Cross-domain sequential recommendation is the task of predicting the next item that the user is most likely to interact with based on past sequential behavior from multiple domains. One of the key challenges in cross-domain sequential recommendation is to grasp and transfer the flow of information from multiple domains so as to promote recommendations in all domains. Previous studies have investigated the flow of behavioral information by exploring the connection between items from different domains. The flow of knowledge (i.e., the connection between knowledge from different domains) has so far been neglected. In this article, we propose a mixed information flow network for cross-domain sequential recommendation to consider both the flow of behavioral information and the flow of knowledge by incorporating a behavior transfer unit and a knowledge transfer unit. The proposed mixed information flow network is able to decide when cross-domain information should be used and, if so, which cross-domain information should be used to enrich the sequence representation according to users’ current preferences. Extensive experiments conducted on four e-commerce datasets demonstrate that the proposed mixed information flow network is able to improve recommendation performance in different domains by modeling mixed information flow. In this article, we focus on the application of mixed information flow networks to a scenario with two domains, but the method can easily be extended to multiple domains.

CCS Concepts: • Information systems → Recommender systems;

Additional Key Words and Phrases: Cross-domain recommendation, sequential recommendation, knowledge base, graph transfer

This research was partially supported by the Key Research and Development Program of China under Grant (2018YFC0831000), the Natural Science Foundation of China (62072279, 61972234, 61902219, 62102234), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), the Natural Science Foundation of Shandong Province (ZR2021QZ230192), the Tencent WeChat Rhino-Bird Focused Research Program (JR-WXG-20214111), the Fundamental Research Funds of Shandong University, the National Key R&D Program of China (2020YFB1406700), the Meituan corporation, and the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organization for Scientific Research, https://hybrid-intelligence-centre.nl. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

Authors’ addresses: M. Ma, P. Ren (corresponding author), Z. Chen (corresponding author), Z. Ren, L. Zhao, and J. Ma, Shandong University, Qingdao 266237, China; emails: muyang0331@gmail.com, jay.ren@outlook.com, {chenzhumin, zhaochun.ren}@sdu.edu.cn, magician233@outlook.com, majun@sdu.edu.cn; P. Liu, Shandong Normal University, Jinan 250014, China; email: liup@sdnu.edu.cn; P. Liu, Shandong Normal University, Jinan 250014, China; email: liup@sdnu.edu.cn; P. Liu, Shandong Normal University, Jinan 250014, China; email: liup@sdnu.edu.cn; M. de Rijke, University of Amsterdam, Amsterdam 1012 WX, The Netherlands; email: m.derijke@uva.nl.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.

ACM Transactions on Knowledge Discovery from Data, Vol. 16, No. 4, Article 64. Publication date: January 2022.
1 INTRODUCTION

Sequential recommendation (SR) aims at predicting the next item that the user is most likely to interact with based on her/his past sequential behavior (e.g., clicks on items) [57]. Recently, cross-domain sequential recommendation (CDSR) has emerged as a way to promote recommendation performance by leveraging and combining information from different domains [47]. Users usually have related preferences in different domains, such as finding a movie with a certain style or looking for a book written by a well-known author, as illustrated in Figure 1. One of the key challenges in CDSR is to capture and transfer useful information about related preferences across different domains.

Zhuang et al. [92] and Ma et al. [47] have shown that behavioral information across domains is helpful for improving recommendation performance. However, behavioral information by itself can only support the use of cross-domain connections in a limited manner. Behavioral information may be insufficient for a model to capture fine-grained connections between item attributes or features. As an example, consider Figure 1 and assume that there is a user who has read Harry Potter (the book) and watched Captain America (the movie). If there is no external knowledge to indicate that both items belong to the category of "fantasy," it is difficult for the model to capture this connection based solely on the user’s behavior from both domains. When supported by external knowledge, the model can capture this characteristic from the knowledge information flow, and this information can be used to model user preferences across different domains. We hypothesize that enabling a flow of knowledge across different domains is able to alleviate this issue. As a result, for a user who has read the book The Great Gatsby, we then can recommend her/him a movie having the same name or movies featuring the same category of "tragic love," such as Atonement, Waterloo Bridge, and so on, when she/he logs on to the movie recommendation system.

There is a growing body of work aimed at improving recommendation performance by using knowledge [43, 59]. Of particular relevance to us is work that has proposed to incorporate knowledge and combine it with behavioral information for SR [see e.g., 34, 35, 82]. However, this work targets a single domain recommendation scenario. The situation is dramatically different in cross-domain scenarios where it is necessary to distinguish information from different domains and effectively link them. We need to select behavioral and knowledge related to users’ current preference, and determine when and what to use in order to learn a better sequence representation.

To address the issue of using behavioral information and knowledge across domains, we propose a mixed information flow network (MIFN) to consider mixed information flow across domains, i.e., the flow of behavioral information as well as the flow of knowledge. The former is based on user’s behavior, which captures the temporal connection between the items they have interacted with, while the latter takes cross-domain knowledge as a bridge to connect different domains to obtain better cross-domain sequence representations. First, we employ a behavior transfer unit (BTU) to grasp useful information from the flow of behavioral information, which can extract information related to the user’s preference and then transfer it to another domain at the level of user behavior level. Then, we propose a knowledge transfer unit (KTU) that is guided by the user’s preference to model the connection between items from different domains; we introduce a cross-domain graph convolutional mechanism to distinguish items in the
knowledge graph (KG) and grasp useful information for fusion at the knowledge level. Finally, we generate recommendations based on the fusion of the two types of information. During learning, mixed information flow network (MIFN) is jointly trained on multiple domains in an end-to-end back-propagation training paradigm. Experiments on the Amazon datasets show that MIFN outperforms state-of-the-art methods in terms of MRR and Recall.

To sum up, the contributions of this work are as follows:

— We propose a mixed information flow framework, MIFN, for CDSR, which consists of a behavior transfer unit and a knowledge transfer unit to simultaneously model the flow of behavioral information and of knowledge across domains.

— We devise a cross-domain graph convolutional mechanism to disseminate item information in the KG, which leads to the better up-to-date item representation.

— We conduct experiments to demonstrate that MIFN is able to improve recommendation performance in different domains by modeling mixed information flow.

2 RELATED WORK

In this section, we briefly introduce related work from the following categories: (1) SR, (2) cross-domain recommendation, and (3) knowledge-aware recommendation.

2.1 Sequential Recommendations

Early work on recommender systems (RSs) typically uses collaborative filtering (CF) to generate recommendations [32] according to users’ preferences reflected in similar items such as K-Nearest neighbors (KNN) or matrix factorization (MF) algorithms. Such methods do not consider sequential aspects. More recently, however, SR and next-basket recommendation have witnessed rapid developments.

Before the widespread application of deep learning, Markov chains (MCs) [10, 24, 62, 93] and Markov decision processes (MDPs) [65, 83] were used to predict users’ next actions given information about their past behavior [77, 86]. All these methods take into account sequential characteristics. However, there are considerable challenges with the size of the state space when considering arbitrary sequences [56].

Recurrent neural networks (RNNs) have been introduced to SR to handle variable-length sequential data. Hidasi et al. [27] are the first to leverage recurrent neural networks (RNNs) for SR. They utilize session-parallel mini-batch training and employ ranking-based loss functions to...
train the model. Then, Tan et al. [69] propose two techniques to improve the performance, i.e., data augmentation and a method to account for shifts in the input data distribution. Li et al. [40] incorporate an attention mechanism into the encoder to capture the users’ main preference in the current sequence. Ren et al. [58] point out the repeat consumption in SR, where the same item is re-consumed repeatedly over time. Quadrana et al. [57] propose a hierarchical RNN model that can relay and evolve latent hidden states of the RNNs across user sequences. Donkers et al. [16] explicitly model user information in a gated architecture with extra input layers for gated recurrent unit (GRU). Chen et al. [12] argue that most work considers user preferences to be static and relies on post-processing the list of recommendations to promote diversity. They propose an intent-aware diversified network to improve accuracy and diversity by introducing self-attention mechanism based on GRU networks. Memory enhanced RNN has been well studied for SR recently. Chen et al. [13] introduce a memory mechanism to SR and design a memory-augmented neural network integrated with the insights of CF. Wang et al. [75] propose two parallel memory modules: one to model a user’s own information in the current sequence and the other to exploit collaborative information in neighborhood sequences. Wu et al. [82] argue that prior work on conventional sequential methods neglects complex transitions between items. They model the sequence as graph-structured data and then represent it as the composition of global preference and the current preference of that sequence using an attention network. Zhang et al. [88] propose a feature-level deeper self-attention network to capture transition patterns between features of items by integrating various heterogeneous features. Sun et al. [68] argue that previous work often assumes a rigidly ordered sequence, which is not always practical. They employ deep bidirectional self-attention to model a user’s behavioral sequences.

In addition to sequential information, auxiliary information is also vital for SR. Hidasi et al. [28] investigate how to add item property information such as text and images to an RNNs framework and introduce a number of parallel RNN (p-RNN) architectures. Liu et al. [44] incorporate contextual information into SR and propose a context-aware RNN model to capture external situations and lengths of time intervals. Bogina and Kuflik [9] explore a user’s dwell time based on an existing RNN-based framework by boosting items above a predefined dwell time threshold. Ma et al. [48] propose a cross-attention memory network for multi-modal tweets via both textual and visual information. Li et al. [39] study how to enlist the semantic signals covered by user reviews for the task of CF. They propose a neural review-driven model by considering users’ intrinsic preference and sequential patterns. To investigate the influence of temporal sentiments on user preference, Zheng et al. [91] propose to generate preferences by guiding user behavior through sequential sentiments. They design a dual-channel fusion mechanism to match and guide sequential user behavior, and to assist in preference generation. Ren et al. [60] model the effect of context information on SR and train the model in an adversarial manner by proposing multiple context-specific discriminators to evaluate the generated sub-sequence from the perspectives of different contexts.

Although these studies have made great progress, none of them has considered how to combine knowledge information under cross-domain situations.

### 2.2 Cross-Domain Recommendations

Cross-domain recommendation has emerged as a potential solution to the cold-start and data-sparse problem [6, 55] in RS. It aims at mitigating the lack of data by exploiting user preference and item attributes in domains distinct but related to the target domain [20].

Traditional methods for cross-domain recommendation can be grouped into two main categories [18]. One category of methods aggregates information across different domains. According to different aggregating strategies, such methods can be further divided into three groups [20]. The
first group is merging user preference (e.g., ratings, transaction behavior, and browsing logs) from different domains to obtain better a preference representation so as to improve the recommendation performance in the target domain. The merge operation is performed by merging a multi-domain rating matrix [5, 63], leveraging users’ social influence [1, 19], linking users’ preference by a multi-domain graph [15, 70] or user behavioral information features [46, 50]. The second group is mediating user modeling data in the source domain to explore the connection between users or items so as to make recommendations in the target domain especially for cold start users. For example, Tirosi and Kuflik [71] and Shapira et al. [66] propose to find similar neighbors and transfer user–user similarity to the target domain. The third group is combining single-domain recommendations (e.g., rating matrices, probability distributions), in which recommendations are generated independently for each domain and later aggregated for the final recommendation. In contrast to the second group, this type of aggregation strategy aims at modeling the weights assigned to recommendations coming from different domains. For example, Givon and Lavrenko [22] focus on book recommendations accomplished by a CF method and model-based recommendations, relying on the similarity of a book and the user’s model, as well as the book content and tags. And the final recommendations are combined in a weighted manner. The other category of cross-domain recommendation aims to transfer information from the source domain to the target domain by means of shared latent features or rating patterns. Hu et al. [33] propose tensor-based factorization to share latent features between different domains by using the same parameters in both factorization models. Li et al. [38] propose a code-book-transfer by co-clustering the source domain rating matrix and exploit it in the target domain to transfer rating patterns across different domains. Similarly, Mirbakhsh and Ling [53] focus on extending clustering-based MF in a single domain into multiple domains through overlapping users. In order to model more complex connections across different domains, a variety of deep learning methods have been proposed for cross-domain recommendation [23, 79, 80]. Elkakhky et al. [17] propose a multi-view deep learning recommendation system by using rich auxiliary features to represent users from different domains. Then, Lian et al. [42] propose a multi-view neural framework of a dual network for user and item, each network models CF information (user and item embeddings) and content information (user preference for item features), which ties CF and content-based filtering together. Hu et al. [31] propose a model using a cross-stitch network [54] to learn complex user behavioral information based on neural CF [26]. Wang et al. [79] propose to combine user behavioral information in information domains and user–user connection in social domains to do recommendation. Wang et al. [80] embed item-level information and cluster-level correlative information from different domains into a unified framework. Gao et al. [21] transfer item embeddings across domains without sharing user-relevant data. Li and Tuzhilin [41] develop a latent orthogonal mapping method to extract user preference over multiple domains while preserving connection between users across different latent spaces based on the mechanism of dual learning. Krishnan et al. [37] propose to guide neural CF with domain-invariant components shared across the dense and sparse domains, improving user and item representations learned in the sparse domains. They leverage contextual invariances across domains to develop these shared modules. Zhao et al. [89] propose to model user preference transfer at the aspect-level derived from reviews, which does not require overlapping users or items in all domains. Bi et al. [7] utilize cross-domain mechanism to promote recommendations for cold start users in insurance domain. They design a meta-path based method over complex insurance products to learn better item representations and learn the mapping function between domains through the overlapping users. Bi et al. [8] are the first to combine a cross-domain mechanism and a heterogeneous information network (including userid, agents, insurance products, and insurance product properties based on data from the insurance application PingAn Jinguanjia) to give personalized recommendations for cold start users in the
insurance domain. Despite the fact that the methods listed above have been proven to be effective, they cannot be directly applied to SRs.

Recently, cross-domain recommendation has been introduced to SRs as well. Zhuang et al. [92] propose a novelty seeking model based on sequences in multi-domains to model an individual’s propensity by transferring novelty seeking traits learned from a source domain for improving the accuracy of recommendations in the target domain. Yang et al. [85] propose a long short-term memory (LSTM) with sequence completion to model the temporal information in a cross-domain scenario. They propose two methods for sequence completion. One uses measures the similarity between item features and temporal features, the other uses an improved LSTM network. Ma et al. [47] study CDSR in a shared-account scenario. They propose a novel gating mechanism to extract and share user-specific information between domains.

Although some studies have begun to explore CDSR, they only focus on user behavioral information to conduct information transfer, and neglect exploring extra knowledge to promote sequence representation across domains.

### 2.3 Knowledge-Aware Recommendations

Side-information, e.g., user reviews and product descriptions, is usually provided together with user–item ratings [3]. Alexandridis et al. [4] propose to extend CF–MF algorithms by fusing the side information into the user/item representations. They propose to use the paragraph vector model to encode user reviews of variable length into feature representations of fixed length. Then, they fuse the resulting feature vectors with the rating scores in a hybrid probabilistic MF algorithm based on maximum a-posteriori estimation. Hu [29] argues that traditional CF approaches only exploit user–item relations (e.g., clicks, likes, and views), which suffer from the data sparsity issue. They design a personalized neural embedding model to fuse side information for effective estimation of user preferences. Furthermore, Hu and Dai [30] integrate item reviews into MF-based Bayesian personalized ranking, which can generate a ranked list of items for individual users.

Considerable efforts have also been made to utilize some structural side-information [74, 81], especially knowledge graphs, to enhance the performance of recommendations. Zhao et al. [90] propose a graph-based method to iteratively update user and item distributions in a heterogeneous user–item graph and incorporate them as features into the MF for item recommendations. Zhang et al. [87] combine CF with structural knowledge, textual knowledge, and visual knowledge in a unified framework. Ai et al. [2] apply TransE on the graph including users, items, and their connections, which casts the recommendation task as a plausibility prediction task. As graph convolutional networks (GCNs) have been shown to be effective on many tasks [36], there have been a number of publications that propose variants of graph convolutional networks (GCNs) for recommendation by considering different types of information. Wang et al. [73] simulate users’ hierarchical preferences over knowledge entities by extending users’ potential preferences among links in a KG. Wang et al. [74] consider the connections among items based on higher-order entity features in KGs. Wang et al. [78] explicitly model the high-order connections in KGs by employing an attention mechanism to discriminate the importance of the neighbors. Ma et al. [49] propose a joint framework to integrate the induction of explainable rules from KGs with the construction of a rule-guided recommendation model. Xian et al. [84] perform explicit reasoning with knowledge so that the recommendations are supported by an interpretable inference procedure via a policy-guided reinforcement learning approach.

Not surprisingly, KGs have also been considered in SRs. Huang et al. [35] are the first to integrate KGs into SR; they utilize RNNs to capture user sequential preference and knowledge memory networks to capture attribute-level user preference. Song et al. [67] model users’ social influence with a graph-attention neural network, which dynamically infers the influencers based on the
users’ current preference. Huang et al. [34] introduce a taxonomy-aware memory-based multi-hop reasoning architecture by incorporating taxonomy data as structural knowledge to enhance the reasoning capacity. Wang et al. [76] formalize the SR problem as a Markov decision process (MDP) and propose a knowledge guided reinforcement learning (RL) model named KERL for fusing KG information into an RL framework, resulting in three main technical contributions concerning state representation, reward function, and learning algorithm. Chen and Wong [11] propose a heterogeneous graph neural network to learn user and item representations by integrating knowledge from user social networks. Meng et al. [52] incorporate the user micro-behavior (such as adding to cart or reading reviews) and item knowledge (item attributes) into a multi-task learning framework. Wang et al. [72] argue that the temporal evolution of the effects caused by different item relations are usually neglected. To model the dynamic meaning of an item in different sequences, they propose a method named Chorus to take both item relations and the corresponding temporal dynamics into consideration.

However, no previous work has considered KGs for SR in a cross-domain scenario, which brings new challenges, e.g., how to find useful and accurate cross-domain knowledge to improve information transfer across domains to promote the performance in both domains.

3 METHOD

In this section, we first give a formulation of the CDSR task. Then, we give an overview of our model MIFN. Finally, we describe each component of MIFN in detail. Table 1 summarizes the main symbols and notation used in this article.

3.1 Task Formulation

Cross-domain sequential recommendation (CDSR) aims at predicting the next item the user is mostly likely to interact with in multiple domains simultaneously, by mining users’ previous sequential behavior over a period of time. In this work, we take two domains (i.e., domain A and B) as an example, e.g., watching movies, reading books. Let $A = \{A_1, A_2, A_3, \ldots, A_n\}$ denote the item set for domain A, which consists of $n$ unique items. Similarly, let $B = \{B_1, B_2, B_3, \ldots, B_m\}$ denote the item set for domain B, which consist of $m$ unique items. A hybrid interaction sequence from the two domains A and B has the form $S = [A_1, B_1, B_2, \ldots, A_i, B_j, \ldots]$, where $A_i \in A \ (1 \leq i \leq n)$ and $B_j \in B \ (1 \leq j \leq m)$ are the indices of consumed items in domain A and B, respectively.

We also associate each $S$ with a knowledge graph (KG), which is defined over an entity set $E$ and a relation set $R$, containing a set of KG triples. A triple $(e_1, r, e_2)$ represents a relation $r \in R$ between two entities $e_1$ and $e_2$ from $E$. In the cross-domain scenario, entities in the KG come from different domains, hence we represent them as $E_A$ and $E_B$. For example, the triple $(e_{A_1}, Is\_the\_same\_category, e_{B_j})$ means that entity $e_{A_1}$ from domain A has the same category as entity $e_{B_j}$ from domain B. Since we aim at linking recommended items to KG entities, an item set can be considered as a subset of KG entity set, i.e., $A \subseteq E_A$ and $B \subseteq E_B$. When extracting KG information for each sequence $S$, we also refer to the “items” in $S$ as “item entities”. We will explain the details of the KG construction method in Section 3.2.

Based on these preliminaries, we are ready to define the CDSR task. Formally, given $S$ and $(E, R)$, we formulate CDSR as a task of evaluating the recommendation probabilities for all candidates in both domains, respectively, as shown in the following equation:

$$P(A_{i+1}|S, (E, R)) \sim f_A(S, (E, R))$$
$$P(B_{j+1}|S, (E, R)) \sim f_B(S, (E, R)),$$

(1)

where $P(A_{i+1}|S, (E, R))$ denotes the probability of recommending the next item $A_{i+1}$ in domain A given the hybrid interaction sequence $S$ and KG $(E, R)$. $f_A(S, (E, R))$ is the model or function used...
Table 1. Summary of the Main Symbols and Notation Used in the Article

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Gloss</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{A} )</td>
<td>Item set for domain ( A ).</td>
</tr>
<tr>
<td>( \mathcal{B} )</td>
<td>Item set for domain ( B ).</td>
</tr>
<tr>
<td>( n )</td>
<td>The number of item set for domain ( A ), i.e., ( n =</td>
</tr>
<tr>
<td>( m )</td>
<td>The number of item set for domain ( B ), i.e., ( m =</td>
</tr>
<tr>
<td>( A_i )</td>
<td>The interacted item at time step ( i ) from domain ( A ).</td>
</tr>
<tr>
<td>( B_j )</td>
<td>The interacted item at time step ( j ) from domain ( B ).</td>
</tr>
<tr>
<td>( S )</td>
<td>Hybrid interaction sequence, i.e., ( S = [A_1, B_1, B_2, \ldots, A_i, \ldots, B_j, \ldots] ).</td>
</tr>
<tr>
<td>( S_A )</td>
<td>A sub-sequence of ( S ) which only contains items from domain ( A ).</td>
</tr>
<tr>
<td>( S_B )</td>
<td>A sub-sequence of ( S ) which only contains items from domain ( B ).</td>
</tr>
<tr>
<td>( e_k \in E )</td>
<td>( e_k ) represents any entity in the KG; ( E ) is the set of all entities.</td>
</tr>
<tr>
<td>( h_{A_i} \in H_A )</td>
<td>( h_{A_i} ) represents the item representation of item ( A_i ); ( H_A ) is the set of all item representations.</td>
</tr>
<tr>
<td>( h_{B_j} \in H_B )</td>
<td>( h_{B_j} ) represents the item representation of item ( B_j ); ( H_B ) is the set of all item representations.</td>
</tr>
<tr>
<td>( h_{(A \rightarrow B)} )</td>
<td>Transferred behavioral information flow from domain ( A ) to domain ( B ) at time step ( i ).</td>
</tr>
<tr>
<td>( N_i(k) )</td>
<td>Neighbor entity set of entity ( e_k ) from the same domain as ( e_k ).</td>
</tr>
<tr>
<td>( N_c(k) )</td>
<td>Neighbor entity set of entity ( e_k ) from the complementary domain.</td>
</tr>
</tbody>
</table>

to estimate \( P(A_{i+1}|S, (E, R)) \). Similar definitions apply to \( P(B_{j+1}|S, (E, R)) \) and \( f_B(S, (E, R)) \). It is important to note that when recommending items (such as \( B_x \)) in the \( B \) domain, the model can only see items before the interaction with \( B_x \) from both domains. As to modeling hybrid sequences, we follow the definition of previous work on CDSR [47].

3.2 Knowledge Graph Construction

Compared with other auxiliary information, KG information has the following advantages. First, it introduces more semantic relationships between items, which can help to discover user preferences at a deeper level. Second, the KG can connect the user’s historical records and recommendation results, thereby improving the user’s satisfaction and acceptance of the recommendation results, and enhancing the user’s trust in the recommender systems (RSs).

In this work, we extract data (entities and relations) from the Amazon product collection\(^1\) as the complete KG, which is collected from massive user logs. Besides, we also crawl some relations between the entities from Wikipedia.\(^2\) The entities include “movies,” “books,” “kitchenware,” and “food,” each of which corresponds to one domain. The relations include “Also_buy,” “Also_view,” “Buy_together,” “Buy_after_viewing,” “Adapted_from,” and “Is_the_same_category.” For

\(^1\)https://jmcauley.ucsd.edu/data/amazon.
\(^2\)https://en.wikipedia.org/.
example, (steak (the food), Buy_together, saucepan (the kitchenware)) means that the user also buys the saucepan while buying the steak. However, the complete KG contains a large number of related entities, and complicated relations among these entities, which will raise memory and computational efficiency issues. Therefore, we propose to extract a KG from the complete KG for each hybrid interaction sequence S. We require that, given any pair of items from both domains, respectively, we can find at least one path in the KG that connects them.

**ALGORITHM 1:** KG Construction Algorithm for Each Hybrid Interaction Sequence.

**Input:**
- Hybrid interaction sequence, S;
- Complete KG triples, (E, R, E);
- Maximum hop count, H;
- Number of entities in the KG, N;

**Output:**
- Multiple relational adjacency matrix, A;
- Complete KG triples, (E, R, E);
- Number of entities in the KG, N;
- Maximum hop count, H;

1. Divide the hybrid interaction sequence S into S_A and S_B;
2. set $\theta^0 = S_A$, $\theta^0 = S_B$, $T^0 = \emptyset$, $T^0 = \emptyset$, connected = false;
3. for $k \in \text{range}(H)$ do
   4. Extract related entities $\theta^k = \{t|(h, r, t) \in (E, R, E) \land h \in \theta^{k-1}\}$ and triples $T^k = \{(h, r, t)|(h, r, t) \in (E, R, E) \land h \in \theta^k\}$;
   5. Extract related entities $\theta^0 = \{t|(h, r, t) \in (E, R, E) \land h \in \theta^{k-1}\}$ and triples $T^0 = \{(h, r, t)|(h, r, t) \in (E, R, E) \land h \in \theta^k\}$;
   6. connected = Is_Connect($\theta^k$, $\theta^k$)
   7. if connected = true or $k = H-1$ then
   8. $\theta = \text{Select_triples}(T^k, T^0, S_A, S_B)$;
   9. $A = \text{Construct_Adjacency_matrix}(\theta)$;
10. break;
11. end if
12. end for
13. return adjacency matrix A;

The KG construction algorithm is shown in Algorithm 1. H represents the maximum hop count and N represents the number of entities in the final KG. We extract all triples and entities that are related to items involved in the hybrid interaction sequence S within H hops. We stop extracting more hops when there is a path for any given pair of items from both domains. Finally, we construct the relational adjacency matrix for the extracted KG. This process can be run in parallel. For about 50,000 sequences, it takes around 3 hours using 10 CPU cores, and each sequence takes 0.2 seconds on average. In this work, we focus on modeling the behavioral and knowledge flow of information across domains. The focus is on how to model this flow rather than how to build the knowledge itself, although a better KG construction algorithm is expected to improve the results a lot (as shown in Section 5.3).

Specifically, for the input S, we first divide it into two sub-sequences S_A and S_B according to the domain to which the items belongs (see line 1). We initialize the related entities $\theta^0_A$ or $\theta^0_B$ with the item entities, and initialize the related triples $T^0_A$ and $T^0_B$ with the empty set (see line 2). At each hop, we extract all related entities in the current KG (i.e., $\theta^k_A$ and $\theta^k_B$), where $\theta^k_A = \{t|(h, r, t) \in (E, R, E) \land h \in \theta^{k-1}_A\}$, and $\theta^k_B = \{t|(h, r, t) \in (E, R, E) \land h \in \theta^{k-1}_B\}$, $k = 1, 2, \ldots, H$, where $\theta^0_A = S_A$ and $\theta^0_B = S_B$. At the same time, we record the related triples as $T^k_A = \{(h, r, t)|(h, r, t) \in (E, R, E)$ and

ACM Transactions on Knowledge Discovery from Data, Vol. 16, No. 4, Article 64. Publication date: January 2022.
Fig. 2. An example of path finding during KG construction.

$h \in \mathcal{B}_1^{k-1}$ and $T^k_2 = \{(h, r, t) | (h, r, t) \in \langle E, R, E \rangle$ and $h \in \mathcal{B}_1^{k-1}$} (see line 4 and line 5). If there is a path for any given pair of items from both domains (e.g., $A_3 \rightarrow e_8 \rightarrow e_{26} \rightarrow e_9 \rightarrow B_6$) or the number of hops reaches the maximum number of hops $H$, we stop extracting other triples from the complete KG and construct the relational adjacency matrix (see line 7 to line 12). Otherwise, we continue to extract other related triples. When constructing the adjacency matrix, we limit the number of entities in the KG to $N$. Therefore, we need to select some triples if the number of entities in all related triples is larger than $N$ (see line 8). To do so, we first gather all the entities that connect a pair of item entities from two domains, e.g., $e_8, e_{26}$ in the path $A_3 \rightarrow e_8 \rightarrow e_{26} \rightarrow B_6$. As shown in Figure 2, $A_3$ is from domain $A$ while $B_6$ is from domain $B$. During the KG construction phase, we aim at finding paths that can connect $A_3$ and $B_6$, i.e., the path marked in orange ($A_3 \rightarrow e_8 \rightarrow e_{26} \rightarrow B_6$).

Then, we select the entities according to their smallest distance w.r.t. any item from the two domains until the number of all entities meets $N$. After that, we construct the relational adjacency matrix $A$ based on the selected entities and their relations (see line 10).

3.3 MIFN

In the following subsections, we will demonstrate the details of MIFN. Generally, MIFN models $P(A_{i+1}|S, \langle E, R \rangle)$ and $P(B_{j+1}|S, \langle E, R \rangle)$ (see Equation (1)) by taking two recommendation modes into consideration, as shown in the following equation:

$$
P(A_{i+1}|S, \langle E, R \rangle) = P(M_{SA}|S, \langle E, R \rangle)P(A_{i+1}|M_{SA}, S, \langle E, R \rangle) + P(M_{GA}|S, \langle E, R \rangle)P(A_{i+1}|M_{GA}, S, \langle E, R \rangle)
$$

$$
P(B_{j+1}|S, \langle E, R \rangle) = P(M_{SB}|S, \langle E, R \rangle)P(B_{j+1}|M_{SB}, S, \langle E, R \rangle) + P(M_{GB}|S, \langle E, R \rangle)P(B_{j+1}|M_{GB}, S, \langle E, R \rangle).
$$

Here, $M_{S}$ and $M_{G}$ denote sequence mode and graph mode, which make recommendations at the user behavior level and the knowledge level, respectively. $P(M_{SA}|S, \langle E, R \rangle)$ and $P(M_{GA}|S, \langle E, R \rangle)$ represent the probabilities under the sequence mode and the graph mode in domain $A$, respectively, $P(A_{i+1}|M_{SA}, S, \langle E, R \rangle)$ and $P(A_{i+1}|M_{GA}, S, \langle E, R \rangle)$ refer to the probabilities of recommending the next item $A_{i+1}$ under the sequence mode and graph mode given a hybrid interaction sequence $S$ and the KG triples $\langle E, R \rangle$. The same definitions apply to domain $B$.

As shown on the left side of Figure 3, MIFN consists of four main components: a sequence encoder, a behavior transfer unit (BTU), a knowledge transfer unit (KTU), and a mixed recommendation decoder. The sequence encoder encodes the interacted item sequence into a sequence of item representations. The BTU takes the representations from the source domain as input, extracts behavioral information flow, and transfers it to the target domain. The KTU aims at grasping useful knowledge from the KG and propagates it to both domains. The mixed recommendation
The BTU extracts behavioral information flow from source domain, and transfers it to the target domain. The KTU aims at grasping useful knowledge from the KG and propagates it to both domains. Section 3.3 contains a walkthrough of the model.

decoder contains two decoders w.r.t. graph mode and sequence mode, respectively. The graph recommendation decoder evaluates the probability for all candidate items from the KG, corresponding to Equation (10). The sequence recommendation decoder evaluates the probability of clicking items, corresponding to Equation (11).

3.4 Sequence Encoder
As with existing studies [27, 28], we use an RNN to encode the sub-sequences $S_A$ and $S_B$. Here, we employ a GRU as the recurrent unit. The initial state of the GRUs is set to zero vectors, i.e., $h_0 = 0$. After that, we can obtain $H_A = \{h_{A1}, h_{A2}, ..., h_{Ai}, ..., h_{An}\}$ for domain $A$, and $H_B = \{h_{B1}, h_{B2}, ..., h_{Bj}, ..., h_{Bm}\}$ for domain $B$. Each $h_{Ai}$ or $h_{Bj}$ is the item representation of an item $A_i$ in sequence $S_A$ or $B_j$ in $S_B$.

3.5 Behavior Transfer Unit
The behavior transfer unit (BTU) takes the representations from the source domain as input, extracts behavioral information, and transfers it to the target domain. It aims to output the transferred behavioral representation. The outputs $H_A$ and $H_B$ from the sequence encoder are representations of user behavior in single domains. It has been shown that there is connection between $H_A$ and $H_B$ [47]. For example, a user who has read the book Harry Potter (e.g., “Harry Potter and the Philosopher’s Stone” or “Harry Potter and the Chamber of Secrets” and so on) has also watched the movie “Pirates of the Caribbean” within the same time period. Based on behavioral information from both domains, it is easier for the model to infer that the user might like some magic and fantasy movies and books.

To achieve this, we employ the BTU to model the flow of behavioral information from domain $A$ to domain $B$, i.e., $h_{A_{i} \rightarrow B}$, as follows:

$$f_{Ai} = \sigma(W_{fa} \cdot h_{Ai} + W_{fb} \cdot h_{Bj} + W_{f} \cdot h_{A_{i-1} \rightarrow B} + b_f)$$

$$\hat{h}_{Ai} = \tanh(W_{h} \cdot h_{Ai} + U_{h} \cdot h_{A_{i-1} \rightarrow B} + b_h)$$

$$h_{A_{i} \rightarrow B} = f_{Ai} \odot \hat{h}_{Ai} + (1 - f_{Ai}) \odot h_{A_{i-1} \rightarrow B},$$

where $h_{Ai}$ and $h_{Bj}$ are the representations of domain $A$ and $B$ at timestamp $i$ and $j$, respectively. $f_{Ai}$ measures the degree of connection between these two representations $h_{A_i}$ and $h_{B_j}$ from both
domains, which employs the gate mechanism to control how much information is to be transferred from domain $A$ to domain $B$. $\hat{h}_{A_i}$ is the updated representation of the current input. $W_A$ and $W_B$ are the parameters, $b_f$ is the bias term, $\odot$ indicates element-wise multiplication. $h_{A_i \rightarrow B}$ can be seen as a combination of $\hat{h}_{A_i}$ and $h_{A_i-1 \rightarrow B}$ balanced by $f_{A_i}$. Note that the BTU can be applied bidirectionally from “domain $A$ to domain $B$” and “domain $B$ to domain $A$”. Here, we take the “domain $A$ to domain $B$” direction and achieve recommendations in domain $B$ as an example.

$h_{A_i \rightarrow B}$ is the information extracted from domain $A$, which is ready to be transferred to domain $B$. Since it still belongs to domain $A$, we employ an RNN structure to transfer it to domain $B$ as follows:

$$h_{(A \rightarrow B)_i} = \text{GRU}(h_{(A \rightarrow B)_{i-1}}, h_{A_i \rightarrow B}).$$

(4)

After that, we can obtain the transferred behavioral representation $h_{(A \rightarrow B)_i}$ in domain $B$ at time step $i$.

### 3.6 Knowledge Transfer Unit

The BTU only models the flow of behavioral information. We hypothesize that this may not be enough for the model to be able to encode the connection between items from the two domains in some cases. For example, if there is no knowledge indicating that both “Pirates of the Caribbean” and “Harry Potter” belong to magic and fantasy, it is difficult for the model to capture the connection between them solely based on behavioral information. To better transfer the information of items from both domains, we propose the **knowledge transfer unit** (KTU); it takes the item representation from sequence encoder and transferred behavioral representations from BTU as input and grasps useful knowledge from KG and propagates it to both domains, as shown in the Figure 4.

For each hybrid interaction sequence $S$, we get the item representations $\{h_{A_1}, h_{A_2}, \ldots, h_{A_i}, \ldots\}$ from the sequence encoder (Section 3.4) and the transferred behavioral representations $\{h_{(A \rightarrow B)_1}, h_{(A \rightarrow B)_2}, \ldots, h_{(A \rightarrow B)_i}, \ldots\}$ from the BTU (Section 3.5). We use the item representation of the last time step to denote the sequence representation $h_A$ and the transferred behavioral representation $h_{(A \rightarrow B)}$. We also obtain the relational adjacency matrix $A$ of the KG, which consists of $N$ entities...
and the corresponding relations (Section 3.2). Here, we represent these \( N \) entities as \( E \), the corresponding relations are represented as \( R \).

We initialize all entities in the KG and can get the initialized entity representations \( h_{E^0} \). That is, for each entity \( e_k \in E \), the initialized entity representation is \( h_{e_k^0} = h_{E^0} \). Then, we learn a transferred entity representation \( h_{e_k^L} \) for each entity \( e_k \) by leveraging the relations in the KG, as shown in the following equation:

\[
\begin{align*}
    r &= \sigma \left( W_r \cdot h_{e_k^0} + U_r \cdot \hat{h}_{e_k^L} + b_r \right) \\
    f &= \sigma \left( W_f \cdot h_{e_k^0} + U_f \cdot \hat{h}_{e_k^L} + V_f \cdot h_{A} + b_f \right) \\
    \hat{h}_{e_k} &= \tanh \left( W_h \cdot \hat{h}_{e_k^L} + U_h \odot \left( r \odot \hat{h}_{e_k^0} \right) \right) \\
    h_{e_k^L} &= (1 - f) \odot h_{e_k^0} + f \odot \hat{h}_{e_k}.
\end{align*}
\]

(5)

The explanations for the main parts of Equation (5) are as follows:

(i) **Gated functions.** \( r \) and \( f \) are the update gate function and the forget gate function, which aim at regulating how much of the update information should be propagated. \( W_r, U_r, W_f, U_f, \) and \( V_f \) are the parameters; \( b_r \) and \( b_f \) are the bias term.

(ii) **Candidate knowledge transfer representation.** \( \hat{h}_{e_k} \) is the candidate knowledge transfer representation, which is calculated based on the cross-domain disseminated entity representation \( \hat{h}_{e_k^L} \) at the \( L \)th hop layer (we will explain this later in Equation (6)) and the updated entity representation \( r \odot h_{e_k^0} \). \( W_h \) and \( U_h \) are the parameters, \( \odot \) indicates element-wise multiplication.

(iii) **Transferred entity representation.** The transferred entity representation \( h_{e_k^L} \) is a combination of the initialized entity representation \( h_{e_k^0} \) and the candidate knowledge transfer representation \( \hat{h}_{e_k} \) balanced by the forget gate \( f \), where the information among entities has been disseminated through \( L \) hop layers.

Graph convolutional techniques are commonly used to disseminate information among entities based on their relations [36, 73, 74]. However, in the cross-domain scenario that we consider, the information disseminated by entities from different domains is different. Hence, we propose a **cross-domain graph convolutional mechanism** that can distinguish entities from different domains and adopt different modeling methods to disseminate their information so as to get better entity representation. In this manner, the information in the KG is disseminated between both domains, which can be considered as a flow of knowledge in the hybrid interaction sequence. Suppose the information can be disseminated within \( L \) hop layers. At the \( l \)th hop layer \( (0 \leq l \leq L) \), the information of each entity and its cross-domain neighbor entities via various relations will be disseminated to the next hop layer \( l + 1 \) and is used to update the entity representation. The process of cross-domain information dissemination is defined in the following equation:

\[
\begin{align*}
    \tilde{h}_{e_k^{l+1}} &= \sigma \left[ f_0 \left( \hat{h}_{e_k^l} \right) + \frac{1}{|N_l(k)|} \sum_{r \in R} \sum_{p \in N_l(k)} f_i \left( \alpha_p \cdot \hat{h}_{e_p^l} \right) + \frac{1}{|N_c(k)|} \sum_{r \in R} \sum_{q \in N_c(k)} f_c \left( \beta_q \cdot \hat{h}_{e_q^l} + c_q \cdot \hat{h}_{e_q^l} \right) \right],
\end{align*}
\]

(6)

where \( e_k \) is any entity in the entity set \( E \); \( N_l(k) \) is the neighbor entity set of entity \( e_k \) from the same domain as \( e_k \); and \( N_c(k) \) is the neighbor entity set of entity \( e_k \) from the complementary domain. In
other words, if \( e_k \) belongs to \( A \) domain, \( N_i(k) \) is the neighbor entity set belonging to domain \( A \), and \( N_c(k) \) is the neighbor entity set belonging to domain \( B \). \( f_0, f_1, \) and \( f_c \) represent the transformation functions for initialization, in-domain, and cross-domain, respectively; \( \alpha_p \) is the in-domain attention weight calculated between each entity \( e_p \in N_i(k) \) and the sequence representation \( \psi_A; \beta_q \) is the cross-domain attention weight calculated between each entity \( e_q \in N_c(k) \) and the transferred behavioral representation \( \psi(A\rightarrow B); \) \( c_q \) shows the sum of similarities between entity \( e_q \) and each entity \( e_p \in N_i(k) \), which is defined as \( c_q = \sum_{p \in N_i(k)} e_p \cdot e_q; \) \( \psi_{c,k} \) denotes the cross-domain disseminated representation of entity \( e_k \) at the \( /th \) hop layer, which aggregates the information from itself and cross-domain neighbor entities as the new representation for the next hop layer. At the first hop layer, the cross-domain disseminated representation is assigned by the gated entity representation, i.e., \( \psi_{c,k} = \psi_{Ak} \) if the entity \( e_k \) is from domain \( A \), otherwise \( \psi_{c,k} = \psi_{Bk} \):

\[
c = \sigma \left( W_c \cdot \text{concat}[h_A, h_{(A\rightarrow B)}, h_{E_k}] + b_c \right)
\]
\[
h_{Ak} = c \odot \left( \alpha_{Ak} \cdot h_{\psi_{Ak}} \right)
\]
\[
h_{Bk} = (1-c) \odot \left( \beta_{Bk} \cdot h_{\psi_{Bk}} \right),
\]

where \( W_c \) is the parameter; \( b_c \) is the bias term; \( \psi_{Ak} \) is the gated entity representation of entity \( e_k \in E_A; \psi_{Bk} \) is the gated representation of entity \( e_k \in E_B \). \( c \) is a cross-domain information gate to handle the situation where the proportion of entities from different domains in the KG is different (e.g., when there are 1,000 entities from domain \( A \), but only 10 entities from domain \( B \)). So we define the gated entity representations \( h_{Ak} \) for domain \( A \) and \( h_{Bk} \) for domain \( B \) based on their initial entity representations, respectively, which aims at balancing information from both domains. \( \alpha_{Ak} \) is the attention weight of \( e_k \) for sequence representation \( h_A; \beta_{Bk} \) is the attention weight of \( e_k \) for the transferred behavioral representation \( h_{(A\rightarrow B)} \). These attention weights act as an information controller to identify entities of different importance in the KG, the definitions of which are shown in the following equation:

\[
\alpha_{Ak} = \text{softmax} \left( \mathbf{v}_1^T \tanh \left( W_{A1} \cdot h_A + W_{A2} \cdot h_{\psi_{Ak}} \right) \right)
\]
\[
\beta_{Bk} = \text{softmax} \left( \mathbf{v}_2^T \tanh \left( W_{B1} \cdot h_{(A\rightarrow B)} + W_{B2} \cdot h_{\psi_{Bk}} \right) \right),
\]

where \( h_{\psi_{Ak}} \) and \( h_{\psi_{Bk}} \) are the initialized entity representations of entity \( e_k \) and \( e_k \), respectively. \( h_A \) is the sequence representation and \( h_{(A\rightarrow B)} \) is the transferred behavioral representation as mentioned above. \( \mathbf{v}_1, \mathbf{v}_2, W_{A1}, W_{A2}, W_{B1}, \) and \( W_{B2} \) are learnable parameters.

### 3.7 Mode Switch

Recall that \( P(M_S\mid B, \langle E, R \rangle) \) and \( P(M_G\mid B, \langle E, R \rangle) \) are the probabilities of conducting recommendations under sequence mode and graph mode, respectively. We model the mode switch as a binary classifier. Specifically, we first combine the sequence representation \( h_B \), the transferred behavioral representation \( h_{(A\rightarrow B)} \) and the sum of all transferred entity representation \( \sum_k (h_{\psi_{c,k}}) \) (where \( h_{\psi_{c,k}} \in h_{E,T} \)). Then, we employ a softmax regression to transform the total representation into the mode probability distributions, as follows:

\[
P(M_G\mid B, \langle E, R \rangle), P(M_S\mid B, \langle E, R \rangle) = \text{softmax} \left( W_m \cdot \text{concat} \left[ h_B, h_{(A\rightarrow B)}, \sum_k (h_{\psi_{c,k}}) \right] + b_m \right),
\]
where $W_m$ is the weight matrix and $b_m$ is the bias term.

### 3.8 Graph Recommendation Decoder

The graph recommendation decoder evaluates the probabilities of recommending items involved in the KG. Here, we directly use the representations of entities to learn the attention weights and take these weights as the final predicted recommendation probability. The recommendation probability for each item $B_{j+1} \in E_B$ is computed as follows:

$$
P(B_{j+1}|M_{GB}, S, \langle E, R \rangle) = \begin{cases} 
0 & \text{if } B_{j+1} \notin B \\
\text{softmax} \left( h_{e_{B_{j+1}}}^T \right) & \text{if } B_{j+1} \in B,
\end{cases}
$$

(10)

where $h_{e_{B_{j+1}}}^T$ is the transferred entity representation of $e_{B_{j+1}}$ (corresponding to $h_{e_k}^T$ in Equation (5) when $e_k$ is an item entity from domain $B$). It is worth noting that this process only calculates the probability in the item set, and that the size of the item set $B$ is $m$. Note that $e_{B_{j+1}} \in E_B$ is an item entity corresponding to item $B_{j+1} \in B$. The recommendation probabilities are set to zero for those items that do not exist in $B$.

### 3.9 Sequence Recommendation Decoder

The sequence recommendation decoder evaluates the probabilities of items in the sequence item set. We first concatenate the sequence representation $h_B$ and the transferred behavioral representation $h_{(A \rightarrow B)}$ into the hybrid representation $c_S$, i.e., $c_S = [h_B, h_{(A \rightarrow B)}]^T$. Then, the recommendation probability for each item $B_{j+1} \in B$ is computed as follows:

$$
P(B_{j+1}|M_{SB}, S, \langle E, R \rangle) = \begin{cases} 
0 & \text{if } B_{j+1} \notin B \\
\text{softmax} \left( W_{j+1} c_S + b_I \right) & \text{if } B_{j+1} \in B,
\end{cases}
$$

(11)

where $W_{j+1} \in W_I$ is the weight matrix, $b_I$ is the bias term. The recommendation probabilities are set to zero for those items that do not exist in the item set $B$.

### 3.10 Objective Function

Our goal is to maximize the prediction probability for each domain given a hybrid interaction sequence. Therefore, we define the negative log-likelihood loss function as follows:

$$
L_R(\theta) = L_{RA}(\theta) + L_{RB}(\theta),
$$

(12)

where $\theta$ are all parameters in MIFN. Specifically, $L_{RA}(\theta)$ and $L_{RB}(\theta)$ can be derived as follows:

$$
L_{RA}(\theta) = -\frac{1}{|S|} \sum_{S \in \mathbb{S}} \sum_{A_i \in S} \log P(A_{i+1}|S, \langle E, R \rangle)
$$

$$
L_{RB}(\theta) = -\frac{1}{|S|} \sum_{S \in \mathbb{S}} \sum_{B_j \in S} \log P(B_{j+1}|S, \langle E, R \rangle),
$$

(13)

where $\mathbb{S}$ is the set of all hybrid interaction sequences in our training set, and $P(A_{i+1}|S, \langle E, R \rangle)$ or $P(B_{j+1}|S, \langle E, R \rangle)$ are the next prediction probabilities, which are as defined in Equation (2).

### 4 EXPERIMENTAL SETUP

#### 4.1 Research Questions

We evaluate MIFN on four e-commerce datasets. We aim at answering the following questions in our experiments:
Table 2. Dataset Statistics

<table>
<thead>
<tr>
<th>Domain</th>
<th>#Items</th>
<th>#Train</th>
<th>#Test</th>
<th>#Valid</th>
<th>Avg_Seq_Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie</td>
<td>36,845</td>
<td>44,732</td>
<td>19,861</td>
<td>9,274</td>
<td>11.98</td>
</tr>
<tr>
<td>Book</td>
<td>63,937</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Food</td>
<td>29,207</td>
<td>25,766</td>
<td>17,280</td>
<td>7,650</td>
<td>9.91</td>
</tr>
<tr>
<td>Kitchen</td>
<td>34,886</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(RQ1) How does MIFN perform compared with the state-of-the-art methods in terms of Recall and MRR? (See Section 5.1.)

(RQ2) Does the KTU help to improve the performance of recommendations? And does the performance differ from the situation when we only allow for the flow of behavioral information? (See Section 5.2.)

(RQ3) Does the knowledge graph construction method have a big effect on the overall recommendation results? (See Section 5.3.)

(RQ4) Is MIFN able to provide better recommendations by incorporating the flow of knowledge across domains? (See Section 5.4.)

(RQ5) Is the overall performance sensitive to the hyperparameters? (See Section 5.5.)

4.2 Datasets

We conduct experiments on the Amazon e-commerce collection,\(^3\) which consists of user interactions (e.g., userid, itemid, ratings, timestamps) from multiple domains and some item meta information (e.g., descriptions, images, product associations). Compared with other recommendation datasets, the Amazon dataset contains overlapping user interactions in multiple domains, which is suitable for cross-domain sequential recommendation (CDSR). Specifically, we pick two pairs of complementary domains “Movie-Book” domains and “Food-Kitchen” domains for experiments. For the “Movie-Book” dataset, the “Movie” domain contains movie watching records. The “Book” domain covers book reading records. For the “Food-Kitchen” dataset, the “Food” domain contains food purchase records. The “Kitchen” domain contains furniture purchase records. We follow the settings of Ma et al. \(^{47}\) to process the data. To satisfy cross-domain characteristics, we first pick users who have interactions in both domains. Since we do not target cold-start users or items in this work, we only keep users who have more than 10 interactions and items whose frequency is larger than 10. To satisfy sequential characteristics which consists of many user interactions within a period of time, we first order the interactions by time for each user, then we split the sequences from each user into several small sequences with each sequence containing interactions within a period, i.e., a month for the “Movie-Book” dataset, and a year for the “Food-Kitchen” dataset. We also require that each sequence contains at least three items from each domain. The statistics of the processed datasets are shown in Table 2.

For knowledge graph construction, we use the Amazon product data to mine the relations of the items.\(^4\) The data are collected from large-scale user logs, which contain the following relations: (1) “Also_buy” (users also buy item X when buying item Y); (2) “Also_view” (users also view item X when viewing item Y); (3) “Buy_together” (users buy item X and Y together frequently); (4) “Buy_after_viewing” (users buy item X after they buy Y); (5) “Is_the_same_category” (item X and Y belong to the same category). Additionally, for the “Movie-Book” dataset, we also

---

Table 3. KG Statistics

<table>
<thead>
<tr>
<th>Domain</th>
<th>#Entities</th>
<th>#Relations</th>
<th>#Triples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie</td>
<td>65,418</td>
<td>6</td>
<td>3,911,284</td>
</tr>
<tr>
<td>Book</td>
<td>315,770</td>
<td></td>
<td>40,048,795</td>
</tr>
<tr>
<td>Food</td>
<td>50,273</td>
<td>5</td>
<td>3,822,123</td>
</tr>
<tr>
<td>Kitchen</td>
<td>82,552</td>
<td></td>
<td>7,836,064</td>
</tr>
</tbody>
</table>

crawl the relation “Adapted_from” between books and movies from Wikipedia. For example, ⟨movie, Adapted_from, book⟩ means that the movie is adapted from the book. We align knowledge entities with Wikipedia titles by fully matching. The statistics of the knowledge information are given in Table 3.

For evaluation, we use the last interacted item in each sequence for each domain as the ground truth item, respectively. We randomly select 80% of each user’s interactions as the training set, 10% as the validation set, and the remaining 10% as the test set.

4.3 Baseline Methods

We compare the proposed model MIFN with baselines from four categories: (1) traditional recommendation methods, (2) SR methods, (3) cross-domain recommendation methods, (4) and knowledge-aware recommendation methods.

4.3.1 Traditional Recommendation Methods. We adapt three commonly used traditional recommendation methods to SRs:

— POP: This method recommends the most popular items in which items are ranked based on their popularity. It is a simple baseline, but is commonly used owing to its simplicity yet effectiveness [26].

— Item-KNN: This method is inspired by the classical KNN model; it looks for items that are similar to other items that have been clicked by a user in the past, where similarity is defined as the cosine similarity between the vector of sequences [64].

— BPR-MF: This method follows the idea of MF with a pairwise ranking objective via stochastic gradient descent [61]. Following Hidasi et al. [27], we represent a new sequence by the average latent factors of items that appeared in the sequence so far.

4.3.2 SR Methods. A number of SR methods have been proposed in the last few years. In this work, we construct/select baselines that are fair (use the same information, similar architectures, etc.) to compare with:

— GRU4REC: It is the first attempt to use GRU for SRs. It utilizes session-parallel mini-batch training strategy and employs a ranking-based loss functions [27].

— HRNN: This method combines the extra user’s information into GRU networks and proposes a hierarchical RNN model based on GRU4REC [57].

— NARM: This method takes an attention mechanism into consideration to capture both sequential-level preferences and the user’s main purpose [40].

— STAMP: This method constructs two network structures to capture a user’s general preferences and the current preferences of the last click within the current sequence [45].

— SRGNN: This method constructs each sequence as a directed graph, where the items in the sequence are entities and the transition relationship between adjacent items represents edge.

5https://en.wikipedia.org/.
By modeling the complex transitions, each session graph and item embeddings of all items involved in each graph can be obtained through gated graph neural networks [82].

### 4.3.3 Cross-Domain Recommendation Methods

We use three popular cross-domain recommendation methods for comparison.

- **NCF-MLP++**: This is a deep learning based method where the model learns the inner product of the traditional CF by using **multilayer perceptron (MLP)** in each domain. The user representations are shared between both domains while item representations are private in each domain, and the final recommendations are aggregated from both domain recommendations probabilities. We adopt the implementation in [47].

- **Conet**: This method transfers information between different domains by a cross-stitch network [54], where information in each domain is captured by neural CF model [26].

- **π-Net**: This method is the only one that considers cross-domain characteristics for SRs. We take this as the fairest baseline to compare with. It designs a new gating mechanism to recurrently extract and share useful information across different domains [47].

### 4.3.4 Knowledge-Aware Recommendation Methods

As knowledge-aware recommendation methods, we choose two baseline methods.

- **KGAT**: This method explicitly models the high-order connectivities in KG in an end-to-end fashion. It recursively propagates the embeddings from an entity’s neighbors (which can be users, items, or attributes) to refine the entity’s embedding, and employs an attention mechanism to discriminate the importance of the neighbors [78].

- **KSR**: This method incorporates KGs into SRs, and it combines the sequential user preference captured by an RNN and attribute-level preferences captured by Key-Value Memory Networks to get the final representation of user preference [35].

### 4.4 Evaluation Metrics

We target the top-K recommendations in this work, so we adopt two widely used ranking-based metrics [14, 25, 51, 58]: MRR@K and Recall@K. Specifically, we report $K = 5, 10, 20$.

- **Recall**: This measures the proportion of the top-K recommended items that are in the evaluation set. It does not consider the actual rank of the item as long as it is amongst the list of recommend items.

- **MRR**: This is the average of reciprocal ranks of the relevant items. And the reciprocal rank is set to zero if the ground truth item is not in the list of recommended items. MRR takes the order of recommendation ranking into account. Since each sample has only one ground truth item, we choose MRR as the ranking metric instead of others, e.g., NDCG.

### 4.5 Implementation Details

For most of the baseline methods, we find the best settings using grid search on the validation set. For those with too many hyperparameters, we follow the reported optimal hyperparameter settings from the original publications that introduced them. For our model, the embedding size and the hidden size are set to 256, and the number of entities $N$ in KG is set to 200. As for the parameter hop layer $H$, we conduct some preliminary experiments and find that when the hop layer is limited to 1, most of the sequences cannot be connected, but when the hop layer increases to 2, around 75% of the sequences can be connected, and when it continues to increase to 5 hops even 7 hops, there is not much growth, so we limit parameter hop layer to 2. We initialize the model parameters randomly using the Xavier method. We take Adam as our optimization method.
Mixed Information Flow for Cross-domain Sequential Recommendations

is implemented in TensorFlow and trained on a GeForce GTX TitanX GPU. The code and dataset used to run the experiments in this article are available at https://github.com/mamuyang/MIFN.

4.6 Discussion

In addition to calculating the prediction probability, we also explore the mode switch module to calculate the mode selection probability between sequence mode and graph mode. We assume that if an item does not exist in the item set, it can just be generated under the graph mode. Here, we can jointly train another mode prediction loss as follows, which adopts the negative log-likelihood loss:

\[ L_{M_A}(\theta) = -\frac{1}{|S|} \sum_{S \in \mathcal{S}} \sum_{A_i \in S} \left[ (1 - \mathbb{1}(A_{i+1} \in \mathcal{A})) \log P(M_{-S_A}|S, \langle E, R \rangle) \right] \]

\[ L_{M_B}(\theta) = -\frac{1}{|S|} \sum_{S \in \mathcal{S}} \sum_{B_j \in S} \left[ (1 - \mathbb{1}(B_{j+1} \in \mathcal{B})) \log P(M_{-S_B}|S, \langle E, R \rangle) \right] \]

\[ L_M(\theta) = L_{M_A}(\theta) + L_{M_B}(\theta), \tag{14} \]

where \( \mathbb{1}(item \in itemset) \) is the indicator function that equals 1 if this item is in the item set and 0 otherwise. \( L_M(\theta) \) is the total mode loss for domain \( A \) and \( B \).

Finally, we adopt a joint-learning strategy, and the final loss combines both recommendation loss and mode loss:

\[ L(\theta) = L_R(\theta) + L_M(\theta). \tag{15} \]

Experimental results for this module are shown in Tables 4 and 5, where MIFN represents the model without mode prediction loss, MIFN+L_M represents the model with this mode prediction loss. An analysis of these results is provided in Section 5.2.

5 RESULTS AND ANALYSIS

5.1 Overall Performance (RQ1)

We report the results of MIFN compared with the baseline methods on the “Movie-Book” and “Food-Kitchen” datasets. The results of all methods are given in Tables 4 and 5, respectively. From the results, we have the following main observations.

First, MIFN outperforms some single-domain SR methods (e.g., STAMP, NARM, SRGNN) and the knowledge-aware methods (KSR) on all datasets. Particularly, on the “Movie-Book” dataset, the largest increase over NARM is 7.5% and 12.4% in terms of MRR@5 and Recall@10 on the “Movie” domain, and on the “Book” domain, the largest increase is 6.2% and 9.1% in terms of MRR@20 and Recall@5. And the increase over KSR on the “Movie” domain is 4.6% and 6.9% in terms of MRR@5 and Recall@10, on the “Book” domain, the increase is 2.1% and 2.9% in terms of MRR@20 and Recall@5. On the “Food-Kitchen” dataset, the largest increase over NARM is 6.5% and 9.3% in terms of MRR@20 and Recall@20 on the “Food” domain, and on the “Kitchen” domain, the increase is 9.1% and 12.3% in terms of MRR@5 and Recall@10. And the increase over KSR is 2.7% and 3.9% in terms of MRR@20 and Recall@10 on the “Food” domain, and on the “Kitchen” domain, the increase is 1.6% and 4.1% in terms of MRR@20 and Recall@20. These improvements demonstrate that jointly considering both cross-domain behavior and knowledge is helpful for SR.

Second, MIFN outperforms the cross-domain recommendation baseline \( \pi \)-Net, which just makes use of information at user behavior level. Specifically, MIFN outperforms \( \pi \)-Net in terms of all metrics on both domains. It demonstrates that considering both knowledge and user behavior level information is better than only behavioral information. Meanwhile, it also proves the effectiveness...
Table 4. Experimental Results (%) on the Amazon ("Movie-Book") Dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>Movie-domain</th>
<th></th>
<th></th>
<th>Book-domain</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR @5</td>
<td>MRR @10</td>
<td>MRR @20</td>
<td>Recall @5</td>
<td>Recall @10</td>
<td>Recall @20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POP</td>
<td>0.10</td>
<td>0.11</td>
<td>0.13</td>
<td>0.20</td>
<td>0.29</td>
<td>0.58</td>
</tr>
<tr>
<td>BPR-MF</td>
<td>0.51</td>
<td>0.57</td>
<td>0.64</td>
<td>0.89</td>
<td>1.35</td>
<td>2.26</td>
</tr>
<tr>
<td>ItemKNN</td>
<td>1.05</td>
<td>1.27</td>
<td>1.48</td>
<td>2.11</td>
<td>3.84</td>
<td>6.99</td>
</tr>
<tr>
<td>GRU4REC</td>
<td>12.80</td>
<td>12.86</td>
<td>12.88</td>
<td>13.69</td>
<td>14.11</td>
<td>14.43</td>
</tr>
<tr>
<td>STAMP</td>
<td>12.44</td>
<td>12.56</td>
<td>12.63</td>
<td>13.66</td>
<td>14.58</td>
<td>15.68</td>
</tr>
<tr>
<td>SRGNN</td>
<td>11.77</td>
<td>11.84</td>
<td>11.88</td>
<td>12.66</td>
<td>13.18</td>
<td>13.87</td>
</tr>
<tr>
<td>NCF-MLP++</td>
<td>1.64</td>
<td>1.86</td>
<td>2.03</td>
<td>2.95</td>
<td>4.61</td>
<td>7.24</td>
</tr>
<tr>
<td>Conet</td>
<td>1.43</td>
<td>1.73</td>
<td>2.01</td>
<td>2.83</td>
<td>5.20</td>
<td>9.24</td>
</tr>
<tr>
<td>π-Net</td>
<td>14.49</td>
<td>14.52</td>
<td>14.54</td>
<td>14.88</td>
<td>15.10</td>
<td>15.37</td>
</tr>
<tr>
<td>KGAT</td>
<td>4.16</td>
<td>4.35</td>
<td>4.48</td>
<td>5.81</td>
<td>7.25</td>
<td>9.18</td>
</tr>
<tr>
<td>KSR</td>
<td>14.18</td>
<td>14.23</td>
<td>14.26</td>
<td>14.91</td>
<td>15.28</td>
<td>15.73</td>
</tr>
<tr>
<td>MIFN-linear</td>
<td>14.00</td>
<td>14.07</td>
<td>14.10</td>
<td>14.88</td>
<td>14.91</td>
<td>15.22</td>
</tr>
<tr>
<td>MIFN+LM</td>
<td>14.73</td>
<td>14.75</td>
<td>14.81</td>
<td>14.87</td>
<td>15.96</td>
<td>16.02</td>
</tr>
<tr>
<td>MIFN</td>
<td><strong>14.84</strong></td>
<td><strong>14.87</strong></td>
<td><strong>14.88</strong></td>
<td><strong>15.13</strong></td>
<td><strong>16.34</strong></td>
<td><strong>16.56</strong></td>
</tr>
</tbody>
</table>

**Bold face** indicates the best result in terms of the corresponding metric. Significant improvements over the results of the baseline (KSR) which performs best on most metrics are marked with ° (t-test, \( p < .05 \)).

of the KTU module of MIFN. With this module, MIFN is able to capture cross-domain knowledge and conduct information transfer in the KG so as to learn better sequence representations.

Third, we can observe that the results of MIFN in the “Book” domain are better than those in the “Movie” domain on the “Movie-Book” dataset. We believe that this is because the data are less sparse in the “Book” domain compared to the “Movie” domain. There is more space for improvement in the “Book” domain. With more interaction data, the models can identify more user preference characteristics in the dense domain so as to transfer it to the sparse domain through both the user behavioral information flow and the knowledge information flow. At the same time, we also find that the results in the “Food” domain are better than the “Kitchen” domain on the “Food-Kitchen” dataset. The final performance is not only related to the difference in sparsity between the datasets, but also to the characteristics of the domain and the degree of correlation between different domains.

Fourth, π-Net outperforms other sequential baselines, which means that cross domain information is beneficial to both domains. At the same time, knowledge aware methods outperform other sequential baselines, which also means that knowledge information can improve recommendation performance. Furthermore, it seems that considering knowledge is more useful than modeling cross-domain characteristics, as KSR slightly outperforms π-Net.

Fifth, the sequential methods achieve much better results in general. This is because RNN-based methods are able to capture the sequential characteristics and can obtain the better representations while the traditional methods neglect this information. What’s more, it seems that STAMP obtains lower results than the sequential method NARM in the “Movie-Book” dataset, while it performs...
Table 5. Experimental Results (%) on the Amazon (“Food-Kitchen”) Dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>Food-domain recommendation</th>
<th>Kitchen-domain recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR @5 @10 @20</td>
<td>MRR Recall @5 @10 @20</td>
</tr>
<tr>
<td>POP</td>
<td>0.40 0.49 0.56</td>
<td>0.82 0.87 0.94</td>
</tr>
<tr>
<td>BPR-MF</td>
<td>0.82 0.87 0.94</td>
<td>1.35 1.82 2.61</td>
</tr>
<tr>
<td>ItemKNN</td>
<td>1.55 1.98 2.43</td>
<td>3.28 6.70 12.47</td>
</tr>
<tr>
<td>GRU4REC</td>
<td>8.10 8.23 8.29</td>
<td>9.46 10.38 11.26</td>
</tr>
<tr>
<td>NARM</td>
<td>9.43 9.54 9.62</td>
<td>10.34 11.86 12.23</td>
</tr>
<tr>
<td>SRGNN</td>
<td>7.31 7.49 7.60</td>
<td>8.68 10.02 11.67</td>
</tr>
<tr>
<td>NCF-MLP++</td>
<td>2.01 2.24 2.42</td>
<td>3.45 5.19 8.02</td>
</tr>
<tr>
<td>Conet</td>
<td>3.38 3.64 3.82</td>
<td>5.07 7.07 9.73</td>
</tr>
<tr>
<td>KGAT</td>
<td>4.74 4.97 5.09</td>
<td>6.49 8.18 9.94</td>
</tr>
<tr>
<td>MIFN</td>
<td><strong>9.91</strong> 10.16 10.25</td>
<td><strong>11.20</strong> 12.25 13.27</td>
</tr>
</tbody>
</table>

**Bold face** indicates the best result in terms of the corresponding metric. Significant improvements over the results of the baseline (KSR) which performs best on most metrics are marked with † (t-test, \( p < .05 \)).

Better performance in the “Kitchen” domain of “Food-Kitchen” dataset. We believe that this is because of differences in the datasets, e.g., we found that the user preferences in the kitchen domain are relatively more focused. And the method SRGNN performs worse than most sequential methods. This is because that SRGNN just employs the transition relation between adjacent items to construct the graph so as to get the representations of different items, however it does not consider the relation between the non-adjacent items (the other sequential methods do consider this), which may also affect item representations.

5.2 Ablation Study (RQ2)

To verify the effectiveness of the proposed modules, we design the ablation study to compare several model variants. The results are listed in Tables 4 and 5.

(1) MIFN is the best performing variant; this variant includes both the BTU and KTU modules, and is trained with the recommendation loss \( L_R \) only.

(2) MIFN-KTU is MIFN without the KTU module and performs information transfer only at the level of behavioral information;

(3) MIFN-linear is MIFN with the KTU module but it replaces the GRU propagation function with a linear mapping layer;

(4) MIFN+LM is MIFN by adding the mode switch loss.

First, by removing KTU, the performance of MIFN-KTU is dramatically less than that of MIFN, which confirms that considering knowledge flow can improve the cross-domain recommendations. In addition, the results of MIFN-KTU are worse than those of the knowledge-aware method KSR.
The different ratios represent the different proportions of the predicted ground truth items appeared in the KG.

5.3 Influence of the Knowledge Graph Construction Algorithm (RQ3)

The number of triples in the complete KG is large (see Table 3), so we propose a knowledge graph construction method as detailed in Algorithm 1 to build KG for each sequence. To study the effect of the knowledge graph construction method, we design an experiment aimed at analyzing the effect of the ratios of the ground truth items in the constructed KG on the final recommendation performance. We achieve this by simulating and controlling the ratios artificially. Specifically, we add the ground truth items to the extracted entities according to the specified ratios in advance. The results are given in Tables 6 and 7.

First, we can see that the performance increases as the ratio of ground truth item appeared in the KG increases on both datasets. For instance, when the ratio varies from 30% to 100%, the value of MRR@20 increases from 16.99% to 67.69% in the “Movie” domain and the value of Recall@20 increases from 22.41% to 95.45% in the “Kitchen” domain. This demonstrates that a good knowledge graph construction algorithm is of vital importance.

### Table 6. Analysis of the Knowledge Graph Construction Algorithm on the Amazon (“Movie-Book”) Dataset

<table>
<thead>
<tr>
<th>Ratios</th>
<th>Movie-domain recommendation</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR</td>
<td>Recall</td>
<td>MRR</td>
<td>Recall</td>
<td>MRR</td>
<td>Recall</td>
<td>MRR</td>
<td>Recall</td>
<td>MRR</td>
<td>Recall</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>@5</td>
<td>@10</td>
<td>@20</td>
<td>@5</td>
<td>@10</td>
<td>@20</td>
<td>@5</td>
<td>@10</td>
<td>@20</td>
<td>@5</td>
<td>@10</td>
<td>@20</td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>19.53</td>
<td>20.43</td>
<td>20.83</td>
<td>30.33</td>
<td>37.07</td>
<td>42.67</td>
<td>25.04</td>
<td>25.68</td>
<td>25.98</td>
<td>36.08</td>
<td>40.75</td>
<td>45.12</td>
<td></td>
</tr>
<tr>
<td>70%</td>
<td>23.41</td>
<td>23.43</td>
<td>23.45</td>
<td>42.54</td>
<td>47.99</td>
<td>50.61</td>
<td>27.12</td>
<td>28.31</td>
<td>28.41</td>
<td>48.93</td>
<td>56.64</td>
<td>57.31</td>
<td></td>
</tr>
<tr>
<td>90%</td>
<td>33.45</td>
<td>34.99</td>
<td>35.09</td>
<td>65.81</td>
<td>76.83</td>
<td>77.97</td>
<td>40.01</td>
<td>40.86</td>
<td>43.00</td>
<td>79.83</td>
<td>80.50</td>
<td>83.86</td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>67.04</td>
<td>67.56</td>
<td>67.69</td>
<td>89.57</td>
<td>93.30</td>
<td>95.15</td>
<td>83.08</td>
<td>83.46</td>
<td>83.62</td>
<td>91.97</td>
<td>94.79</td>
<td>97.02</td>
<td></td>
</tr>
</tbody>
</table>

While MIFN outperforms KSR on all domains. This indicates that the KTU module is able to make good use of the cross-domain knowledge and is able to better capture user preferences by modeling the cross-domain knowledge flow.

Second, by replacing the GRU propagation function (Equation (5)) with a linear mapping layer, the performance of MIFN-linear drops a little compared with the full model MIFN. This demonstrates that the GRU propagation is also useful for the combination of cross-domain knowledge for CDSR. At the same time, we find that the MIFN-linear outperforms MIFN-KTU in terms of most metrics. This indicates that the cross-domain graph convolution mechanism without GRU propagation is also capable of capturing cross-domain knowledge information flow. But the performance of MIFN-linear is still lower than the KSR in terms of most metrics in both datasets and MIFN outperforms KSR in terms of all metrics of both datasets. This again shows that GRU propagation is more effective for cross-domain knowledge representation.

Third, if we jointly train the recommendation loss $L_R$ and the mode switch loss $L_M$, the performance drops a little but its performance is still than that of the baselines. The switch loss $L_M$ assumes that if the next item does not exist in the item set, it must be recommended under the graph mode, which makes the model tend to recommend items existing in the graph. However, since there already exists a similar supervision signal in $L_R$, which assumes that each item is recommended under the graph mode, the sequence mode, or a combination of both. Further adding the $L_M$ loss introduces unnecessary bias toward graph mode.
Table 7. Analysis of the Knowledge Graph Construction Algorithm on the Amazon (“Food-Kitchen”) Dataset

<table>
<thead>
<tr>
<th>Ratios</th>
<th>Food-domain recommendation</th>
<th>Kitchen-domain recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR @5</td>
<td>MRR @10</td>
</tr>
<tr>
<td>30%</td>
<td>11.75</td>
<td>12.30</td>
</tr>
<tr>
<td>50%</td>
<td>22.14</td>
<td>22.61</td>
</tr>
<tr>
<td>70%</td>
<td>26.01</td>
<td>26.74</td>
</tr>
<tr>
<td>90%</td>
<td>36.58</td>
<td>37.50</td>
</tr>
<tr>
<td>100%</td>
<td>47.40</td>
<td>48.51</td>
</tr>
</tbody>
</table>

The different ratios represent the different proportions of the predicted ground truth items appeared in the KG.

Second, we notice that when the ratio increases linearly, the results do not increase linearly. On both datasets, the increase is relatively slow from 30% to 90%. However, when we simulate the ratio from 90% to 100%, the performance is greatly improved in terms of both MRR and Recall on the “Movie-Book” dataset. We believe that this is because when the ratios reaches a certain value, the model can easily capture the characteristics of recommended items from the graph mode in most cases and relies mostly on the KG to do recommendations. We also notice that with 100% ratio, Recall is improved largely on “Food-Kitchen” dataset while MRR is not. We think this is because of the density of the KG. As shown in Table 3, there are more triples in the “Movie-Book” dataset, especially the “Book” domain. The richer knowledge makes it relatively easier to rank the ground truth items.

Currently, using Algorithm 1, the ratios of the ground truth items in the KG of the “Movie-Book” and “Food-Kitchen” datasets are 14% and 12%, respectively, both of which are relatively low. Algorithm 1 is only based on the entity distance calculated using the pretrained entity representations in the KG, which is insufficient. To this end, we think an important future research direction in order to further improve MIFN is to design a more effective knowledge graph construction algorithm.

5.4 Qualitative Analysis with Case Studies (RQ4)

To analyze the recommendation results with and without a flow of knowledge, we list some examples from the “Movie-Book” dataset. Figure 5 shows recommendations when the extracted KG is relevant to the current user preference, and Figure 6 shows recommendations when the extracted KG is irrelevant. Figures 5(a) and 6(a) are recommendations from MIFN, while Figures 5(b) and 6(b) are recommendations from MIFN-KTU. In each figure, the orange color represents the interactions in the “Movie” domain, and the blue color represents the interactions in the “Book” domain. The meaning of the different colored fonts and lines are explained in the legend. The green tick indicates that the recommendation is correct, and the red cross indicates it is wrong.

From Figure 5, we can observe that when using the extracted KG (Figure 5(a)), MIFN can give correct recommendations for both domains, however, the recommendation is wrong in the “Book” domain when the KG is not used (Figure 5(b)). Furthermore, it should be noted that the mode switch probabilities in Figure 5(a) are different for the two domains. The probability of sequence mode is 1.0 in the “Movie” domain, but in the “Book” domain, the probability of graph mode is 0.75, which means that the recommendation of the “Movie” domain comes from the item set, while the recommendation mostly relies on the KG for the “Book” domain. From the KG, the book entity
“One Fine Stooge: Larry Fine’s Frizzy Life In Pictures” gets the highest recommendation score of 0.65 in the “Book” domain. The reason is that there exists a knowledge triple, that is, people who watch the movie “The Three Stooges Go Around the World in a Daze” will also view the book “One Fine Stooge: Larry Fine’s Frizzy Life In Pictures.” This knowledge is well transferred by the flow of knowledge to obtain a better recommendation for the “Book” domain. In contrast, as shown in Figure 5(b), the recommendation is wrong because the model only relies on the flow of behavioral information to recommend items from the item set.
On the other hand, we also found that the flow of knowledge is not always helpful. As shown in Figure 6(a), MIFN still gives a wrong recommendation even by modeling knowledge information flow with KTU. MIFN recommends “The confucian Transformation of Korea” with the graph mode recommendation probability of 0.77, which means, in this case, it still relies mostly on the KG to do recommendation. However, we can observe that most of the extracted entities in the KG are thriller movies or reference books which are not relevant to the current user preference. As a result, MIFN performs somewhat worse than MIFN-KTU, because although MIFN-KTU also recommends the wrong item, the recommended item seems more relevant to the user preference,
Table 8. Analysis of the Number of Entities $N$ on the Amazon (“Movie-Book”) Dataset

<table>
<thead>
<tr>
<th>$N$</th>
<th>Movie-domain recommendation</th>
<th>.Book-domain recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR @5</td>
<td>Recall @5</td>
</tr>
<tr>
<td>100</td>
<td>14.38</td>
<td>14.43</td>
</tr>
<tr>
<td>150</td>
<td>14.50</td>
<td>14.51</td>
</tr>
<tr>
<td>180</td>
<td>14.70</td>
<td>14.73</td>
</tr>
<tr>
<td>200</td>
<td>14.84</td>
<td>14.87</td>
</tr>
</tbody>
</table>

Different $N$ represents the different number of entities of the extracted KG.

Table 9. Analysis of the Number of Entities $N$ on the Amazon (“Food-Kitchen”) Dataset

<table>
<thead>
<tr>
<th>$N$</th>
<th>Food-domain recommendation</th>
<th>Kitchen-domain recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MRR @5</td>
<td>Recall @5</td>
</tr>
<tr>
<td>100</td>
<td>9.56</td>
<td>9.77</td>
</tr>
</tbody>
</table>

Different $N$ represents the different number of entities of the extracted KG.

i.e., “children|family.” This suggests that the quality of the extracted KG has a large impact on the final recommendation performance, which further verifies the conclusion in Section 5.3.

5.5 Hyperparameters Analysis (RQ5)

We conduct experiments to investigate the impact of the hyper-parameters in the MIFN framework, including the number of entities $N$ and the maximum hop count $H$.

The number of entities $N$ is used to limit the size of the KG during KG construction. The results of MIFN for different numbers of $N$ are shown in Tables 8 and 9, with $N$ ranging within $[50, 100, 150, 180, 200]$ for the “Movie-Book” dataset and $[100, 150, 170, 180, 200]$ for the “Food-Kitchen” dataset. The maximum hop count $H$ is also used to limit the KG size. We show the results in Tables 10 and 11 when $H = 1, 2, 3, 5$ for both datasets. From these results, we can observe that when values of $N$ and $H$ drop, the performance does not decrease dramatically. This demonstrates that MIFN is not sensitive to both hyperparameters.

6 Conclusion and Future Work

In this article, we study how to incorporate knowledge into the cross-domain sequential recommendation task. We present MIFN, which jointly models two types of information flow across domains, i.e., of behavioral information and of knowledge. To verify the effectiveness of MIFN, we conduct experiments on datasets from four Amazon domains. The results demonstrate that MIFN outperforms other state-of-the-art baselines. Through extensive analysis experiments, we confirm
that the flow of knowledge helps improve the recommendation performance in general, which means mixed flow of information can be used to enhance the CDSR performance.

There are also some limitations in this work. Currently the knowledge graph construction algorithm used to extract graphs is simple. So the quality of extracted graph is limited, which we show can be made more effective by including as much relevant knowledge as possible. And we only focus on the applications of MIFN to the scenario with two domains, which can be extended to multiple domains simultaneously. What’s more, some prior correlations are required between different domains. Now we assume that the users are overlapped between different domains during modeling the information flow, which limits the cross-domain research.

As to future work, MIFN can be enhanced in at least four directions correspondingly. First, we will further study knowledge graph construction algorithms to improve the quality of the KG without increasing its size. Second, MIFN is limited to information flow between two domains in this work. Therefore, we want to study how to make cross-domain recommendations across multiple domains. Third, it is worth trying to study the cross-domain transfer learning so as to improve the performance of both domains when there is no relationship between them. Fourth, we have only experimented on e-commerce datasets from Amazon as other datasets that meet our requirements do not seem to be available. We hope our work can inspire other researchers working on different recommendation scenarios.

**DATA AND CODE**

To facilitate reproduction of the results in the article, we are sharing the code and resources used to produce our results at https://github.com/mamuyang/MIFN.
REFERENCES


ACM Transactions on Knowledge Discovery from Data, Vol. 16, No. 4, Article 64. Publication date: January 2022.


[90] Wayne Xin Zhao, Jimpeng Wang, Yulan He, Ji-Rong Wen, Edward Y. Chang, and Xiaoming Li. 2016. Mining product adopter information from online reviews for improving product recommendation. ACM Transactions on Knowledge Discovery from Data 10, 3 (2016), 1–23.


Received December 2020; revised July 2021; accepted September 2021