 3111–3119, 2013.

11. J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, [in:] *Proceedings of the NAACL*, Minneapolis, MN, USA, June 2–7, pp. 4171–4186, 2019.

12. T. Brown *et al.*, Language models are few-shot learners, *Advances in Neural Information Processing Systems*, **33**: 1877–1901, 2020.

13. A. Rietzler, S. Stabinger, P. Opitz, S. Engl, Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification, [in:] *Proceedings of the LREC*, Marseille, 11–16 May, pp. 4933–4941, 2020.

14. Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, *arXiv*, 2020, arXiv: 2003.00911.

15. Y. Li, J.J. Xu, P.P. Zhao, J.H. Fang, W. Chen, L. Zhao, ATLRec: An attentional adversarial transfer learning network for cross-domain recommendation, *Journal of Computer Science and Technology*, **35**(4): 794–808, 2020, doi: 10.1007/s11390-020-0314-8.

16. Z. Huang, H. Wang, E.P. Xing, D. Huang, Self-challenging improves cross-domain generalization, [in:] *European Conference on Computer Vision-ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, J.M. Frahm [Eds.], Lecture Notes in Computer Science, Vol. 12347, pp. 124–140, Springer, Cham, 2020, doi: 10.1007/978-3-030-58536-5_8.

17. H. Liu, L. Guo, P. Li, P. Zhao, X. Wu, Collaborative filtering with a deep adversarial and attention network for cross-domain recommendation, *Information Sciences*, **565**: 370–389, 2021.

18. H. Wang, D. Amagata, T. Makeawa, T. Hara, N. Hao, K. Yonekawa, M. Kurokawa, A DNN-based cross-domain recommender system for alleviating cold-start problem in e-commerce, *IEEE Open Journal of the Industrial Electronics Society*, **1**: 194–206, 2020, doi: 10.1109/OJIES.2020.3012627.

19. H.J. Xue, X.Y. Dai, J. Zhang, S. Huang, J. Chen, Deep matrix factorization models for recommender systems, [in:] *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pp. 3203–3209, 2017, doi: 10.24963/ijcai.2017/447.

20. J. Tang, K. Wang, Personalized top-n sequential recommendation via convolutional sequence embedding, [in:] *WSDM'18: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 565–573, 2018, doi: 10.1145/3159652.3159656.

21. S.T. Zhong, L. Huang, C.D. Wang, J.H. Lai, P.S. Yu, An autoencoder framework with attention mechanism for cross-domain recommendation, *IEEE Transactions on Cybernetics*, **52**(6): 5229–5241, 2022, doi: 10.1109/TCYB.2020.3029002.

22. T. Man, H. Shen, X. Jin, X. Cheng, Cross-domain recommendation: An embedding and mapping approach, [in:] *Proceeding of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, pp. 2464–2470, 2017, doi: 10.24963/ijcai.2017/343.

23. F. Zhu, Y. Wang, C. Chen, G. Liu, M. Orgun, J. Wu, A deep framework for cross-domain and cross-system recommendations, *arXiv*, 2020, arXiv: 2009.06215.

24. F. Yuan, L. Yao, B. Benatallah, DARec: Deep domain adaptation for cross-domain recommendation via transferring rating pattern, *arXiv*, 2019, arXiv: 1905.10760.

25. C. Zhao, C. Li, C. Fu, Cross-domain recommendation via preference propagation GraphNet, [in:] *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 2165–2168, 2019, doi: 10.1145/3357384.3358166.

26. C. Zhao, C. Li, R. Xiao, H. Deng, A. Sun, CATN: Cross-domain recommendation for cold-start users via aspect transfer network, [in:] *SIGIR'20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 229–238, 2020, doi: 10.1145/3397271.3401169.

27. C. Li, C. Quan, L. Peng, Y. Qi, Y. Deng, L. Wu, A capsule network for recommendation and explaining what you like and dislike, [in:] *SIGIR'19: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 275–284, 2019, doi: 10.1145/3331184.3331216.

28. Y. Xu, Z. Peng, Y. Hu, X. Hong, W. Fu, Cross-domain recommendation for mapping sentiment review pattern, [in:] *Knowledge Science, Engineering and Management Springer – KSEM 2018*, W. Liu, F. Giunchiglia, B. Yang [Eds.], Lecture Notes in Computer Science, Vol. 11061, pp. 324–336, Springer, Cham, 2018, doi: 10.1007/978-3-319-99365-2_29.

29. Y. Xu, Z. Peng, Y. Hu, X. Hong, SARFM: A sentiment-aware review feature mapping approach for cross-domain recommendation, [in:] *Web Information Systems Engineering – WISE 2018*, H. Hacid *et al.* [Eds.], Lecture Notes in Computer Science, Vol. 11234, pp. 3–18, Springer, Cham, 2018, doi: 10.1007/978-3-030-02925-8_1.

30. Y. Wang, H. Yu, G. Wang, Y. Xie, Cross-domain recommendation based on sentiment analysis and latent feature mapping, *Entropy*, **22**(4): 473, 2020, doi: 10.3390/e22040473.

31. W. Hong, N. Zheng, Z. Xiong, Z. Hu, A parallel deep neural network using reviews and item metadata for cross-domain recommendation, *IEEE Access*, **8**: 41774–41783, 2020, doi: 10.1109/ACCESS.2020.2977123.

32. Y. Cai, W. Ke, E. Cui, F. Yu, A deep recommendation model of cross-grained sentiments of user reviews and ratings, *Information Processing & Management*, **59**(2): 102842, 2022, doi: 10.1016/j.ipm.2021.102842.

33. N. Taneja, H.K. Thakur, Evaluating the scalability of matrix factorization and neighborhood-based recommender systems, *International Journal of Information Technology and Computer Science (IJITCS)*, **15**(1): 21–29, 2023, doi: 10.5815/ijitcs.2023.01.03.

34. G. Hu, Y. Zhang, Q. Yang, CoNet: Collaborative cross networks for cross-domain recommendation, [in:] *CIKM'18: Proceedings of the 27th ACM international Conference on Information and Knowledge Management*, pp. 667–676, 2018, doi: 10.1145/3269206.3271684.

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

36. X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.S. Chua, Neural collaborative filtering, [in:] *WWW'17: Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182, 2017, doi: 10.1145/3038912.3052569.

37. N. Taneja, H.K. Thakur, RNNCore: Lexicon aided recurrent neural network for sentiment analysis, *International Journal of Computing and Digital System*, **12**(1): 1561–1568, 2021, doi: 10.12785/ijcds/1201126.

38. M. Song, T. Chambers, Text mining with the Stanford CoreNLP, [in:]*Measuring Scholarly Impact*, Y. Ding, R. Rousseau, D. Wolfram [Eds.], pp. 215–234, Springer, Cham, 2014, doi: 10.1007/978-3-319-10377-8_10.

39. Z. Liu, L. Zheng, J. Zhang, J. Han, S.Y. Philip, JSCN: Joint spectral convolutional network for cross domain recommendation, [in:] *2019 IEEE International Conference on Big Data (Big Data)*, pp. 850–859, IEEE, 2019.

40. S. AlZu'bi, A. Alsmadiv, S. AlQatawneh, M. Al-Ayyoub, B. Hawashin, Y. Jararweh, A brief analysis of Amazon online reviews, [in:] *Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pp. 555–560, 2019, doi: 10.1109/SNAMS.2019.8931816.

41. Y. Liu, Y. Zhou, S. Wen, C. Tang, A strategy on selecting performance metrics for classifier evaluation, *International Journal of Mobile Computing and Multimedia Communications (IJMCMC)*, **6**(4): 20–35, 2014, doi: 10.4018/IJMCMC.2014100102.

42. A.K. Sahu, P. Dwivedi, Knowledge transfer by domain-independent user latent factor for cross-domain recommender systems, *Future Generation Computer Systems*, **108**: 320–333, 2020, doi: 10.1016/j.future.2020.02.024.

43. C. Xing, X. Yang, Cross-domain recommendation model based on SVD++ and tag, *Computer Engineering*, **44**(4): 225–230, 2018.

44. C. Li, M. Zhao, H. Zhang, C. Yu, L. Cheng, G. Shu, B. Kong, D. Niu, RecGURU: Adversarial learning of generalized user representations for cross-domain recommendation, [in:] *WSDM'22: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 571–581, 2022, doi: 10.1145/3488560.3498388.

45. Z. Li, D. Amagata, Y. Zhang, T. Maekawa, T. Hara, K. Yonekawa, M. Kurokawa, HML4Rec: Hierarchical meta-learning for cold-start recommendation in flash sale e-commerce, *Knowledge-Based Systems*, **255**: 109674, 2022, doi: 10.1016/j.knosys.2022.109674.

46. Y. Deng, Recommender systems based on graph embedding techniques: A review, *IEEE Access*, **10**: 51587–51633, 2022, doi: 10.1109/ACCESS.2022.3174197.

Received September 28, 2022; revised version February 14, 2023; accepted February 27, 2023; published online June 19, 2024.